

**Universidad Empresarial Siglo 21**

*Carrera de Licenciatura en Informática*

*Trabajo Final de Graduación*

---

**Título del Trabajo**

*Sistema Informático de Apoyo a la Lecto-Escritura para Niño con Discapacidad  
Múltiple*

**Tipo de Trabajo**

*Proyecto de Aplicación Profesional*

**Fecha Presentación**

*Abril de 2005*

**Autor**

Diego Gonzalo Obregón - (Leg: INF 159)

---

## *Agradecimientos*

---

Fue ardua la tarea, muchas las horas trasnochadas...

Agradezco a mi querida familia: la paciencia y el aguante...;

a la Universidad y a mis tutores: la dirección en el proceso...;

a los profesores Jorge Cassi y Techy Reyna: por permitirme enriquecer la versión final de este trabajo con sus aportes y recomendaciones;

al Ing. Diego Beltramone: por su colaboración en los aspectos electrónicos;

y al Señor Joseph Katz, de SoftVoice: la desinteresada colaboración, apostando a un proyecto tan lejano, con el sólo afán de colaborar para mejorar la calidad de vida de otro ser humano!

# Índice General

---

<b>PRIMERA PARTE</b> .....	<b>1</b>
<b>1. CONSIDERACIONES GENERALES DEL PRESENTE TRABAJO</b> .....	<b>2</b>
1.1. TEMA DEL TRABAJO .....	2
1.2. SOBRE EL TÍTULO DEL PRESENTE TRABAJO FINAL .....	2
1.3. SOBRE LA DENOMINACIÓN UTILIZADA: PERSONAS CON DISCAPACIDAD .....	2
<b>2. INTRODUCCIÓN GENERAL</b> .....	<b>8</b>
2.1. EL USUARIO .....	9
2.2. ANTECEDENTES .....	10
2.3. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA .....	11
2.4. FORMULACIÓN DEL PROBLEMA .....	12
2.5. JUSTIFICACIÓN.....	12
2.5.1. <i>Disponibilidad, Alcance y Posibilidades de soluciones existentes</i> .....	12
2.5.2. <i>Posibilidades de desarrollo superadoras para el sujeto</i> .....	13
2.5.3. <i>Aporte tecnológico</i> .....	13
<b>3. OBJETIVOS</b> .....	<b>14</b>
3.1. OBJETIVOS ESPECÍFICOS DETALLADOS.....	14
3.2. LÍMITE Y ALCANCE .....	15
<b>4. MARCO REFERENCIAL TEÓRICO</b> .....	<b>16</b>
4.1. PROBLEMÁTICA EDUCATIVA EN NIÑOS NO VIDENTES.....	17
4.2. DESARROLLOS TECNOLÓGICOS EXISTENTES EN EL CAMPO EN ESTUDIO .....	21
4.3. ELEMENTOS TECNOLÓGICOS DISPONIBLES PARA EL DESARROLLO .....	24
4.4. EL CICLO DE VIDA DEL PROYECTO Y LOS MODELOS DE A+D.....	27
<b>5. DIAGNÓSTICO Y PROPUESTA</b> .....	<b>30</b>
5.1. DIAGNÓSTICO .....	30
5.2. PROPUESTA.....	31
5.3. CONSIDERACIONES ADICIONALES .....	32

<b>6. ESTRATEGIA METODOLÓGICA .....</b>	<b>34</b>
6.1. CICLO DE VIDA.....	34
6.2. METODOLOGÍA DE ANÁLISIS Y DISEÑO DE SISTEMAS.....	36
6.3. DISEÑO DE INTERFACES DE USUARIO .....	37
6.4. MODELO DE PROTOTIPADO.....	40

## **SEGUNDA PARTE.....42**

### **7. ESTRUCTURA DE DESARROLLO.....43**

### **8. INTERFAZ FÍSICA.....44**

8.1. POSIBILIDADES MOTRICES Y SENSORIALES DE LUCAS .....	44
8.2. PORTABILIDAD, MATERIALES Y DISPOSICIÓN FÍSICA.....	45
8.3. APLICABILIDAD A LAS CARACTERÍSTICAS DEL SOFTWARE A DESARROLLAR.....	45
8.4. EXPERIENCIAS PREVIAS DEL SUJETO .....	46
8.5. CONCLUSIÓN Y DISEÑO FINAL DE LA INTERFAZ FÍSICA.....	47

### **9. INTERFAZ ELECTRÓNICA Y COMUNICACIÓN CON EL SOFTWARE .....56**

9.1. PRIMER PROTOTIPO: INTERFAZ ELECTRÓNICA VIA TECLADO. ....	57
9.1.1. <i>Implementación</i> .....	58
9.2. PROTOTIPO FINAL: INTERFAZ VÍA PUERTO COM.....	60
9.3. MECANISMO ANTIRREBOTE Y ANTIRREPETICIÓN .....	60
9.3.1. <i>Parámetros de configuración Puerto COM</i> .....	62
9.3.2. <i>Protocolo definido para la comunicación entre el dispositivo y el software</i> ...	63

### **10. SÍNTESIS DE VOZ POR SOFTWARE.....65**

10.1. FONACIÓN POR VOCABLOS ALMACENADOS PREDEFINIDOS .....	65
10.2. FONACIÓN POR COMBINACIÓN SILÁBICA .....	66
10.3. FONACIÓN POR SÍNTESIS MATEMÁTICA DE FONEMAS .....	67
10.4. PARTICULARIDADES REQUERIDAS PARA EL APLICATIVO.....	69
10.5. IMPLEMENTACIÓN TECNOLÓGICA DEL MOTOR.....	69

10.6.	DISPONIBILIDAD .....	69
10.7.	CONTROL APLICATIVO .....	71
<b>11.</b>	<b>MOTORES DE PALABRA PRESELECCIONADOS .....</b>	<b>74</b>
11.1.	BÚSQUEDA, PRESELECCIÓN.....	74
11.2.	COMPARATIVA DE CARACTERÍSTICAS DE LOS MOTORES (BENCHMARKING) .....	75
11.2.1.	<i>Consideraciones sobre SAPI y el motor Lernout &amp; Hauspie® TruVoice Text-To-Speech Engine</i> .....	75
11.2.2.	<i>Consideraciones generales sobre el motor SVTTS de SoftVoice Inc.</i> .....	76
11.2.3.	<i>Ambientes de Prueba de los motores: Especificaciones técnicas</i> .....	76
11.2.4.	<i>Tiempos de falla observados en cada ambiente de prueba</i> .....	77
11.2.5.	<i>Consideraciones sobre las implementaciones tecnológicas</i> .....	79
11.2.6.	<i>Consideraciones sobre Disponibilidad</i> .....	80
11.2.7.	<i>Consideraciones sobre el Control Aplicativo</i> .....	81
11.3.	SÍNTESIS COMPARATIVA.....	82
11.4.	SOFTVOICE TEXT-TO-SPEECH : SVTTS.....	84
11.4.1.	<i>Sobre la compañía fabricante: SoftVoice Inc.</i> .....	84
11.4.2.	<i>Sobre el sistema SoftVoice Text-To-Speech</i> .....	85
11.4.3.	<i>Implementación, instalación y disponibilidad</i> .....	87
11.4.4.	<i>Funciones Disponibles</i> .....	88
11.5.	ASPECTOS FUNCIONALES Y DECISIONES DE DISEÑO .....	90
<b>12.</b>	<b>DISEÑO DE LA API INDEPENDIENTE .....</b>	<b>94</b>
12.1.	IMPLEMENTACIÓN PARA EL LENGUAJE MICROSOFT VISUAL BASIC .....	94
12.2.	INTERFAZ DE PUNTEROS FAR Y ACCESO DIRECTO A MEMORIA.....	95
12.2.1.	<i>Alternativas tecnológicas para solucionar el problema</i> .....	96
12.2.2.	<i>Llamadas a Funciones Compiladas en una DLL</i> .....	97
12.2.3.	<i>Equivalencias entre los tipos de datos</i> .....	98
12.2.4.	<i>Nombre interno de la función (punto de entrada en la DLL)</i> .....	99
12.2.5.	<i>Resultado de la Investigación y Aplicación referida</i> .....	100
12.3.	CONSTANTES SIMBÓLICAS.....	102
12.4.	INCORRECTA LECTURA DE SIGNOS Y SÍMBOLOS DEL LENGUAJE .....	103
12.5.	INCORRECTA LECTURA DE EXPRESIONES NUMÉRICAS .....	104

12.6.	REFERENCIA DE PROCEDIMIENTOS Y FUNCIONES IMPLEMENTADOS EN LA API PARA EL DESARROLLO DE SOFTWARE DESDE VISUAL BASIC .....	105
12.7.	CÓDIGO FUENTE DE LA API CONSTRUIDA.....	111
<b>13.</b>	<b>DESARROLLO DE LA APLICACIÓN DE SOFTWARE.....</b>	<b>139</b>
13.1.	ESTRUCTURA DE RELEASES ENTREGADOS .....	139
13.2.	VERSIÓN 1.0 Y SUBSIGUIENTES .....	140
13.2.1.	<i>Requerimientos iniciales</i> .....	140
13.2.2.	<i>Características del Software</i> .....	142
13.3.	VERSIÓN 2.0 Y SUBSIGUIENTES .....	151
13.3.1.	<i>Requerimientos adicionales para la segunda versión.....</i>	152
13.3.2.	<i>Características del Software</i> .....	154
13.4.	RELEASE FINAL: VERSIÓN 3.0.....	159
13.4.1.	<i>Nuevos requerimientos para la versión 3.0.....</i>	159
13.4.2.	<i>Prototipo de Interfaz.....</i>	163
13.4.3.	<i>Modelos UML para los diferentes Casos de Uso.....</i>	166
	<b>CASO DE USO: 1.1. INGRESO DE PUNTOS BRAILLE .....</b>	<b>169</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 1.2. ACEPTACIÓN DEL CARÁCTER INGRESADO .....</b>	<b>171</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 1.3. CANCELACIÓN DEL CARÁCTER INGRESADO .....</b>	<b>173</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 2.1. LECTURA DEL ÚLTIMO CARÁCTER INGRESADO .....</b>	<b>178</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 2.2. LECTURA DE LA ÚLTIMA PALABRA .....</b>	<b>180</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 2.3. LECTURA DE LA ÚLTIMA ORACIÓN .....</b>	<b>183</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 2.4. LECTURA DEL ÚLTIMO PÁRRAFO .....</b>	<b>187</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 2.5. LECTURA DESDE LA ÚLTIMA MARCA DE BLOQUE.....</b>	<b>191</b>
	<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>

<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 2.6. LECTURA DE TODO EL TEXTO .....</b>	<b>195</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 3. INTERVENCIÓN EXTERNA EN EL TEXTO.....</b>	<b>197</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 4. ACCESO AL MENÚ DE LA APLICACIÓN .....</b>	<b>199</b>
<b>DIAGRAMA DE RELACIONES DE C.U. ....</b>	<b>169</b>
<b>CASO DE USO: 5.1.1. ELIMINACIÓN CARÁCTER INMEDIATO ANTERIOR.....</b>	<b>205</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.1.2. ELIMINACIÓN ÚLTIMA PALABRA.....</b>	<b>207</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.1.3. ELIMINACIÓN ÚLTIMA ORACIÓN .....</b>	<b>210</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.1.4. ELIMINACIÓN ÚLTIMO PÁRRAFO .....</b>	<b>214</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.1.5. ELIMINACIÓN DE TODO EL TEXTO.....</b>	<b>218</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.1. DESPLAZAR EL CURSOR UN CARÁCTER HACIA ATRÁS .....</b>	<b>221</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.2. DESPLAZAR EL CURSOR AL COMIENZO DE LA PALABRA .....</b>	<b>223</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.3. DESPLAZAR EL CURSOR AL COMIENZO DE LA ORACIÓN .....</b>	<b>226</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.4. DESPLAZAR EL CURSOR AL COMIENZO DEL PÁRRAFO .....</b>	<b>229</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.5. DESPLAZAR EL CURSOR HASTA EL MARCADOR ANTERIOR... 232</b>	
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.2.6. DESPLAZAR EL CURSOR HASTA EL INICIO DEL TEXTO .....</b>	<b>235</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.3.1. DESPLAZAR EL CURSOR UN CARÁCTER HACIA DELANTE.....</b>	<b>238</b>
<b>CASO DE USO: 5.3.2. DESPLAZAR EL CURSOR AL COMIENZO DE LA SGTE. PALABRA .....</b>	<b>240</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.3.3. DESPLAZAR EL CURSOR AL COMIENZO DE LA SGTE. ORACIÓN .....</b>	<b>243</b>
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.3.4. DESPLAZAR EL CURSOR AL COMIENZO DEL SGTE. PÁRRAFO 246</b>	
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.3.5. DESPLAZAR EL CURSOR HASTA EL MARCADOR SIGUIENTE.. 249</b>	
<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
<b>CASO DE USO: 5.3.6. DESPLAZAR EL CURSOR HASTA EL FINAL DEL TEXTO .....</b>	<b>252</b>

	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 6.1. INGRESO DE UNA OPERACIÓN ARITMÉTICA .....</b>	<b>256</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 6.2.1. EVALUAR Y LEER EL RESULTADO .....</b>	<b>260</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 6.2.2. EVALUAR Y PEGAR EL RESULTADO .....</b>	<b>262</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 6.2.3. EVALUAR Y PEGAR EL CÁLCULO Y RESULTADO.....</b>	<b>264</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 6.3. LEER OPERACIÓN.....</b>	<b>266</b>
	<b>CASO DE USO: 6.4. PEGAR OPERACIÓN EN EL TEXTO EN EDICIÓN.....</b>	<b>267</b>
	<b>CASO DE USO: 6.5. INICIALIZAR CALCULADORA PARA UNA NUEVA OPERACIÓN..</b>	<b>268</b>
	<b>CASO DE USO: 6.6. LEER MENÚ DE LA CALCULADORA .....</b>	<b>269</b>
	<b>CASO DE USO: 6.7. CERRAR LA CALCULADORA .....</b>	<b>270</b>
	<b>CASO DE USO: 7.1.A. ABRIR ARCHIVO (FUNCIÓN DE INTERFAZ AUDITIVA PARA EL</b> <b>USUARIO NO VIDENTE) .....</b>	<b>272</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 7.1.B. ABRIR ARCHIVO (FUNCIÓN DE INTERFAZ VISUAL) .....</b>	<b>274</b>
	<b>CASO DE USO: 7.2.A. GUARDAR ARCHIVO (FUNCIÓN DE INTERFAZ AUDITIVA PARA</b> <b>EL USUARIO NO VIDENTE) .....</b>	<b>275</b>
	<b>CASO DE USO: 7.2.B. GUARDAR ARCHIVO (FUNCIÓN DE INTERFAZ VISUAL) .....</b>	<b>276</b>
	<b>CASO DE USO: 8.1. SELECCIÓN DE VOZ PREDETERMINADA .....</b>	<b>278</b>
	<b>DIAGRAMA DE COLABORACIÓN .....</b>	<b>169</b>
	<b>CASO DE USO: 8.2. CONFIGURACIÓN DE CARACTERÍSTICAS FUNCIONALES.....</b>	<b>280</b>
	<b>CASO DE USO: 9. AUTOGRABACIÓN DEL TEXTO EN EDICIÓN.....</b>	<b>283</b>
13.4.4.	<i>Aspectos de Modelado e Implementación de Objetos de Datos.....</i>	<i>166</i>
<b>14.</b>	<b>CONCLUSIONES .....</b>	<b>288</b>
	RESULTADO APLICADO PARA LA EVOLUCIÓN ESCOLAR DE LUCAS GARCÍA .....	288
	ADQUISICIÓN DE NUEVAS HABILIDADES PSICOPEDAGÓGICAS POR PARTE DEL USUARIO DE LA HERRAMIENTA .....	288
	APORTE INTELECTUAL PARA SOFTVOICE Y LA COMUNIDAD DE DESARROLLADORES DE SOFTWARE .....	289
<b>15.</b>	<b>BIBLIOGRAFIA .....</b>	<b>288</b>
	<b>EPÍLOGO .....</b>	<b>296</b>

**FORMULACIÓN DEL  
PROYECTO DE APLICACIÓN  
PROFESIONAL**

## **1. Consideraciones generales del presente trabajo**

---

Se presentan a continuación las consideraciones sobre el área específica del presente Proyecto de Aplicación Profesional.

### **1.1. Tema del Trabajo**

El presente proyecto tiene como meta el desarrollo de una solución informática para un niño con discapacidad, impedido de acceder a los mecanismos convencionales de lecto-escritura.

El trabajo consistirá, en una primera etapa, en el relevamiento relacionado a las necesidades y capacidades del usuario, y la consecuente indagación en materia de recursos tecnológicos disponibles. Asimismo, en una segunda etapa, se desarrollarán los diferentes aspectos relacionados a la construcción de una aplicación específica que satisfaga sus necesidades, incluyendo el análisis, diseño, desarrollo e implementación de la misma.

### **1.2. Sobre el Título del Presente Trabajo Final**

*“Sistema Informático de Apoyo a la Lecto-Escritura para Niño con Discapacidad Múltiple”*

El título del trabajo refiere el rol de la solución informática a construir en relación al desarrollo del proceso intelectual de lecto-escritura en un niño, cuyo perfil se analizará más adelante, quien por su discapacidad se clasifica como multidéficit o multiimpedido<sup>1</sup>.

Dicha función de “apoyo” se planteará a través del trabajo coordinado con el equipo terapéutico-pedagógico del niño, de manera interdisciplinaria. Se detallarán los requerimientos y las habilidades que se observen en el usuario (paciente) que posibilitan el enfoque dado a este desarrollo, obteniendo el mayor provecho de los recursos tecnológicos en el caso de este niño, tanto en la integración escolar como

---

<sup>1</sup> Personas con Discapacidad Múltiple (Multiimpedimento): Persona que presenta más de una discapacidad sensorial, física o mental de manera simultánea.

en la disponibilidad de fuentes de información textuales a las que no tiene acceso de otro modo.

El alcance del proyecto desarrollado y sus implicaciones se analizarán más detenidamente en la fundamentación del presente trabajo.

### **1.3. Sobre la Denominación utilizada: Personas con Discapacidad**

De acuerdo a la clasificación establecida por la Organización Mundial de la Salud, en el documento denominado “Clasificación Internacional de Deficiencias, Discapacidades y Minusvalías”, conocido por sus siglas **CIDDM** publicado en 1980 por dicho organismo internacional, se establece un nuevo criterio para el abordaje de la problemática de la discapacidad:

*El objetivo planteado en la versión de 1980 de la CIDDM se centra en traspasar las barreras de la enfermedad, entendida en su concepción clásica. Se trata de ir más allá del proceso mismo de la enfermedad y clasificar las consecuencias que ésta deja en el individuo tanto en su propio cuerpo, como en su persona y en su relación con la sociedad. De este modo, se va más allá del esquema de la enfermedad como:*

***Etiología → Patología → Manifestación***

*Así, la OMS propone con la CIDDM un esquema nuevo:*

***Enfermedad → Deficiencia → Discapacidad → Minusvalía***

#### **Conceptos y terminología de la CIDDM**

*En este nuevo esquema ha de entenderse que:*

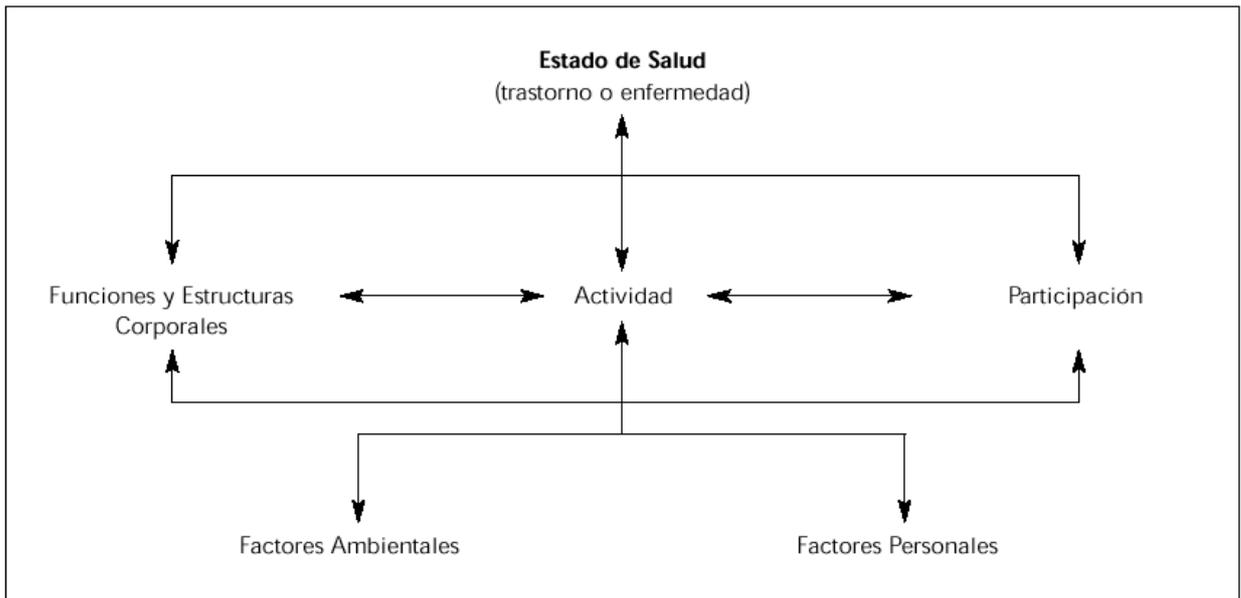
- *La enfermedad es una situación intrínseca que abarca cualquier tipo de enfermedad, trastorno o accidente. La enfermedad está clasificada por la OMS en su Clasificación Internacional de Enfermedades y Problemas de Salud Relacionados (CIE).*
- *La deficiencia es la exteriorización directa de las consecuencias de la enfermedad y se manifiesta tanto en los órganos del cuerpo como en sus funciones (incluidas las psicológicas).*

- *La discapacidad es la objetivación de la deficiencia en el sujeto y con una repercusión directa en su capacidad de realizar actividades en los términos considerados normales para cualquier sujeto de sus características (edad, género,...).*
- *La minusvalía es la socialización de la problemática causada en un sujeto por las consecuencias de una enfermedad, manifestada a través de la deficiencia y/o la discapacidad, y que afecta al desempeño del rol social que le es propio.*

*(\*) Extraído del trabajo “Clasificaciones de la OMS sobre discapacidad” publicado en 1997 por Carlos Egea García (Pedagogo) y Alicia Sarabia Sánchez (Trabajadora Social) internacionalmente reconocidos en esta materia.*

Dicho documento (CIDDM) fue posteriormente revisado y ampliado, llegando en el año 1997 a su segunda versión oficial, denominada CIDDM-2. La versión final aprobada en 2001 de dicho documento se denominó «Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud», siendo a partir de entonces referida por sus siglas en castellano CIF .

Al esquema causal y lineal de la CIDDM de 1980, la CIF responde con un esquema de múltiples interacciones entre sus dimensiones y áreas. Aunque, según se declara en el texto de esta Clasificación, la CIF «no establece un “proceso” del funcionamiento y la discapacidad», se intentan aclarar dichas interacciones mediante el siguiente esquema:



En la CIF, la Discapacidad «engloba las deficiencias, las limitaciones en la actividad y las restricciones en la participación» y, así, deja de emplearse como una parte de las consecuencias de la enfermedad y se eleva a término «paraguas» para todas las condiciones de salud negativas.

Así describen T.B. Üstün y colaboradores el proceso por el que se llega a adoptar el uso de «discapacidad» como término globalizador de los factores negativos del funcionamiento humano:

*«Si bien “discapacidad” (“disability”) es un término utilizado universalmente, tanto en el lenguaje cotidiano como en la literatura profesional y científica, resulta ambiguo. El término podría referirse a una anomalía funcional o estructural en el ámbito corporal (por ejemplo, un problema en el metabolismo de las proteínas*

*o la pérdida de una pierna); un problema de actuación o comportamiento en el ámbito de la persona (por ejemplo, ser incapaz de vestirse o de conducir un coche); o, incluso, en el ámbito social al estar socialmente en desventaja a causa de los problemas funcionales en el ámbito corporal o personal (por ejemplo, perder el trabajo o que te denieguen el permiso de conducir).*

*Para evitar la confusión entre estas tres nociones, muy diferentes entre sí, la CIDDM de 1980 utilizó los términos “deficiencia” (“impairment”), “discapacidad” (“disability”) y “minusvalía” (“handicap”) para distinguir estas*

*tres dimensiones, con el término paraguas “disablement” (que no tiene paridad en castellano, pero*

*podríamos traducir como “discapacitación” o “discapacitamiento”) cubriendo a las tres.*

*En el proceso de revisión, se decidió que la CIF no debía ser una clasificación de los problemas funcionales que las personas pueden experimentar, sino una clasificación universal del funcionamiento humano en sí mismo, tanto positivo como negativo. Por esta causa, y por la importancia de expresar la clasificación en un lenguaje neutral y flexible, los tres ámbitos fueron renombrados “estructuras y funciones corporales” (por “deficiencias”), “actividad” (por “discapacidad”) y “participación” (por “minus-valía”). Puesto que el término “disablement” resultó difícil de traducir (como sucedía en castellano) y ahora “discapacidad” estaba liberada de su asociación con el ámbito personal de los problemas funcionales, se decidió volver a “dis-capacidad” como término comprensivo de los tres ámbitos de dificultad funcional. El término “disablement” se mantuvo como un término acordado para nombrar el proceso interactivo por el cual los ámbitos de la discapacidad suceden (lo que podríamos denominar en castellano “discapacitación” o “discapacitamiento”)».*

Extraído de **Disability and Culture: Universalism and Diversity**, publicado por la Organización Mundial de la Salud y Hogrefe & Huber Publishers, 2001.

Adicionalmente a lo anterior, y aún sin guardar los aspectos de universalidad que los documentos referidos establecen, pero basándose en ellos, resulta oportuno citar la definición de Personas con Discapacidad presente en el artículo 3º de la Ley Orgánica de Atención Integral a Personas con Discapacidad, de la República Bolivariana de Venezuela. Allí se establece:

*“Personas con discapacidad son niños, niñas, adolescentes, hombres, mujeres y adultos mayores que, por diversas razones genéticas, congénitas o adquiridas como consecuencia de accidentes perinatales, laborales, de tránsito, domésticos o de otros tipos, de enfermedades laborales o no, intervenciones quirúrgicas,*

*tratamientos médicos, aplicación de medicamentos, exposición a sustancias o ambientes contaminantes, evidencian una disminución de diverso valor estadístico, o ausencia prolongada o permanente, de alguna de sus capacidades sensoriales, motrices o intelectuales”.*

### **Importancia de las Capacidades de las Personas: *Individuos con Capacidades Especiales***

Independientemente de la nomenclatura internacionalmente establecida y las diversas fuentes referidas, el enfoque personal del autor de este trabajo sobre las capacidades de las personas estriba fundamentalmente en el concepto de **Resiliencia**<sup>2</sup> y el desarrollo que desde la misma pueden llevar a cabo todos los seres humanos de sus capacidades, en sus diferentes dimensiones.

Por lo anterior, considero que la problemática de la discapacidad debe entenderse en realidad en el contexto de “**Personas con Capacidades Especiales**, o cuando menos **Capacidades Diferentes**”, cuyo potencial puede desarrollarse más allá de las limitaciones sensoriales, motrices o intelectuales.

No obstante, para mantener la nomenclatura establecida en los ámbitos profesionales y científicos, en el resto de este trabajo se conservará la denominación de Personas con Discapacidad, según las referencias citadas anteriormente.

---

<sup>2</sup> *La **resiliencia** es la capacidad de una persona o grupo para seguir proyectándose en el futuro a pesar de acontecimientos desestabilizadores, de condiciones de vida difíciles y de traumas a veces graves. La resiliencia se sitúa en una corriente de psicología positiva y dinámica de fomento de la salud mental y parece una realidad confirmada por el testimonio de muchísimas personas que, aún habiendo vivido una situación traumática, han conseguido encajarla y seguir desenvolviéndose y viviendo, incluso, en un nivel superior, como si el trauma vivido y asumido hubiera desarrollado en ellos recursos latentes e insospechados.*

## 2. Introducción General

---

La aplicación de recursos tecnológicos para mejorar la calidad de vida de las personas con discapacidad es una temática que ha adquirido últimamente mayor relevancia, particularmente desde el punto de vista mediático. Pero más allá de algunos avances, el actual nivel de desarrollo de este tipo de aplicaciones es aún incipiente.

Las posibilidades que ofrece hoy en día la tecnología son innumerables; y si bien se han desarrollado hasta aquí muchas aplicaciones para personas con discapacidad en general, todo lo hecho tiene dos características que podrían considerarse contrarias a los requisitos del presente proyecto:

- Son soluciones generales, producidas en serie - en el caso de componentes de hardware o elementos físicos - o como paquetes de “software enlatado”<sup>3</sup>. Paradójicamente, una de las principales características de las personas con multidéficit es que cada caso es diferente a los demás, por lo que las soluciones generales tienen una aplicación relativa en ellos.
- Muchos de los desarrollos son importados, y los que no, están siendo llevados adelante por especialistas u organizaciones radicados principalmente en la Capital Federal, con lo que las posibilidades de contacto con el paciente, su seguimiento y adaptación son complicados desde el punto de vista operativo, a la vez que muy costosos.

El presente trabajo pretende cubrir - al menos en parte - precisamente ese vacío de disponibilidad de soluciones tecnológicas y su aplicación con una orientación bien específica: *aportar elementos que posibiliten mejorar la calidad de vida del sujeto<sup>4</sup> y sus posibilidades de desarrollo.*

La categoría general de este tipo de herramientas, que posibilitan ampliar las capacidades comunicacionales del sujeto, se denomina en el ámbito de los terapeutas de personas con discapacidad y rehabilitadores como instrumentos de “comunicación aumentativa”. Esta es

---

<sup>3</sup> **Software Enlatado:** Referencia general a los programas de computación que son distribuidos como paquete cerrado, desarrollados con objetivos de satisfacción de necesidades generales de una cierta población, y no con un enfoque en la solución de los problemas específicos de un cierto usuario.

<sup>4</sup> **Sujeto:** En términos de nomenclatura, “sujeto” refiere al “usuario” (desde el punto de vista informático) o “paciente” (desde el punto de vista terapéutico).

una taxonomía realmente amplia, y cubre desde interfaces que posibiliten la expresión de personas que por sus impedimentos motrices no pueden hacerlo, hasta soluciones que amplíen o desarrollen potenciales comunicacionales no desarrollados en el paciente.

Se describirá a continuación el caso del sujeto, y el modo en que esta solución aportará a sus posibilidades de desarrollo.

### **2.1. El usuario**

Lucas García<sup>5</sup> es un niño de 10 años de edad, quien presenta una patología de origen congénito compleja. Su cuadro clínico se denomina “anoftalmia”, es decir: ausencia de globos oculares. Como se deduce, es ciego. Utiliza dos ojos de vidrio por cuestiones estéticas, pero naturalmente tiene sus cavidades oculares vacías.



**Lucas García**

Normalmente, un cuadro de esta naturaleza no se presenta de manera aislada, sino como una más de las manifestaciones de un problema congénito a nivel sistémico. En el caso de Lucas, la evaluación clínica de sus terapeutas señala una hipotonía muscular generalizada en su primera infancia, particularmente notoria en lo que refiere a la motricidad fina. Si bien, gracias al tratamiento y estimulación temprana que recibió, logró mejorar notablemente el tono muscular general y controlar movimientos involuntarios, no ha podido desarrollar la motricidad fina las falanges. Dicho en otros términos, Lucas es incapaz de mover sus dedos de manera independiente y delicada. Mueve bien sus manos y maneja relativamente bien el espacio, pero no puede desarrollar las habilidades motrices que le permitan acceder a la lectura, y mucho menos a la escritura, utilizando el Sistema Braille convencional. Leer una línea de Braille implica el desarrollo de una sensibilidad

---

<sup>5</sup> El nombre completo del usuario y su imagen son divulgados en el presente trabajo con el correspondiente consentimiento y autorización de sus padres, por tratarse de un menor de edad, según consta en el documento anexo al final de este.

especial en las yemas de los dedos, y habilidades motrices finas muy precisas; si se quiere, superiores a las normalmente observadas en personas dotadas de visión normal.

En cuanto a su capacidad intelectual, es muy buena. No se manifiestan en Lucas diferencias significativas respecto a otros niños de su edad. Está inserto en una escuela normal, y es asistido en la misma por una maestra especial de integración - profesora de ciegos.

En el aspecto emocional ha tenido una fuerte incidencia el tratamiento temprano, por lo que no presenta en la actualidad alteraciones emocionales importantes.

Considerando su cuadro general, su familia y equipo terapéutico considera presente en él una capacidad no explotada para desarrollar los mecanismos de lecto-escritura, impedidos precisamente por los aspectos físicos señalados.

## **2.2. Antecedentes**

La problemática de soluciones tecnológicas adaptativas o de comunicación aumentativa para personas con discapacidad múltiple, es un terreno considerado prácticamente como “virgen” por los terapeutas y especialistas consultados. Desde hace varias décadas existen posibilidades tecnológicas de desarrollo de soluciones en este sentido, aunque no hayan sido convenientemente explotadas.

No obstante, al menos en lo que a soluciones para personas con discapacidad en general se refiere, pareciera observarse que en la sociedad - al menos en el ámbito de la provincia -, este tipo de cuestiones estuvieran cobrando mayor auge.

Sin embargo, el nivel de desarrollo actual en este campo no es muy alto. Los profesionales y organizaciones seriamente comprometidos en este tipo de aplicaciones no son demasiados, y más allá de algún desarrollo académico aislado es poco lo que se conoce. Existen trabajos documentados al respecto, pero coherentemente con lo antes señalado, cada sujeto que presenta una discapacidad múltiple es muy particular, y la generalización de soluciones es difícilmente aplicable.

### **2.3. Descripción del área problemática**

Como resulta natural, el principal aspecto considerado en el proyecto es la posibilidad de desarrollo de una solución tecnológica que permita superar los impedimentos físicos del sujeto en relación al desarrollo de habilidades de lecto-escritura, y su posterior construcción e implementación. Sin embargo, en el ámbito del área problemática a trabajar, no sólo caben considerarse las necesidades específicas del sujeto, sino su interrelación con diferentes aspectos, a saber:

➤ **Pertinencia y usabilidad de la solución desarrollada**

*El contexto educativo, el ambiente áulico, la transportabilidad de la solución y la adecuación funcional de la herramienta deben ser considerados.*

➤ **Autonomía del sujeto en el uso de la herramienta**

*Refiere a las posibilidades de utilización sin asistencia de terceros y el trabajo autónomo e independiente del sujeto con la herramienta.*

➤ **Posibilidades de aplicación pedagógica**

*La utilidad pedagógica de la solución en función de su objetivo determina fuertemente las características a considerar para el desarrollo.*

➤ **Interactividad entre el sujeto y su profesora, de cara a la misma solución**

*El nivel de interacción entre el sujeto y su maestra de integración, en torno a la herramienta, es un aspecto importante a considerar. Esto marcará de manera determinante los aspectos de interfaz para ambos actores que habrán de considerarse.*

➤ **Posibilidades de ampliación de la solución (escalabilidad)**

*Es importante que el desarrollo de la solución sea evolutivo y escalable, como característica central del trabajo. No resulta adecuado pensar en una solución estanca y completa de manera inicial, sino en un marco flexible que posibilite la ampliación de las capacidades de la herramienta*

*acompañando la evolución en el plan pedagógico que sigue el sujeto, inserto en una escuela normal.*

Los aspectos antes señalados se consideran a partir de las observaciones realizadas y los elementos de relevamiento reunidos, y serán oportunamente considerados en este trabajo.

#### **2.4. *Formulación del Problema***

El problema a considerar, será entonces: *De qué manera interrelacionar los elementos tecnológicos disponibles en el diseño e implementación de una solución de las características señaladas como herramienta para el desarrollo de la actividad de lecto-escritura por parte del sujeto, posibilitando su adaptación y evolución según las necesidades pedagógicas y funcionales en materia de lenguaje escrito.*

#### **2.5. *Justificación***

El presente proyecto se justifica desde diferentes puntos de vista, según se detalla a continuación:

##### **2.5.1. *Disponibilidad, Alcance y Posibilidades de soluciones existentes***

Los desarrollos disponibles en materia de sistemas de apoyo al proceso de lectoescritura para personas con discapacidad visual conjunta con multidéficit son muy pocos. Existen algunos desarrollos comerciales, de escasa aplicación al sujeto, cuyas características de interfaz con el usuario las tornan inaccesibles para Lucas en función de su discapacidad motriz. Se trata de sistemas de aplicación general, que por su misma naturaleza no satisfacen las necesidades pedagógicas específicas o particulares que se plantean como requerimientos para el presente desarrollo. Además, dichas aplicaciones están básicamente planteadas en términos de interfaz de usuario para personas – y más específicamente, niños – con sólo una discapacidad presente: la ceguera o disminución visual. Según los especialistas consultados, no hay soluciones disponibles – al menos localmente – que integren la discapacidad motriz fina con la ceguera.

### *2.5.2. Posibilidades de desarrollo superadoras para el sujeto*

La principal motivación para el desarrollo de este trabajo es posibilitar a Lucas su desarrollo escolar e intelectual, a través del acceso a la lectura y escritura. La capacidad puesta de manifiesto en el sujeto, y la opinión profesional de sus terapeutas y maestra de integración, le auguran importantes posibilidades. Su potencial se ve limitado por la falta de recursos técnicos para dar curso a su capacidad intelectual en esta materia, por lo que abrir la puerta de salida a dicho potencial es un objetivo tentador y apasionante, en post del desarrollo humano de la persona.

### *2.5.3. Aporte tecnológico*

Mediante este proyecto se aportará al campo de la integración y el desarrollo de las personas con discapacidad: en particular los niños con capacidades intelectuales normales y discapacidades visual y funcional. Si bien el desarrollo estará orientado a la satisfacción de las necesidades específicas de Lucas, la tarea desarrollada aportará a otros proyectos que puedan desarrollarse en este campo para sujetos con similares problemáticas, y diferentes tipos de disfunciones motrices.

### **3. Objetivos**

---

En consonancia con los aspectos introductorios señalados, se proponen para este trabajo los siguientes objetivos:

- Posibilitar el desarrollo de habilidades de lecto-escritura para el usuario destinatario de este proyecto mediante la implantación de un sistema informático que posibilite su acceso físico-sensorial a los componentes de la solución
- Permitir al usuario autonomía e independencia en la utilización de la solución.
- Aportar a la comunidad informática y terapéutica de personas con discapacidad una base tecnológica para el desarrollo de soluciones de similar naturaleza.

#### **3.1. Objetivos Específicos Detallados**

Se plantean para el presente trabajo los objetivos específicos y detallados que siguen a continuación:

- a. Indagar, Seleccionar y Evaluar diferentes motores de software de Síntesis de palabra, considerando entre sus diferentes aspectos la disponibilidad, estabilidad, costos y posibilidades de integración como componente del software de aplicación.
- b. Relevar la interfaz de programación del motor seleccionado y diseñar una API Independiente y uniforme para su aplicación en Visual Basic.
- c. Implementar y documentar la API diseñada, estableciendo un modelo de programación uniforme. Esto incluye:
  - ✓ Traducción de los protocolos de llamada a las funciones requeridas implementadas en la API seleccionada
  - ✓ Conversión de tipos para todas constantes y macros presentes en las funciones de API
- d. Comprender las capacidades motrices, sensoriales e intelectuales del sujeto, y determinar las posibilidades de aplicación tecnológica en pos de los objetivos planteados.
- e. Diseñar una solución que combine recursos de Hardware y Software, conjuntamente con el desarrollo de una aplicación específica y a medida, que

permita al sujeto realizar de manera autónoma la actividad de lecto-escritura, aportando un medio adecuado para el desarrollo del lenguaje escrito por parte de él. En este sentido deberán considerarse particularmente los aspectos y características requeridas, señalados en el apartado 2.3 precedente.

### **3.2. Límite y Alcance**

De acuerdo a los elementos específicos de relevamiento a determinar – y a la luz de los aspectos terapéutico-pedagógicos que se establezcan como necesidades –, se cubrirá en el proyecto la creación de una herramienta que contribuya a solucionar la problemática planteada, incluyendo la evaluación y selección de recursos tecnológicos disponibles y su utilización en el desarrollo de la solución.

En tal sentido, se enfocarán los siguientes aspectos:

- Búsqueda de un medio de comunicación con la aplicación que posibilite una interfaz adecuada para el sujeto, como así también para su entorno.
- Determinación de los recursos tecnológicos más adecuados para cada componente de la solución a desarrollar.
- Análisis, Diseño, Desarrollo e Implantación de la solución a construir, satisfaciendo las necesidades del sujeto en relación a la disponibilidad del soporte de lecto-escritura requerido.

No serán desarrolladas en este trabajo las estrategias académicas y de integración a desarrollar por parte de las psicopedagogas que asistirán al sujeto en la aplicación de la herramienta, ni ningún otro aspecto terapéutico, amén de los requerimientos funcionales que se establezcan.

## **4. Marco Referencial Teórico**

---

Fueron consultadas diversas fuentes, procurando establecer el marco referencial en cuatro grandes grupos o categorías conceptuales y metodológicas:

### ***1º. Respetto de la problemática educativa en niños no videntes.***

Se consideraron los siguientes aspectos:

- Características generales del trabajo terapéutico-pedagógico con niños con discapacidad visual adicional a otro déficit. Elementos de experiencias previas desarrolladas en relación a la problemática planteada.
- Requerimientos funcionales para el desarrollo de las habilidades y mecánica de lecto-escritura.

### ***2º. Respetto de los desarrollos tecnológicos existentes***

Se consideraron los siguientes aspectos:

- Instrumentos tecnológicos utilizados en el área de comunicación aumentativa para personas con discapacidad visual.
- Soluciones documentadas / disponibles.

### ***3º. Respetto de los elementos tecnológicos disponibles para el desarrollo de una solución a medida***

Se consideraron los siguientes aspectos:

- Interconexión de dispositivos de hardware específicos a una PC. Puertos de comunicación, tecnologías y protocolos: cableado, electrónica y programación.
- Productos y soluciones de software de bajo nivel de Síntesis de Palabra, y su disponibilidad de programación.
- Lenguajes de programación con capacidad de gestión de elementos de interfaz auditiva y gráfica.

#### **4º. *Respecto del ciclo de vida del proyecto y los modelos de análisis y diseño del sistema***

Se consideraron los siguientes aspectos:

- Especificaciones del lenguaje unificado de modelado (UML) para el análisis y diseño del sistema.
- Modelos de ciclos de vida de sistemas
- Elementos y guías de diseño de interfaces visuales y auditivas orientadas a la usabilidad<sup>6</sup> del software a construir

Se mencionan a continuación los elementos más significativos analizados para cada uno de los aspectos señalados, indicando las fuentes consultadas en cada caso:

##### **4.1. *Problemática educativa en niños no videntes.***

Las características y principales dificultades de los pacientes infantiles ciegos de nacimiento, en el contexto del trabajo terapéutico-pedagógico que con ellos se desarrolla, fueron establecidos en entrevistas con diferentes terapeutas especializados. Se consultó a la Prof. María de las Mercedes Rigo, a la Prof. Soledad Martín, al Lic. Hugo Cantore, y especialmente a la Prof. Gabriela Migueltoarena, quien es actualmente la maestra de integración de Lucas. Los aportes realizados por los profesionales se sustentan no sólo en los conceptos de forma establecidos en sus respectivas disciplinas, sino también en relación a las experiencias por ellos desarrolladas. En este sentido, se indagó además sobre diferentes experiencias documentadas en relación a esta temática específica.

También se consideraron en el marco referencial varios artículos técnicos específicos de la problemática educativa en niños invidentes y/o multiimpedidos, de los cuales los principales se citan a continuación:

---

<sup>6</sup> Usabilidad de un producto de software: (a) Indica el grado en que un programa puede ser usado por un usuario que se enfrenta a él por primera vez, con la menor asistencia posible. (b) Refiere también a la simpleza y agilidad con que la Interfaz Hombre-Máquina implementada en el Software permite a su usuario realizar la tarea para la cual fue diseñado.

- “Alfabetización de las personas que son sordas e invidentes” (por Barbara Miles) Disponible en el Centro de Información Nacional sobre Niños que son Sordos y Ciegos de Estados Unidos, en Internet: <http://www.tr.wou.edu/dblink/lit-span.htm><sup>7</sup>
- “Discapacidad visual y acceso a la información” (por Soledad Mochales López < [soledadml@correo.cop.es](mailto:soledadml@correo.cop.es) >) Disponible en <http://usuarios.iponet.es/casinada/25ciegos.htm><sup>8</sup>
- Notas disponibles en las publicaciones del 3º Congreso Iberoamericano de Informática Educativa:
  - “As Tecnologias de Informação e Comunicação (T.I.C.) e a pessoa portadora de deficiência motora severa: Construção de um modelo de avaliação.”, disponible en: [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/EULALIA/EULALIA.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/EULALIA/EULALIA.HTML)<sup>9</sup>
  - “CantaLetras: Sistema multimedial de apoyo al proceso de enseñanza de la lectoescritura para niños ciegos”, disponible en [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/125\\_E\\_126/DEMO1.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/125_E_126/DEMO1.HTML)<sup>10</sup>
  - “FALAS: Ferramenta Alternativa de Aquisição Simbólica”, disponible en [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/145/FALAS.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/145/FALAS.HTML)<sup>11</sup>
- Fue también considerada la biografía de Hellen Keller, sus consideraciones respecto de ese “mundo interior” del que ella habla, la percepción con privación

---

<sup>7</sup> Fecha de actualización de la página consultada: 10/03/2004

<sup>8</sup> Fecha de actualización de la página consultada: 10/03/2004

<sup>9</sup> Fecha de actualización de la página consultada: 17/05/2004

<sup>10</sup> Fecha de actualización de la página consultada: 17/05/2004

<sup>11</sup> Fecha de actualización de la página consultada: 17/05/2004

de sentidos, y aspectos referenciales de profesionales y asistentes que con ella tuvieron oportunidad de interactuar. Mucha de la información considerada fue obtenida en Internet, en la dirección: <http://www.sordoceguera.org/Historias Reales/Helen Keller.htm> <sup>12</sup>

- Videoclips relacionados a los aspectos de comunicación sobre ciegos. En el sitio de SENSE (Organización Británica sin fines de lucro de apoyo a personas con discapacidad sensorial múltiple), en <http://www.sense.org.uk> <sup>13</sup>, están disponibles diferentes materiales fílmicos, siendo los que más aportaron al presente trabajo los siguientes:
  - Documental sobre Hellen Keller: incluyendo imágenes de Hellen, en edad adulta, desplegando sus posibilidades comunicacionales mediante diferentes herramientas.
  - Documentales sobre técnicas comunicacionales y pedagógicas con niños ciegos multiimpedidos graves, basado en experiencias desarrolladas en un centro rehabilitador italiano de avanzada.
  - Documentales sobre la adquisición de lenguaje en sordociegos.
- Trabajo de Lois Harrel y Nancy Akeson, publicado en 1987 por la American Foundation for the Blind, titulado “*IT’S MORE THAN A FLASHLIGHT: Developmental Perspectives for Visually and Multi Handicapped Infants and Preschoolers*”, presentado en el International Council for Education of the Visually Handicapped, Región Latinoamericana, Córdoba, Argentina, en 1988. Este trabajo es un “clásico” en la problemática estudiada, y su lectura resultó importantísima para formar un claro panorama de las implicancias psicológicas, pedagógicas y conceptuales de los niños ciegos, y en particular los multiimpedidos.

---

<sup>12</sup> Fecha de actualización de la página consultada: 22/04/2004

<sup>13</sup> Fecha de actualización de la página consultada: 20/04/2004

Archivos descargados: comunicacion.wmv , Hellen Keller parte 1.wmv , Hellen Keller parte 2.wmv , Hellen Keller parte 3.wmv , sistemas de comunicacion parte 1.zip , sistemas de comunicacion parte 2.zip , sistemas de comunicacion parte 3.zip

Los principales aspectos a considerar, resultantes de las fuentes consideradas, se resumen a continuación:

- ✓ Los niños ciegos de nacimiento tienen en general dificultad para manejar conceptos relacionados al espacio, y particularmente en lo referido a orden o secuencias espaciales. En relación al proceso de lecto-escritura propiamente dicho, les resulta especialmente difícil seguir y registrar el ordenamiento secuencial de las letras que constituyen una palabra, si no cuentan con un importante feedback. En general ese mecanismo de feedback está naturalmente dado por los aspectos sensoriales involucrados en el tacto sobre los puntos que constituyen los caracteres del sistema Braille.
- ✓ El foco atencional es también un elemento a tener en cuenta. Los niños ciegos no alfabetizados se dispersan fácilmente y es requerida una guía o conducción permanente de su entrenador hasta que se desarrolle en ellos la mecánica requerida. Esta dispersión se produce tanto ante estímulos externos al proceso que se está desarrollando, como por el simple paso del tiempo. Los períodos de atención son breves e interrumpidos -algunos segundos o pocos minutos entre cada dispersión-, por lo que la terapia pedagógica considera en general estos aspectos. En la medida que se va desarrollando el programa y el sujeto va adquiriendo más elementos de acción e interacción con la herramienta, los periodos se van prolongando, como así también la duración total de la actividad.
- ✓ El aspecto lúdico, por lo mismo antes mencionado, resulta imperativo para lograr el acercamiento al proceso de lecto-escritura. Los niños ciegos son, ante todo, niños. Quieren divertirse, encontrar pertenencia con su entorno, y al mismo tiempo poder superar desafíos que les interesen.
- ✓ La interacción Educando-Entrenador-Instrumento tiene una fuerza mucho mayor que la necesaria en procesos de aprendizaje dirigidos en niños con capacidades visuales. El trabajo del terapeuta o educador con el niño ciego, de cara a las herramientas utilizadas es altamente interactivo. La atención y dedicación requerida es muy grande, por lo que en los casos de integración de los educandos en escuelas normales, se cuenta con una profesora de integración normalmente dedicada de manera exclusiva al niño, haciendo de “interfaz” entre

él y su entorno -especialmente las consignas desarrolladas en el aula-, combinando en dicho proceso el uso de las herramientas específicas que se dispongan para el proceso educativo.

#### **4.2. Desarrollos tecnológicos existentes en el campo en estudio**

La indagación realizada se basó fundamentalmente en los recursos disponibles en la World Wide Web. Los documentos y publicaciones considerados no se limitan al comentario respecto de soluciones desarrolladas, sino también en los fundamentos o conceptos pedagógicos sobre los que se apoyan. En este tópico, los elementos más significativos en su aporte al marco referencial son los siguientes:

- “SISTEMAS MULTIMEDIALES DE APOYO AL APRESTO ESCOLAR Y A LA LECTURA Y ESCRITURA INICIAL PARA NIÑOS DISCAPACITADOS VISUALES”, artículo desarrollado por profesionales de la Escuela de Psicología, la Escuela de Diseño, y la Escuela de Ingeniería / Servicio Ciencias de la Computación, de la Pontificia Universidad Católica de Chile. Disponible en <http://www.puc.cl/toquemagico/html/resumen.pdf> <sup>14</sup>  
En este artículo se analizan dos herramientas específicas: a. El toque mágico y b. Cantaletas.
- Varios de los artículos y publicaciones del 3º Congreso Iberoamericano de Informática Educativa, que fueron señalados en la categoría anterior, también lo hacen en relación a las herramientas específicas utilizadas en la comunicación aumentativa de no videntes, y en sus formas de evaluación, por lo que también se mencionan en esta sección, a la que aportan de manera específica:
  - “As Tecnologias de Informação e Comunicação (T.I.C.) e a pessoa portadora de deficiência motora severa: Construção de um modelo de avaliação.”, disponible en:  
[http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/EULALIA/EULALIA.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/EULALIA/EULALIA.HTML) <sup>15</sup>

---

<sup>14</sup> Fecha de descarga del archivo consultado: 03/10/2004

<sup>15</sup> Fecha de actualización de la página consultada: 12/05/2004

- “CantaLetras: Sistema multimedial de apoyo al proceso de enseñanza de la lectoescritura para niños ciegos”, disponible en [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/12\\_5\\_E\\_126/DEMO1.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/12_5_E_126/DEMO1.HTML)<sup>16</sup>
- “FALAS: Ferramenta Alternativa de Aquisição Simbólica”, disponible en [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/14\\_5/FALAS.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/14_5/FALAS.HTML)<sup>17</sup>
- Soluciones de Hardware disponibles para la integración del ciego en el uso de la PC, mediante el sistema Braille. Los desarrollos considerados aportan elementos concretos de referencia, cada uno por caso, según su objetivo.
  - Sistema BRAILEX®: Display braille bidimensional. Acceso del usuario invidente al contenido de la pantalla, su organización física y contenidos, a la vez que proporciona una línea braille para la lectura propiamente dicha. Información disponible en <http://www.sighted.com/spanish/braillex.html><sup>18</sup>
  - Documento de la ONCE (Organización Nacional de Ciegos de España) que cubre el estado del arte en materia de Tiflotecnología<sup>19</sup>. En este documento se describen diferentes soluciones existentes, sus características y aplicabilidad. Allí se cubre el siguiente contenido:

---

<sup>16</sup> Fecha de actualización de la página consultada: 12/05/2004

<sup>17</sup> Fecha de actualización de la página consultada: 12/05/2004

<sup>18</sup> Fecha de actualización de la página consultada: 12/05/2004

<sup>19</sup> **Tiflotecnología:** El conjunto de técnicas, conocimientos y recursos encaminados a procurar a los ciegos y deficientes visuales los medios oportunos para la correcta utilización de la tecnología con el fin de favorecer su autonomía personal y plena integración social, laboral y educativa, se conoce como Tiflotecnología.

Como material Tiflotécnico se entiende todo material específico para ciegos y deficientes visuales, desde los materiales más sencillos y de fácil manejo (bajo nivel de especialización), hasta los materiales que por su

➤ **INSTRUMENTOS ELECTRÓNICOS DE LECTURA Y ACCESO A LA INFORMACIÓN.**

Podríamos subdividir este apartado de la siguiente forma:

- ◆ Instrumentos para acceder a la información en una pantalla de ordenador:
  - Programas de ampliación de caracteres.
  - Lectores de pantalla.
- ◆ Instrumentos que permiten leer textos impresos:
  - Lupa-Tv.
  - Escáner y O.C.R.
  - Lectores Ópticos autónomos.
- ◆ Equipos autónomos de almacenamiento y proceso de la información:
  - Braille'n Speak.
  - Sonobrilie.

**MÁQUINAS DE ESCRIBIR E IMPRESORAS BRAILLE**

- **GRABADORES Y REPRODUCTORES DE SONIDO**
- **DICCIONARIO INFORMATIZADOS**
- **CALCULADORAS CIENTÍFICAS**

- Se considera también, de manera crítica, y a la luz de los elementos hasta acá reseñados, una solución desarrollada por un ingeniero de Capital Federal. El nombre del sistema es “Pizarra Interactiva” y fue desarrollado por el Ing. Campos. En los tópicos siguientes se comentan algunos aspectos de la herramienta que aportan al marco referencial de este proyecto.

Los principales puntos a tener en cuenta según la información considerada son los siguientes:

- ✓ Las soluciones tecnológicas aplicadas a la comunicación aumentativa en personas ciegas o disminuidos visuales, aprovechan en general el sentido del

---

especial complejidad requieren de un entrenamiento previo para su correcto manejo (alto nivel de especialización). ((Citado del documento referido))

oído para implementar interfaces con el usuario. Algunas otras -menos- se valen del tacto.

- ✓ La autonomía ofrecida al usuario de cara al uso de la herramienta es otro aspecto importante a considerar, según se observa también en los elementos considerados. Si bien muchos de los elementos no son comparables al presente proyecto, en cuanto a lo amplio y general de sus destinatarios, en contraste al alto grado de especificidad de la solución a desarrollar para Lucas, la mayoría de las soluciones disponibles ofrecen un importante nivel de independencia al usuario. Sin embargo, son soluciones, en su mayoría, dirigidas a personas ciegas (sin ninguna otra discapacidad), y es considerable que en la medida que las posibilidades de autonomía se reducen, o se suman nuevas discapacidades en el usuario, se reducen también las posibilidades de autonomía.
- ✓ Las soluciones autónomas existentes son aparatos bastante portátiles, generalmente alimentados con baterías, por lo que cumplen uno de sus principales objetivos: asistir al usuario con discapacidad sin restricciones de tiempo ni lugar. Por su parte, los desarrollados como dispositivos periféricos para una computadora no cuidan ese aspecto, pues la carencia de portabilidad está dada en este caso en el ordenador en sí mismo.
- ✓ Sobre el particular de la solución denominada “Pizarra Interactiva”, por sus características inherentes, sí considera el problema desde una perspectiva integral, y no de manera aislada como las demás. Es decir, trata con la multidiscapacidad del sujeto en este caso, consistiendo en una adaptación físico-ergonómica del tablero de acceso a la herramienta. No obstante, tiene importantes deficiencias técnicas, en varias dimensiones: tanto desde el punto de vista del hardware y su conectividad a la PC, como desde la perspectiva pedagógica y de desarrollo de las funciones del aplicativo, y sus posibilidades de escalabilidad.

#### ***4.3. Elementos tecnológicos disponibles para el desarrollo de una solución a medida***

Los aspectos en este sentido considerados fueron sólo referenciales, pues es tanto y tan amplio lo que pude considerarse que sin un foco específico, derivado

exclusivamente de la propuesta de sistema, resulta impracticable. Se consideraron las siguientes fuentes:

- Textos sobre Hardware y conexión de dispositivos al PC. Específicamente los aspectos relacionados a puertos de comunicación, tecnologías y protocolos: cableado y conexionado, electrónica específica e interfaces de programación de bajo nivel (low-level).
  - PC INTERNO 5  
Autor M. TISCHER, B. JENNRICH  
Editorial MARCOMBO
  - GUIA COMPLETA PARA PC  
Autor GILSTER RON  
Editorial MCGRAW-HILL
  - ARQUITECTURA DE EQUIPOS Y SISTEMAS INFORMATICOS  
Autor VALDIVIA MIRANDA CARLOS  
Editorial PARANINFO
  
- Información sobre productos y soluciones de software de bajo nivel de Síntesis de Palabra, y su disponibilidad de programación.
  - Documentación On Line - Motor de Búsquedas GOOGLE.COM
    - Información sobre Síntesis de Palabra por Ordenador
    - Información sobre Proveedores de Motor de Texto a Palabra
  - Documentación On Line - MSDN (Red de Desarrolladores de Microsoft)
    - Información sobre Microsoft Agent 2.0
  - Documentación On Line - Sitio de SoftVoice Text-To-Speech
    - Información sobre el motor SVTTS
    - Elementos técnicos en los documentos facilitados por el Sr. Joseph Katz, de SoftVoice Inc.
  
- Textos sobre lenguajes de programación que posibiliten el desarrollo de los aspectos de múltiple interfaz en una aplicación (gráfica y auditiva). Si bien el aspecto de doble interfaz corresponde a una decisión atendida a la propuesta y

es un elemento específico de diseño, se consideró su relevamiento por destacarse como aspecto fuerte desde el principio en el relevamiento del marco referencial de soluciones existentes y las necesidades a satisfacer del sujeto y del proceso.

- Textos de Programación en Visual Basic
  - PROGRAMACION AVANZADA CON MICROSOFT VISUAL BASIC 6.0  
Autor BALENA FRANCESCO  
Editorial MCGRAW-HILL
  - A VISUAL BASIC 6 PROGRAMMER'S TOOLKIT  
Autor MARQUIS HANK, SMITH ERIC  
Editorial APRESS
- Textos sobre Programación en Visual C/C++
  - PROGRAMACION EN C/C++  
Autor PAPPAS  
Editorial ANAYA MULTIMEDIA
  - EFFICIENT C/C++ PROGRAMMING  
Autor HELLER  
Editorial ACADEMIC PRESS
- Textos sobre Programación de la API / Windows Services
  - PROGRAMMING WINDOWS SERVICES  
Autor MORIN RANDY  
Editorial JOHN WILEY & SONS
  - WIN32 API PUZZLE BOOK AND TUTORIAL FOR VISUAL BASIC PROGRAMMERS  
Autor APPLEMAN DAN  
Editorial APRESS

Se contemplan en este marco referencial, a partir de los elementos indicados, los siguientes puntos:

- ✓ Interfaces electrónicas específicas de los dispositivos periféricos conectables a una PC. Se profundizarán, en función de los elementos indagados, aquellos que correspondan a los medios elegidos para posibilitar el intercambio de datos con requerido para dotar de accesibilidad al sistema a desarrollar.
- ✓ Componentes de software de síntesis de palabra, con sus correspondientes interfaces de programación. Se trabajará en detalle el relevamiento de

alternativas disponibles y se enfocarán los aspectos específicos de la que resulte seleccionada.

- ✓ Características y herramientas ofrecidas por los lenguajes de programación para dotar a la aplicación de interfaces duales: auditivas y gráficas, ajustando los objetos requeridos a las necesidades que se determinen en el perfil del sistema a desarrollar.
- ✓ Integración de funciones de reproducción de sonido y síntesis de palabra con interfaces gráficas.

#### **4.4. El ciclo de vida del proyecto y los modelos de análisis y diseño del sistema**

Los elementos de marco teórico y referencial en este tópico fueron considerados a partir de los textos que se mencionan a continuación:

- Respetto de los modelos de ciclos de vida en proyectos informáticos
  - “Lifecycle Planning Rapid Development” por Steve McConnell, editorial Microsoft Press.
- Respetto de los modelos de análisis y diseño de sistemas
  - UML GOTA A GOTA  
Autor FOWLER MARTIN y SCOTT KENDALL  
Editorial ADDISON-WESLEY IBEROA
  - INTEGRAR UML EN LOS PROYECTOS  
Autor LOPEZ NATALIE  
Editorial GESTION 2000
  - UML Y PATRONES – Introducción al Análisis y Diseño Orientado a Objetos y Proceso Unificado  
Autor LARMAN CRAIG  
Editorial PRENTICE-HALL
  - Documento electrónico titulado “Ejercicio de Aplicación de UML al Análisis de Sistemas”, escrito por Alexis Gigena y Juan Pablo Ferreira Centeno, disponible para alumnos y egresados del Colegio Universitario IES Siglo 21, en la dirección de Internet [http://www.ies21.com.ar/trabajos/tp\\_inf\\_CasoUML.pdf](http://www.ies21.com.ar/trabajos/tp_inf_CasoUML.pdf)<sup>20</sup>

---

<sup>20</sup> Fecha de descarga del documento consultado: 03/02/2005

- Trabajo de documentación de principios y directrices de diseño de interfaces de usuario publicado en la dirección Web [http://rss.utalca.cl/web/Estandares y Guias.doc](http://rss.utalca.cl/web/Estandares_y_Guias.doc)<sup>21</sup> en el que se compendian los diferentes estándares y recomendaciones de los principales referentes en la industria.
- Capítulo 4 del trabajo de Tesis del Lic. Héctor Aragón Cantú presentado ante la Universidad de las Américas – Puebla – México. El mismo se encuentra disponible en Internet, en la dirección Web [http://mail.udlap.mx/~tesis/lis/aragon\\_c\\_h/capitulo4.pdf](http://mail.udlap.mx/~tesis/lis/aragon_c_h/capitulo4.pdf)<sup>22</sup> El título de este cuarto capítulo es “Lineamientos de Diseño para Interfaces de Pantalla Amplia en UIML”, y resultan muy valiosos sus aportes en materia de Diseño Metafórico.
- Capítulos 1, 2 y 3 del texto de “Programación Lógica”, Ed. Colegio Universitario IES Siglo 21, del mismo autor del presente proyecto, tanto en su edición impresa como multimedial. Se enfocan en este trabajo especialmente los aspectos referidos a las técnicas de diseño de interfaces de usuario y a los atributos que las mismas deben reunir en pos de su simpleza y facilidad de uso.

El desarrollo del sistema objeto de este proyecto estará determinado por los siguientes elementos:

- ✓ Los aspectos propios del ciclo de vida del proyecto, de características de Prototipo Evolutivo.
- ✓ La metodología de análisis y diseño establecida por el estándar UML, como modelo de documentación.
- ✓ Los aspectos de diseño específicos en materia de interfaz de usuario, desde una doble perspectiva:

---

<sup>21</sup> Fecha de descarga del documento consultado: 03/02/2005

<sup>22</sup> Fecha de descarga del documento consultado: 03/02/2005

- La interacción del sujeto con la herramienta, mediante una interfaz auditiva
- La interacción de la maestra de integración y demás asistentes del sujeto en su aproximación a la herramienta y el desarrollo de la mecánica de lecto-escritura a través de la misma.

Estas dos interacciones consideradas juegan un rol fundamental en lo referido a la interacción sujeto-capacitador, de cara a la misma herramienta, según se señaló precedentemente.

## **5. Diagnóstico y Propuesta**

---

Para la elaboración del Diagnóstico y Propuesta del presente proyecto se consideraron los diferentes elementos recogidos en la etapa de indagación preliminar, a saber:

- Consideraciones pedagógicas, terapéuticas y tecnológicas reseñadas en el Marco Teórico Referencial.
- Elementos considerados en las entrevistas mantenidas con los terapeutas del sujeto (actuales y anteriores) y con su maestra especial de integración.
- Entrevistas con los padres del sujeto.
- Aspectos observados directamente sobre el sujeto y sus posibilidades motrices, de coordinación e intelectuales.
- Consideraciones sobre el relevamiento preliminar de recursos tecnológicos disponibles.

Sustentados en los elementos mencionados, se propuso el siguiente diagnóstico y su consecuente propuesta.

### **5.1. Diagnóstico**

Los puntos referentes a la problemática psicológica, terapéutica y pedagógica fueron establecidos de manera interdisciplinaria con los especialistas consultados, considerando la perspectiva informática de cara al presente proyecto.

El diagnóstico se sintetiza en los siguientes puntos:

- El sujeto dispone de la capacidad intelectual para utilizar herramientas de software de interfaz auditiva. Su estado emocional e inquietud intelectual lo predisponen de manera muy favorable para el logro efectivo de las habilidades de lecto-escritura planteadas como objetivo. Su capacidad de concentración y sus periodos atencionales no presentan mayores inconvenientes para el manejo de una herramienta de comunicación aumentativa que le posibilite el proceso de lecto-escritura.
- Sus posibilidades motrices no le permiten operar interfaces convencionales como por ejemplo un teclado convencional; no obstante, sí se encuentra en capacidad de reconocer texturas y operar una interfaz física simplificada. El

control motriz de sus manos y el manejo de su espacio de trabajo son muy buenos y le permiten trabajar sin problemas con herramientas físicas (interfaz).

- La utilización de una herramienta informática en el ámbito educativo del sujeto es viable, tanto en la escuela como en su casa. Dispone de los recursos tecnológicos suficientes, incluyendo un computadora portátil con capacidades de audio.
- Las herramientas tecnológicas disponibles posibilitan el desarrollo de una solución específica, combinando recursos de hardware y software, soportando interfaces físicas, auditivas y visuales.
- Las características pedagógicas de la herramienta a desarrollar están claramente definidas, lo que posibilita su implementación e integración al proceso de aprendizaje del sujeto.
- Los requerimientos de interfaz con los dos actores implicados en el proceso de aprendizaje (sujeto y maestra especial de integración), tanto de la herramienta como, mediante su uso, de la temática escolar específica, están claramente definidos.
- Los instrumentos utilizados anteriormente por el sujeto no cubren sus necesidades en relación a la asistencia en el proceso de lecto-escritura, pero su utilización previa facilita el acceso a la nueva herramienta a desarrollar.

## **5.2. Propuesta**

Coherentemente con los aspectos señalados en el diagnóstico, y teniendo en cuenta todos los elementos considerados en este apartado, los diferentes aspectos de la propuesta se detallan a continuación:

- Diseñar una interfaz física adecuada a las posibilidades motrices del sujeto, que le permita operar el sistema de lecto-escritura. Esto incluye:
  - Diseño conceptual del tablero o componente físico
  - Diseño e Implementación de la interfaz lógica de comunicación entre el tablero y la PC

- Diseñar y desarrollar el aplicativo de software que posibilite al sujeto el desarrollo de la lecto-escritura de manera independiente y autónoma, a la vez que su interacción con su docente especial de integración en dicho proceso. En particular, los siguientes se constituyen en subobjetivos específicos de esta propuesta:
  - Diseñar una interfaz auditiva que posibilite la operación del sistema, que se constituya en canal de información para el sujeto y que le proporcione la retroalimentación necesaria para su correcto uso.
  - Diseñar una interfaz gráfica que permita al entrenador del sujeto interactuar con el mismo en relación al objeto de uso – el sistema – posibilitando la intervención en el proceso de escritura con fines correctivos y la asistencia general en el uso de la herramienta.
  - Implementar en el sistema las funciones de lectura de texto mediante un software de síntesis electrónica de voz, aplicando dicha capacidad en las diferentes funciones y contextos de la aplicación, permitiéndole tanto la retroalimentación en el proceso de escritura como la lectura de cualquier texto digitalizado en el ordenador.
  - Posibilitar la integración del sistema desarrollado con otros programas computacionales, de manera que puedan traerse y llevarse hacia ellos los textos a los que el sujeto acceda o que el mismo produzca.
  - Facilitar la autogestión de la herramienta por parte del sujeto, independizándolo de asistencia externa para su uso.
  - Posibilitar la interacción e intervención directa en el sistema del entrenador del sujeto, tanto con fines correctivos en el proceso de escritura como así también con diferentes objetivos pedagógicos canalizados en dicha interacción.

### **5.3. Consideraciones Adicionales**

Desde el punto de vista interdisciplinario, el alcance de este proyecto excede sus límites, integrando mediante un trabajo interdisciplinario, los aportes y opiniones

profesionales de diferentes especialistas. En este sentido se prevén los siguientes aspectos:

- En la producción del tablero de operación para el sujeto se trabajará de manera interdisciplinaria con una diseñadora industrial (Natalia Hernández).
- En la interfaz electrónica de dicho tablero con el ordenador, se contará con la asistencia de un ingeniero electrónico (Diego Beltramone), relacionando los aspectos de construcción, cableado, diseño e implementación del dispositivo y su conexión al ordenador.
- En el trabajo de campo con el sujeto, y en el desarrollo de los aspectos operacionales y de requerimientos funcionales y pedagógicos del instrumento, se desarrollará un estrecho trabajo interdisciplinario con su maestra especial de integración, la Prof. Gabriela Migueltoarena. En este sentido, una particularidad del sistema a desarrollar será su múltiple enfoque de usuario, considerando no sólo las capacidades y desarrollo evolutivo del sujeto, sino también las observaciones e interacción con la herramienta de su maestra.

## 6. Estrategia Metodológica

---

Como estrategia metodológica para el presente proyecto se utilizarán distintos elementos de modelos de análisis y diseño de sistemas, como así también diferentes instrumentos técnicos / tecnológicos para su desarrollo e implementación.

En este sentido, se consideran los modelos y elementos para cada aspecto del desarrollo que más pertinentes resulten al proyecto, realizando adaptaciones Ad-Hoc de las herramientas e instrumentos propuestos en cada metodología considerada, de manera de aportar el mayor valor y al mismo tiempo no recargar la documentación con modelos o herramientas innecesarias.

Se presentan a continuación los diferentes elementos de la estrategia metodológica considerada:

### 6.1. *Ciclo de Vida*

Habiendo considerados los diferentes modelos de ciclos de vida, se considera que el más adecuado para este proyecto es el denominado “Ciclo de Vida de Prototipación Evolutiva”. El mismo consiste en el desarrollo del concepto del sistema a medida que se avanza en el proyecto. Esta característica es, en este caso, imprescindible, pues no puede saberse de antemano exactamente qué interfaces deberán generarse, ni que funciones específicas deberán proporcionarse en el software.

Según se definió anteriormente, un claro requisito para el sistema a construir es la ***interfaz dual simultánea***: el sistema deberá presentar una interfaz auditiva para su principal usuario, disponiendo al mismo tiempo de una interfaz gráfica simultánea y paralela, que le permita al entrenador del sujeto “observar” lo que el está gestionando con la herramienta, a la vez que intervenir en el proceso de escritura.

En consideración de lo anterior, el prototipado evolutivo proporcionará los mecanismos para desarrollar el ciclo de vida de manera exitosa al ir presentando a los usuarios (en este caso, por partida doble) cada elemento de diseño, posibilitando la retroalimentación a partir de sus consideraciones en un proceso de avance y refinamiento de características.

La utilización de este modelo de ciclo de vida se ve formalmente sustentada en que se cumplen, de manera particular y específica, las condiciones que recomiendan su aplicación:

- Los requerimientos no están completamente definidos en el aspecto detallado y operativo del sistema a construir. Si bien están claramente especificadas las características y propiedades que deberá reunir la herramienta, los aspectos específicos en materia de funciones detalladas e interfaces con los usuarios no se encuentran completamente definidos.
- Las especificaciones detalladas no pueden ser aportadas por los usuarios mediante ninguna de las técnicas convencionales de relevamiento, pues ni ellos mismos son completamente concientes de las capacidades que pueden obtenerse de la herramienta. El proceso de definición será iterativo y de avance, en la medida que vayan siendo expuestos a cierta funcionalidad podrán comprender las capacidades implícitas y solicitar detalles en materia de prestaciones e interfaz.
- Del mismo modo tampoco se puede, desde el punto de vista informático, establecer de antemano las funcionalidades detalladas. Como se acaba de señalar, el conjunto final de funciones detalladas, características específicas y prestaciones de la herramienta resultará de la confluencia de necesidades pedagógicas y posibilidades técnicas / tecnológicas.

Concientes de las características de este tipo de ciclo de vida, resultará muy difícil y carente de precisión cualquier estimación de tiempos de desarrollo e implementación que se pretenda establecer. En este sentido se establecerá una Línea Base cuando el conjunto inicial de prestaciones esté disponible, depurado e implementado, y a partir de ese momento, se avanzará siguiendo este modelo de ciclo de vida generando nuevas “releases”<sup>23</sup> de la herramienta al ir incluyendo nuevas características que vayan resultando de la evolución intelectual y pedagógica que se vaya produciendo en el sujeto con el paso del tiempo.

## **6.2. Metodología de Análisis y Diseño de Sistemas**

Dadas las características del presente proyecto, tanto en materia de relevamiento de necesidades como en lo referente a las funciones a desarrollar que deben ser analizadas y diseñadas, resultan muy particulares las implementaciones directas de modelos de análisis y diseño clásicos, fundamentalmente por la fuerte impronta de la problemática empresarial, o cuando menos informacional, que se observa en las mismas. Precisamente por esto, la metodología a emplear debía, ante todo, reunir como característica fundamental su uniformidad u universalidad de aplicación.

Considerando el estado del arte actual en materia de análisis y diseño de sistemas informáticos, las tendencias, y las características antes mencionadas, se observa que el lenguaje UML<sup>24</sup>, específicamente mediante algunos de sus modelos, reúne las características necesarias para el desarrollo del presente proyecto. Su utilización soporta el ciclo de vida de prototipado evolutivo definido, a la vez que las interrelaciones de sus modelos - sobre todo cuando son soportados por alguna herramienta CASE<sup>25</sup> - permite mantener la integridad conceptual del producto a desarrollar.

En este sentido, de los indicados por la metodología, serán utilizados de manera principal los siguientes modelos:

---

<sup>23</sup> Release: Término (en inglés) utilizado para referir en número de versión de un producto de software liberado al usuario, el cual reúne una serie de características específicas adicionales respecto de la versión anterior del mismo.

<sup>24</sup> UML: Lenguaje de Modelado Unificado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables. ((Definición tomada del texto “El lenguaje unificado de modelado” de los autores Booch, Rumbaugh y Jacobson, creadores del UML. Editorial Addison Wesley. 1999))

<sup>25</sup> Herramienta CASE (Computer Aided Software Engineering): Conjunto de programas de software que posibilitan el desarrollo de los modelos de ingeniería de software. Es un software que asiste al informático en el análisis, diseño, documentación y mantenimiento de un sistema informático.

- Modelos de Casos de Uso: Permiten referenciar cada instancia de uso de la solución, indicando los actores involucrados y la funcionalidad a resolver en cada situación. Es un modelo adecuado para explicar la aplicación requerida, de manera funcional, en cada caso de uso del sistema, valga la redundancia.
- Modelos de colaboración de objetos: Su utilización permitirá definir específicamente la interrelación de los componentes de interfaz, proceso y acceso a datos de los procesos modelados. Las especificaciones establecidas en los mismos sirven de especificación técnica para la programación, complementando su referencia con los prototipos previstos.
- Modelos de componentes: Las herramientas para modelar los componentes permitirán las especificaciones de interrelación y despliegue de los mismos de cara a la implementación. Son un enfoque arquitectural de las piezas que componen en conjunto la solución.
- Modelos de estados y secuencias: Para los casos en que la secuencia de comandos, o acciones sobre la interfaz lo requiere, estos modelos permitirán llevar un efectivo control del estado de la aplicación a cada momento y la interrelación de objetos de programación en cada momento. Su utilización es sugerida precisamente por la modelización de dichos estados, como se detalla más adelante.

El lenguaje UML define muchos otros modelos, y su utilización quedará supeditada a su efectivo aporte a la claridad conceptual del análisis, diseño y modelo de implementación de la herramienta. En principio se consideran como imprescindibles los anteriores. Los propios autores del Lenguaje Unificado de Modelado señalan la importancia de seleccionar inteligentemente los modelos a utilizar, sin incurrir en el error de querer utilizarlos a todos, pues sólo complica las cosas congestionar con diferentes modelos el mismo objeto de sistema.

### **6.3. Diseño de Interfaces de Usuario**

Los aspectos a considerar en el diseño específico de las interfaces de usuario deben reunir una serie de características específicas, alineadas en dotar a las mismas de la usabilidad requerida, tanto para su usuario principal como para su maestra integradora

En este sentido, caben señalarse como más destacables las siguientes:

- Homogeneidad: Las interfaces deben representar las mismas categorías siempre del mismo modo. A similares estados, similares mensajes (tanto visuales como auditivos).
- Intuitividad: Esto se aplica más específicamente a las interfaces gráficas del proyecto, por cuanto deben proveerse elementos representativos que permitan a su usuario (visualmente capaz) operar el sistema por transferencia de sus conocimientos previos en el uso de interfaces de computadora.

En cuanto a la intuitividad de la interfaz auditiva, no son tan fuertes los elementos que pueden reutilizarse pues son muy pocos los objetos con este tipo de interfaz a los que el sujeto ha estado expuesto. No obstante, en consonancia con la homogeneidad antes descrita, el producto será intuitivo para Lucas en la medida que se respeten los mismos patrones durante todo su diseño.

- Balance, Equilibrio y correcta distribución: esta característica también es aplicable a cada modelo de interfaz, pero nuevamente con mayor incidencia en el modelo gráfico que en el auditivo. En el primero de ellos, en adecuado manejo del espacio y la adecuada distribución de los controles GUI<sup>26</sup> aportará a la usabilidad de la interfaz, manteniendo organizada e intelectualmente accesible toda la información en la pantalla. Por su parte, la equivalencia auditiva de este concepto, pasa por procurar el mayor equilibrio posible en los mensajes audibles que reciba el usuario. Un sistema de menú auditivo con estructura homogénea y opciones bien distribuidas es una aplicación del balance y equilibrio referido. Mensajes consistentes y opciones con similares estructuras permitirán cumplir este objetivo en materia de interfaz de audio.
- Ajuste a estándares y códigos universales: Esto sólo tiene aplicación en cuanto a la interfaz gráfica. Seguir los estándares de diseño de interfaces GUI<sup>27</sup> será

---

<sup>26</sup> Controles GUI (Graphical User Interface): Se denomina así a cada uno de los componentes gráficos de un entorno o interfaz GUI. Cada botón, cada icono, cada etiqueta de texto es un control.

<sup>27</sup> Interfaces GUI (Graphical User Interface): El conjunto de elementos, físicos o virtuales, de percepción sensorial (normalmente visual y auditiva) que posibilitan la interacción hombre-máquina, disponiendo de un

productivo en razón del aporte a la usabilidad anteriormente mencionado. En cuanto a la interfaz auditiva, como se señaló, no existen antecedentes significativos para el sujeto en este sentido, por lo que sus modelos previos de retroalimentación / control de dispositivos y aparatos son insignificantes. Esto posibilitará desarrollar un diseño en procura de la optimización del tiempo requerido para operarlo y su facilidad de uso, que más que ajustarse a estándares, defina en este sentido los propios.

- Retroalimentación visual y auditiva: La retroalimentación al usuario sobre cada una de sus acciones sobre la herramienta es fundamental, tanto en su interfaz visual, como en la auditiva. En este punto se debe prestar especial atención, considerando que el único mecanismo de feedback disponible para el Usuario No Vidente es, en el caso de esta solución, los mensajes e indicadores de estado que vaya recibiendo de manera auditiva del sistema. Asimismo, la capacidad de síntesis de palabra y su utilización en la herramienta, no sólo en los procesos de lectura, sino también como retroalimentación o “reacción” ante cada “acción” se constituye en el principal elemento o canal para lograr los objetivos de usabilidad antes mencionados.

Un elemento particularmente importante en materia de diseño es que, al tratarse de un software de asistencia al sujeto en el proceso de lecto-escritura, permitiéndole crear y reproducir documentos de texto electrónicos, todas las funciones implementadas y disponibles en el mismo estarán gobernadas por el “estado” de la aplicación. Se mencionó anteriormente que los momentos o estados en que se encuentra el programa determinarán el conjunto de acciones o funciones a los que podrá acceder el usuario principal. Esto resulta natural como aspecto diferencial, pues existe una enorme diferencia en materia de información suministrada al usuario en una interfaz gráfica que en una interfaz auditiva: el usuario de interfaz gráfica literalmente está “viendo” la pantalla completa, en la que puede observar multitud de elementos gráficos y textuales que le permiten comprender cada

---

ambiente multimedial, especialmente determinado por el monitor gráfico, un puntero sobre el mismo (mouse) y demás elementos accesorios.

aspecto de estado en el programa, entendiendo por estado los datos gestionados, el documento en creación y el conjunto de funciones disponibles. Si embargo, un usuario ciego que sólo cuenta con la interfaz auditiva del programa, deberá “oír” todo lo necesario para tomar la misma conciencia que el usuario con capacidades visuales plenas respecto del “estado” de la aplicación. Esto no será tarea sencilla... Es muy cierto aquello de que “una imagen dice más que mil palabras”; no obstante, manejando estados discretos y bien definidos, y ofreciendo la retroalimentación auditiva necesaria, siguiendo los lineamientos y propiedades que deben reunir las interfaces de usuario efectivas -pero extrapolando el medio visual al auditivo-, podrá lograrse una correcta comprensión del estado aplicativo por parte del usuario. Ese es precisamente el mayor desafío de este proyecto!

#### **6.4. Modelo de Prototipado**

Considerando el tipo de ciclo de vida del proyecto a desarrollar, y las particularidades en materia de requerimientos y funcionalidades a implementar, la correcta utilización, seguimiento y control de los prototipos propuestos resulta fundamental. En este sentido se prevé un mecanismo de observación y evaluación muy detallado del sujeto frente al prototipo, especialmente en lo que a su interfaz auditiva se refiere. De esta manera, a través de las observaciones “de campo” del usuario (sus acciones y especialmente sus “reacciones” sobre el objeto) podrán establecerse, en discusión interdisciplinaria con su equipo terapéutico/pedagógico, los ajustes, adaptaciones e implementaciones funcionales que sean requeridas para maximizar el provecho que pueda ser obtenido del uso de la herramienta.

Del mismo modo, la interfaz gráfica de la solución deberá tener igualmente cuidados los aspectos de diseño, para lo cual el prototipazo posibilitará que su maestra de integración, padres y demás asistentes, puedan interactuar de la mejor manera con la herramienta. Serán también considerados en las pruebas de campo, fundamentalmente por dos cuestiones:

- El asistente del sujeto deberá observar en pantalla, de manera clara, ágil e intuitiva, los elementos necesarios para comprender y seguir paso a paso la experiencia de uso del usuario principal.

- Quienes asistan a Lucas en el uso de la herramienta serán, a su vez, actores de la solución: no sólo indicando al usuario las acciones requeridas en la fase de aprendizaje y aprestamiento; sino también interactuando con él, frente a la herramienta, en la corrección “de proceso”, frente al desafío de lograr automaticidad y autonomía en la mecánica de lecto-escritura, como fuera señalado en los objetivos del presente proyecto.

Finalmente, cabe considerar que dadas las características evolutivas del sujeto, expuesto a una experiencia de aprendizaje que va mucho más allá de la herramienta en sí misma (se trata precisamente de aprender todo lo necesario en la escuela, no sólo escribir, sino también gramática, matemática, ciencias y demás elementos curriculares...) cada release del producto se convertirá, a su vez, en prototipo funcional para la determinación de futuras necesidades. El objetivo final del proyecto incluye la mantenibilidad de la solución en el tiempo, ajustándola y ampliándola a las nuevas necesidades que vayan siendo detectadas por el equipo profesional que trabaja con el sujeto.

**DESARROLLO**

## 7. Estructura de Desarrollo

---

Se plantea para el desarrollo del presente Trabajo de Aplicación Profesional una estructura de integración tecnológica de diferentes componentes, acorde a la solución requerida.

Los componentes referidos son los siguientes:

➤ **Interfaz física.** *Capítulo 8.*

Se detallan los aspectos de diseño y distribución de componentes para el acceso físico del usuario a la herramienta desarrollada. Sus dimensiones y esquema funcional.

➤ **Interfaz electrónica y comunicación con el software.** *Capítulo 9.*

Se establecen las características de la interfaz electrónica entre la interfaz anterior y la computadora, reseñando las diversas evoluciones que la misma sufrió durante el desarrollo del proyecto y específicamente detallando las actuales características de dicha interfaz.

➤ **Síntesis de voz por software y construcción de una API independiente para la utilización de la misma.** *Capítulos 10, 11 y 12.*

Como componente central del proyecto, la capacidad de síntesis de voz generada en la PC se explora, en cuanto a las tecnologías disponibles, benchmarking de diferentes motores y demás aspectos, hasta la selección del motor en cuestión para este desarrollo y puntualmente, el diseño y producción de un software de middleware que actúe de interfaz entre el motor de síntesis fonética y la aplicación del usuario final, con el agregado de las mejoras y correcciones de errores y falencias en el entorno nativo del motor.

➤ **Software de aplicación e Integración de componentes.** *Capítulo 13.*

Se presenta el análisis y diseño de la solución de software propiamente dicha, utilizando los modelos metodológicos en la aplicación específica, y detallando todas y cada una de las funcionalidades provistas por la aplicación.

Finalmente, se presentarán las conclusiones del trabajo en el capítulo 14.

## 8. Interfaz Física

---

En el diseño de la interfaz física para el usuario de la herramienta debieron considerarse, entre otros, los siguientes aspectos:

- Posibilidades motrices y sensoriales del sujeto
- Portabilidad, materiales y disposición física
- Aplicabilidad a las características del software a desarrollar
- Experiencias previas del sujeto

La solución consistiría fundamentalmente en un **teclado o tablero especialmente desarrollado para Lucas** siguiendo estas consideraciones, sus capacidades y los requerimientos de usabilidad para el software a desarrollar.

Respecto de cada uno de estos tópicos, se mencionan a continuación los aspectos más relevantes.

### ***8.1. Posibilidades Motrices y Sensoriales de Lucas***

Como se detalló en las etapas precedentes de este trabajo, las posibilidades motrices del usuario son muy buenas en el sentido grueso, lo que hace viable una interfaz de control manual. Sin embargo, no deben obviarse dos aspectos funcionales de Lucas que condicionaron el diseño de la interfaz:

- Su tono muscular es relativamente bueno, aunque en el particular de la utilización de sus dedos no es capaz de ejercer demasiada fuerza sobre los objetos con los que interactúa.
- La motricidad fina que puede desarrollar con sus dedos es muy pobre. Si bien se mantienen en sus aspectos terapéuticos ejercicios específicos para desarrollarla, la posibilidad de operar individualmente los movimientos de sus falanges es baja.

Cabe también considerar que su sentido del tacto se encuentra muy bien desarrollado, lo que le permite reconocer superficies y texturas.

En cuanto a las capacidades auditivas del usuario, las mismas son óptimas. Tanto su respuesta a frecuencias audibles como a la intensidad de los sonidos son completamente normales.

En síntesis: *puede oír bien, reconocer texturas, y oprimir botones grandes utilizando la yema de todos los dedos de la mano de manera conjunta, no disociada.*

## **8.2. Portabilidad, materiales y disposición física**

La frecuencia esperada de uso de la solución es considerablemente alta: varias horas por día, al menos cinco días por semana. Esto obedece a la utilización de la herramienta en toda actividad de lecto-escritura a desarrollar por el usuario; tanto en el ámbito escolar como doméstico.

También resulta importante considerar la necesidad de portabilidad de la solución, dada precisamente por el hecho de disponer del software correspondiente a la misma en al menos tres computadoras diferentes: la NoteBook y la PC de escritorio de su familia, y la computadora asignada para su uso en la escuela. Si bien la recomendación profesional es disponer de un teclado específico para su escuela y otro para su casa, la posibilidad de reemplazar un tablero por otro ante la eventual rotura de uno de ellos es fundamental; de ahí deviene su requisito de portabilidad.

Finalmente, también se consideró el riesgo de rotura por accidentes, caídas o golpes. Considerando estas posibilidades en un niño de la edad del usuario, los materiales a utilizar debían de ser resistentes, pero al mismo tiempo flexibles, para posibilitar su operación.

## **8.3. Aplicabilidad a las características del software a desarrollar**

La interfaz debía diseñarse de acuerdo a las características de entrada (INPUT) requeridas en la aplicación de software a desarrollar. En este sentido, se previeron dos aspectos diferentes y concurrentes:

- La aplicación debería permitir la pulsión individual de los 6 (seis) puntos que componen cada carácter Braille, más dos pulsadores adicionales utilizados para indicar la aceptación y cancelación respectivamente del carácter en curso.

- Las diferentes acciones a desarrollar con la aplicación en términos de lecto-escritura requerirían poder disponer de algún mecanismo de control y selección de comandos, tanto para las funciones específicas de edición de texto como para el desplazamiento en el texto y las operaciones de lectura. En este sentido resultaba adecuado pensar en un sistema de menús auditivos, controlables desde el mismo teclado especial a desarrollar. Para el desarrollo de estas funciones se aprovecharon los restantes ocho botones.

Podía darse satisfacción a estos dos aspectos de múltiples maneras, organizando los botones del tablero a desarrollar de diferentes maneras, según principios de ergonomía, usabilidad, frecuencia de uso de cada botón (en condiciones de operación normales de la herramienta), etc... No obstante, los aspectos considerados en el apartado siguiente marcaron de manera específica las características de diseño a respetar.

#### **8.4. Experiencias previas del sujeto**

Lucas contaba ya con algún tiempo de experiencia en el uso de uno de los productos mencionados en la indagación de herramientas tecnológicas disponibles: la “Pizarra Interactiva”. Las características técnicas de esta herramienta la hacían no recomendable desde varios puntos de vista:

- Facilidad para romperse: los materiales con que estaba construida eran demasiado rígidos y habían tenido ya episodios de rotura por caídas accidentales.
- Dificultad de conexionado universal: se conectaba a la computadora utilizando el puerto paralelo y el puerto de Joystick, lo que impedía de conectar dispositivos adicionales como impresoras, palancas de juego, y cualquier otro periférico que utilizara tales puertos de comunicación. Esto traía aparejado como complicación adicional que la frecuente acción de conectar y desconectar dispositivos terminara por falsear o romper los conectores.
- Al tratarse de un producto adaptado, sólo funcionaban algunos de los botones de la misma, confundiendo al usuario respecto de su utilización.

- Las prestaciones del software de esta herramienta eran sumamente precarias: sólo consistía en posibilitar la pulsión de los puntos Braille y el mensaje grabado asociado a cada uno, **sin posibilitar de modo alguno la actividad de lecto-escritura planteada como objetivo.**

No obstante, su familiaridad con la disposición física de sus botones y en especial la mecánica de reconocimiento de los botones mediante texturas, hicieron al equipo terapéutico de Lucas recomendar el diseño de un tablero de similares dimensiones, aunque con una aplicación completamente diferente.

### **8.5. Conclusión y diseño final de la interfaz física**

Considerando los aspectos hasta aquí señalados, las características de la solución propuesta y desarrollada son las siguientes.

#### **Dimensiones y Construcción**

El tablero tiene unas dimensiones totales de 35 cm de ancho por 24 cm de fondo. Sobre el se encuentran dispuestos 16 áreas rectangulares delimitadas por separadores. Cada una de estas áreas tiene 7,5 cm de ancho por 4,5 cm de alto. Debajo de cada una de estas áreas se encuentra instalado un pulsador tipo microswitch de alta sensibilidad, hermético. En adelante se denominará “botón” a cada una de estas áreas, pues de hecho, ofician de tales.

La separación entre los botones está sobreelevada respecto de la altura normal de los mismos, lo que permite distinguirlos al tacto. Dichas separaciones son todas de 1 cm de ancho.

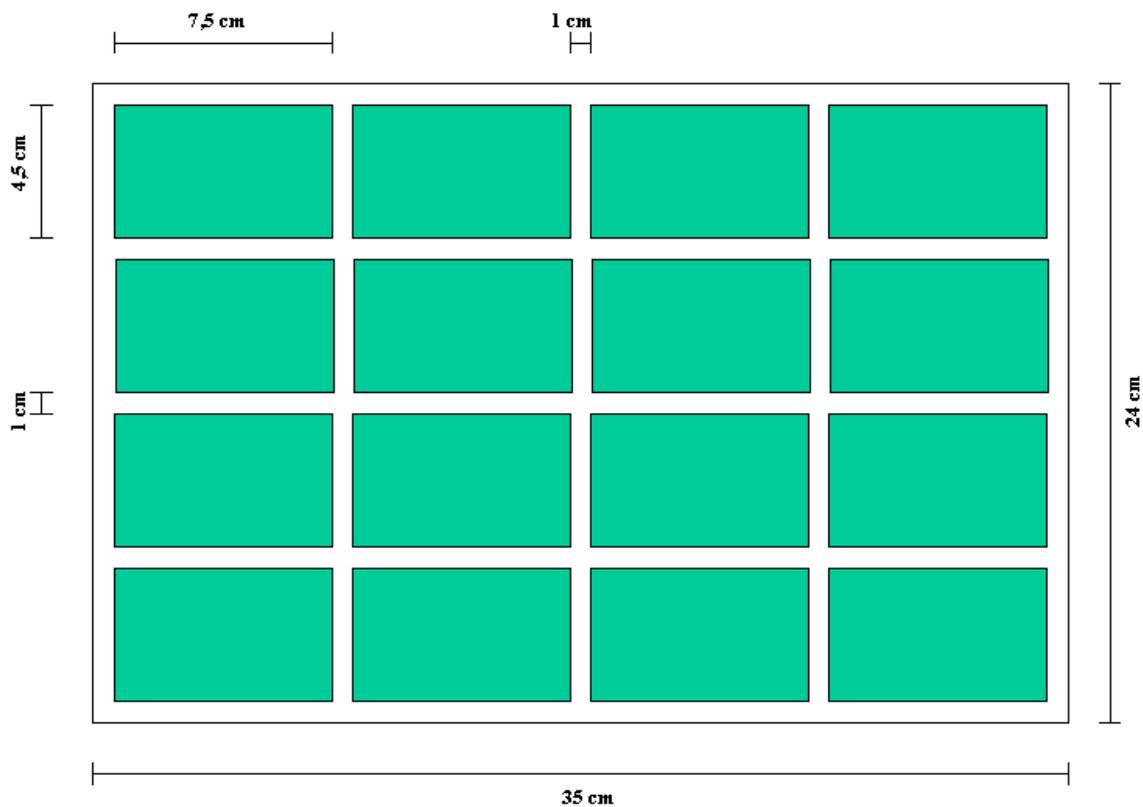
La altura total del teclado es de 1,7 cm, suficientes para poder albergar a los microswitchs mencionados más el recorrido de los cables que conectan a los mismos, en la cavidad entre las dos placas con que está construido.

La placa inferior (base) es de plástico de alto impacto de mayor rigidez que la placa superior (superficie de los botones). De este modo, el dispositivo completo soporta la presión ejercida sobre los botones sin problemas. Sobre la placa inferior, para dar estructura al dispositivo, se encuentran dispuestos tabiques de madera, pegados a la placa, que soportan los separadores anteriormente mencionados.

Estos tabiques posibilitan organizar cómodamente los manojos de cables que recorren todo el tablero en su interior, desde cada uno de los pulsadores hacia el circuito electrónico que se encarga de efectuar al interfaz con la PC.

Finalmente, los cuatro bordes del tablero se encuentran cubiertos con una canaleta flexible, resistente a golpes, que permite sujetar el conjunto a la vez que darle robustez.

Se presenta a continuación un esquema con las dimensiones referidas indicadas.



Este tablero de prototipo, con sus dimensiones y materiales, fue construido por la Dis. Ind. Natalia Hernández, quien colaboró desinteresadamente en la realización de este proyecto.

Cabe señalar que posteriormente a la implementación del proyecto, se fabricaron otros dos tableros, todos de idéntica funcionalidad, a los efectos de contar con el BackUp necesario por eventuales roturas.

Se presentan a continuación algunas fotos del tablero descrito, y su cableado interior.



Primer Prototipo: Vista en Perspectiva



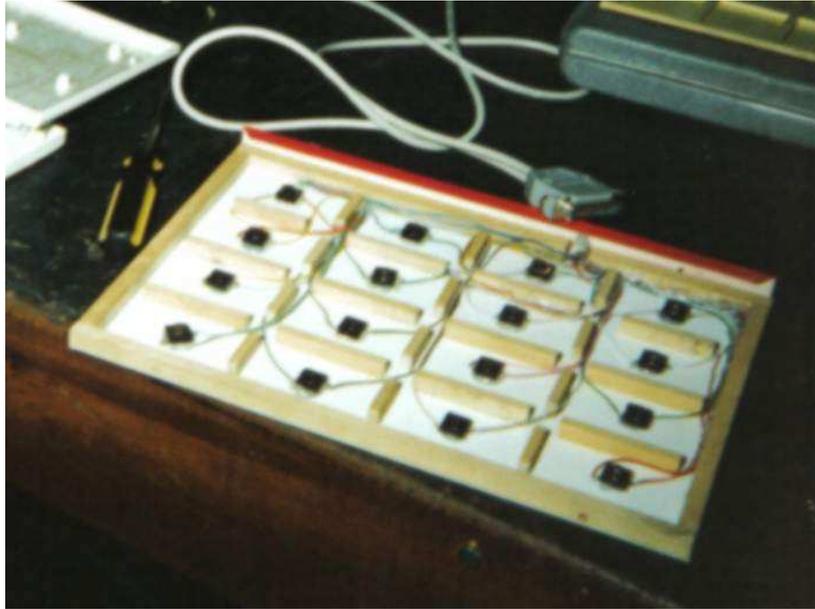
Primer Prototipo: Vista Superior



Primer Prototipo: *Vista Frontal*



Primer Prototipo: *Banco de Trabajo – Sin cobertura*



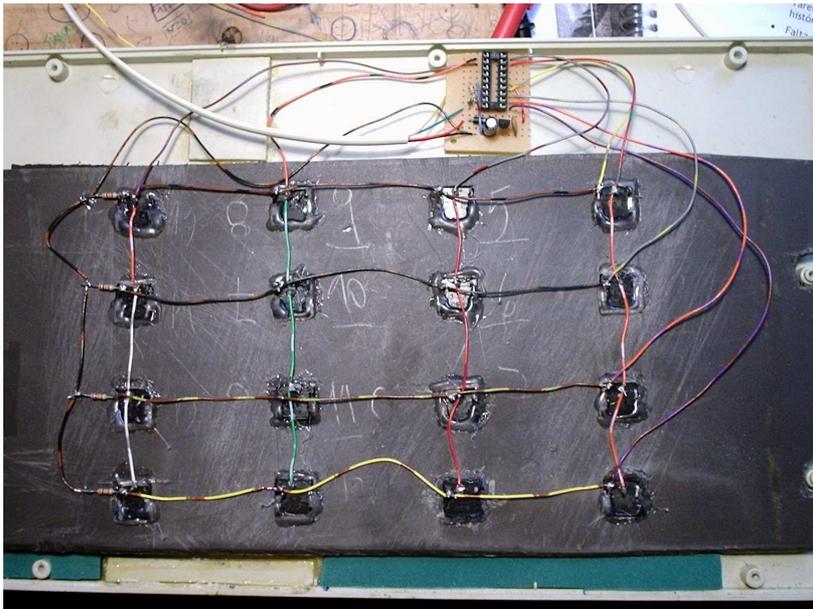
Primer Prototipo: *Conexión Interno Microswitchs*



Prototipo Final: *Vista en Perspectiva*



Prototipo Final: Vista en Perspectiva



Prototipo Final: Matriz conexionado

## **Conexión del Tablero a la PC**

Los aspectos electrónicos de la interfaz entre este tablero y la computadora personal (PC) o NoteBook serán descriptos en detalle en el siguiente capítulo; no obstante, fue parte de las directrices de diseño consideradas que la comunicación con la computadora debía ser cableada con la suficiente independencia, para que el tablero pudiera disponerse a comodidad del ambiente de trabajo del usuario.

Considerando que el modelo de interfaz completo involucra no sólo el tablero sino también la retroalimentación o Feedback auditiva con el usuario, y que la misma se implementaría mayormente con auriculares conectados al ordenador, la longitud del cable del tablero debía al menos ser de 2 metros, para evitar inconvenientes de conexión. Esto debió ser también considerado en los aspectos electrónicos, como se detallará posteriormente.

## **Disposición Funcional de los Botones**

Los botones requeridos fueron organizados aprovechando la distribución espacial del tablero, en cuanto a las características mencionadas anteriormente.

La disposición matricial de 4 (cuatro) filas por 4 (cuatro) columnas permite separar los botones en dos grupos funcionales:

- **La mitad izquierda** (dos primeras columnas: 8 botones) utilizados para el proceso de escritura, utilizándolos para la referenciación de los 6 puntos Braille más los botones para Aceptar el carácter especificado, y el de Cancelar. Sobre la funcionalidad de este último botón cabe señalar que debiera ser contextual, en el siguiente sentido: si el usuario se encuentra en proceso de escritura de un carácter Braille, el botón cancela su edición; en tanto que si no se encuentra editando, sino que acaba de aceptar un carácter y descubre que no es el que quería ingresar, con este botón eliminará dicho carácter del texto. Es un funcionamiento similar al BackSpace de un teclado convencional.
- **La mitad derecha** (dos últimas columnas: 8 botones) utilizados para la gestión del menú de la aplicación en cuanto a sus aspectos de

configuración y desplazamiento sobre el texto, pero fundamentalmente, para comandar el proceso de “lectura” mediante la voz generada sintéticamente por el ordenador.

Se presenta a continuación un esquema con la funcionalidad diseñada para cada uno de los botones:

<b>Botón #1</b>	<b>Botón #5</b>	<b>Botón #9</b>	<b>Botón #13</b>
<b>Botón #2</b>	<b>Botón #6</b>	<b>Botón #10</b>	<b>Botón #14</b>
<b>Botón #3</b>	<b>Botón #7</b>	<b>Botón #11</b>	<b>Botón #15</b>
<b>Botón #4</b>	<b>Botón #8</b>	<b>Botón #12</b>	<b>Botón #16</b>

**Botón #1** : Punto Braille 1

**Botón #2** : Punto Braille 2

**Botón #3** : Punto Braille 3

**Botón #5** : Punto Braille 4

**Botón #6** : Punto Braille 5

**Botón #7** : Punto Braille 6

**Botón #4** : Espaciador Braille → Aceptar el carácter tecleado

**Botón #8** : Cancelar Carácter / Eliminar último carácter tecleado (carácter a la izquierda del cursor)

**Botón #9** : Función de Menú (según contexto)

**Botón #10** : Función de Menú (según contexto)

**Botón #11** : Función de Menú (según contexto)

**Botón #12** : Función de Menú (según contexto)

**Botón #13 :** Función de Menú (según contexto)

**Botón #14 :** Función de Menú (según contexto)

**Botón #15 :** Función de Menú (según contexto)

**Botón #16 :** Leer Menú Actual (independiente de contexto)

### **Texturas aplicadas sobre los botones**

Como se mencionó anteriormente, las posibilidades sensoriales del usuario y su experiencia previa hacían recomendable, al menos en la instancia inicial, la utilización de texturas sobre la cara de los 16 botones del tablero.

Estas texturas debían de ser fácilmente identificables para Lucas, por lo que la definición de las mismas estuvo a cargo de su terapeuta: Prof. Esp. Gabriela Migueltoarena.

Las texturas aplicadas fueron las siguientes:

Lija	Goma Espuma	Felpa	Botón
Cartón coarrug.	Cartón Liso	Cañamazo	Goma coarrugada
Goma EVA	Paño lenci	Palito de helado	Plástico rayado
Cinta	Lana	Globitos	Goma con puntitos

## 9. Interfaz Electrónica y comunicación con el Software

---

El tablero especial desarrollado en el capítulo anterior debía conectarse a la PC para poder detectar desde el software a desarrollar la pulsión de los diferentes botones que lo componen. En este sentido se consideraron las siguientes premisas, en términos de diseño:

- El total de señales diferentes a detectar es de 16 (correspondientes a cada uno de los botones)
- No se prevén en el funcionamiento de la herramienta la pulsión simultánea de dos o más botones
- Es necesario detectar el estado neutro: no pulsión de ningún botón
- La interfaz electrónica, por cuestiones de seguridad, debía montarse dentro del tablero, saliendo directamente desde el mismo con un cable hacia la PC
- La longitud del cable para conectar el tablero a la PC debía cuando menos ser de 2 mts, siendo recomendable que el mecanismo de comunicación soportara una distancia aún mayor, para evitar cualquier pérdida de señal significativa.
- El hardware específico debía poder conectarse a un puerto de comunicaciones del PC lo más estandarizado posible, de modo que pudiera estar presente en los diferentes computadores en los que se conectara la solución.
- En la medida de lo posible, era deseable que el hardware específico desarrollado no requiriera de una fuente de alimentación externa, sino que la alimentación del circuito electrónico de interfaz fuera adquirida a través del mismo cable.

Analizando los requisitos anteriores, de manera interdisciplinaria con el Ing. Diego Beltramone - quien colaboró en esta etapa del proyecto en el diseño e implementación de la interfaz electrónica -, se consideraron de manera inicial tres puertos diferentes:

- Puerto Paralelo (Centronix)
- Puerto de Teclado
- Puerto Serial (RS-232)

En el primer análisis, se diseñó un circuito electrónico digital compuesto de compuertas NAND y NOR que implementaba la multiplexación de las dieciséis señales diferentes en

cuatro bits. Se utilizarían los cuatro bits entrantes del puerto paralelo (disponible en cualquier PC). De este modo,  $2^4=16$  combinaciones posibles. No obstante, este diseño no llegó jamás a implementarse, principalmente por los inconvenientes detectados en la posterior etapa de análisis detallado:

- Si bien son 16 las señales diferentes a recibir, existía un decimoséptimo estado a considerar: la no pulsión de ningún botón.
- La longitud total recomendable para un cable de conexión al puerto paralelo está al límite de las necesidades, lo que arrojaba un margen de seguridad demasiado pequeño.
- El tamaño, costo y complejidad del circuito a construir era considerable, teniendo en cuenta las alternativas.

A consideración de lo anterior, se decidió por tomar la segunda de las opciones: utilizar el puerto de teclado.

### **9.1. Primer prototipo: Interfaz electrónica vía teclado.**

La propuesta de utilizar el conector del teclado arrojaba visibles ventajas, como por ejemplo la de contar con la circuitería completa a muy bajo costo.

Sin embargo, esta alternativa implicaba en primera instancia un problema: deberían conectarse de manera alternativa el teclado normal, o el tablero especial. Esto iba en contra de una de las características deseables de la aplicación: que pudieran interactuar con la herramienta su usuario principal (Lucas) y quien estuviere asistiéndolo en su uso (normalmente su maestra).

Considerando este problema, se planteó una solución: sobrecablear la circuitería específica de un teclado normal, disponiendo los cables pegados al chip de montaje superficial que realiza toda la tarea específica en un conector externo, adosado al mismo teclado. De este modo, la solución consistiría en un teclado normal “adaptado”, con un conector externo adicional, al que se conectaría el tablero especial desarrollado para Lucas.

### 9.1.1. Implementación

Esta alternativa de interfaz electrónica se implementó. El teclado utilizado fue el modelo KB6055 de BTC, distribución Latinoamericana, de interfaz mini-DIN (PS/2).

La distribución de pines del microchip de control del teclado elegido permitía detectar la pulsión física de cualquier tecla mediante el cortocircuito que la misma produce sobre las diferentes patas o microbornes del circuito. De este modo, y según el esquema de conexionado desarrollado sobre el circuito, las diferentes pulsiones sobre los microswitchs del Tablero diseñado para el usuario producían los cortocircuitos correspondientes sobre el chip de control del teclado, de manera que se emulara la siguiente distribución de teclas, detectables desde el software a desarrollar.

“8”	“4”	Espacio	BackSpace
“7”	“3”	“L”	“S”
“6”	“2”	“K”	“D”
“5”	“1”	“J”	“F”

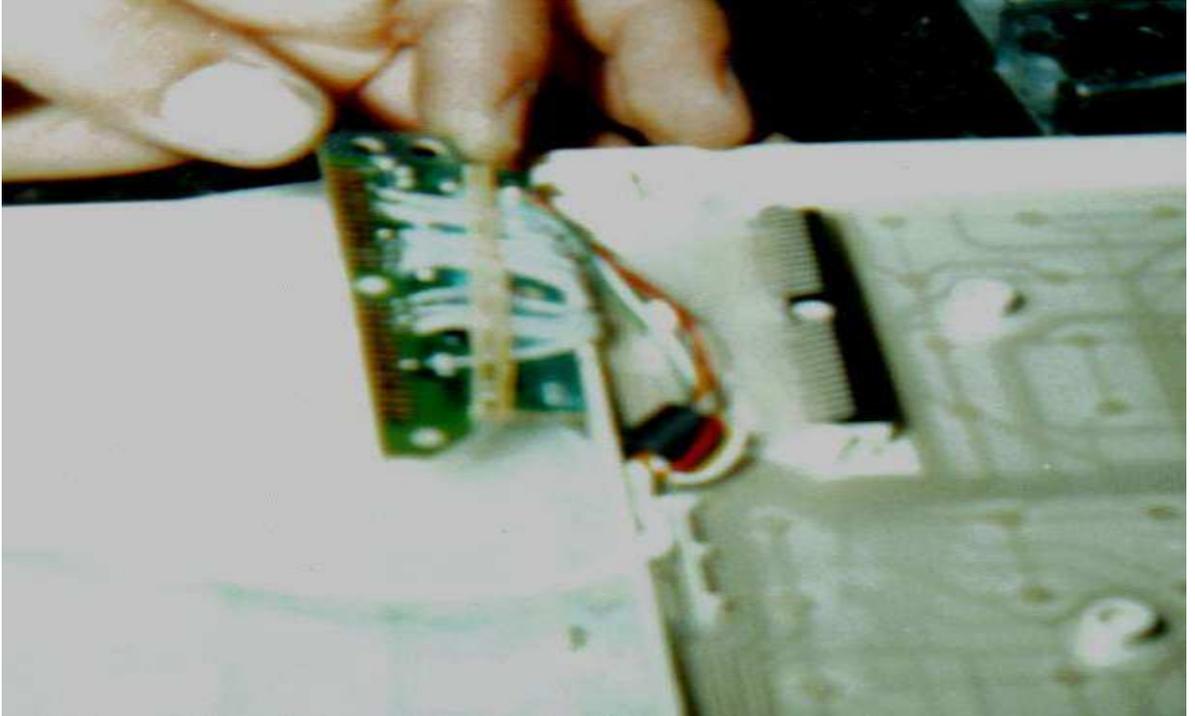
**Distribución de caracteres para interfaz vía Teclado**

Dicho de otro modo, cuando el usuario pulsaba el botón correspondiente a la esquina superior izquierda de su Tablero, la computadora recibía la señal como si se hubiera pulsado la tecla “8” del teclado normal; si se pulsaba el botón de la esquina superior derecha, la entrada de teclado detectada es la correspondiente a la tecla de BackSpace del teclado normal.

El prototipo fue utilizado durante algún tiempo, y si bien funcionaba correctamente, se presentó un inconveniente: la naturaleza de las conexiones dispuestas con montaje superficial sobre la circuitería del teclado era extremadamente delicada, lo que produjo que ante golpes y

caídas involuntarias del mismo produjeran falsos contactos y la aplicación fallara.

Se presenta a continuación una imagen del sobremontaje realizado:



**Montaje Superficial sobre los pines del circuito de teclado**

Ante este inconveniente, se decidió optar finalmente por la tercera alternativa: puerto de comunicaciones seriales COM.

### **9.2. Prototipo Final: Interfaz vía puerto COM**

Si bien esta era la más cara de las alternativas, el costo de sus componentes seguían siendo despreciable en contraste a los beneficios de su aplicación.

Para el desarrollo de esta interfaz se utilizó un Microcontrolador Programable (PIC) con la lógica interna de comunicación vía protocolo RS-232 del puerto Serie ya disponible en la misma circuitería.

La estrategia electrónica consistió en un scanner activo que recorriera el conexionado de Filas y Columnas del tablero para detectar las colisiones. De este modo, se determina con precisión el botón pulsado a cada momento.

El intervalo (frecuencia) de Barrido es altísimo: del orden de los microsegundos, por lo que cualquier pulsión, por breve que fuera, sería detectable sin inconvenientes.

Cuando el circuito diseñado detecta la pulsión de una tecla, envía a través del puerto serie a la PC un carácter específico, codificado en ASCII.

La tabla de caracteres recibidos según sea el botón presionado, se detalla a continuación:

“0”	“1”	“2”	“3”
“4”	“5”	“6”	“7”
“8”	“9”	“A”	“B”
“C”	“D”	“E”	“F”

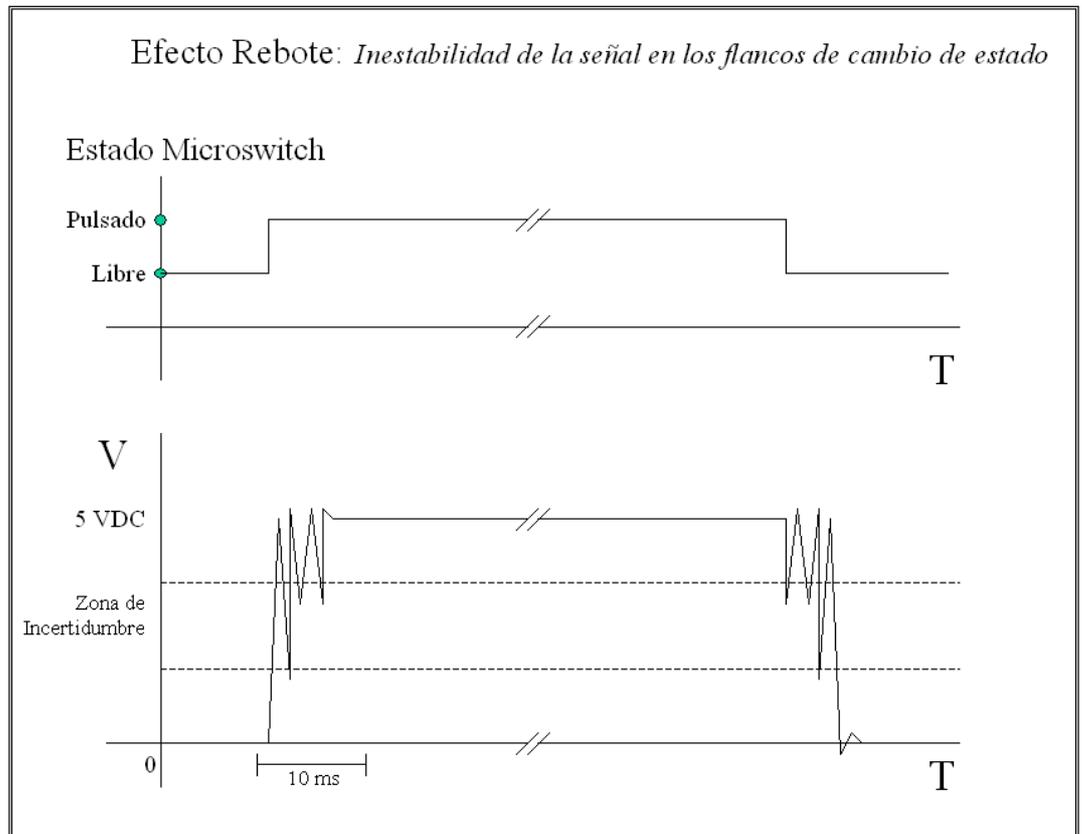
**Distribución de caracteres para interfaz RS-232**

### **9.3. Mecanismo Antirrebote y Antirrepetición**

Una problemática a considerar en esta interfaz es la identificación de cada pulsación voluntaria, y su no repetición automática.

Cada pulsión voluntaria sobre cualquiera de los botones produce un “rebote electrónico” inicial al ser pulsado, cuyo tiempo de inestabilidad puede superar el

intervalo o frecuencia de barrido de coordenadas. Esto es visible en el siguiente gráfico:



A consideración de lo anterior, era posible que el puerto serie de la PC recibiera dos o tres caracteres consecutivos correspondientes en realidad a una misma acción del usuario. Esto ocasionaría su posible mala interpretación desde el software, afectando la funcionalidad prevista para la interfaz.

Adicionalmente, el tiempo durante el cual permaneciera pulsado cada botón resultaba variable y difícil de controlar, pues dependía del aprestamiento y seguridad del usuario. El software no debía interpretar una pulsión de larga duración como varias pulsiones diferentes, pues en realidad la acción de pulsar un botón es única y no repetible, hasta tanto el mismo no se suelte; al menos para el software que se diseñaría.

Para solucionar estos dos inconvenientes las alternativas analizadas fueron dos:

- Implementar un mecanismo Antirrebote / Antirrepetición por firmware en el mismo PIC

- Implementar un mecanismo Antirrebote / Antirrepetición desde el software de captación de datos del puerto serie en la PC, coordinando un protocolo preestablecido con el Hardware específico.

Consideradas ambas posibilidades, se decidió por la segunda de ellas, pues aportaba dos ventajas significativas:

- Simplificaba la complejidad en la programación del PIC (de mucho más bajo nivel que el aplicativo sobre la PC, y por ende, más costosa)
- Permitiría futuras ampliaciones o reutilizaciones de la interfaz de Hardware, como por ejemplo, la utilización del mismo tablero para implementar la interfaz con otros programas que se desarrollaran en el futuro, y sobre los que existían algunas intenciones previamente consideradas: implementar un software musical (pianito electrónico, batería, etc...), juegos de control activo, etc... Si bien estas aplicaciones exceden el objetivo del presente trabajo, tenerlas en cuenta fue un elemento definitorio a la hora de establecer el protocolo entre Firmware (PIC) y Software (PC).

### *9.3.1. Parámetros de configuración Puerto COM*

Para la configuración de los atributos de protocolo serial RS-232 se convinieron con el diseñador del circuito electrónico los siguientes parámetros:

Velocidad de Transmisión: 2400 baudios (bits por segundo)

Bits de Datos: 8

Bit de Paridad: Ninguna

Bit de Stop: 1

Ventana de datos: 1024 Bytes

Sentido de transmisión: Simplex (sólo del dispositivo a la PC)

Comunicación: Asíncrona (no requiere Handshaking)

Encontramos después de algunas pruebas que esta configuración permitía una comunicación entre dispositivos lo suficientemente rápida y segura.

Si bien las características de los puertos seriales involucrados (tanto el del PIC con el de la PC) posibilitaban la transmisión de datos a una tasa de transferencia mucho mayor, la misma no aportaría beneficio alguno, dado que las señales a transmitir eran pequeñísimas en cuanto a la información transmitida: simplemente un carácter por pulsión. En tanto, haber incrementado la velocidad de transferencia sólo habría producido el inconveniente de reducir la estabilidad en la señal transmitida o cuando menos habría limitado la longitud máxima aceptable para el cable de interfaz.

En cuanto a las características específicas de las ráfagas de datos a recibir desde el dispositivo periférico diseñado, se detallan en la siguiente sección sus aspectos más relevantes.

### *9.3.2. Protocolo definido para la comunicación entre el dispositivo y el software*

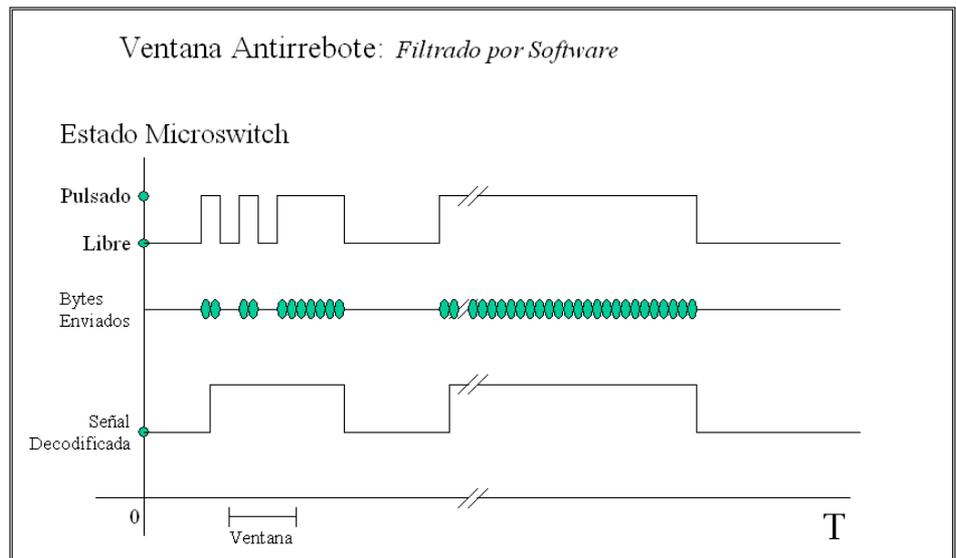
Considerando no sólo su utilización en esta aplicación, sino también posibles utilidades del dispositivo en otros programas a desarrollar para el usuario, y teniendo en cuenta la simplicidad ya referida en términos de programación del PIC, se decidió establecer un protocolo de Ráfaga de datos ante presencia de señal, implementando los algoritmos antirrebote en el mismo software cliente del lado de la PC.

Así, el PIC envía a través del puerto serie a la PC cadenas continuas de caracteres correspondientes a cada pulsión de teclas en el tablero. En otras palabras: mientras un botón o switch del tablero esté pulsado, el PIC envía a la PC el carácter que le corresponde al mismo de manera continua, hasta que el botón sea soltado. Si se pulsan simultáneamente dos botones, no se envía ningún carácter.

Por su parte, el software de captura de señales del puerto COM del lado de la PC se diseñó y construyó de manera que partiendo de una situación inicial de “no señal”, cuando un dato arriba por el puerto serie se toma su tiempo de entrada (Ticktime) para considerarlo en el algoritmo antirrebote desarrollado. De este modo, para que el siguiente dato

entrante sea considerado por el sistema como válido, deberá haber transcurrido un tiempo mínimo establecido y configurable -llamado Ventana- sin datos entrantes en el puerto para que el siguiente dato sea considerado como válido. Así, si por error de uso se pulsa el mismo botón dos veces demasiado próximas en tiempo, o si una simple pulsión genera por la misma característica de rebote en el microswitch dos señales interpretadas por el PIC como dos pulsaciones, las mismas son tan próximas en tiempo que no superan la ventana, y por ende se interpretarán como una única acción.

Visto en un gráfico, la situación de señales producidas, datos enviados por el PIC a la PC, y datos considerados, puede representarse del siguiente modo:



Cabe destacar que la Ventana definida es variable y configurable desde la aplicación, de modo que puede ajustarse a la máxima sensibilidad que le permita al usuario operar el tablero tan rápidamente como le sea posible y que al mismo tiempo no produzca interpretaciones erróneas por parte del software de aplicación sobre sus acciones.

## **10. Síntesis de voz por software**

---

En materia de motores de palabra, en la investigación desarrollada se determinó que existen básicamente tres modelos funcionales diferentes. Cada solución tecnológica existente en este campo, tanto en materia de software como en soluciones híbridas – combinando hardware especializado –, hace uso de al menos uno de esos modelos. Los mismos son:

- ✓ Fonación por vocablos almacenados predefinidos (con o sin posibilidades de expansión, contextuales o libres de contexto)
- ✓ Fonación por combinación silábica
- ✓ Fonación por síntesis matemática de fonemas

A continuación se desarrollan brevemente los conceptos tecnológicos que los sustentan, para poder luego justificar el modelo elegido para el desarrollo del trabajo.

### ***10.1. Fonación por vocablos almacenados predefinidos***

Esta técnica consiste, básicamente, en tener almacenados un conjunto de vocablos preestablecido (comúnmente denominado como “diccionario”). La función del aplicativo consiste, en estos casos, en identificar en la base de datos cada palabra a pronunciar y simplemente reproducirla mediante la interfaz de audio disponible.

Cabe señalar que esta fue la primera de las implementaciones tecnológicas en este campo que se desarrolló, y que desde sus orígenes hasta la actualidad, de la mano del crecimiento de las capacidades de hardware, los tiempos de procesamiento y el rápido acceso a grandes volúmenes de datos, se fue enriqueciendo notablemente. En la actualidad, los sistemas de estas características, posibilitan la adición de nuevos vocablos al diccionario, haciéndolos dinámicos. Otras características inteligentes provistas a esta tecnología consiste en la detección de contexto, para que una misma palabra escrita sea pronunciada de manera diferente según el contexto en que se encuentra.<sup>28</sup>

---

<sup>28</sup> Sensibilidad de Contexto en sistemas de vocablos predefinidos. Esto se aplica particularmente en otros idiomas, como el Inglés, en el que una misma palabra puede variar su pronunciación en función del contexto

Las principales debilidades de este tipo de sistemas son:

- Alto costo operativo: el hardware y software requeridos son considerables, a la vez que su desarrollo implica la grabación de prácticamente todas las palabras de una lengua. Obviamente esto no se alcanza ni cerca, por lo que cabe la siguiente debilidad.
- Limitaciones para vocablos desconocidos: permanentemente hay que estar ampliando el vocabulario que difícilmente sea completo. Esto incluye como desventaja que se pierda la uniformidad en la voz que se escucha, obteniendo generalmente mezclas muy desagradables al oído al combinar en una misma oración audible la voz de dos o más personas, pronunciando cada una de ellas una palabra diferente.

### ***10.2. Fonación por combinación silábica***

Atacando las principales debilidades del sistema anteriormente mencionado, estas otras soluciones tecnológicas estriban en un concepto diferente. Todas las lenguas se componen de cierto conjunto finito y pequeño de sílabas fonéticas posibles. Básicamente, este tipo de soluciones identifican dichas sílabas en las palabras que deben pronunciar, haciendo un análisis o scanning de la letras que componen cada palabra, y luego ponen en secuencia de audio la pronunciación de los recortes de fonemas, denominados “difonos”, que constituyen cada vocablo, separando obviamente las palabras. Estos “difonos” son recortes de la voz real grabada, organizados para poder ser combinados en secuencia procurando la mejor pronunciación, en cuanto a tono, timbre, velocidad y demás características sonoras.

A diferencia de lo que su nombre permitiría intuir, cada “difono” no constituye la pronunciación recortada de un sílaba del lenguaje, como fonema; sino que por el contrario se trata del recorte de audio que va del centro de un fonema al centro del siguiente. De este modo, la parte más crítica del aspecto sonoro de nuestra

---

gramatical en que se encuentra. En castellano no tiene tanta importancia, excepto por la entonación en las preguntas, que es una característica común incluso a otras lenguas.

fonación, que es precisamente la concatenación de sílabas con los diferentes matices que la anterior y la próxima aportan a cada palabra queda relativamente bien resuelta.

El principal inconveniente tecnológico presente en el anterior modelo queda aquí resuelto. El tamaño de la base de datos de fonemas requerido es en el orden de miles de veces más pequeño que en el caso de la anterior solución, por cuanto su velocidad de respuesta y sus tiempos de acceso se simplifican. Uno puede tener almacenados registros de diferentes voces, dotando al sistema de la posibilidad de pronunciar según diferentes voces pregrabadas. No obstante, la construcción de una de estas bases de datos es un proceso bastante complejo y costoso, por cuanto la variedad es bastante reducida.

Estos sistemas cuentan con algunas desventajas, principalmente relacionadas al origen de las lenguas. Puntualmente, esta limitación no es demasiado seria para lenguas como el Castellano o el Portugués, en las que la pronunciación de las diferentes letras es siempre la misma; sola excepción hecha de algunas reglas de pronunciación claramente definidas para algunas combinaciones de letras. Pero para el caso de lenguas anglosajonas, sirvan como ejemplo el Inglés o Alemán, sucede que este reconocimiento no es directo en todos los casos. Por ello, se hacen imprescindibles diccionarios de excepciones a las reglas fonéticas de pronunciación, que están ya implementados en este tipo de soluciones.

Adicionalmente, sucede que no todos los difonos muestreados encajan correctamente, por lo que se producen en la fonación gorjeos, falsas consonantes y otros efectos sonoros indeseables, que confunden la comprensión de las palabras pronunciadas.

### ***10.3. Fonación por síntesis matemática de fonemas***

Finalmente, este último modelo se aparta radicalmente de los anteriores. No consiste en lo absoluto en la reproducción de sonidos prealmacenados en el sistema, ni de su combinación. Por el contrario, esta nueva alternativa, consiste en la síntesis matemática de las funciones que generan las formas de onda principal y armónicas que definen cada sonido de la fonación humana. El sonido resultante a estas funciones es sintetizado en la placa de audio del sistema, secuenciando los

diferentes sonidos “formantes”, constituyendo en tiempo real los fonemas de cada palabra.

Estos sonidos “formantes” son los resultantes de la forma que adopta todo el tracto vocal de los seres humanos para producir cada sonido de nuestro lenguaje oral. Es a lo largo de todo el tracto vocal (desde las cuerdas vocales, pasando por la garganta, la lengua y hasta los labios) que se producen ciertas frecuencias resonantes mayores, correspondientes a cada parte involucrada en la pronunciación de cada consonante o vocal. Estas frecuencias cambian en la medida que cambia la forma y configuración muscular del tracto vocal, por lo que la técnica consiste en identificar cada uno de estos “formantes” en la lengua (descompuesto como la producción simultánea de tres frecuencias resonantes asociadas a la principal), hacer corresponder la secuencia de ellos a cada clase de sonido producido para constituir cada fonema, y componer la síntesis final.

Los sintetizadores de voces disponibles implementando esta técnica de fonación simulan el mecanismo de producción de la fonación humana utilizando osciladores digitales (para producir las frecuencias mencionadas), fuentes de ruidos y filtros sonoros, logrando de ese modo la síntesis de cada sonido formante.

Este método tiene como principal ventaja la posibilidad de darle al sonido producido las características que puedan imaginarse. Sólo es cuestión de establecer el correspondiente modelo matemático generador de las funciones involucradas y ponerlo en obra. Pueden definirse literalmente millones de voces diferentes, en función a características como la frecuencia media, el timbre, velocidad de síntesis, y demás elementos.

Asimismo, la principal desventaja de este método estriba en la necesidad de una alta capacidad de cálculo a alta velocidad, pues para lograr resultados aceptables hacen falta del orden de millones de operaciones de punto flotante por segundo. Afortunadamente, la capacidad de cálculo de los procesadores actuales cubre con creces esa necesidad, por lo que puede sintetizarse voz mediante esta técnica en una computadora actual, sin notar un sacrificio de performance importante.

#### ***10.4. Particularidades requeridas para el aplicativo en materia de fonación***

Acorde al objetivo final definido para el aplicativo, la tecnología de fonación a utilizar deberá ser capaz de pronunciar de manera natural cualquier combinación de letras, sin que deban existir en este sentido limitaciones silábicas. Es esperable que en su proceso de desarrollo del lenguaje escrito, el usuario combine letras que no correspondan exactamente con las sílabas posibles del lenguaje castellano, lo que generaría serios inconvenientes al no poder oír exactamente lo escrito.

En función de lo señalado, se tornan inadecuadas las dos primeras técnicas señaladas, por lo que la solución a desarrollar deberá pasar necesariamente por la utilización de un motor de síntesis matemática de los fonemas, que posibilite la correcta lectura castellana de cualquier combinación de letras.

#### ***10.5. Implementación Tecnológica del Motor***

Como el objetivo desarrollar una API independiente de acceso a las funciones de fonación del motor de palabra, la implementación del mismo tendrá que ser tan simple y ágil como sea posible. En este sentido, se observa que la forma más efectiva de lograr las características necesarias es mediante la disponibilidad e instalación en el sistema, como parte de su conjunto de funciones de API extendida, de las bibliotecas de funciones compiladas relacionadas a todos los aspectos de configuración y fonación del motor.

De esta manera, la implementación posibilita una interfaz limpia e independiente, permitiendo el acceso a las diferentes funciones requeridas desde cualquier lenguaje de programación que soporte los tipos de datos y el protocolo de llamada de funciones de 32 bits según el modelo de API de cualquier Sistema Operativo Windows de 32 bits.

#### ***10.6. Disponibilidad***

En materia de características de disponibilidad del motor a implementar, resultan muy importantes las características que se detallan a continuación, en función de la utilización que para el mismo se prevé:

✓ **Estabilidad del servicio**

La estabilidad en la ejecución es quizá la más importante característica de disponibilidad requerida. Considerando que el aplicativo final está dirigido a una persona no vidente, no es una opción que el software no sea estable. Trasladando el ejemplo, sería como si en un sistema de interfaz gráfica el monitor sólo funcionara a veces! El servicio debe estar disponible “siempre”, o al menos tan próximo de esa meta como sea posible.

✓ **Independencia de Dispositivo**

En relación a las características particulares de hardware de diferentes placas de audio, el motor a implementar debe ser capaz de integrarse a cualquiera de estos dispositivos, independientemente de las características particulares que unos u otros posean. En tal sentido, la interfaz con el hardware de sonido será una de alto nivel, accediendo a sus funciones específicas a través de la interfaz provista por Windows en su Capa de Abstracción de Hardware (HAL<sup>29</sup>), para todas las aplicaciones que sobre este sistema operativo corran.

Este requisito se sustenta en la necesidad planteada, a futuro, de poder distribuir libremente las aplicaciones que utilicen los servicios del motor independientemente de las características técnicas o la configuración de hardware que posea cada uno de sus destinatarios.

✓ **Interfaz de Aplicación Win32**

---

<sup>29</sup> HAL : Siglas por “Hardware Abstraction Layer”. Es la designación que reciben los componentes del sistema operativo que se encargan de aislar las particularidades específicas de cada dispositivo de hardware, cubriendo los drivers específicos y brindando acceso a sus funciones mediante una interfaz unificada para los programas de aplicación.

Resulta requisito de especificación la plataforma operativa sobre la que habrá de desarrollarse el software. En tal sentido, considerando que se desarrollará sobre la familia de sistemas operativos de 32 bits de Microsoft, el motor a seleccionar deberá ser instalable y ejecutable sobre este entorno, implementándose como un programa ejecutable Win32. Adicionalmente, considerando las características de ciertas configuraciones antivirus y la sobrecarga de procesos, resulta recomendable que los servicios sean implementados como funciones encapsuladas compiladas en archivos DLL<sup>30</sup> de acceso directo para aplicaciones Win32, en tanto se respete el protocolo de llamada.

### **10.7. Control Aplicativo**

En cuanto a las características de control del motor desde los aplicativos que se desarrolle, el mismo deberá ofrecer funciones que como mínimo garanticen las siguientes categorías de funciones:

#### **✓ Control de Estado**

*Tiene que posibilitar el monitoreo del estado del motor desde el código fuente del aplicativo. Es decir, debe ser posible establecer si el motor se encuentra activo, si tiene algún problema de conexión con los componentes, si está con una fonación en proceso o está detenida, etc... Tiene permitir interrumpir un proceso de fonación abruptamente, antes de que termine naturalmente, pues es también requisito en función de las características generales de la categoría de aplicaciones a desarrollar. Obviamente, deberá posibilitar el inicio de*

---

<sup>30</sup> DLL : Siglas por “Dynamic Linked Library”. Extensión que identifica a las Bibliotecas de Enlace Dinámico en el ambiente Windows. Se trata de conjuntos de funciones compiladas y empaquetadas en un archivo que no se monta hasta tanto no se requieran sus funciones, las que pese a estar compiladas, no están enlazadas a las aplicaciones que las usar, sino que el enlace se produce en tiempo de ejecución, mediante las interfaces provistas a tal efecto por el sistema operativo. Esta característica le aporta al modelo una gran dinamicidad y capacidad de portabilidad y distribución, sin necesidad de recompilar a cada modificación.

*fonación de una expresión castellana (o al menos de pronunciación tal) tan corta o extensa como sea necesario.*

✓ **Control de Características Audibles de la Fonación**

*Debe posibilitar el control desde el código del aplicativo de las diferentes características anteriormente mencionadas en este trabajo en relación a los atributos que definen cada voz, como lo son el tono, el timbre, la velocidad de pronunciación, el nivel de naturalidad u oscilación de frecuencia en la pronunciación, etc.*

✓ **Capacidad para leer en castellano números con decimales y otros signos frecuentes**

Esta característica es también requerida, pues en función de los textos a procesar con el motor, será necesario que el usuario oiga cada símbolo presente en la gramática original, excepción hecha de los signos naturales de puntuación como la coma, punto y coma, dos puntos y punto. Estos últimos son en verdad pronunciados al realizar en la lectura las correspondientes pausas que indican.

Sucede que muchas veces, la fonación de los símbolos presentes en un texto lo hace comprensible. Por ejemplo, si el motor menciona la apertura y cierre de paréntesis cada vez que los encuentra en un texto, el usuario (oyente) tendrá conciencia de la existencia de tales signos y el lugar que ocupaban en el texto, dando a su contenido una importancia interpretativa acorde a su rol gramatical. En tanto que si dichos signos no hubieran sido mencionados, el usuario podría incluso malinterpretar su contenido.

El ejemplo no se limita a los paréntesis. También sucede con otros signos, y particularmente con las secuencias de números. Si al encontrar dos o más dígitos numéricos adyacentes en el texto se leyeran de manera individual, y no pronunciando la lectura castellana del número resultante, la capacidad de comprensión y la unificación de código lingüístico con otras personas sin

inconvenientes visuales se vería seriamente afectada. Por ejemplo, imaginemos que el usuario esté oyendo la lista de ingredientes de una receta, la que incluye 250 gramos de manteca, y se diera con que escucha algo como “dos cinco cero gramos de manteca”, en vez de oír lo que en verdad debió, como “doscientos cincuenta gramos de manteca”.

Evidentemente habrán sobrados ejemplos en los que es el criterio y el discernimiento de contexto lo que nos indica como se leen las cosas en castellano, pero el objetivo es que al menos en la mayor parte de las situaciones posibles el motor siga dichos patrones.

## **11. Motores de Palabra preseleccionados**

---

En función del objetivo del presente trabajo, y de los diversos aspectos señalados precedentemente en relación a los tipos de sistemas informáticos de síntesis de palabra o lectura sintética, y de las características desarrolladas como requerimientos, se desarrolló una investigación de productos que se ajustaran a los requerimientos mencionados.

### **11.1. Búsqueda, Preselección**

La principal fuente de datos en este sentido fue la propia red Internet, en la que se buscó de manera amplia e intensa utilizando servicios de búsqueda como yahoo, altavista, etc... Aunque la más productiva de las búsquedas fue la alcanzada utilizando Google.com.ar

Una vez identificado un conjunto amplio de desarrollos tecnológicos en esta materia, se fueron realizando descartes según diferentes criterios:

- ✓ Primero se descartaron soluciones que requerían de Hardware específico.
- ✓ Luego fueron separados los motores desarrollados para ambientes propietarios y otras plataformas.
- ✓ Después se eliminaron de la lista candidata los que no correspondían a la categoría de sintetizadores matemáticos de voz
- ✓ Finalmente, fueron separados aquellos cuyas implementaciones sólo estaban disponibles para sistemas fonéticos de lenguas foráneas (la mayoría hallada sólo soportaba Inglés o Alemán)

Se produjo un análisis de diferentes productos / aplicativos, con el objetivo de probar el cumplimiento en los mismos de las características señaladas como requeridas en los apartados correspondientes de este trabajo.

Algunos de los productos más relevantes analizados fueron los siguientes:

- ✓ Cool Speech
- ✓ DigIt
- ✓ Xylek's Text Reader
- ✓ Microsoft Agents Sample

- ✓ Just Talk
- ✓ SoftVoice Demonstration

En realidad se trata de productos diferentes, desarrollados por distintos fabricantes, e incluso con distintos objetivos; pero la mayoría de ellos comparten una característica en común: Utilizan el motor Lernout & Hauspie® TruVoice Text-To-Speech Engine, protocolizando su uso mediante la API de Speech para motores compatibles desarrollada por Microsoft, denominada SAPI (versión actual: 4.0). Este es el motor oficial de Microsoft para el uso de sus funciones con los servicios de Agentes (Microsoft Agent).

Por otra parte, un producto que llamó la atención por la claridad verbal de la pronunciación del motor y la estabilidad fue un aplicativo de ejemplo desarrollado utilizando el motor de palabra SVTTS, de la empresa SoftVoice Inc.

En síntesis, a partir de los aspectos preliminares de selección y de las características requeridas para los motores, fueron preseleccionados motores, para su comparativa más detallada:

- ✓ Lernout & Hauspie® TruVoice Text-To-Speech Engine, protocolizando SAPI 4.0 de Microsoft Corp.
- ✓ SVTTS de SoftVoice Inc.

Ambos motores fueron comparados en función de los diferentes criterios señalados, detallando el informe de dicha comparativa en el siguiente apartado.

## ***11.2. Comparativa de características de los motores (Benchmarking)***

### ***11.2.1. Consideraciones sobre SAPI y el motor Lernout & Hauspie® TruVoice Text-To-Speech Engine***

Cabe señalar que en todos los casos en que se verificó la utilización de Lernout & Hauspie® TruVoice Text-To-Speech Engine como motor en Background para la implementación de los diferentes aplicativos, se pudo observar en los programas un nivel de estabilidad inaceptable. Todos, sin excepción, sometidos a pruebas de uso intensivas, produjeron

errores. Lo que no pudo establecerse es si el problema radica en el motor propiamente dicho o en la interfaz de programación con la que se lo gestiona desde las aplicaciones (la llamada SAPI). Lo cierto es que no pudo hallarse información sobre el motor separada de su utilización por parte de SAPI, y más concretamente, su programación para dotar de voz a los agentes Microsoft Agents.

### *11.2.2. Consideraciones generales sobre el motor SVTTS de SoftVoice Inc.*

Las dos implementaciones analizadas para este motor se comportaron de manera muy estable. No se observaron requerimientos de componentes preinstalados y el proceso de instalación resultó tan simple como la descarga de los archivos y su ejecución directa.

Este motor no es compatible con el protocolo SAPI, por lo que definen sus propias funciones de gestión, configuración y control, implementando para ellas su propio protocolo.

No obstante lo escueta de la información disponible en la Web de la compañía, al tomar contacto con su representante, el Sr. Joseph Katz, la información recibida fue abundante y detallada. La compañía me proveyó del Kit de Desarrollo (SDK) y de toda la documentación necesaria para el análisis e implementación del motor.

### *11.2.3. Ambientes de Prueba de los motores: Especificaciones técnicas*

Los motores fueron probados en tres ambientes de prueba diferentes, con distintas características de Hardware y Software de Base. Los equipos en los que se realizó el testing fueron computadoras del estándar IBM PC-Compatible, con las siguientes configuraciones:

#### ➤ **Equipo #1**

Procesador Intel Pentium de 100 MHz  
64 MB de memoria RAM  
Disco duro de 1 GB  
Placa de sonido SoundBlaster Pro  
Monitor, Teclado y Mouse estándar  
Sistema Operativo: Microsoft Windows 95

- **Equipo #2**
  - Procesador Intel Celeron de 450 MHz
  - 64 MB de memoria RAM
  - Disco duro de 2 GB
  - Placa de sonido SoundBlaster Live
  - Monitor, Teclado y Mouse estándar
  - Sistema Operativo: Microsoft Windows 98
  
- **Equipo #3**
  - Procesador AMD Duron de 800 MHz
  - 128 MB de memoria RAM
  - Disco duro de 6 GB
  - Placa de sonido compatible on-board (placa base PC-Chips 810)
  - Monitor, Teclado y Mouse estándar
  - Sistema Operativo: Microsoft Windows Me
  
- **Equipo #4**
  - Procesador Intel Pentium III de 750 MHz
  - 256 MB de memoria RAM
  - Disco duro de 20 GB
  - Placa de sonido compatible on-board (placa base Intel 815EEA2)
  - Monitor, Teclado y Mouse estándar
  - Sistema Operativo: Microsoft Windows 2000 Profesional

Los resultados observados se detallan en las tablas y gráficos descriptos en el siguiente apartado.

Los diferentes programas pudieron instalarse exitosamente en los cuatro ambientes, a excepción del primero, en el que algunos de los software testeados reportaron capacidad insuficiente en el sistema.

#### *11.2.4. Tiempos de falla observados en cada ambiente de prueba*

Se detallan a continuación los tiempos observados antes de falla en cada uno de los ambientes de prueba, incluyendo algunas consideraciones particulares en los casos señalados.

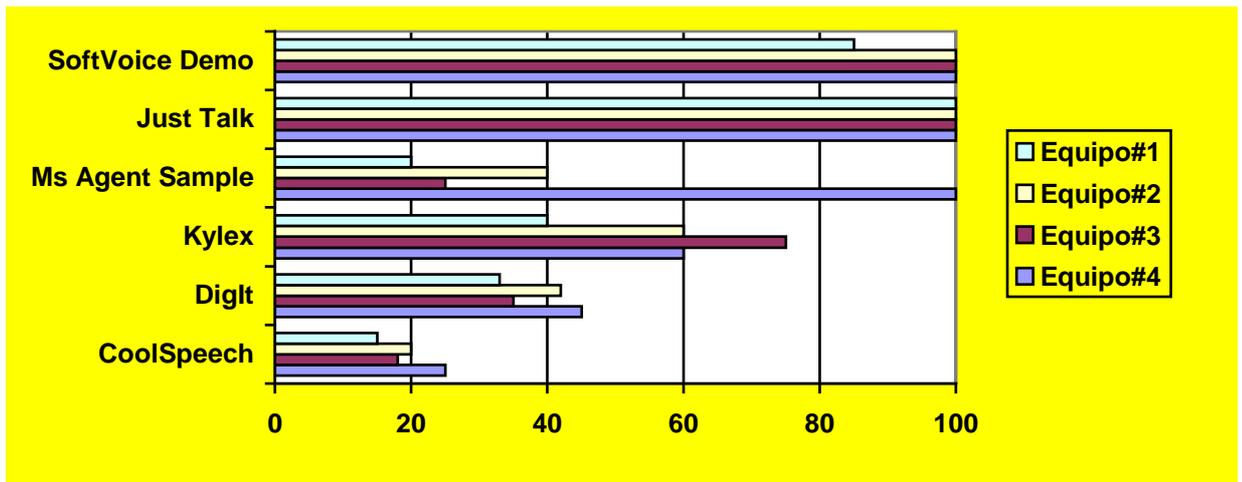


Fig: Tiempo Medio de Falla de cada Software / Ambiente

Los primeros dos productos señalados, de hecho, corrieron sin problemas durante varios minutos, pero a los efectos de la gráfica se consideraron los primeros 100 segundos como indicadores. Cabe señalar que en el caso del Demo provisto por SoftVoice se observó, para el primero de los equipos, un corte en la secuencia de pronunciación después de aplicar de manera masiva cambios en la configuración de la voz. No obstante, dicho corte fue transitorio y no desestabilizó el sistema, que siguió corriendo sin necesidad de detenerlo ni reiniciarlo, a la siguiente petición de pronunciación. En el resto de los equipos no se observó ningún inconveniente.

También cabe señalar que, si bien no está indicado en el gráfico, para el caso del Equipo#3, se observó tanto en el Demo de SoftVoice como en el programa Just Talk (versión lúdica de interfaz toy-style del mismo aplicativo) un pequeño problema de sonido: se producían de manera aislada microcortes en el inicio de la corriente sonora, pero luego pudo constatar que se trataba de algún inconveniente técnico en la tarjeta de audio del sistema. La misma falla se observaba al iniciar la reproducción de cualquier clip de audio, y también de manera aleatoria. Al correr reproducciones continuas enganchadas (iniciando un antes de que termine la anterior) la falla no se observó. Finalmente, probando otra placa de audio (una genérica compatible) en el mismo sistema, el síntoma

desapareció, por lo que en síntesis la falla puede ser atribuida a algún mal funcionamiento de hardware.

Para el caso de los últimos cuatro programas, en todos sin excepción se observaron comportamientos irregulares y problemas de estabilidad, siendo el más característico de los problemas detectados la interrupción abrupta de la corriente de audio generada. Cuando se presentaba este inconveniente, el mismo era revertido, en algunos casos, al iniciar una nueva corriente. En otros casos debió cerrarse el programa y reiniciarlo. Finalmente, se observaron también situaciones que requirieron el reinicio del sistema completo.

Cabe señalar asimismo que el sistema produjo Hangs-Up repentinos, particularmente utilizando el software denominado Xylek's Text Reader, presentándose incluso pantallas azules, volcados de pila, y en la mayoría de las pruebas, falta de respuesta permanente en el sistema (hasta su reinicio).

Sólo el programa MS Agents Sample corrió exitosamente en la mejor de las configuraciones de hardware por más de 100 minutos, aunque cabe señalar que ante pruebas de uso intensas, también produjo dos cortes en las corrientes de audio.

#### *11.2.5. Consideraciones sobre las implementaciones tecnológicas*

El motor Lernout & Hauspie® TruVoice Text-To-Speech Engine debe instalarse como componente separado en el sistema y mediante el proceso de instalación y registro ponerse disponible como servicio, para poder utilizar la interfaz SAPI en los programas de aplicación que lo requieran. Si bien se puede detectar la presencia del motor desde el software, no se determinó de qué manera y la documentación consultada no menciona nada específico al respecto.

Por otra parte, el motor SVTTS de SoftVoice Inc. no requiere la instalación de componentes, sino tan sólo la presencia de dos archivos DLL en el sistema. El modo de ponerlos operativos es: Registrándolos en alguna ubicación específica, o simplemente teniendo las DLL's grabadas

en la misma carpeta que el aplicativo (.EXE) Cabe señalar que si bien esta última técnica ahora la necesidad de registrarlas, tiene como inconveniente la duplicidad de las Bibliotecas en caso de instalar varios aplicativos en rutas diferentes en el mismo equipo, por lo que se recomienda la primera.

#### *11.2.6. Consideraciones sobre Disponibilidad*

En cuando a la estabilidad, quedó ya debidamente explicado en el apartado precedente sobre tiempos antes de fallas en el gráfico analizado.

Respecto de la independencia de dispositivos de hardware de audio específicos de cada sistema, ambos motores soportan dicha independencia. En el caso del SVTTS, no acceden directamente a la placa de audio, sino protocolizando el acceso en los controladores instalados en el propio sistema operativo. Por su parte no se encontró información específica respecto del modo en que lo hace el motor Lernout & Hauspie® TruVoice Text-To-Speech Engine. No obstante, sí informa el fabricante como característica dicha independencia.

Finalmente, en relación a la interfaz de programación Win32, ambos motores están sólo disponibles para el sistema operativo de 32 bits de Microsoft. No obstante, cabe enfatizar una importante diferencia:

Los mecanismos de comunicación entre componentes de ambos motores son diferentes. En el caso de Lernout & Hauspie® TruVoice Text-To-Speech Engine, se apoya en el gestor de mensajes entre objetos del sistema operativo Windows (protocolo de envío y recepción de mensajes), por lo que el adecuado funcionamiento queda supeditado a la capacidad del sistema de resolver a tiempo la secuencia de mensajes hacia el motor. Por su parte, el motor SVTTS accede a todas sus funciones de manera directa al estar implementadas como código compilado de enlace dinámico y directo en tiempo de ejecución, en forma de los archivos de Biblioteca DLL's que lo conforman.

### *11.2.7. Consideraciones sobre el Control Aplicativo*

En lo referido al control de estado de los motores, ambos proveen mecanismos de control de estado, aunque cabe señalar que el tiempo de respuesta es más inmediato en el caso del motor SVTTS que en el del Lernout & Hauspie® TruVoice Text-To-Speech Engine.

Ambos motores proveen funciones y configuraciones de las características audibles de la fonación, permitiendo establecer ciertos conjuntos de voces preestablecidas y la modificación, a partir de las mismas, del resto de las características como timbre, tono, velocidad, etc... No obstante, las capacidades de configuración del motor SVTTS son superiores, pues permite establecer cada componente matemático de las síntesis de voz.

En lo que respecta a la capacidad de ambos sistemas de leer letras, palabras, oraciones y párrafos, ambos los satisfacen.

En cuanto a la lectura de números, especialmente aquellos con parte entera y parte decimal, el motor Lernout & Hauspie® TruVoice Text-To-Speech Engine se comportó de mejor manera que el SVTTS. En éste último se presentan irregularidades en la lectura de números enteros de cuatro cifras, siendo además esta la máxima cantidad de cifras que es capaz de leer naturalmente. Además, cabe considerar que este último motor no lee de manera numérica expresiones con decimales. Por ejemplo, puesto a leer el número [123,45] la fonación indica “uno dos tres coma cuatro cinco” y no “ciento veintitrés con cuarenta y cinco”. Asimismo, si el separador decimal utilizado es el punto la lectura es más acertada, pero aún incompatible con el modo castellano de leer números con decimales. Así, si la expresión a leer es el número [123.45] la fonación resultante indica “ciento veintitrés y cuatro cinco decimal”. Finalmente, si se pone un espacio luego de la coma [123, 45] la lectura indica “ciento veintitrés, cuarenta y cinco”. Estas características indican la falta de sensibilidad del sistema a la configuración regional establecida en el equipo.

Los signos especiales son en general reconocidos y correctamente pronunciados por ambos motores, aunque con diferencias de criterios.

### **11.3. Síntesis Comparativa**

Se resumen en la siguiente tabla los aspectos considerados en la comparativa de los dos motores.

<b>Característica</b>	<b>Lernout &amp; Hauspie® TruVoice TTS Engine</b>	<b>SoftVoice Text-To-Speech Engine (SVTTS)</b>
Dependencia de Hardware específico	No depende de un Hardware específico, sino cualquier dispositivo de salida de sonido compatible.	No depende de un Hardware específico, sino cualquier dispositivo de salida de sonido compatible.
Ambiente de Ejecución	Sistema Operativo Windows de 32 bits	Sistema Operativo Windows de 16 o 32 bits
Soporte Multilingual	Sí, como módulos instalables	Sí. Es una DLL específica para cada idioma soportado
Soporte de Idioma Español	Sí. La voz natural es claramente inteligible, aunque presenta diferencias de fonación.	Sí. La voz natural es claramente inteligible, aunque presenta diferencias de fonación.
Sistema de Fonación	Síntesis Matemática (sin sonidos pregrabados)	Síntesis Matemática (sin sonidos pregrabados)
Diccionario de Excepciones	Interno. No posee interfaces de configuración / personalización por parte del usuario.	Interno y Externo. Incluye una amplia lista de excepciones y posibilita el registro de nuevas excepciones, permitiendo especificar los fonemas para la pronunciación específica de una palabra que no se ajuste a las

		reglas normales.
Implementación e Interfaz de Programación	Objetos Modelo COM, con interfaz de mensajes entre objetos. SAPI.	Funciones compiladas disponibles en DLL's con interfaces de programación definidas para los tipos de datos de C/C++
Estabilidad del Servicio	Inestable al someter el sistema a pruebas intensivas, con cascadas de mensajes a frecuencias mayores que la capacidad de respuesta del sistema.	Estable en todos los casos, incluso sometiendo el sistema a cascadas de llamadas a funciones y requerimientos al motor.
Control de Estado del Motor	Dispone métodos específicos para monitorear el estado.	Dispone funciones específicas para monitorear el estado.
Configuración de Características de Fonación.	Posibilita el control de las características de fonación de alto nivel, permitiendo el ajuste de sólo ciertas variables.	Posibilita el control de las características de fonación de alto, medio y bajo nivel, permitiendo el control de cada variable participante en la síntesis matemática de la fonación.
Lectura de números con decimales y otros signos en castellano	Las funciones de lectura interpretada son bastante completas, exceptuando algunos signos especiales.	Las funciones de lectura interpretada de números no es aceptable, en tanto que para los signos sólo unos pocos no considerados.
Nivel de información y documentación disponible	Abundante. No obstante, es muy poco lo disponible en materia de arquitectura y en historial de fallas conocidas.	Bueno. No obstante, la asistencia personalizada disponible compensa la escasa información en línea.

SDK (Software Development Kit)	Disponible. Orientado de manera no exclusiva a la fonación, sino a la programación de agente.	Disponible y bien ejemplificado, pero sólo para lenguaje C/C++
Instalación y disponibilidad	Requiere la instalación voluntaria y deliberada de los componentes necesarios en varios pasos hasta tener disponibles los servicios, incluyendo el reinicio del sistema.	No requiere la instalación particular de ningún componente, sino el libre acceso a las DLL's que componen el motor y la base idiomática.

## Motor de SoftVoice Inc.

---

### ***11.4. SoftVoice Text-To-Speech : SVTTS***

El sistema SoftVoice Text-to-Speech<sup>31</sup> es un motor de conversión de texto ASCII a discurso verbal en lenguaje natural, de quinta generación. Soporta tanto la pronunciación según las reglas fonéticas del idioma Inglés y del Español, según como se configure. El sistema convierte texto irrestricto en cualquiera de ambos idiomas en un discurso altamente inteligible de sonido natural en tiempo real.

#### ***11.4.1. Sobre la compañía fabricante: SoftVoice Inc.***

SoftVoice Inc. tiene más de 25 años en síntesis de voz. El primer sintetizador de voz completamente basado en software para microcomputadores comercialmente disponible fue escrito por la gente de SoftVoice en 1979. S.A.M. (Software Automatic Mouth) fue líder de ventas en plataformas Apple, Atari y Commodore.

Al principio de los 80's, SoftVoice desarrolló sistemas de TTS (Text-To-Speech / Texto a Voz) para las corporaciones Apple y Amiga, para sus respectivas nuevas máquinas a desarrollar. De hecho, en su lanzamiento

---

<sup>31</sup> La información considerada en este apartado ha sido, en parte, transcrita de la web de SVTTS.

en 1984, la Macintosh anunció su propia existencia al público utilizando este software, la versión original del “Macin Talk”. Por su parte, la Commodore Amiga tuvo uno incipiente sistema de TTS conocido como “narrador.device” incluido en su sistema operativo.

En la actualidad, hay más de ocho millones de copias del sistema SoftVoice TTS en uso a lo largo del mundo, haciendo de SoftVoice uno de los mayores proveedores de sistemas de texto a palabra oral en el mundo.

#### *11.4.2. Sobre el sistema SoftVoice Text-To-Speech*

Implementado como simples DLL's de Windows, el SoftVoice TTS es un sistema experto para la conversión de texto irrestricto en Inglés y Español a discurso de alta calidad en tiempo real. Cuenta con un conjunto de 20 voces predefinidas que pueden seleccionarse según las preferencias del usuario. Estas voces pueden modificarse en sus diferentes parámetros, como timbre y velocidad, de manera prácticamente ilimitada, permitiendo la creación de voces completamente nuevas. Puede también elegirse la calidad vocal entre tres modos preestablecidos: normal, agitado y susurro. Más de 30 comandos de control de fonación pueden ser incluidos en el mismo texto para modificar su salida audible, e incluso puede controlarse el tono, dando al sistema la posibilidad de “cantar”.

Durante el discurso, el programador puede recibir una serie de mensajes (en realidad implementados como lectura en las variables de estado internas del motor) que le permiten sincronizar diferentes eventos a los estados del sintetizados, tales como indicadores de inicio de palabra, inicio de sílaba u otros puntos definidos por el programador.

El sistema de SoftVoice utiliza reglas de conversión de Letra a Sonido, un preprocesador numérico, y un diccionario para determinar la correcta pronunciación de cada palabra. Esta última característica, como se mencionó anteriormente, es particularmente necesaria en el idioma inglés, guardando su aplicación en la lectura de letras individuales por su

nombre, los signos y la lectura de secuencias numéricas según las reglas de composición numéricas específicas del lenguaje.

El sistema SoftVoice está construido en base a la síntesis de formantes, según se detalló en el apartado correspondiente a sistemas de síntesis de voz, en la que se modela matemáticamente el mecanismo de producción del lenguaje humano, y en particular, las resonancias acústicas (formantes) del tracto vocal. A diferencia de los sistemas de base pregrabada (semisílabas, difonos, etc...) en los que piezas pequeñas de discurso de voz real se combinan, la síntesis de formantes no está restringida a una sola voz, ni está limitada a un rango de tono limitado. En sistemas como éste, el usuario puede personalizar de la manera más amplia posible las características vocales. Además, no se producen en este tipo de sistemas los efectos desagradables de gorjeo, falsas consonantes y microinterrupciones que reducen la inteligibilidad e incrementan la fatiga del oyente.

Sigue a continuación una lista de las características más destacables de este motor:

Conversión en tiempo real de texto irrestricto a discurso de alta calidad.

Lenguaje Inglés y Castellano disponibles.

Veinte personalidades vocales predefinidas, incluyendo efectos masculinos, femeninos, niños, robots y alienígenas.

Número ilimitado de personalidades y calidades vocales a ser definido y configurado por el usuario.

Amplio conjunto de comandos para el motor embebibles en el texto (simplificando la llamada a funciones de configuración en el medio de una corriente de texto a pronunciar)

Ratio de pronunciación configurable (de 20 a 800 palabras por minuto)

Control de tono en un rango muy amplio (de 10 a 2000 Hz)

Efectos de modulación de tono (vibrato, perturbación, oscilación verbal, etc...)

Control de calidad vocal (normal, agitado y susurro)

Control del modo de pronunciación (continuo, palabra por palabra y deletreado)

Datos de estado extraíbles en tiempo real, posibilitando la coordinación de eventos de la aplicación (como movimientos de boca en una animación, o seguimiento de las sílabas de discurso)

#### *11.4.3. Implementación, instalación y disponibilidad*

Como se mencionó anteriormente, todas las funciones del motor están implementadas en simples DLL's de Windows, por lo que no existe un procedimiento de instalación y registro particular, excepto el común a cualquier conjunto de funciones agrupadas en una Biblioteca de Enlace Dinámico.

Los archivos que componen el motor son dos DLL's:

SVCTL32.DLL : El módulo de control

SVSPAN32.DLL : El módulo de lenguaje español (\*)

(\*) En caso de necesitar disponer de capacidades de pronunciación en inglés, el módulo que corresponde es el SVENG32.DLL

Adicionalmente, el Kit de Desarrollo provee los archivos de cabecera (.h) y la librería de compilación para lenguaje C/C++, junto con una aplicación de ejemplo desarrollada bajo el modelo de las MFC (Microsoft Foundation Classes) y el manual de referencia de funciones.

No se incluye información ni referencias respecto de las posibilidades de programación de las funciones del motor desde otros lenguajes de programación. De hecho, consultados los fabricantes, indicaron que no creían viable la comunicación a nivel de protocolo con otros lenguajes de programación, en función de las convenciones de parámetros particulares del C y las características de punteros de memoria de referencia absoluta que el motor utiliza.

#### 11.4.4. Funciones Disponibles

Las funciones implementadas en las DLL's de control y las específicas del lenguaje se agrupan, a continuación, por categorías funcionales.

##### a. Funciones de Acción del Motor

- ✓ **SVOpenSpeech** Abre el motor de palabra y establece la voz por defecto
- ✓ **SVCloseSpeech** Cierra el motor de palabra
- ✓ **SVRegister** Registra el software de aplicación
- ✓ **SVAbort** Aborta el discurso actualmente en progreso
- ✓ **SVPause** Pausa el discurso actualmente en progreso
- ✓ **SVResume** Continúa con el discurso actualmente en progreso
- ✓ **SVGetErrorText** Convierte el código de error del motor en una cadena ASCII
- ✓ **SVGetVersionInfo** Provee información de versión útil
- ✓ **SVGetVolume** Devuelve el volumen de salida de voz establecido
- ✓ **SVSetVolume** Establece el volumen de salida de voz
- ✓ **SVCommand** Ejecuta un comando extendido

##### b. Funciones de establecimiento de lenguaje (Inglés / Español)

- ✓ **SVSetLanguage** Establece el lenguaje
- ✓ **SVGetAvailableLanguages** Devuelve una lista de los módulos de lenguaje disponibles en el sistema

##### c. Funciones de diccionario de excepciones

- ✓ **SVLoadUserDictionary** Carga un diccionario de excepciones definido por el usuario
- ✓ **SVUnloadUserDictionary** Descarga un diccionario de excepciones definido por el usuario

##### d. Funciones de generación de síntesis fonética (fonación propiamente dicha)

- ✓ **SVTextToPhon** Traduce el texto en el idioma cargado en códigos fonéticos
- ✓ **SVNarrate** Convierte una entrada fonética en discurso de voz de salida (sintetiza y pronuncia la voz)

- ✓ **SVTTS** Combina la funcionalidad de **SVTextToPhon** y **VNarrate** en una única llamada a función

#### e. Funciones de configuración vocal

- ✓ **SVSetPersonality** Establece una personalidad vocal a partir del conjunto de voces predefinidas
- ✓ **SVSetUserVoice** Establece la configuración de voz a una previamente definida por el usuario
- ✓ **SVGetVoiceInfo** Recupera los parámetros de voz
- ✓ **SVSetVoiceInfo** Establece todos los parámetros de la voz actualmente activa
- ✓ **SVSetWAVFile** Establece un archivo .WAV para la salida

#### f. Funciones de configuración vocal

Este conjunto de llamadas a funciones establecen parámetros específicos para la voz activa. En tanto puede usarse **SVSetVoiceInfo**, a veces es más conveniente llamar una llamada a la API simple para modificar un parámetro específico.

- ✓ **SVSetPitch** Establece la línea base para el tono de la voz
- ✓ **SVSetRate** Establece la ratio de palabras por minuto
- ✓ **SVSetVoicingMode** Establece el estilo vocal (normal, agitado, susurro, etc)
- ✓ **SVSetF0Perturb** Establece la perturbación de tono
- ✓ **SVSetF0Range** Establece el rango de variación natural del tono
- ✓ **SVSetF0Style** Establece la curva de tono
- ✓ **SVSetSpeakingMode** Establece el modo de lectura (natural, palabra por palabra, deletreado, etc)

## Análisis de Características

---

### 11.5. Aspectos funcionales y decisiones de diseño

Se reseñan a continuación, a manera de resumen, los principales elementos considerados a partir de la investigación hasta aquí documentada en relación al motor de Text-To-Speech seleccionado (SVTTS), señalando asimismo los aspectos de diseño y desarrollo de los componentes de software necesarios, consecuencia de los anteriores.

Dichos componentes consisten en los siguientes elementos:

- ✓ API Independiente para la implementación de las funciones de acceso y gestión del motor de palabra SVTTS.
- ✓ Aplicación en la que se ejemplifique la utilización de la API construida y que posibilite la configuración y prueba del motor de Text-To-Speech seleccionado.

Si bien los diferentes aspectos propuestos y considerados en materia de diseño son detallados en profundidad en el siguiente apartado del presente trabajo, este cuadro tiene por objetivo referirlos de manera sintética para su evaluación de conjunto, en contraposición a los aspectos diagnosticados.

<b>Aspecto considerado</b>	
Protocolo de acceso a las funciones compiladas en las Bibliotecas que ofician de interfaz entre los programas de aplicación y el motor.	
<b>Característica observada</b>	<b>Aspecto de Diseño de la Solución</b>
El diseño, implementación y documentación del motor considera solamente la implementación de los protocolos en lenguaje C/C++	Investigando las equivalencias sintácticas y funcionales entre los diferentes tipos de datos y el tratamiento de direcciones absolutas de memoria (punteros FAR) entre Visual Basic y el modelo de C/C++, se protocolizó el acceso a las principales funciones del motor y la disponibilidad de los datos provistos y devueltos por las mismas.

**Aspecto considerado**

Lectura de símbolos y caracteres especiales por parte del motor.

<b>Característica observada</b>	<b>Aspecto de Diseño de la Solución</b>
<p>El motor SVTTS presenta inconvenientes de pronunciación incorrecta (o al menos no coherente con los estándares lingüísticos del castellano que hablamos en Argentina) de algunos caracteres específicos, a la vez que se observa la ausencia de expresión de lectura de los signos de puntuación jerárquico/matemáticos, como son los paréntesis ( ), corchetes [ ] y llaves { }.</p>	<p>Diseñar e implementar, en las funciones de lectura a desarrollar en la API, un analizador sintáctico contextual, que se encargue de reconocer y reemplazar en línea los caracteres que naturalmente el motor pronunciaría de manera incorrecta, u omitiría en su pronunciación. En dicha función de reemplazo, se hará llegar al motor la lectura deseada para la cadena, y no la cadena original con la que habría fallado.</p>

**Aspecto considerado**

Incorrecta lectura de expresiones numéricas.

<b>Característica observada</b>	<b>Aspecto de Diseño de la Solución</b>
<p>El motor sólo dispone de capacidad para la lectura de expresiones enteras de hasta cuatro dígitos. En éstas, además, se observan errores e irregularidades.</p> <p>Adicionalmente, no interpreta adecuadamente las expresiones numéricas con parte decimal, por lo que su lectura es incorrecta.</p>	<p>Incluir en el analizador sintáctico contextual a desarrollar la capacidad de reconocimiento y sustitución de expresiones numéricas.</p> <p>Se establece como alcance en este sentido la correcta pronunciación de números de hasta seis dígitos enteros y con hasta cuatro cifras decimales.</p>

<b>Aspecto considerado</b>	
Documentación de desarrollo provista en el SDK. Nombres de funciones y constantes.	
<b>Diagnóstico</b>	<b>Propuesta de Diseño</b>
La documentación ofrecida por SoftVoice en su SDK utiliza las constantes simbólicas definidas en sus bibliotecas de cabecera para el tratamiento conceptual de los parámetros en muchas de sus funciones. Dichas constantes resultan mnemotécnicas para el desarrollador, a la vez que su reemplazo podría conducir a incompatibilidades entre la implementación a desarrollar y otras fuentes de documentación existentes.	En función de lo señalado como diagnóstico, se decide mantener los nombres de constantes simbólicas referidos en la API independiente a desarrollar, como así también el nombre de cada una de las funciones provistas que se protocolizará de manera adecuada desde Visual Basic. De esta manera, el usuario que acceda a la documentación natural del motor no deberá ocuparse de traducciones ni equivalencias intermedias, sino que podrá acceder de manera directa a las funciones según sus especificaciones naturales, como una de las formas de acceso provistas.

<b>Aspecto considerado</b>	
Instalación y disponibilidad del motor.	
<b>Diagnóstico</b>	<b>Propuesta de Diseño</b>
El sistema se implementa como un conjunto de dos DLL's: La Biblioteca de Control y la propia del idioma castellano.	Al tratarse de Bibliotecas de enlace dinámico, el protocolo de acceso a las funciones de las mismas no requiere de su instalación ni registro al contarlas en una ubicación (ruta) alcanzable por la aplicación. Se implementará su disponibilidad en el mismo subdirectorio en que se encuentre el aplicativo, de modo que la ruta esté siempre disponible al programa.

<b>Aspecto considerado</b>	
Complejidad en las funciones involucradas en el proceso de puesta en marcha y fonación (lectura) del motor	
<b>Diagnóstico</b>	<b>Propuesta de Diseño</b>
<p>En relación a las funciones anteriores, reseñadas en el punto de diagnóstico anterior, se observa que si bien su complejidad no es demasiado grande, es ciertamente mucho mayor a lo estrictamente necesario. De hecho hay conjuntos de funciones que si bien son necesarias para el funcionamiento del motor, no resulta importante que el programador se encargue de controlarlas.</p>	<p>Se implementará como característica de diseño una simplificación considerable en las funciones necesarias para acceder a las características y funciones principales del motor. Esto posibilitará el desarrollo de código en el que el acceso a las actividades de lectura y fonación esté reducido a tan sólo la invocación de una función definida en la API independiente desarrollada.</p> <p>Esto podrá observarse en el código fuente de la aplicación de ejemplo desarrollada al final del presente trabajo.</p>

## **12. Diseño de la API Independiente**

---

### ***12.1. Implementación para el lenguaje Microsoft Visual Basic***

Habiendo seleccionado el motor SVTTS para el desarrollo del proyecto, es importante señalar que independientemente de las fortalezas que lo hacen adecuado para esta aplicación - las que han sido detalladas precedentemente -, se observan sobre el mismo algunas importantes debilidades, que se señalan a continuación, y sobre la solución de las cuales serán orientadas la estrategia y líneas de diseño de la API Independiente que este trabajo tiene por objetivo.

Los principales inconvenientes a solucionar son los siguientes:

✓ **Interfaz de punteros FAR y acceso directo a memoria: Problemas de lenguaje.**

El protocolo de acceso a las funciones del motor está diseñado para accesos absolutos a la memoria y gestionan punteros a las estructuras de datos, lo que es incompatible con los tipos de datos gestionados por Visual Basic, que es el lenguaje elegido para el desarrollo del aplicativo, en función de sus características particulares.

En este sentido, cabe también señalar que un segundo inconveniente a resolver es que el código requiere la utilización de constantes simbólicas, cuya codificación se ofrece en el SDK para C/C++, por lo que también estas definiciones debían de traducirse para Visual Basic.

✓ **Incorrecta lectura de ciertos signos y símbolos del lenguaje**

El motor tiene una implementación particular respecto de la lectura de ciertos símbolos. Muchos de los signos que pueden escribirse en un teclado convencional son directamente ignorados por el motor, en tanto que otros son leídos de manera incorrecta.

✓ **Incorrecta lectura de expresiones numéricas**

Como se señaló en la comparativa entre los motores preseleccionados, una importante debilidad del SVTTS es la incorrecta lectura de números de más de tres cifras. Además, trata incorrectamente los decimales y se limita a números de hasta cuatro dígitos.

Considerando estas limitaciones, el enfoque requerido para desarrollar la API Independiente deberá considerar estos aspectos, a la vez que procurar su principal objetivo: *Contar con funciones de alto nivel, de acceso directo desde el código del aplicativo, que simplemente permitan Leer expresiones y Configurar las diferentes características del motor*. El resto, son complejidades intrínsecas de un motor de TTS tan versátil como el escogido, pero que lejos de interesar al desarrollo, sólo constituyen una complicación innecesaria. Precisamente por ello, la API Independiente a desarrollar se documentará como el simple protocolo necesario para el acceso a las pocas funciones elementales señaladas, resolviendo internamente en esta Biblioteca todos los aspectos de protocolo real con el motor, y la conversión y correcta lectura de signos y números.

Se señalan a continuación, para cada uno de los aspectos señalados como debilidades, los detalles particulares al respecto y las líneas de diseño seguidas para afrontar cada uno de estos inconvenientes.

## **12.2. Interfaz de punteros FAR y acceso directo a memoria**

Como ya se mencionó, la implementación del motor se basa en dos DLL's a cuyas funciones tienen acceso a los programadores. Estas llamadas a funciones constituyen la interfaz de funcionamiento (comandos + datos) entre los programas de aplicación y el motor SVTTS.

El problema es que en toda la documentación provista en el SDK<sup>32</sup> de SoftVoice, e incluso preguntado directamente su representante, Joseph Katz, señaló que el

---

<sup>32</sup> **SDK** (*Software Development Kit*) – Kit de Desarrollo de Software

protocolo de acceso a todas las funciones ofrecidas por el motor estaba diseñado específicamente para su utilización desde código escrito en lenguaje C/C++ o, directamente, en Assembler. Indicó además, que de hecho, el propio motor estaba desarrollado utilizando estos lenguajes, y que la única excepción de la que tenía conocimiento era la protocolización de las llamadas desde Java, que por sus características particulares, comparte en gran medida las capacidades de acceso a memoria y los tipos de datos que ofrece C++.

Esto se debe a que el motor utiliza direcciones absolutas de memoria para referir las estructuras de datos que procesa y devolver a los programas la información de estado y demás elementos de intercambio de datos.

El principal inconveniente que se presenta es que Visual Basic, por sus mismas características de RAD<sup>33</sup> de alto nivel, no soporta los tipos de datos requeridos por estas funciones del motor, implementadas en sus respectivas DLL's.

### *12.2.1. Alternativas tecnológicas para solucionar el problema*

Ante esta situación a resolver, los caminos evaluados fueron dos:

Desarrollar una biblioteca de acceso intermedia.

Resolver las equivalencias de protocolo y acceder directamente desde Visual Basic a las funciones.

La primera de las alternativas consistía en desarrollar e implementar una biblioteca propia, desarrollada también en lenguaje C/C++ que se encargara de recibir los parámetros y función a invocar desde el aplicativo en Visual Basic y por su parte hiciera las conversiones de protocolo necesarias y comandara su ejecución en las propias Bibliotecas DLL del motor. De esta manera, esta nueva biblioteca officaría de “capa

---

<sup>33</sup> **RAD** (*Rapid Application Development*) – Sistemas de desarrollo de aplicaciones rápidos. Son lenguajes cuya interfaz de diseño y programación de aplicaciones simplifican al programador muchas de las tareas que normalmente insumen mucho tiempo en el proceso de creación de un aplicativo. Se caracterizan por disponer de funciones de alto nivel y resolver internamente, sin mucha intervención del programador, los diferentes aspectos de implementación y las rutinas de bajo nivel.

intermedia” entre el aplicativo a desarrollar en Visual Basic y el motor propiamente dicho.

Como esta biblioteca habría de ser desarrollada en el mismo lenguaje que el motor, mantendría compatibilidad a nivel de tipos de datos con el mismo, pero no con el aplicativo. Por cuanto, para posibilitar dicho pasaje de parámetros o valores, una posible solución consistiría en implementar un archivo transitorio que oficiara de tubería entre estos dos procesos, lo que implicaría un sacrificio de performance considerable.

Este modelo no resistía sus argumentos en contra, por lo que se decidió avanzar en dirección a la otra opción.

Esta segunda alternativa consistía en investigar los protocolos de llamadas a funciones compiladas y el tratamiento de los tipos de datos, considerando que las especificaciones estaban definidas en la documentación, pero para lenguaje C/C++.

Se desarrollan a continuaciones los diferentes elementos investigados y tenidos en cuenta para el desarrollo de esta solución.

#### *12.2.2. Llamadas a Funciones Compiladas en una DLL*

El protocolo para llamar a funciones compiladas en una DLL desde Visual Basic, se compone de la siguiente manera:

Declare Function <b><u>N1</u></b> Lib " <b><u>Biblioteca.dll</u></b> " Alias " <b><u>N2</u></b> " ( <b><u>Parámetros</u></b> ) As <b><u>Resultado</u></b>
---

Las partes de la sintaxis en **negrita subrayada** se detallan a continuación:

**N1** Nombre de la función para su referencia e invocación desde el código fuente en VB.

**Biblioteca.dll** Nombre del archivo .DLL que contiene la función compilada. Puede ser un nombre de archivo simple o calificado con su ruta o path.

**N2** Nombre interno de la función (identificador utilizado por el sistema operativo para buscar el entry point de la función en la biblioteca)

**Resultado** Tipo de dato del valor de salida de la función

**Parámetros** Los parámetros que reciben las funciones son separados por comas, y el tipo de dato para cada uno de ellos es particular, dependiendo de la función. El caso es que para pasar los parámetros deseados con el tipo de dato requerido por la biblioteca, resultaba necesario “convertir” a su equivalente, no documentada en Visual Basic, los tipos que se gestionarían en la aplicación.

### *12.2.3. Equivalencias entre los tipos de datos*

Se desarrolló una ardua investigación bibliográfica, procurando diferentes fuentes de referencia, junto a una extensa prueba de laboratorio mediante mecanismos de prueba y error, hasta determinar el modo de pasar por referencia a direcciones absolutas de memoria los datos disponibles y gestionados desde la aplicación en Visual Basic.

El resultado de dicha investigación se resume en la siguiente tabla de equivalencias entre tipos de datos protocolizados en C/C++ y su contraparte en Visual Basic.

<b>Descripción del Tipo de Dato</b>	<b>Sintaxis C/C++</b>	<b>Sintaxis Visual Basic</b>
Puntero FAR a la estructura HSPEECH	HSPEECH FAR * <i>lphSpeech</i>	ByRef lphSpeech As Long
Handle a una ventana	HWND <i>hWnd</i>	ByVal hWnd As Long
Entero sin signo de 16 bits	UINT <i>wDeviceID</i>	ByVal wDeviceID As Integer
Entero sin signo de 32 bits	ULONG <i>ulFlags</i>	ByVal ulFlags As Long
Handle de 32 bits para una instancia del motor	HSPEECH <i>hSpeech</i>	ByVal m_hSpeech As Long
Puntero FAR a una cadena terminada en null	LPSTR <i>lpszCompany</i>	ByVal lpszCompany As String

De esta forma, contando con la información de protocolo de cada función en lenguaje C/C++ provista en la documentación del SDK de SoftVoice para su motor SVTTS, y convirtiendo las especificaciones de tipo de dato para cada parámetro a pasar y recibir de las funciones según el detalle de la tabla anterior, se pudo resolver la incompatibilidad referida.

#### *12.2.4. Nombre interno de la función (punto de entrada en la DLL)*

Otro inconveniente que surgió en relación a esta problemática de acceso y llamada a las funciones compiladas en las DLL's del motor, fue que Visual Basic requiere el nombre interno de la función para poder accederla, como se reseñó anteriormente, en tanto que los protocolos de

llamas suministrados como documentación para su acceso utilizando C/C++ no requieren de ese dato.

Para solucionar este problema y poder averiguar los nombres o puntos de entrada se utilizaron diferentes herramienta de decompilación de código ejecutable y análisis de dependencias entre módulos. Fue especialmente útil una utilidad llamada **Dependency Walker**. Utilizándola puede verse el interior de una DLL, identificando la posición del entry point para cada función que la misma contiene, a qué otros archivos / módulos invoca y el respectivo entry point en aquellos. Con este programa puede inspeccionarse una DLL y obtener el nombre interno de cada función allí compilada. Finalmente con esta información decompilada de las DLL's del motor se pudo construir el protocolo de llamada desde Visual Basic.

#### *12.2.5. Resultado de la Investigación y Aplicación referida*

El resultado fue, finalmente, positivo. Se logró resolver toda la traducción implicada en las llamadas y protocolos de parámetros, por lo que se pudo acceder desde Visual Basic a las funciones contenidas en las DLL's del motor.

Sigue a continuación un fragmento de código donde puede observarse el modo de protocolizar desde Visual Basic algunas de las principales funciones del motor.

```
Public Declare Function SVOpenSpeech _
    Lib "SVCTL32" Alias "_SVOpenSpeech@20" _
    (ByRef lphSpeech As Long, _
    ByVal hWnd As Long, _
    ByVal wDeviceID As Long, _
    ByVal ulFlags As Long, _
    ByVal ulReserved As Long ) As Long
```

```
Public Declare Function SVRegister _  
    Lib "SVCTL32" Alias "_SVRegister@20" _  
    (ByVal m_hSpeech As Long,  
     ByVal lpszCompany As String,  
     ByVal lpszProduct As String, _  
     ByVal ulID As Long,  
     ByVal ulReserved As Long ) As Long
```

```
Public Declare Function SVTTS _  
    Lib "SVCTL32" Alias "_SVTTS@32"  
    (ByVal m_hSpeech As Long, _  
     ByVal lpszInput As String, _  
     ByRef lpLpszOutput As Long, _  
     ByRef lpcbOutput As Long, _  
     ByVal hWnd As Long, _  
     ByVal ulTranslateOptions As Long, _  
     ByVal ulNarrateOptions As Long, _  
     ByVal ulReserved As Long ) As Long
```

```
Public Declare Function SVAbort _  
    Lib "SVCTL32" Alias "_SVAbort@4" _  
    (ByVal lphSpeech As Long ) As Long
```

```
Public Declare Function SVCloseSpeech _  
    Lib "SVCTL32" Alias "_SVCloseSpeech@4" _  
    (ByVal m_hSpeech As Long ) As Long
```

El resto de los protocolos establecidos para cada función, puede verse en el código fuente de la API en la sección siguiente.

### **12.3. Constantes Simbólicas**

En relación a las constantes simbólicas utilizadas en las llamadas a las funciones y del motor y en el análisis y gestión de sus resultados, las mismas venían especificadas por el fabricante en un archivo de cabecera para incluir en los proyectos C/C++ que se desarrollaron. Por lo ya señalado, dichas especificaciones fueron también traducidas, considerando todas las equivalencias de tipo y valor entre ambos lenguajes, en función de lo ya señalado precedentemente.

Esta traducción también incluye las estructuras de datos específicas del motor, que son también utilizadas en las funciones. Aquí nuevamente fueron consideradas todas las equivalencias sintácticas y de tipos, por lo que su implementación también se logró con éxito.

Como ejemplo de las traducciones desarrollados, se ejemplifica a continuación un fragmento del código en lenguaje C/C++ y su equivalente en lenguaje Visual Basic.

#### **Ejemplo Constantes Cabecera API en C/C++**

```
#define sv_EVENT_SPEECH_STARTED      1000 // speaking has begun
#define sv_EVENT_SPEECH_DONE        1001 // speaking has finished
#define sv_EVENT_NEW_SENTENCE       1002 // start of sentence
#define sv_EVENT_ENGLISH_WORD_START 1003 // start of word (in English)
#define sv_EVENT_PHON_WORD_START    1004 // start of word (in phonetics)
#define sv_EVENT_SYLLABLE_START     1005 // start of syllable
#define sv_EVENT_PHONEME_START      1006 // start of phoneme
#define sv_EVENT_USER_SYNC          1007 // user specified event
#define sv_EVENT_MOUTH_SHAPE        1008 // new mouth shape
```

#### **Ejemplo Constantes API en Visual Basic**

```
Public Const sv_EVENT_SPEECH_STARTED = 1000 ' El speech ha comenzado
Public Const sv_EVENT_SPEECH_DONE = 1001 ' El speech ha terminado
Public Const sv_EVENT_NEW_SENTENCE = 1002 ' Inicio de una nueva
                                           ' oración
Public Const sv_EVENT_ENGLISH_WORD_START = 1003 ' Inicio de palabra (modo
                                           ' lenguaje)
Public Const sv_EVENT_PHON_WORD_START = 1004 ' Inicio de palabra (modo
                                           ' fonema)
```

Public Const sv_EVENT_SYLLABLE_START = 1005	' Inicio de sílaba
Public Const sv_EVENT_PHONEME_START = 1006	' Inicio de fonema
Public Const sv_EVENT_USER_SYNC = 1007	' Evento especificado por el usuario
Public Const sv_EVENT_MOUTH_SHAPE = 1008	' Nueva forma boca
Public Const sv_EVENT_BUFFER_READY = 1009	' Buffer PCM disponible
Public Const sv_EVENT_END_OF_SENTENCE = 1010	' Fin de oración

El código fuente de las tablas de constantes simbólicas del sistema también es parte de la API desarrollada, por lo que puede revisarse en su código fuente.

#### **12.4. Incorrecta lectura de signos y símbolos del lenguaje**

Como se señaló anteriormente, el sistema SVTTS con su motor de lenguaje Castellano, presenta ciertas irregularidades u omisiones, en relación a la correcta “lectura” o pronunciación de algunos signos utilizados en castellano.

Los siguientes caracteres son ignorados por el motor. Esto quiere decir que ante su presencia en una cadena a ser “leída”, simplemente no emite ningún sonido especial. Para estos casos, la decisión de diseño fue hacer pronunciar al motor una cadena equivalente al nombre en castellano de cada uno de estos símbolos, según el siguiente detalle

Carácter	Lectura generada
(	Abrir paréntesis
)	Cerrar paréntesis
[	Abrir corchete
]	Cerrar corchete
{	Abrir llave
}	Cerrar llave
“	Comilla
\$	Pesos ( <i>la pronunciación original indica “dollar”</i> )
-	Guión
/	Barra ( <i>la pronunciación original indica “barra oblicua”</i> )

\	Barra Invertida
---	-----------------

Para implementar esto, junto a otras características particulares cuyo desarrollo se indica más adelante, se desarrolló como parte de la API un Analizador Sintáctico de cada cadena a leer, haciendo que el mismo reemplace en la cadena a ser requerida para el motor los signos que naturalmente no pronuncia, por su correspondiente nombre castellano. De este modo, por ejemplo, si la cadena a leer fuera:

“Hola (dijo alarmado)”
------------------------

El analizador sintáctico pasaría como cadena a leer al motor la siguiente:

“Hola, <b>abrir paréntesis</b> , dijo alarmado, <b>cerrar paréntesis</b> ”
--

### ***12.5. Incorrecta lectura de expresiones numéricas***

Finalmente, como se comentó, la lectura numérica del motor no es correcta, particularmente para números de cuatro dígitos; además de que no lee correctamente las expresiones con decimales.

Para solucionar esto, también se incluye, en la API desarrollada, un función que se encarga de generar la cadena de caracteres correspondiente a la lectura en castellano de cualquier número de hasta seis cifras enteras y hasta cuatro decimales. El desarrollo de esta función fue hecho considerando el modo en que se leen las expresiones numéricas en castellano; particularmente, en Argentina; pues suele haber diferencias sutiles con otros países de habla hispana.

Esta función, cuyo código fuente también se adjunta como parte de la API, es llamada internamente por el analizador sintáctico, produciendo el reemplazo en la cadena a leer de las expresiones numéricas por su correspondiente versión en palabras de los números. Para desarrollar esta tarea, el analizador sintáctico identifica las secuencias numéricas en el texto, las aísla, y reemplaza cada una de ellas por el valor devuelto por esta función, llamada **NroaTxt**.

## **12.6. Referencia de procedimientos y funciones implementados en la API para el desarrollo de software desde Visual Basic**

La API desarrollada se organiza en forma de Módulo que podrá ser incluido en cualquier proyecto en Visual Basic para acceder, mediante sus procedimientos y funciones, a las características, servicios y funcionalidades del motor SVTTS.

El nombre dado a este módulo es APISVTTS.bas, pero huelga señalar que puede modificarse sin afectar su funcionamiento.

Las funciones que ofrece se presentan con su protocolo a continuación.

### **Sub InicializarMotor(HandleWindow As Variant)**

Arranca el motor, registrándolo y dejándolo listo para recibir comandos.

Parámetros:

**HandleWindow** Debe indicar el Handle de la ventana que recibirá los mensajes del motor (Propiedad hWnd del formulario)

### **Function Leer(ByVal Texto As String) As String**

Se encarga de que el motor reproduzca verbalmente el texto recibido como parámetro. El procedimiento se encarga de todas las conversiones de formato necesarias, incluyendo el análisis sintáctico del texto, la conversión de números en expresiones castellanas de lectura y la pronunciación de los signos de puntuación no contemplados como ( ) , [ ] , { } , -

Parámetros:

**Texto** Cadena de texto a ser pronunciada.

Valor de Retorno:

**CadenaDeLectura** Cadena de lectura a ser procesada por el motor. Es la cadena formateada, resultado del análisis y modificación sintáctica sobre la cadena de entrada

### **Sub PausarLeer()**

Interrumpe la secuencia de lectura actual, manteniendo en la misma un puntero para su posterior continuación.

Parámetros:

*<No lleva parámetros>*

### **Sub ReanudarLeer()**

Continúa una secuencia de lectura interrumpida previamente.

Parámetros:

*<No lleva parámetros>*

### **Sub AbortarLeer()**

Aborta definitivamente la secuencia de lectura en curso.

Parámetros:

*<No lleva parámetros>*

### **Sub CerrarMotor()**

Cierra el motor, desinstancia todos sus objetos y libera los recursos.

Parámetros:

*<No lleva parámetros>*

### **Function GetVolumen() As Long**

Devuelve el nivel de volumen de voz actual.

Parámetros:

<*No lleva parámetros*>

Valor de Retorno:

*Nivel de Volumen* Entero largo (32 bits). Entre 0% y 100%

### **Sub SetVolumen(ByVal Volumen As Long)**

Establece el nivel de volumen deseado.

Parámetros:

*Volumen* Nivel de volumen a establecer para la voz del motor.  
Entre 0% y 100%

### **Sub SetPersonalidad(ByVal Personalidad As Long)**

Establece el nivel de volumen deseado.

Parámetros:

*Personalidad* Identificador de personalidad a establecer. Existen valores de macro definidos en la API que pueden utilizarse:

0 : sv\_PERS\_MALE  
Voz masculina (valor por defecto)

1 : sv\_PERS\_FEMALE  
Voz femenina

2 : sv\_PERS\_LARGE MALE  
Voz masculina grave

3 : sv\_PERS\_CHILD  
Voz de niño

4 : sv\_PERS\_GIANTMALE  
Voz masculina de bajo registro

5 : sv\_PERS\_MELLOWFEM  
Voz Femenina madura

6 : sv\_PERS\_MELLOWMALE  
Voz Masculina madura

7 : sv\_PERS\_CRISPMALE  
Voz Masculina crispy

8 : sv\_PERS\_THEFLY  
Voz de alto registro

9 : sv\_PERS\_ROBOTOID  
Voz Robotoide

10 : sv\_PERS\_MARTIAN  
Voz de Alien

11 : sv\_PERS\_COLOSSUS  
Voz de Colossus computarizado

12 : sv\_PERS\_FASTFRED  
Voz de rápida pronunciación

13 : sv\_PERS\_OLDWOMAN  
Voz de Mujer mayor

14 : sv\_PERS\_MUNCHKIN  
Voz de Enano

15 : sv\_PERS\_TROLL  
Un poco más grave que enano

16 : sv\_PERS\_NERD  
Voz Masculina NERD

17 : sv\_PERS\_MILKTOAST  
Leche tostada

18 : sv\_PERS\_GIPSY  
Voz Masculina Gitano

19 : sv\_PERS\_CHOIRBOY  
Voz cantora de alto registro

255 : sv\_PERS\_ACTIVEVOICE  
Voz activa actual

### **Sub SetVelocidad(ByVal Velocidad As Long)**

Establece la velocidad o ratio de pronunciación del motor, en palabras por minuto (como medida de aproximación).

Parámetros:

**Velocidad** Indica la velocidad a establecer. Entre 20 y 500 ppm (palabras por minuto).

### **Sub SetTono(ByVal Tono As Integer)**

Establece el Tono vocal (o Pitch) del motor, en Hz de frecuencia base.

Parámetros:

**Tono** Valor de tono actual (entre 10 y 2000 Hz)

### **Sub SetModoVocal(ByVal Modo As Integer)**

Establece el modo vocal de pronunciación.

Parámetros:

**Modo** Identificador de modo vocal a establecer. Existen valores de macro definidos en la API que pueden utilizarse:

0 : sv\_VMODE\_NORMAL

Modo Normal

1 : sv\_VMODE\_BREATHY

Modo Agitado (con más respiración)

2 : sv\_VMODE\_WHISPERED

Modo Susurro

### **Sub SetPerturbacion(ByVal Perturbacion As Integer)**

Establece el nivel de perturbación de frecuencia base (F0) para la voz en curso.

Parámetros:

***Perturbacion*** Jitter a utilizar. Entre 0 y 30, siendo 0 el valor por defecto.

### **Sub SetRango(ByVal Rango As Integer)**

Establece el rango de excursión de la frecuencia base (F0) para la voz en curso.

Parámetros:

***Rango*** Porcentaje de variación. Entre 0% y 500% (el valor por defecto es 100%)

### **Sub SetEstilo(ByVal Estilo As Integer)**

Establece la curva de oscilación de la frecuencia base (F0) a nivel de oración para la voz en curso.

Parámetros:

***Estilo*** Identificador de estilo a establecer. Existen valores de macro definidos en la API que pueden utilizarse:

0 : sv\_F0\_NATURAL

Estilo natural

2 : sv\_F0\_MONOTONE

Estilo monotonal (sin oscilaciones de curva en la oración)

3 : sv\_F0\_SING

Estilo cantado (oscilaciones de alta escala)

4 : sv\_F0\_RANDOM

Estilo aleatorio (notas aleatorias de escla mayor por sílaba)

## 12.7. Código Fuente de la API Construida

Archivo: APISVTTS.bas

```
Option Explicit

'Declaración de las variables Públicas del Programa
'-----

Public hSpeech As Long 'Handler de la sesión abierta con SVOpenSpeech
Public ResultadoSVTTS As Long 'Resultado de cada invocación a una función de la API de SVTTS

'Declaración de variables globales para gestionar
'los parámetros de las llamadas a las funciones de la API de SVTTS
'-----

Public X_hwnd As Long      'Handler de la ventana en que corre la sesión
Public X_wDeviceID As Long 'Identificador de Dispositivo de WAV
Public X_ulFlags As Long   'Banderas para Lenguaje y Calidad de sonido
Public X_ulReserved As Long 'Reservado para futuros usos

'Declaración de las funciones de las DLL del motor SVTTS
'-----

Public Declare Function SVOpenSpeech Lib "SVCTL32" _
    Alias "_SVOpenSpeech@20" _
    (ByRef lphSpeech As Long, _
    ByVal hwnd As Long, _
    ByVal wDeviceID As Long, _
    ByVal ulFlags As Long, _
    ByVal ulReserved As Long) As Long

Public Declare Function SVRegister Lib "SVCTL32" _
    Alias "_SVRegister@20" _
    (ByVal m_hSpeech As Long, _
    ByVal lpszCompany As String, _
    ByVal lpszProduct As String, _
    ByVal ulID As Long, _
    ByVal ulReserved As Long) As Long

Public Declare Function SVTTS Lib "SVCTL32" _
```

```

Alias "_SVTTS@32" _
  (ByVal m_hSpeech As Long, _
   ByVal lpszInput As String, _
   ByRef lpLpszOutput As Long, _
   ByRef lpcbOutput As Long, _
   ByVal hwnd As Long, _
   ByVal ulTranslateOptions As Long, _
   ByVal ulNarrateOptions As Long, _
   ByVal ulReserved As Long) As Long

Public Declare Function SVPause Lib "SVCTL32" _
  Alias "_SVPause@4" _
  (ByVal m_hSpeech As Long) As Long

Public Declare Function SVResume Lib "SVCTL32" _
  Alias "_SVResume@4" _
  (ByVal m_hSpeech As Long) As Long

Public Declare Function SVAbort Lib "SVCTL32" _
  Alias "_SVAbort@4" _
  (ByVal lphSpeech As Long) As Long

Public Declare Function SVCloseSpeech Lib "SVCTL32" _
  Alias "_SVCloseSpeech@4" _
  (ByVal m_hSpeech As Long) As Long

Public Declare Function SVGetVolume Lib "SVCTL32" _
  Alias "_SVGetVolume@8" _
  (ByVal m_hSpeech As Long, _
   ByRef lpdwVolume As Long) As Long

Public Declare Function SVSetVolume Lib "SVCTL32" _
  Alias "_SVSetVolume@8" _
  (ByVal m_hSpeech As Long, _
   ByVal dwVolume As Long) As Long

Public Declare Function SVSetPersonality Lib "SVCTL32" _

```

```

Alias "_SVSetPersonality@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nPersonality As Long) As Long

Public Declare Function SVSetRate Lib "SVCTL32" _
  Alias "_SVSetRate@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nRate As Integer) As Long

Public Declare Function SVSetPitch Lib "SVCTL32" _
  Alias "_SVSetPitch@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nPitch As Integer) As Long

Public Declare Function SVSetVoicingMode Lib "SVCTL32" _
  Alias "_SVSetVoicingMode@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nVoicingMode As Integer) As Long

Public Declare Function SVSetF0Perturb Lib "SVCTL32" _
  Alias "_SVSetF0Perturb@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nF0Perturb As Integer) As Long

Public Declare Function SVSetF0Range Lib "SVCTL32" _
  Alias "_SVSetF0Range@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nF0Range As Integer) As Long

Public Declare Function SVSetF0Style Lib "SVCTL32" _
  Alias "_SVSetF0Style@8" _
  (ByVal m_hSpeech As Long, _
   ByVal nF0Style As Integer) As Long

```

'Declaraciones de Constantes para el SVTTS (Traducción de la biblioteca SVTTS.h original)

```

'-----
Public Const sv_EVENT_SPEECH_STARTED = 1000      ' El speech ha comenzado
Public Const sv_EVENT_SPEECH_DONE = 1001        ' El speech ha terminado
Public Const sv_EVENT_NEW_SENTENCE = 1002       ' Inicio de una nueva oración
Public Const sv_EVENT_ENGLISH_WORD_START = 1003 ' Inicio de palabra (modo lenguaje)
Public Const sv_EVENT_PHON_WORD_START = 1004    ' Inicio de palabra (modo fonema)
Public Const sv_EVENT_SYLLABLE_START = 1005     ' Inicio de sílaba
Public Const sv_EVENT_PHONEME_START = 1006      ' Inicio de fonema
Public Const sv_EVENT_USER_SYNC = 1007         ' Evento especificado por el usuario
Public Const sv_EVENT_MOUTH_SHAPE = 1008       ' Nueva forma boca
Public Const sv_EVENT_BUFFER_READY = 1009       ' Buffer PCM disponible
Public Const sv_EVENT_END_OF_SENTENCE = 1010    ' Fin de oración

```

' Opciones de Lenguaje.

' Estas opciones son especificadas en el campo ulFlag de la

' llamada a la API SVOpensPeech. Múltiples lenguajes pueden especificarse

' en una sola llamada a SVOpenSpeech haciendo "OR" simultáneo entre

' estas banderas.

```

Public Const sv_LANGUAGE_ENGLISH = &H1 ' Lenguaje por defecto

```

```

Public Const sv_LANGUAGE_SPANISH = &H2

```

```

Public Const sv_LANGUAGE_GERMAN = &H4

```

```

Public Const sv_LANGUAGE_FRENCH = &H8

```

```

Public Const sv_LANGUAGE_JAPANESE = &H10

```

```

Public Const sv_LANGUAGE_ITALIAN = &H20

```

```

Public Const sv_LANGUAGE_DUTCH = &H40

```

```

Public Const sv_FLAGS_22KHZ = &H8000000

```

```

Public Const sv_FLAGS_11KHZ = &H4000000 ' Frecuencia de muestreo por defecto

```

```

Public Const sv_FLAGS_8KHZ = &H2000000

```

```

Public Const sv_FLAGS_MULAW = &H1000000 ' Codificación mulaw de 8 bits

```

```

Public Const sv_FLAGS_16BIT = &H800000 ' Tamaño de muestreo por defecto

```

```

Public Const sv_FLAGS_8BIT = &H400000

```

```

' Opciones usadas por las llamadas a SVNarrate y SVTTS.
' Todas estas opciones operan sobre el texto fonético, no en un idioma.

Public Const sv_OPT_WORD_START = &H1           ' evento generar inicio de palabra
Public Const sv_OPT_SYLLABLE_START = &H2       ' evento generar inicio de sílaba
Public Const sv_OPT_PHONEME_START = &H4       ' evento generar inicio de fonema
Public Const sv_OPT_MOUTH_SHAPE = &H8        ' evento generar movimiento de boca
Public Const sv_OPT_WORDBYWORD = &H10        ' pronunciar cada palabra aislada
Public Const sv_OPT_OVERRIDE = &H20         ' aborar requerimientos en progreso
                                           ' e iniciar este requerimiento
Public Const sv_OPT_NOSYNCS = &H40          ' no generar mensajes de eventos
Public Const sv_OPT_IGNORE_EMBCMDS = &H80    ' ignorar comandos incrustados
Public Const sv_OPT_WRITE_WAVFILE = &H100    ' escribir salida en archivo .WAV

```

```

' Opciones usadas por las llamadas SVTextToPhon y SVTTS.
' Estas banderas se usan para dirigir de qué manera SVTextToPhon traduce
' texto idiomático en códigos fonéticos.

```

```

Public Const sv_TMODE_NATURAL = &H1         ' traduce normalmente (default)
Public Const sv_TMODE_WORDBYWORD = &H2     ' ejecutar traducción palabra por palabra
Public Const sv_TMODE_SPELL = &H4         ' deletrear cada palabra
Public Const sv_TMODE_WORD_SYNC = &H8     ' insertar comandos de sincronización de palabra
Public Const sv_TMODE_READ_EMBCMDS = &H10  ' leer comandos incrustados en la salida
Public Const sv_TMODE_IGNORE_EMBCMDS = &H20 ' ignorar comandos incrustados
Public Const sv_TMODE_IGNORE_QM = &H40    ' ignorar signos de interrogación
Public Const sv_TMODE_RETURN_PHONBFR = &H80 ' devolver el buffe fonético (sólo SVTTS)

```

```

' Opciones usadas en la llamada a SVSetWAVFile
' Estas banderas conrolan como es accedido el archivo .WAV

```

```

Public Const sv_WAV_NEWFILE = &H1         ' crear nuevo archivo .WAV
Public Const sv_WAV_OVERWRITE = &H2     ' sobrescribir archivo .WAV
Public Const sv_WAV_APPEND = &H4        ' agregar al archivo .WAV

```

```

' Códigos de error retornables. La llamada a SVGetErrorText (qv) puede ser usada

```

' para devolver una descripción de texto del error.

```
Public Const sv_ERR_BASE = 7000
Public Const sv_ERR_NOERROR = 0 ' sin error
Public Const sv_ERR_PARSE = (sv_ERR_BASE + 1) ' error de interpretación
Public Const sv_ERR_NOMEM = (sv_ERR_BASE + 2) ' no se puede ubicar memoria
Public Const sv_ERR_NOCOEFMEM = (sv_ERR_BASE + 3) ' no se puede ubicar memoria (coef)
Public Const sv_ERR_NOWAVEHDRMEM = (sv_ERR_BASE + 4) ' no se puede ubicar memoria (whdr)
Public Const sv_ERR_NOWAVEMEM = (sv_ERR_BASE + 5) ' no se puede ubicar memoria (wmem)
Public Const sv_ERR_CANTLOCK = (sv_ERR_BASE + 6) ' no se puede bloquear memoria
Public Const sv_ERR_CANTPREPHDR = (sv_ERR_BASE + 7) ' no se puede preparar el header
Public Const sv_ERR_CANTWRITE = (sv_ERR_BASE + 8) ' error de escritura en la salida WAV
Public Const sv_ERR_INVALIDHANDLE = (sv_ERR_BASE + 9) ' handle de speech incorrecto
Public Const sv_ERR_OUTOFRANGE = (sv_ERR_BASE + 10) ' parámetro fuera de rango
Public Const sv_ERR_AUDIODEVS = (sv_ERR_BASE + 11) ' no se detectan dispositivos WAV
Public Const sv_ERR_AUDIOFMT = (sv_ERR_BASE + 12) ' formato de audio incorrecto
Public Const sv_ERR_AUDIOOPEN = (sv_ERR_BASE + 13) ' falla en la apertura de la salida WAV
Public Const sv_ERR_SPEECHBUSY = (sv_ERR_BASE + 14) ' motor de speech ocupado
Public Const sv_ERR_EMBCMD = (sv_ERR_BASE + 15) ' error en comando incrustado
Public Const sv_ERR_CANTABORT = (sv_ERR_BASE + 16) ' no se puede saltar el error
Public Const sv_ERR_NOHWND = (sv_ERR_BASE + 17) ' no se puede crear la ventana
Public Const sv_ERR_UDCANTOPEN = (sv_ERR_BASE + 18) ' no se puede abrir el dicc. de usr.
Public Const sv_ERR_UDEOF = (sv_ERR_BASE + 19) ' fin de archivo en el dicc. de usr.
Public Const sv_ERR_UDFORMAT = (sv_ERR_BASE + 20) ' error de formato en el dicc. de usr.
Public Const sv_ERR_EXPIRED = (sv_ERR_BASE + 21) ' el periodo de evaluación ha expirado
' (sólo versiones de demo)
Public Const sv_ERR_NOLANG = (sv_ERR_BASE + 22) ' lenguaje no soportado
Public Const sv_ERR_NOTIMERS = (sv_ERR_BASE + 23) ' no se puede ubicar servicios de timer
Public Const sv_ERR_INVALIDADDR = (sv_ERR_BASE + 24) ' dirección inválida
Public Const sv_ERR_UNREGISTERED = (sv_ERR_BASE + 25) ' versión no registrada
Public Const sv_ERR_NOTIMPLEMENTED = (sv_ERR_BASE + 26) ' comando no implementado
Public Const sv_ERR_NOUSERVOICE = (sv_ERR_BASE + 27) ' voz de usuario no encontrada
Public Const sv_ERR_WAVFILEOPEN = (sv_ERR_BASE + 28) ' no se puede abrir el archivo WAV
Public Const sv_ERR_WAVFILEWRITE = (sv_ERR_BASE + 29) ' error de escritura en archivo .WAV
Public Const sv_ERR_INTERNAL = (sv_ERR_BASE + 500) ' errores internos
Public Const sv_ERR_MAXERRORLENGTH = 80 ' longitud máxima de mensaje de error
```

```

' Longitudes del tracto vocal.

Public Const sv_VTRACT_MALE = 0           ' masculino
Public Const sv_VTRACT_FEMALE = 1        ' femenino
Public Const sv_VTRACT_CHILD = 2         ' tracto vocal de niño
Public Const sv_VTRACT_BIGMALE = 3       ' tracto vocal masculino voz grave

' Modos de contorno de tono (F0).

Public Const sv_F0_NATURAL = 0           ' natural
Public Const sv_F0_STYLE1 = 0            ' natural (#define obsoleto)
Public Const sv_F0_RESERVED = 1          ' reservado para usos posteriores
Public Const sv_F0_MONOTONE = 2          ' monotonal
Public Const sv_F0_SING = 3              ' cantando
Public Const sv_F0_RANDOM = 4            ' notas aleatorias de escala mayor por sílaba

' Modos vocales.

Public Const sv_VMODE_NORMAL = 0         ' normal
Public Const sv_VMODE_BREATHY = 1        ' agitado
Public Const sv_VMODE_WHISPERED = 2      ' susurro

' Registros de fuente glotal

Public Const sv_GLOT_DEFAULT = 0         ' usa el género para seleccionar la fuente
Public Const sv_GLOT_MALE = 1            ' adulto masculino
Public Const sv_GLOT_FEMALE = 2          ' adulto femenino
Public Const sv_GLOT_CHILD = 3           ' niño pequeño o femenino agudo
Public Const sv_GLOT_HIGH = 4            ' para voces muy agudas, > 300Hz
Public Const sv_GLOT_MELLOW = 5          ' adulto masculino maduro
Public Const sv_GLOT_IMPULSE = 6         ' para voces graves
Public Const sv_GLOT_ODD = 7             ' sólo para armónicas extrañas
Public Const sv_GLOT_COLOSSUS = 8        ' voz de coloso computarizada

```

' Personalidades vocales predefinidas.

```
Public Const sv_PERS_MALE = 0           ' voz masculina por defecto
Public Const sv_PERS_FEMALE = 1        ' voz femenina por defecto
Public Const sv_PERS_LARGE_MALE = 2    ' voz masculina grave por defecto
Public Const sv_PERS_CHILD = 3         ' voz de niño por defecto
Public Const sv_PERS_GIANT_MALE = 4    ' voz masculina de bajo registro
Public Const sv_PERS_MELLOW_FEM = 5   ' femenina madura
Public Const sv_PERS_MELLOW_MALE = 6   ' masculino maduro
Public Const sv_PERS_CRISP_MALE = 7    ' masculino claro
Public Const sv_PERS_THE_FLY = 8       ' voz de alto registro
Public Const sv_PERS_ROBOTOID = 9      ' robotoide
Public Const sv_PERS_MARTIAN = 10      ' alien
Public Const sv_PERS_COLOSSUS = 11     ' la voz del Colossus computarizado
Public Const sv_PERS_FAST_FRED = 12    ' voz de rápida pronunciación
Public Const sv_PERS_OLD_WOMAN = 13    ' mujer mayor
Public Const sv_PERS_MUNCHKIN = 14     ' enano
Public Const sv_PERS_TROLL = 15        ' un poco más grave que enano
Public Const sv_PERS_NERD = 16         ' masculino NERD
Public Const sv_PERS_MILK_TOAST = 17   ' leche tostada
Public Const sv_PERS_TIPSY = 18        ' masculino borracho
Public Const sv_PERS_CHOIR_BOY = 19    ' voz cantora de alto registro
Public Const sv_PERS_ACTIVE_VOICE = 255 ' voz activa actual

Public Const sv_PERS_RICHLOW_MALE = sv_PERS_GIANT_MALE ' compatibilidad hacia atrás
Public Const sv_PERS_SEXY_FEM = sv_PERS_MELLOW_FEM   ' compatibilidad hacia atrás
```

' Constantes varias de SoftVoice

```
Public Const sv_MIN_PITCH = 10         ' tono mínimo admisible
Public Const sv_MAX_PITCH = 2000       ' tono máximo admisible
Public Const sv_MIN_RATE = 20          ' ratio de palabras por minuto mínimo
Public Const sv_MAX_RATE = 500         ' ratio de palabras por minuto máximo
Public Const sv_MAX_VOICENAME_LEN = 32 ' máxima longitud de la cadena de nombre de voz
```

'Procedimientos y Funciones específicos de esta API

Public Sub InicializarMotor(Handlewindow As Variant)

'Abrir una sesión Speech con SVOpenSpeech

'-----

hSpeech = 0 'No importa el valor, porque se pasa por referencia a SVOpenSpeech y al  
          ' volver viene instanciada con el Handler para la sesión de Speech abierta!

X\_hwnd = Handlewindow 'hwnd del formulario llamador

X\_wDeviceID = -1 'Le pide a windows que lo establezca dinámicamente

X\_ulFlags = sv\_LANGUAGE\_SPANISH + sv\_FLAGS\_22KHZ + sv\_FLAGS\_16BIT

X\_ulReserved = 0

ResultadoSVTTS = SVOpenSpeech(hSpeech, X\_hwnd, X\_wDeviceID, X\_ulFlags, X\_ulReserved)

If ResultadoSVTTS <> sv\_ERR\_NOERROR Then

MsgBox "SVOpenSpeech[" & ResultadoSVTTS & "]: No se puede iniciar la sesión con el motor  
Softvoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"

End

End If

'Registrar el motor (Es un requisito! Si no no funciona!)

'-----

ResultadoSVTTS = SVRegister(hSpeech, "XXXXXXXXXX", "XXXXXXXXXX", &HFFFFFFF, 0)

If ResultadoSVTTS <> sv\_ERR\_NOERROR Then

MsgBox "SVRegister[" & ResultadoSVTTS & "]: No se puede registrar la sesión con el motor  
Softvoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"

SVCloseSpeech (hSpeech)

End

End If

End Sub

Public Function Leer(ByVal Texto As String) As String

'Declaración variables propias para pasarlas a la función SVTTS del motor!

Dim X\_lpszInput As String

```

Dim X_lplpszOutput As Long
Dim X_lpcbOutput As Long
Dim X_ulTranslateOptions As Long
Dim X_ulNarrateOptions As Long

'Declaración de variable auxiliares de la rutina
Dim i As Long, j As Long
Dim Caracter As String, Caracter2 As String
Dim TextoNumerico As String
Dim flagUnaComa As Boolean
Dim TextoFormateado As String

'Formatear la cadena a leer, reemplazando los números que haya con la lectura
'correcta de dichos números (el motor los lee mal)
i = 1
Do While i <= Len(Texto)
    Caracter = Mid(Texto, i, 1)
    TextoNumerico = ""
    If Caracter >= "0" And Caracter <= "9" Then
        'Comienza una secuencia numérica
        j = 0
        'TextoNumerico = Carácter
        Caracter2 = Mid(Texto, i + j, 1)
        flagUnaComa = False
        Do While (Caracter2 >= "0" And Caracter2 <= "9") Or (Caracter2 = "," And flagUnaComa
= False)
            If Caracter2 = "," And flagUnaComa = False Then
                flagUnaComa = True
            End If
            TextoNumerico = TextoNumerico & Caracter2
            j = j + 1
            Caracter2 = Mid(Texto, i + j, 1)
        Loop
    End If
    If Right(TextoNumerico, 1) = "," Then
        'La secuencia terminó en una coma
        j = j - 1
    End If

```

```

'Si hubo un texto numérico, concatenar su lectura
'Si no, concatenar el caracter en cuestión
If Len(TextoNumerico) > 0 Then
    TextoNumerico = Replace(TextoNumerico, ",", ".")
    TextoFormateado = TextoFormateado & " " & NroaTxt(Val(TextoNumerico))
    i = i + j
Else
    Select Case Caracter
        Case "("
            TextoFormateado = TextoFormateado & ", abrir paréntesis, "
        Case ")"
            TextoFormateado = TextoFormateado & ", cerrar paréntesis, "
        Case "["
            TextoFormateado = TextoFormateado & ", abrir corchete, "
        Case "]"
            TextoFormateado = TextoFormateado & ", cerrar corchete, "
        Case "{"
            TextoFormateado = TextoFormateado & ", abrir llave, "
        Case "}"
            TextoFormateado = TextoFormateado & ", cerrar llave, "
        Case "\""
            TextoFormateado = TextoFormateado & ", comillas, "
        Case "$"
            TextoFormateado = TextoFormateado & ", pesos, "
        Case "-"
            TextoFormateado = TextoFormateado & ", guión, "
        Case "/"
            TextoFormateado = TextoFormateado & ", barra, "
        Case "\"
            TextoFormateado = TextoFormateado & ", barra invertida, "
        Case Else
            TextoFormateado = TextoFormateado & Carácter
    End Select
    i = i + 1
End If
Loop

'Devolver la cadena a reproducir

```

```

Leer = TextoFormateado

'Ojo! Filtrar que no se trate de una cadena vacía!
If TextoFormateado = "" Then
    Exit Function
End If

'Carga valores en las variables
X_lpszInput = TextoFormateado
X_lplpszOutput = 0
X_lpcbOutput = 0
X_ulTranslateOptions = sv_TMODE_NATURAL 'Modo de pronunciación natural
X_ulNarrateOptions = sv_OPT_OVERRIDE 'Detiene cualquier pronunciación en progreso e inicia
la nueva!

'Leer el texto
ResultadoSVTTS = SVTTS(hSpeech, X_lpszInput, X_lplpszOutput, X_lpcbOutput, X_hwnd,
X_ulTranslateOptions, X_ulNarrateOptions, X_ulReserved)
If ResultadoSVTTS <> sv_ERR_NOERROR Then
    Select Case ResultadoSVTTS
        Case sv_ERR_PARSE
            MsgBox "SVTTS[" & ResultadoSVTTS & "]: Fonema inválido detectado. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        Case sv_ERR_NOMEM
            MsgBox "SVTTS[" & ResultadoSVTTS & "]: Memoria insuficiente para pronunciar el
texto solicitado!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        Case Else
            MsgBox "SVTTS[" & ResultadoSVTTS & "]: Se produjo un error interno con el motor
Softvoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
    End Select
    SVCloseSpeech (hSpeech)
End
End If

End Function

Public Sub PausarLeer()

'Pausar la lectura en curso
ResultadoSVTTS = SVPause(hSpeech)

```

```

If ResultadoSVTTS <> sv_ERR_NOERROR Then
    Select Case ResultadoSVTTS
        Case sv_ERR_INVALIDHANDLE
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        Case Else
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
    End Select
    SVCloseSpeech (hSpeech)
End
End If

End Sub

Public Sub ReanudarLeer()

'Reanudar la lectura en curso
ResultadoSVTTS = SVResume(hSpeech)
If ResultadoSVTTS <> sv_ERR_NOERROR Then
    Select Case ResultadoSVTTS
        Case sv_ERR_INVALIDHANDLE
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        Case Else
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
    End Select
    SVCloseSpeech (hSpeech)
End
End If

End Sub

Public Sub AbortarLeer()

'Abortar la lectura en curso
ResultadoSVTTS = SVAbort(hSpeech)

```

```

If ResultadoSVTTS <> sv_ERR_NOERROR Then
    Select Case ResultadoSVTTS
        Case sv_ERR_INVALIDHANDLE
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        Case Else
            MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
    End Select
    SVCloseSpeech (hSpeech)
End
End If

End Sub

Public Sub CerrarMotor()

    'Cerrar la conexión con el motor y liberar los recursos
    SVCloseSpeech (hSpeech)

End Sub

Public Function GetVolumen() As Long

    'Tomar el volumen actual
    Dim Volumen As Long
    ResultadoSVTTS = SVGetVolume(hSpeech, Volumen)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        GetVolumen = 0
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                SVCloseSpeech (hSpeech)
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
    End If
End Function

```

```

        End
    Else
        GetVolumen = Volumen
    End If

End Function

Public Sub SetVolumen(ByVal Volumen As Long)

    'Establecer el nivel de volumen
    ResultadoSVTTS = SVSetVolume(hSpeech, Volumen)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetPersonalidad(ByVal Personalidad As Long)

    'Establecer la personalidad vocal
    ResultadoSVTTS = SVSetPersonality(hSpeech, Personalidad)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
    End
End Sub

```

```

        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetVelocidad(ByVal velocidad As Long)

    'Establecer la velocidad de pronunciación (ratio de palabras por minuto)
    ResultadoSVTTS = SVSetRate(hSpeech, velocidad)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetTono(ByVal Tono As Integer)

    'Establecer el tono de pronunciación
    ResultadoSVTTS = SVSetPitch(hSpeech, Tono)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
    End If
End Sub

```

```

        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetModoVocal(ByVal Modo As Integer)

    'Establecer el modo vocal
    ResultadoSVTTS = SVSetVoicingMode(hSpeech, Modo)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetPerturbacion(ByVal Perturbacion As Integer)

    'Establecer el nivel de perturbación
    ResultadoSVTTS = SVSetF0Perturb(hSpeech, Perturbacion)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
    End If
End Sub

```

```

        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetRango(ByVal Rango As Integer)

    'Establecer el rango de variación del tono
    ResultadoSVTTS = SVSetF0Range(hSpeech, Rango)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
        SVCloseSpeech (hSpeech)
    End
End If

End Sub

Public Sub SetEstilo(ByVal Estilo As Integer)

    'Establecer el estilo de pronunciación
    ResultadoSVTTS = SVSetF0Style(hSpeech, Estilo)
    If ResultadoSVTTS <> sv_ERR_NOERROR Then
        Select Case ResultadoSVTTS
            Case sv_ERR_INVALIDHANDLE
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Controlador de TTS Invalido. Presione
ENTER!", vbCritical + vbOKOnly, "Error Motor SVTTS"
            Case Else
                MsgBox "SVAbort[" & ResultadoSVTTS & "]: Se produjo un error interno con el
motor SoftVoice TTS!", vbCritical + vbOKOnly, "Error Motor SVTTS"
        End Select
    End If
End Sub

```

```

    End Select
    SVCloseSpeech (hSpeech)
    End
End If

End Sub

' Funciones particulares internas de la API
' -----

Public Function NroaTxt(ByVal Num As Currency) As String
    Dim R As String, Dec As String

    R = ""
    If Num >= 1000000 Then
        R = "## Error: Parámetro fuera de los límites ##"
    Else
        ' Centena de Mil
        If Num >= 900000 Then
            R = R & "novecientos"
            Num = Num - 900000
        ElseIf Num >= 800000 Then
            R = R & "ochocientos"
            Num = Num - 800000
        ElseIf Num >= 700000 Then
            R = R & "setecientos"
            Num = Num - 700000
        ElseIf Num >= 600000 Then
            R = R & "seiscientos"
            Num = Num - 600000
        ElseIf Num >= 500000 Then
            R = R & "quinientos"
            Num = Num - 500000
        ElseIf Num >= 400000 Then
            R = R & "cuatrocientos"
            Num = Num - 400000
        ElseIf Num >= 300000 Then

```

```

    R = R & "trescientos"
    Num = Num - 300000
ElseIf Num >= 200000 Then
    R = R & "doscientos"
    Num = Num - 200000
ElseIf Num >= 101000 Then
    R = R & "ciento"
    Num = Num - 100000
ElseIf Num >= 100000 Then
    R = R & "cien"
    Num = Num - 100000
End If
If LTrim(R) <> "" And Right(R, 1) <> " " Then
    R = R & " "
End If

```

#### ' Decena de Mil

```

If Num >= 90000 Then
    R = R & "noventa"
    Num = Num - 90000
ElseIf Num >= 80000 Then
    R = R & "ochenta"
    Num = Num - 80000
ElseIf Num >= 70000 Then
    R = R & "setenta"
    Num = Num - 70000
ElseIf Num >= 60000 Then
    R = R & "sesenta"
    Num = Num - 60000
ElseIf Num >= 50000 Then
    R = R & "cincuenta"
    Num = Num - 50000
ElseIf Num >= 40000 Then
    R = R & "cuarenta"
    Num = Num - 40000
ElseIf Num >= 30000 Then
    R = R & "treinta"
    Num = Num - 30000

```

```
ElseIf Num >= 29000 Then
    R = R & "veintinueve"
    Num = Num - 29000
ElseIf Num >= 28000 Then
    R = R & "veintiocho"
    Num = Num - 28000
ElseIf Num >= 27000 Then
    R = R & "veintisiete"
    Num = Num - 27000
ElseIf Num >= 26000 Then
    R = R & "veintiseis"
    Num = Num - 26000
ElseIf Num >= 25000 Then
    R = R & "veinticinco"
    Num = Num - 25000
ElseIf Num >= 24000 Then
    R = R & "veinticuatro"
    Num = Num - 24000
ElseIf Num >= 23000 Then
    R = R & "veintitres"
    Num = Num - 23000
ElseIf Num >= 22000 Then
    R = R & "veintidos"
    Num = Num - 22000
ElseIf Num >= 21000 Then
    R = R & "veintiun"
    Num = Num - 21000
ElseIf Num >= 20000 Then
    R = R & "veinte"
    Num = Num - 20000
ElseIf Num >= 19000 Then
    R = R & "diecinueve"
    Num = Num - 19000
ElseIf Num >= 18000 Then
    R = R & "dieciocho"
    Num = Num - 18000
ElseIf Num >= 17000 Then
    R = R & "diecisiete"
```

```

    Num = Num - 17000
ElseIf Num >= 16000 Then
    R = R & "dieciseis"
    Num = Num - 16000
ElseIf Num >= 15000 Then
    R = R & "quince"
    Num = Num - 15000
ElseIf Num >= 14000 Then
    R = R & "catorce"
    Num = Num - 14000
ElseIf Num >= 13000 Then
    R = R & "trece"
    Num = Num - 13000
ElseIf Num >= 12000 Then
    R = R & "doce"
    Num = Num - 12000
ElseIf Num >= 11000 Then
    R = R & "once"
    Num = Num - 11000
ElseIf Num >= 10000 Then
    R = R & "diez"
    Num = Num - 10000
End If
If LTrim(R) <> "" And Right(R, 1) <> " " Then
    R = R & " "
    If Num >= 1000 Then
        R = R & "y "
    End If
End If

'Unidad de Mil
If Num >= 9000 Then
    R = R & "nueve"
    Num = Num - 9000
ElseIf Num >= 8000 Then
    R = R & "ocho"
    Num = Num - 8000
ElseIf Num >= 7000 Then

```

```

R = R & "siete"
Num = Num - 7000
ElseIf Num >= 6000 Then
    R = R & "seis"
    Num = Num - 6000
ElseIf Num >= 5000 Then
    R = R & "cinco"
    Num = Num - 5000
ElseIf Num >= 4000 Then
    R = R & "cuatro"
    Num = Num - 4000
ElseIf Num >= 3000 Then
    R = R & "tres"
    Num = Num - 3000
ElseIf Num >= 2000 Then
    R = R & "dos"
    Num = Num - 2000
ElseIf Num >= 1000 Then
    R = R & "un"
    Num = Num - 1000
End If

'Indicador de mil
If LTrim(R) <> "" Then
    If Right(R, 1) <> " " Then
        R = R & " "
    End If
    R = R & "mil "
End If

'Centena
If Num >= 900 Then
    R = R & "novecientos"
    Num = Num - 900
ElseIf Num >= 800 Then
    R = R & "ochocientos"
    Num = Num - 800
ElseIf Num >= 700 Then

```

```

R = R & "setecientos"
Num = Num - 700
ElseIf Num >= 600 Then
R = R & "seiscientos"
Num = Num - 600
ElseIf Num >= 500 Then
R = R & "quinientos"
Num = Num - 500
ElseIf Num >= 400 Then
R = R & "cuatrocientos"
Num = Num - 400
ElseIf Num >= 300 Then
R = R & "trescientos"
Num = Num - 300
ElseIf Num >= 200 Then
R = R & "doscientos"
Num = Num - 200
ElseIf Num >= 101 Then
R = R & "ciento"
Num = Num - 100
ElseIf Num = 100 Then
R = R & "cien"
Num = Num - 100
End If
If LTrim(R) <> "" And Right(R, 1) <> " " Then
R = R & " "
End If

```

' Decena

```

If Num >= 90 Then
R = R & "noventa"
Num = Num - 90
ElseIf Num >= 80 Then
R = R & "ochenta"
Num = Num - 80
ElseIf Num >= 70 Then
R = R & "setenta"
Num = Num - 70

```

```
ElseIf Num >= 60 Then
    R = R & "sesenta"
    Num = Num - 60
ElseIf Num >= 50 Then
    R = R & "cincuenta"
    Num = Num - 50
ElseIf Num >= 40 Then
    R = R & "cuarenta"
    Num = Num - 40
ElseIf Num >= 30 Then
    R = R & "treinta"
    Num = Num - 30
ElseIf Num >= 29 Then
    R = R & "veintinueve"
    Num = Num - 29
ElseIf Num >= 28 Then
    R = R & "veintiocho"
    Num = Num - 28
ElseIf Num >= 27 Then
    R = R & "veintisiete"
    Num = Num - 27
ElseIf Num >= 26 Then
    R = R & "veintiseis"
    Num = Num - 26
ElseIf Num >= 25 Then
    R = R & "veinticinco"
    Num = Num - 25
ElseIf Num >= 24 Then
    R = R & "veinticuatro"
    Num = Num - 24
ElseIf Num >= 23 Then
    R = R & "veintitres"
    Num = Num - 23
ElseIf Num >= 22 Then
    R = R & "veintidos"
    Num = Num - 22
ElseIf Num >= 21 Then
    R = R & "veintiuno"
```

```

    Num = Num - 21
ElseIf Num >= 20 Then
    R = R & "veinte"
    Num = Num - 20
ElseIf Num >= 19 Then
    R = R & "diecinueve"
    Num = Num - 19
ElseIf Num >= 18 Then
    R = R & "dieciocho"
    Num = Num - 18
ElseIf Num >= 17 Then
    R = R & "diecisiete"
    Num = Num - 17
ElseIf Num >= 16 Then
    R = R & "dieciseis"
    Num = Num - 16
ElseIf Num >= 15 Then
    R = R & "quince"
    Num = Num - 15
ElseIf Num >= 14 Then
    R = R & "catorce"
    Num = Num - 14
ElseIf Num >= 13 Then
    R = R & "trece"
    Num = Num - 13
ElseIf Num >= 12 Then
    R = R & "doce"
    Num = Num - 12
ElseIf Num >= 11 Then
    R = R & "once"
    Num = Num - 11
ElseIf Num >= 10 Then
    R = R & "diez"
    Num = Num - 10
End If
If LTrim(R) <> "" Then
    If Right(R, 1) <> " " Then
        R = R & " "
    
```

```
End If
If Num >= 1 Then
    R = R & "y "
End If
End If
```

'Unidad

```
If Num >= 9 Then
    R = R & "nueve"
    Num = Num - 9
ElseIf Num >= 8 Then
    R = R & "ocho"
    Num = Num - 8
ElseIf Num >= 7 Then
    R = R & "siete"
    Num = Num - 7
ElseIf Num >= 6 Then
    R = R & "seis"
    Num = Num - 6
ElseIf Num >= 5 Then
    R = R & "cinco"
    Num = Num - 5
ElseIf Num >= 4 Then
    R = R & "cuatro"
    Num = Num - 4
ElseIf Num >= 3 Then
    R = R & "tres"
    Num = Num - 3
ElseIf Num >= 2 Then
    R = R & "dos"
    Num = Num - 2
ElseIf Num >= 1 Then
    R = R & "uno"
    Num = Num - 1
End If
```

'Centavos

```
If Trim(R) = "" Then
```

```

    R = R & "cero"
End If
If Right(R, 1) <> " " Then
    R = R & " "
End If
Dec = Trim(Str(Num))
Dec = Replace(Dec, ".", "")
Dec = Left(Dec, 4)
If Val(Dec) >= 1 Then
    R = R & "coma "
    Do While Left(Dec, 1) = "0"
        R = R & "cero "
        Dec = Right(Dec, Len(Dec) - 1)
    Loop
    R = R & NroaTxt(Val(Dec))
End If

End If

NroaTxt = R

End Function

```

## 13. Desarrollo de la Aplicación de Software

---

Para el desarrollo de la Aplicación de Software de la solución diseñada, se consideraron los diferentes aspectos funcionales, pedagógicos y aplicativos según fuentes señaladas en el marco teórico. Adicionalmente, todo el proceso fue llevado a cabo con permanente Feed-Back del usuario, su maestra integradora y su familia, quienes concientes de sus capacidades e intereses, aportaron elementos conceptuales para la definición final de las funcionalidades de la aplicación.

### 13.1. Estructura de Releases entregados

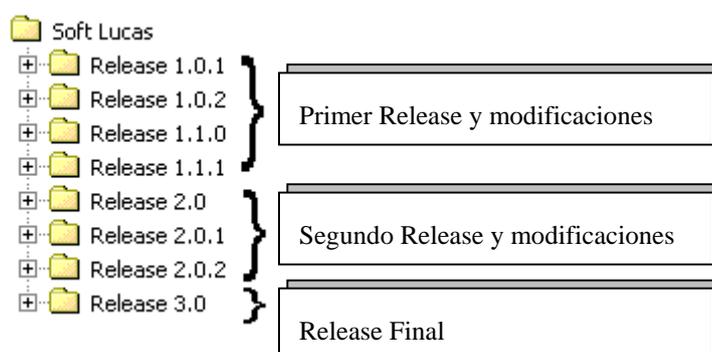
Como se mencionó en la introducción del trabajo, el modelo de ciclo de vida que se siguió para su desarrollo fue el de Prototipado Evolutivo.

Este modelo de ciclo de vida permite la construcción de sucesivas versiones o “releases” del software, a través de las cuales se va avanzando en la satisfacción del conjunto de necesidades del usuario que no pueden establecerse con precisión a priori.

Cada iteración del modelo implica el análisis de la información obtenida a partir de la experiencia de uso del software en la versión anterior, más la consideración de nuevos requerimientos que surgen de su uso, hasta la definición de un conjunto de características que justifica la producción de una nueva release.

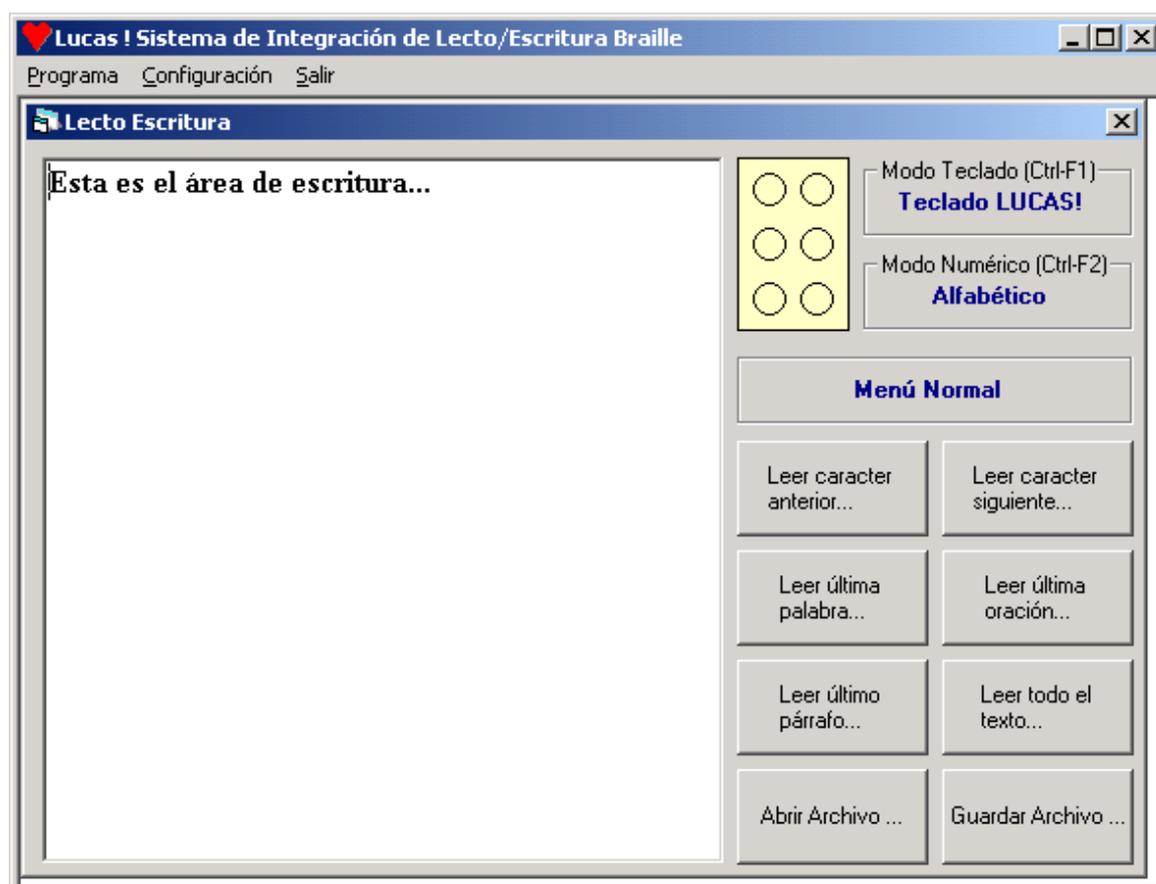
Siguiendo este esquema, el aplicativo desarrollado evolucionó a través de 3 releases o versiones principales, con algunas sub-VERSIONES adicionales orientadas básicamente a la solución de errores menores o inclusión de nuevas características de menor envergadura.

Sigue a continuación la estructura de versiones liberadas y sus características:



### 13.2. Versión 1.0 y subsiguientes

En la primera implementación de la solución, las características que se desarrollaron fueron las básicas del primer modelo aplicativo anticipado, en función de los requerimientos indagados inicialmente y la integración con la solución de Hardware construida. Estas características se resumen a continuación, presentando la interfaz del software:



#### 13.2.1. Requerimientos iniciales

El sistema debía:

- Permitir el ingreso de caracteres Braille, mediante la selección de los puntos que lo componen (identificados como “punto 1”, “punto 2”, ... , “punto 6”).
- Reflejar en pantalla cada carácter ingresado, constituyendo el texto en el área de escritura.

- Proporcionar un Feedback auditivo al usuario que confirme el efecto de cada tecla del tablero pulsada.
- Indicar al usuario, también de manera auditiva, cada carácter escrito correctamente, mediante su nombre.
- Identificar y procesar adecuadamente la codificación de caracteres numéricos del alfabeto Braille.
- Posibilitar el ingreso de caracteres de símbolos especiales contemplados en el alfabeto Braille.
- Permitir la “lectura” (mediante la pronunciación audible para el usuario) de cada palabra que haya terminado de escribir.
- Permitir la lectura del texto completo o fragmentos del mismo, tales como la última oración escrita o el último párrafo.
- Aportar información de contexto, indicando a solicitud del usuario la letra o carácter que se encuentre inmediatamente antes del cursor en el texto, o inmediatamente después, a los efectos de adquirir destreza en la corrección de los errores de escritura que pudiese producir.
- Salvar automáticamente y a intervalos regulares el texto en edición en un archivo, sin intervención del usuario, de manera que no se pierda lo escrito de una sesión de uso a otra.
- Posibilitar guardar el texto en edición en un archivo a elección del usuario.
- Posibilitar la recuperación de texto previamente salvado en un archivo, poniendo el mismo en el área de edición del usuario.
- Permitir la intervención de la maestra integradora - u otra persona que asista al usuario - en el proceso de escritura, pudiendo ubicar el cursor en cualquier punto del texto y posibilitando la modificación del texto propiamente dicho, agregando, quitando o corrigiendo los caracteres que lo componen.
- Permitir al asistente del usuario las operaciones de edición sobre el texto en edición (copiar, cortar y pegar) desde otros orígenes de texto, con el

fin de proponer al usuario actividades o lecturas adicionales a lo escrito por él mismo.

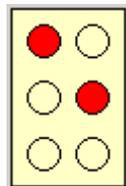
### 13.2.2. Características del Software

Como soporte a los requerimientos establecidos, las características del software se compusieron de la siguiente manera:

#### ➤ **Escritura de puntos braille**

El software reacciona ante la pulsión de los botones del tablero correspondientes a los 6 puntos Braille señalados de dos maneras:

- En primer término, produce un FeedBack auditivo al usuario indicándole la textura que presionó. De este modo, cuando pulsa el punto correspondiente al primer punto Braille, escucha por sus auriculares la pronunciación de “Lija”, con lo que la experiencia del usuario se enriquece en el sentido de la seguridad sobre la acción realizada.
- En segundo término, se ilumina en pantalla el punto correspondiente en la ficha de carácter Braille que se observa en la interfaz, conservando el color rojo sobre el punto seleccionado, de manera que cada nuevo punto ingresado se agrega al patrón hasta constituir todos los puntos que componen el carácter en cuestión.



Esta funcionalidad permite a su terapeuta, o a quién asista a Lucas en esta actividad, ir chequeando que los puntos que va ingresando coincidan con los del carácter que pretende escribir, permitiéndole intervenir ante un error u omisión, fundamentalmente en las primeras etapas de uso del aplicativo y hasta tanto adquiera la seguridad y destreza necesaria para la correcta escritura de los diferentes caracteres que componen el alfabeto Braille.

Si bien este FeedBack implica un sistema de traslación de códigos demasiado complejo (de la textura al punto Braille, y de la sumatoria de estos a la constitución del carácter en cuestión), este fue un requerimiento de su terapeuta.

Los puntos pulsados no pueden “de-seleccionarse”, y repetir un punto ya pulsado no produce diferencias en la codificación. Este aspecto fue intencionalmente considerado en razón que en su proceso de aprendizaje, y hasta que el usuario adquiriese la certeza y seguridad necesaria sobre el patrón de cada punto Braille, era esperable que repitiera el ingreso de algunos de los puntos que constituyen los caracteres.

Para la aplicación se consideraron los caracteres Braille universales como conjunto básico, que fue ampliado como parte de este trabajo para cubrir algunos caracteres especiales no contemplados en el alfabeto original. Sigue a continuación la tabla braille considerada.

Punto1	Punto2	Punto3	Punto4	Punto5	Punto6	Carácter	Pronunciación
0	0	0	0	0	0		Espacio
1	0	0	0	0	0	A	A
1	1	0	0	0	0	B	Be Larga
1	0	0	1	0	0	C	Ce
1	0	0	1	1	0	D	De
1	0	0	0	1	0	E	E
1	1	0	1	0	0	F	Efe
1	1	0	1	1	0	G	Ge
1	1	0	0	1	0	H	Hache
0	1	0	1	0	0	I	I
0	1	0	1	1	0	J	Jota
1	0	1	0	0	0	K	Ka
1	1	1	0	0	0	L	Ele
1	0	1	1	0	0	M	Eme
1	0	1	1	1	0	N	Ene
1	1	0	1	1	1	Ñ	Eñe
1	0	1	0	1	0	O	O
1	1	1	1	0	0	P	Pe
1	1	1	1	1	0	Q	Cu
1	1	1	0	1	0	R	Erre
0	1	1	1	0	0	S	Ese
0	1	1	1	1	0	T	Te

Punto1	Punto2	Punto3	Punto4	Punto5	Punto6	Carácter	Pronunciación
1	0	1	0	0	1	U	U
1	1	0	0	1	1	Ü	U con diéresis
1	1	1	0	0	1	V	Ve Corta
0	1	0	1	1	1	W	Doble ve
1	0	1	1	0	1	X	Equis
1	0	1	1	1	1	Y	Y Griega
1	0	1	0	1	1	Z	Zeta
1	1	1	0	1	1	Á	á acentuada
0	1	1	1	0	1	É	é acentuada
0	0	1	1	0	0	Í	í acentuada
0	0	1	1	0	1	Ó	ó acentuada
0	1	1	1	1	1	Ú	ú acentuada
0	0	1	0	0	0	.	Punto
0	1	0	0	0	0	,	Coma
0	1	1	0	0	0	;	Punto y coma
0	1	0	0	1	0	:	Dos puntos
0	0	1	0	0	1	-	Guión
0	1	1	0	1	0	+	Signo de suma
0	1	0	0	1	1	/	Signo de división
0	1	1	0	0	1	*	Signo de multiplicación
0	1	1	0	1	1	=	Signo de igual
1	1	0	0	0	1	(	Abrir paréntesis
0	0	1	1	1	0	)	Cerrar paréntesis
1	0	0	1	1	1	¿	Abrir Signo de Pregunta
0	1	0	0	0	1	?	Cerrar Signo de Pregunta
1	0	0	1	0	1	¡	Signo de Pregunta
0	0	0	0	1	1	!	Signo de Admiración
0	1	0	1	0	1	@	Arroba
0	0	0	1	1	0	[	Abrir corchete
0	0	0	1	1	1	]	Cerrar corchete
0	0	1	0	1	0	{	Abrir llave
0	0	1	0	1	1	}	Cerrar llave
<b>Caracteres Especiales</b>							
0	0	1	1	1	1		Modo Numérico
0	0	0	1	0	1		Mayúscula
1	0	0	0	0	1	<*>	Marca de Bloque
1	1	1	1	1	1		Nuevo Párrafo

(\*) En las columnas de Puntos Braille, 0 significa punto apagado, y 1 significa punto encendido.

➤ **Acción del Botón Espaciador (Aceptar el carácter ingresado)**

La funcionalidad desarrollada para este botón del tablero es dual, acorde a las características mismas del sistema Braille, en el que el espacio separador de palabras se representa como un espacio sin puntos en la línea. Su comportamiento es el siguiente:

Al pulsar el botón se analizan los puntos Braille ingresados y el carácter correspondiente a la codificación de los mismos es agregado al texto en el punto de inserción – o posición del cursor – sobre el área de escritura.

Si se pulsa sin ningún punto seleccionado previamente, el carácter ingresado es el espacio en blanco, y el usuario recibe FeedBack auditivo de tal carácter.

Finalmente, si es pulsado y los puntos Braille ingresados no se corresponden a ningún carácter del alfabeto, el usuario recibe un mensaje audible que le indica el error, y el conjunto de puntos ingresado se resetea, es decir, queda la ficha braille nuevamente en blanco y el usuario deberá ingresar nuevamente los puntos que corresponden al carácter que deseaba introducir. De este modo, se mantiene la interfaz dual, y quien asiste al usuario puede orientarlo acerca del error producido o el modo correcto de ingresar el carácter deseado.

➤ **Acción del Botón de Cancelación o Retroceso**

Similarmente al caso anterior, este botón cuenta con una doble funcionalidad:

Si es pulsado luego de que se hayan presionado uno o más puntos Braille, el carácter que empezó a escribirse es “cancelado”, con lo que la ficha Braille queda nuevamente en blanco. A su vez, el usuario recibe un Beep como FeedBack audible de dicha cancelación. Esto le permite que en caso de reconocer un error, pueda dar marcha atrás en el progreso de escritura y volver a ingresar el carácter correcto.

Por otra parte, si este botón es pulsado sin haber comenzado a ingresar ningún carácter, actúa de borrado hacia atrás del último carácter ingresado. Algo equivalente a la función del BackSpace en un teclado convencional de computadora. De esta forma, si el usuario descubre su error luego de haber

ingresado un carácter Braille correcto, pero diferente del que en realidad quería ingresar, puede eliminarlo y corregir su error. Cuando es esta la funcionalidad desarrollada, el usuario recibe un mensaje audible que le indica cuál fue el carácter eliminado, lo que corrobora y ratifica la corrección del error producido.

#### ➤ **Edición multilínea**

La edición de texto que realiza el usuario es presentada en pantalla en un cuadro multilínea. De este modo, su integradora puede ver el texto que se encuentra en progreso y monitorear su avance, indicando correcciones o verificando consignas. El espacio completo del texto es manejado independientemente del área visible en pantalla. De hecho, se trata de un único texto plano sin limitaciones específicas de tamaño o longitud; pero el hecho de hacer visible permanentemente la parte de dicho texto en la que se encuentra escribiendo el usuario hace que quien interactúa con él en esta actividad pueda tener la información necesaria de manera visual.

#### ➤ **Modo numérico**

En el alfabeto Braille, los dígitos numéricos se escriben exactamente del mismo modo que las diez primeras letras (de la A a la J). La diferencia radica en la utilización de un carácter especial, llamado “símbolo de número”, cuya semántica consiste simplemente en considerar a los caracteres que siguen como números y no como letras, hasta que vuelva a utilizarse el símbolo de número con lo cual los caracteres siguientes vuelven a ser considerados como letras, y no como números. Es por eso que en este trabajo se designa como “modo numérico” al estado interpretativo de la aplicación luego de recibir como entrada por parte del usuario el carácter correspondiente al símbolo de número. Este estado se mantiene hasta que vuelva a ser ingresado el símbolo de número con lo que se desactiva.

Cuando la aplicación se encuentra en este estado, la pantalla lo indica para que el acompañante pedagógico del usuario pueda estar al tanto de la situación e

intervenir en lo que fuere necesario. Asimismo, el estado de “modo numérico” es controlable desde el teclado normal de la computadora, mediante la combinación de teclas **Ctrl-F2** que fue programado como Switch entre el modo numérico y el modo normal.



También es importante señalar que el usuario recibe el FeedBack correspondiente cada vez que ingresa el código Braille correspondiente al símbolo de número (p3-p4-p5-p6), indicándole explícitamente, de manera audible, el mensaje “Modo Numérico Activado” o “Modo Numérico Desactivado”, según el estado en que quede la aplicación. Además, cuando el modo numérico está activado, sólo son aceptados por la aplicación los caracteres correspondientes a los dígitos numéricos (letras A a J del estado normal), y los caracteres correspondientes a signos de puntuación, signos de operación aritmética, y caracteres especiales; pero son rechazados los correspondientes al resto de las letras. Finalmente, y aunque resulte obvio, al encontrarse la aplicación en estado numérico no sólo se aceptan como números los caracteres correspondientes (y son incorporados como tales al texto en edición), sino que el usuario oye la pronunciación del dígito numérico en lugar del nombre de la letra que hubiera representado si no se hallara en modo numérico.

### ➤ **Modo intervención manual**

De modo similar a lo que ocurre con el switching o cambio manual entre modo numérico y modo alfabético producido a través del teclado convencional por la persona que asiste a al usuario en el uso de la herramienta, puede también alternarse la posibilidad de interactuar sobre el texto mediante el teclado normal. Para hacer esto, la combinación de teclas capturada es **Ctrl-F1**.



Con esta característica no sólo se posibilita la intervención manual externa sobre el texto en edición (con los fines pedagógicos ya señalados), sino que también se logra establecer una frontera de mutua exclusión entre un método de acceso a la aplicación y otro, de manera que ni las acciones de Lucas sobre su tablero interfieran el proceso de corrección que esté desarrollando su terapeuta, ni las acciones o eventos de teclado producidos por otra persona

sobre el teclado convencional de la computadora afecten el proceso de escritura que está llevando adelante el usuario principal de la herramienta. Esta característica es importante, sobre todo considerando que uno de los ámbitos en los que se utiliza la herramienta es el aula de la escuela normal, por lo que otros niños podrían sin querer interferir en el proceso de escritura generando mucha confusión a Lucas.

### ➤ **Lectura y Botones Funcionales**

Las diferentes funciones provistas al usuario en esta primera versión se dispusieron paralelamente en pantalla, para su utilización mediante la botonera visible en la parte derecha de la interfaz. Adicionalmente, el usuario principal de la aplicación accede a las mismas utilizando las dos columnas derechas de su tablero especial. Así, se guarda una relación de aspecto y disposición entre la forma de accionar los botones de función desde el tablero especial como desde la interfaz en pantalla, lo cual mantiene la coherencia e isomorfia entre ambos caminos de acceso.

Las teclas de función fueron etiquetadas “Función 1” a “Función 8” según el siguiente esquema:



Este esquema responde a la distribución de orden utilizada también en la asignación de botones para los puntos Braille, con lo que se simplifica su recordación por analogía con aquel sistema de distribución.

Las funciones consideradas en esta primera implementación guardan relación básicamente con las funciones de lectura de la aplicación, posibilitando las siguientes operaciones:

- **Función 1 y Función 5**: Leen, respectivamente, el carácter que se encuentra inmediatamente antes e inmediatamente después de la posición actual del cursor. Esto le permite al usuario conocer el contexto de escritura para tipear el carácter correcto. Si bien se preveía que podían producirse errores en la escritura (como componente natural del proceso de aprendizaje), la indicación pedagógica para la primera implementación fue que la asistente de Lucas le ayudara a organizarse en el proceso.
- **Función 2**: El software “lee” la última palabra más cercana a la posición del cursor. Es decir: si el cursor se encuentra en medio o al final de una palabra, el usuario la escuchará completa, en tanto que si se encuentra al principio de una palabra, escuchará completa la lectura del carácter inmediato anterior, típicamente, un espacio o signo de puntuación, que son los separadores de palabras definidos en el software. Esta funcionalidad aporta igualmente al sentido de contexto y posición de escritura en términos pedagógicos como se comentó anteriormente.
- **Función 6**: Con esta tecla el software “lee” la última oración escrita, siguiendo el mismo criterio que el enunciado anteriormente para determinar la última palabra. La diferencia radica en que los caracteres considerados separadores de oración son un subconjunto de los considerados separadores de palabra. Básicamente, son los signos de punto y seguido, punto y aparte, signo de exclamación o de interrogación, y salto de párrafo. El sentido de esta función es específico para la tarea de lectura y comprensión por parte del usuario.
- **Función 3**: Nuevamente, esta función posibilita la lectura de un conjunto aún más amplio: todo el párrafo en el que se encuentra el cursor. En este sentido, sólo los punto y aparte indican cambios de párrafo.
- **Función 7**: Finalmente, esta tecla le permite oír la lectura de todo el texto en edición.

Además de las señaladas, los dos últimos botones le permiten voluntariamente abrir un archivo o guardar la edición del actual, mediante las teclas de Función 4 y 8 respectivamente.

➤ **Grabación automática del texto en edición**

Se implementó en la aplicación un Timer automático que a intervalos regulares grabe automáticamente el texto en edición en un archivo especial. En este sentido, la configuración inicial de este evento autodisparado se estableció en un tiempo de ciclo de 1 minuto, y el nombre del archivo en el que se graba el texto en edición es “TEXTO.TXT” y se encuentra ubicado en la misma carpeta de la aplicación. De esta manera, pueden quienes asisten al usuario salvar, recuperar y proponer diferentes contenidos para este que es un simple archivo plano de texto ASCII.

➤ **Operaciones de Copiar y Pegar sobre el texto en Edición**

La utilización de un control de texto plano multilineal como contenedor del texto en edición permite la implementación de las operaciones de edición requeridas con otras aplicaciones. De esta manera, la asistente pedagógica puede copiar el texto que el usuario edite en otros documentos de otras aplicaciones, y viceversa: disponer en el área de edición del usuario un texto copiado desde otro origen de datos. Cabe señalar que estas operaciones se basan en texto plano ASCII sin formato, lo cual es natural dada las características y aplicación de la solución desarrollada.

### 13.3. Versión 2.0 y subsiguientes

La segunda versión se generó a partir de los nuevos requerimientos académicos que debía enfrentar Lucas en la escuela, al pasar de grado.

Básicamente se aportó a la utilidad pedagógica / áulica de la herramienta dotando a la aplicación de una Calculadora integrada en el software, con el mismo sistema de utilización y características reactivas. En principio se trató de una calculadora básica, y en posteriores evoluciones de la versión se amplió la funcionalidad de la misma para posibilitar la edición y cálculo automático de resultado de fórmulas o ejercicios combinados, utilizando paréntesis para jerarquizar las operaciones de los cálculos a resolver.

La interfaz de uso principal de la aplicación se mantuvo, a excepción de un área de pantalla reservada para la “lectura visual” de los mensajes auditivos que recibía Lucas, permitiendo a su maestra integradora “ver” lo que Lucas “oye”. La interfaz de la calculadora desarrollada y sus características finales siguen a continuación:



### *13.3.1. Requerimientos adicionales para la segunda versión*

La lista de requerimientos iniciales se mantuvo, sin modificaciones sustanciales, pero para la liberación de esta segunda release se agregaron los siguientes requerimientos:

#### ❖ **Requerimientos Funcionales**

##### ➤ **Calculadora**

El sistema debía proveer de una calculadora que posibilitara la realización de las operaciones matemáticas fundamentales (suma, resta, multiplicación y división).

Posteriormente, en una segunda sub-versión, este requerimiento se amplió para contemplar la correcta resolución de ejercicios combinados, utilizando paréntesis para jerarquizar el orden de las operaciones.

Finalmente, respecto de las operaciones con resultados fraccionarios, el sistema debía presentar tanto la parte entera como los decimales de las mismas.

##### ➤ **Área de visualización de feedback auditivo**

Como el conjunto de informaciones provistas por vía auditiva al usuario resultaba creciente, y la interacción con su asistente dependía en gran medida de dichos elementos, el sistema debía poder “mostrar gráficamente” lo que el usuario “oye” como mecanismo de interfaz con el software.

##### ➤ **Implementación de diferentes voces sintéticas**

A los efectos de procurar diferentes combinaciones de timbre y velocidad de pronunciación, con el fin de hacer tan clara como fuera posible la voz sintética y atendiendo del mismo modo a algunos objetivos lúdicos, se solicitó que se pudiera configurar las características de la pronunciación del software.

➤ **Sonidos indentificatorios para las teclas del tablero**

Atendiendo a la creciente habilidad en el uso de la interfaz física (tablero especial) y auditiva (feedback por los auriculares), se solicitó que pudiera desactivarse la lectura de las texturas correspondientes a cada botón del tablero en el momento que el mismo era pulsado. No obstante, debía de algún modo indicarse al usuario la pulsión en cuestión, pues de no realizarse se produciría un efecto contraproducente en términos del *nivel de incertidumbre* respecto del efecto producido con las pulsiones sobre los botones.

En cuanto a los requisitos no funcionales de la aplicación, se incorporaron a la misma múltiples características configurables, por lo que fueron añadidos los siguientes requerimientos:

❖ **Requerimientos NO Funcionales**

➤ **Menú de Configuración General del Software**

Atendiendo a las características funcionales requeridas, el software debía disponer de un mecanismo sencillo y fácil de utilizar para la configuración del software, en cuanto a los aspectos de fonación sintética y los sonidos asociados a las teclas principales del tablero – como feedback auditivo para el usuario –.

➤ **Parametrización manual de características de comunicación**

La experiencia de uso en diferentes computadoras (la de su casa, la de la escuela, la NoteBook y otras de algunos familiares), se detectó que las características del puerto de comunicación utilizado diferían de uno a otro equipo, y si bien el tablero era simple de conectar y fácil de trasladar, la reconfiguración de cada uno de estos equipos se hacía dificultosa. Se requirió entonces un mecanismo de configuración para adaptar los parámetros

esperados por el software a diferentes puertos de comunicaciones y sus correspondientes características.

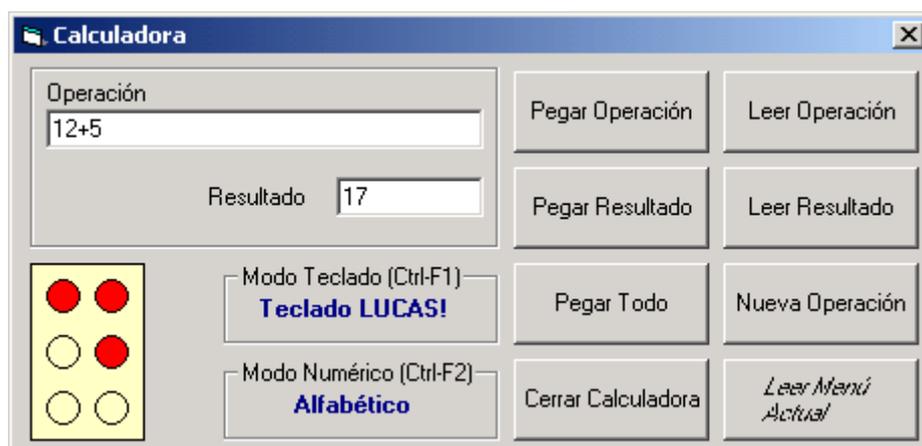
### 13.3.2. Características del Software

#### ➤ Calculadora Interactiva

La calculadora implementada en el software de lectoescritura tiene, como principal característica, la de mantener una interfaz de uso homogénea respecto del resto de la aplicación para su usuario: utilización mediante las teclas de puntos braille para el ingreso de los números y las operaciones a resolver, y manejo de las funcionalidades previstas en el menú comandando el mismo con la mitad derecha de su tablero.

En cuanto a la interfaz gráfica de este nuevo componente del software, también se respetó el estilo anterior, aunque utilizando una ventana de menores dimensiones, modal a la ventana principal del software de lectoescritura, con el objeto de gestionar adecuadamente su disponibilidad en el frente de la aplicación hasta que sea voluntariamente cerrada, para dar paso al programa de fondo. Este criterio de homogeneidad y coherencia permite que la curva de aprendizaje de la nueva herramienta sea sensiblemente menor a la correspondiente a cualquier otro estilo o características de interfaz.

Se presenta a continuación la interfaz específica de la calculadora:



Como podrá observarse, las funcionalidades implementadas incluyen la gestión de la operación matemática escrita en la misma y su resultado,

combinando el transporte – pegado – al texto principal en edición tanto de la operación como de su resultado en caso que sea conveniente para el usuario.

La implementación de la calculadora contó, como se reseñaba anteriormente, con una primera versión en la que sólo se permitían operaciones elementales (suma, resta, producto y cociente) entre dos operandos numéricos. La estrategia lógica del programa para resolver tales situaciones era bastante sencilla: identificar el signo de operación en la cadena y convertir ambos miembros de dicha operación (parte izquierda y derecha de la cadena) a valores numéricos que, según fuera la operación, se resolvían obteniendo y verbalizando el resultado.

Este modelo sólo se sostuvo durante un tiempo, hasta que fue necesaria una nueva evolución del software hacia una calculadora recursiva, capaz de resolver ejercicios combinados complejos. Esto radica en los requerimientos académicos y pedagógicos que fueron planteados por la maestra especial de Lucas. Así también fue necesario configurar el tratamiento de decimales cuando correspondiere, tanto en los resultados como en los valores numéricos ingresados como operandos para las distintas operaciones a realizar.

Las características sobresalientes del software desarrollado en cuanto al tratamiento sintáctico y analítico de las operaciones fueron las siguientes:

Analizador sintáctico recursivo

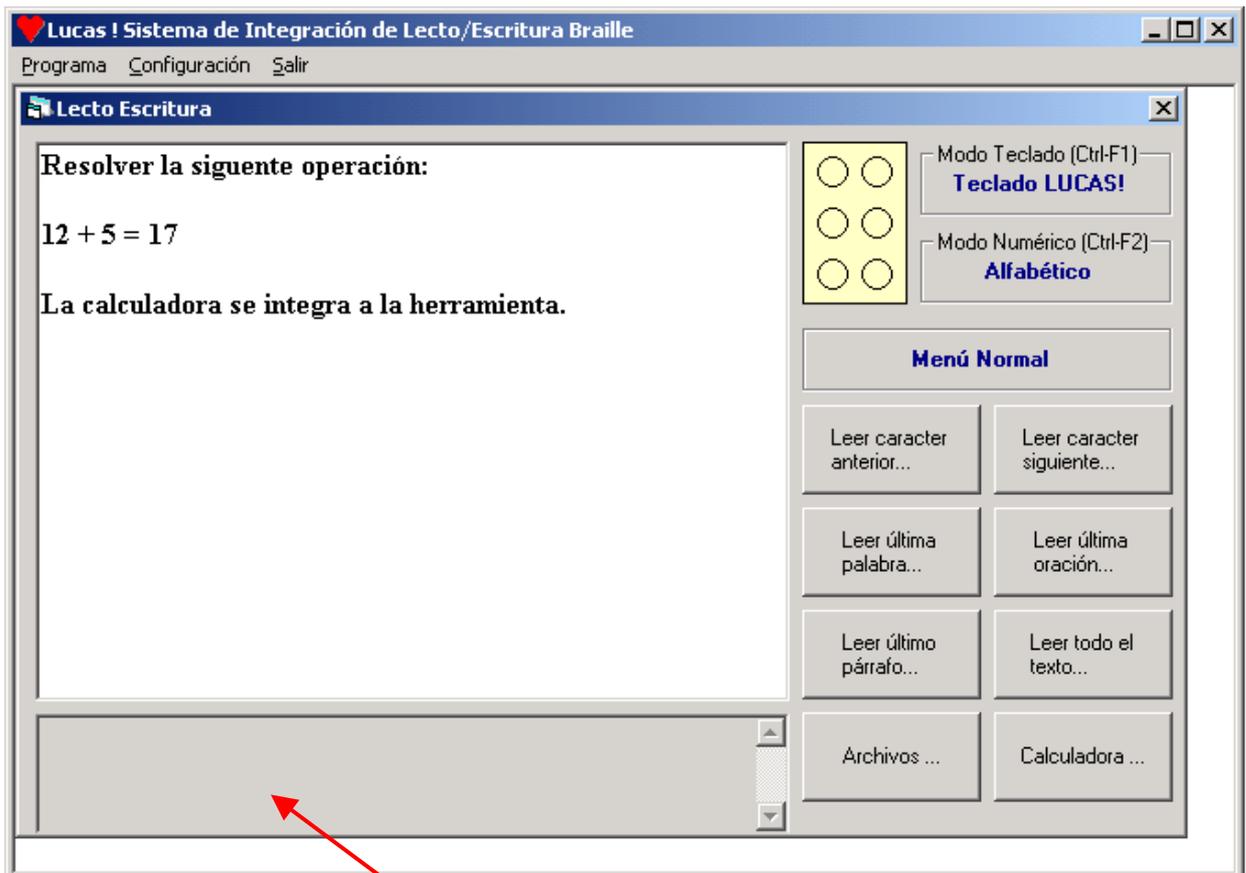
Compilador en tiempo de ejecución

La cadena de caracteres que compone la operación es analizada recursivamente, identificando en cada nueva llamada a la función analizadora el operador principal, y sus operandos, llamándose a sí misma para cada uno de estos operandos, hasta que el mismo sea un número simple. Una vez alcanzado el mínimo nivel, empiezan a devolverse los resultados parciales hasta llegar al nivel final con el resultado global de la operación. Dicho resultado es presentado en pantalla y verbalizado para

que el usuario pueda continuar la edición, realizar un nuevo cálculo, y/o pegar en el texto en edición ambos elementos.

### ➤ Área de FeedBack visual de operación

La implementación del área visual de feedback para la integradora de Lucas – o quien lo asista en el uso de la herramienta – se señala en la siguiente impresión de la pantalla.

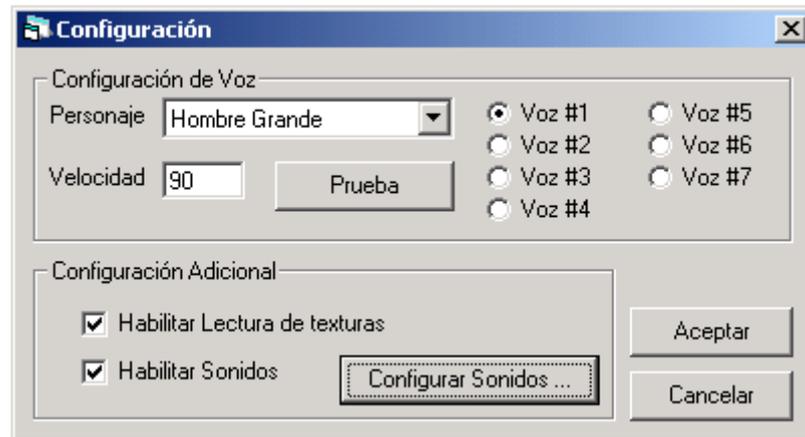


Esta es el área de comunicación visual al usuario secundario

El modo de funcionamiento es simple: se presentan en la caja de texto deshabilitada los mensajes que el usuario “oye” para que la otra persona pueda “verlos” sin necesidad de colocarse un auricular en paralelo.

### ➤ Menú de configuración de la Aplicación

Considerando la complejidad agregada a la aplicación en cuanto a sus aspectos de configuración, y atendiendo al requerimiento de poder modificarlos desde la aplicación misma, sin necesidad de utilizar herramientas adicionales ni complicados archivos de parametrización, se desarrolló e implementó una opción de menú que presenta el cuadro de diálogo que sigue a continuación, desde el que pueden configurarse los aspectos allí presentes de la aplicación.



Esta funcionalidad añadida atiende también al requerimiento de poder configurar diferentes voces para la aplicación, como así también las características del feedback auditivo para el usuario.

Para cada voz, como se observa en la gráfica, pueden configurarse básicamente dos atributos:

Estilo de la voz (define su timbre y otros elementos de fonación)

Velocidad de pronunciación

Desde la opción de “Configurar sonidos...” se accede a un segundo cuadro de diálogo en el que se puede especificar, seleccionar y probar el sonido que desee asignarse a la pulsión de cada tecla de la interfaz física de la aplicación.

Las categorías identificadas son:

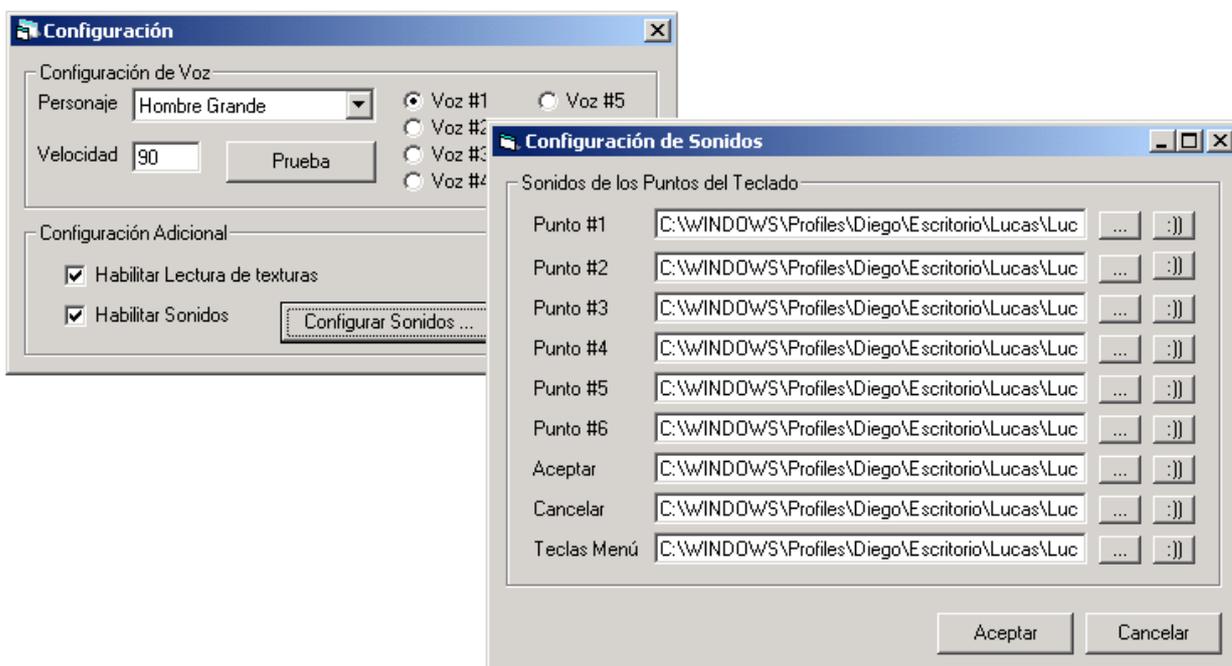
Sonido de cada tecla correspondiente a un punto braille

Sonido de la tecla de confirmación (espaciador)

Sonido de la tecla de cancelación (o BackSpace)

## Sonido de las teclas de menú

Sigue a continuación este cuadro de diálogo:



La totalidad de los parámetros configurados, más otros de uso interno de la aplicación, se registran en dos archivos planos de configuración, llamados Config.dat y Sonidos.dat

En el primero de ellos se establecen los aspectos de configuración del primer cuadro de diálogo, en tanto que en el segundo se graba la ruta absoluta de acceso a los archivos de sonido configurados. Ambos archivos deben estar disponibles en el mismo directorio de la aplicación. Cabe señalar asimismo que si por algún motivo estos archivos fueran eliminados involuntariamente, o alterados en su contenido, los mismos son regenerados automáticamente por la aplicación con los valores por defecto establecidos en el código mismo del software para todos los parámetros señalados.

### **13.4. Release Final: Versión 3.0**

Esta es la release actual del software, resultado de las múltiples evoluciones que el mismo tuvo desde su implementación inicial. En esta versión se implementaron básicamente tres aspectos nuevos bien diferenciados:

- Nuevo sistema de menús jerárquicos.
- Utilidades de movimiento del cursor sobre el texto y controles de edición del mismo.
- Ampliación de las características configurables en la aplicación.

Estas responden a los requerimientos que se señalan en el siguiente apartado. Considerando que esta es la versión actual, se presentan los documentos de especificaciones de casos de uso del software según la metodología UML señalada en el Marco Teórico y en la Estrategia Metodológica de desarrollo establecida al principio para el presente trabajo.

#### **13.4.1. Nuevos requerimientos para la versión 3.0**

Los requerimientos establecidos para las versiones anteriores de la herramienta, como así también los elementos del software que les dan cumplimiento se mantuvieron, cuando menos en líneas generales. Surgieron sí, de manera notable, los nuevos requerimientos se concentran en los puntos siguientes:

##### **➤ Operaciones de desplazamiento y edición del texto**

A partir de la experiencia de uso de la aplicación por parte del usuario, y del mecanismo de interacción con su maestra integradora, surgió un nuevo “concepto” manejado por él: el de la *posición del cursor en el texto*. Experiencias pedagógicas anteriores con el usuario – incluso de varios años – no le habían permitido manejar los conceptos de orden y posición relativa dentro de un conjunto con éxito sustancial. Lo sorprendente de esta aplicación es precisamente el efecto organizativo y las habilidades intelectuales despertadas en Lucas a partir del uso de esta herramienta.

En su interacción con la maestra empezó a familiarizarse con este nuevo concepto a partir de la situación de omisión involuntaria de una de las letras que componen una palabra y la interrupción de la maestra quien hacía explícito que era necesario corregir la posición del cursor para que él pudiera insertar allí la letra faltante. De esta manera fue aproximándose experiencialmente a este concepto tan abstracto. Luego, adquiriendo más habilidad en el manejo de un texto con sus oraciones y párrafos, y el abordaje de la redacción de textos como componente curricular en la escuela, esta habilidad fue trasladándose a la posición del cursor dentro del texto, para poder modificar lo ya redactado agregando o quitando palabras e incluso oraciones completas: tarea que debía requerir a su maestra integradora que realizara para él poder continuar.

Surge entonces esta necesidad como nuevo requerimiento: que pueda Lucas manejar la posición del cursor, tanto dentro de la palabra que está escribiendo como dentro del resto del texto, moviéndose en ambos sentidos (hacia atrás y hacia delante). También se hacía necesario que él pudiera voluntariamente borrar bloques completos de texto, sin necesidad de ir borrando una por una las letras, tal cual las prestaciones de la versión anterior del sistema.

El requerimiento establecía sólo lo anterior. No obstante, para implementar en la solución características que le permitieran llevar adelante tal diversidad de tareas, a la vez que algunos elementos adicionales que fueron propuestos – como la posibilidad de establecer “marcas” en el texto a los efectos de poder desplazarse en ambos sentidos entre tales marcas – se consideró el rediseño de la estructura de acceso a las funciones autocomandadas por el usuario en la aplicación, sin complicar la interfaz física de la solución. De esta manera se propuso la definición de un sistema de menú jerárquico donde el usuario pudiera elegir la categoría de acciones que deseaba llevar a cabo, y una vez elegida la misma,

poder seleccionar sí la acción en cuestión. Por ejemplo: si elegía la categoría de moverse hacia atrás en el texto, que tuviera en el siguiente menú la posibilidad de establecer si deseaba moverse un carácter hacia atrás, o una palabra, o una oración completa, o al párrafo anterior, a la marca anterior, o incluso al principio del texto. De idéntico modo para los movimientos relativos del cursor en la otra dirección y en las funciones de eliminación de texto.

➤ **Tratamiento sintáctico particular para la lectura de signos de puntuación**

Dadas las características de complejidad sintáctica que el usuario comenzó a trabajar en la escuela, fue requerido que los signos de puntuación pudieran ser leídos de manera explícita por la aplicación. Si bien el requerimiento llegaba hasta aquí, se consideró a la hora de darle forma a la implementación de una solución que esta característica podía ser deseable en determinados contextos de trabajo, e indeseable en otros. Por ejemplo, forzar la lectura de cada signo de puntuación en una actividad de lectura comprensiva, o incluso en lecturas que fueran acordes a sus intereses particulares podía resultar seguramente contraproducente, tanto por el tiempo que se agregaría innecesariamente al proceso, como por la distracción que dicha lectura produciría sobre el sentido del texto analizado.

➤ **Flexibilidad y características de configuración**

Finalmente, se solicitó – nuevamente por cuestiones más psicológicas que funcionales – que el usuario pudiera él mismo, sin intervención ajena, establecer la voz que quisiera oír.

Asimismo, se requería añadir algunas características de configuración adicionales, como por ejemplo el tipo de tratamiento

dado a la parte decimal de los resultados generados por la calculadora.

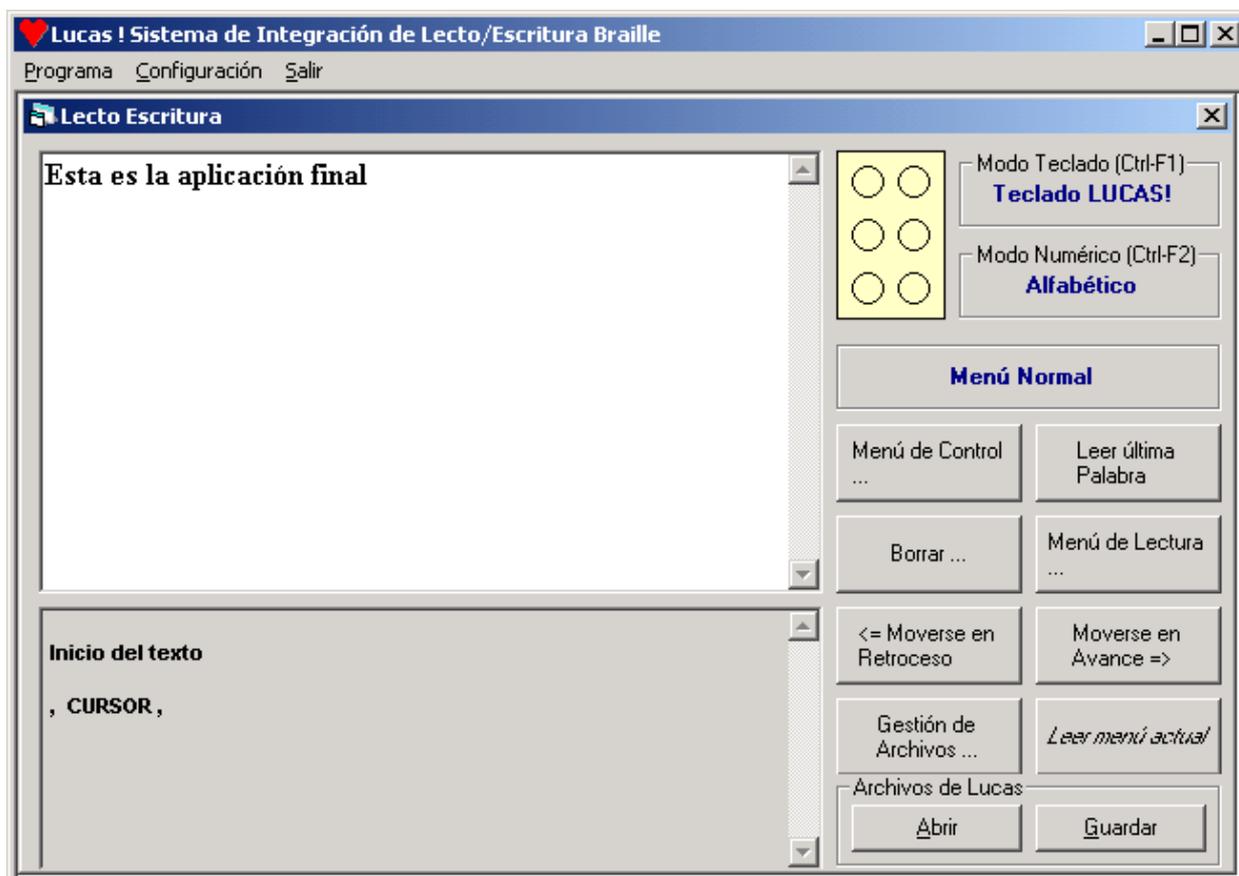
Considerando estos requerimientos, más otros aspectos de configuración específica resultantes de las demás características adicionales, resultó necesario el rediseño y ampliación de los cuadros de diálogo de configuración para dar soporte a estos nuevos elementos. Entre ellos se cuentan los siguientes aspectos:

- Posibilidad de establecer el tono de pronunciación de las voces
- Capacidad de indicar si el sistema debe leer con un tono y velocidad diferenciados los signos de puntuación que el resto del texto
- Posibilitar la función de lectura previa de las opciones de menú, hasta que el usuario se familiarizara con la disposición de las mismas, y que pudiera además repetir dicha lectura de un modo ágil, preciso e inequívoco.
- Hacer explícita también la lectura de los saltos de nuevo párrafo, y/o señalar los mismos de un modo simplificado para la lectura de corrido de un texto.
- Configurar si el tratamiento de decimales debía hacerse Redondeando, Truncando, o dejando los decimales reales de los resultados.
- Configurar si el dispositivo de interfaz física se conectaría al puerto COM1 o COM2 de la computadora (para aportar a la movilidad que seguía incrementándose).

Para satisfacer estos requerimiento se rediseño la interfaz y se desarrollaron nuevos elementos de software. Los mismos no se detallan dado que cada caso de uso del sistema se presentará modelado para esta última versión del software según se definió.

### 13.4.2. Prototipo de Interfaz

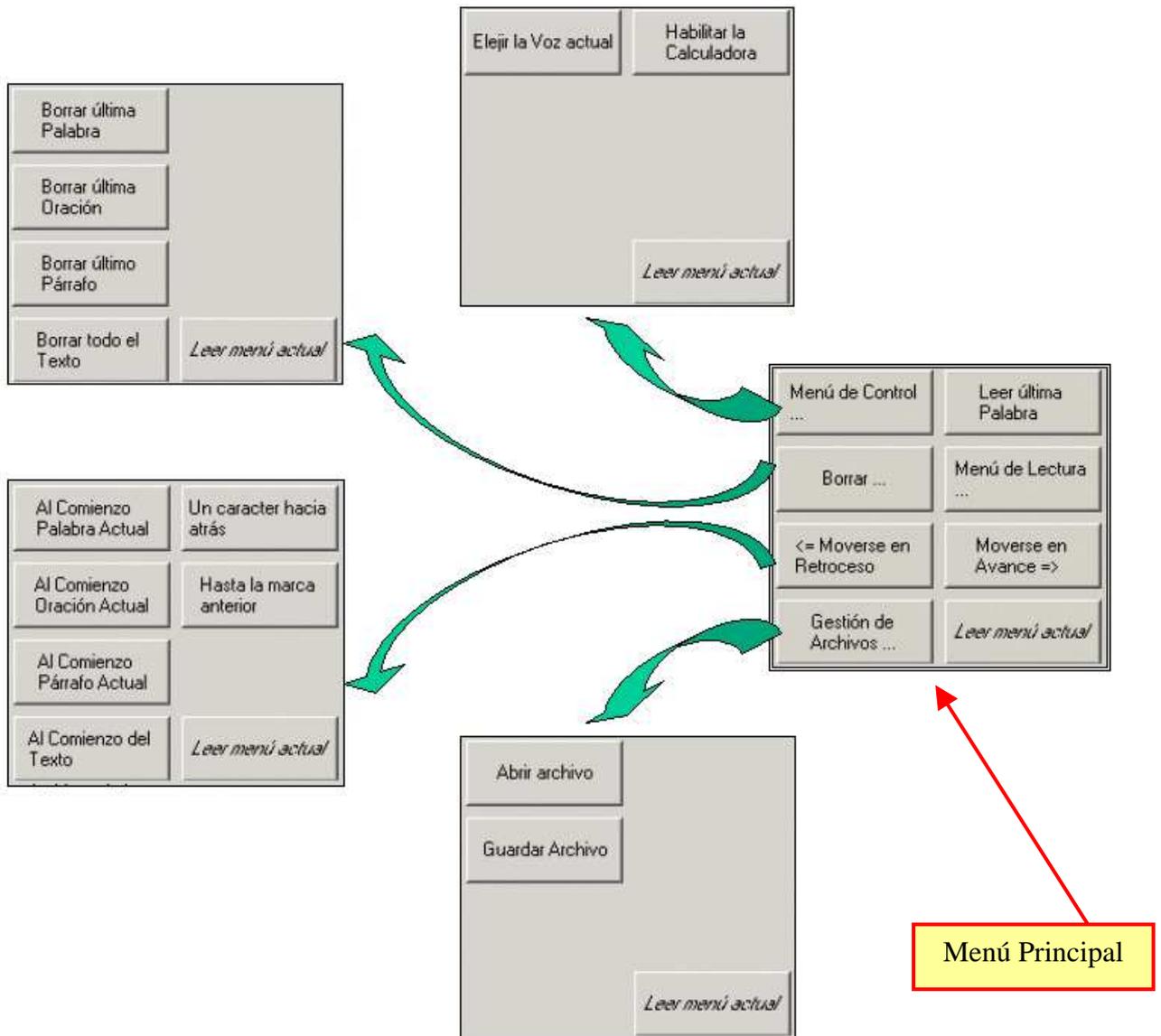
El prototipo final de la aplicación es el que se presenta a continuación.



La principal modificación que el mismo sufrió desde su versión precedente es la estructura de menú que se implementó. Evidentemente, al tratarse de un sistema de menú jerárquico y sensible al contexto, no puede representarse adecuadamente en una fotografía, por lo que se presenta a continuación un esquema de su estructura:

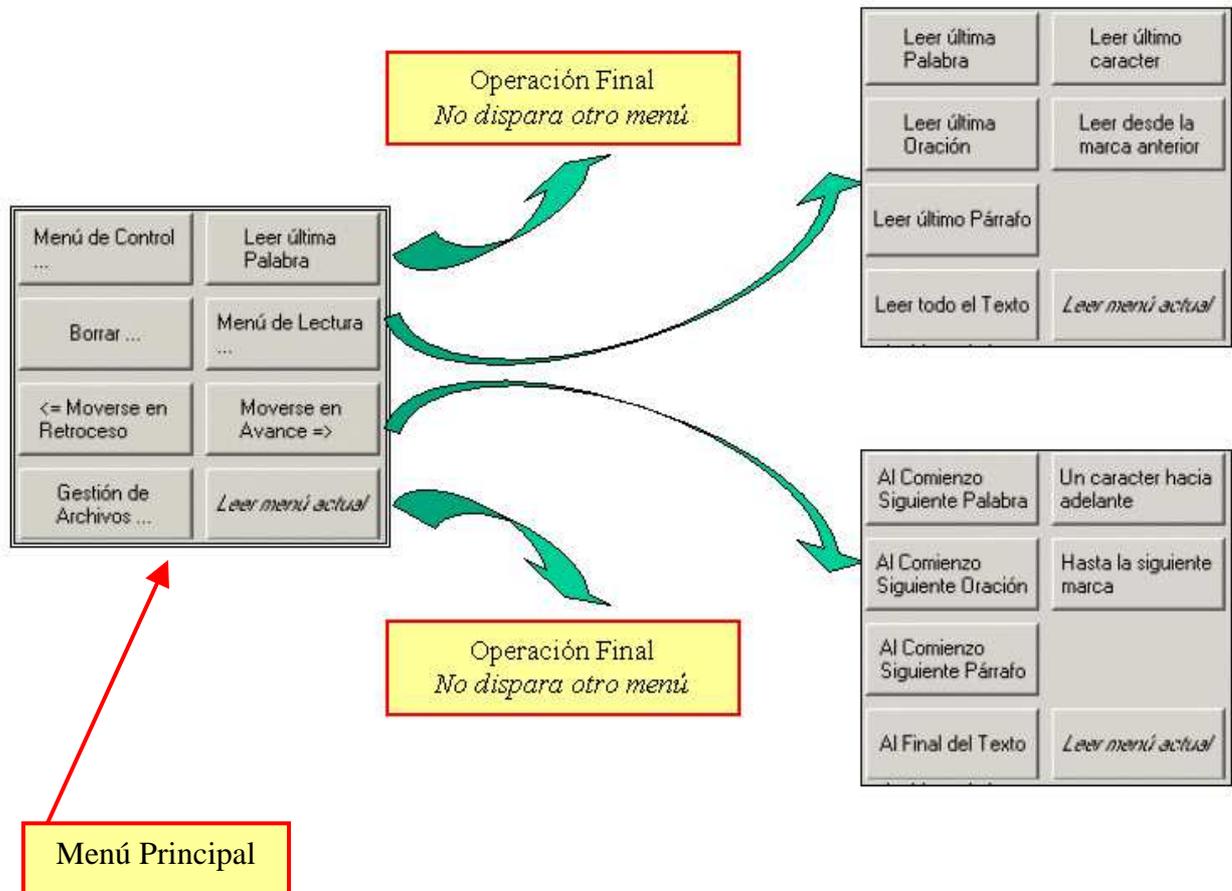
## Estructura de Menú Principal y SubMenús disponibles

Vista parcial: Opciones de Menú de los Botones Izquierdos



## Estructura de Menú Principal y SubMenús disponibles

Vista parcial: Opciones de Menú de los Botones Derechos



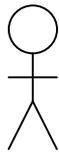
Asimismo, otro cambio que debió realizarse en la interfaz gráfica fue la ampliación del área de “lectura” de mensajes audibles, pues al incorporar las nuevas características señaladas anteriormente – fundamentalmente en cuanto a la lectura de los signos de puntuación y mensajes del sistema en la lectura de los menús – la cantidad de FeedBack auditivo que recibe el usuario se incrementó considerablemente, por lo que las reducidas dimensiones anteriores dejaban en muchas situaciones buena parte de los mensajes sin leer.

### 13.4.3. Modelos UML para los diferentes Casos de Uso

A continuación se detalla la estructura de casos de uso del sistema desarrollado, utilizando las herramientas metodológicas de UML según su pertinencia. Como es bien sabido e indicado en la misma metodología, los diagramas deben posibilitar la descripción de los casos de uso y sus interacciones en la medida que la complejidad de los mismos lo justifique. Por ello, si bien se considera a continuación la estructura completa de casos de uso del sistema, sólo se aplicarán los restantes diagramas para los casos más significativos, cuya complejidad requiera de los para el desarrollo y mantenimiento del software producido.

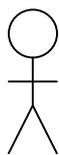
#### **Usuarios del Software**

El sistema sólo cuenta con dos usuarios normales, coherentemente con las interfaces que impone:



**Usuario Invidente**

El usuario normal y más frecuente del sistema es el propio Lucas, cuya interfaz con el software es auditiva, y su operación del sistema es realizada a través del tablero específicamente diseñado para él según sus capacidades motrices.



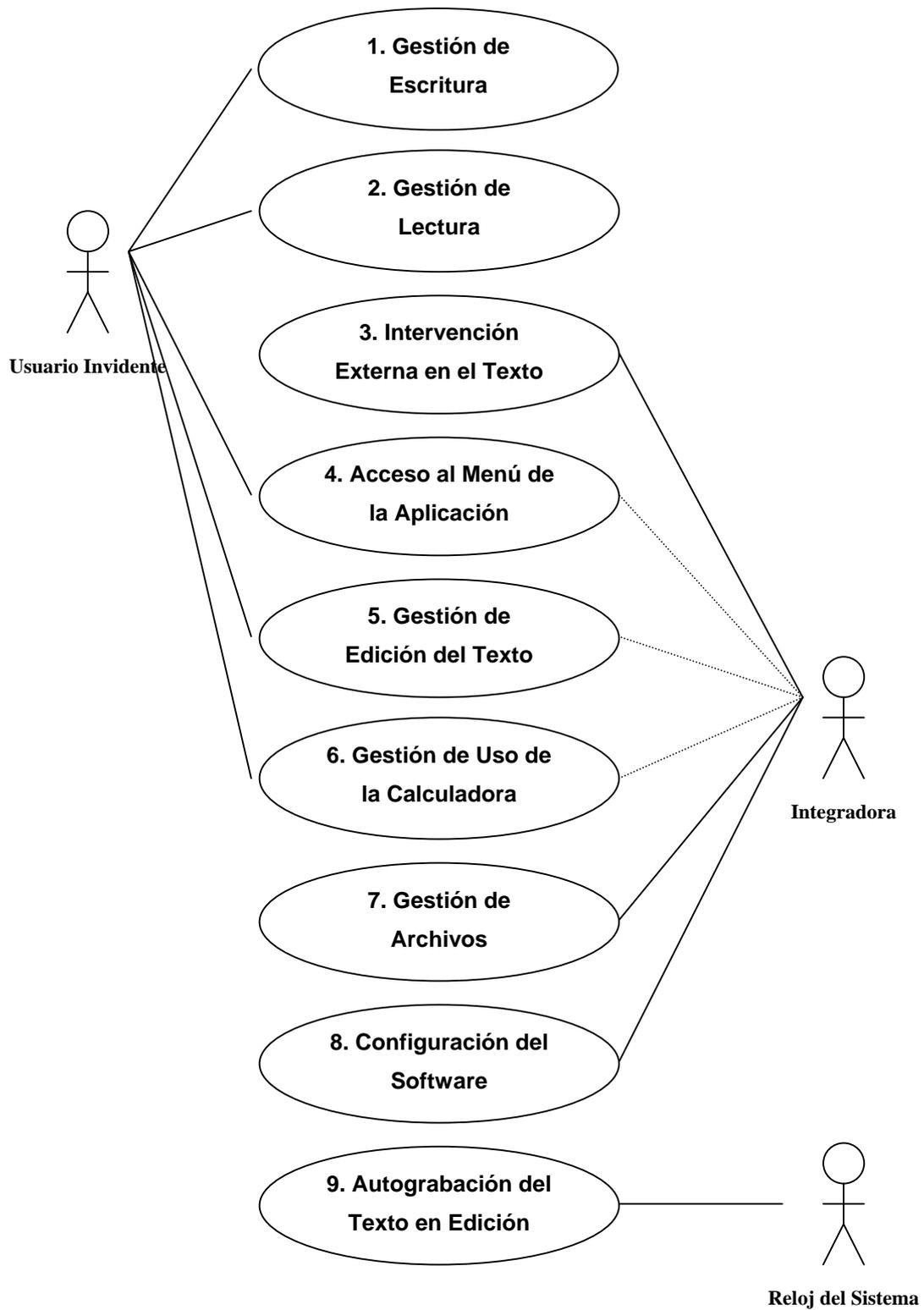
**Integradora**

El rol de usuario de la maestra integradora está concebido no sólo para la propia integradora, sino también para cualquier otra persona que colabore con Lucas en su utilización de la herramienta, tanto interviniendo en su proceso de lecto-escritura, como configurando las diferentes características del software para adecuarlo a las necesidades del usuario principal.

Adicionalmente, existe desde el punto de vista del análisis un tercer usuario que es el Reloj del Sistema, encargado de disparar un caso de uso particular (la grabación automática del texto en edición).

Estos serán entonces los actores considerados en la estructura de casos de uso que se define a continuación.

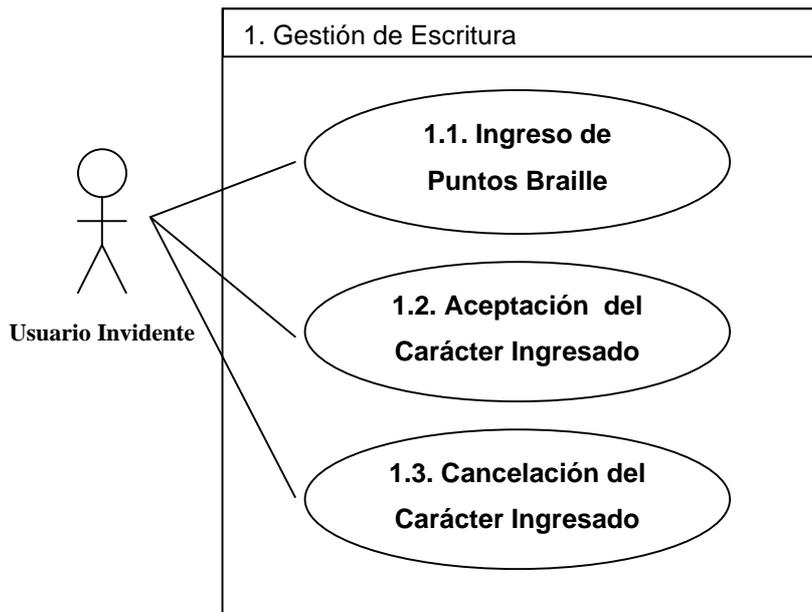
### Vista Global de Casos de Uso del Sistema



Las funcionalidad de los casos de uso identificados se descomponen según el siguiente detalle. Para cada burbuja final se detallará la funcionalidad requerida en las correspondientes fichas de caso de uso.

### **Sucesivas Explosiones componentes de los Casos de Uso**

#### **Explosión Caso de Uso 1. Gestión de Escritura**



La gestión de escritura secuencial en la aplicación se implementa con tres situaciones de uso bien diferenciadas:

- Ingreso de cada uno de los puntos braille que componen cada carácter
- Aceptación de los puntos ingresados (Espaciador Braille)
- Cancelación de los puntos braille ingresados cuando la escritura de un carácter se encuentra en progreso

Cabe señalar que esta última función operativa del botón de cancelación del tablero se encuentra “sobrecargada”, para que su funcionamiento sea dual: Si un carácter se encuentra en progreso, lo cancela; pero si no se encuentra ningún carácter en progreso, la funcionalidad del mismo botón es la de eliminar el carácter inmediato anterior a la posición del cursor.

A continuación se consideran las funcionalidades de los casos de uso considerados en relación a la Gestión de Escritura:

### CU: 1.1. Ingreso de Puntos Braille

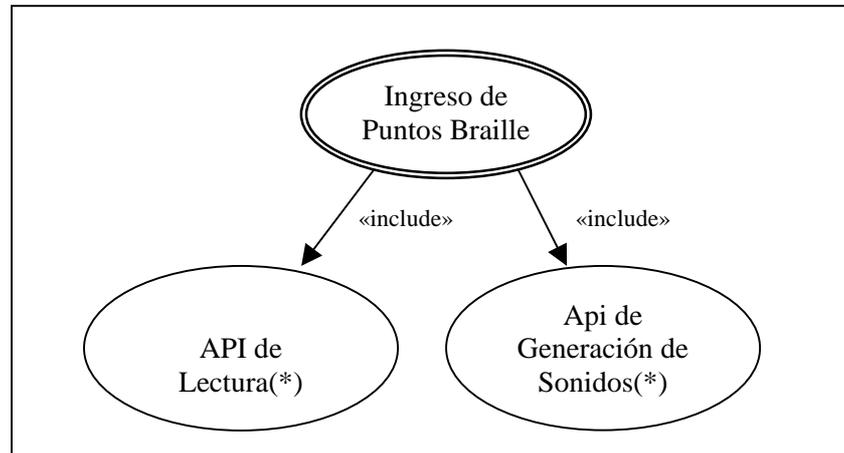
#### Ficha de Caso de Uso

<b>Caso de uso:</b> 1.1. Ingreso de Puntos Braille	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Ingresar al sistema uno de los seis puntos que conforman un carácter braille	
<b>Resumen:</b> Este caso de uso le permite al usuario, mediante sus sucesivas realizaciones, indicar al sistema cada uno de los puntos que conforman un carácter braille que desee escribir. Esta funcionalidad no incluye el ingreso propiamente dicho, si no la recepción por parte del software de los puntos que constituirán el carácter en caso que sea aceptado. Cuando el punto es seleccionado, el usuario recibe un feedback auditivo que confirma su selección, a la vez que en la pantalla se representa dicho punto como marcado (en color rojo) sobre la ficha braille (amarilla por default), de manera que el asistente pedagógico corrobore las acciones del usuario principal.	
<b>Requisitos:</b> El sistema debe encontrarse en estado de espera de puntos braille. Este es el estado natural del sistema. La única situación en la que no pueden ingresarse caracteres braille es cuando se ha activado el acceso al menú de la aplicación con el fin de acceder a alguna de sus funcionalidades.	
<b>Precondiciones:</b> El sistema se encuentra en estado de ingreso de puntos braille, es decir, con un carácter braille en progreso.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario pulsa uno de los seis botones correspondientes a los puntos braille en el tablero	1.1. El sistema le devuelve auditivamente la información sobre el punto braille seleccionado. Este feedback será el nombre de la textura correspondiente a dicho punto, o un sonido que lo caracterice, según se establezca en las opciones de configuración. 1.2. El punto braille ingresado es agregado a la matriz de puntos del carácter en progreso. Si el mismo no había sido previamente seleccionado, se selecciona, reflejando en pantalla el punto que le corresponde en la ficha braille graficada. 1.3. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Si se encontraba en progreso una operación de lectura, o el acceso a una función de menú, se aplican las excepciones.	
<b>Poscondiciones:</b> El estado del carácter braille en progreso es actualizado para incluir el punto que se añadió en la ejecución de este caso de uso, manteniendo en memoria los puntos que van formando el carácter a escribir.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El sistema se encontraba con una pronunciación (lectura) en progreso	1.1. La pronunciación es cancelada y se indica un beep audible (interpretado como situación de error o cancelación). El punto braille no es ingresado a la matriz de puntos ni reflejado gráfica ni auditivamente. 1.2. Fin del caso de uso.
2. El sistema se encontraba en estado de acceso a funciones de menú: había sido presionada alguna tecla de función y un menú diferente al estándar se presentaba al usuario.	2.1. El menú en curso es cancelado y la aplicación retorna al estado normal (por default). Se indica un beep audible (interpretado como situación de error o cancelación). 2.2. Fin del caso de uso.
<b>Casos de uso utilizados:</b> «ninguno»	

**Interfaces requeridas:**

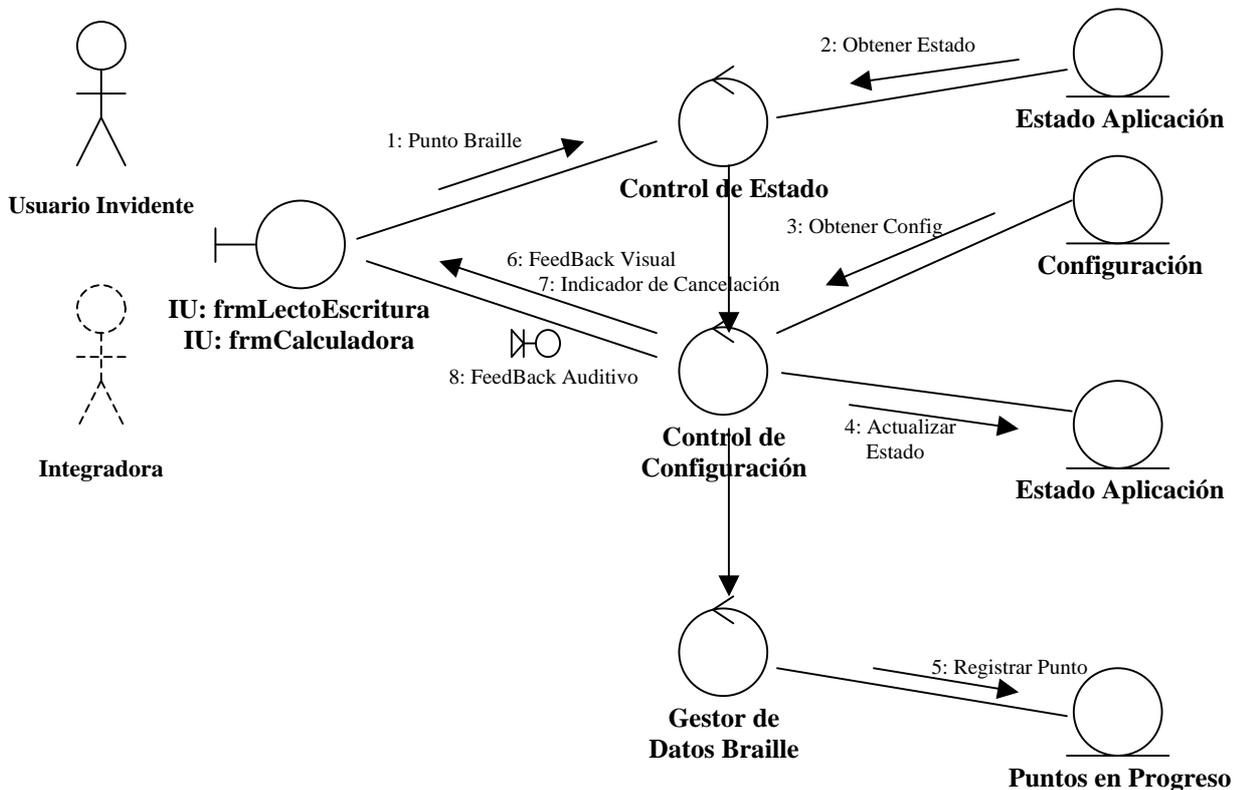
- 1. Formulario principal de Lecto-Escritura
- 2. Formulario de Calculadora (alternativamente)

*Diagrama de Relaciones de CU: 1.1. Ingreso de Puntos Braille*



(\*) Estas API se incluyen en todas las relaciones de CU de los que el usuario no vidente participa activamente, por lo que no se referirán adicionalmente en los diagramas posteriores para mantener la simplicidad de los modelos.

*Diagrama de Colaboración CU: 1.1. Ingreso de Puntos Braille*



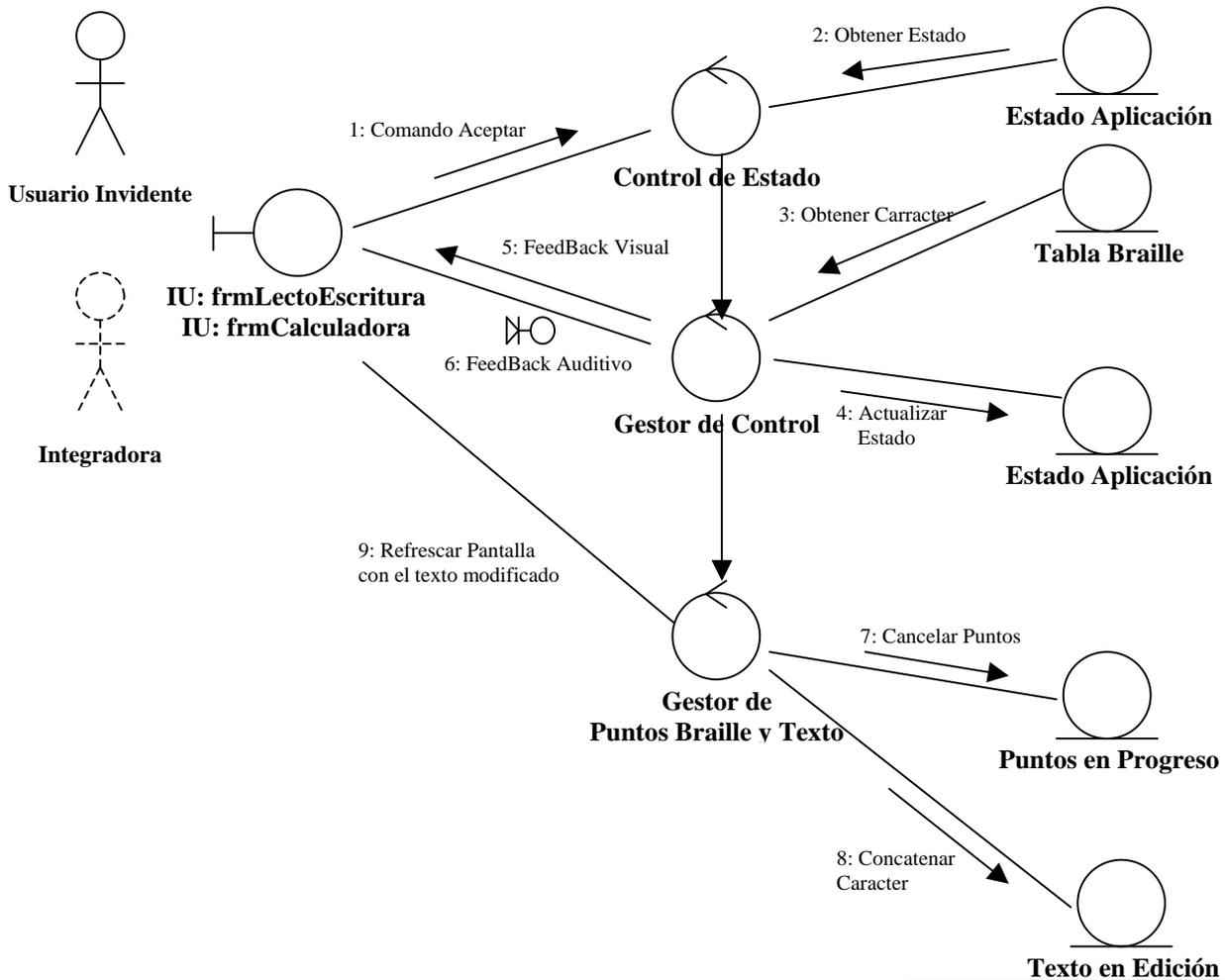
## CU: 1.2. Aceptación del Carácter Ingresado

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 1.2. Aceptación del Carácter Ingresado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Confirmar la aceptación del carácter braille ingresado y añadirlo al texto en la posición del cursor	
<b>Resumen:</b> Mediante la implementación de este caso de uso el usuario indica al sistema que acepte el carácter braille ingresado – compuesto por el conjunto de puntos braille ingresados en el caso de uso 1.1. Si no se ingresó ningún punto braille, el carácter aceptado es el espacio. El sistema identificará el carácter correspondiente a los puntos ingresados –según definición de la tabla braille implementada en el sistema – y lo añadirá al texto en edición en la posición actual del cursor. Para el caso que los puntos ingresados no se correspondan con ningún carácter, informará al usuario del error y reestablecerá a cero el conjunto de puntos ingresados para que se vuelva a intentar la operación seleccionando los puntos correctos.	
<b>Requisitos:</b> El sistema debe encontrarse en estado de espera de puntos braille, con o sin puntos ya ingresados. Este es el estado natural del sistema. La única situación en la que no pueden aceptarse caracteres braille es cuando se ha activado el acceso al menú de la aplicación con el fin de acceder a alguna de sus funcionalidades.	
<b>Precondiciones:</b> El sistema se encuentra en modo normal, con ninguno, uno o varios puntos braille ingresados, correspondientes a realizaciones anteriores de caso de uso 1.1.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario pulsa el botón de aceptación del carácter braille (espaciador braille)	1.1. El sistema analizará el conjunto de puntos brailles seleccionados hasta el momento, buscando su patrón en la tabla Braille definida en la aplicación. También se considerará si el sistema se encuentra en Modo Numérico (por haber sido introducido previamente un carácter braille de número), para la correcta lectura.
	1.1.1. Si el sistema no se encuentra en modo numérico, se verbalizará como feedback auditivo para el usuario el nombre del carácter braille ingresado.
	1.1.2. Si el sistema se encuentra en modo numérico (ya sea por haberse introducido anteriormente un carácter de número, como por intervención externa estableciendo manualmente el estado de modo numérico), se verbalizará como feedback auditivo para el usuario el nombre del número correspondiente al carácter braille ingresado.
	1.2. El carácter correspondiente será concatenado en el texto en edición en la posición del cursor. Si el modo numérico estaba activado, se concatenará el número que le corresponde.
	1.3. El conjunto de puntos braille de carácter es puesto en cero (y su para dar paso al ingreso de un nuevo carácter.
	1.4. Fin del caso de uso
<b>Nota sobre el escenario:</b> Si se encontraba en progreso una operación de lectura, o el acceso a una función de menú, se aplican las excepciones 3 y 4.	
<b>Poscondiciones:</b> El carácter correspondiente a los puntos braille que se habían ingresado es concatenado al texto en la posición del cursor, y el usuario se percata de tal acción por el feedback recibido.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El conjunto de puntos braille ingresados no se corresponde con un carácter de la tabla braille del sistema.	1.1. Se informa de manera auditiva la situación de error al usuario y se inicializa el conjunto de puntos braille para que proceda al ingreso nuevamente. La situación se refleja también en la interfaz gráfica (blanqueo de los puntos braille).

<b>Caso de uso: 1.2. Aceptación del Carácter Ingresado</b> <span style="float: right;">(Cont...)</span>	
2. El sistema se encuentra en Modo Numérico y se pretende dar aceptación a un carácter no permitido (no correspondiente a las letras "A" a "J")	2.1. Se informa de manera auditiva la situación de error al usuario y se inicializa el conjunto de puntos braille para que proceda al ingreso nuevamente. La situación se refleja también en la interfaz gráfica (blanqueo de los puntos braille). 2.2. Fin del caso de uso.
3. El sistema se encontraba con una pronunciación (lectura) en progreso	3.1. La pronunciación es cancelada y se indica un beep audible (interpretado como situación de error o cancelación). El punto braille no es ingresado a la matriz de puntos ni reflejado gráfica ni auditivamente. 3.2. Fin del caso de uso.
4. El sistema se encontraba en estado de acceso a funciones de menú: había sido presionada alguna tecla de función y un menú diferente al estándar se presentaba al usuario.	4.1. El menú en curso es cancelado y la aplicación retorna al estado normal (por default). Se indica un beep audible (interpretado como situación de error o cancelación). 4.2. Fin del caso de uso.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i> 2. <i>Formulario de Calculadora (alternativamente)</i>	

*Diagrama de Colaboración CU: 1.2. Aceptación del Carácter Ingresado*



## CU: 1.3. Cancelación del Carácter Ingresado

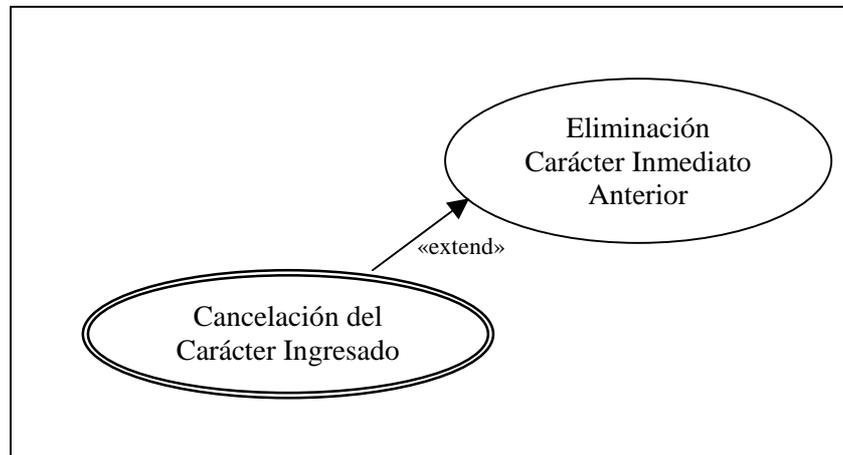
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 1.3. Cancelación del Carácter Ingresado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Abortar el conjunto de puntos braille ingresados para el carácter en progreso.	
<p><b>Resumen:</b> Este caso de uso le permite al usuario abortar la escritura de un carácter braille que se encontraba en progreso. Esto es: habiendo ingresado al sistema uno o más puntos braille de los que componen un carácter, poder cancelar dicho ingreso devolviendo la matriz de puntos a su estado inicial (sin puntos seleccionados). El usuario utilizará esta funcionalidad cuando en progreso de escritura de un carácter se de cuenta de que ingresó un punto que no correspondía al carácter que quisiera escribir, o en su defecto, en el progreso de ingreso se arrepiente y decide cancelar.</p> <p>La iniciación de este caso de uso se produce al presionar la tecla de cancelación/borrado de su tablero, que como su nombre lo indica, permite también otra funcionalidad, por lo que en función del estado de la aplicación este caso de uso podrá extender al de eliminación del carácter inmediato anterior a la posición del cursor, cancelando su funcionalidad.</p>	
<b>Requisitos:</b> El sistema debe encontrarse en estado de espera de puntos braille, con al menos un punto braille ya ingresado. Si el sistema no se encuentra con un carácter en progreso, la funcionalidad del botón que dispara este caso de uso será diferente.	
<b>Precondiciones:</b> El sistema se encuentra en estado de ingreso de un carácter, con uno o varios puntos braille preseleccionados, correspondientes a realizaciones anteriores del caso de uso 1.1.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario pulsa el botón correspondiente a la funcionalidad de Cancelación/Borrado en el tablero de control de la aplicación.	1.1. La matriz de puntos braille se inicializa (todos los puntos ingresados son cancelados), lo que se refleja en la ficha braille en pantalla. 1.2. El sistema le devuelve auditivamente un sonido que indica la cancelación de la operación de ingreso de puntos braille que venía en curso. 1.3. Fin del caso de uso
<p><b>Nota sobre el escenario:</b> Si se encontraba en progreso una operación de lectura, o el acceso a una función de menú, se aplican las excepciones.            Si la aplicación no se encontraba en edición de un nuevo carácter, se aplican las excepciones.</p>	
<b>Poscondiciones:</b> El sistema queda en estado listo para el ingreso de puntos braille, sin ningún punto preseleccionado.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El sistema se encontraba con una pronunciación (lectura) en progreso	1.1. La pronunciación es cancelada y se indica un Bel audible (interpretado como situación de error o cancelación). La matriz de puntos braille queda en estado inicializada: sin ningún punto preseleccionado. 1.2. Fin del caso de uso
2. El sistema se encontraba en estado de acceso a funciones de menú: había sido presionada alguna tecla de función y un menú diferente al estándar se presentaba al usuario.	2.1. El menú en curso es cancelado y la aplicación retorna al estado normal (por default) sin ningún punto braille preseleccionado. Se indica un beep audible (interpretado como situación de error o cancelación). 2.2. Fin del caso de uso
3. El sistema se encontraba en estado de espera de puntos braille, pero sin ningún punto aún seleccionado.	3.1. El control es transferido al caso de uso de eliminación del carácter inmediato anterior a la posición del cursor (5.1.1) 3.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «extend» 5.1.1. Eliminación Carácter Inmediato Anterior	

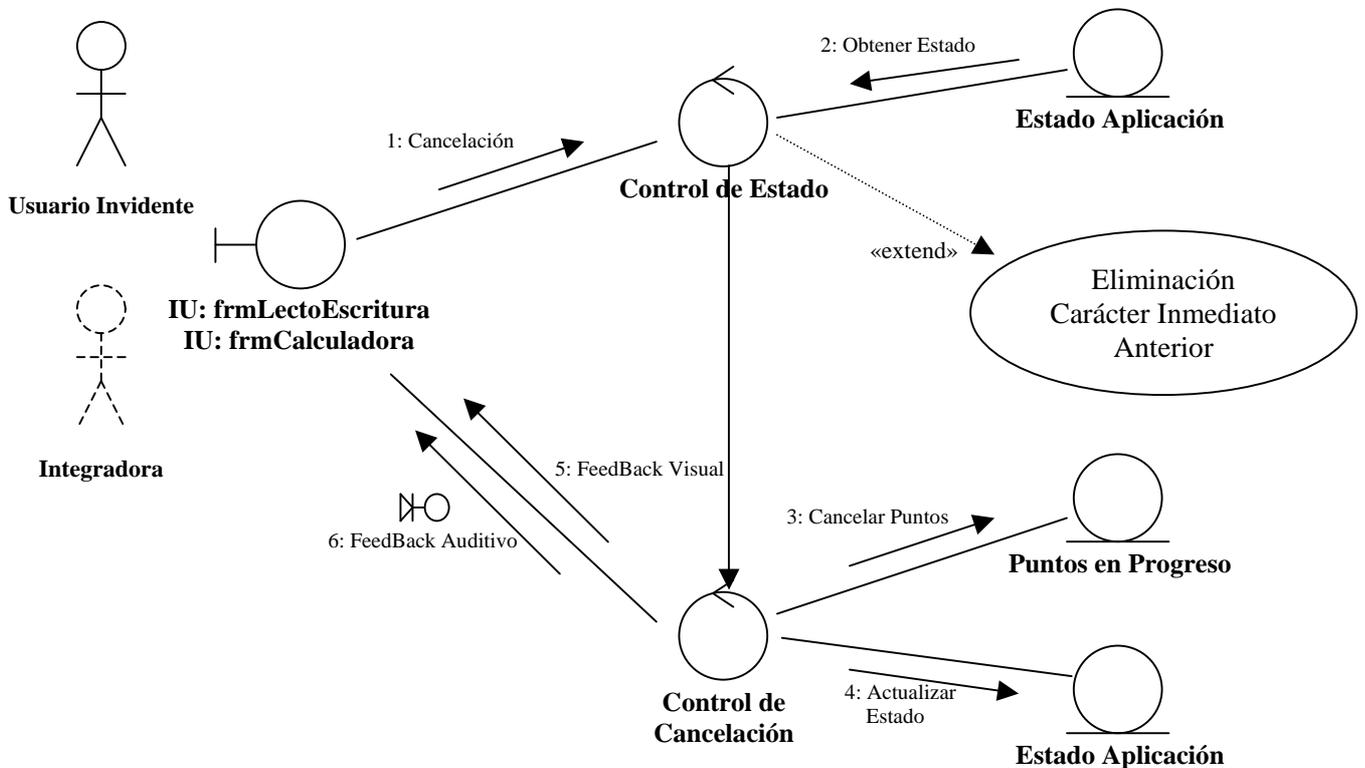
**Interfaces requeridas:**

- 1. Formulario principal de Lecto-Escritura
- 2. Formulario de Calculadora (alternativamente)

*Diagrama de Relaciones de CU: 1.3. Cancelación del Carácter Ingresado*

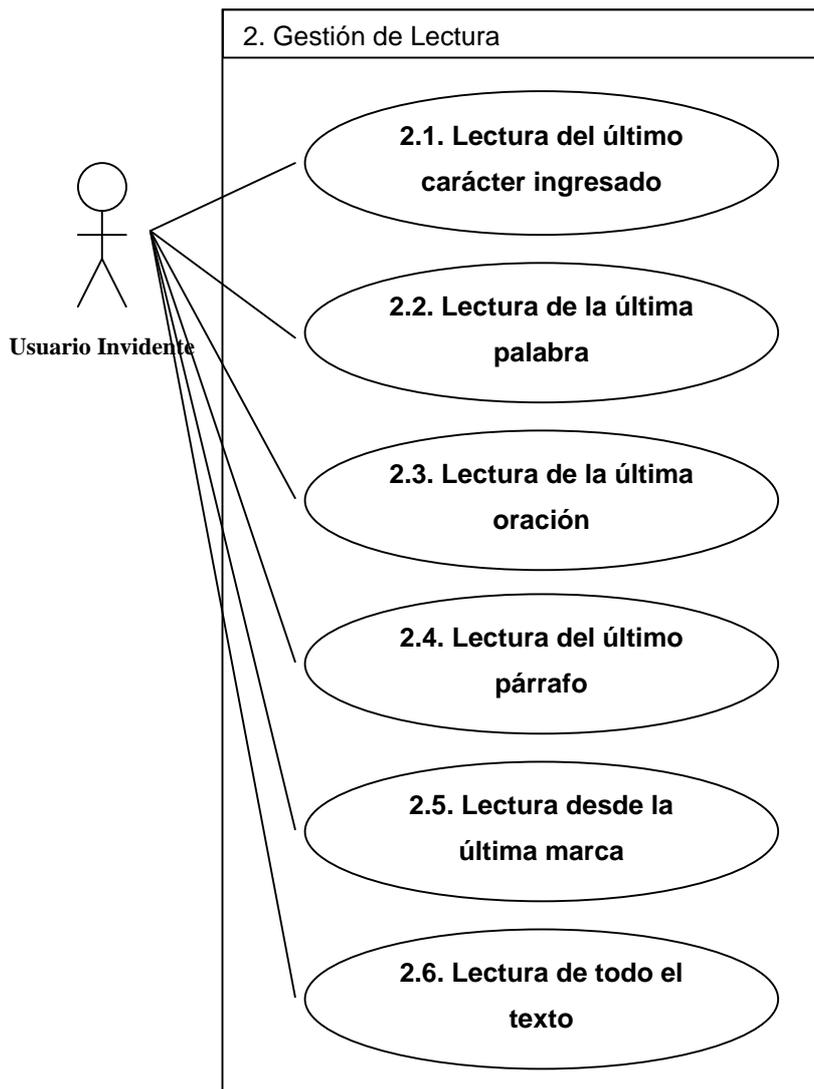


*Diagrama de Colaboración CU: 1.3. Cancelación del Carácter Ingresado*





## **Explosión Caso de Uso 2. Gestión de Lectura**



Las funcionalidades de lectura (verbalización por voz sintética) de la aplicación se implementan según el contexto elegido por el usuario. En este sentido, se ofrecen como posibilidades las siguientes subcadenas del texto en edición, en función de la posición en que se encuentre el cursor:

- Lectura del carácter inmediato anterior a la posición del cursor.
- Lectura de la palabra sobre la cual se encuentra el cursor.
- Lectura de la oración sobre la cual se encuentra el cursor.
- Lectura del párrafo sobre el cual se encuentra el cursor.
- Lectura desde el último marcador de posición existente en el texto, y hasta el próximo. (bloque que contiene al cursor)

- Lectura de todo el texto.

En todas las operaciones de lectura solicitadas, el software lee de corrido (con fonación sintética) y utilizando la voz elegida por el usuario hasta la posición del cursor, luego hace explícita la misma – indicando la palabra “CURSOR” con un tono diferenciado – y luego continúa con la lectura de corrido hasta el final del contexto indicado. Asimismo, y en función de las características de configuración que se establezcan, se hace también explícita la lectura del *nombre de los signos de puntuación* – utilizando una velocidad de pronunciación y timbre de voz diferenciables y configurables por el usuario –, para fortalecer los aspectos gramaticales de la lectura.

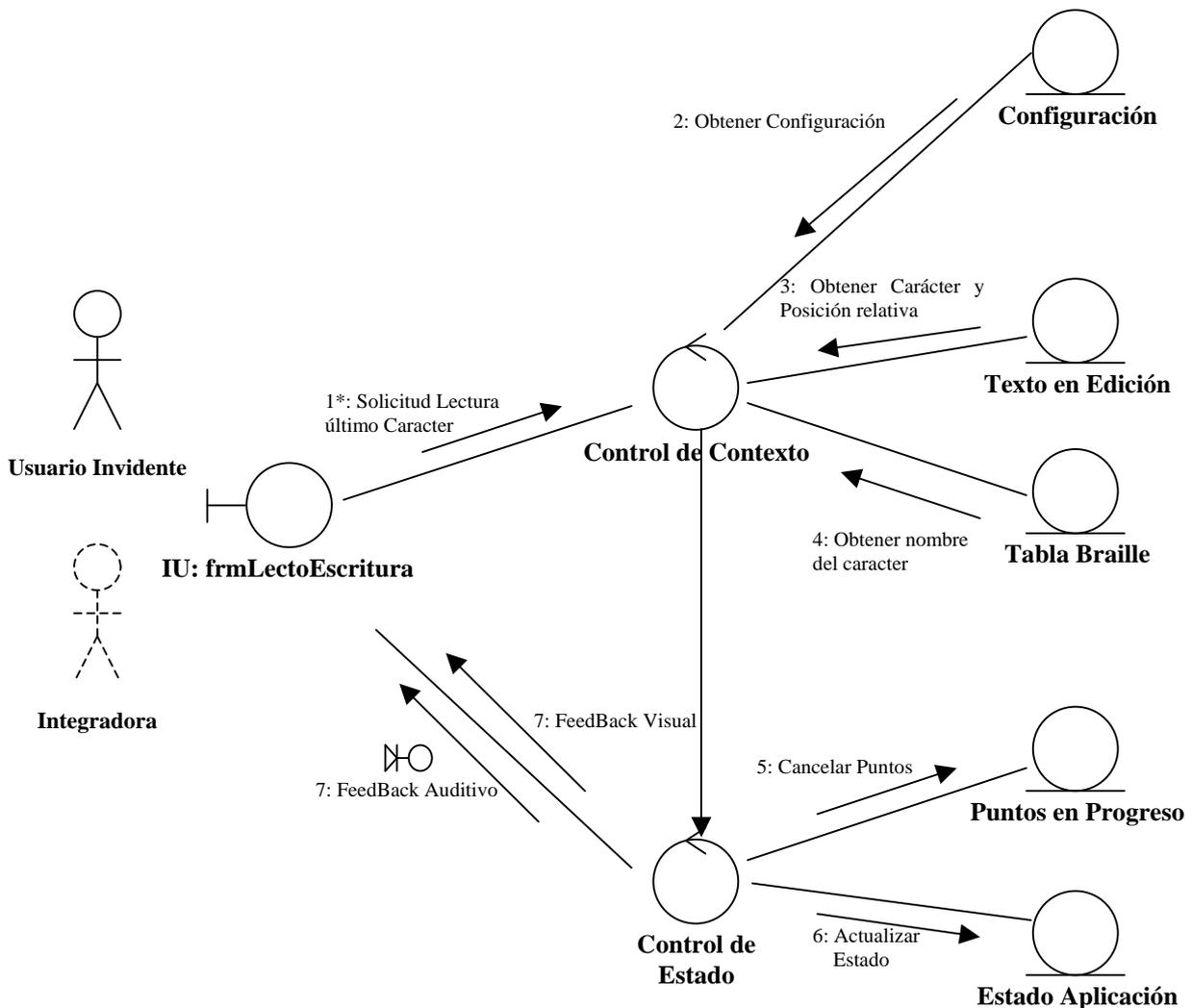
Cabe señalar que independientemente de cual fuere la operación de lectura desarrollada, mientras la misma se encuentra en progreso el usuario dispone de la funcionalidad de abortarla, para continuar con la edición del texto o el uso de la calculadora.

## CU: 2.1. Lectura del último carácter ingresado

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.1. Lectura del último carácter ingresado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, el nombre el carácter inmediato anterior a la posición del cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación conocer el carácter que se encuentra inmediatamente a la izquierda de la posición actual del cursor. Asimismo, si el cursor se encuentra ubicado al principio del texto, o al final del mismo, también se lo indica de manera verbal.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
<p>1. El usuario indica al sistema el comando de menú de leer el último carácter ingresado. Esto puede realizarse de dos maneras diferentes:</p> <p>a. Estando en estado de edición, con el menú normal de la aplicación activo: pulsando sobre la tecla definida en la esquina superior derecha del tablero (función #5).</p> <p>b. Seleccionando la tecla de menú de lectura (tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer último carácter" (con la tecla de función #5)</p>	1.1. El sistema analiza la posición del cursor en el texto, y se desencadena la pronunciación del nombre del carácter que se encuentra inmediatamente a la izquierda del cursor.
	1.2. Se añade a la cadena a pronunciar, con timbre y velocidad diferenciados y en función de las características de configuración, la palabra "CURSOR" para indicar la posición del mismo.
	1.3. La verbalización entregada como feedback al usuario es también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encontraba al principio del texto, es decir, justo después de ningún carácter. Esto también aplica para cuando el texto en edición no contenga ningún carácter (texto en blanco).	1.1. Como no hay carácter previo a pronunciar, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.2 del caso de uso. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso
2. El cursor se encontraba al final del texto en edición, justo después del último carácter del mismo. Esto también aplica para cuando el texto en edición no contenga ningún carácter (texto en blanco).	2.1. A las palabras pronunciadas como feedback auditivo para el usuario se concatena la indicación de "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	

*Diagrama de Colaboración CU: 2.1. Lectura del último carácter ingresado*



Referencias:

**1\*:** **Solicitud Lectura último carácter.** Este mensaje es implementado de manera indirecta por el usuario, mediante la realización por parte de éste del caso de uso **4. Acceso al menú de la aplicación**, tanto en su utilización directa de la función #5, como desde el acceso al menú de lectura (función #4) y desde éste a la función #5.

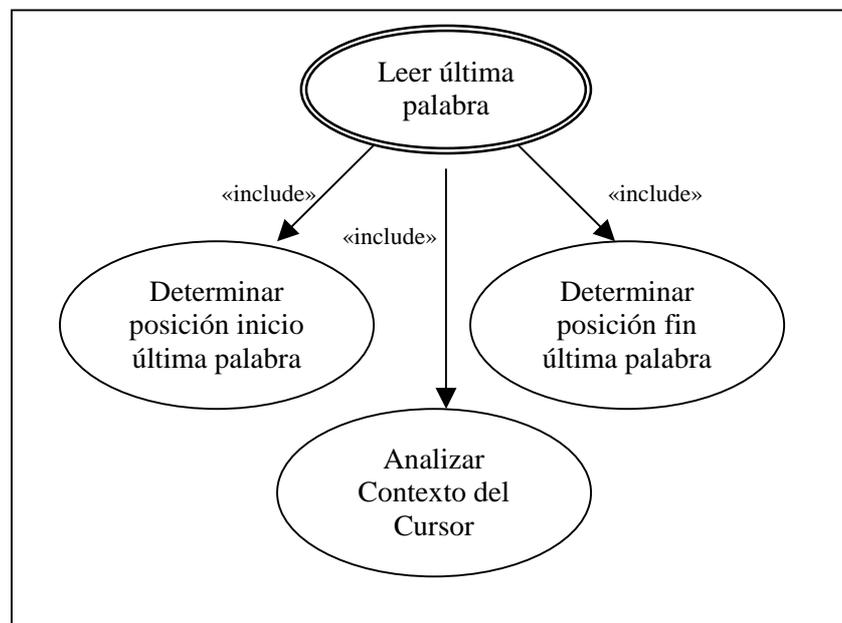
## CU: 2.2. Lectura de la última palabra

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.2. Lectura de la última palabra	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, la palabra sobre la que se encuentra situada el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación reconocer la palabra sobre la que se encuentra situada el cursor. Esto es: cuando el cursor se encuentra entre dos letras que la componen, o justo después del último carácter de la misma. Si el cursor se encuentra justo antes de la primer letra de una palabra, la lectura deberá indicar el carácter inmediato anterior a dicha palabra. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario. Lo mismo se aplica para cuando el cursor se encuentre posicionado al final del texto.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>leer la última palabra</u> . Esto se realiza seleccionando la tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer última palabra" (con la tecla de función #1)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra justo antes del primer carácter de una palabra, establece el nombre del carácter inmediato anterior para su pronunciación.  b. Si el cursor se encuentra entre dos caracteres (posición intermedia) de una palabra, determina las posiciones del texto en que comienza y termina dicha palabra, junto a la posición relativa del cursor sobre la misma.  c. Si el cursor se encuentra exactamente al final de una palabra, determina la posición de inicio y fin de la misma.
	1.2. Se genera la verbalización (pronunciación) de la palabra, determinada por la subcadena del texto comprendida entre el inicio y el fin determinado según la lógica explicada, siguiendo el siguiente esquema:  a. Para el caso 1.1.a. se pronuncia el nombre del carácter inmediato anterior a la posición del cursor, y se añade a la cadena a pronunciar, con timbre y velocidad diferenciados y en función de las características de configuración, la palabra "CURSOR" para indicar la posición del mismo.  b. Para el caso 1.1.b. se pronuncia la parte de la palabra comprendida desde el inicio de la misma hasta la posición del cursor, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo. Y finalmente se finaliza la verbalización con la parte de la palabra comprendida entre la posición del cursor y el final de la misma.  c. Para el caso 1.1.c. se pronuncia de corrido la palabra comprendida entre la posición de inicio y la posición de fin determinadas, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo.
	1.3. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.

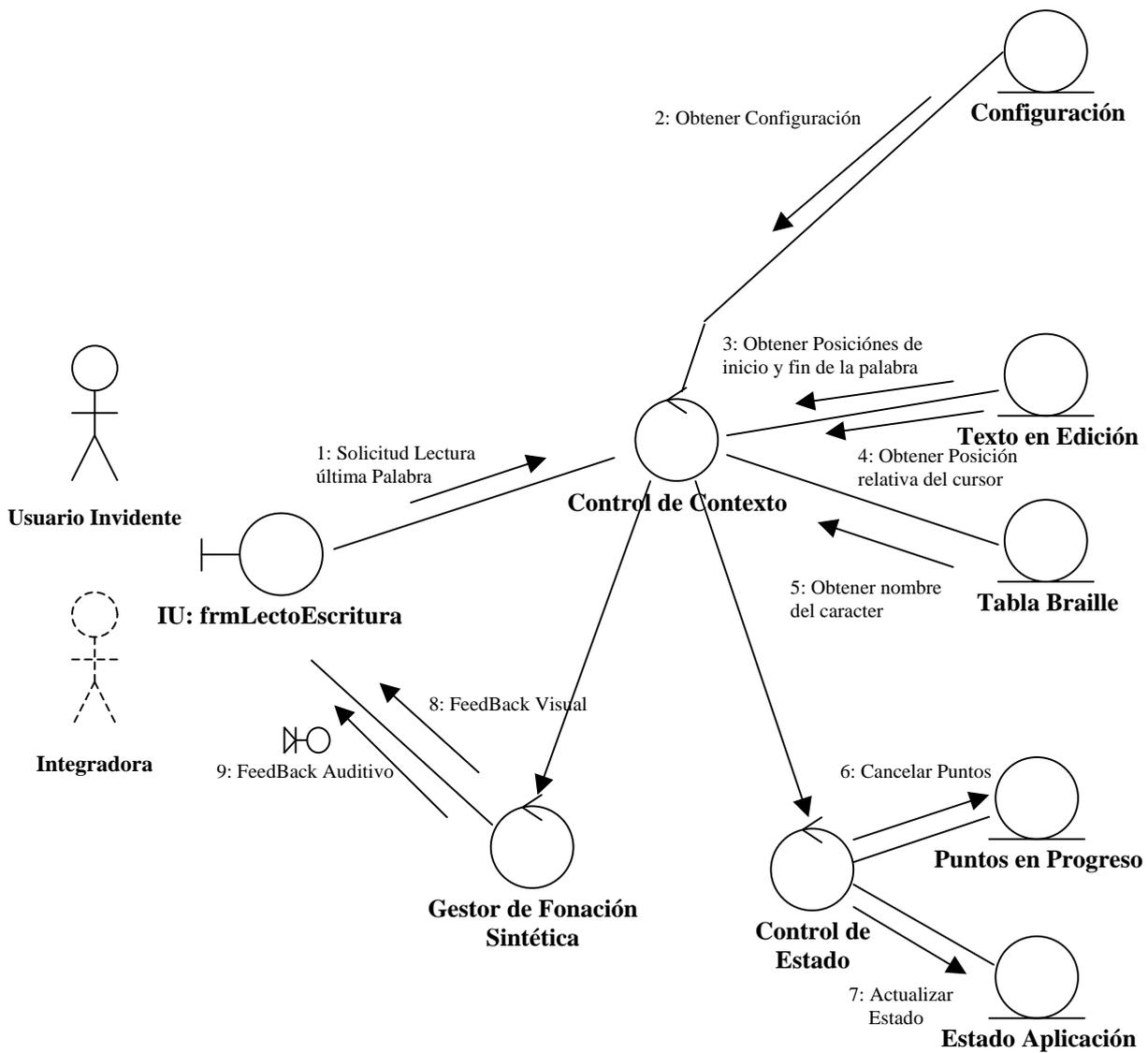
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. Si el cursor se encuentra al final del texto en edición, es decir, en la posición inmediato siguiente al último carácter que lo compone.	1.1. Se añade a la cadena de lectura la indicación expresa de "Fin del Texto" para dar al usuario información de contexto.
	1.2. Fin del caso de uso
2. Si el cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	2.1. Como no hay palabra previa a pronunciar, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.2 del caso de uso, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU: 2.2. Lectura de la última palabra*



*Nota:* Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

*Diagrama de Colaboración CU: 2.2. Lectura de la última palabra*



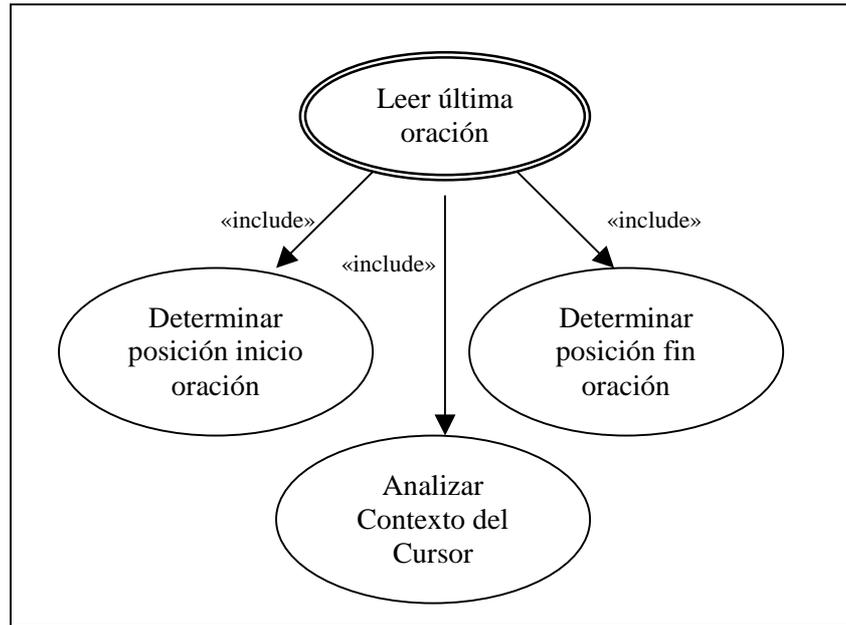
## CU: 2.3. Lectura de la última oración

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.3. Lectura de la última oración	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, la oración completa sobre la que se encuentra situada el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación oír la lectura de la oración sobre la que se encuentra situada el cursor, sea que se encuentre al principio de la misma, al medio, o al final. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son leídos expresamente. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario. Lo mismo se aplica para cuando el cursor se encuentre posicionado al final del texto.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>leer la última oración</u> . Esto se realiza seleccionando la tecla de menú de lectura (tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer última oración" (con la tecla de función #2)	<p>1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:</p> <p>a. Si el cursor se encuentra exactamente al principio de una oración, es decir, en la posición inmediatamente previa al primer carácter que la compone; el sistema determina la posición de inicio y fin de la oración que allí comienza.</p> <p>b. Si el cursor se encuentra exactamente al final de una oración, es decir, en la posición inmediatamente posterior al último carácter que la compone; el sistema determina la posición de inicio y fin de la oración que allí termina.</p> <p>c. Si el cursor se encuentra en cualquier posición intermedia de una oración, determina las posiciones del texto en que comienza y termina dicha oración, junto a la posición relativa del cursor sobre la misma.</p> <p>1.2. Se genera la verbalización (pronunciación) de la oración, determinada por la subcadena del texto comprendida entre el inicio y el fin determinado según la lógica explicada, siguiendo el siguiente esquema:</p> <p>a. Para el caso 1.1.a. se pronuncia de corrido la oración comprendida entre la posición de inicio y la posición de fin determinadas, precedida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo.</p> <p>b. Para el caso 1.1.b. se pronuncia de corrido la oración comprendida entre la posición de inicio y la posición de fin determinadas, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo.</p> <p>c. Para el caso 1.1.c. se pronuncia la parte de la oración comprendida desde el inicio de la misma hasta la posición actual del cursor, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo. Y finalmente se finaliza la verbalización con la parte restante de la oración, comprendida entre la posición del cursor y el final de la misma.</p> <p>1.3. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.</p>

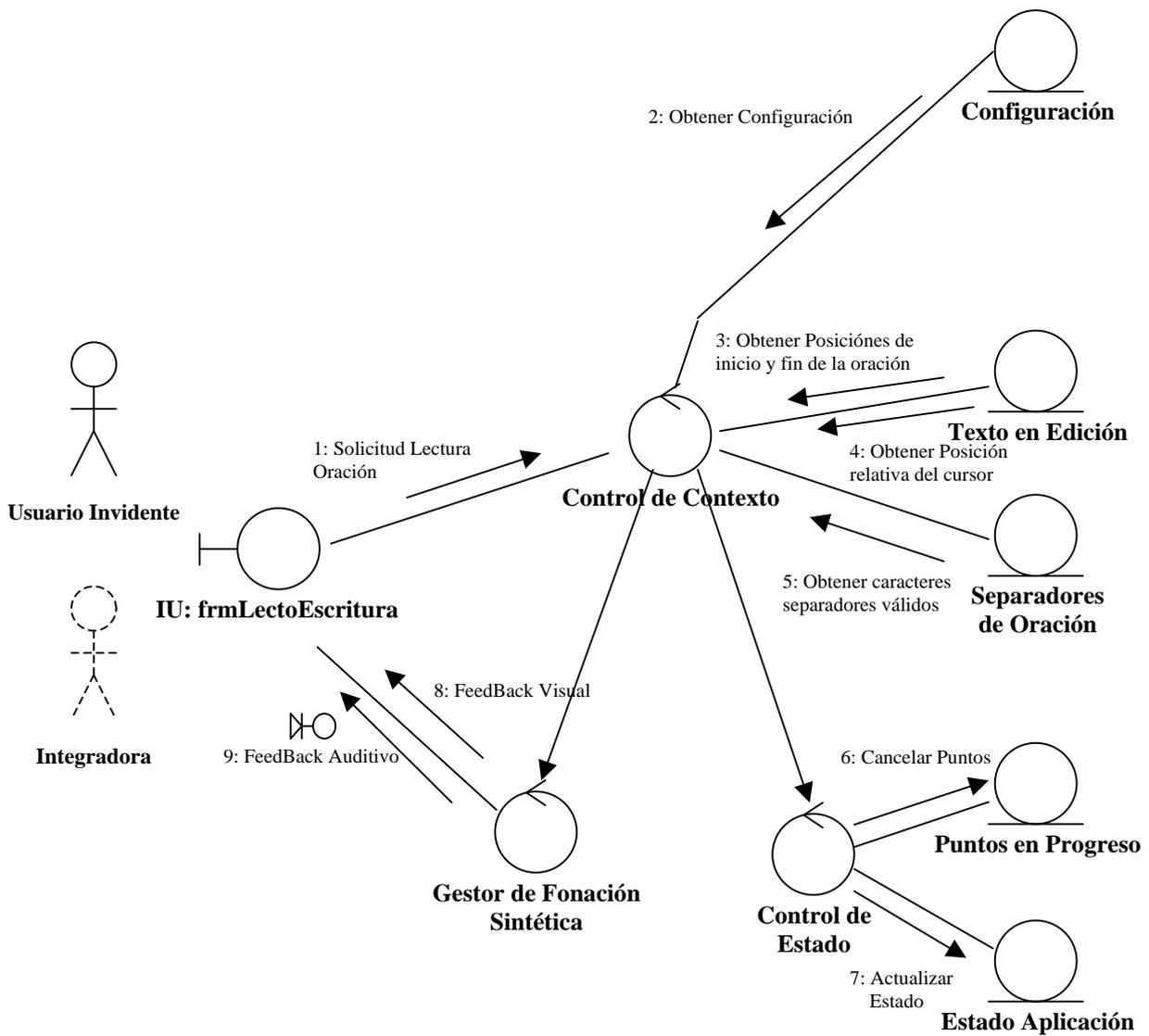
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. Si el cursor se encuentra al inicio del texto.	1.1. La cadena de lectura es reemplazada por la indicación expresa de "Inicio del Texto", seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, dando al usuario información de contexto.
2. Si el cursor se encuentra al final del texto.	2.1. Se añade a la cadena de lectura la indicación expresa de "Fin del Texto" para dar al usuario información de contexto.
3. El cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	3.1. Se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 3.2. Fin del caso de uso
4. El cursor se encontraba inmediatamente a continuación de un salto de línea: carácter de nuevo párrafo (equivalente a un ENTER en un editor de texto).	4.1. Se reproduce audiblemente la oración anterior a la posición del cursor, haciendo explícita la lectura del carácter de "nuevo párrafo", seguido de la indicación del "cursor". 4.2. Fin del caso de uso
5. El cursor se encontraba inmediatamente antes de un salto de línea: carácter de nuevo párrafo (equivalente a un ENTER en un editor de texto).	5.1. Se reproduce audiblemente la oración anterior a la posición del cursor, haciendo explícita a continuación la indicación del "cursor", continuado por la lectura del carácter de "nuevo párrafo". Esto aporta a la lectura de contexto que realiza el usuario. 5.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU: 2.3. Lectura de la última oración*



*Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.*

*Diagrama de Colaboración CU: 2.3. Lectura de la última oración*



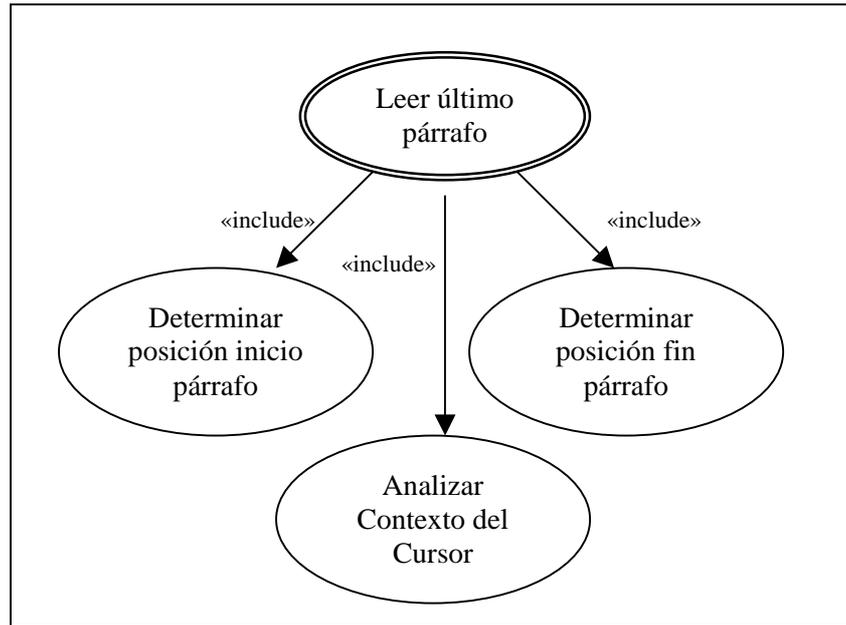
## CU: 2.4. Lectura del último párrafo

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.4. Lectura del último párrafo	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, el párrafo completo sobre el que se encuentra situado el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación oír la lectura del párrafo sobre el que se encuentra situado el cursor, sea que se encuentre al principio del mismo, al medio, o al final. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son leídos expresamente. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario. Lo mismo se aplica para cuando el cursor se encuentre posicionado al final del texto.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>leer el último párrafo</u> . Esto se realiza seleccionando la tecla de menú de lectura (tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer último párrafo" (con la tecla de función #3)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica: <ul style="list-style-type: none"> <li>a. Si el cursor se encuentra exactamente al principio de un párrafo, es decir, en la posición inmediatamente previa al primer carácter que lo compone; el sistema determina la posición de inicio y fin del párrafo que allí comienza.</li> <li>b. Si el cursor se encuentra exactamente al final de un párrafo, es decir, en la posición inmediatamente posterior al último carácter que lo compone; el sistema determina la posición de inicio y fin del párrafo que allí termina.</li> <li>c. Si el cursor se encuentra en cualquier posición intermedia de un párrafo, determina las posiciones del texto en que comienza y termina dicho párrafo, junto a la posición relativa del cursor sobre el mismo.</li> </ul>
	1.2. Se genera la verbalización (pronunciación) del párrafo, determinada por la subcadena del texto comprendida entre el inicio y el fin determinado según la lógica explicada, siguiendo el siguiente esquema: <ul style="list-style-type: none"> <li>a. Para el caso 1.1.a. se pronuncia de corrido el párrafo comprendido entre la posición de inicio y la posición de fin determinadas, precedido de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo.</li> <li>b. Para el caso 1.1.b. se pronuncia de corrido el párrafo comprendido entre la posición de inicio y la posición de fin determinadas, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo.</li> <li>c. Para el caso 1.1.c. se pronuncia la parte del párrafo comprendida desde el inicio de la misma hasta la posición actual del cursor, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo. Y finalmente se finaliza la verbalización con la parte restante del párrafo, comprendida entre la posición del cursor y el final del mismo.</li> </ul>
	1.3. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.

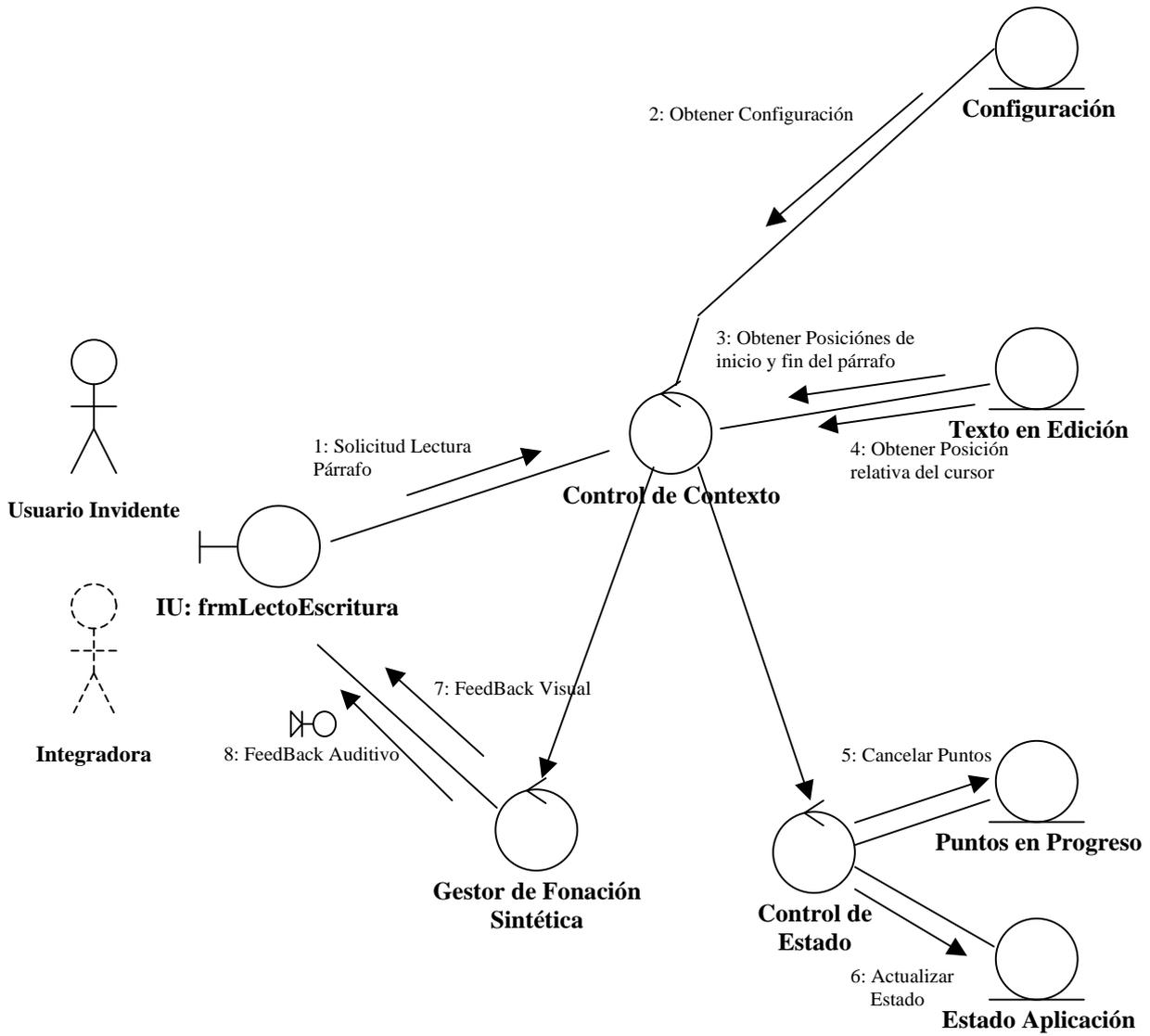
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. Si el cursor se encuentra al inicio del texto.	1.1. La cadena de lectura es reemplazada por la indicación expresa de "Inicio del Texto", seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, dando al usuario información de contexto.
2. Si el cursor se encuentra al final del texto.	2.1. Se añade a la cadena de lectura la indicación expresa de "Fin del Texto" para dar al usuario información de contexto.
3. El cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	3.1. Se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 3.2. Fin del caso de uso
4. El cursor se encontraba inmediatamente a continuación de un salto de línea: carácter de nuevo párrafo (equivalente a un ENTER en un editor de texto).	4.1. Se reproduce audiblemente el párrafo anterior a la posición del cursor, haciendo explícita la lectura del carácter de "nuevo párrafo", seguido de la indicación del "cursor". 4.2. Fin del caso de uso
5. El cursor se encontraba inmediatamente antes de un salto de línea: carácter de nuevo párrafo (equivalente a un ENTER en un editor de texto).	5.1. Se reproduce audiblemente el párrafo anterior a la posición del cursor, haciendo explícita a continuación la indicación del "cursor", continuado por la lectura del carácter de "nuevo párrafo". Esto aporta a la lectura de contexto que realiza el usuario. 5.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU: 2.4. Lectura del último párrafo*



*Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.*

*Diagrama de Colaboración CU: 2.4. Lectura del último párrafo*



## CU: 2.5. Lectura desde la última marca de bloque

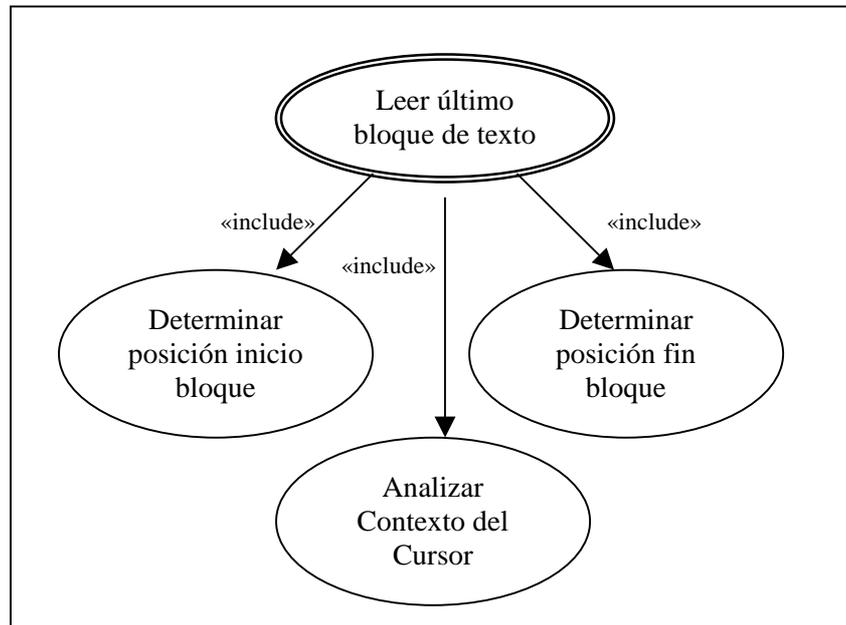
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.5. Lectura desde la última marca de bloque	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, el bloque de texto completo sobre el que se encuentra situado el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación oír la lectura del bloque de texto sobre el que se encuentra situado el cursor, sea que se encuentre al principio del mismo, al medio, o al final. Los marcadores de bloque (carácter braille especial específicamente diseñado para la aplicación) permiten establecer conjuntos de párrafos formados por el texto que se encuentra entre dos marcas. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario. Lo mismo se aplica para cuando el cursor se encuentre posicionado al final del texto.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>leer desde la última marca de bloque</u> . Esto se realiza seleccionando la tecla de menú de lectura (tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer desde la marca anterior" (con la tecla de función #6)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra en medio de un área de texto comprendida entre dos marcadores de bloque, el sistema determina la posición de inicio y fin de dicho bloque de texto.  b. Si el cursor se encuentra posicionado entre el principio del texto y una marca de bloque, el sistema determina como posición inicial del bloque a leer el inicio mismo del texto, y como posición final de dicho bloque la ubicación en que se encuentra la marca referida.  c. Si el cursor se encuentra posicionado entre una marca de bloque y el fin del texto, el sistema determina como posición inicial del bloque a leer la ubicación en que se encuentra la marca referida, y como posición final la del fin del texto.
	1.2. Se genera la verbalización (pronunciación) del bloque de texto, determinado por la subcadena del texto comprendida entre el inicio y el fin determinado según la lógica explicada. Durante la operación de lectura en curso, y al igual que sucede con todas las operaciones de lectura implementadas en el texto, la lectura explícita de los nombres de los signos de puntuación, y su tono y velocidad de pronunciación diferenciados, dependerán de la configuración de la aplicación.  La posición relativa del cursor en el texto será también indicada al usuario de manera explícita, según la configuración de la aplicación.
	1.3. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	

<b>Caso de uso: 2.5. Lectura desde la última marca de bloque</b> <span style="float: right;">(Cont...)</span>	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. Si el cursor se encuentra al inicio del texto.	1.1. La cadena de lectura es reemplazada por la indicación expresa de "Inicio del Texto", seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, dando al usuario información de contexto.
2. Si el cursor se encuentra al final del texto.	2.1. Se añade a la cadena de lectura la indicación expresa de "Fin del Texto" para dar al usuario información de contexto.
3. El cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	3.1. Se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 3.2. Fin del caso de uso
4. El cursor se posiciona encima de un marcador de bloque. Considerar que su instrumentación visible y lógica consiste en tres caracteres consecutivos: "<" + "*" + ">" normalmente precedidos y sucedidos por saltos de línea (CR + LF ).	4.1. El cursor es desplazado de manera automática a la primera posición inmediatamente consecutiva a la marca de bloque, evitando de este modo que se pueda iniciar un proceso de lectura con el cursor mal posicionado. 4.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

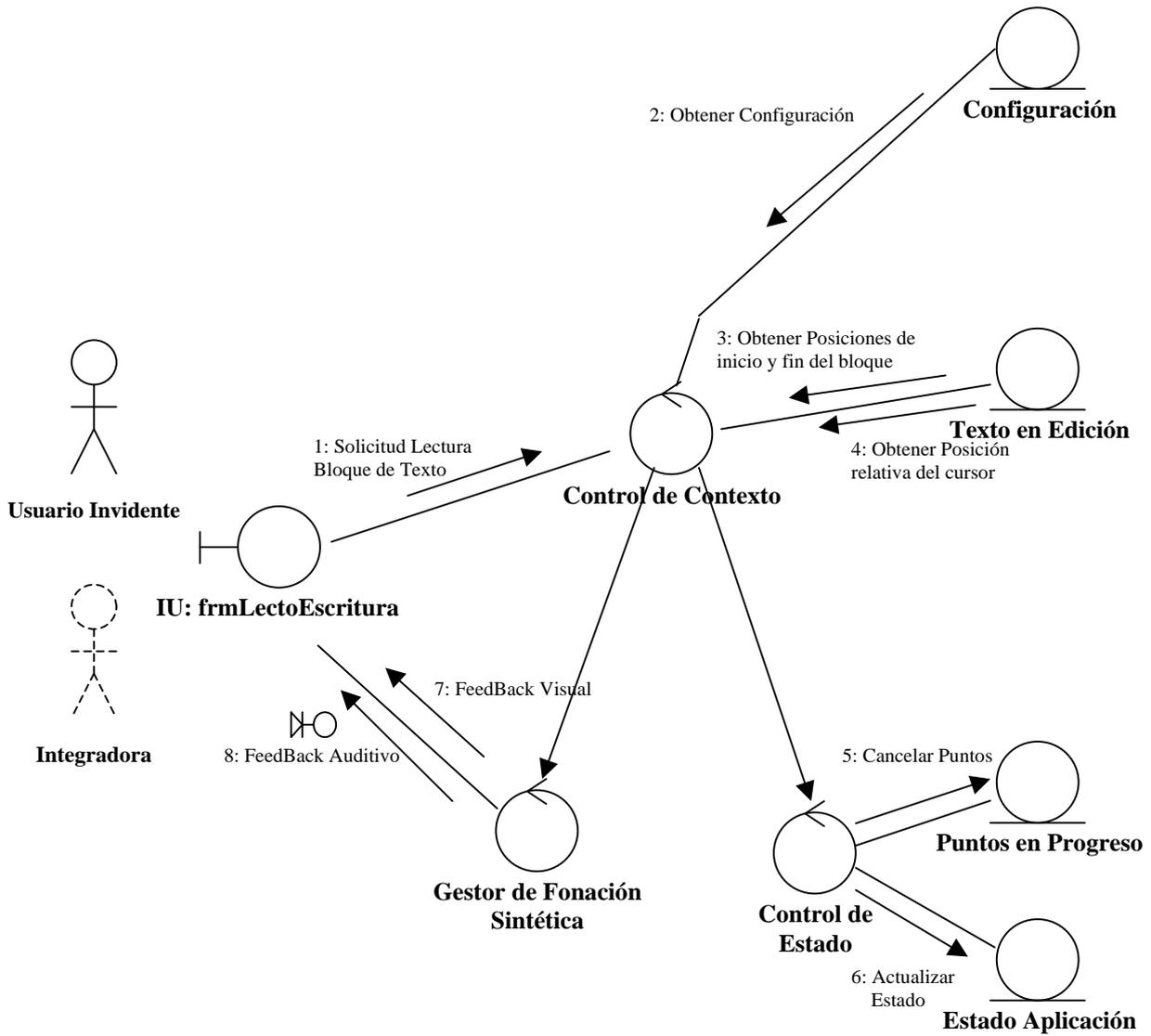
2.5. Lectura desde la última marca de bloque



*Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.*

Diagrama de Colaboración CU:

2.5. Lectura desde la última marca de bloque

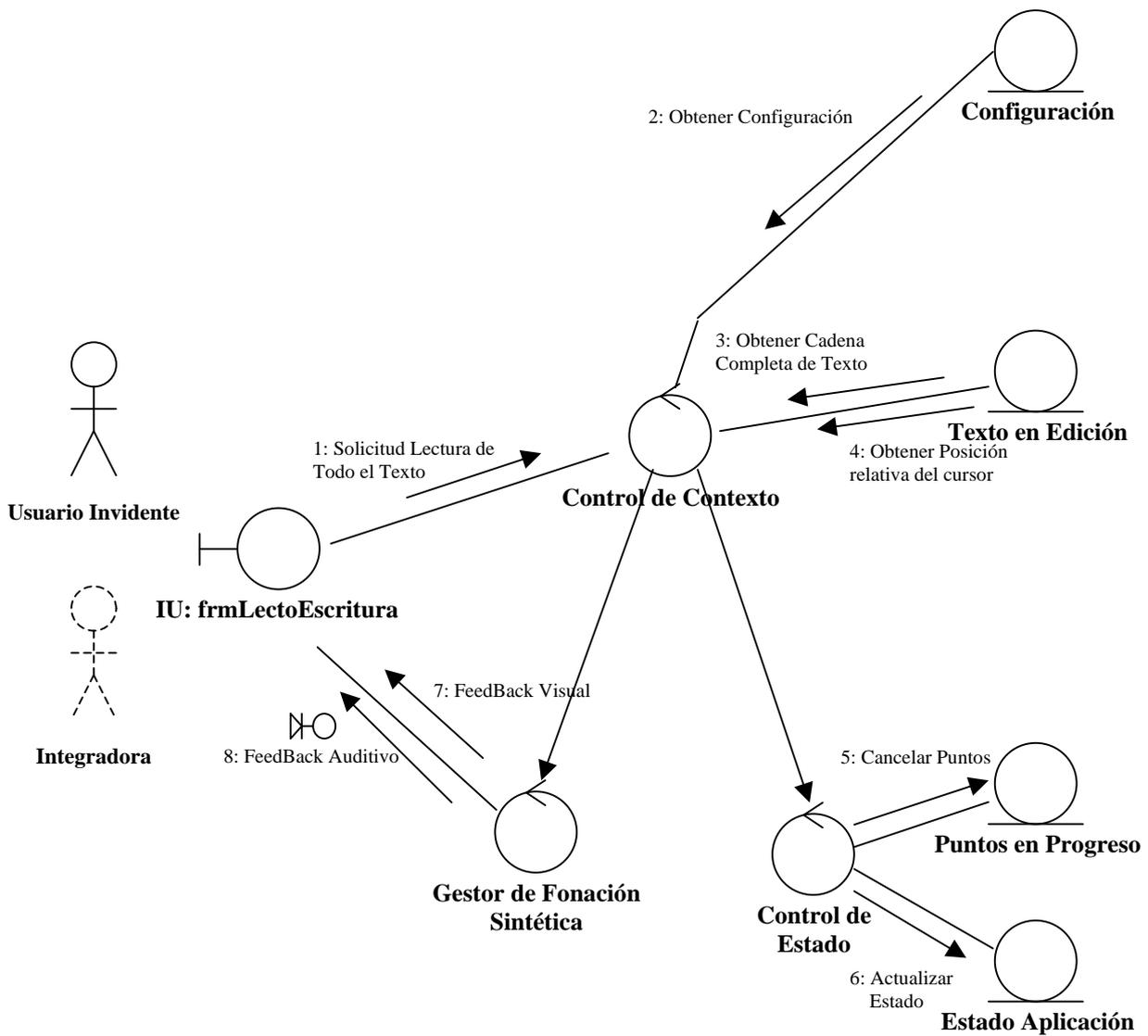


## CU: 2.6. Lectura de todo el texto

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 2.6. Lectura de todo el texto	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente, por fonación sintética, el texto completo en edición.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación oír la lectura del texto completo que se encuentra actualmente en edición.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de leer desde la última marca de bloque. Esto se realiza seleccionando la tecla de menú de lectura (tecla de función #6), y en el menú emergente, eligiendo la opción de "Leer todo el texto" (con la tecla de función #4)	<p>1.1. Se genera la verbalización (pronunciación) de la totalidad del texto en edición.</p> <p>Los signos de puntuación serán leídos explícitamente, reproduciendo su nombre con timbre y velocidad diferenciados y en función de las características de configuración.</p> <p>La posición relativa del cursor en el texto será también indicada al usuario de manera explícita, según la configuración de la aplicación.</p> <p>1.2. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.</p> <p>1.3. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.</p> <p>1.4. Fin del caso de uso.</p>
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. Si el texto en edición está vacío (compuesto por ningún carácter)	1.1. La cadena de lectura es reemplazada por la indicación expresa de "Fin del Texto", para indicar la posición del mismo, dando al usuario información de contexto.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	

Diagrama de Colaboración CU: 2.6. Lectura de todo el texto



### Explosión Caso de Uso 3. Intervención Externa en el Texto

Este caso de uso no sufre posteriores explosiones, pues sea cual fuere la intención de la intervención externa en la edición del texto por parte de la asistente pedagógica del usuario, su acción es siempre de una misma naturaleza y características de uso.

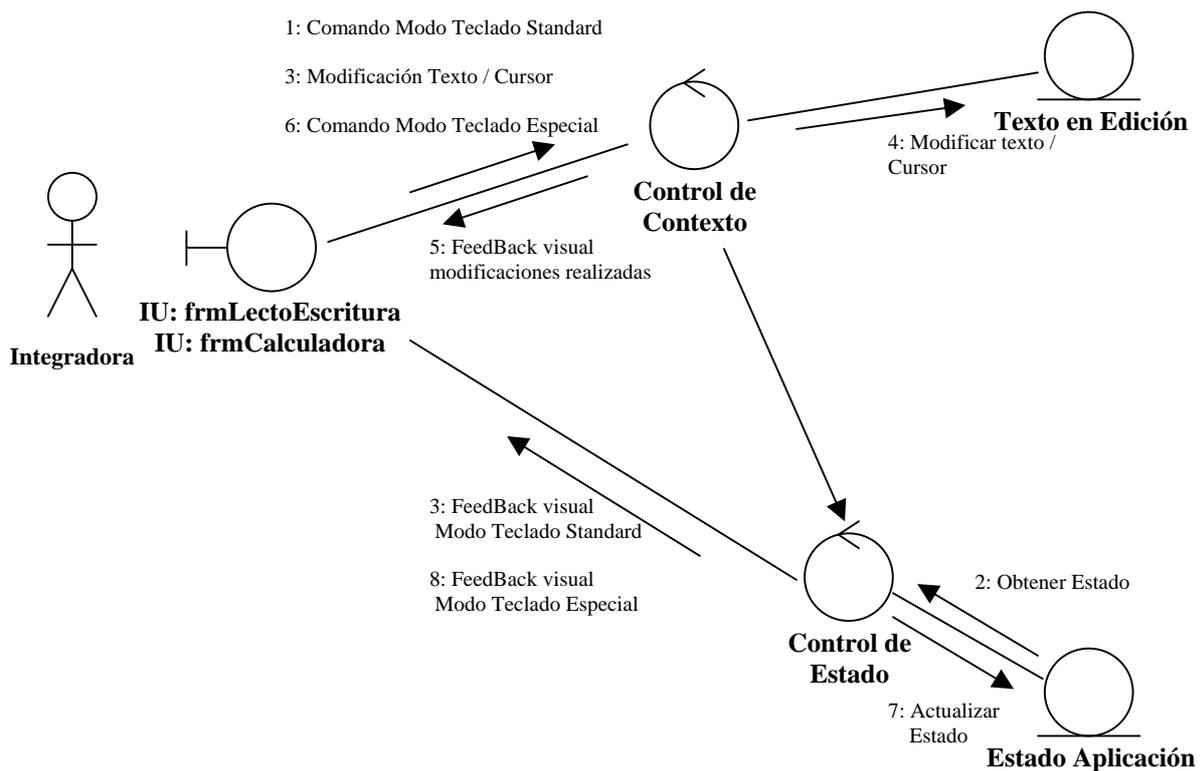
Se detallan a continuación los modelos que definen este caso de uso.

#### Ficha de Caso de Uso

<b>Caso de uso:</b> 3. <i>Intervención Externa en el Texto</i>	
<b>Actor Principal:</b> <i>Asistente Pedagógica Integradora</i>	
<b>Actor Secundario:</b> «ninguno»	
<b>Objetivos / propósito:</b> <i>Intervenir externamente en el texto en edición, sea modificando su contenido y/o la posición del cursor en el mismo.</i>	
<b>Resumen:</b> <i>Este Caso de Uso le permite a la asistente pedagógica, integradora, o a la persona que esté asistiendo al usuario principal de la aplicación en su uso, realizar en el texto modificaciones, tanto en su contenido como en la posición relativa del cursor sobre el mismo. De este modo, y mediante la interacción externa con el usuario principal del sistema, se hacen posibles correcciones y otras actividades de naturaleza pedagógica, tendientes a la asistencia y/o corrección hacia el usuario principal.</i> <i>Para que esta intervención sea realizada, el usuario externo hace uso de un comando de teclado que inactiva temporalmente el uso del tablero especial, abriendo la edición del texto al teclado convencional de la computadora. Y una vez finalizada la modificación deseada, vuelve a invertirse el dispositivo de entrada de la aplicación, anulando el teclado y reactivando el uso del tablero especial.</i>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario externo de la aplicación indica al sistema el comando de activación del modo de "Teclado Standard". Esto se realiza mediante la pulsión de la combinación de teclas definida para tal fin: <b>Ctrl+F1</b>	1.1. Se desactiva la recepción de señales desde el dispositivo especial de acceso a la aplicación (el tablero especial), y se activa la normal interpretación de las teclas del teclado Standard de la computadora. 1.2. La Interfaz gráfica (tanto la pantalla de edición de texto como la calculadora) reflejan el cambio de estado en el indicador de "Modo de Teclado", presentando en color rojo la leyenda específica de "Teclado Standard"
2. El usuario externo realiza las modificaciones que desee sobre el texto en edición y la posición relativa del cursor en el mismo.	2.1. El sistema responde a los comandos de teclado y mouse del usuario del modo convencional para la edición de contenido de la caja de texto multilínea que contiene el texto en edición.
3. El usuario externo de la aplicación indica al sistema el comando de regreso al modo de "Teclado Especial". Esto se realiza mediante la pulsión de la combinación de teclas definida para tal fin: <b>Ctrl+F1</b> (estas teclas funcionan como un switch entre ambos modos)	3.1. Se desactiva la recepción de señales desde el teclado Standard de la PC, y se reactiva el uso del dispositivo especial de acceso a la aplicación (el tablero especial). 3.2. La Interfaz gráfica (tanto la pantalla de edición de texto como la calculadora) reflejan el cambio de estado en el indicador de "Modo de Teclado", presentando en color azul la leyenda específica de "Teclado Lucas!"
4. Fin del caso de uso.	

<b>Caso de uso:</b> 3. <i>Intervención Externa en el Texto</i> (Cont...)	
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El texto en edición y la posición relativa del cursor en el mismo se modificaron. El control de la aplicación vuelve a estar en su estado original, a espera del ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. Si el usuario intenta posicionar el cursor en medio de una marca de bloque "<*>".	1.1. La posición del cursor será automáticamente modificada para situarse al final de dicha marca, inmediatamente a continuación del carácter ">". Esto tiene por objeto que no pueda iniciarse ninguna operación de edición con el cursor situado en medio de una marca, lo que afectaría la legibilidad de la misma por parte del análisis de contexto del cursor.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i> 2. <i>Formulario de Calculadora (alternativamente)</i>	

Diagrama de Colaboración CU: 3. Intervención Externa en el Texto



### Explosión Caso de Uso 4. Acceso al Menú de la Aplicación

Al igual que el caso anterior, este caso de uso no sufre posteriores explosiones. Dada la naturaleza jerárquica del menú de la aplicación oportunamente detallado, ambos usuarios – el usuario principal y su asistente pedagógica – hacen uso del menú siguiendo mecanismos de interacción diferentes, pero exponiéndose a una misma e idéntica funcionalidad.

El uso del menú sirve de disparador de otros casos de uso, a los cuales no incluye en su funcionalidad, sino que la misma llega precisamente hasta la activación del que corresponda, según se detalla en los respectivos escenarios.

Se detallan a continuación los modelos que definen este caso de uso.

#### Ficha de Caso de Uso

<b>Caso de uso:</b> 4. Acceso al menú de la aplicación
<b>Actor Principal:</b> Usuario No Vidente
<b>Actor Secundario:</b> Asistente Pedagógica Integradora
<b>Objetivos / propósito:</b> Permitir la navegación del sistema de menú jerárquico en que están organizados todos los comandos de la aplicación, y la activación del comando deseado por el usuario.
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal, y eventualmente a la integradora – sólo con fines de entrenamiento para el usuario principal – navegar las diferentes categorías de comandos para la aplicación disponibles en el sistema de menú jerárquico descrito precedentemente. La implementación del menú se realiza mediante la redefinición de la funcionalidad de cada uno de los ocho botones de comando (llamados "Función #1" a "Función #8") para cada menú o submenú accedido. Una vez accedido al conjunto de botones que disparan acciones, al pulsar sobre el botón elegido se dispara el caso de uso que corresponda a su funcionalidad. Un caso de excepción es la funcionalidad otorgada al botón de Función #8, cuya funcionalidad es siempre la misma, en cualquier contexto: la lectura fonética de las opciones disponibles en el menú actual. Una vez accedida a una función específica – implementación de un caso de uso de la aplicación – el estado del menú vuelve a su configuración original: <b>el menú principal</b> .  La estructura de menú de la aplicación y los casos de uso disparados desde cada una de sus funciones se presentan en el siguiente esquema:
<ul style="list-style-type: none"><li>➤ <u>Menú Principal</u><ul style="list-style-type: none"><li>➤ Función#1: Dispara <u>Menú de Control</u></li><li>➤ Función#2: Dispara <u>Menú de Borrar</u></li><li>➤ Función#3: Dispara <u>Menú para Moverse en Retroceso</u></li><li>➤ Función#4: Dispara <u>Menú de Gestión de Archivos</u></li><li>➤ Función#5: Dispara C.U. 2.2. Lectura de la última palabra</li><li>➤ Función#6: Dispara <u>Menú de Lectura</u></li><li>➤ Función#7: Dispara <u>Menú para Moverse en Avance</u></li><li>➤ Función#8: Reproduce lectura de menú actual</li></ul></li></ul>

- Menú de Control
  - ❖ *Función#1: Dispara Menú para Elegir la Voz Actual*
  - ❖ *Función#2: < Sin funcionalidad asignada >*
  - ❖ *Función#3: < Sin funcionalidad asignada >*
  - ❖ *Función#4: < Sin funcionalidad asignada >*
  - ❖ *Función#5: Dispara Aplicativo Calculadora*
  - ❖ *Función#6: < Sin funcionalidad asignada >*
  - ❖ *Función#7: < Sin funcionalidad asignada >*
  - ❖ *Función#8: Reproduce lectura de menú actual*
  
  - ❖ Menú para Elegir la Voz Actual
    - *Función#1: Dispara C.U. 8.1. Selecc. Voz Predeterminada (1)*
    - *Función#2: Dispara C.U. 8.1. Selecc. Voz Predeterminada (2)*
    - *Función#3: Dispara C.U. 8.1. Selecc. Voz Predeterminada (3)*
    - *Función#4: Dispara C.U. 8.1. Selecc. Voz Predeterminada (4)*
    - *Función#5: Dispara C.U. 8.1. Selecc. Voz Predeterminada (5)*
    - *Función#6: Dispara C.U. 8.1. Selecc. Voz Predeterminada (6)*
    - *Función#7: Dispara C.U. 8.1. Selecc. Voz Predeterminada (7)*
    - *Función#8: Reproduce lectura de menú actual*
  
- Menú de Borrar
  - ❖ *Función#1: Dispara C.U. 5.1.2. Eliminar última palabra*
  - ❖ *Función#2: Dispara C.U. 5.1.3. Eliminar última oración*
  - ❖ *Función#3: Dispara C.U. 5.1.4. Eliminar último párrafo*
  - ❖ *Función#4: Dispara C.U. 5.1.5. Eliminar todo el texto*
  - ❖ *Función#5: < Sin funcionalidad asignada >*
  - ❖ *Función#6: < Sin funcionalidad asignada >*
  - ❖ *Función#7: < Sin funcionalidad asignada >*
  - ❖ *Función#8: Reproduce lectura de menú actual*
  
- Menú para Moverse en Retroceso
  - ❖ *Función#1: Dispara C.U. 5.2.2.  
Despl. Cursor al Comienzo de la Palabra Actual*
  - ❖ *Función#2: Dispara C.U. 5.2.3.  
Despl. Cursor al Comienzo de la Oración Actual*
  - ❖ *Función#3: Dispara C.U. 5.2.4.  
Despl. Cursor al Comienzo del Párrafo*
  - ❖ *Función#4: Dispara C.U. 5.2.6.  
Despl. Cursor hasta el Inicio del Texto*
  - ❖ *Función#5: Dispara C.U. 5.2.1.  
Despl. Cursor un Carácter hacia Atrás*
  - ❖ *Función#6: Dispara C.U. 5.2.5.  
Despl. Cursor hasta el Marcador Anterior*
  - ❖ *Función#7: < Sin funcionalidad asignada >*
  - ❖ *Función#8: Reproduce lectura de menú actual*
  
- Menú de Gestión de Archivos
  - ❖ *Función#1: Dispara C.U. 7.1. Abrir Archivo*
  - ❖ *Función#2: Dispara C.U. 7.2. Guardar Archivo*
  - ❖ *Función#3: < Sin funcionalidad asignada >*
  - ❖ *Función#4: < Sin funcionalidad asignada >*
  - ❖ *Función#5: < Sin funcionalidad asignada >*
  - ❖ *Función#6: < Sin funcionalidad asignada >*
  - ❖ *Función#7: < Sin funcionalidad asignada >*
  - ❖ *Función#8: Reproduce lectura de menú actual*
  
- Menú de Lectura
  - ❖ *Función#1: Dispara C.U. 2.2. Lectura de la última palabra*
  - ❖ *Función#2: Dispara C.U. 2.3. Lectura de la última oración*
  - ❖ *Función#3: Dispara C.U. 2.4. Lectura del último párrafo*
  - ❖ *Función#4: Dispara C.U. 2.6. Lectura de todo el texto*
  - ❖ *Función#5: Dispara C.U. 2.1. Lectura del último carácter ingresado*
  - ❖ *Función#6: Dispara C.U. 2.5. Lectura desde la última marca*
  - ❖ *Función#7: < Sin funcionalidad asignada >*
  - ❖ *Función#8: Reproduce lectura de menú actual*

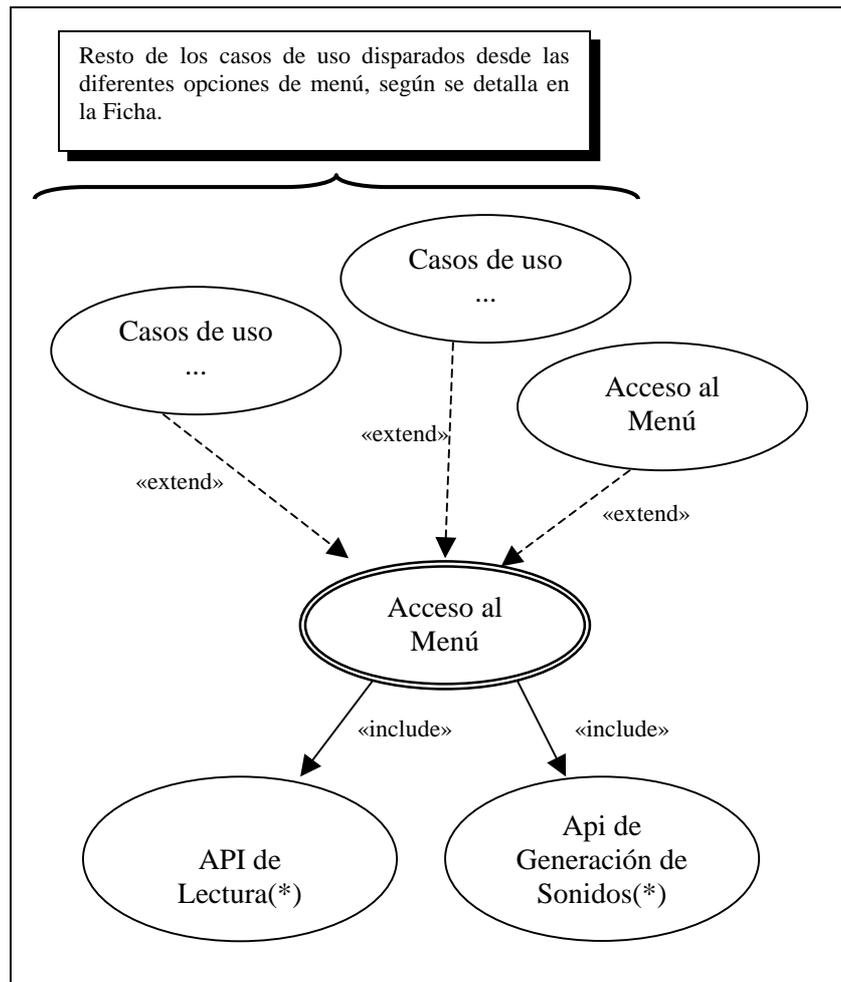
<ul style="list-style-type: none"> <li>➤ <u>Menú para Moverse en Avance</u> <ul style="list-style-type: none"> <li>❖ <i>Función#1: Dispara</i> C.U. 5.3.2. <i>Despl. Cursor al Comienzo de la Sgte. Palabra</i></li> <li>❖ <i>Función#2: Dispara</i> C.U. 5.3.3. <i>Despl. Cursor al Comienzo de la Sgte. Oración</i></li> <li>❖ <i>Función#3: Dispara</i> C.U. 5.3.4. <i>Despl. Cursor al Comienzo del Sgte. Párrafo</i></li> <li>❖ <i>Función#4: Dispara</i> C.U. 5.3.6. <i>Despl. Cursor hasta el Final del Texto</i></li> <li>❖ <i>Función#5: Dispara</i> C.U. 5.3.1. <i>Despl. Cursor un Carácter hacia Adelante</i></li> <li>❖ <i>Función#6: Dispara</i> C.U. 5.3.5. <i>Despl. Cursor hasta el Marcador Siguiete</i></li> <li>❖ <i>Función#7: &lt; Sin funcionalidad asignada &gt;</i></li> <li>❖ <i>Función#8: Reproduce lectura de menú actual</i></li> </ul> </li> </ul>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra en estado de edición de texto, esperando puntos braille, o está dentro de algún menú (fue presionada una tecla de menú que disparó otro menú).	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario Pulsa una de las teclas de función definidas para el menú en curso.	1.1. Si la tecla corresponde a una funcionalidad: se extiende al caso de uso que la implementa.
	1.2. Si la tecla corresponde a otro menú: se extiende a este mismo caso de uso para el menú en cuestión, con lo que se presenta – y según la configuración, se oyen las opciones disponibles – el nuevo menú.
2. Fin del caso de uso.	
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El nuevo menú correspondiente a la opción elegida es presentado y queda en estado de disponible para que se acceda a sus opciones; o en caso de haberse ejecutado un comando, el menú se encuentra restaurado a su configuración por defecto (menú principal)	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario pulsa una tecla de función que no tiene definida ninguna funcionalidad para el menú en curso.	1.1. Se produce un Beep audible que indica la cancelación.
	1.2. Se vuelve el menú a su estado normal (por defalut), volviendo el foco al texto en edición.
	1.3. Fin del caso de uso.
2. El usuario pulsa la tecla de cancelación, sobre la mitad izquierda del tablero.	2.1. Se produce un Beep audible que indica la cancelación.
	2.2. Se vuelve el menú a su estado normal (por defalut), volviendo el foco al texto en edición.
	2.3. Fin del caso de uso.
3. El usuario oprime cualquier otra tecla del tablero	3.1. Se produce un Beep audible que indica la cancelación.
	3.2. Se vuelve el menú a su estado normal (por defalut), volviendo el foco al texto en edición.
	3.3. Fin del caso de uso.
4. El usuario secundario (integrador) pulsa la tecla Esc sobre el teclado	4.1. Se produce un Beep audible que indica la cancelación.
	4.2. Se vuelve el menú a su estado normal (por defalut), volviendo el foco al texto en edición.
	4.3. Fin del caso de uso.
<b>Casos de uso utilizados:</b> «extend a sí mismo» C.U. 4. Acceso al menú de la aplicación. «extend» C.U. 2.2. Lectura de la última palabra «extend» C.U. 8.1. Selecc. Voz Predeterminada «extend» C.U. 5.1.2. Eliminar última palabra «extend» C.U. 5.1.3. Eliminar última oración «extend» C.U. 5.1.4. Eliminar último párrafo	

«extend» C.U. 5.1.5. Eliminar todo el texto  
 «extend» C.U. 5.2.2. Despl. Cursor al Comienzo de la Palabra Actual  
 «extend» C.U. 5.2.3. Despl. Cursor al Comienzo de la Oración Actual  
 «extend» C.U. 5.2.4. Despl. Cursor al Comienzo del Párrafo  
 «extend» C.U. 5.2.6. Despl. Cursor hasta el Inicio del Texto  
 «extend» C.U. 5.2.1. Despl. Cursor un Carácter hacia Atrás  
 «extend» C.U. 5.2.5. Despl. Cursor hasta el Marcador Anterior  
 «extend» C.U. 7.1. Abrir Archivo  
 «extend» C.U. 7.2. Guardar Archivo  
 «extend» C.U. 2.2. Lectura de la última palabra  
 «extend» C.U. 2.3. Lectura de la última oración  
 «extend» C.U. 2.4. Lectura del último párrafo  
 «extend» C.U. 2.6. Lectura de todo el texto  
 «extend» C.U. 2.1. Lectura del último carácter ingresado  
 «extend» C.U. 2.5. Lectura desde la última marca  
 «extend» C.U. 5.3.2. Despl. Cursor al Comienzo de la Sgte. Palabra  
 «extend» C.U. 5.3.3. Despl. Cursor al Comienzo de la Sgte. Oración  
 «extend» C.U. 5.3.4. Despl. Cursor al Comienzo del Sgte. Párrafo  
 «extend» C.U. 5.3.6. Despl. Cursor hasta el Final del Texto  
 «extend» C.U. 5.3.1. Despl. Cursor un Carácter hacia Adelante  
 «extend» C.U. 5.3.5. Despl. Cursor hasta el Marcador Siguiente

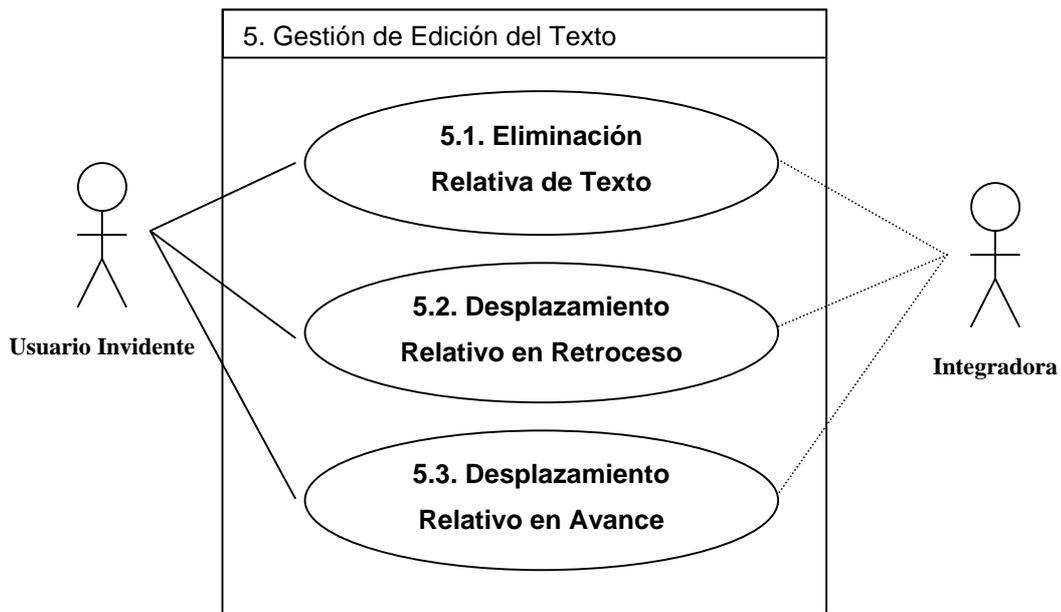
**Interfaces requeridas:**

1. Formulario principal de Lecto-Escritura

*Diagrama de Relaciones de CU: 4. Acceso al Menú de la Aplicación*



### **Explosión Caso de Uso 5. Gestión de Edición del Texto**

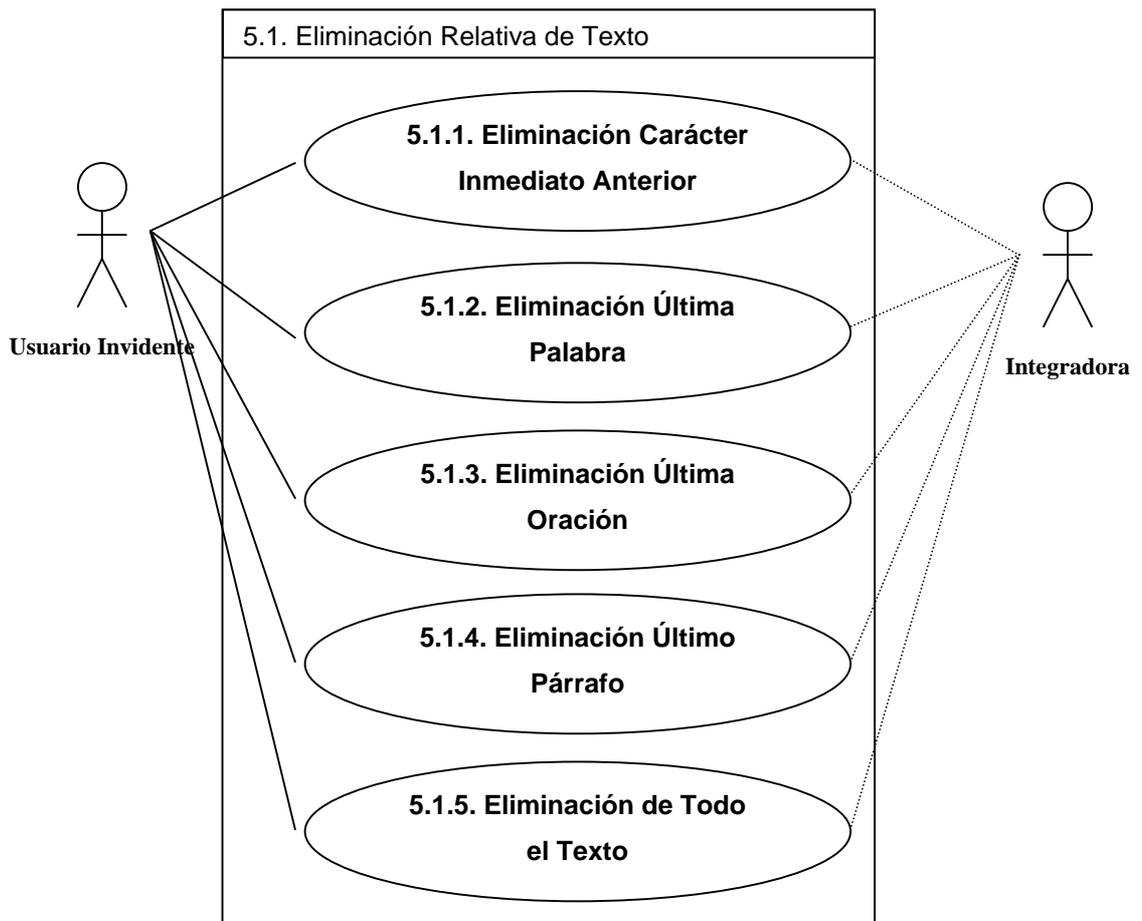


En lo referente a las funcionalidades de edición de texto a proveer por la aplicación, las mismas se dividen funcionalmente en tres grupos:

- Operaciones de eliminación relativa ( o borrado parcial ) del texto, en función de la posición del cursor en el mismo.
- Operaciones de desplazamiento del cursor o punto relativo de ubicación en el texto hacia atrás ( en dirección al principio del texto)
- Operaciones de desplazamiento del cursor o punto relativo de ubicación en el texto hacia adelante ( en dirección al final del texto)

Estas herramientas, consideradas como categorías funcionales, se explotan respectivamente en cada una de las funciones específicas provistas por la aplicación en los siguientes modelos.

**Explosión Caso de Uso 5.1. Eliminación Relativa de Texto**



Estas funciones tienen como usuario principal al niño no vidente, pero admiten también su uso por parte de la integradora, sobre todo con fines de enseñanza de uso de la herramienta y/o correcciones de procedimiento.

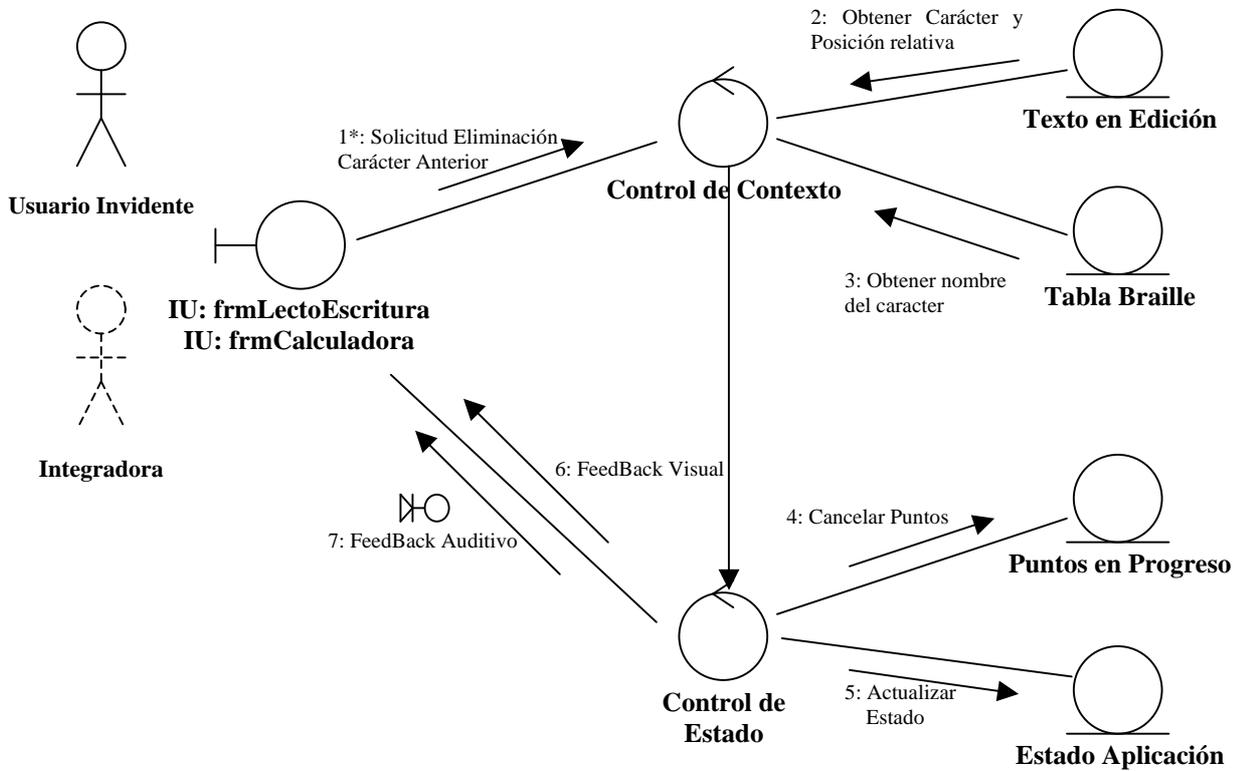
## CU: 5.1.1. Eliminación Carácter Inmediato Anterior

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.1.1. Eliminación Carácter Inmediato Anterior	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Suprimir (eliminar) el carácter ubicado inmediatamente a la izquierda del cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación eliminar el carácter que se encuentra inmediatamente a la izquierda de la posición actual del cursor, ofreciendo al mismo en forma de feedback auditivo la confirmación de dicha acción. Asimismo, si el cursor se encuentra ubicado al principio del texto, no elimina ningún carácter, sino que le indica la posición relativa de manera verbal.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
El usuario indica al sistema el comando de eliminar el carácter anterior al cursor. Esto se realiza del siguiente modo: Estando en estado de edición, sin ningún punto braille ingresado, pulsa la tecla de cancelación, con lo que se produce la extensión referida en el caso de uso 1.3. Cancelación del Carácter Ingresado	1.1. El sistema analiza la posición del cursor en el texto, y su contexto.
	1.2. Se suprime de la cadena en edición el carácter de la posición inmediato anterior a la ubicación relativa del cursor.
	1.3. Se produce la verbalización de la acción de eliminación producida, que es entregada como feedback al usuario, además de ser presentada en la interfaz gráfica al usuario secundario. Esta verbalización es de la forma "Borrar X", donde X es el nombre del carácter eliminado.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición no se modifica. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El cursor se encontraba al principio del texto, es decir, justo después de ningún carácter. Esto también aplica para cuando el texto en edición no contenga ningún carácter (texto en blanco).	1.1. Como no hay carácter previo a eliminar, simplemente se produce un Beep audible para el usuario, indicativo genérico en el contexto de todo el sistema de condición o comando erróneo.
	1.2. Fin del caso de uso
2. El cursor se encontraba justo después del un carácter compuesto (representado por más de un código ASCII).	2.1. El sistema identifica el carácter compuesto. Los casos posibles son el salto de línea o "nuevo párrafo" (CR + LF), o el "marcador de bloque" (<*>).
	2.2. Se realiza la eliminación de todos los componentes ASCII de dicho carácter de manera simultánea.
	2.3. Se verbaliza el mismo mensaje normal de "Borrar X", donde X es "Nuevo Párrafo" o "Marca de Bloque".
	2.4. El cursor queda situado en la posición inmediato anterior al primer componente ASCII del carácter compuesto eliminado.
	2.5. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno» Este caso de uso no utiliza otros, pero su contexto de ejecución es utilizado (mediante relación de «extend») desde el caso de uso <b>1.3. Cancelación del Carácter Ingresado</b> en las condiciones allí descritas.	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	
2. Formulario de Calculadora (alternativamente)	

Diagrama de Colaboración CU:

5.1.1. Eliminación Carácter Inmediato Anterior



Referencias:

**1\*: Solicitud Lectura último carácter.** Este mensaje es implementado mediante una extensión en el caso de uso **1.3. Cancelación del Carácter Ingresado** en las condiciones descritas en el escenario de la ficha de Caso de Uso.

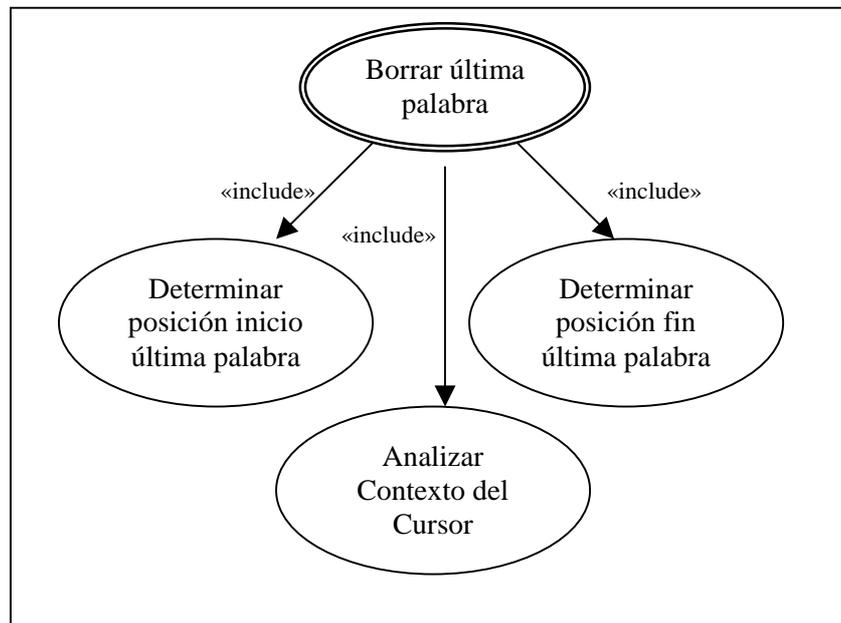
## CU: 5.1.2. Eliminación Última Palabra

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.1.2. Eliminación Última Palabra	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Eliminar del texto en edición la palabra sobre la que se encuentra situada el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación eliminar del texto la palabra sobre la que se encuentra situada el cursor. Esto es: cuando el cursor se encuentra entre dos letras que la componen, o justo después del último carácter de la misma. Si el cursor se encuentra justo antes de la primer letra de una palabra, la eliminación a producir deberá ser el carácter inmediato anterior a dicha palabra. Adicionalmente, si el cursor se encuentra al principio del texto, deberá ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>borrar la última palabra</u> . Esto se realiza seleccionando la tecla de menú de borrado (tecla de función #2), y en el menú emergente, eligiendo la opción de "Borrar última palabra" (con la tecla de función #1)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra justo antes del primer carácter de una palabra, se selecciona el carácter inmediato anterior para su eliminación.  b. Si el cursor se encuentra entre dos caracteres (posición intermedia) de una palabra, determina las posiciones del texto en que comienza y termina dicha palabra, junto a la posición relativa del cursor sobre la misma.  c. Si el cursor se encuentra exactamente al final de una palabra, determina la posición de inicio y fin de la misma.
	1.2. Se genera la verbalización (pronunciación) del mensaje de eliminación de la palabra, determinada por la subcadena del texto comprendida entre el inicio y el fin determinado según la lógica explicada, siguiendo el siguiente esquema:  a. Para el caso 1.1.a. se pronuncia "Borrar" seguido del nombre del carácter inmediato anterior a la posición del cursor, y se añade a la cadena a pronunciar, con timbre y velocidad diferenciados y en función de las características de configuración, la palabra "CURSOR" para indicar la posición del mismo.  b. Para el caso 1.1.b. se pronuncia la palabra comprendida desde las posiciones de inicio y fin determinadas, seguida de "Borrado". Con esto, si por ejemplo la palabra a eliminar es "Hola", el feedback auditivo entregado será "Hola Borrado".  c. Para el caso 1.1.c. se pronuncia la palabra comprendida desde las posiciones de inicio y fin determinadas, seguida de "Borrado".
	1.3. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.4. Se elimina del texto la subcadena comprendida entre las posiciones de inicio y fin determinadas, quedando el cursor situado en la posición relativa del texto en la que se encontraba la palabra eliminada.

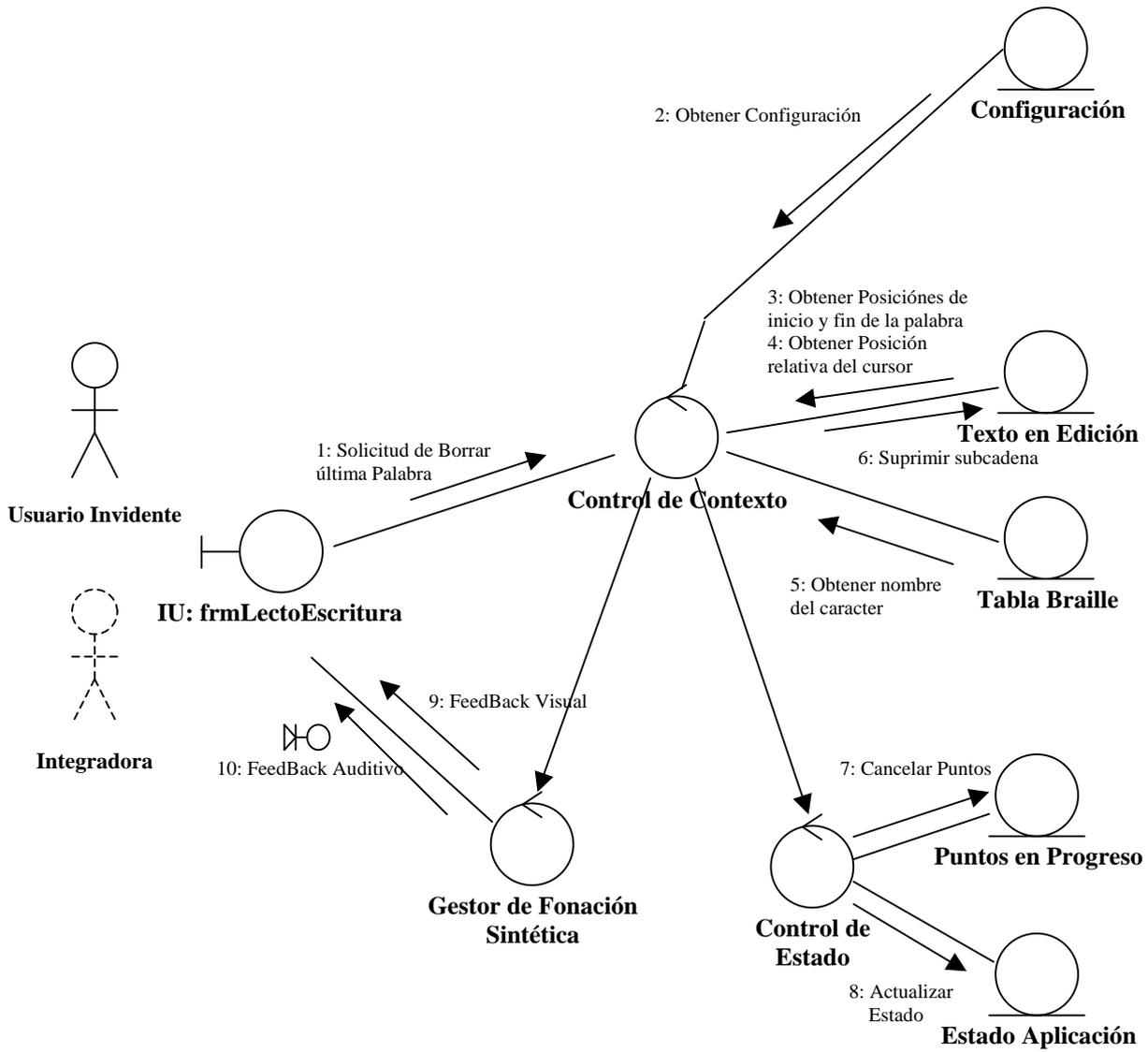
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La palabra seleccionada ha sido eliminada del texto. La posición relativa del cursor en el texto en edición se modifica para quedar situado en la ubicación relativa en que se encontraba la palabra eliminada. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al principio del texto.	1.1. Como no hay palabra previa a eliminar, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.2 del caso de uso. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	

*Diagrama de Relaciones de CU: 5.1.2. Eliminación Última Palabra*



*Nota:* Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

*Diagrama de Colaboración CU: 5.1.2. Eliminación Última Palabra*



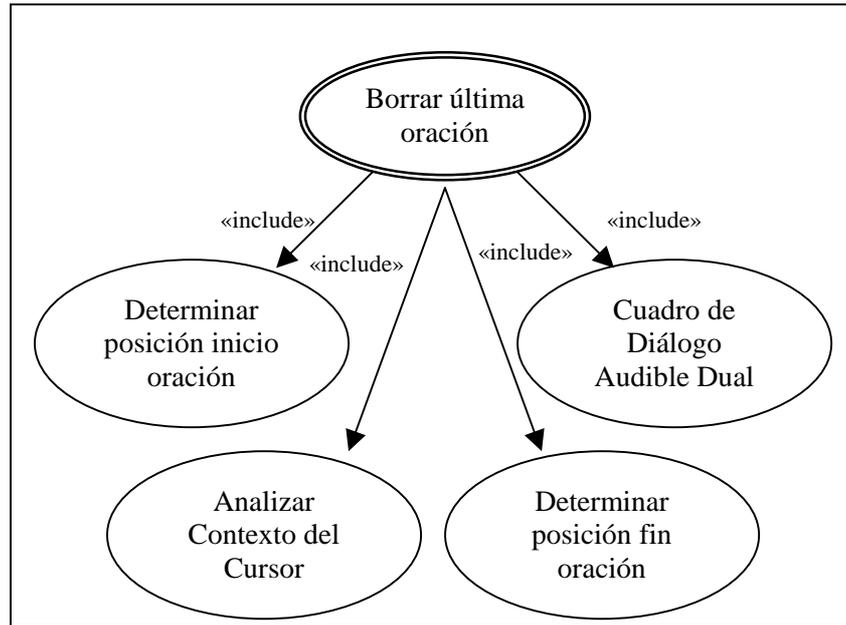
## CU: 5.1.3. Eliminación Última Oración

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.1.3. Eliminación Última Oración	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Eliminar del texto en edición la oración completa sobre la que se encuentra situada el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación eliminar del texto que se está editando la oración sobre la que se encuentra situada el cursor, sea que se encuentre al principio de la misma, al medio, o al final. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son tratados en sí mismos como oraciones, del mismo modo que los marcadores de bloque. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>borrar la última oración</u> . Esto se realiza seleccionando la tecla de menú de borrado (tecla de función #2), y en el menú emergente, eligiendo la opción de "Borrar última oración" (con la tecla de función #2)	1.1. El sistema solicita confirmación al usuario de su deseo de eliminar la oración actual. Esto se implementa mediante un cuadro de diálogo de diseño específico y reutilizable que propone auditivamente la pregunta de confirmación "¿Estás seguro que quieres borrar la oración?", ofreciéndole las opciones de "Sí" y "No" asociadas respectivamente a las teclas de función #1 y #5. Este cuadro de diálogo también se presenta visualmente para dar al usuario secundario las mismas opciones de operación.
2. El usuario confirma que desea eliminar la oración pulsando la tecla de función #1.	2.1. Se cierra la ventana emergente del cuadro de diálogo. 2.2. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica: a. Si el cursor se encuentra exactamente al principio de una oración, es decir, en la posición inmediatamente previa al primer carácter que la compone; el sistema determina la posición de inicio y fin de la oración anterior a la que allí comienza. b. Si el cursor se encuentra exactamente al final de una oración, es decir, en la posición inmediatamente posterior al último carácter que la compone; el sistema determina la posición de inicio y fin de la oración que allí termina. c. Si el cursor se encuentra en cualquier posición intermedia de una oración, determina las posiciones del texto en que comienza y termina dicha oración, junto a la posición relativa del cursor sobre la misma. 2.3. Se elimina del texto en edición la subcadena correspondiente a la oración, comprendida entre los límites de inicio y fin establecidos en el paso previo. 2.4. Se verbaliza la confirmación de la operación realizada, con el mensaje "Oración Borrada", seguida de la palabra "Cursor" (para volver a poner en contexto al usuario). 2.4.1. Si luego de la eliminación el cursor queda situado al final del texto, se añade a la verbalización la indicación de tal situación. 2.5. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.

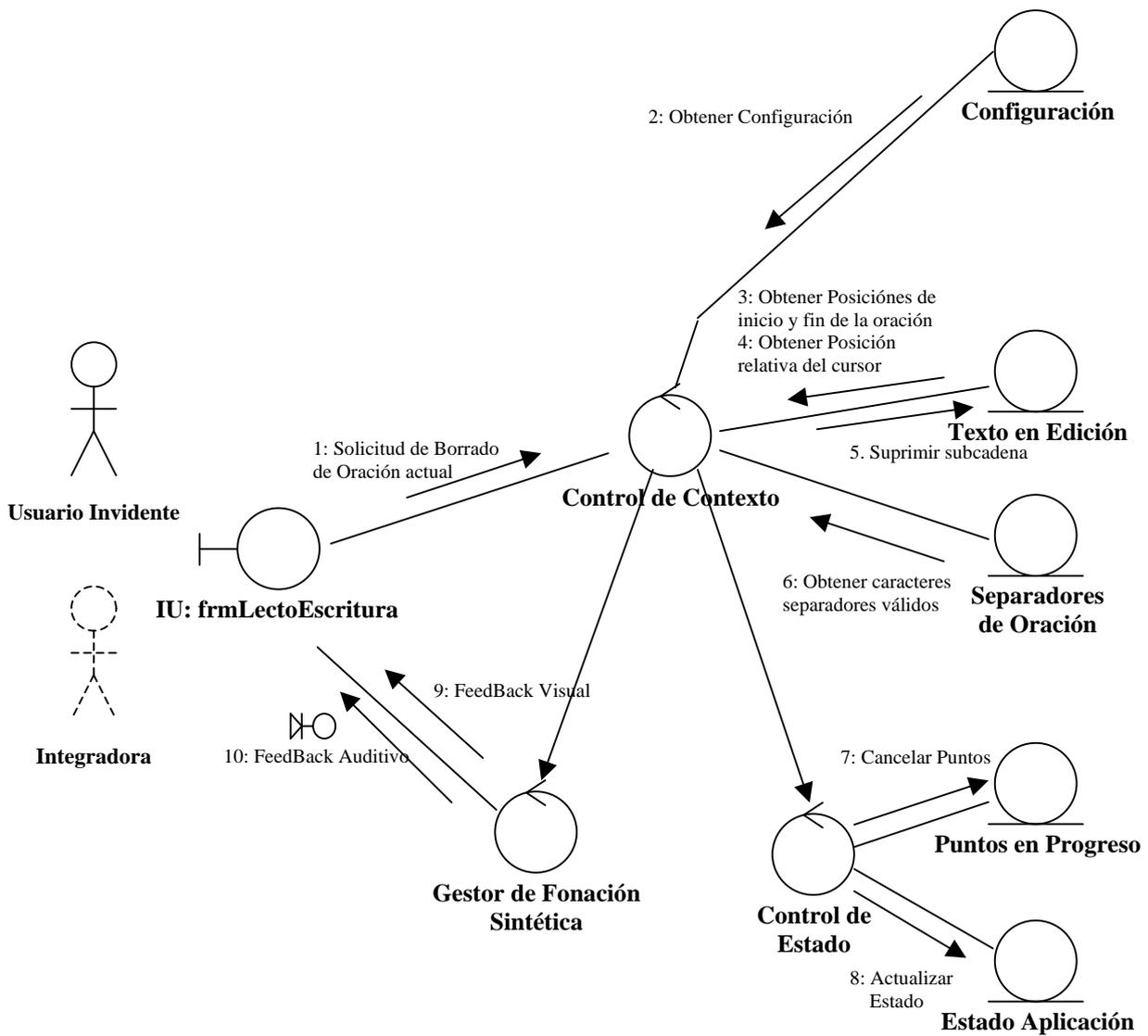
	2.6. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	2.7. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La oración seleccionada ha sido eliminada del texto. La posición relativa del cursor en el texto en edición se modifica para quedar situado en la ubicación relativa en que se encontraba la oración eliminada. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. Si el cursor se encuentra al inicio del texto.	1.1. No se elimina ninguna oración. 1.2. La cadena de lectura es reemplazada por la indicación expresa de "Inicio del Texto", seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, dando al usuario información de contexto. 1.3. Fin del caso de uso.
2. El cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	2.1. Se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 2.2. Fin del caso de uso.
3. El cursor se encontraba inmediatamente a continuación de un salto de línea: carácter de nuevo párrafo (equivalente a un ENTER en un editor de texto).	3.1. Se elimina la oración anterior a la posición del cursor, incluyendo el salto de línea. 3.2. Se continua el curso normal a partir del punto 2.3.
4. El cursor se encontraba inmediatamente a continuación un marcador de bloque: carácter especial (compuesto por "<*>").	4.1. Se elimina la marca de bloque y el salto de línea que le sigue. 4.2. Se reproduce audiblemente la indicación de "oración borrada", haciendo explícita a continuación la indicación del "cursor". 4.3. Fin del caso de uso
5. En el cuadro de diálogo presentado en el punto 1.1. del curso normal el usuario selecciona la opción "No" (tecla de función #5), o pulsa cualquier otra tecla excepto la de "Sí" (función #1).	5.1. Se vuelve al estado por defecto de la aplicación, con el menú normal disponible. 5.2. Se cancela el caso de uso.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU: 5.1.3. Eliminación Última Oración*



*Nota: Las cuatro funciones incluidas se destacan por su reutilización desde otros casos de uso.*

*Diagrama de Colaboración CU: 5.1.3. Eliminación Última Oración*



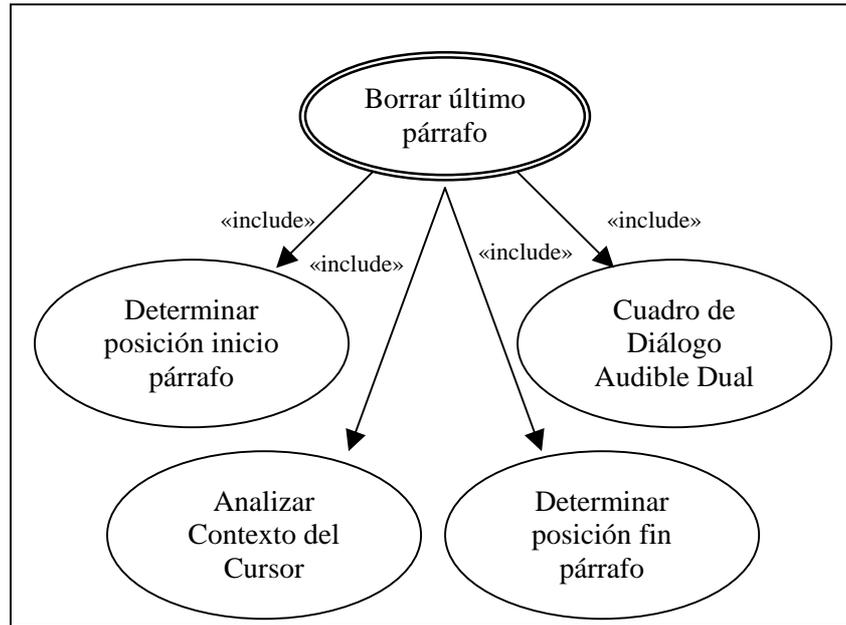
## CU: 5.1.4. Eliminación Último Párrafo

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.1.4. Eliminación Último Párrafo	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Eliminar del texto en edición el párrafo completo sobre el que se encuentra situado el cursor.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación eliminar del texto que se está editando el párrafo sobre el que se encuentra situado el cursor, sea que se encuentre al principio del mismo, al medio, o al final. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son los delimitadores de párrafos considerados. Los marcadores de bloque son considerados párrafos en sí mismos. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>borrar el último párrafo</u> . Esto se realiza seleccionando la tecla de menú de borrado (tecla de función #2), y en el menú emergente, eligiendo la opción de "Borrar último párrafo" (con la tecla de función #3)	1.1. El sistema solicita confirmación al usuario de su deseo de eliminar el párrafo actual. Esto se implementa mediante un cuadro de diálogo de diseño específico y reutilizable que propone auditivamente la pregunta de confirmación "¿Estás seguro que quieres borrar el párrafo?", ofreciéndole las opciones de "Sí" y "No" asociadas respectivamente a las teclas de función #1 y #5. Este cuadro de diálogo también se presenta visualmente para dar al usuario secundario las mismas opciones de operación.
2. El usuario confirma que desea eliminar el párrafo pulsando la tecla de función #1.	2.1. Se cierra la ventana emergente del cuadro de diálogo. 2.2. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica: a. Si el cursor se encuentra exactamente al principio de un párrafo, es decir, en la posición inmediatamente previa al primer carácter que lo compone; el sistema determina la posición de inicio y fin del párrafo que allí comienza. b. Si el cursor se encuentra exactamente al final de un párrafo, es decir, en la posición inmediatamente posterior al último carácter que lo compone; el sistema determina la posición de inicio y fin del párrafo que allí termina. c. Si el cursor se encuentra en cualquier posición intermedia de un párrafo, determina las posiciones del texto en que comienza y termina dicho párrafo, junto a la posición relativa del cursor sobre el mismo.
	2.3. Se elimina del texto en edición la subcadena correspondiente al párrafo comprendido entre los límites de inicio y fin establecidos en el paso previo, incluyendo también el carácter de nuevo párrafo al principio del mismo.
	2.4. Se verbaliza la confirmación de la operación realizada, con el mensaje "Párrafo Borrado", seguida de la palabra "Cursor" (para volver a poner en contexto al usuario).
	2.4.1. Si luego de la eliminación el cursor queda situado al final del texto, se añade a la verbalización la indicación de tal situación.
	2.5. La verbalización es entregada como feedback al usuario, y también presentada en pantalla, en el área de texto audible, para su visualización pasiva por parte de la integradora.

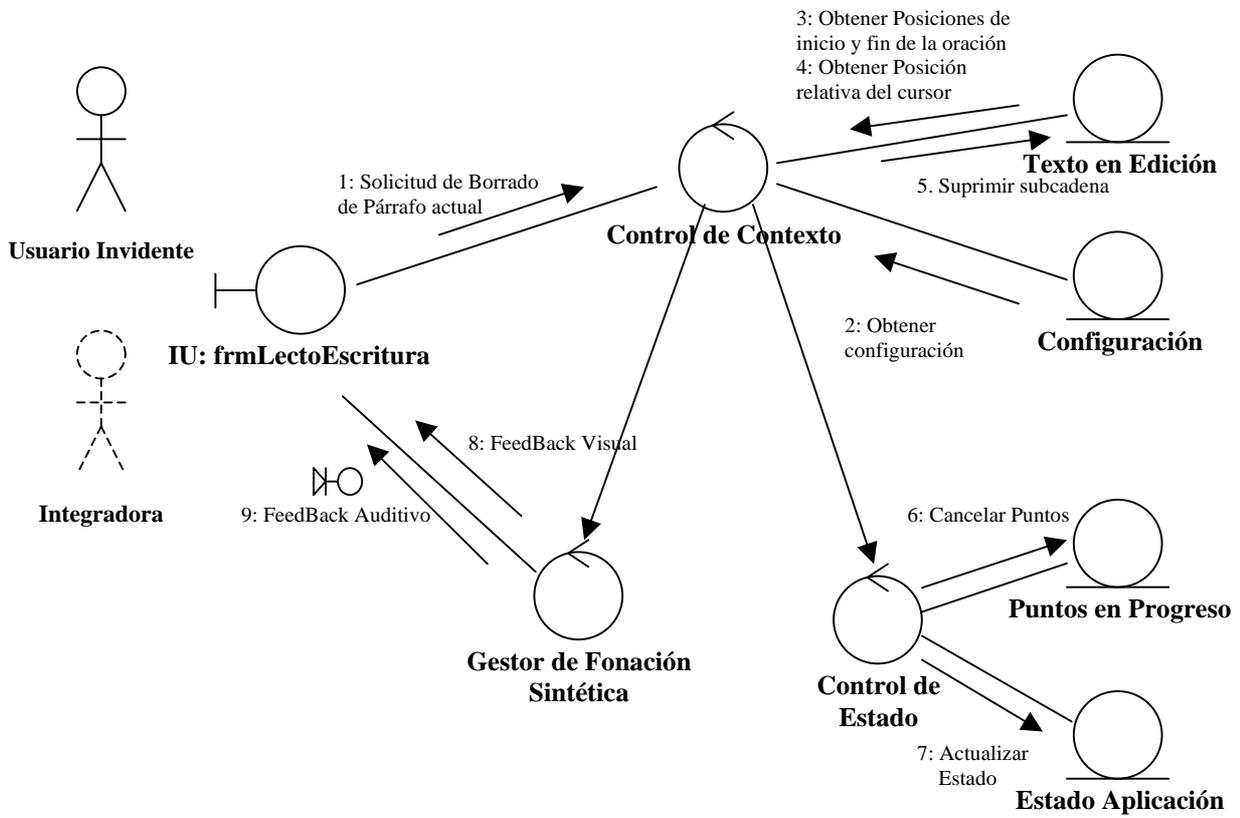
	2.6. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	2.7. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El párrafo seleccionado ha sido eliminado del texto. La posición relativa del cursor en el texto en edición se modifica para quedar situado en la ubicación relativa en que se encontraba el párrafo eliminado. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
Eventos (del Actor o Usuario) / Estado del Sistema	Respuesta del Sistema
1. Si el cursor se encuentra al inicio del texto.	1.1. No se elimina ningún párrafo. 1.2. La cadena de lectura es reemplazada por la indicación expresa de "Inicio del Texto", seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, dando al usuario información de contexto. 1.3. Fin del caso de uso.
2. El cursor se encontraba al principio del texto y al mismo tiempo al final del mismo, es decir, cuando el texto en edición no contenga ningún carácter (texto en blanco).	2.1. Se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "CURSOR" (con timbre y velocidad diferenciados y en función de las características de configuración), para indicar la posición del mismo, seguida de la indicación "Fin del Texto". Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 2.2. Fin del caso de uso.
4. El cursor se encontraba inmediatamente a continuación un marcador de bloque: carácter especial (compuesto por "<*>").	4.1. Se elimina la marca de bloque y el salto de línea que le sigue. 4.2. Se reproduce audiblemente la indicación de "párrafo borrado", haciendo explícita a continuación la indicación del "cursor". 4.3. Fin del caso de uso
5. En el cuadro de diálogo presentado en el punto 1.1. del curso normal el usuario selecciona la opción "No" (tecla de función #5), o pulsa cualquier otra tecla excepto la de "Sí" (función #1).	5.1. Se vuelve al estado por defecto de la aplicación, con el menú normal disponible. 5.2. Se cancela el caso de uso.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU: 5.1.4. Eliminación Último Párrafo



Nota: Las cuatro funciones incluidas se destacan por su reutilización desde otros casos de uso.

*Diagrama de Colaboración CU: 5.1.4. Eliminación Último Párrafo*

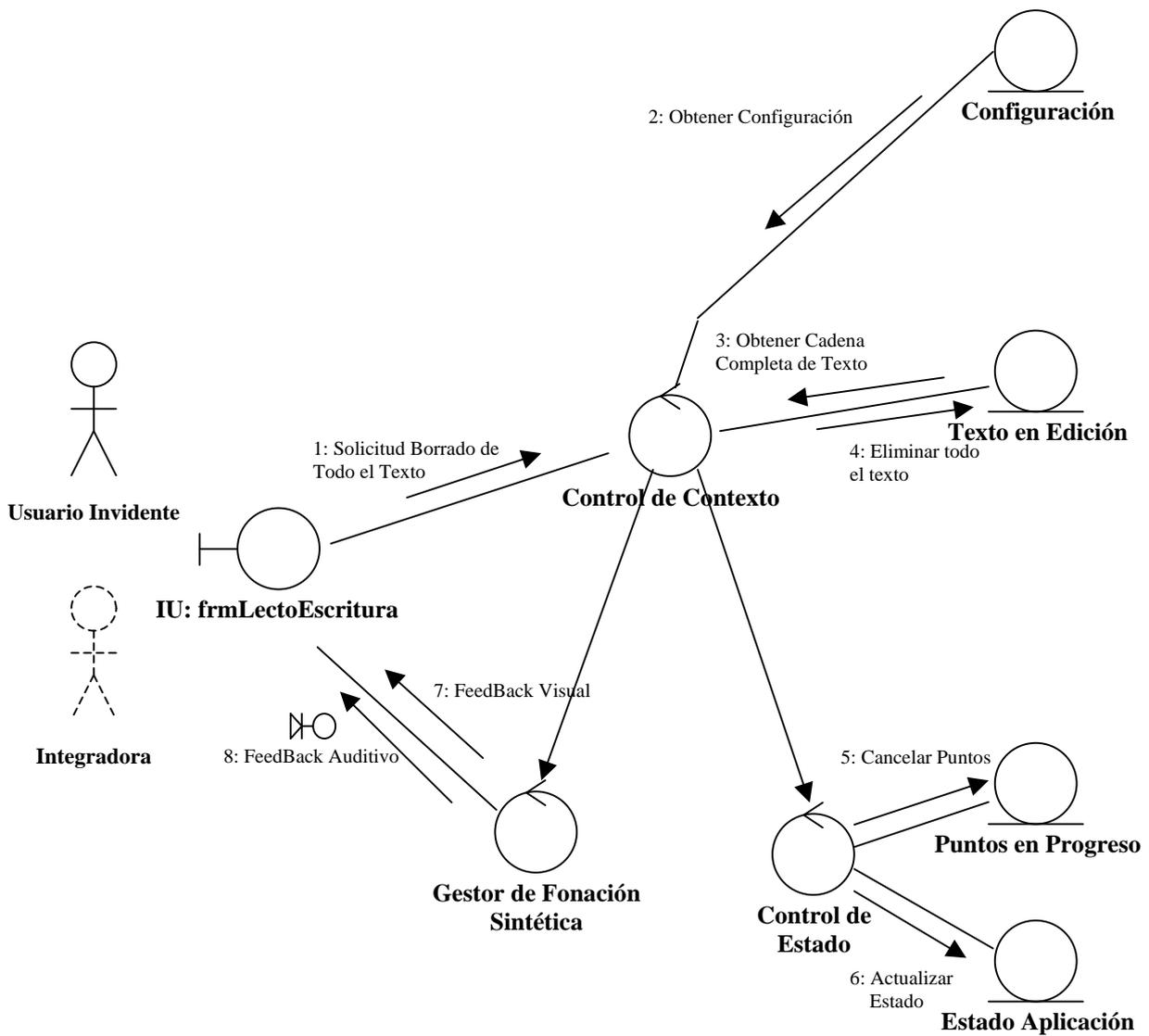


## CU: 5.1.5. Eliminación de Todo el Texto

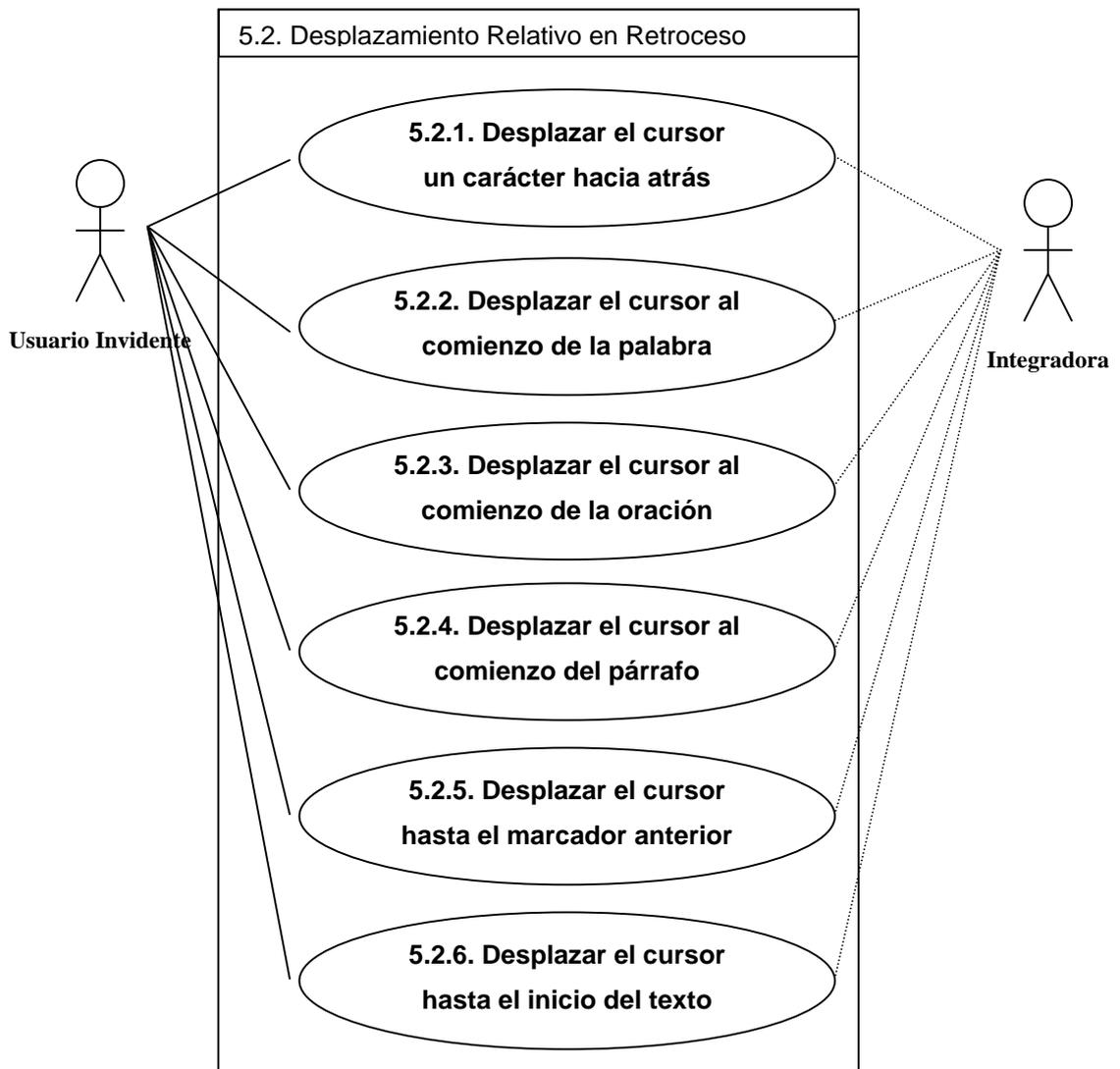
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.1.5. Eliminación de Todo el Texto	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Eliminar el texto completo en edición, dejando el área de escritura en blanco.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación eliminar el texto completo que se encuentra actualmente en edición, previa doble confirmación de su parte.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <b>Borrar todo el texto</b> . Esto se realiza seleccionando la tecla de menú de borrado (tecla de función #2), y en el menú emergente, eligiendo la opción de "Borrar todo el texto" (con la tecla de función #4)	1.1. Se solicita al usuario una doble confirmación del comando, mediante la implementación consecutiva de dos cuadros de diálogo audibles, a los que debe responder positivamente ambas veces para que el comando sea ejecutado. El primero de los cuadros le interroga "¿Estás seguro que quieres borrar todo el texto?", dándole como opciones "Sí" (función #1) y "No" (función #5). En caso de ser respondido afirmativamente, el segundo cuadro se presenta, interrogándole "¿Estás muy, pero muy, seguro?", nuevamente con las mismas opciones.
2. El usuario respondió positivamente a ambos cuadros de diálogo.	2.1. El texto completo en edición es eliminado, dejando el cursor posicionado en la cadena vacía resultante.
	2.2. Se verbaliza "Se ha borrado todo el texto", seguido de la indicación de "Cursor", según configuración, como feedback para el usuario.
	2.3. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	2.4. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El texto quedó vacío, con el cursor posicionado al principio de la cadena nula.	
<b>Excepciones:</b> No se aplican excepciones a este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Colaboración CU: 5.1.5. Eliminación de Todo el Texto



### **Explosión Caso de Uso 5.2. Desplazamiento Relativo en Retroceso**



Las funcionalidades relativas al posicionamiento relativo del cursor ( o punto de inserción ) en el texto, se estructuraron las opciones disponibles en movimientos en retroceso, y movimientos en avance, refiriendo respectivamente los desplazamientos de la posición actual del cursor hacia el inicio y el fin del texto.

En este caso de uso global, se desarrollaron seis diferentes funciones relativas al desplazamiento en retroceso del cursor, descritas en detalle en los casos de uso particulares que se presentan a continuación.

## CU: 5.2.1. Desplazar el cursor un carácter hacia atrás

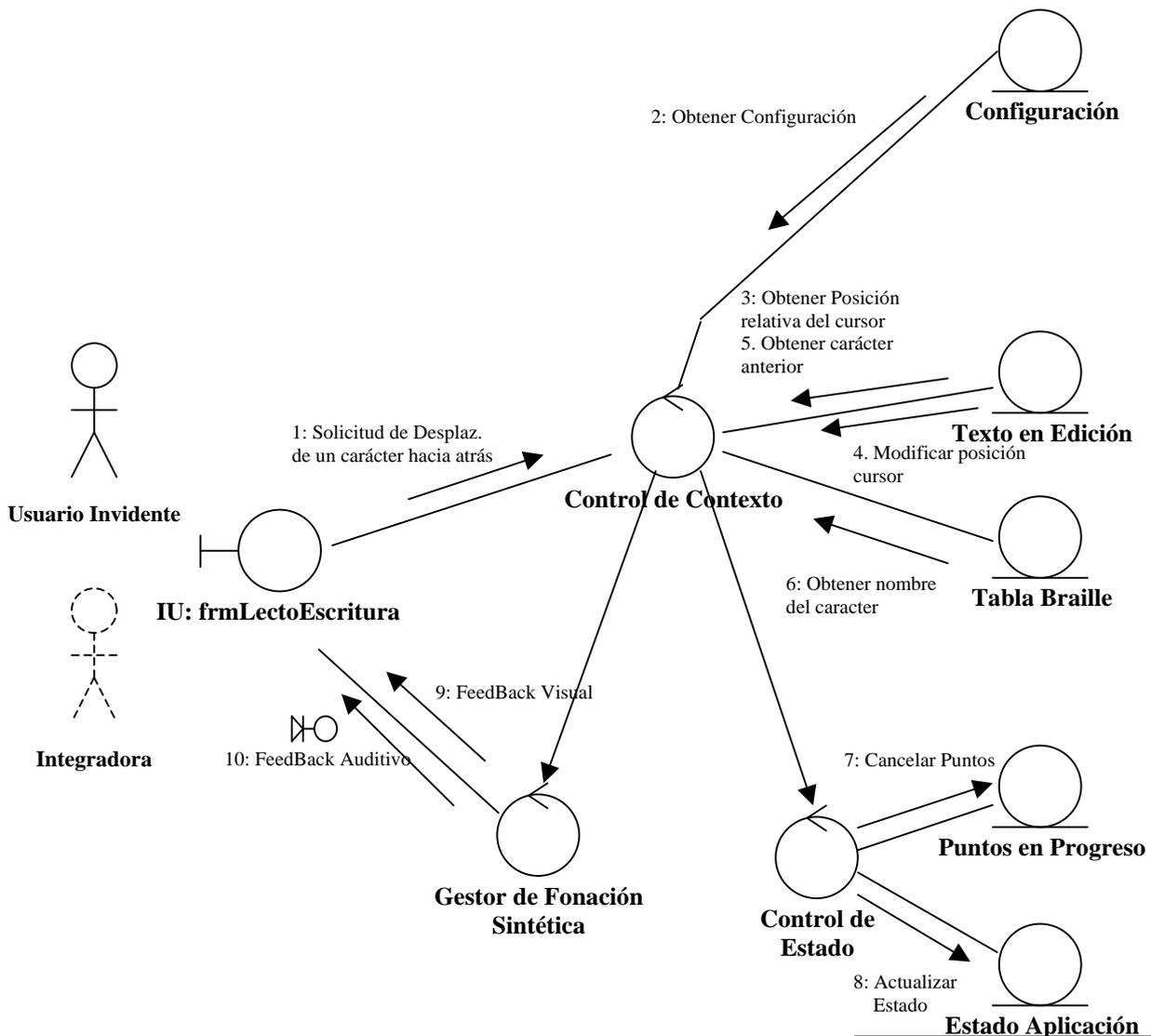
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.2.1. Desplazar el cursor un carácter hacia atrás	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición un carácter hacia atrás.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo un carácter hacia atrás.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor un carácter hacia atrás</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Un carácter hacia atrás" (con la tecla de función #5)	1.1. El sistema analiza la posición del cursor en el texto, y su contexto, según el siguiente criterio: a. Si el carácter inmediato anterior es LF (Line Feed), y el anterior a este es CR (Carriage Return), la posición de destino será la actual menos 2. b. Si el carácter inmediato anterior es ">" y los dos anteriores a ese son "<*", se trata de un marcador de bloque, por lo que la posición de destino será la actual menos 3. c. En cualquier otro caso, la posición de destino será la actual menos 1. (Ver Excepciones)
	1.2. Se sitúa el cursor en la posición de destino establecida según el criterio anterior.
	1.3. Se identifica el carácter anterior a la nueva posición del cursor, siguiendo el mismo criterio expresado en 1.1: a. Si el carácter inmediato anterior es LF (Line Feed), y el anterior a este es CR (Carriage Return), el carácter compuesto es un "nuevo párrafo". b. Si el carácter inmediato anterior es ">" y los dos anteriores a éste son "<*", se trata de una "marca de bloque". c. En cualquier otro caso, el carácter es un ASCII simple. En este caso se buscará el mismo en la tabla Braille para determinar su denominación.
	1.4. Se produce la verbalización del nombre del carácter inmediato anterior a la nueva posición del cursor, que es entregada como feedback al usuario, además de ser presentada en la interfaz gráfica al usuario secundario.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modifica al carácter (simple o compuesto) inmediato anterior a la previa. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. En las consideraciones del punto 1.1 del escenario, el cursor se encontraba al inicio del texto (posición relativa = 0).	1.1. Se informa al usuario mediante la interfaz auditiva que se encuentra al "Inicio del Texto", seguido del indicador de "Cursor", según configuración de la aplicación. No se produce ningún desplazamiento del cursor. 1.2. Fin del caso de uso.

<b>Caso de uso:</b> 5.2.1. <i>Desplazar el cursor un carácter hacia atrás</i> (Cont...)	
2. En las consideraciones del punto 1.3.c. del escenario, el carácter inmediato anterior a la nueva posición del cursor no existe en la tabla Braille.	2.1. Se refiere a ese carácter como "carácter desconocido", respetando su valor ASCII y ubicación, pese a no ser legible para la aplicación.
	2.2. Se verbaliza dicha denominación seguida del indicador de "Cursor", según configuración de la aplicación, para situar al usuario en contexto.
	2.3. Se continúa con el flujo normal desde el punto 1.4. del escenario.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Colaboración CU:

5.2.1. *Desplazar el cursor un carácter hacia atrás*



## CU: 5.2.2. Desplazar el cursor al comienzo de la palabra

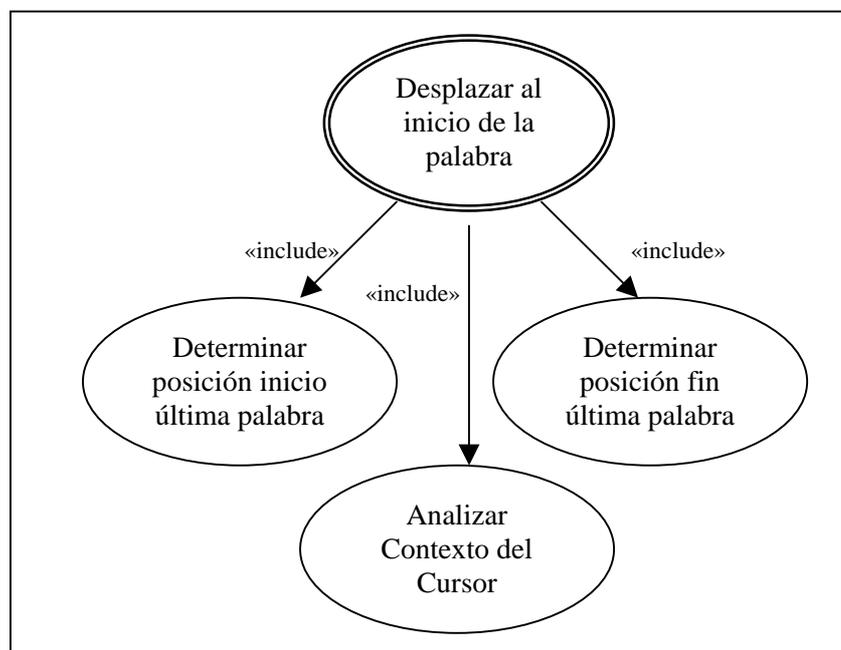
### Ficha de Caso de Uso

<b>Caso de uso:</b> 5.2.2. Desplazar el cursor al comienzo de la palabra	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio de la palabra actual.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el inicio de la palabra sobre la cual se encuentra. Esto es: cuando el cursor se encuentra entre dos letras que la componen, o justo después del último carácter de la misma. Si el cursor se encuentra justo antes de la primer letra de una palabra, el desplazamiento a producir deberá ser al carácter inmediato anterior a dicha palabra. Adicionalmente, si el cursor se encuentra al principio del texto, deberá ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo de la palabra</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Al comienzo palabra actual" (con la tecla de función #1)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra justo antes del primer carácter de una palabra, se selecciona la posición del carácter inmediato anterior para su reubicación.  b. Si el cursor se encuentra entre dos caracteres (posición intermedia) de una palabra, determina la posición del texto en que comienza dicha palabra.  c. Si el cursor se encuentra exactamente al final de una palabra, determina la posición del texto en que comienza dicha palabra.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se reanaliza la cadena de texto en edición, a partir de la nueva posición del cursor, identificando la palabra anterior a la misma para su devolución como feedback al usuario, siguiendo el mismo criterio enunciado en el punto 1.1.
	1.4. Se verbaliza la palabra (o nombre del carácter si se trata de un carácter simple en vez de un vocablo), seguido del indicativo de "cursor" según la configuración de la aplicación, para aportar información del nuevo contexto al usuario.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al inicio de la palabra sobre la cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	

<b>Caso de uso: 5.2.2. Desplazar el cursor al comienzo de la palabra</b> <span style="float: right;">(Cont...)</span>	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al principio del texto al comenzar el caso de uso.	1.1. Como no hay palabra previa al principio de la cual mover el cursor, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.4. del escenario. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 1.2. Fin del caso de uso
2. El cursor se encuentra sobre la primer palabra del texto.	2.1. Se realiza el desplazamiento del cursor según se describe en el escenario, hasta el punto 1.2. inclusive. Como no hay palabra previa a la nueva ubicación del cursor, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.4. del escenario. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU:*

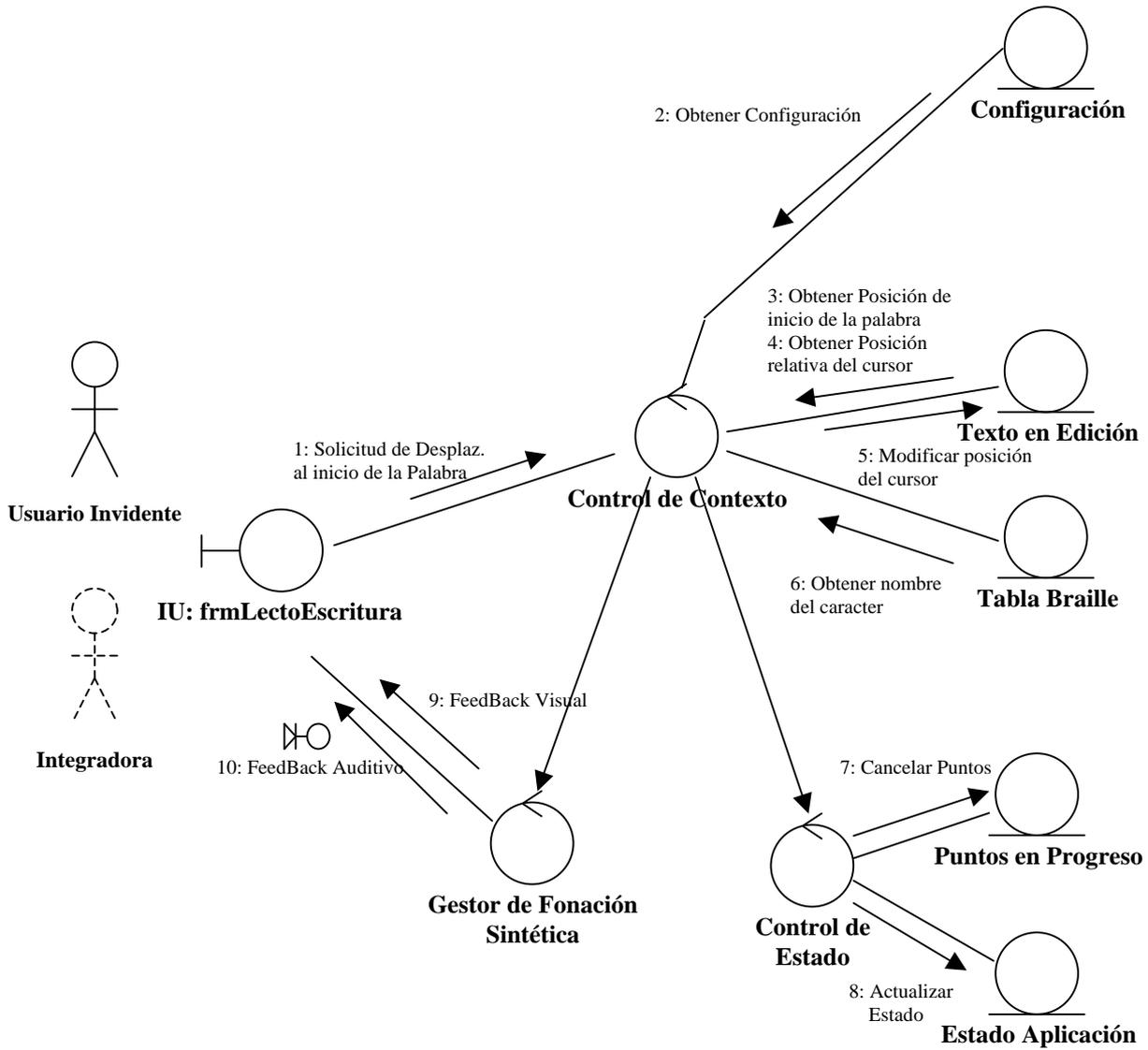
*5.2.2. Desplazar el cursor al comienzo de la palabra*



*Nota:* Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.2.2. Desplazar el cursor al comienzo de la palabra



### CU: 5.2.3. Desplazar el cursor al comienzo de la oración

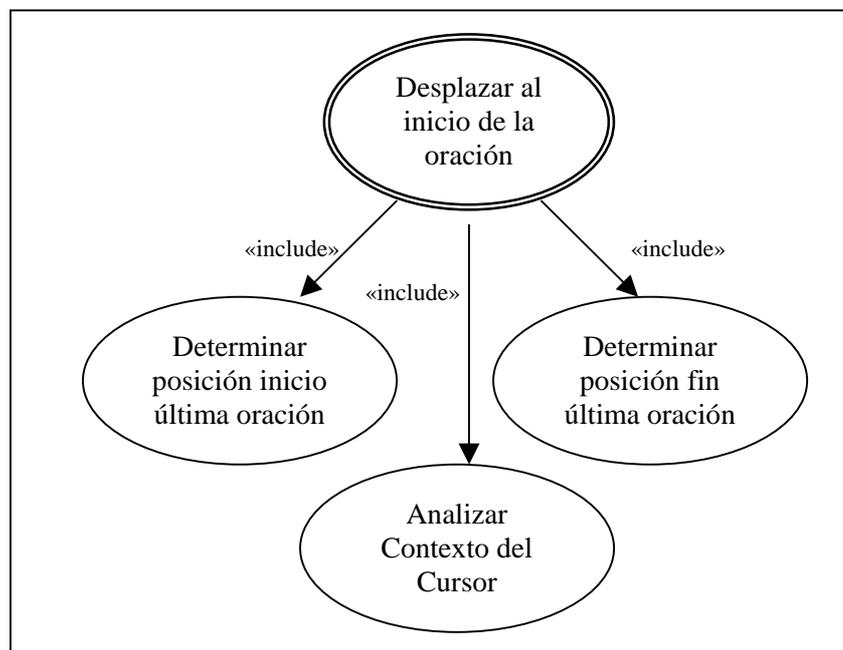
#### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.2.3. Desplazar el cursor al comienzo de la oración	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio de la oración actual.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el inicio de la oración sobre la cual se encuentra. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son tratados en sí mismos como oraciones, del mismo modo que los marcadores de bloque. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo de la oración</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Al comienzo oración actual" (con la tecla de función #2)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra exactamente al principio de una oración, es decir, en la posición inmediatamente previa al primer carácter que la compone; el sistema determina la posición de inicio de la oración anterior a la que allí comienza.  b. Si el cursor se encuentra exactamente al final de una oración, es decir, en la posición inmediatamente posterior al último carácter que la compone; el sistema determina la posición de inicio de la oración que allí termina.  c. Si el cursor se encuentra en cualquier posición intermedia de una oración, determina la posición del texto en que comienza dicha oración, junto a la posición relativa del cursor sobre la misma.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al inicio de la oración sobre la cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al principio del texto al comenzar el caso de uso.	1.1. Como no hay ninguna oración previa a la cual mover el cursor, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso

<b>Caso de uso:</b> 5.2.3. <i>Desplazar el cursor al comienzo de la oración</i> (Cont...)	
2. El cursor se encuentra inmediatamente a continuación de un salto de párrafo.	2.1. Se realiza el desplazamiento del cursor a la posición inmediato anterior al CR del Salto de Párrafo.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

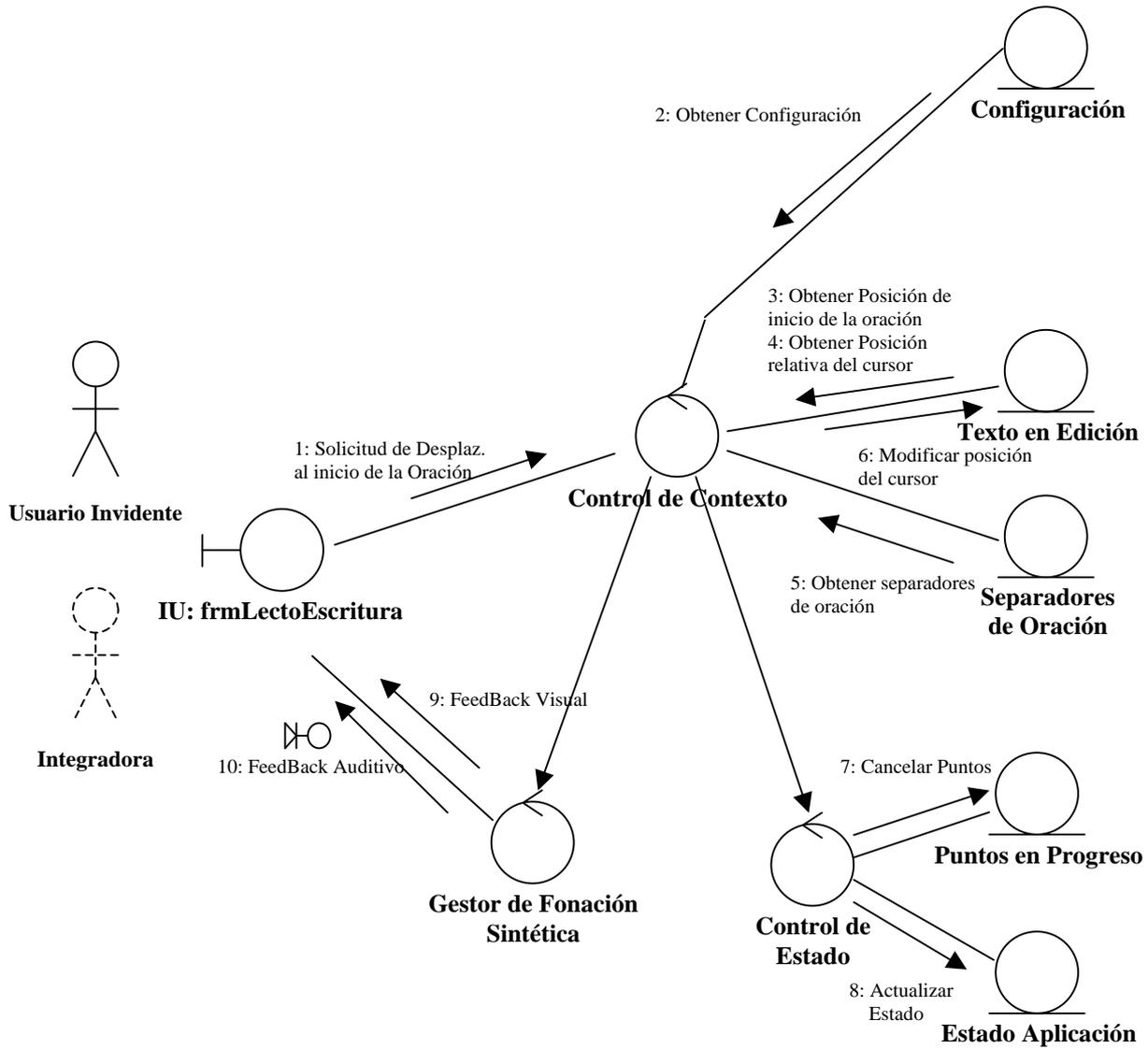
5.2.3. *Desplazar el cursor al comienzo de la oración*



*Nota:* Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.2.3. Desplazar el cursor al comienzo de la oración



## CU: 5.2.4. Desplazar el cursor al comienzo del párrafo

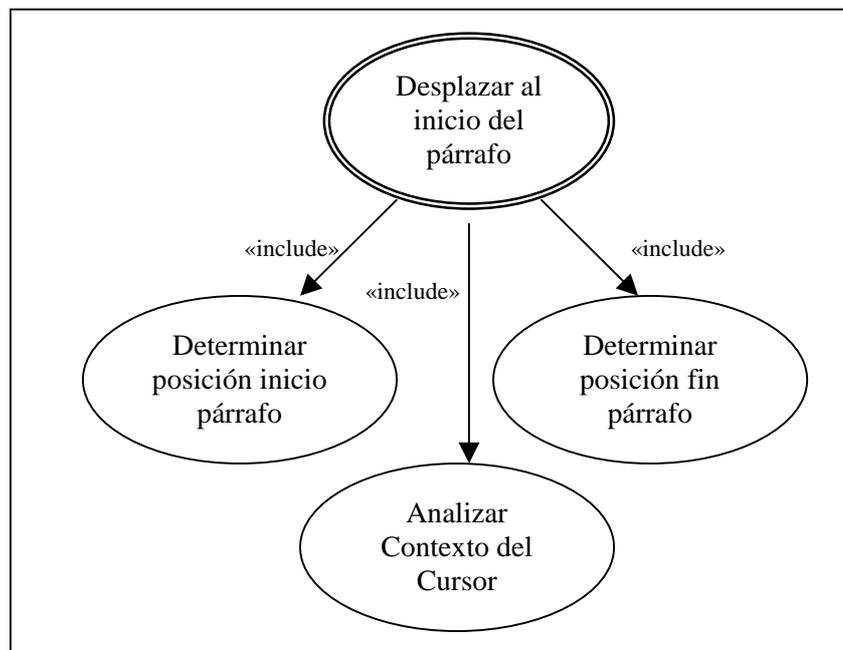
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.2.4. Desplazar el cursor al comienzo del párrafo	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio del párrafo actual.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el inicio del párrafo sobre el cual se encuentra. Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo del párrafo</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Al comienzo párrafo actual" (con la tecla de función #3)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra exactamente al principio de un párrafo, es decir, en la posición inmediatamente previa al primer carácter que lo compone; el sistema determina la posición de inicio del párrafo anterior al que allí comienza.  b. Si el cursor se encuentra exactamente al final de un párrafo, es decir, en la posición inmediatamente posterior al último carácter que lo compone; el sistema determina la posición de inicio del párrafo que allí termina.  c. Si el cursor se encuentra en cualquier posición intermedia de un párrafo, determina la posición del texto en que comienza dicho párrafo, junto a la posición relativa del cursor sobre el mismo.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al inicio del párrafo sobre el cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al principio del texto al comenzar el caso de uso.	1.1. Como no hay ningún párrafo previo a la cual mover el cursor, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso

<b>Caso de uso:</b> 5.2.4. <i>Desplazar el cursor al comienzo del párrafo</i> (Cont...)	
2. El cursor se encuentra inmediatamente a continuación de un salto de párrafo.	2.1. Se realiza el desplazamiento del cursor a la posición inmediato anterior al CR del Salto de Párrafo.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

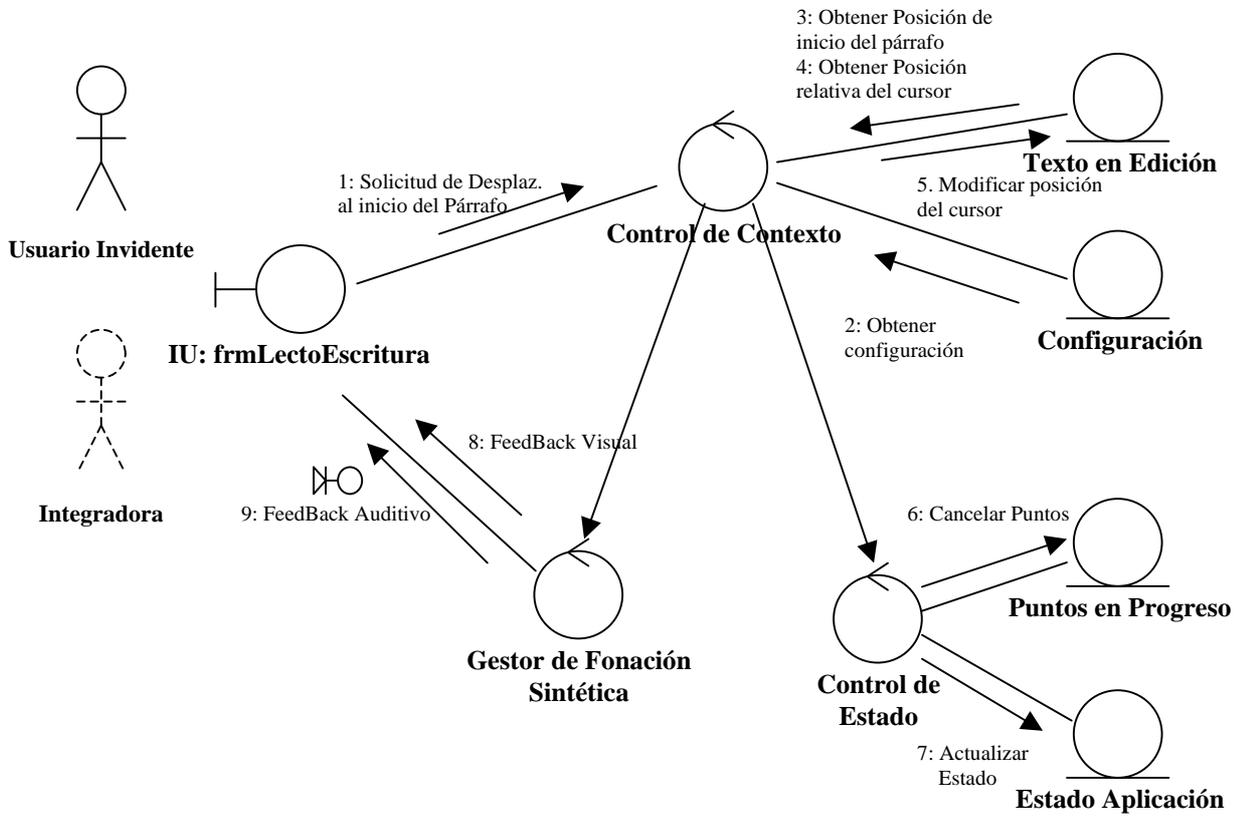
5.2.4. *Desplazar el cursor al comienzo del párrafo*



Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.2.4. Desplazar el cursor al comienzo del párrafo



## CU: 5.2.5. Desplazar el cursor hasta el marcador anterior

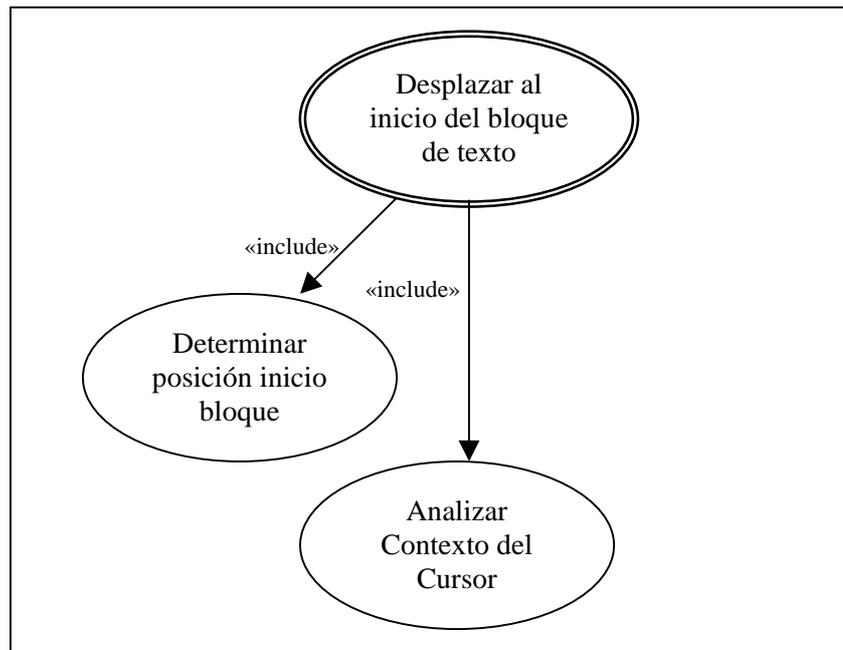
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.2.5. Desplazar el cursor hasta el marcador anterior	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio del Bloque de Texto actual.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el inicio del Bloque de Texto sobre el cual se encuentra. Un bloque de texto se delimita con los marcadores de bloque: carácter especial compuesto "<*>". Adicionalmente, si el cursor se encuentra al principio del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo del bloque actual</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Hasta la marca anterior" (con la tecla de función #6)	<p>1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:</p> <p>a. Si el cursor se encuentra exactamente al principio de un bloque, es decir, en la posición inmediatamente posterior a la marca de bloque que lo delimita; el sistema determina la posición de inicio del bloque de texto anterior al que allí comienza.</p> <p>b. Si el cursor se encuentra en una posición intermedia o exactamente al final de un bloque, es decir, en la posición inmediatamente previa a la siguiente marca de bloque; el sistema determina la posición de inicio del bloque de texto sobre el que se encuentra.</p> <p>1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.</p> <p>1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.</p> <p>1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.</p> <p>1.5. Fin del caso de uso.</p>
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al inicio del bloque de texto sobre el cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al principio del texto al comenzar el caso de uso.	<p>1.1. Como no hay ningún bloque previo al cual mover el cursor, simplemente se reproducen audiblemente las palabras "Inicio del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.</p> <p>1.2. Fin del caso de uso</p>

<b>Caso de uso:</b> 5.2.5. <i>Desplazar el cursor hasta el marcador anterior</i> (Cont...)	
2. El cursor se encuentra en una posición intermedia entre el inicio del texto y la primer marca de bloque; o no existen marcas de bloque.	2.1. Se realiza el desplazamiento del cursor a la posición inicial del texto.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

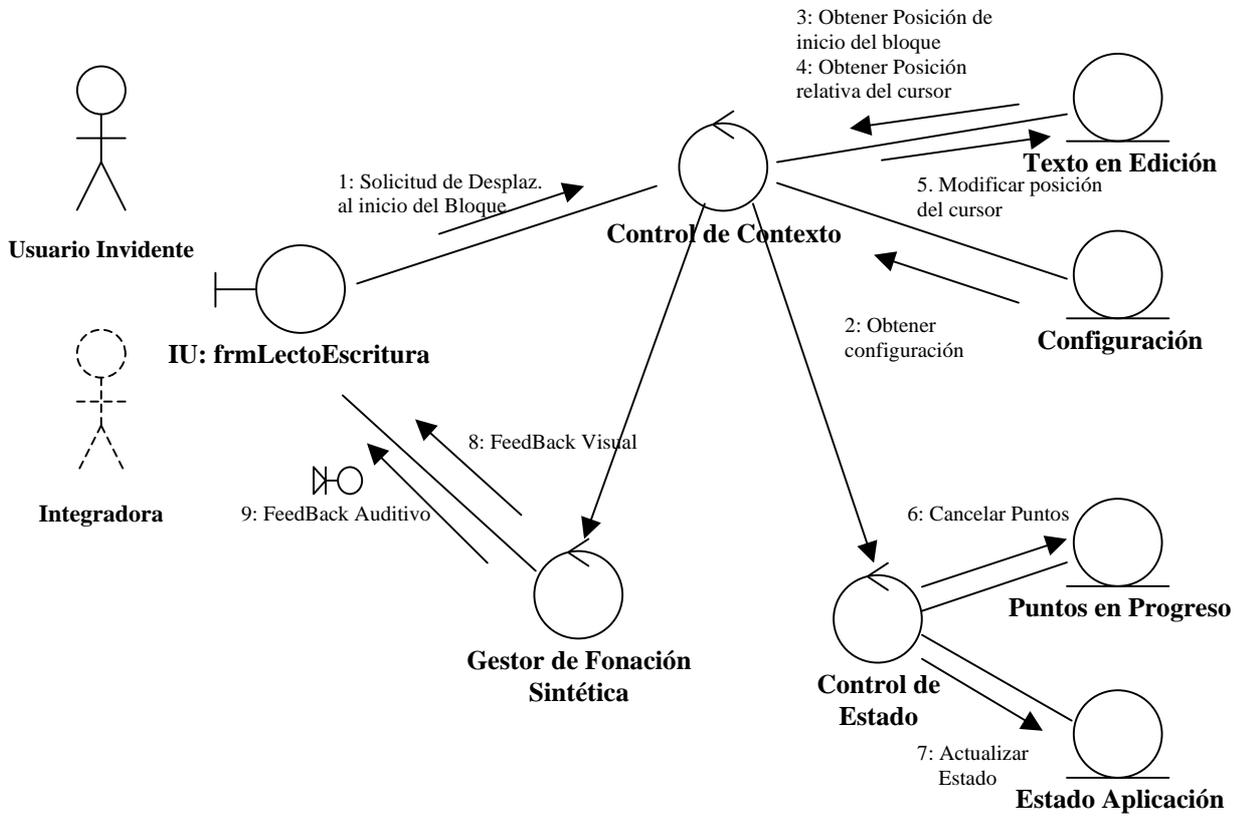
5.2.5. *Desplazar el cursor hasta el marcador anterior*



Nota: Las dos funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.2.5. Desplazar el cursor hasta el marcador anterior



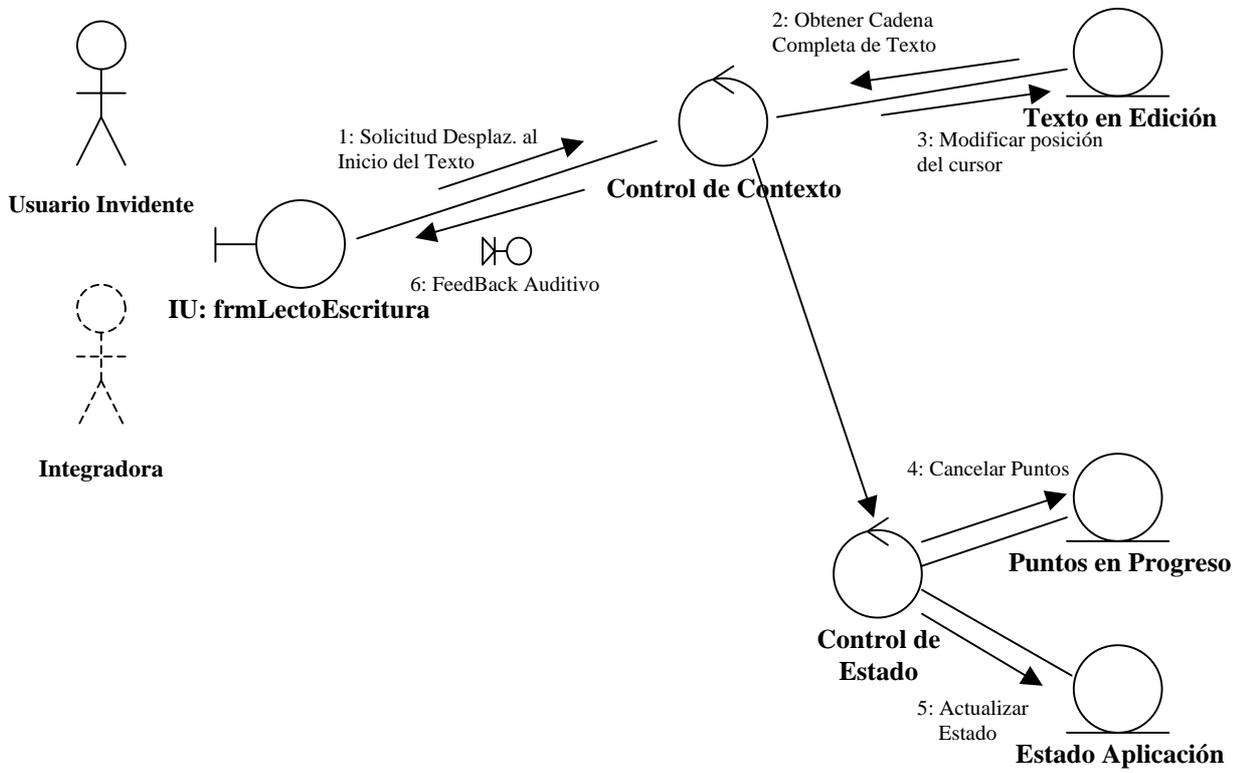
## CU: 5.2.6. Desplazar el cursor hasta el inicio del texto

### *Ficha de Caso de Uso*

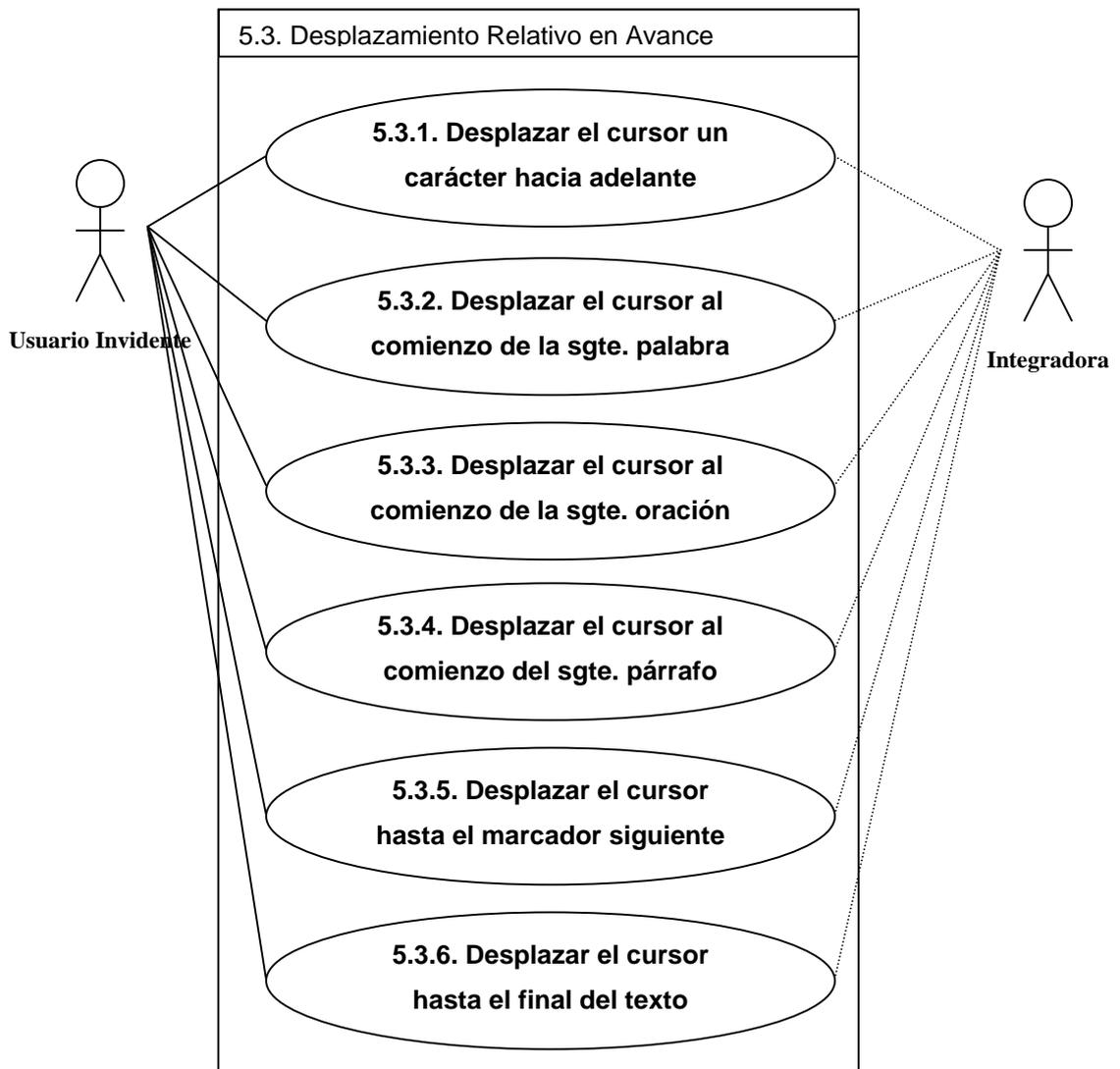
<b>Caso de uso:</b> 5.2.6. Desplazar el cursor hasta el inicio del texto	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio del Texto.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hasta el inicio del mismo: posición cero.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo del texto</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #3), y en el menú emergente, eligiendo la opción de "Al comienzo del texto" (con la tecla de función #4)	1.1. El cursor es ubicado en la posición cero.
	1.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.3. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.4. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El cursor quedó ubicado en la posición cero del texto, y la aplicación con el menú normal, lista para recibir el ingreso de nuevos caracteres.	
<b>Excepciones:</b> No se aplican excepciones a este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	

Diagrama de Colaboración CU:

5.2.6. Desplazar el cursor hasta el inicio del texto



### **Explosión Caso de Uso 5.3. Desplazamiento Relativo en Avance**



Las funcionalidades relativas al posicionamiento relativo del cursor ( o punto de inserción ) en el texto, se estructuraron las opciones disponibles en movimientos en retroceso, y movimientos en avance, refiriendo respectivamente los desplazamientos de la posición actual del cursor hacia el inicio y el fin del texto.

En este caso de uso global, se desarrollaron seis diferentes funciones relativas al desplazamiento en avance del cursor, descritas en detalle en los casos de uso particulares que se presentan a continuación.

### 5.3.1. Desplazar el cursor un carácter hacia adelante

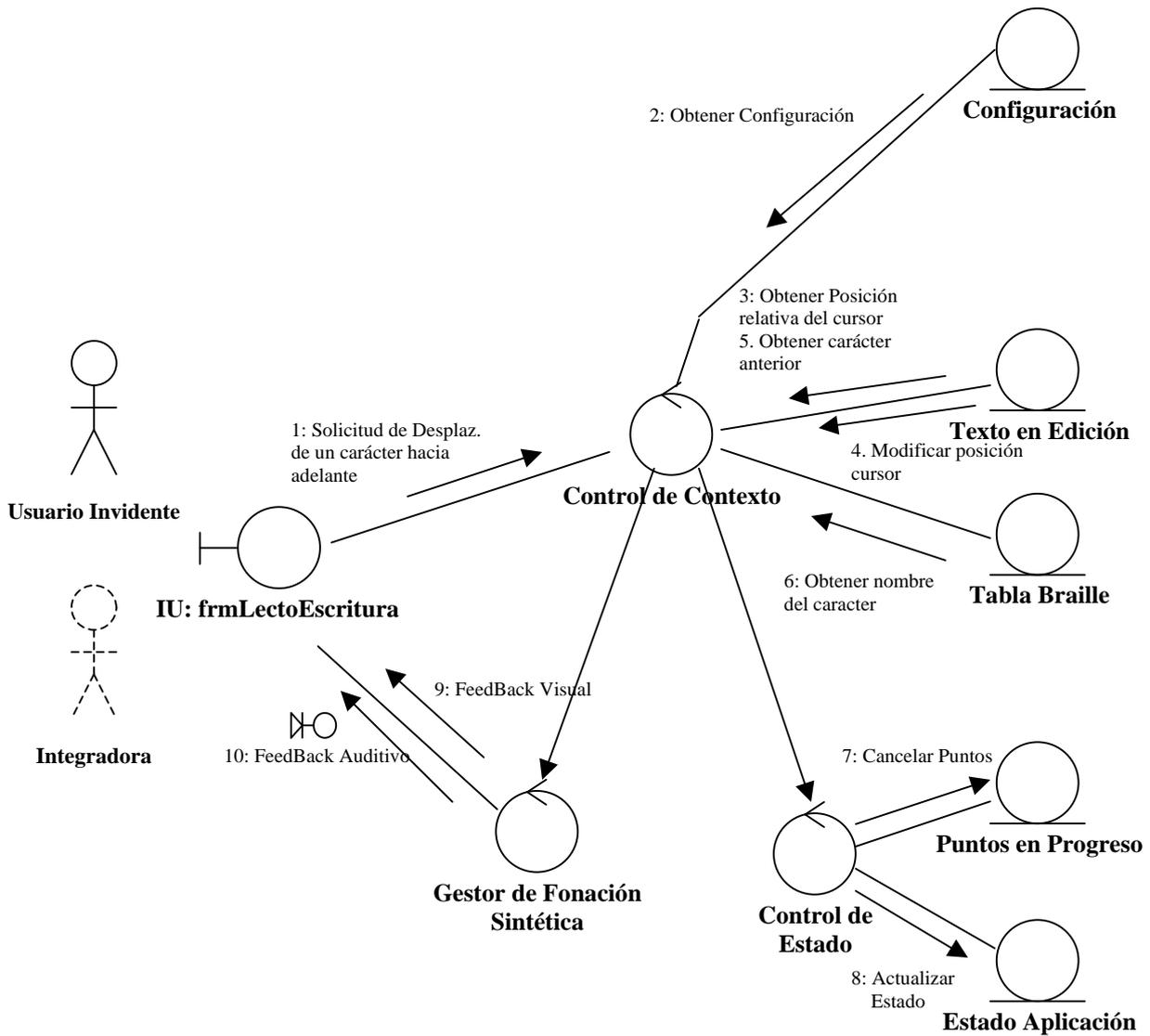
#### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.3.1. Desplazar el cursor un carácter hacia adelante	
<b>Actor Principal:</b> <i>Usuario No Vidente</i>	
<b>Actor Secundario:</b> <i>Asistente Pedagógica Integradora</i>	
<b>Objetivos / propósito:</b> <i>Desplazar el cursor en el texto en edición un carácter hacia adelante.</i>	
<b>Resumen:</b> <i>Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo un carácter hacia adelante.</i>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor un carácter hacia adelante</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Retroceso (tecla de función #7), y en el menú emergente, eligiendo la opción de "Un carácter hacia adelante" (con la tecla de función #5)	1.1. El sistema analiza la posición del cursor en el texto, y su contexto, según el siguiente criterio: a. Si el carácter inmediato posterior es CR (Carriage Return), y el siguiente a este es LF (Line Feed), la posición de destino será la actual más 2. b. Si el carácter inmediato posterior es "<" y los dos siguientes a ese son ">", se trata de un marcador de bloque, por lo que la posición de destino será la actual más 3. c. En cualquier otro caso, la posición de destino será la actual más 1. (Ver Excepciones)
	1.2. Se sitúa el cursor en la posición de destino establecida según el criterio anterior.
	1.3. Se identifica el carácter anterior a la nueva posición del cursor, siguiendo el siguiente criterio: a. Si el carácter inmediato anterior es LF (Line Feed), y el anterior a este es CR (Carriage Return), el carácter compuesto es un "nuevo párrafo". b. Si el carácter inmediato anterior es ">" y los dos anteriores a éste son "<*", se trata de una "marca de bloque". c. En cualquier otro caso, el carácter es un ASCII simple. En este caso se buscará el mismo en la tabla Braille para determinar su denominación.
	1.4. Se produce la verbalización del nombre del carácter inmediato anterior a la nueva posición del cursor, que es entregada como feedback al usuario, además de ser presentada en la interfaz gráfica al usuario secundario.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> <i>Sin comentarios adicionales. Ver excepciones.</i>	
<b>Poscondiciones:</b> <i>La posición relativa del cursor en el texto en edición se modifica al carácter (simple o compuesto) inmediato posterior a la previa. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.</i>	
<b>Excepciones:</b> <i>Las excepciones consideradas son las siguientes</i>	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. En las consideraciones del punto 1.1 del escenario, el cursor se encontraba al final del texto.	1.1. Se informa al usuario mediante la interfaz auditiva que se encuentra al "Inicio del Texto". No se produce ningún desplazamiento del cursor.
	1.2. Fin del caso de uso.
2. En las consideraciones del punto 1.3.c. del escenario, el carácter inmediato anterior a la nueva posición del cursor no existe en la tabla Braille.	2.1. Se refiere a ese carácter como "carácter desconocido", respetando su valor ASCII y ubicación, pese a no ser legible para la aplicación.
	2.2. Se verbaliza dicha denominación.
	2.3. Se continúa con el flujo normal desde el punto 1.4. del escenario.

<b>Caso de uso:</b> 5.3.1. Desplazar el cursor un carácter hacia delante	(Continuación...)
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Colaboración CU:

5.3.1. Desplazar el cursor un carácter hacia adelante



## CU: 5.3.2. Desplazar el cursor al comienzo de la sgte. palabra

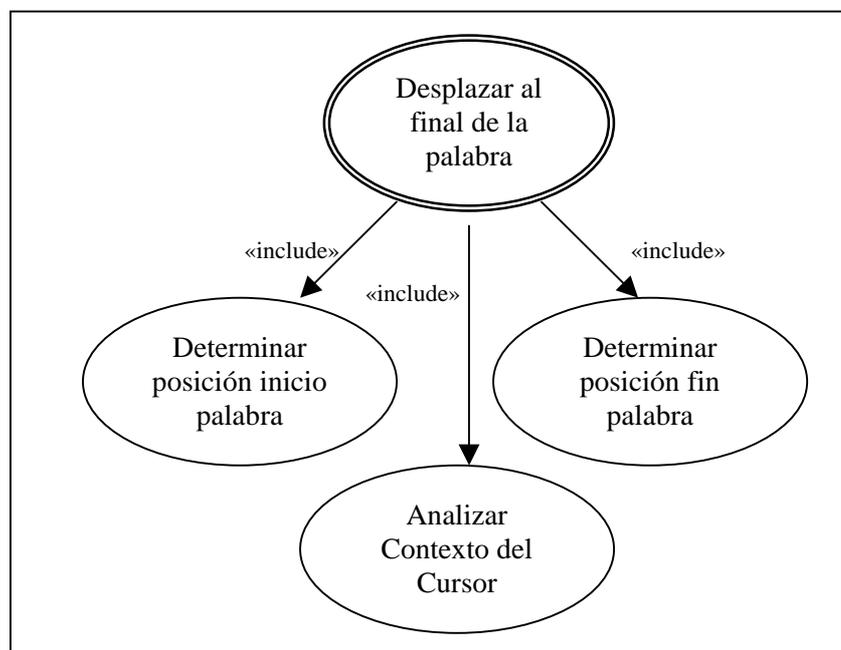
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.3.2. Desplazar el cursor al comienzo de la sgte. Palabra	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el fin de la palabra actual (inicio de la palabra siguiente).	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el final de la palabra sobre la cual se encuentra. Esto es: cuando el cursor se encuentra entre dos letras que la componen, o justo antes del primer carácter de la misma. Si el cursor se encuentra justo después de la última letra de una palabra, el desplazamiento a producir deberá ser el carácter inmediato posterior a dicha palabra. Adicionalmente, si el cursor se encuentra al final del texto, deberá ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo de la siguiente palabra</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Avance (tecla de función #7), y en el menú emergente, eligiendo la opción de "Al comienzo siguiente palabra" (con la tecla de función #1)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra justo después del último carácter de una palabra, se selecciona la posición donde termina el carácter inmediato posterior para su reubicación.  b. Si el cursor se encuentra entre dos caracteres (posición intermedia) de una palabra, determina la posición del texto en que finaliza dicha palabra.  c. Si el cursor se encuentra exactamente al principio de una palabra, determina la posición del texto en que finaliza dicha palabra.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se reanaliza la cadena de texto en edición, a partir de la nueva posición del cursor, identificando la palabra anterior a la misma para su devolución como feedback al usuario, siguiendo el criterio opuesto al enunciado en el punto 1.1.
	1.4. Se verbaliza la palabra (o nombre del carácter si se trata de un carácter simple en vez de un vocablo), seguido del indicativo de "cursor" según la configuración de la aplicación, para aportar información del nuevo contexto al usuario.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al final de la palabra sobre la cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	

<b>Caso de uso: 5.3.2. Desplazar el cursor al comienzo de la sgte. Palabra</b> (Cont...)	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al final del texto al comenzar el caso de uso.	1.1. Como no hay palabra posterior al final de la cual mover el cursor, simplemente se reproducen audiblemente las palabras "Fin del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.4. del escenario. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 1.2. Fin del caso de uso
2. El cursor se encuentra sobre la última palabra del texto.	2.1. Se realiza el caso de uso de modo normal, sólo que se añade la reproducción audible de palabras "Fin del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor" como se señala en el punto 1.4. del escenario. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora. 2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

*Diagrama de Relaciones de CU:*

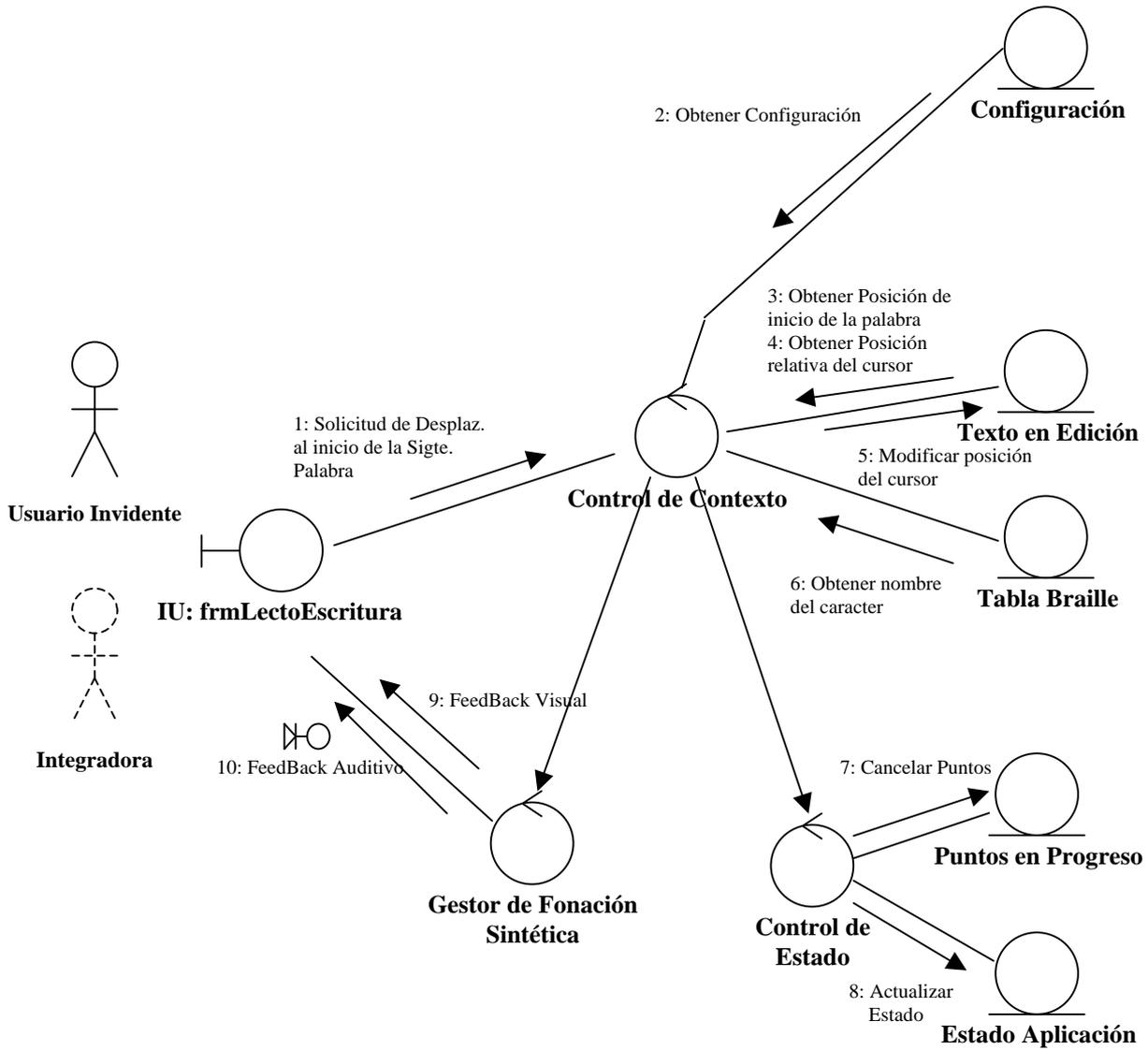
*5.3.2. Desplazar el cursor al comienzo de la sgte. palabra*



*Nota:* Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.3.2. Desplazar el cursor al comienzo de la sigte. palabra



### CU: 5.3.3. Desplazar el cursor al comienzo de la sgte. oración

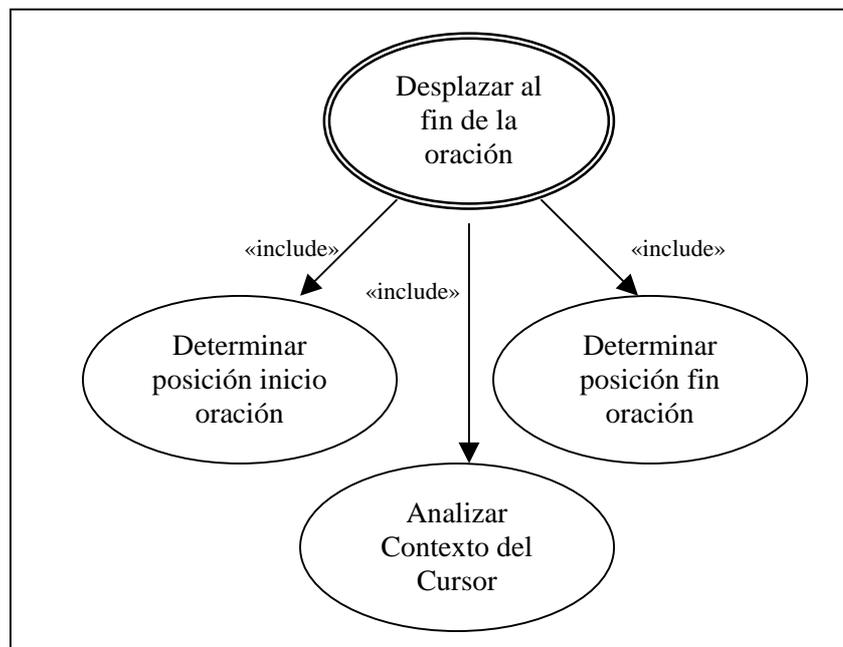
#### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.3.3. Desplazar el cursor al comienzo de la sgte. Oración	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio de la siguiente oración. (Fin de la oración actual).	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el final de la oración sobre la cual se encuentra. Los saltos de línea en el texto (carácter de Nuevo Párrafo) son tratados en sí mismos como oraciones, del mismo modo que los marcadores de bloque. Adicionalmente, si el cursor se encuentra al final del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo de la siguiente oración</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Avance (tecla de función #7), y en el menú emergente, eligiendo la opción de "Al comienzo siguiente oración" (con la tecla de función #2)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra exactamente al final de una oración, es decir, en la posición inmediatamente posterior al último carácter que la compone; el sistema determina la posición de inicio de la oración siguiente a la que allí termina.  b. Si el cursor se encuentra exactamente al principio de una oración, es decir, en la posición inmediatamente anterior al primer carácter que la compone; el sistema determina la posición de fin de la oración que allí comienza.  c. Si el cursor se encuentra en cualquier posición intermedia de una oración, determina la posición del texto en que finaliza dicha oración, junto a la posición relativa del cursor sobre la misma.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al final de la oración sobre la cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al final del texto al comenzar el caso de uso.	1.1. Como no hay ninguna oración posterior a la cual mover el cursor, simplemente se reproducen audiblemente las palabras "Fin del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso

<b>Caso de uso:</b> 5.2.3. <i>Desplazar el cursor al comienzo de la oración</i> (Continuación...)	
2. El cursor se encuentra inmediatamente antes de un salto de párrafo.	2.1. Se realiza el desplazamiento del cursor a la posición inmediato posterior al LF del Salto de Párrafo.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

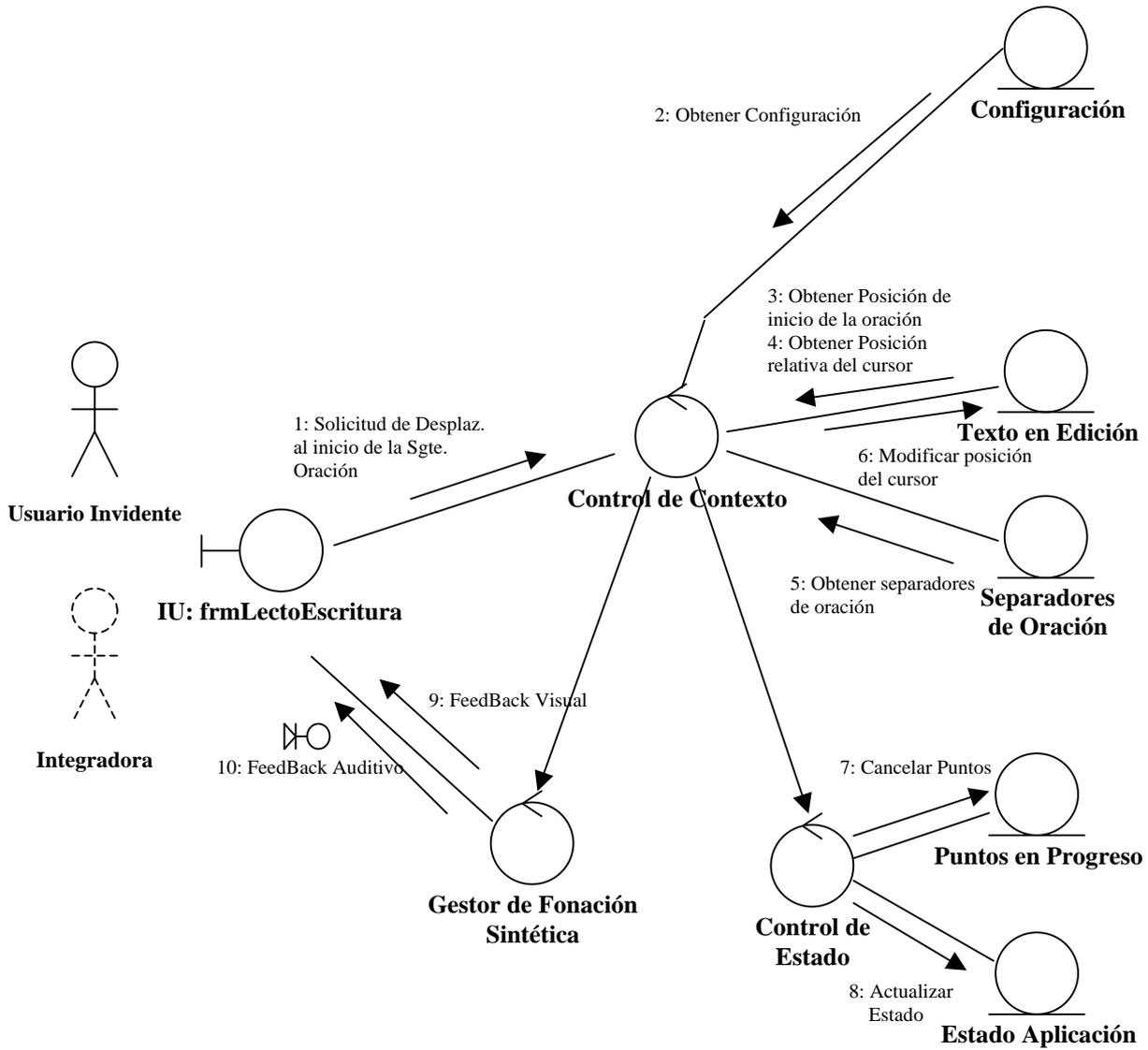
5.3.3. *Desplazar el cursor al comienzo de la sgte. oración*



Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.3.3. Desplazar el cursor al comienzo de la sgte. oración



## CU: 5.3.4. Desplazar el cursor al comienzo del sgte. párrafo

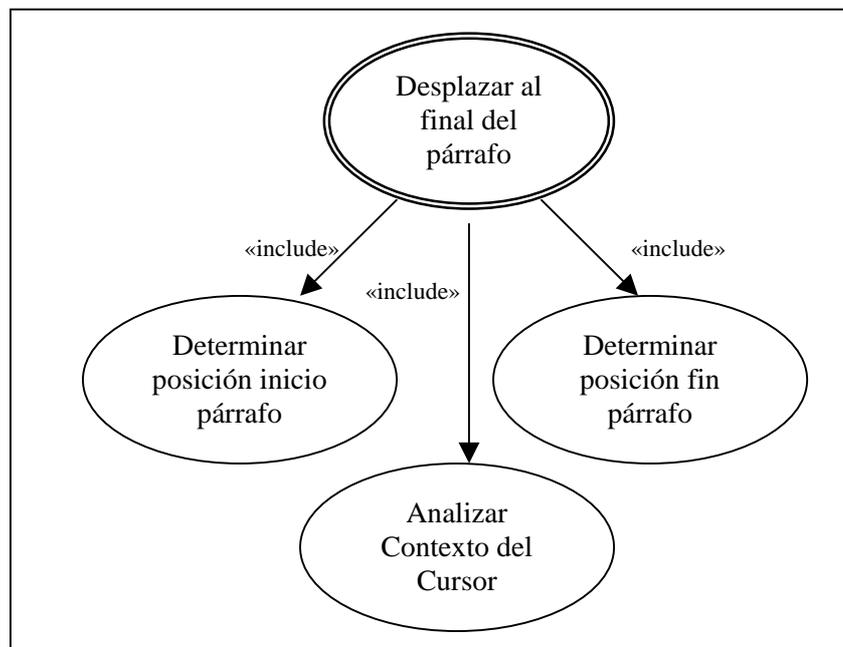
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.3.4. Desplazar el cursor al comienzo del sgte. Párrafo	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio del párrafo siguiente. (Fin del párrafo actual).	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el final del párrafo sobre el cual se encuentra. Adicionalmente, si el cursor se encuentra al final del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al final del párrafo</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Avance (tecla de función #7), y en el menú emergente, eligiendo la opción de "Al comienzo siguiente párrafo" (con la tecla de función #3)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra exactamente al final de un párrafo, es decir, en la posición inmediatamente posterior al último carácter que lo compone; el sistema determina la posición de inicio del párrafo siguiente al que allí termina.  b. Si el cursor se encuentra exactamente al principio de un párrafo, es decir, en la posición inmediatamente anterior al primer carácter que lo compone; el sistema determina la posición de fin del párrafo que allí comienza.  c. Si el cursor se encuentra en cualquier posición intermedia de un párrafo, determina la posición del texto en que finaliza dicho párrafo, junto a la posición relativa del cursor sobre el mismo.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.5. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.6. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al final del párrafo sobre el cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al final del texto al comenzar el caso de uso.	1.1. Como no hay ningún párrafo posterior al cual mover el cursor, simplemente se reproducen audiblemente las palabras "Fin del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso

<b>Caso de uso:</b> 5.2.4. <i>Desplazar el cursor al comienzo del párrafo</i> (Continuación...)	
2. El cursor se encuentra inmediatamente a antes de un salto de párrafo.	2.1. Se realiza el desplazamiento del cursor a la posición inmediato posterior al LF del Salto de Párrafo.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

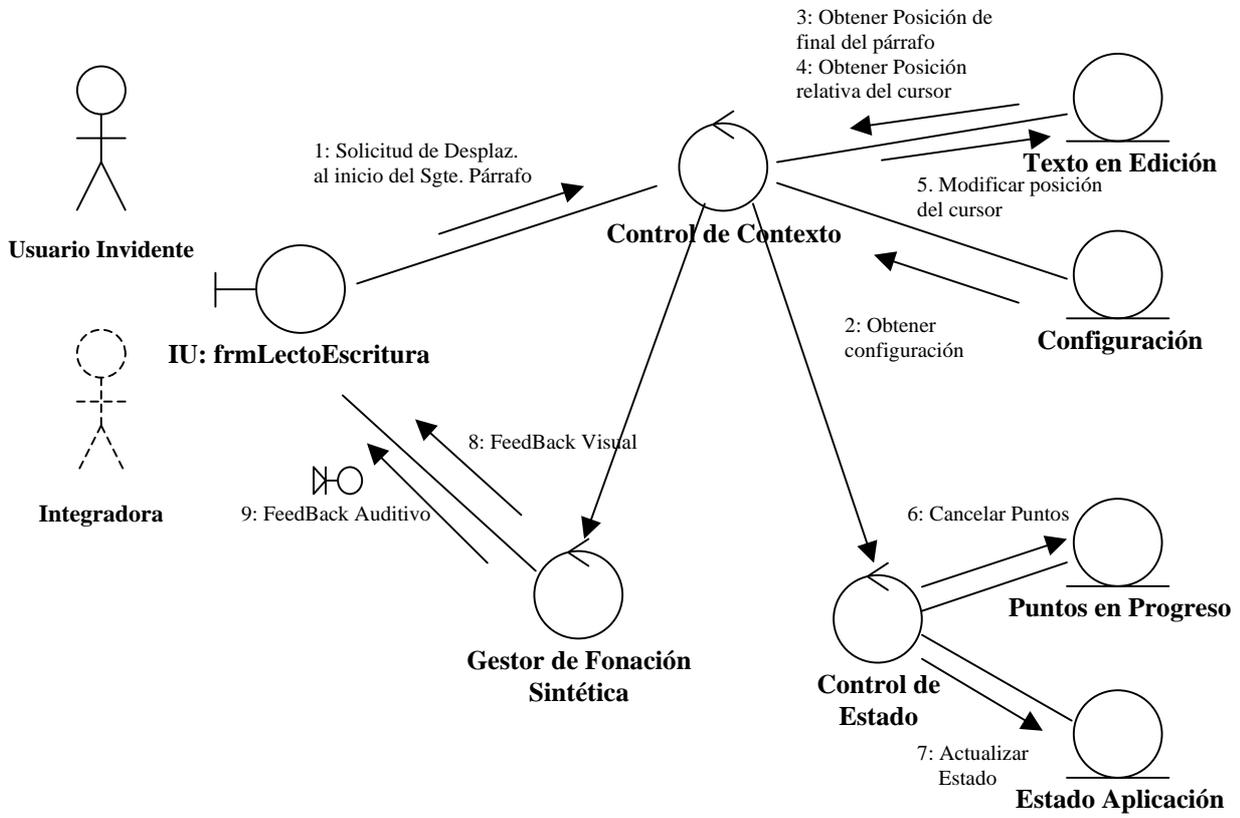
5.3.4. *Desplazar el cursor al comienzo del sgte. párrafo*



Nota: Las tres funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.3.4. Desplazar el cursor al comienzo del sgte. párrafo



## CU: 5.3.5. Desplazar el cursor hasta el marcador siguiente

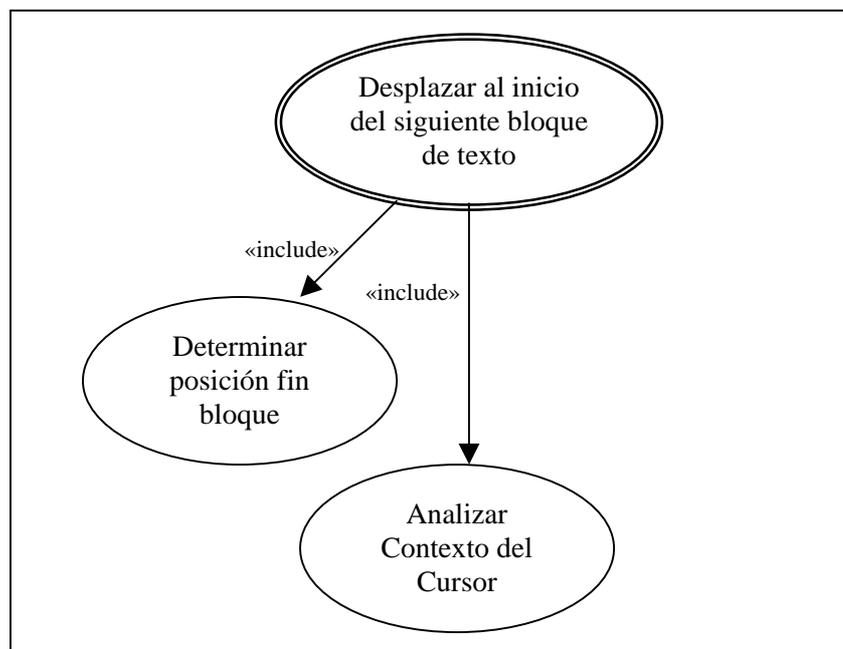
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 5.3.5. Desplazar el cursor hasta el marcador siguiente	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el inicio del Bloque de Texto actual.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hacia el inicio del Bloque de Texto siguiente a aquel en el cual se encuentra. Un bloque de texto se delimita con los marcadores de bloque: carácter especial compuesto "<*>". Adicionalmente, si el cursor se encuentra al final del texto, deberá también ser indicado al usuario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al comienzo del siguiente bloque</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Avance (tecla de función #7), y en el menú emergente, eligiendo la opción de "Hasta la siguiente marca" (con la tecla de función #6)	1.1. El sistema analiza la posición del cursor en el texto, considerando el contexto que lo rodea, según la siguiente lógica:  a. Si el cursor se encuentra exactamente al principio de un bloque, es decir, en la posición inmediatamente posterior a la marca de bloque que lo delimita; el sistema determina la posición de inicio del bloque de texto siguiente al que allí comienza.  b. Si el cursor se encuentra en una posición intermedia o exactamente al final de un bloque, es decir, en la posición inmediatamente previa a la siguiente marca de bloque; el sistema determina la posición de inicio del bloque de texto siguiente al que se encuentra.
	1.2. Se reubica el cursor en la cadena (texto en edición) en la posición de destino seleccionada.
	1.3. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.4. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.5. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La posición relativa del cursor en el texto en edición se modificó para quedar situado en la ubicación correspondiente al inicio del siguiente bloque de texto sobre el cual se encontraba. El control de la aplicación vuelve a estar en estado de espera de ingreso de puntos braille.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El cursor se encuentra al final del texto al comenzar el caso de uso.	1.1. Como no hay ningún bloque posterior al cual mover el cursor, simplemente se reproducen audiblemente las palabras "Fin del Texto", para dar a conocer la situación especial al usuario, seguida de la palabra "cursor", según configuración de la aplicación. Lo mismo se refleja en el área de texto audible, para su visualización pasiva por parte de la integradora.
	1.2. Fin del caso de uso

<b>Caso de uso:</b> 5.2.5. <i>Desplazar el cursor hasta el marcador anterior</i> (Continuación...)	
2. El cursor se encuentra en una posición intermedia entre el fin del texto y la última marca de bloque; o no existen marcas de bloque.	2.1. Se realiza el desplazamiento del cursor a la posición final del texto.
	2.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	2.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario principal de Lecto-Escritura</i>	

Diagrama de Relaciones de CU:

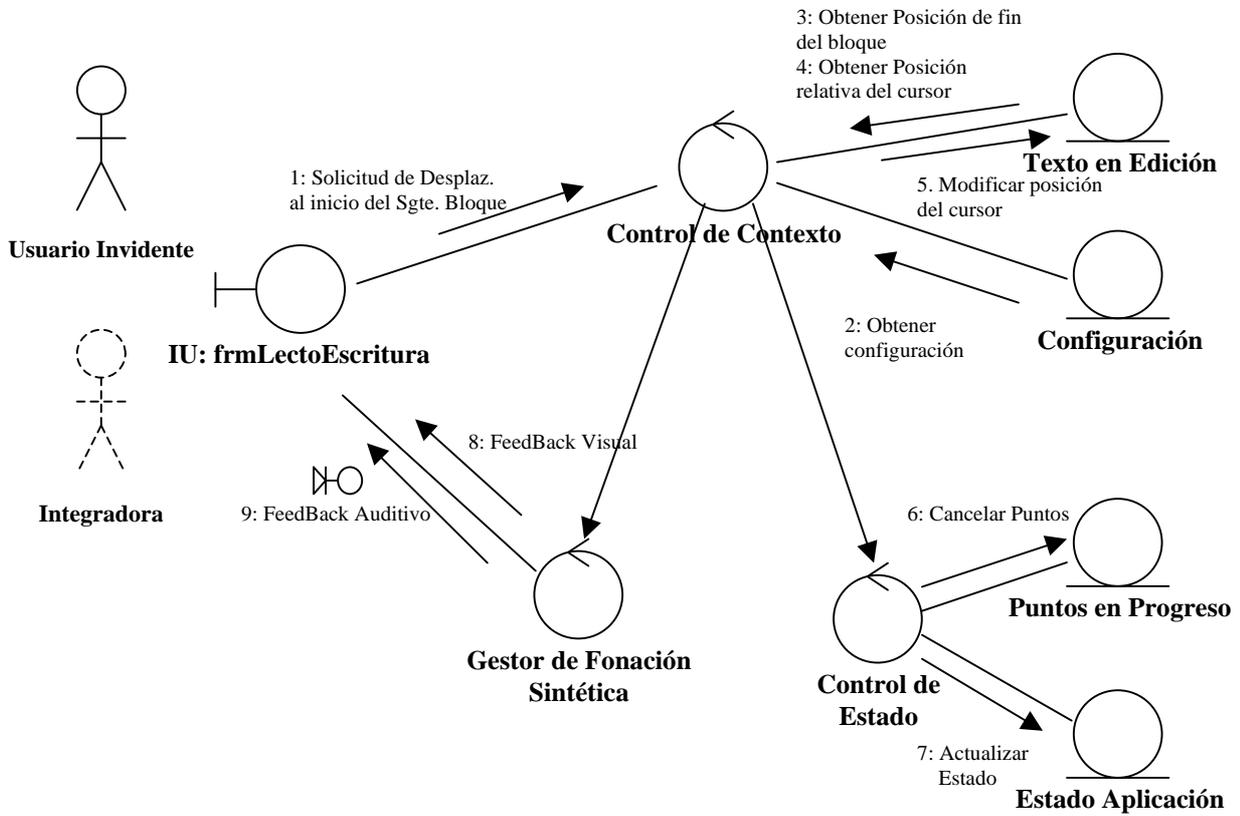
5.3.5. *Desplazar el cursor hasta el marcador siguiente*



*Nota:* Las dos funciones incluidas se destacan por su reutilización desde otros casos de uso.

Diagrama de Colaboración CU:

5.3.5. Desplazar el cursor hasta el marcador siguiente



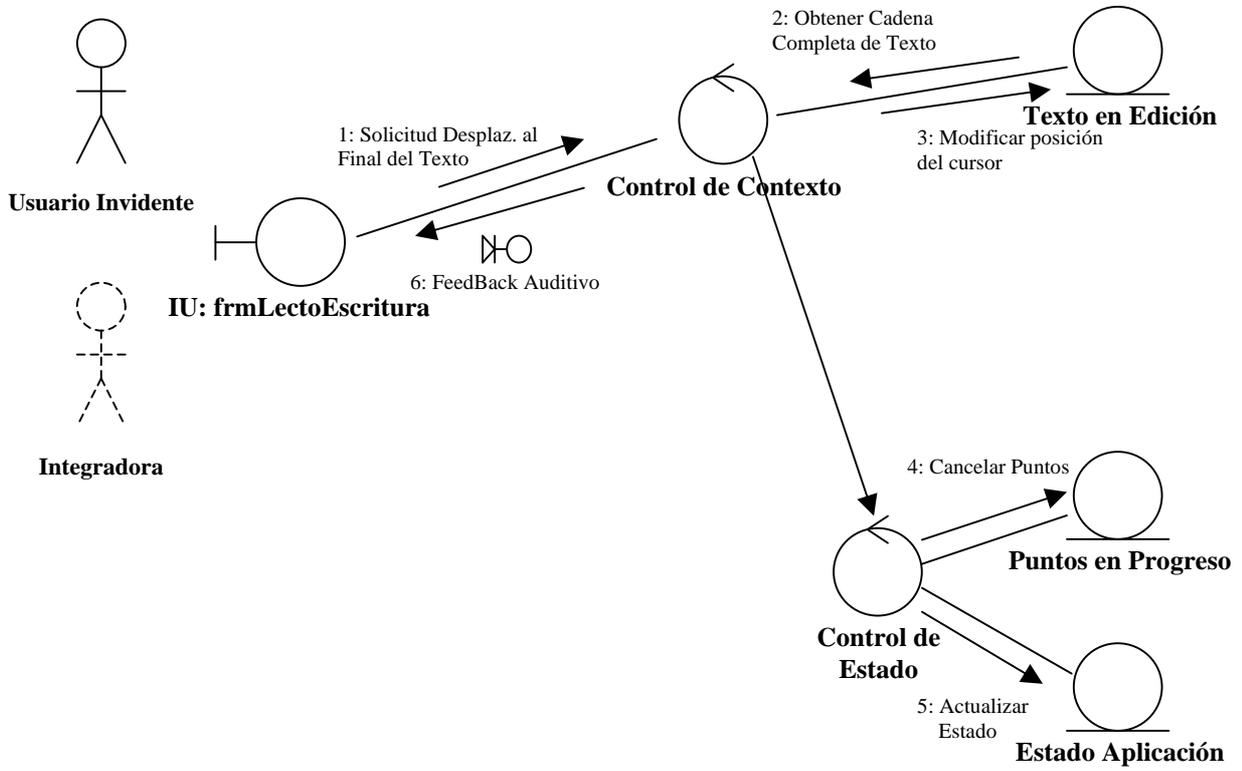
## CU: 5.3.6. Desplazar el cursor hasta el final del texto

### *Ficha de Caso de Uso*

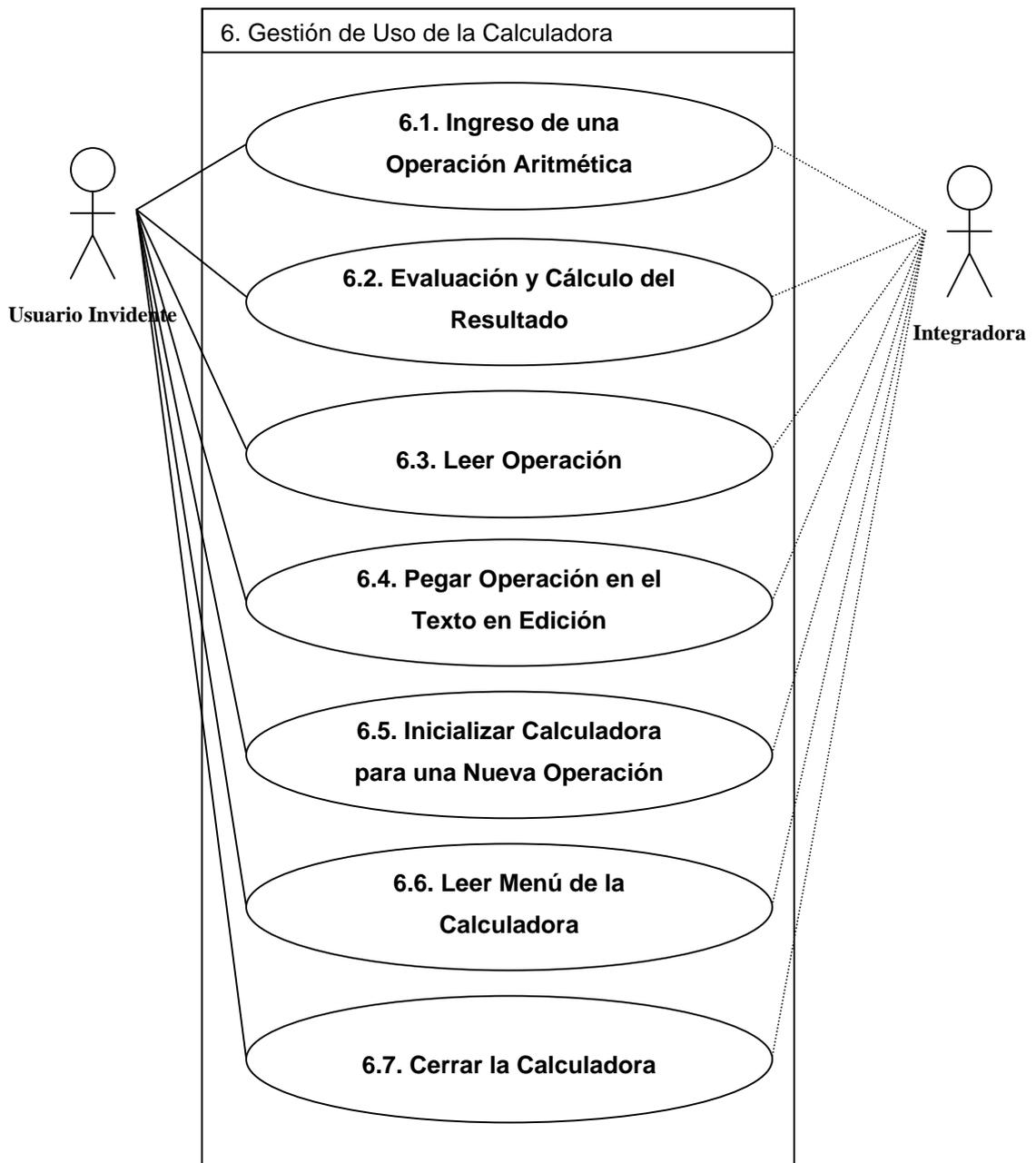
<b>Caso de uso:</b> 5.3.6. Desplazar el cursor hasta el final del texto	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Desplazar el cursor en el texto en edición hasta el final del Texto.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación modificar la ubicación relativa del cursor en el texto en edición, desplazándolo hasta el final del mismo.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del texto en edición.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario indica al sistema el comando de menú de <u>Mover el cursor al final del texto</u> . Esto se realiza seleccionando la tecla de menú de Desplazamiento en Avance (tecla de función #7), y en el menú emergente, eligiendo la opción de "Al final del texto" (con la tecla de función #4)	1.1. El cursor es ubicado en la última posición del texto.
	1.2. Se produce como feedback auditivo para el usuario un sonido característico de la acción realizada.
	1.3. El conjunto de puntos braille se inicializa para quedar en estado listo para recibir la escritura de un nuevo carácter.
	1.4. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El cursor quedó ubicado en la posición final del texto, y la aplicación con el menú normal, lista para recibir el ingreso de nuevos caracteres.	
<b>Excepciones:</b> No se aplican excepciones a este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura	

Diagrama de Colaboración CU:

5.3.6. Desplazar el cursor hasta el final del texto

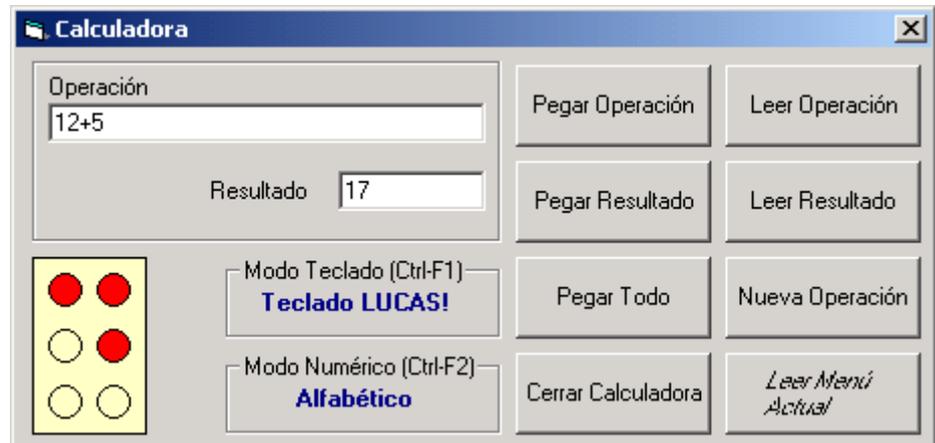


### Explosión Caso de Uso 6. Gestión de Uso de la Calculadora



Las funcionalidades soportadas por la calculadora se describen mediante la realización de los casos de uso detallados, implementando dichas características a través de una interfaz homogénea con el sistema de lecto-escritura propiamente dicho, fundamentalmente en cuanto a sus conceptos de base, su mecanismo de control (por acceso a funciones dispuestas en los botones del área derecha del tablero), por el mecanismo de comunicación auditivo implementado con su usuario, y por la similitud de la interfaz gráfica que se expone a la integradora.

La interfaz propiamente dicha fue ya detallada, pero se incluye aquí a modo de referencia próxima para la correcta comprensión e interpretación de los casos de uso desarrollados.



Cabe recordar que esta ventana se implementa “encima” de la ventana principal de la aplicación: el formulario de lecto-escritura, de manera modal a la misma.

Se desarrollan a continuación los modelos para los casos de uso identificados.

## CU: 6.1. Ingreso de una Operación Aritmética

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.1. Ingreso de una Operación Aritmética	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Ingresar una expresión aritmética a resolver, tanto operaciones simples (dos operandos con un operador) como expresiones complejas recursivas (ejercicios combinados).	
<b>Resumen:</b> Este Caso de Uso le permite al usuario principal de la aplicación ingresar al sistema la expresión matemática a evaluar, sin incluir la evaluación propiamente dicha, que como se realiza a demanda se tipifica en otros casos de uso, según requerimiento. La escritura de números y signos de operación se realiza utilizando el mismo alfabeto braille. Las únicas salvedades en ese sentido son las siguientes: <ul style="list-style-type: none"> <li>➤ Para facilitar el proceso de escritura, no se requiere el ingreso del carácter braille de número precedente a cada dígito numérico, pues de hecho no se escriben en la calculadora expresiones alfabéticas. Es decir: los caracteres braille de la "a" a la "j" son automáticamente interpretados numéricamente como los dígitos "0" a "9" respectivamente.</li> <li>➤ Adicionalmente a los dígitos numéricos, se admiten los signos específicos de operación, el punto decimal, y los paréntesis (utilizados para jerarquizar el orden de operación en expresiones complejas).</li> </ul> <p>Todo otro carácter ingresado, es rechazado. El análisis de la corrección sintáctica de la expresión ingresada y su evaluación numérica son desarrollados al momento de requerir el resultado, detallado en los correspondientes casos de uso posteriores.</p>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> el software se encuentra con el cursor posicionado en algún punto del área de texto de la calculadora.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario ingresa los puntos braille que componen el carácter que desea agregar al texto de la operación en el punto de inserción.	1.1. El sistema responde implementando para cada punto el <u>caso de uso 1.1.</u> manteniendo en memoria el conjunto de puntos hasta que sea aceptado o cancelado el carácter.
2. El usuario "acepta" el carácter ingresado, pulsando la tecla correspondiente.	1.1. El sistema responde implementando el <u>caso de uso 1.2.</u> y verificando a continuación que el carácter en cuestión sea válido, según se describió. 1.2. Si el carácter no es válido se le informa auditivamente al usuario y se inicializa el conjunto de puntos braille para que continúe con el paso 1. 1.3. Si el carácter es válido, se añade al texto en edición en el punto de inserción.
3. El usuario "cancela" el carácter que venía ingresando, pulsando la tecla correspondiente.	3.1. El sistema responde implementando el <u>caso de uso 1.3.</u> con lo que el conjunto braille queda inicializado para continuar con el ingreso de un nuevo carácter.
4. El usuario "borra" el carácter que acaba de escribir, cuando descubre que fue un error.	4.1. El sistema responde implementando el <u>caso de uso 5.1.1.</u> con lo que el carácter es removido de la cadena y el conjunto braille queda inicializado para continuar con el ingreso de un nuevo carácter.
<b>Nota sobre el escenario:</b> Las cuatro actividades señaladas se realizan de manera iterativa y recurrente hasta completar la expresión aritmética que el usuario quiere evaluar.	
<b>Poscondiciones:</b> La expresión aritmética a evaluar ha quedado ya escrita en la caja de texto correspondiente, con lo que el usuario puede proceder a su lectura y/o cálculo de resultado.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario pulsa alguna de las teclas de menú	1.1. El sistema desencadena la función asociada a la misma, según el siguiente esquema, implementando el caso de uso que corresponda en cada caso.

Esquema de funciones de menú implementadas:

- **Función#1:** Pegar Operación  
Pega en el texto en edición de la ventana principal la operación editada.
- **Función#2:** Pegar Resultado  
Pega en el texto en edición de la ventana principal el resultado de la operación editada, procediendo a su cálculo previo en caso de no disponer ya del resultado.
- **Función#3:** Pegar Todo  
Pega en el texto en edición de la ventana principal la operación editada seguida del signo "=" seguido a su vez del resultado. Al igual que la anterior, esta función desencadena a su vez el cálculo del resultado en caso de ser necesario.
- **Función#4:** Cerrar Calculadora  
Cierra la ventana de la calculadora regresando el control a la edición del texto en la ventana principal de lecto-escritura.
- **Función#5:** Leer Operación  
Lee el texto ingresado como operación, haciendo explícita la lectura de cada carácter que la compone y pronunciando los números según su lectura castellana.
- **Función#6:** Leer Resultado  
Calcula la operación introducida y verbaliza su resultado.
- **Función#7:** Nueva Operación  
Inicializa el texto de la operación en edición y el resultado, dejando la calculadora en el estado original en que se abre, para proceder a una nueva operación aritmética.
- **Función#8:** Leer Menú Actual  
Verbaliza las opciones disponibles del menú, del mismo modo que funciona en la ventana de edición de texto principal.

**Casos de uso utilizados:**

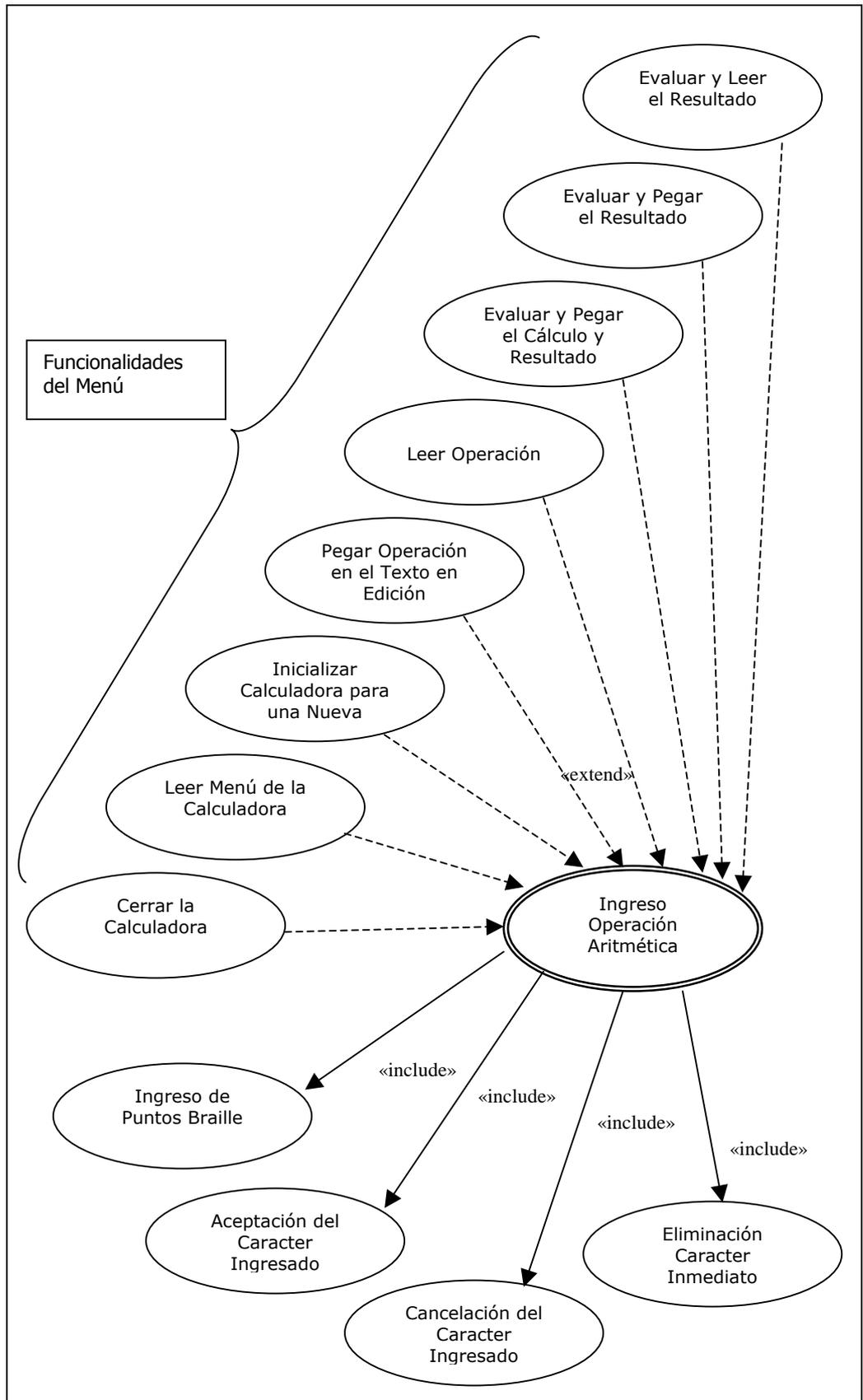
«include» C.U. 1.1. Ingreso de Puntos Braille  
«include» C.U. 1.2. Aceptación del Carácter Ingresado  
«include» C.U. 1.3. Cancelación del Carácter Ingresado  
«include» C.U. 5.1.1. Eliminación Carácter Inmediato Anterior  
«extend» C.U. 6.2.1. Evaluar y Leer el Resultado  
«extend» C.U. 6.2.2. Evaluar y Pegar el Resultado  
«extend» C.U. 6.2.3. Evaluar y Pegar el Cálculo y Resultado  
«extend» C.U. 6.3. Leer Operación  
«extend» C.U. 6.4. Pegar Operación en el Texto en Edición  
«extend» C.U. 6.5. Inicializar Calculadora para una Nueva Operación  
«extend» C.U. 6.6. Leer Menú de la Calculadora  
«extend» C.U. 6.7. Cerrar la Calculadora

**Interfaces requeridas:**

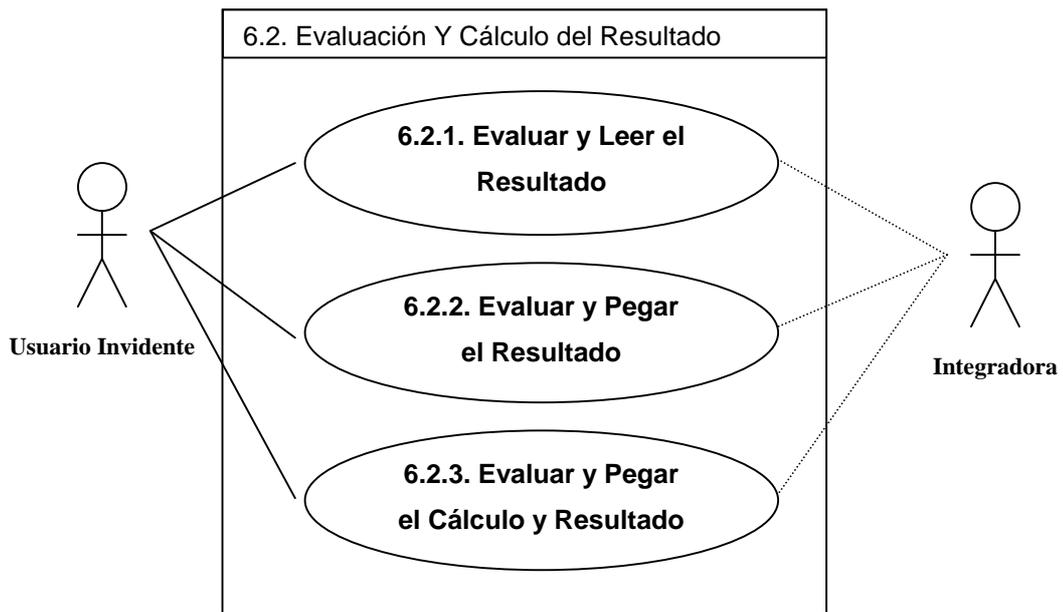
1. Formulario de Calculadora
2. Formulario principal de Lecto-Escritura (de fondo)

*Nota: Cabe señalar que las relaciones de extend con el resto de los casos de uso establecer relaciones de transferencia momentánea de control, pues en realidad dichas tareas no son parte constituyente del proceso de edición propiamente dicho (como sí sucede con los include), sino que son en sí mismos casos de uso independientes y no relacionados de modo directo al proceso de escritura de la expresión aritmética a evaluar.*

*Diagrama de Relaciones de CU: 6.1. Ingreso de una Operación Aritmética*



### **Explosión Caso de Uso 6.2. Evaluación y Cálculo del Resultado**



En lo referente a las funcionalidades de evaluación y cálculo del resultado de la expresión aritmética ingresada por el usuario, la funcionalidades previstas son las de los tres casos de uso que se detallan a continuación.

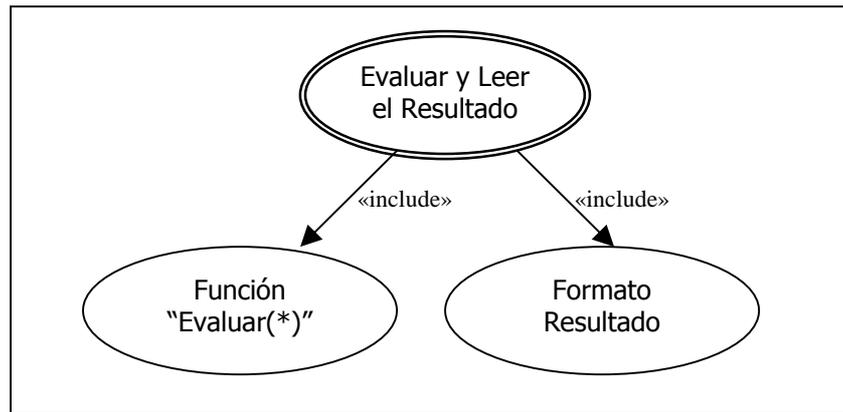
Cabe señalar como punto central de estas funciones **la implementación de un algoritmo recursivo de análisis de corrección sintáctica y cálculo de resultado para expresiones aritméticas con múltiples operaciones jerarquizadas con paréntesis**, desarrollado por el autor del presente trabajo.

## CU: 6.2.1. Evaluar y Leer el Resultado

### *Ficha de Caso de Uso*

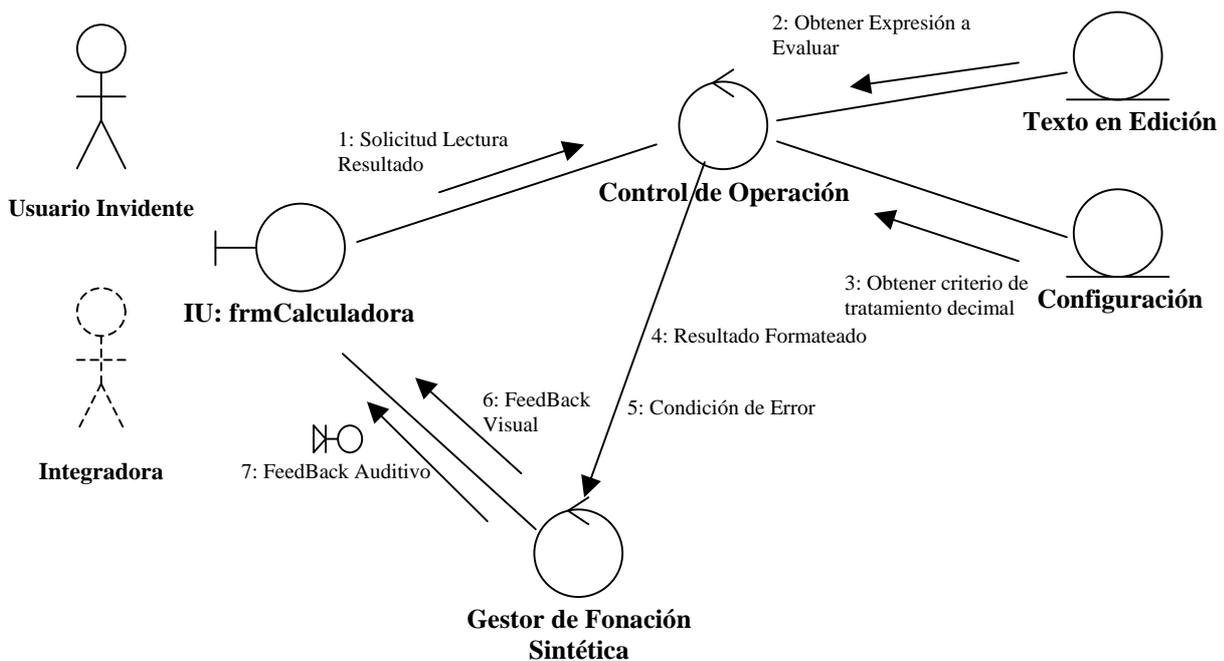
<b>Caso de uso:</b> 6.2.1. Evaluar y Leer el Resultado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Evaluar la operación aritmética ingresada y oír el resultado.	
<p><b>Resumen:</b> Este Caso de Uso permite al usuario principal de la aplicación oír el resultado de la evaluación de la operación aritmética ingresada. Si el resultado tiene parte entera y parte decimal, esta última será tratada según la configuración de la aplicación, del siguiente modo:</p> <ul style="list-style-type: none"> <li>➤ <b>Redondear:</b> sólo se entrega como resultado la parte entera, resultante de redondear los decimales.</li> <li>➤ <b>Truncar:</b> sólo se entrega como resultado la parte entera, resultante de truncar los decimales.</li> <li>➤ <b>Decimal:</b> se entrega como resultado el resultado completo, incluyendo hasta cuatro cifras decimales.</li> </ul> <p>El resultado, además de presentarse en el área de lectura (escrito en castellano), es también presentado en pantalla para su disponibilidad visual para la integradora.</p>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> la expresión a evaluar ya ha sido introducida en el área de texto correspondiente de la ventana, y aún no se ha calculado su resultado..	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. Luego de haber ingresado la fórmula a evaluar, el usuario solicita al sistema la lectura del resultado de la misma, pulsando la tecla de Función #6 del menú.	1.1. El sistema realiza la comprobación sintáctica de la expresión a evaluar.
	1.2. Se ejecuta la función recursiva de cálculo de resultado para toda la expresión.
	1.3. Se formatea el resultado según la configuración de la calculadora (Redondeando, Truncando, o con decimales)
	1.4. Se presenta el resultado en pantalla, y se devuelve como feedback auditivo al usuario.
	1.5. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La expresión aritmética ha sido evaluada y su resultado devuelto auditivamente al usuario principal, y visualmente al usuario secundario.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El usuario ingresó una expresión sintácticamente incorrecta, o con resultado infinito, y solicitó su evaluación.	1.1. La comprobación sintáctica falla, y en consecuencia se informa al usuario de manera auditiva con el mensaje "Error". Adicionalmente, se ilumina en la pantalla un indicador de Error.
2. El usuario no ingresó ninguna expresión para evaluar (cadena vacía), pero solicitó de todos modos su evaluación.	2.1. El resultado se fuerza a cero, y se informa del modo normal al usuario de manera audible y se presenta en pantalla.
<b>Casos de uso utilizados:</b> «include» Función Recursiva <b>Evaluar(*expresión*)</b>	
<b>Interfaces requeridas:</b> 1. Formulario de Calculadora	

Diagrama de Relaciones de CU: 6.2.1. Evaluar y Leer el Resultado



*Nota: La función Evaluar recibe como parámetro la cadena que contiene la expresión aritmética a evaluar. Su funcionamiento es recursivo, identificando el operador principal de la expresión y volviéndose a llamar a sí misma con cada una de las dos expresiones que son operandos del operador identificado. Finalmente, al terminar el descenso recursivo, comienzan a volver los resultados y según sea el operador que vincula los dos operandos es la operación matemática que entre aquellos se realiza, arribando así finalmente al resultado de la expresión completa a partir de sus partes.*

Diagrama de Colaboración CU: 6.2.1. Evaluar y Leer el Resultado

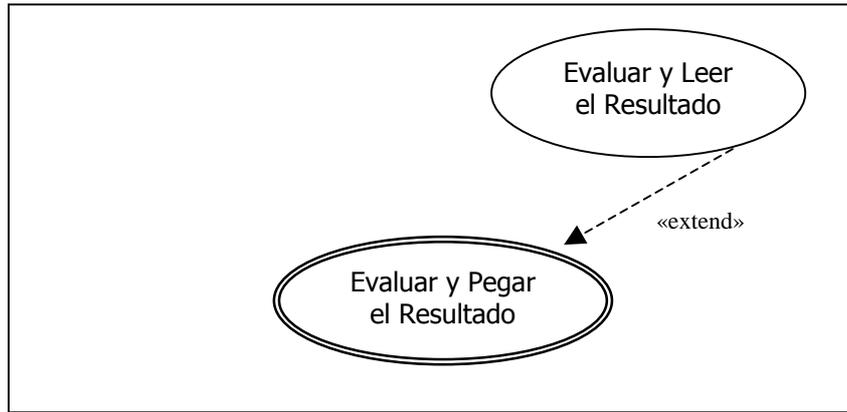


## CU: 6.2.2. Evaluar y Pegar el Resultado

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.2.2. Evaluar y Pegar el Resultado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Evaluar la operación aritmética ingresada y pegar en el texto en edición en la ventana principal de la aplicación el resultado de la expresión evaluada.	
<b>Resumen:</b> Este Caso de Uso permite al usuario principal de la aplicación oír el resultado de la evaluación de la operación aritmética ingresada, y al mismo tiempo insertar dicho resultado en la posición actual del cursor sobre el texto en edición en el formulario principal de la aplicación (ventana de Lecto-Escritura). Todo el tratamiento de la expresión aritmética y su evaluación y resultado son gestionados por el caso de uso 6.2.1. que es extendido desde este.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> la expresión a evaluar ya ha sido introducida en el área de texto correspondiente de la ventana, y aún no se ha calculado su resultado..	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. Luego de haber ingresado la fórmula a evaluar, el usuario solicita al sistema la evaluación del resultado de la misma, y la transferencia de dicho resultado al texto en edición, pulsando la tecla de Función #2 del menú.	1.1. El sistema realiza la comprobación sintáctica de la expresión a evaluar y el cálculo de su resultado, extendiendo al caso de uso 6.2.1.
	1.2. El resultado es pegado en la posición del cursor sobre el texto en edición en el formulario principal de lecto-escritura.
	1.3. La ventana de la calculadora se cierra indicando auditivamente al usuario el mensaje "Calculadora cerrada"
	1.4. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La expresión aritmética ha sido evaluada y su resultado devuelto auditivamente al usuario principal, y visualmente al usuario secundario. El resultado ha sido pegado en el texto en edición.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El usuario ingresó una expresión sintácticamente incorrecta, o con resultado infinito, y solicitó su evaluación.	1.1. Se da la alerta correspondiente (funcionalidad del C.U. 6.2.1) y no se traslada nada al texto en edición.
<b>Casos de uso utilizados:</b> «extend» C.U. 6.2.1. Evaluar y Leer el Resultado	
<b>Interfaces requeridas:</b> 1. Formulario de Calculadora 2. Formulario principal de Lecto-Escritura (de fondo)	

*Diagrama de Relaciones de CU: 6.2.2. Evaluar y Pegar el Resultado*



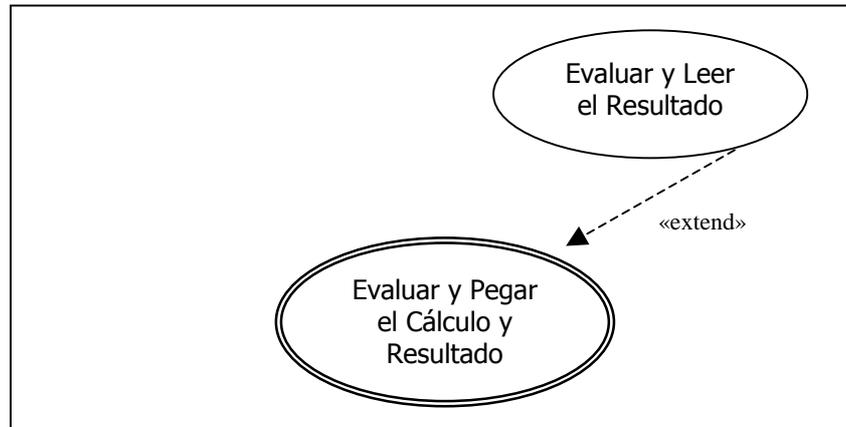
### CU: 6.2.3. Evaluar y Pegar el Cálculo y Resultado

#### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.2.3. Evaluar y Pegar el Cálculo y Resultado	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Evaluar la operación aritmética ingresada y pegar en el texto en edición en la ventana principal de la aplicación la expresión evaluada y su resultado.	
<b>Resumen:</b> Este Caso de Uso permite al usuario principal de la aplicación oír el resultado de la evaluación de la operación aritmética ingresada, y al mismo tiempo insertar dicho resultado junto a la expresión a la que responde en la posición actual del cursor sobre el texto en edición en el formulario principal de la aplicación (ventana de Lecto-Escritura). Todo el tratamiento de la expresión aritmética y su evaluación y resultado son gestionados por el caso de uso 6.2.1. que es extendido desde este. El formato con que se insertan los datos en el texto es <b>Expresión = Resultado</b>	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> la expresión a evaluar ya ha sido introducida en el área de texto correspondiente de la ventana, y aún no se ha calculado su resultado.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. Luego de haber ingresado la fórmula a evaluar, el usuario solicita al sistema la evaluación del resultado de la misma, y la transferencia de dicho resultado junto a la expresión de la operación, al texto en edición, pulsando la tecla de Función #3 del menú.	1.1. El sistema realiza la comprobación sintáctica de la expresión a evaluar y el cálculo de su resultado, extendiendo al caso de uso 6.2.1.
	1.2. La expresión evaluada y su resultado son pegados en la posición del cursor sobre el texto en edición en el formulario principal de lecto-escritura, con el formato " <b>Expresión = Resultado</b> ".
	1.3. La ventana de la calculadora se cierra indicando auditivamente al usuario el mensaje "Calculadora cerrada"
	1.4. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La expresión aritmética ha sido evaluada y su resultado devuelto auditivamente al usuario principal, y visualmente al usuario secundario. La expresión y el resultado han sido pegados en el texto en edición.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El usuario ingresó una expresión sintácticamente incorrecta, o con resultado infinito, y solicitó su evaluación.	1.1. Se da la alerta correspondiente (funcionalidad del C.U. 6.2.1) y no se traslada nada al texto en edición.
<b>Casos de uso utilizados:</b> «extend» C.U. 6.2.1. Evaluar y Leer el Resultado	
<b>Interfaces requeridas:</b> 1. Formulario de Calculadora 2. Formulario principal de Lecto-Escritura (de fondo)	

Diagrama de Relaciones de CU:

6.2.3. *Evaluar y Pegar el Cálculo y Resultado*



### CU: 6.3. Leer Operación

#### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.3. Leer Operación	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Leer la expresión introducida como operación a evaluar.	
<b>Resumen:</b> Este Caso de Uso permite al usuario oír la lectura de la expresión por él ingresada en la caja de texto de la calculadora como expresión a evaluar. Le permite corroborar si es la expresión correcta o si requiere alguna modificación.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> la expresión a evaluar – o parte de ella – ha sido introducida en el área de texto correspondiente de la ventana, y aún no se ha calculado su resultado.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la lectura de la expresión introducida como operación a ser evaluada, pulsando la tecla de Función #5 del menú.	1.1. El sistema analiza y produce la fonación de la expresión introducida en el área de texto. 1.2. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La expresión aritmética ha sido leída al usuario, y aún sigue sin haberse intentado su cálculo.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario no ingresó ninguna expresión a evaluar, y no obstante solicitó su lectura.	1.1. El sistema le informa de la situación verbalizando "No hay ninguna operación"
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. Formulario de Calculadora	

Nota: Considerando la simpleza de este caso de uso, no se justifica la producción de modelos adicionales para el mismo.

## CU: 6.4. Pegar Operación en el Texto en Edición

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.4. Pegar Operación en el Texto en Edición	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Pegar en el texto en edición la expresión introducida como operación a evaluar.	
<b>Resumen:</b> Este Caso de Uso permite al usuario trasladar al texto en edición en la ventana principal de la aplicación la expresión por él ingresada en la caja de texto de la calculadora como expresión a evaluar.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> la expresión a evaluar – o parte de ella – ha sido introducida en el área de texto correspondiente de la ventana, y aún no se ha calculado su resultado.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la transferencia de la expresión ingresada al texto en edición en la ventana principal, pulsando la tecla de Función #1 del menú.	1.1. El sistema copia en la posición actual del cursor en el texto en edición en la ventana principal de la aplicación la expresión introducida en el área de texto.
	1.2. La ventana de la calculadora se cierra indicando auditivamente al usuario el mensaje "Calculadora cerrada"
	1.3. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La expresión aritmética ha sido leída al usuario, y aún sigue sin haberse intentado su cálculo.	
<b>Excepciones:</b> No se consideran excepciones para este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario de Calculadora	
2. Formulario principal de Lecto-Escritura (de fondo)	

*Nota: Considerando la simpleza de este caso de uso, no se justifica la producción de modelos adicionales para el mismo.*

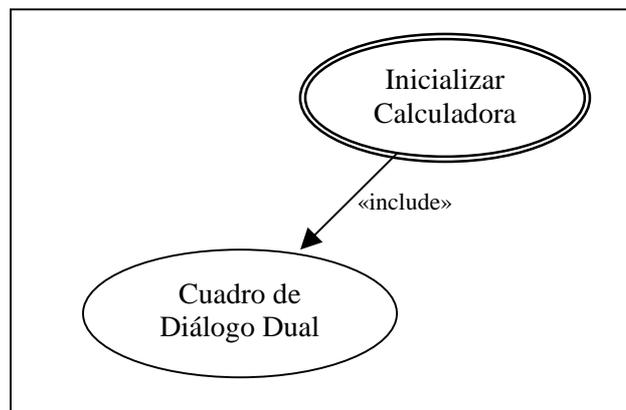
## CU: 6.5. Inicializar Calculadora para una Nueva Operación

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.5. Inicializar Calculadora para una Nueva Operación	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Inicializar la Calculadora, reestableciendo tanto el texto de la operación a realizar como el resultado a su estado inicial: vacío.	
<b>Resumen:</b> Este Caso de Uso cancela todas las operaciones pendientes con la calculadora, dejándola en estado inicial.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> Expresión y/o resultados calculados y disponibles.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la reinicialización de la calculadora para disponer el ingreso de un nuevo cálculo, pulsando la tecla de Función #7 del menú.	1.1. El sistema solicita confirmación al usuario, a través de la implementación de un cuadro de diálogo modal de interfaz dual: audible y visible.
	1.2. En caso que la operación haya sido confirmada, se inicializan ambas cajas de texto y la calculadora queda lista para la realización de un nuevo cálculo.
	1.3. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> La calculadora queda en estado inicial, sin texto en el área de ingreso de las operaciones a realizar, ni resultados disponibles..	
<b>Excepciones:</b> No se consideran excepciones para este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario de Calculadora	

### *Diagrama de Relaciones de CU:*

#### *6.5. Inicializar Calculadora para una Nueva Operación*



## CU: 6.6. Leer Menú de la Calculadora

### *Ficha de Caso de Uso*

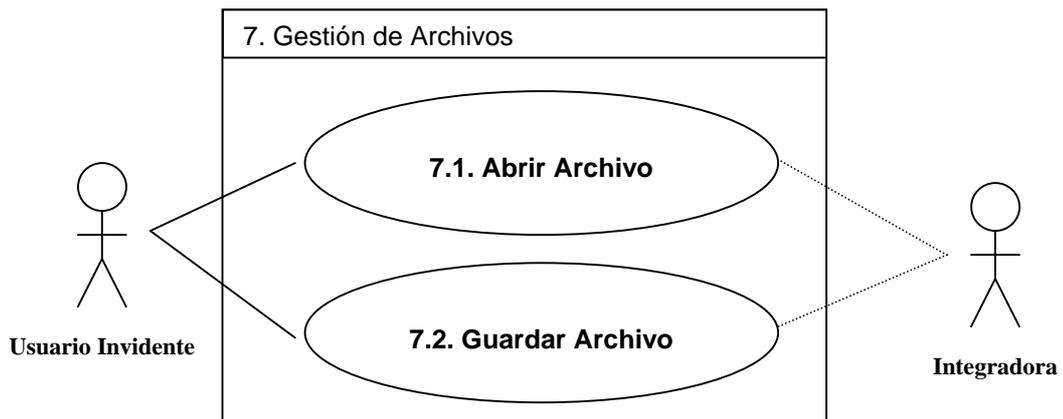
<b>Caso de uso:</b> 6.6. Leer Menú de la Calculadora	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Reproducir audiblemente las opciones de menú disponibles en la calculadora.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario recordar qué funciones de la calculadora están asignadas a cada una de las teclas de control. La configuración de menú es verbalizada posibilitando su reconocimiento por parte del usuario no videntes.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> «no relevante»	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la lectura de las funciones disponibles en el menú, pulsando la tecla de Función #8.	1.1. El sistema verbaliza las funciones disponibles y el botón al que cada una de las mismas está asociada. 1.2. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> «no relevante»	
<b>Excepciones:</b> No se consideran excepciones para este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. Formulario de Calculadora	

## CU: 6.7. Cerrar la Calculadora

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 6.7. Cerrar la Calculadora	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Finalizar el trabajo con la calculadora, cerrando el formulario modal de la misma.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario cerrar la ventana en que corre la calculadora, volviendo a tomar foco sobre el formulario de lecto-escritura subyacente.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> «no relevante»	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema el cierre de la calculadora, pulsando la tecla de Función #4.	1.1. El sistema cierra la ventana de la calculadora, devolviendo el foco a la ventana de lecto-escritura que está por detrás.
	1.2. El sistema informa auditivamente al usuario que la calculadora ha sido cerrada.
	1.3. La aplicación pone foco sobre el formulario de lecto-escritura, estableciendo el cursor en la posición en que se encontraba antes de activar la calculadora.
	1.4. Fin del caso de uso
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> «no relevante»	
<b>Excepciones:</b> No se consideran excepciones para este caso de uso.	
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario de Calculadora	
2. Formulario principal de Lecto-Escritura (de fondo)	

### **Explosión Caso de Uso 7. Gestión de Archivos**



Las funcionalidades de abrir y guardar archivos se implementan de dos modos diferentes:

- La interfaz auditiva le permite al usuario seleccionar el archivo que quiere abrir utilizando las teclas de función. Para guardar, simplemente se salva el archivo en curso.
- La interfaz gráfica, en cambio, se implementa con los cuadros de diálogo Standard de Windows de “Abrir...” y “Guardar como...” para su utilización por parte de la integradora.

Se detallan a continuación los casos de uso respectivos.

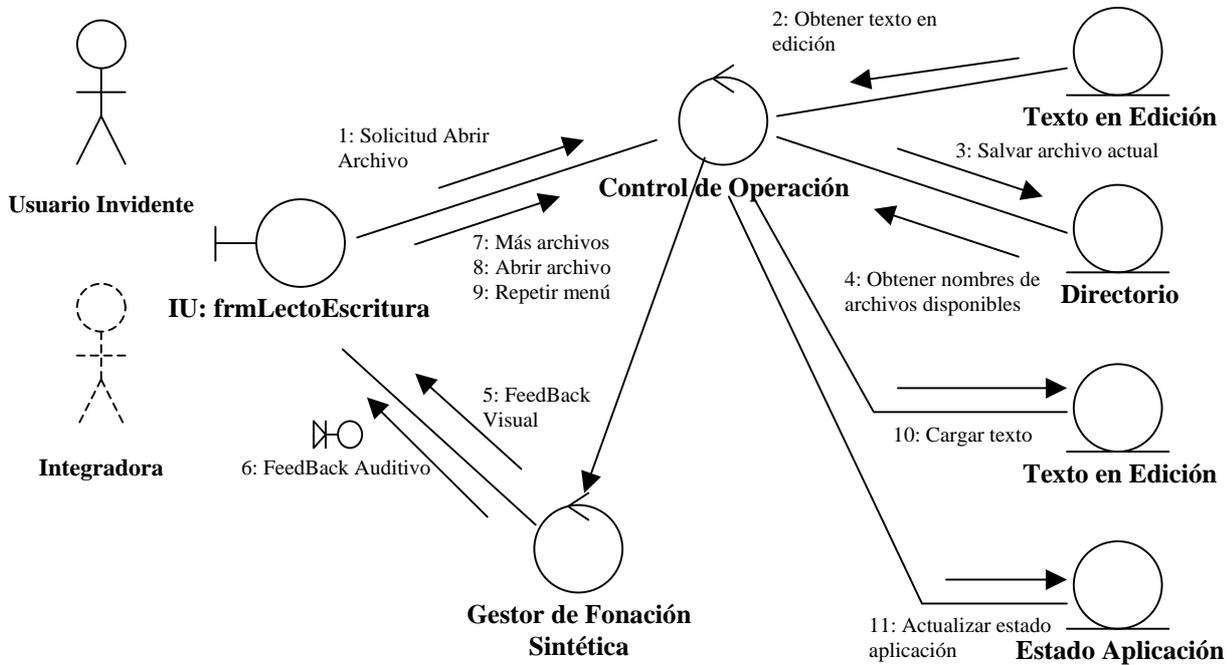
## CU: 7.1. Abrir Archivo

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 7.1.a. Abrir Archivo (función de interfaz auditiva para el usuario no vidente)	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Permitir abrir un archivo de texto previamente salvado para trabajar sobre su edición.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario seleccionar el archivo que desea abrir y proceder a su apertura, salvando previamente los cambios realizados sobre el archivo en edición.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra con un archivo en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la función de apertura de un archivo de trabajo. Esto se realiza seleccionando la tecla de menú de gestión de archivos (tecla de función #4), y en el menú emergente, eligiendo la opción de "Abrir Archivo" (con la tecla de función #1)	1.1. Se graba al disco el archivo actualmente en edición.
	1.2. Se recorre el directorio actual, escaneando los archivos de texto disponibles, y se presentan los primeros seis asociándolos respectivamente a los botones de Función #1 a #6.
	1.3. Se verbalizan sus nombres y la tecla de función a la que quedaron asociados, para que el usuario pueda elegir cuál abrir. También se indica que la tecla de función #7 permite avanzar a los siguientes archivos, con el mensaje "Función 7: Más archivos", sólo en el caso de que existan más entradas de directorio válidas. Finalmente se indica que la tecla de función #8, al igual que ocurre en toda la aplicación, permite repetir las opciones de menú (Leer menú actual).
2. El usuario elige el archivo a abrir, pulsando la tecla de función que le corresponde (de la #1 a la #6)	2.1. El sistema abre el archivo elegido y lo carga en el área de edición.
	2.2. El cursor se posiciona al principio del archivo.
	2.3. Se reestablece el menú de la aplicación a su estado normal.
	2.4. Fin del caso de uso.
3. El usuario opta por continuar con los siguientes archivos disponibles, presionando la tecla de función #7.	3.1. El sistema escanea las siguientes entradas de directorio válida, implementando la funcionalidad descrita en el paso 1.2 en adelante.
4. El usuario solicita oír nuevamente las opciones disponibles, presionando la tecla de función #8.	4.1. El sistema vuelve a presentar auditivamente las opciones de menú disponibles.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El archivo que se encontraba anteriormente en edición se guardó actualizado en el disco, y el archivo elegido por el usuario se cargó en reemplazo de aquel para su edición. El cursor se encuentra posicionado al principio del mismo.	

<b>Caso de uso:</b> 7.1.a. Abrir Archivo (función de interfaz auditiva) (Continuación...)	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<b>Eventos (del Actor o Usuario) / Estado del Sistema</b>	<b>Respuesta del Sistema</b>
1. El usuario decide cancelar la operación de apertura, presionando la tecla de cancelación en su tablero.	1.1. Se restaura el menú normal de la aplicación, indicando auditivamente la cancelación producida.
	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. Formulario principal de Lecto-Escritura (de fondo)	

*Diagrama de Colaboración CU: 7.1. Abrir Archivo*



### Ficha de Caso de Uso

<b>Caso de uso:</b> 7.1.b. Abrir Archivo (función de interfaz visual)	
<b>Actor Principal:</b> Asistente Pedagógica Integradora	
<b>Actor Secundario:</b> «no»	
<b>Objetivos / propósito:</b> Permitir abrir un archivo de texto previamente salvado para trabajar sobre su edición.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario seleccionar el archivo que desea abrir y proceder a su apertura, salvando previamente los cambios realizados sobre el archivo en edición.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra con un archivo en edición.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la función de apertura de un archivo de trabajo. Esto se realiza haciendo click en el botón de comando para tal función al pie de la interfaz gráfica.	1.1. Se graba al disco el archivo actualmente en edición.
	1.2. Se presenta el cuadro de diálogo estándar de "Abrir archivo..." filtrando exclusivamente archivos de texto.
2. El usuario elige un archivo a abrir, y acepta su apertura en el cuadro de diálogo.	2.1. El sistema abre el archivo elegido y lo carga en el área de edición.
	2.2. El cursor se posiciona al principio del archivo.
	2.4. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El archivo que se encontraba anteriormente en edición se guardó actualizado en el disco, y el archivo elegido por el usuario se cargó en reemplazo de aquel para su edición. El cursor se encuentra posicionado al principio del mismo.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario decide cancelar la operación de apertura, presionando el botón cancelar en el cuadro de diálogo.	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura (de fondo)	

## CU: 7.2. Guardar Archivo

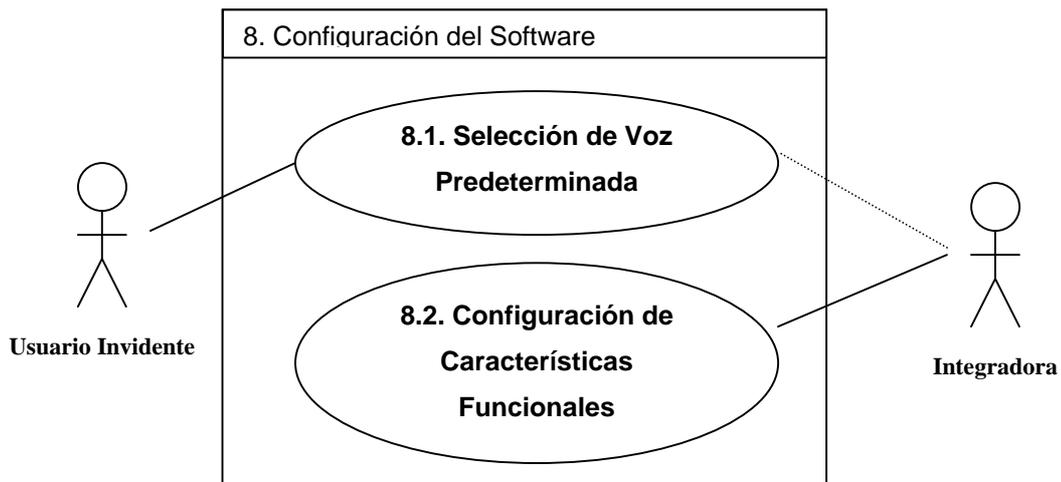
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 7.2.a. Guardar Archivo (función de interfaz auditiva para el usuario no vidente)	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Permitir grabar al disco el archivo de texto actualmente en edición.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario grabar en el disco duro el archivo que se encuentra actualmente en edición, bajo el nombre de archivo ya establecido al momento de haberlo abierto.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra con un archivo en edición, sin certeza de que el mismo esté actualizado en el disco.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la función de grabación del archivo en edición. Esto se realiza seleccionando la tecla de menú de gestión de archivos (tecla de función #4), y en el menú emergente, eligiendo la opción de "Guardar Archivo" (con la tecla de función #2)	1.1. Se graba al disco el archivo actualmente en edición.
	2.3. Se reestablece el menú de la aplicación a su estado normal.
	2.4. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El archivo que se encontraba en edición se guardó actualizado en el disco.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario decide cancelar la operación de grabación, presionando la tecla de cancelación en su tablero.	1.1. Se restaura el menú normal de la aplicación, indicando auditivamente la cancelación producida.
	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura (de fondo)	

### Ficha de Caso de Uso

<b>Caso de uso:</b> 7.2.b. Guardar Archivo (función de interfaz visual)	
<b>Actor Principal:</b> Asistente Pedagógica Integradora	
<b>Actor Secundario:</b> «no»	
<b>Objetivos / propósito:</b> Permitir guardar en un archivo el texto que se encuentra actualmente edición.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario elegir o establecer el nombre del archivo en que grabará el texto actualmente en edición, procediendo a la grabación propiamente dicha en caso de confirmar la operación.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra con un archivo en edición, sin certeza de que el mismo esté actualizado en el disco.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. El usuario solicita al sistema la función de grabación del archivo de trabajo. Esto se realiza haciendo click en el botón de comando para tal función al pie de la interfaz gráfica.	1.1. Se presenta el cuadro de diálogo estándar de "Guardar como..." filtrando exclusivamente archivos de texto.
	1.2. Se graba con el nombre de archivo y ubicación especificados el texto actualmente en edición, con formato de archivo ASCII plano. Si el archivo seleccionado ya existía, se sobrescribe con el contenido actual. Si no existía, se crea.
	1.3. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El archivo que se encontraba anteriormente en edición se guardó actualizado en el disco, con el nombre y ubicación elegido por el usuario.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El usuario decide cancelar la operación de grabación, presionando el botón cancelar en el cuadro de diálogo.	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura (de fondo)	

### **Explosión Caso de Uso 8. Configuración del Software**



En lo referente a las funcionalidades de configuración del software, dada la complejidad de dichas configuraciones y la pertinencia terapéutico-pedagógica de las mismas, sólo se ofrece al usuario principal de la aplicación la posibilidad de elegir con qué voz quiere que el software se comunique con él, pudiendo optar de entre las configuraciones de voces predeterminadas.

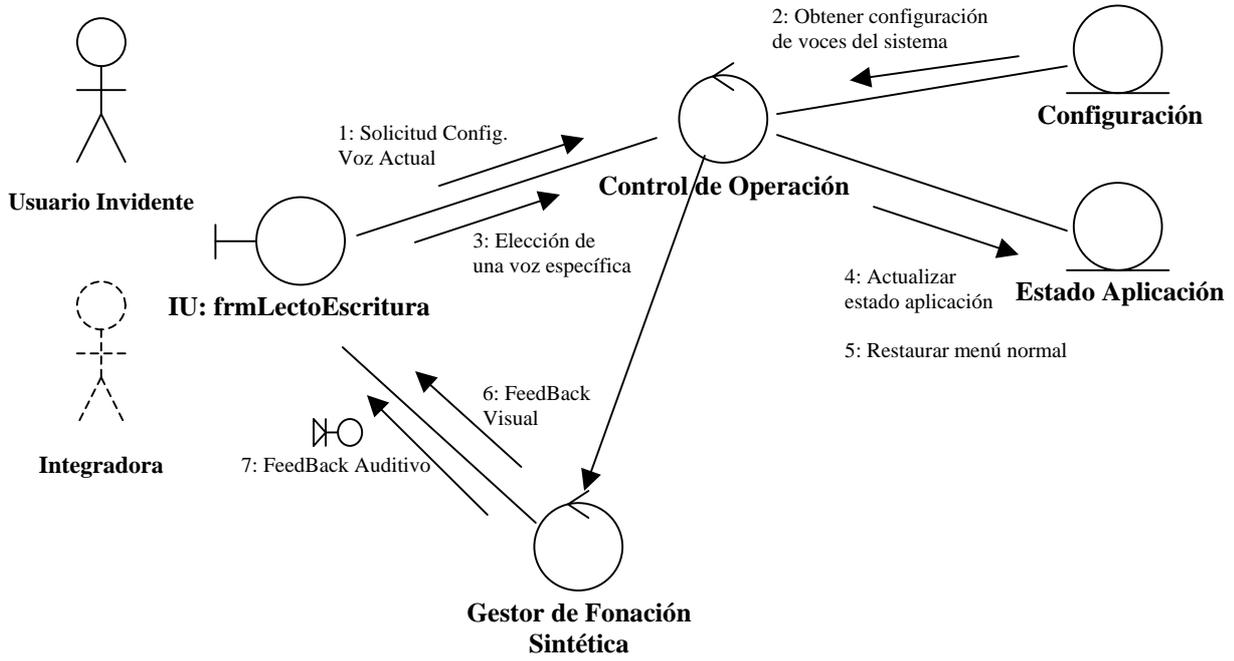
Por su parte, la integradora o asistente pedagógica del usuario, tiene la posibilidad de configurar los diferentes aspectos de comportamiento e interfaz del software a los efectos de ir produciendo las adaptaciones necesarias conforme los estadios académicos en que se va situando el usuario, y la destreza que va adquiriendo en el uso de la herramienta.

## CU: 8.1. Selección de Voz Predeterminada

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 8.1. Selección de Voz Predeterminada	
<b>Actor Principal:</b> Usuario No Vidente	
<b>Actor Secundario:</b> Asistente Pedagógica Integradora	
<b>Objetivos / propósito:</b> Permitir elegir y establecer la voz con la que se comunicará la aplicación con el usuario..	
<b>Resumen:</b> Este Caso de Uso le permite al usuario seleccionar de entre las siete voces preconfiguradas en el sistema, con cuál de ellas trabajar.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra con una determinada voz configurada.	
<b>Escenario</b>	
Eventos (del Actor o Usuario)	Respuesta del Sistema
1. El usuario solicita al sistema la función de elección de la voz de la aplicación. Esto se realiza seleccionando la tecla de menú de control (tecla de función #1), y en el menú emergente, eligiendo la opción de "Elegir la voz actual" (con la tecla de función #1)	1.1. Se presenta al usuario mediante la interfaz auditiva el menú de posibles voces preconfiguradas, asignando cada una de las siete voces a las respectivas teclas de función #1 a #7. La tecla de función #8 mantiene, como en el resto de la aplicación, la funcionalidad de leer el menú actual.
2. El usuario elige la configuración de voz a establecer como activa, pulsando la tecla de función que le corresponda (de la #1 a la #7)	2.1. El sistema reconfigura las características vocales de la fonación sintética en función de la elección hecha por el usuario y la configuración de las voces preestablecida.
	2.2. Se informa auditivamente al usuario, utilizando la nueva voz seleccionada, que la operación tuvo éxito mediante el mensaje "Nueva voz establecida".
	2.3. Se reestablece el menú de la aplicación a su estado normal.
	2.4. Fin del caso de uso.
3. El usuario solicita oír nuevamente las opciones disponibles, presionando la tecla de función #8.	3.1. El sistema vuelve a presentar auditivamente las opciones de menú disponibles, como se detalló en 1.1.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El sistema quedó reconfigurado para utilizar la voz elegida por el usuario para todas las comunicaciones verbales hacia él.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
Eventos (del Actor o Usuario) / Estado del Sistema	Respuesta del Sistema
1. El usuario decide cancelar la operación de configuración, presionando la tecla de cancelación en su tablero.	1.1. Se restaura el menú normal de la aplicación, indicando auditivamente la cancelación producida.
	1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura (de fondo)	

*Diagrama de Colaboración CU: 8.1. Selección de Voz Predeterminada*



## CU: 8.2. Configuración de Características Funcionales

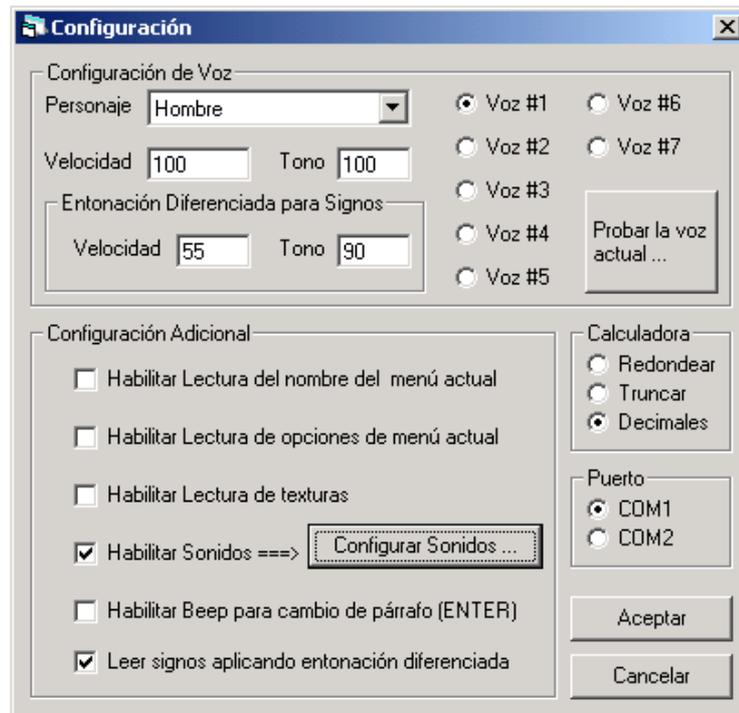
### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 8.2. Configuración de Características Funcionales	
<b>Actor Principal:</b> Asistente Pedagógica Integradora	
<b>Actor Secundario:</b> «no»	
<b>Objetivos / propósito:</b> Establecer el valor de cada una de las características configurables de la aplicación, controlando el funcionamiento de la misma según necesidades del usuario principal de la herramienta.	
<b>Resumen:</b> Este Caso de Uso le permite al usuario establecer el modo de funcionamiento elegido para cada una de las características configurables de la aplicación, según el siguiente detalle:	
<ul style="list-style-type: none"> <li>➤ <u>Configuración de la voz de la aplicación</u> Permite configurar 7 (siete) voces diferentes, estableciendo para cada una de ellas las siguientes variables: <ul style="list-style-type: none"> <li>• Personaje.</li> <li>• Velocidad de pronunciación.</li> <li>• Tono.</li> <li>• Entonación diferenciada para la lectura de los signos de puntuación, especificando la velocidad y tono particulares.</li> </ul> ** Cada voz puede probarse, escuchando un texto predefinido según las características de fonación establecidas. </li> <li>➤ Configuración del tratamiento de decimales en la calculadora <ul style="list-style-type: none"> <li>• Redondear decimales</li> <li>• Truncar decimales</li> <li>• Presentar decimales</li> </ul> </li> <li>➤ Configuración de conexión del tablero especial con el software <ul style="list-style-type: none"> <li>• Puerto COM a utilizar (COM1 o COM2)</li> </ul> </li> <li>➤ Configuración de características de feedback auditivo <ul style="list-style-type: none"> <li>• Habilitar lectura del nombre del menú actual al momento de establecerlo</li> <li>• Habilitar lectura de las opciones disponibles en el menú actual al momento de establecerlo</li> <li>• Habilitar lectura de texturas al presionar los botones</li> <li>• Habilitar reproducción de sonidos para la pulsión de los botones. Se especifica el archivo a reproducir para cada botón también como aspecto configurable.</li> <li>• Reproducir un Beep para cada cambio de párrafo en la lectura</li> <li>• Leer signos de puntuación utilizando la entonación diferenciada establecida</li> </ul> </li> </ul>	
** Se adjuntan prototipos de interfaz para implementación de este caso de uso.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El sistema se encuentra funcionando según la configuración vigente.	
<b>Escenario</b>	
<b>Eventos (del Actor o Usuario)</b>	<b>Respuesta del Sistema</b>
1. El usuario abre el formulario de configuración mediante el menú principal de la aplicación: interfaz MDI.	1.1. Se presenta la pantalla de configuración con los parámetros actuales del sistema, permitiendo al usuario editar/modificar el contenido de todos los campos.
2. El usuario selecciona la opción de "Habilitar sonidos" y hace click en el botón de "Configurar sonidos"	2.1. Se abre de manera modal la ventana de configuración de sonidos para cada tecla del tablero, permitiéndole editar / seleccionar / testear (oír) cada sonido configurado.
2.1. El usuario confirma las modificaciones realizadas en los parámetros de sonidos.	2.1.1. Se graba en el archivo de configuración las rutas establecidas para cada sonido.
	2.1.2. Se cierra el formulario de configuración de sonidos, devolviendo el foco a la pantalla de configuración.

<b>Caso de uso: 8.2. Configuración de Características Funcionales</b> <span style="float: right;"><i>(Continuación...)</i></span>	
3. El usuario confirma las modificaciones realizadas en los parámetros de configuración, haciendo click en el botón "Aceptar".	3.1. Se graba en el archivo de configuración de la aplicación el nuevo perfil de parámetros, para hacerlo permanente.
	3.2. Se cierra el formulario de configuración, devolviendo el foco a la pantalla principal de la aplicación.
	3.3. Fin del caso de uso.
<b>Nota sobre el escenario:</b> <i>Sin comentarios adicionales. Ver excepciones.</i>	
<b>Poscondiciones:</b> El sistema queda en funcionamiento reconfigurado según los parámetros gestionados en el caso de uso.	
<b>Excepciones:</b> <i>Las excepciones consideradas son las siguientes</i>	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
2.2. El usuario cancela (aborta) las modificaciones realizadas en los parámetros de sonidos.	2.1.2. Se cierra el formulario de configuración de sonidos sin grabar las modificaciones realizadas, devolviendo el foco a la pantalla de configuración.
4. El usuario cancela (aborta) las modificaciones realizadas en los parámetros de configuración, haciendo click en el botón "Cancelar".	4.1. Se cierra el formulario de configuración sin grabar las modificaciones realizadas en los parámetros, devolviendo el foco a la pantalla principal de la aplicación.
	4.2. Fin del Caso de Uso.
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b> 1. <i>Formulario General de configuración de la aplicación</i> 2. <i>Formulario de configuración de sonidos de Feedback</i>	

## *Interfaz de Usuario:8.2. Configuración de Características Funcionales*

Esta interfaz es disparada desde la opción Configuración del menú principal de la pantalla de la aplicación:



Al pulsar el botón “Configurar Sonidos...” se presenta de manera modal a esta ventana la siguiente, con la que se configuran los sonidos a reproducir para la pulsión de cada tecla del tablero.



Cada archivo de sonido puede elegirse y probarse con los botones correspondientes.

## CU: 9. Autograbación del Texto en Edición

### *Ficha de Caso de Uso*

<b>Caso de uso:</b> 9. Autograbación del Texto en Edición	
<b>Actor Principal:</b> Reloj del Sistema (CPU Clock)	
<b>Actor Secundario:</b> «no»	
<b>Objetivos / propósito:</b> Salvar en un archivo no volátil el texto que se encuentra en edición, a intervalos regulares de tiempo.	
<b>Resumen:</b> Este Caso de Uso se ejecuta automáticamente a intervalos regulares de tiempo (1 minuto por default) produciendo en cada ciclo la grabación automática y sin intervención del usuario del texto que se encuentra actualmente en edición. Dicho texto se graba en el archivo "Texto.dat" ubicado en el mismo directorio que la aplicación, y puede ser editado externamente de ser necesario.	
<b>Requisitos:</b> «ninguno»	
<b>Precondiciones:</b> El texto en edición puede haberse modificado sin que se salven los cambios.	
<b>Escenario</b>	
<i>Eventos (del Actor o Usuario)</i>	<b>Respuesta del Sistema</b>
1. Transcurrió el intervalo definido por la autoejecución de este caso de uso (por default 1 minuto)	1.1. El texto que se encuentra actualmente en edición es grabado con formato de archivo ASCII plano en "Texto.dat", en el mismo directorio de la aplicación. Si el archivo existe, se sobrescribe. Si no existe, se crea. 1.2. Fin del caso de uso.
<b>Nota sobre el escenario:</b> Sin comentarios adicionales. Ver excepciones.	
<b>Poscondiciones:</b> El texto en edición se salvó para su posterior recuperación, al momento de reiniciar el programa, en el archivo especificado.	
<b>Excepciones:</b> Las excepciones consideradas son las siguientes	
<i>Eventos (del Actor o Usuario) / Estado del Sistema</i>	<b>Respuesta del Sistema</b>
1. El archivo de destino está ocupado por otro proceso que lo mantiene abierto.	1.1. Se presenta un mensaje de error al usuario y se detiene la aplicación. 1.2. Fin del caso de uso
<b>Casos de uso utilizados:</b> «ninguno»	
<b>Interfaces requeridas:</b>	
1. Formulario principal de Lecto-Escritura (de fondo)	

#### 13.4.4. Aspectos de Modelado e Implementación de Objetos de Datos

Por la naturaleza de la aplicación desarrollada, el soporte e implementación de los objetos de datos modelados en los diagramas y documentos precedentes es particular para cada caso.

El software no trabaja con bases de datos ni otras formas de organización y relación de datos por sus mismas características, sino que los diferentes objetos se implementan como archivos planos, variables de entorno en tiempo de ejecución y matrices estructuradas, según resulte adecuado en cada caso.

Sigue a continuación el detalle de implementación de cada objeto modelado:



**Tabla Braille**

Matriz Pública de Dimensiones Fijas y registros estructurados.

```
Public B(57) As Braille 'Array para contener la tabla Braille
Public Type Braille
    Puntos As String * 6
    Caracter As String
    Pronunciacion As String
    Especial As Boolean
End Type
```

Su contenido se carga con la tabla Braille definida para la aplicación, indicando para cada fila de la misma los puntos que componen cada carácter, el valor ASCII del carácter en cuestión, su pronunciación para las funciones de fonación, y una bandera que permite el tratamiento correcto de aquellos caracteres Braille que no guardan una relación directa con caracteres ASCII sobre el texto: indicador de Mayúscula, indicador de Número, carácter de Nuevo Párrafo, marca de Bloque.



**Puntos en Progreso**

Define los puntos Braille que se encuentran seleccionados al momento de escritura de un carácter en progreso. El dato es registrado en el mismo objeto gráfico que se utiliza para su presentación en pantalla (puntos rojos en la ficha braille), implementando los mismos como un Array de Controles Shape de 6 elementos, uno para cada punto.



**Estado Aplicación**

Variables Globales numéricas de valores predefinidos. Esto permite la sincronización de los diferentes estados entre los procedimientos y funciones que se van ejecutando en los diferentes puntos o estados del software, de manera coordinada. Las variables que implementan el estado de la aplicación y sus posibles valores son:

**StatusTeclado:** indica si el teclado se encuentra en modo “standar” (para la intervención externa sobre el texto en edición) o en modo “LUCAS”, para no permitir la intervención externa, sino la edición exclusiva desde el tablero especial.

**StatusNumerico:** indica si se ingresó el carácter Braille de número, para interpretar correctamente el carácter siguiente (correspondiente a las letras “A” a “J”).

**StatusMayuscula:** indica si se ingresó el carácter Braille de mayúscula, para interpretar correctamente el carácter siguiente y su correspondiente codificación ASCII al momento de ser añadido al texto en edición.

**flagGuardado:** indica si el texto en edición fue ya guardado al archivo de trabajo, o no, para mantener la sincronización entre ambos elementos de dato.

**StatusMenu:** indica el nivel de menú y submenú en que se encuentra la aplicación (variable multivaluada de 0 a 7).



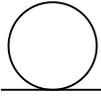
**Texto en Edición**

String de longitud variable y abierta que contiene la totalidad del texto en edición y la posición relativa del cursor sobre el mismo. Se implementa en la propiedad Text del objeto TextBox que presenta el texto en pantalla.



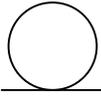
**Directorio**

Array virtual devuelto por el lenguaje al acceder a los componentes de una carpeta o subdirectorio del disco del sistema. Posibilita la recuperación de archivos guardados.



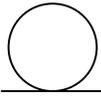
### **Separadores de Oración**

Vector que contiene los caracteres o códigos ASCII que son interpretados por la aplicación como indicadores de límite de las oraciones. Se cotejan las búsquedas de límite de oración con cada posición de este vector para determinar si un carácter en cuestión es o no un separador de oración.



### **Separadores de Palabras**

Vector que contiene los caracteres o códigos ASCII que son interpretados por la aplicación como indicadores de límite de las palabras. Se cotejan las búsquedas de límite de palabra con cada posición de este vector para determinar si un carácter en cuestión es o no un separador de palabra en las funciones de lectura y desplazamiento.



### **Configuración**

Se implementa como un conjunto de variables globales a la aplicación que mantienen el estado y configuración de cada característica customizable de la aplicación.

```
Public Voces(1 To 7) As Voz 'Configuraciones de la voz
Public VozPorDefecto As Integer
Public flagLeerNombreMenu As Boolean
Public flagLeerMenus As Boolean
Public flagLeerTexturas As Boolean
Public flagSonidos As Boolean
Public flagRedondeo As Integer '0=Redondear;1=Truncar;2=Decimales
Public flagBeepENTER As Boolean
Public flagTonoDistintivoSignos As Boolean 'True=Pronunciar el
nombre de los signos y Aplicar modificación de tono y velocidad a la
lectura de los signos; False=No leer el nombre de los signos!
Public ArchivoP1 As String
Public ArchivoP2 As String
Public ArchivoP3 As String
Public ArchivoP4 As String
Public ArchivoP5 As String
Public ArchivoP6 As String
Public ArchivoAceptar As String
Public ArchivoCancelar As String
Public ArchivoTeclasMenu As String
Public SeparadorDecimal As String * 1
Public PuertoCom As Byte
```

El vector voces ejemplificado, que mantiene las voces configuradas en la aplicación, responde a la siguiente estructura:

```
Public Type Voz
 Codigo As Integer      'Identificador de la voz asignada
Velocidad As Integer   'Rate para la pronunciación normal
Tono As Integer        'Pitch para la pronunciación normal
VelocidadSignos As Integer 'Rate para la pronunciación de los signos
TonoSignos As Integer  'Pitch para la pronunciación de los signos
End Type
```

Todos los valores de estas variables globales son establecidos al inicio de la aplicación a partir de los valores grabados en el archivo plano "" en el que se registran los valores de configuración establecidos para la aplicación. Cuando se realiza el caso de uso de configuración del software se modifican estos valores y el correspondiente archivo, en tanto que en la práctica totalidad de las funciones del software son considerados los valores de estas variables para condicionar el comportamiento de la aplicación.

## 14. Conclusiones

---

El desarrollo del Proyecto de Aplicación Profesional documentado en este Trabajo Final de Graduación posibilitó varios aportes y conclusiones, según los siguientes aspectos.

### ***Resultado aplicado para la evolución escolar de Lucas García***

A partir de la primera implementación de la herramienta construida, *Lucas pudo pasar de grado*. Sin dudas este fue el principal y más importante logro de este proyecto.

Se encontraba repitiendo el 1er grado de la escuela primaria por tercera vez al momento de conocerlo, pues se hallaba condicionado al requerimiento académico formal de saber leer y escribir, e incapacitado de adquirir dichas destrezas por sus limitaciones físicas, no podía hacerlo.

Luego de poner en marcha la solución construida, continuó con su desarrollo escolar y fueron siendo necesarias las diversas ampliaciones detalladas con el esquema de versiones seguido en la herramienta, proveyendo los recursos necesarios para que su aplicación desde el punto de vista psicopedagógico, dirigido por su maestra integradora, le permitiera seguir avanzando sin nuevas repeticiones en la escuela, encontrándose hoy en el 4to grado, y avanzando a buen ritmo en su educación de ciclo primario.

### ***Adquisición de nuevas habilidades psicopedagógicas por parte del usuario de la herramienta***

Resultó también importante (y sorprendente incluso para su psicopedagoga) la capacidad adquirida por Lucas para el desarrollo de ciertas habilidades intelectuales que hasta el momento no había podido desarrollar: el concepto de secuencia, ordenamiento, y posición relativa en un conjunto, disparados a partir del recurso del “cursor” en el texto, y la utilización de la herramienta instrumentada en su plan pedagógico. Habían trabajado anteriormente con él estos elementos desde lo concreto, con recursos materiales (bolitas, fichas, etc...) con pocos resultados positivos, y con la consecuente frustración. Considera su pedagoga que el desarrollo de las habilidades referidas pudo deberse en gran parte a la fuerte motivación que lo guió en esta nueva posibilidad de acercarse a sus

pares, integrándose socialmente en mayor medida en tanto participante de las mismas actividades áulicas y los mismos requisitos y exigencias académicos.

### ***Aporte intelectual para SoftVoice y la comunidad de desarrolladores de software***

El resultado de esta tarea de investigación y desarrollo, la API Independiente para Visual Basic para el motor SVTTS, junto a la aplicación de ejemplo, fue enviado al Señor Joseph Katz para su libre distribución a todos quienes se interesen en su producto y en su utilización desde esta herramienta de desarrollo.

Cabe señalar que aunque no fue incluida en esta tarea documental, la API fue remitida a SoftVoice tanto en su idioma original (Inglés) como la versión aquí documentada, en español, autorizando a la compañía a que pueda distribuirla libremente a quienes lo requieran.

Asimismo, en cuanto a los hallazgos realizados respecto de los protocolos de llamada a funciones compiladas en DLL con parámetros de punteros tipo FAR, específicos para su tratamiento desde los lenguajes de programación C/C++ y Java, la configuración del protocolo equivalente para las llamadas desde Visual Basic sirve a cualquier proyecto de desarrollo en el que se requiera el acceso desde este último lenguaje a dicho tipo de funciones. En calidad de tal, la información fue publicada en diversos foros de programadores y sitios Web dedicados a la provisión de recursos para los desarrolladores de software, constituyendo un aporte a la comunidad.

## 15. Bibliografía consultada

---

Las referencias documentales y bibliográficas consideradas en este trabajo se resumen a continuación:

- a. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “Alfabetización de las personas que son sordas e invidentes”  
**Autor:** Barbara Miles  
**Origen:** Centro de Información Nacional sobre Niños que son Sordos y Ciegos de Estados Unidos  
**URL:** <http://www.tr.wou.edu/dblink/lit-span.htm>  
**Fecha de Acceso:** 10/03/2004
  
- b. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “Discapacidad visual y acceso a la información”  
**Autor:** Soledad Mochales López  
**URL:** <http://usuarios.iponet.es/casinada/25ciegos.htm>  
**Fecha de Acceso:** 10/03/2004
  
- c. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “As Tecnologias de Informação e Comunicação (T.I.C.) e a pessoa portadora de deficiência motora severa: Construção de um modelo de avaliação.”  
**Origen:** 3º Congreso Iberoamericano de Informática Educativa  
**URL:** [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/EULALIA/EULALIA.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/EULALIA/EULALIA.HTML)  
**Fecha de Acceso:** 17/05/2004
  
- d. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “CantaLetras: Sistema multimedial de apoyo al proceso de enseñanza de la lectoescritura para niños ciegos”  
**Origen:** 3º Congreso Iberoamericano de Informática Educativa  
**URL:** [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/125\\_E\\_126/DEMO1.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/125_E_126/DEMO1.HTML)  
**Fecha de Acceso:** 17/05/2004

- e. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “FALAS: Ferramenta Alternativa de Aquisição Simbólica”  
**Origen:** 3º Congreso Iberoamericano de Informática Educativa  
**URL:** [http://www.niee.ufrgs.br/ribie98/CONG\\_1996/CONGRESSO\\_HTML/125\\_E\\_126/DEMO1.HTML](http://www.niee.ufrgs.br/ribie98/CONG_1996/CONGRESSO_HTML/125_E_126/DEMO1.HTML)  
**Fecha de Acceso:** 17/05/2004
- f. **Tipo de Material:** Biografía – Página Web  
**Título:** “Biografía de Hellen Keller”  
**Origen:** SENSE Internacional (Latinoamérica) – Centro de Recursos sobre Sordoceguera en Español  
**URL:** <http://www.sordoceguera.org/Historias Reales/Helen Keller.htm>  
**Fecha de Acceso:** 22/04/2004
- g. **Tipo de Material:** Video Digitalizado  
**Área Temática:** Hellen Keller y el desarrollo de sus capacidades  
**Origen:** SENSE  
**URL:** <http://www.sense.org.uk>  
**Archivos:** Hellen Keller parte 1.wmv  
Hellen Keller parte 2.wmv  
Hellen Keller parte 3.wmv  
**Fecha de Acceso:** 20/04/2004
- h. **Tipo de Material:** Video Digitalizado  
**Área Temática:** Técnicas comunicacionales y pedagógicas con niños ciegos multiimpedidos graves  
**Origen:** SENSE  
**URL:** <http://www.sense.org.uk>  
**Archivos:** comunicacion.wmv  
sistemas de comunicacion parte 1.zip  
sistemas de comunicacion parte 2.zip  
sistemas de comunicacion parte 3.zip  
**Fecha de Acceso:** 20/04/2004

- i. **Tipo de Material:** Publicación Técnica – Página Web  
**Título:** “IT’S MORE THAN A FLASHLIGHT: Developmental Perspectives for Visually and Multi Handicapped Infants and Preschoolers”  
**Autores:** Lois Harrel y Nancy Akeson  
**Origen:** American Foundation for the Blind - International Council for Education of the Visually Handicapped, Región Latinoamericana, Córdoba, Argentina (1988)  
**URL:** <http://www.ropard.org/bibliography.shtml>  
**Fecha de Acceso:** 11/04/2004
- j. **Tipo de Material:** Publicación Técnica – Documento electrónico  
**Título:** “SISTEMAS MULTIMEDIALES DE APOYO AL APRESTO ESCOLAR Y A LA LECTURA Y ESCRITURA INICIAL PARA NIÑOS DISCAPACITADOS VISUALES”  
**Autores:** Trabajo conjunto de varios profesionales de la Escuela de Psicología, la Escuela de Diseño, y la Escuela de Ingeniería / Servicio Ciencias de la Computación, de la Pontificia Universidad Católica de Chile  
**Origen:** Pontificia Universidad Católica de Chile  
**URL:** <http://www.puc.cl/toquemagico/html/resumen.pdf>  
**Fecha de Acceso:** 30/10/2004
- k. **Tipo de Material:** Texto Técnico  
**Título:** “PC INTERNO 5”  
**Autor:** M. TISCHER, B. JENNRICH  
**Editorial:** MARCOMBO
- l. **Tipo de Material:** Texto Técnico  
**Título:** “GUIA COMPLETA PARA PC”  
**Autor:** GILSTER RON  
**Editorial:** MCGRAW-HILL
- m. **Tipo de Material:** Texto Técnico  
**Título:** “ARQUITECTURA DE EQUIPOS Y SISTEMAS INFORMATICOS”  
**Autor:** VALDIVIA MIRANDA CARLOS  
**Editorial:** PARANINFO

- n. **Tipo de Material:** Texto Técnico  
**Título:** “PROGRAMACION AVANZADA CON MICROSOFT VISUAL BASIC 6.0”  
**Autor:** BALENA FRANCESCO  
**Editorial:** MCGRAW-HILL
- o. **Tipo de Material:** Texto Técnico  
**Título:** “A VISUAL BASIC 6 PROGRAMMER'S TOOLKIT”  
**Autor:** MARQUIS HANK, SMITH ERIC  
**Editorial:** APRESS
- p. **Tipo de Material:** Texto Técnico  
**Título:** “PROGRAMACION EN C/C++”  
**Autor:** PAPPAS  
**Editorial:** ANAYA MULTIMEDIA
- q. **Tipo de Material:** Texto Técnico  
**Título:** “EFFICIENT C/C++ PROGRAMMING”  
**Autor:** HELLER  
**Editorial:** ACADEMIC PRESS
- r. **Tipo de Material:** Texto Técnico  
**Título:** “PROGRAMMING WINDOWS SERVICES”  
**Autor:** MORIN RANDY  
**Editorial:** JOHN WILEY & SONS
- s. **Tipo de Material:** Texto Técnico  
**Título:** “WIN32 API PUZZLE BOOK AND TUTORIAL FOR VISUAL BASIC PROGRAMMERS”  
**Autor:** APPLEMAN DAN  
**Editorial:** APRESS

- t. **Tipo de Material:** Texto Técnico  
**Título:** “Lifecycle Planning Rapid Development”  
**Autor:** Steve McConnell  
**Editorial:** Microsoft Press
- u. **Tipo de Material:** Texto Técnico  
**Título:** “UML GOTA A GOTA”  
**Autor:** FOWLER MARTIN y SCOTT KENDALL  
**Editorial:** ADDISON-WESLEY IBEROA
- v. **Tipo de Material:** Texto Técnico  
**Título:** “INTEGRAR UML EN LOS PROYECTOS”  
**Autor:** LOPEZ NATALIE  
**Editorial:** GESTION 2000
- w. **Tipo de Material:** Texto Técnico  
**Título:** “UML Y PATRONES – Introducción al Análisis y Diseño Orientado a Objetos y Proceso Unificado”  
**Autor:** LARMAN CRAIG  
**Editorial:** PRENTICE-HALL
- x. **Tipo de Material:** Documento Técnico – Archivo PDF  
**Título:** “Ejercicio de Aplicación de UML al Análisis de Sistemas”  
**Autores:** Alexis Gigena y Juan Pablo Ferreira Centeno  
**Editorial:** Colegio Universitario IES Siglo 21  
**URL:** [http://www.ies21.com.ar/trabajos/tp\\_inf\\_CasoUML.pdf](http://www.ies21.com.ar/trabajos/tp_inf_CasoUML.pdf)  
**Fecha de Descarga:** 03/02/2005
- y. **Tipo de Material:** Documento Técnico – Archivo DOC  
**Título:** “Principios y directrices de diseño de interfaces de usuario”  
**URL:** <http://rss.atalca.cl/web/Estandares y Guias.doc>  
**Fecha de Descarga:** 03/02/2005

- z. **Tipo de Material:** Trabajo Académico (Tesis Licenciatura) – Archivo PDF  
**Título:** “Lineamientos de Diseño para Interfaces de Pantalla Amplia en UIML”  
**Autor:** Héctor Aragón Cantú  
**Origen:** Universidad de las Américas – Puebla – México  
**URL:** [http://mail.udlap.mx/~tesis/lis/aragon\\_c\\_h/capitulo4.pdf](http://mail.udlap.mx/~tesis/lis/aragon_c_h/capitulo4.pdf)  
**Fecha de Descarga:** 03/02/2005
- aa. **Tipo de Material:** Texto Técnico  
**Título:** “Programación Lógica”  
**Autor:** Diego Obregón – Érica Bongiovanni – Alejandra Farneti  
**Editorial:** Colegio Universitario IES Siglo 21
- bb. **Tipo de Material:** Publicación OMS - 1980  
**Título:** “Clasificación Internacional de Deficiencias, Discapacidades y Minusvalías”  
**Origen:** Organización Mundial de la Salud (OMS)
- cc. **Tipo de Material:** Publicación OMS - 2001  
**Título:** “Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud”  
**Origen:** Organización Mundial de la Salud (OMS)
- dd. **Tipo de Material:** Definición Técnica – Publicación Web  
**Término:** “Resiliencia”  
**URL:** <http://www.psicologia-positiva.com/resiliencia.html>  
**Fecha de Acceso:** 28/06/2005

**FORMACIÓN DEL  
EQUIPO PROFESIONAL  
“ÉBANO SOLUCIONES”**

Como epílogo de este trabajo, quisiera comentar que fue el inicio del desarrollo de este proyecto fue el puntapié inicial para la conformación de un equipo interdisciplinario de profesionales dedicados al desarrollo de soluciones y herramientas tecnológicas para personas con discapacidad, de características particulares y no convencionales.

Notamos, junto al Ing. Diego Beltramone, que podía haber muchas otras personas en la situación de Lucas: con discapacidades no convencionales y conjuntas, llamadas en general Multidéficit, que difícilmente podían acceder a herramientas que les permitieran mejorar su calidad de vida, por no tratarse de patologías comunes para las que existen herramientas estandarizadas.

Un paralítico puede comprar una silla de ruedas en cualquier ortopedia; un ciego puede escribir utilizando una máquina Braille o incluso el teclado convencional de la computadora y leer páginas de texto escrito en braille sin inconvenientes; pero ... ¿qué puede hacer para comunicarse con el mundo una persona con parálisis cerebral que no puede controlar sus movimientos involuntarios, que carece de precisión motriz, y que no puede hablar, pese a que ve y oye correctamente? ¿Cómo puede una persona parapléjica accionar una silla de ruedas eléctrica utilizando los movimientos de su mentón? Estas y muchas otras preguntas no tienen respuestas para buena parte de los seres humanos que nos rodean, y poder aportar nuestro granito de arena para mejorar la calidad de vida de ellos fue el espíritu que nos unió, al que se fueron sumando otros especialistas, y que nos permitió finalmente conformar el grupo *Ébano Soluciones*.

### ***Constitución del Equipo***

EBANO Soluciones es una asociación libre de profesionales de diferentes disciplinas, que intertrabajan en el diseño y fabricación de herramientas según las características de cada proyecto. En la actualidad está compuesta por los siguientes miembros:

- **ASC Diego G. Obregón**

*Analista de Sistemas de Computación*

Análisis, diseño y desarrollo de las aplicaciones informáticas de facilitación de recursos para los personas con discapacidad con las que se trabaja.

- **Prof. Soledad Martín**

*Profesora de ciegos – Especialidad en multiimpedidos*

Terapista especializada en psicopedagogía y desarrollo de niños con discapacidades múltiples, responsable del análisis y diagnóstico clínico-terapéutico de los pacientes para quienes se desarrollen proyectos.

- **Lic. Hugo Cantore**

*Fisioterapeuta – Kinesiólogo – Especializado en rehabilitación*

Relevamiento e identificación de capacidades motrices y posibles aplicaciones para explotarlas, según los objetivos planteados.

- **Dis. José María Espósito**

*Diseñador Industrial*

Responsable del diseño y desarrollo de los aspectos físicos ergonómicos para los destinatarios de las soluciones desarrolladas, especialmente en los casos de dificultades motoras severas.

- **Ing. Diego Beltramone**

*Ingeniero Electrónico*

Encargado de los aspectos de hardware y tecnología electrónica presentes en los desarrollos que se llevan a cabo.

Por las características de organización señaladas, EBANO Soluciones no tiene figura legal como organización, sino que son sus miembros quienes responden de manera independiente mediante sus propias habilitaciones profesionales y su correspondiente inscripción ante el fisco. No obstante, se incorporó recientemente a la Fundación Senda Azul, como uno de sus proyectos, conjuntamente con Terrara (Centro Terapéutico Integral) y PINET (Proyecto Educativo Terapéutico).

La forma de trabajo es por proyectos. Analizadas las posibilidades de desarrollo de cada paciente, se ponen en marcha las diferentes especialidades según necesidades en cada área, mediante un trabajo interdisciplinario e integrando de sus miembros. No se limitan a las posibilidades profesionales de sus integrantes, sino que se asocian nuevos profesionales en función de las necesidades. El único requisito para la incorporación de otros especialistas en un proyecto, además de la idoneidad y capacidad profesional, es la adhesión a los fuertes principios éticos que pautan los miembros de EBANO Soluciones, considerando el fuerte compromiso con el desarrollo de los beneficiarios de las aplicaciones en sus diferentes dimensiones humanas.

La visión del equipo queda reflejada en su slogan:



El enfoque epistemológico de la tarea desarrollada se basa en el enfoque de resiliencia: *centrando su trabajo en el desarrollo de las capacidades de las personas y no en sus limitaciones.*

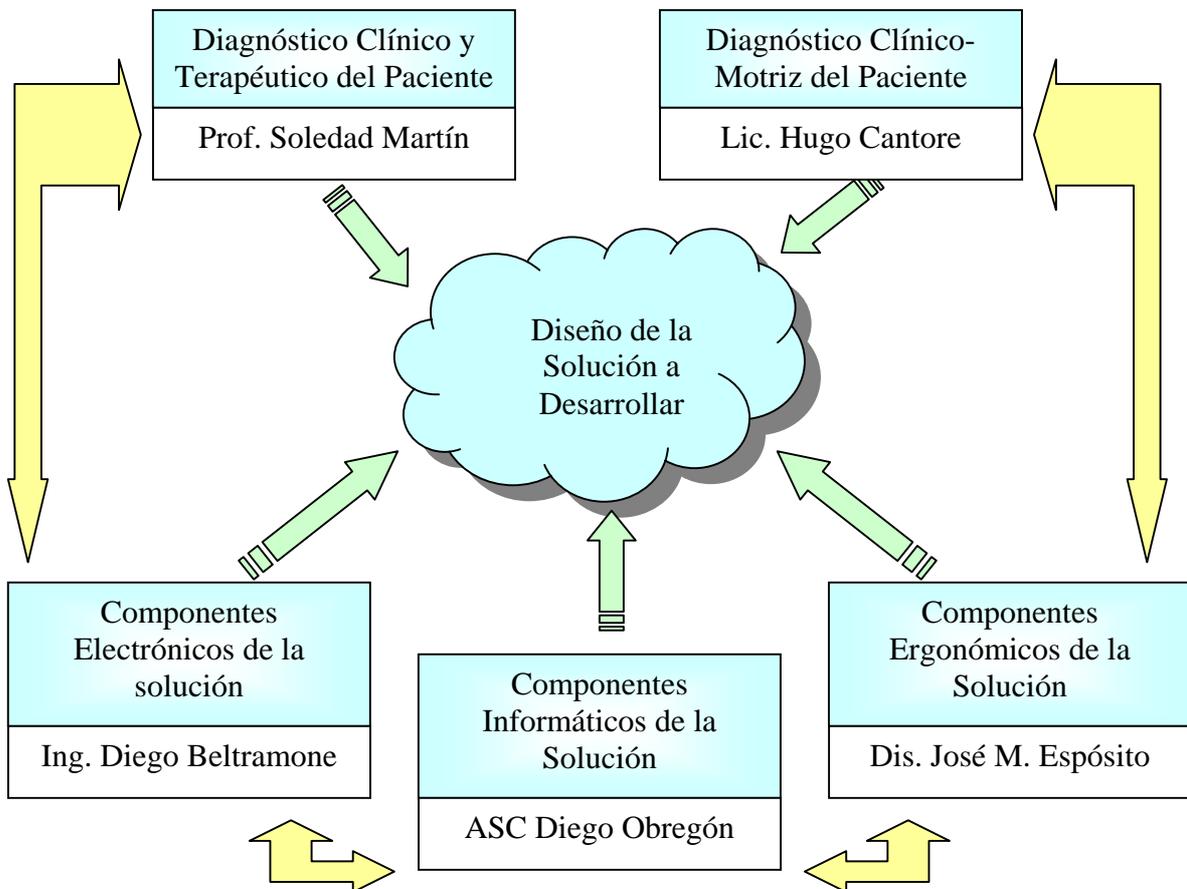
El trabajo consiste en el desarrollo de soluciones tecnológicas que permitan mejorar la calidad de vida de los usuarios.

Dichas soluciones pertenecen -de manera no excluyente- a alguna de las siguientes categorías:

- **Comunicación Aumentativa:** Soluciones integrales orientadas a proveer capacidades comunicacionales a personas cuyas discapacidades no le permitan desarrollarlas. Abarca lectoescritura, soluciones para personas no videntes o disminución visual, sordos e hipoacúsicos, etc.
- **Asistencia en la rehabilitación:** Diferentes recursos tecnológicos que puedan ser combinados para el desarrollo de herramientas e instrumentos de asistencia para los rehabilitadores, desde aparatología específica para actividades como la gimnasia sin esfuerzo o la rehabilitación motriz, hasta sistemas electrónicos para entrenar a pacientes sordos, o instrumentos que les permitan el auto control de su fonación.
- **Biofeedback:** Información suministrada en forma de retroalimentación para que la persona con discapacidad aprenda a mover voluntariamente partes del cuerpo cuyos movimientos son naturalmente reflejos. Incluye aplicaciones de indicadores de posición corporal
- **Robótica y Domótica:** Aplicaciones para hacer accesible el control de los elementos del hogar a personas con discapacidad (particularmente con problemas de motricidad).
- **Entretenimiento:** Las soluciones tecnológicas en materia lúdica incluyen, desde adaptaciones para el uso de videojuegos o navegación por Internet, hasta desarrollos específicos, según el nivel intelectual del beneficiario y sus capacidades. El enfoque en estos casos se centra en lo pedagógico.

## ***Estructura Organizacional y Funcional***

Como se desprende de lo anteriormente mencionado, no existe una organización formal interna del equipo de trabajo, sino que la interrelación de sus diferentes miembros deviene del natural trabajo interdisciplinario que ante cada nuevo proyecto se establece. Resulta por esto imposible trazar un organigrama formal de la organización. No obstante pueden establecerse las relaciones funcionales entre las áreas señaladas:



Como se observa, a partir del acercamiento profesional clínico-terapéutico y el análisis y diagnóstico del sujeto, son los profesionales de la salud quienes trazan los objetivos de la solución a desarrollar, a partir de las necesidades y demandas – del propio paciente o su grupo familiar – y las posibilidades o capacidades desarrolladas o desarrollables. A partir de allí, todos los profesionales aportan –desde sus respectivas especialidades– al diseño de la solución, desarrollando el trabajo interdisciplinario señalado.

## ***Estado de Desarrollo y Proyección***

Ébano soluciones fue evolucionando, aceitando los mecanismos de relevamiento, diagnóstico, propuestas, diseño, construcción e implementación de las soluciones desarrolladas. La principal referencia del equipo es para con los profesionales de la salud, que requieren de nuestros servicios. Hemos establecido como norma no trabajar a partir de requerimientos particulares si no es que los mismos vienen solicitados y avalados por los respectivos equipos terapéuticos que tratan a los pacientes.

En este sentido, actualmente contamos como una de nuestras actividades, la difusión de nuestros servicios y los casos de aplicación tecnológica desarrollados, mediante charlas, seminarios y conferencias dirigidos a profesionales de las salud. En el último año, hemos desarrollado este tipo de actividades en los principales nosocomios y centros de rehabilitación de la ciudad de Córdoba, incluyendo entre otros:

- Centro de Tomografía Computada Oulton
- Hospital Misericordia – Servicio de Rehabilitación
- ALPI
- Sanatorio Allende
- Centro Terapéutico San Camilo
- Centro Terapéutico Un Camino
- Fundación Rayo de Sol – Dr. Castillo Morales –

Hemos desarrollado hasta el momento decenas de soluciones, desde herramientas enteramente mecánicas, mecánico-electrónicas, integrales (incluyendo software), aplicaciones informáticas, etc...

Dar a conocer nuestra actividad profesional nos ha permitido desarrollar una cantidad creciente de soluciones, tendencia que se mantiene e incrementa día a día.

## ***Datos de Contacto***

**Domicilio (Fundación Senda Azul):** Columbres 1048 – B° San Martín – Córdoba

**Teléfono:** (0351) 473-1297

**E-Mail:** [info@ebano-soluciones.com.ar](mailto:info@ebano-soluciones.com.ar)



**Web Site:** [www.ebano-soluciones.com.ar](http://www.ebano-soluciones.com.ar)