

# Universidad Siglo 21



## **Trabajo Final De Grado. Prototipado Tecnológico** **Carrera: Licenciatura En Informática**

### **Aplicación Interactiva Para La Enseñanza De Metodologías De Desarrollo De Software A Jóvenes**

Autor: Lucas Antu Fernández

Legajo: VINF04231

Colón - Buenos Aires, 17 julio de 2023

## Índice

<b>Resumen</b> .....	<b>4</b>
<b>Abstract</b> .....	<b>4</b>
<b>Introducción</b> .....	<b>6</b>
Antecedentes .....	6
Descripción Del Área Problemática .....	7
<b>Justificación</b> .....	<b>7</b>
<b>Objetivo General Del Proyecto</b> .....	<b>7</b>
<b>Objetivos Específicos Del Proyecto</b> .....	<b>8</b>
<b>Marco Teórico Referencial</b> .....	<b>8</b>
Dominio Del Problema .....	8
TICs .....	9
Competencia .....	10
<b>Diseño Metodológico</b> .....	<b>11</b>
<b>Relevamiento</b> .....	<b>12</b>
Relevamiento Estructural .....	12
Relevamiento Funcional .....	12
<b>Procesos De Negocios</b> .....	<b>14</b>
<b>Diagnóstico Y Propuesta</b> .....	<b>16</b>
Diagnóstico .....	16
Propuesta .....	17
<b>Objetivo, Límites Y Alcances Del Prototipo</b> .....	<b>17</b>
Objetivo Del Prototipo .....	17
Límites .....	17
Alcances .....	17
<b>Descripción Del Sistema</b> .....	<b>18</b>
Requerimientos Funcionales .....	18
Requerimientos No Funcionales .....	18
Diagrama De Casos De Uso .....	19
Descripciones De Casos De Uso .....	20
Diagramas De Secuencia .....	28
Estructura De Datos .....	34
Prototipos De Interfaces De Pantallas .....	35
<b>Seguridad</b> .....	<b>39</b>
Acceso A La Aplicación .....	39
Política De Respaldo De Información .....	40

<b>Análisis De Costos</b> .....	<b>41</b>
<b>Análisis De Riesgos</b> .....	<b>43</b>
<b>Conclusiones</b> .....	<b>48</b>
<b>Referencias</b> .....	<b>50</b>

### **Índice De Imágenes**

Diagrama de Gantt .....	12
Diagrama de proceso de negocio .....	15
Diagrama de casos de uso .....	20
Diagrama de secuencia para el caso de uso Iniciar Sesión.....	29
Diagrama de secuencia para el caso de uso Registrar Nuevo Estudiante.....	30
Diagrama de secuencia para el caso de uso Iniciar Un Curso .....	31
Diagrama de secuencia para el caso de uso Mostrar Lección Actual Del Curso .....	32
Diagrama de secuencia para los casos de uso Solicitar Explicación Básica, Solicitar Explicación Avanzada y Realizar Evaluación Actual .....	33
Diagrama de secuencia para el caso de uso Consultar Progreso Del Estudiante .....	34
Diagrama de clases.....	35
Prototipo de interfaz de pantalla 1.....	36
Prototipo de interfaz de pantalla 2.....	37
Prototipo de interfaz de pantalla 3.....	37
Prototipo de interfaz de pantalla 4.....	38
Prototipo de interfaz de pantalla 5.....	38

### **Índice De Tablas**

Tabla comparativa entre plataformas .....	10
Proceso de negocio Dictado De Clases .....	16
Proceso de negocios Aprendizaje Autodidacta .....	16
Descripción del caso de uso Iniciar Sesión .....	20
Descripción del caso de uso Registrar Estudiante.....	22
Descripción del caso de uso Iniciar Un Curso .....	23
Descripción del caso de uso Mostrar Lección Actual Del Curso .....	24
Descripción del caso de uso Solicitar Explicación Básica .....	25
Descripción del caso de uso Solicitar Explicación Avanzada.....	26
Descripción del caso de uso Realizar Evaluación Actual .....	26
Descripción del caso de uso Consultar Progreso Del Estudiante .....	27

Costo en recursos humanos: escenario pesimista.....	41
Costo en recursos humanos: escenario optimista.....	42
Costos en hardware y software para el desarrollo.....	43
Costos finales en ambos escenarios .....	43
Riesgos identificados del proyecto.....	44
Análisis cuantitativo y grado de exposición.....	45
Plan de contingencia para los riesgos.....	46

## Resumen

En la actualidad la educación hacia los jóvenes en el campo de la informática y, más precisamente en el desarrollo de software, se centra en exponer principalmente solo un área de este campo: la programación. Se llegó a la conclusión de que la educación en este tema puede ir mucho más allá de la enseñanza de la programación, ya que en el mundo profesional los procesos de desarrollo de software son muy grandes y complejos e implican una gran cantidad de actividades diferentes además de la programación. Para realizar un aporte al respecto, se propuso el desarrollo de una aplicación interactiva con fines educativos, la cual buscar dar un primer acercamiento al estudiante al conocimiento del mundo del desarrollo de software profesional mediante una serie de cursos interactivos basados en metodologías de desarrollo de software. Para lograr este objetivo se diseñó un proyecto para llevar adelante el desarrollo de dicha aplicación teniendo en cuenta su alcance, costos monetarios, tiempo de ejecución y sus riesgos, y se desarrolló un prototipo para poner a prueba las funciones nucleares de la misma.

**Palabras clave:** educación, programación, desarrollo de software, aplicación interactiva, metodologías de desarrollo de software.

## Abstract

Currently, education for young people in the field of computer science, specifically software development, focuses primarily on exposing only one area of this field: programming. It has been concluded that education in this subject can go far beyond teaching programming since professional software development processes are large, complex, and involve a variety of activities beyond just programming. To contribute to this issue, the development of an interactive educational application was proposed, aiming to provide students with an initial approach to the knowledge of the professional software development world through a series of interactive courses based on software development methodologies. To achieve this objective, a project was designed to carry out the development of the application, considering its scope, financial costs, execution time, and risks. Additionally, a prototype was developed to test its core functions.

**Keywords:** education, programming, software development, interactive application, software development methodologies.

## Introducción

Durante este trabajo se analizó la manera en que los jóvenes son formados en el tema del desarrollo de software. Para ello se analizaron diferentes plataformas educativas disponibles en línea a nivel global, se analizó la forma en que se educa en las instituciones, y finalmente se analizó la manera en que los estudiantes autodidactas realizan su aprendizaje.

Se llegó a la conclusión de que el aprendizaje de los jóvenes en el tema del desarrollo de software está incompleto, ya que ellos aprenden principalmente a **programar** a pesar de que en el mundo profesional la programación es **solo una actividad más** dentro de un **proceso de desarrollo de software** mucho más grande.

Para cumplir con el objetivo de una educación más completa, se desarrolló una aplicación interactiva dirigida principalmente a jóvenes que tiene como objetivo facilitarles el aprendizaje de una serie de conocimientos fundamentales en el tema del desarrollo de software.

### *Antecedentes*

Actualmente se ofrecen una gran cantidad de plataformas educativas digitales para la formación en desarrollo de software dirigidas a jóvenes.

Estas plataformas, en general, fomentan el interés en la programación, facilitan el aprendizaje, la enseñanza y la formación de comunidades de estudiantes y profesores.

Muchas de estas plataformas, como por ejemplo Tynker y Scratch, ofrecen principalmente un lenguaje que permite el desarrollo de videojuegos sencillos a través la programación visual (sin escribir código) y también ofrecen otras herramientas necesarias para desarrollar estos videojuegos sencillos; además de cursos, material para los profesores y desafíos para los alumnos.

Otras plataformas además ofrecen, por ejemplo, la capacidad de desarrollar aplicaciones simples y el aprendizaje de la programación a través de un sistema que permite una transición simple desde un lenguaje de programación visual a lenguajes reales como JavaScript o Swift (como por ejemplo la plataforma Kodable).

En resumen, en todas estas plataformas se hace énfasis principalmente en la enseñanza de la **programación**.

### *Descripción Del Área Problemática*

La manera actual de aprender a desarrollar software de los jóvenes puede mejorarse expandiéndose más allá de la programación, ya que los estudiantes descubren en el futuro una realidad muy distinta a la aprendida, donde los procesos de desarrollo de software son muy grandes y complejos e implican una gran cantidad de actividades diferentes además de la programación.

### **Justificación**

En la industria del desarrollo de software la manera más habitual y recomendada para trabajar es llevar a cabo los procesos de desarrollo de software siguiendo los lineamientos de una **metodología de desarrollo** para poder afrontar el tamaño de los proyectos, la calidad del software requerida y la cantidad de personas que deben organizarse para llevar a cabo estos proyectos.

El desarrollo de una aplicación interactiva adaptada especialmente para los jóvenes y con un contenido más completo, el cual enseñaría a desarrollar software siguiendo metodologías de desarrollo de manera teórico y práctica, aportaría una educación más completa desde el principio mejorando así las bases para estudios superiores.

### **Objetivo General Del Proyecto**

- Desarrollar una aplicación interactiva para la enseñanza, a través de la práctica, de metodologías de desarrollo de software a jóvenes, es decir, que exponga una serie de conocimientos básicos para el entendimiento del tema mientras permite ponerlos en práctica de manera simplificada,



logrando de esta manera que los usuarios realicen un proceso de desarrollo de software simplificado.

## Objetivos Específicos Del Proyecto

- Analizar las diferentes ofertas educativas en TICs disponibles en internet.
- Diseñar un **sistema de cursos** que el usuario pueda elegir: cada curso permite al usuario aprender una metodología de desarrollo de software concreta, mientras desarrolla una aplicación sencilla dentro del propio sistema, siguiendo las etapas de dicha metodología.
- Diseñar un sistema de pantallas que representen, una a una, las etapas de una metodología de desarrollo de software.
- Diseñar una interfaz de usuario general para todas las pantallas basada en bloques predefinidos que se puedan seleccionar, arrastrar y soltar, que muestren explicaciones, etc., para así lograr facilitar el aprendizaje de sus usuarios jóvenes.
- Desarrollar un prototipo, en base a lo anterior, que permita el aprendizaje de al menos dos metodologías de desarrollo de software.

## Marco Teórico Referencial

### *Dominio Del Problema*

Para comenzar a hablar sobre enseñanza del desarrollo de software deberíamos empezar por definir que son el **software** y la **ingeniería de software**:

El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como

en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos. La ingeniería de software está formada por un proceso, un conjunto de métodos (prácticas) y un arreglo de herramientas que permite a los profesionales elaborar software de cómputo de alta calidad. (Pressman, 2010, p 1)

Y para la enseñanza de metodologías de desarrollo de software empezaremos por definir que son los **modelos del proceso del software**:

Cuando se trabaja en la construcción de un producto o sistema, es importante ejecutar una serie de pasos predecibles —el mapa de carreteras que lo ayuda a obtener a tiempo un resultado de alta calidad—. El mapa que se sigue se llama “proceso del software”. (Pressman, 2010, p 26)

Y porque son importantes estos modelos:

Porque da estabilidad, control y organización a una actividad que puede volverse caótica si se descontrola. Sin embargo, un enfoque moderno de ingeniería de software debe ser “ágil”. Debe incluir sólo aquellas actividades, controles y productos del trabajo que sean apropiados para el equipo del proyecto y para el producto que se busca obtener. (Pressman, 2010, p 26)

### *TICs*

Se utilizará el **motor de desarrollo de videojuegos Unity**: “Unity es la plataforma de desarrollo 3D en tiempo real líder en el mundo. Te ofrece lo que necesitas para crear, operar y monetizar juegos y experiencias increíbles en la gama de plataformas más amplia de la industria” (Unity Technologies, s.f.). A través de esta herramienta se podrá implementar una interfaz con gráficos en tres dimensiones para hacerla más atractiva para su público joven.

Además, junto con el motor de Unity, se utilizará el **lenguaje de programación C#**: “es un lenguaje de programación moderno, basado en objetos y con seguridad de

tipos. C# permite a los desarrolladores crear muchos tipos de aplicaciones seguras y sólidas que se ejecutan en .NET.” (Microsoft, 2022).

### *Competencia*

Hay muchas ofertas educativas en internet y por eso se tomaron tres de las opciones más representativas y se analizaron sus principales características desde el punto de vista educativo:

- **Scratch:** “Scratch es la comunidad de programación para niños y niñas más grande del mundo, y un lenguaje de programación con una interfaz sencilla que permite a los jóvenes crear historias digitales, juegos y animaciones.” (Fundación Scratch, s.f.).
- **Code.org:** “Code.org® es una organización sin fines de lucro dedicada a ampliar el acceso a la ciencia de la computación en las escuelas y aumentar la participación de mujeres jóvenes y estudiantes de otros grupos subrepresentados.” (Code.org, s.f.).
- **Kodable:** “Kodable les enseña a los niños a codificar en casa o en la escuela mediante divertidos juegos interactivos, al mismo tiempo que les brinda a los maestros un plan de estudios de codificación K-5 completo para el aula” (Kodable, s.f.).

A continuación se muestra una tabla comparativa para las diferentes plataformas analizadas.

	<b>Scratch</b>	<b>Kodable</b>	<b>Code.org</b>
<b>Enseñanza mediante programación visual</b>	Sí	Sí	Sí
<b>Programación mediante código</b>	No	Sí	Sí
<b>Enseñanza mediante desarrollo de aplicaciones</b>	No	No	Sí

<b>Enseñanza mediante desarrollo de videojuegos</b>	Sí	Sí	Sí
<b>Enseñanza práctica del desarrollo mediante metodologías</b>	No	No	No
<b>Enseñanza integral de la informática</b>	No	No	Sí

## Diseño Metodológico

Se siguieron los lineamientos del **proceso unificado de desarrollo de software**, el cual consiste en cuatro etapas iterativas e incrementales: inicio, elaboración, construcción y transición. Se utilizó UML como principal lenguaje de modelado.

A continuación, se detalla lo que se realizó principalmente en cada etapa del proceso unificado.

### Inicio:

- Se elaboró una descripción del producto final teniendo en cuenta las características de los productos ofrecidos por la competencia, los cuales fueron analizados leyendo la documentación de estos y utilizándolos.
- Se establecieron los principales requerimientos a través de casos de uso.
- Se establecieron los límites y alcances del sistema, y la factibilidad del proyecto.
- Se identificaron los principales riesgos.

### Elaboración:

- Se diseñó la arquitectura del sistema utilizando principalmente UML.
- Se diseñaron las interfaces de usuario.
- Se diseñaron los casos de uso.

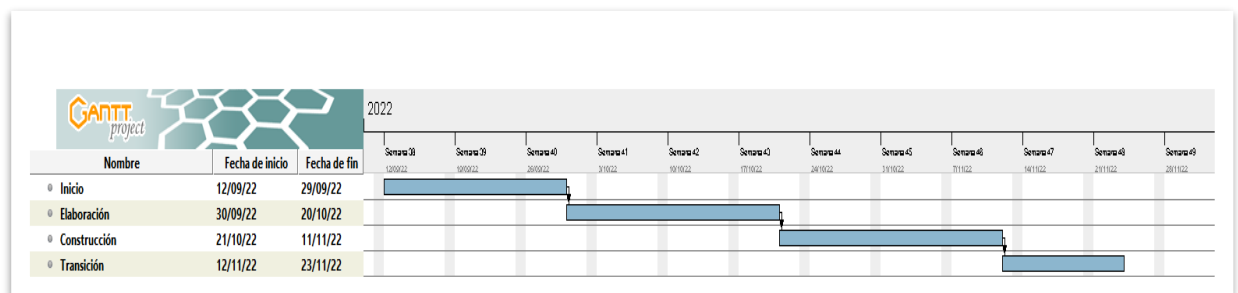
### Construcción:

- Se desarrolló principalmente dentro del marco de trabajo que ofrece Unity donde se gestionan los recursos que componen el videojuego, como los recursos gráficos, sonidos, código fuente, etc.
- Se codificó el software utilizando el IDE Visual Studio y el lenguaje de programación C#.
- Se realizaron las interfaces de usuario utilizando Unity.

### Transición:

- Se completó el software hasta cumplir con todos los requerimientos.
- Se realizaron las pruebas finales del producto.

Diagrama de Gantt realizado utilizando Gantt Project:



## Relevamiento

### *Relevamiento Estructural*

No se identificó una localización concreta para el proyecto, primero porque el dictado de clases se da en instituciones educativas en distintas ubicaciones, segundo porque el aprendizaje por medio de plataformas como Scratch esta localizadas en servidores remotos en internet, lo mismo para el caso de cursos en línea.

### *Relevamiento Funcional*

El relevamiento funcional se llevó a cabo, primero, a través de la utilización de diferentes plataformas educativas en TICs disponibles en línea a nivel global y la lectura de información disponible en sus sitios web oficiales y su documentación. También se analizó la forma en que se educa en las instituciones de nivel primario y secundario principalmente, y finalmente, se analizó la manera en que los estudiantes autodidactas realizan su aprendizaje.

Se identificaron los siguientes procesos que engloban las diferentes formas de aprendizaje analizadas:

**Nombre del proceso:** Dictado de clases.

**Roles:** Profesor, estudiante.

**Pasos:**

1. En la primera clase el profesor se presenta ante los alumnos al inicio de la materia en una institución educativa.
2. El profesor da la clase para impartir conocimientos en programación, según el plan de estudio del que dispone.
3. El estudiante intenta comprender dichos conocimientos.
4. En algún punto del cursado el estudiante debe ser evaluado y se enfrenta a un ejercicio.
5. El estudiante resuelve el ejercicio por si mismo o pide ayuda al profesor, si es posible.
6. Si pide ayuda al profesor este puede ayudarlo hasta que lo resuelva.
7. Si el alumno finalmente resuelve el ejercicio y la materia aún no finalizó se vuelve al paso 2.
8. El cursado de la materia finaliza.

**Nombre del proceso:** Aprendizaje autodidacta.

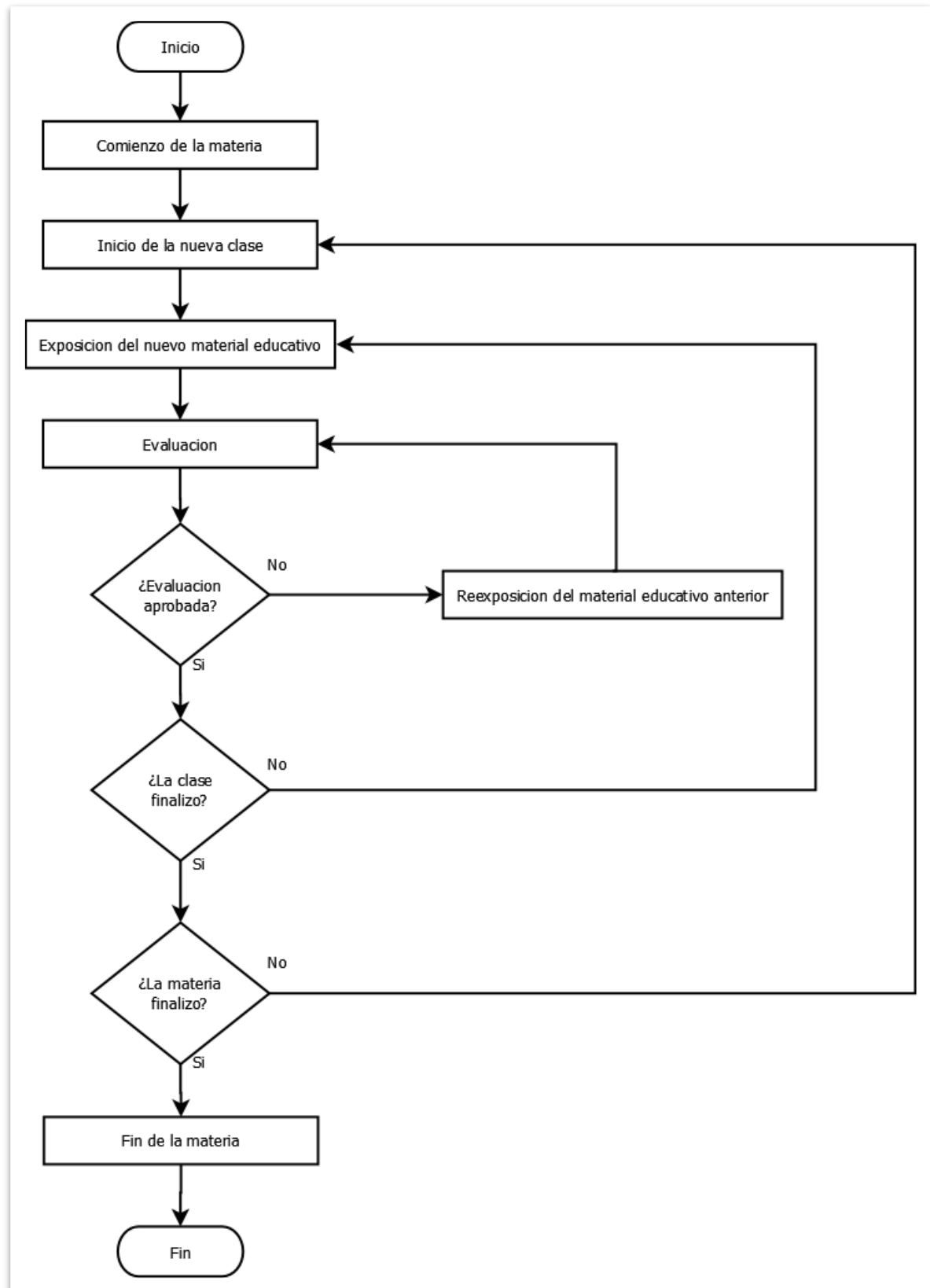
**Roles:** Estudiante autodidacta.

**Pasos:**

1. El estudiante autodidacta comienza sus estudios en desarrollo de software.
2. Como no dispone de un plan de estudio, realiza búsquedas en internet que generalmente lo llevan a cursos en línea enfocados en el área de la programación.
3. Generalmente realiza el curso de forma irregular en el tiempo y su aprendizaje se caracteriza por ser muy práctico y poco teórico.
4. Es poco frecuente que sus conocimientos sean evaluados por un profesor durante el proceso, ya que esta restricción suele no estar en los cursos en internet.
5. El estudiante finaliza el curso, aunque también es muy frecuente que el estudiante abandone el curso antes de terminarlo.

**Procesos De Negocio**

El siguiente diagrama, realizado con el programa Dia, involucra los procesos de negocio relevados representando las actividades nucleares de los mismos en un solo proceso.





## Diagnóstico Y Propuesta

### Diagnóstico

En los procesos de negocio relevados se han encontrado una serie de problemas detallados en las siguientes tablas, una por cada proceso.

#### DICTADO DE CLASES

Problemas	Causas
Se les enseña a los alumnos que el desarrollo de software profesional consiste únicamente en programar.	No se incluye en los planes de estudio en las escuelas una enseñanza completa.
Se les enseña a los alumnos a desarrollar software casi siempre de forma individual, cuando en la práctica es una actividad grupal.	Principalmente solo se ejercita y evalúa de forma individual.

#### APRENDIZAJE AUTODIDACTA

Problemas	Causas
El aprendizaje teórico del estudiante autodidacta está poco presente y es de baja calidad.	La gran mayoría de cursos disponibles en internet solo enseñan los conocimientos más prácticos y además no hay evaluación por parte de profesores.
Generalmente el estudiante autodidacta cree que el desarrollo de software profesional consiste únicamente en programar.	Falta de un plan de estudios completo.
El estudiante autodidacta aprende a desarrollar software solamente de forma individual.	Modalidad de estudio en solitario.

El cursado se irregular y a veces abandonado, es decir, poco disciplinado.	El cursado se realiza fueran de una institución y sus reglas.
--	---

### *Propuesta*

Se propone el desarrollo de una aplicación interactiva que ofrecerá a los jóvenes una enseñanza **más completa** que les aporte un mejor conocimiento del desarrollo de software del **mundo profesional**. Esto se logrará a través de la presentación de una serie de conceptos fundamentales de manera teórico y práctica en un formato de cursos basados en metodologías de desarrollo de software que contendrán una guía con explicaciones y evaluaciones en cada una de sus etapas, y que podrán ser realizados de manera **grupal**.

## **Objetivo, Límite Y Alcance Del Prototipo**

### *Objetivo Del Prototipo*

Cubrir los principales procesos para la enseñanza simplificada de **dos** metodologías de desarrollo de software completas.

### *Límites*

Desde que el estudiante joven comienza sus cursos introductorios en el área del desarrollo de software hasta que los termina.

### *Alcances*

- Registro del profesor y de los estudiantes en el sistema.
- Mostrar explicaciones al profesor y estudiantes.
- Evaluación del estudiante.
- Seguimiento del progreso de los estudiantes.

## Descripción Del Sistema

### *Requerimientos Funcionales*

El prototipo permitirá realizar ciertas acciones dependiendo de si se está realizando actualmente un **curso** o no. Cada curso permite el aprendizaje de una metodología de desarrollo de software concreta.

Deberá cumplir con los siguientes requerimientos mientras **no se está** realizando ningún curso:

- Permitir crear un perfil simplificado del usuario, el cual contenga su nombre y progreso actual de aprendizaje de cada curso.
- Permitir al usuario consultar el progreso actual de todos los cursos.
- Guardar automáticamente el estado del progreso del usuario en todos los cursos.

Además, deberá cumplir con los siguientes requerimientos en forma **secuencial** para realizar un curso:

1. Permitir al usuario seleccionar un curso de un listado, cada curso permite el aprendizaje de una metodología de desarrollo.
2. Iniciar el curso actual desde la primera etapa de la metodología del curso actualmente seleccionado mostrando dicha etapa en la interfaz del usuario.
3. Explicar al usuario a través de textos e imágenes en la interfaz del usuario, para que pueda comprender la etapa actual de la metodología del curso actualmente seleccionado.
4. Evaluar al usuario para que pueda superar la etapa actual de la metodología del curso actualmente seleccionado.
5. Cuando el usuario supera la etapa actual de la metodología del curso actualmente seleccionado el sistema debe pasar a la siguiente etapa.

### *Requerimientos No Funcionales*

La **interfaz de usuario** del prototipo deberá cumplir con los siguientes requerimientos:

- Fácil de utilizar para personas jóvenes.

En cuanto a la **portabilidad** del prototipo, deberá:

- Funcionar en una computadora personal con hardware limitado, para asegurar el alcance de la mayoría de los usuarios.
- Poder ejecutarse en Windows 10.

*Diagrama De Casos De Uso*

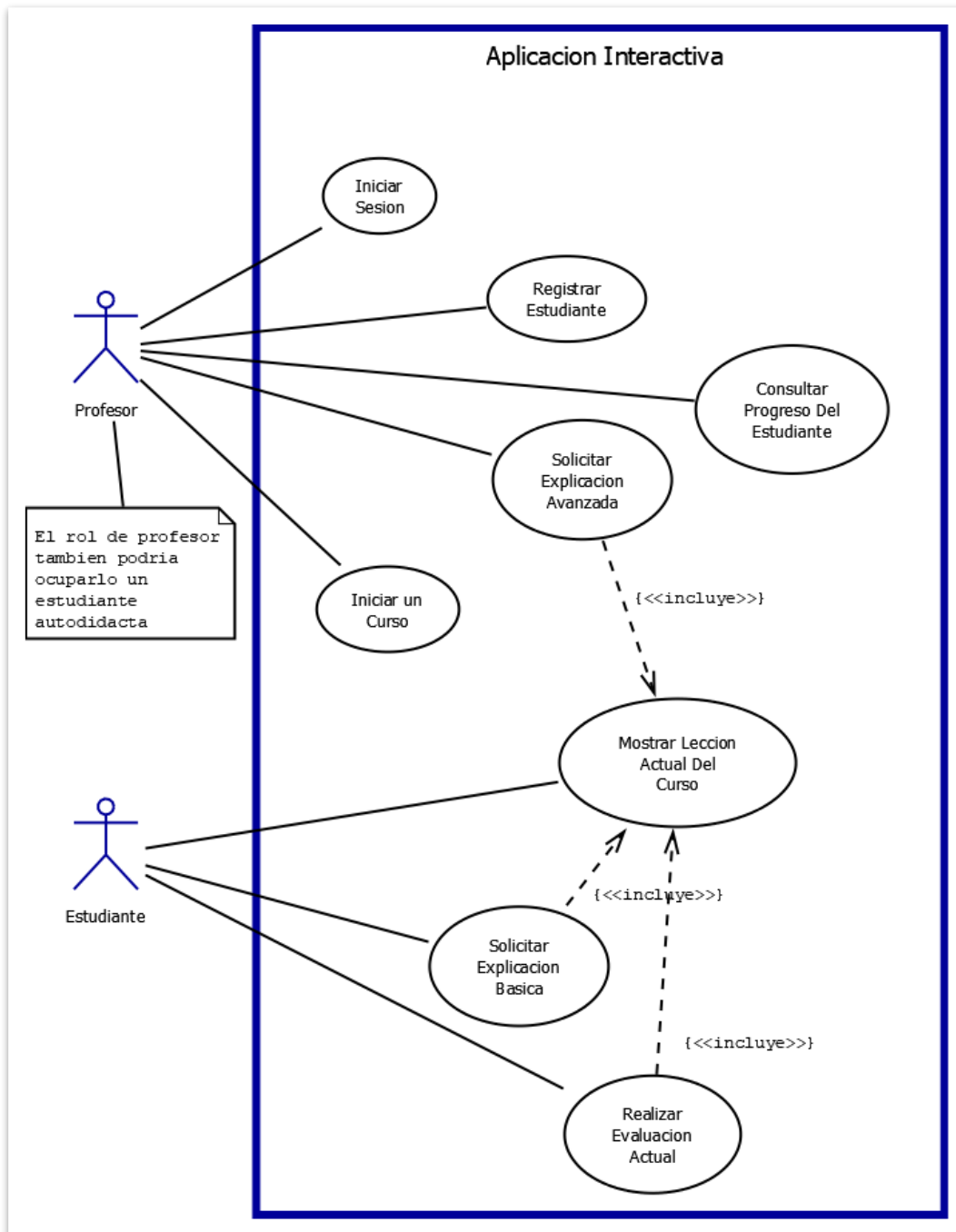


Diagrama realizado con el programa Dia

### Descripciones De Casos De Uso

Nombre del caso de uso	INICIAR SESIÓN
------------------------	----------------

<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitir que el profesor pueda iniciar sesión y administrar el sistema.	
<b>Descripción</b>	El profesor ingresa al sistema si tiene su perfil registrado, sino crea un nuevo perfil en el sistema, introduciendo un nombre de usuario y contraseña.	
<b>Precondición</b>	La UI para iniciar sesión se debe estar mostrando en pantalla.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor introduce su nombre de usuario y contraseña.
	2	El sistema le da acceso como administrador y muestra en la UI el listado de alumnos registrados, si existe alguno.
<b>Postcondición</b>	La sesión está iniciada por el profesor en el sistema.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	1 - A	El sistema registra un error al intentar acceder a los perfiles almacenados y muestra en la UI un mensaje de error.
	2 - B	El sistema no encuentra ningún perfil con el nombre de usuario y contraseña especificados por el profesor.
	3 - B	Se muestra en la UI la opción de crear nuevo perfil.
	4 - B	El profesor introduce nombre de usuario y contraseña nuevos.
	5 - B	El sistema archiva el nuevo perfil de profesor y se vuelve al paso 1.
	5 - C	El sistema registra un error al intentar almacenar el nuevo perfil y muestra en la UI un mensaje de error.
<b>Frecuencia esperada</b>	Frecuente.	

<b>Importancia</b>	Muy importante.
<b>Comentarios</b>	-

<b>Nombre del caso de uso</b>	<b>REGISTRAR ESTUDIANTE</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitir que el profesor pueda registrar al estudiante en el sistema.	
<b>Descripción</b>	El profesor registra un nuevo estudiante en el sistema.	
<b>Precondición</b>	El profesor debe haber iniciado sesión.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor introduce el nombre y el apellido del nuevo estudiante.
	2	El sistema almacena el perfil del nuevo estudiante con los datos que introdujo el profesor.
	3	Se muestra en la UI un mensaje de éxito.
<b>Postcondición</b>	El nuevo perfil del estudiante está archivado en el sistema.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	2 - A	Lo datos del nuevo estudiante ya habían sido registrados previamente.
	3 - A	Se muestra en la UI un mensaje de error: el estudiante ya está registrado en el sistema.
	2 - B	El sistema registra un error al intentar almacenar el nuevo perfil del estudiante.
	3 - B	Se muestra en la UI un mensaje de error.
<b>Frecuencia esperada</b>	Poco frecuente.	
<b>Importancia</b>	Muy importante.	
<b>Comentarios</b>	-	

<b>Nombre del caso de uso</b>	<b>INICIAR UN CURSO</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitirle al profesor iniciar uno de los cursos disponibles para que sea cursado por el estudiante.	
<b>Descripción</b>	El profesor selecciona uno de los cursos disponibles y le asigna un alumno para que este comience a cursarlo.	
<b>Precondición</b>	Debe haber algún perfil de estudiante creado.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor selecciona uno de los cursos disponibles mostrados en la UI.
	2	El sistema muestra un listado de estudiantes para que el profesor seleccione uno de ellos para que inicie el curso.
	3	El profesor selecciona uno o varios estudiantes del listado.
	4	El sistema completa la asignación del curso al estudiante o estudiantes seleccionados, almacenando esta información en el perfil del estudiante o estudiantes.
	5	Se muestra en la UI un mensaje de éxito.
<b>Postcondición</b>	Al estudiante se le asignó un nuevo curso.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	2 - A	El sistema detecta un error al intentar recuperar el listado de cursos disponibles.
	3 - A	Se muestra en la UI un mensaje de error.
	4 - B	El sistema registra un error al intentar almacenar el inicio del curso.
	5 - B	Se muestra en la UI un mensaje de error.
<b>Frecuencia esperada</b>	Poco frecuente.	



<b>Importancia</b>	Muy importante.
<b>Comentarios</b>	-

<b>Nombre del caso de uso</b>	<b>MOSTRAR LECCIÓN ACTUAL DEL CURSO</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitirle al estudiante reanudar un curso iniciado por el profesor.	
<b>Descripción</b>	El estudiante selecciona uno de los cursos iniciados, el sistema reanuda el curso desde el punto en que se lo abandonó mostrando en la UI la lección actual.	
<b>Precondición</b>	El profesor debe haber seleccionado algún estudiante con un curso asignado.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante selecciona uno de los cursos iniciados mostrados en la UI.
	2	El sistema reanuda el curso mostrando en la UI la última lección cursada por el alumno, si el curso está recién comenzado se muestra la primera lección.
<b>Postcondición</b>	La última lección actual curso se está mostrando en la UI.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	2 - A	El sistema registra un error al intentar recuperar la información de progreso del curso actual.
	3 - A	Se muestra en la UI un mensaje de error.
<b>Frecuencia esperada</b>	Frecuente.	
<b>Importancia</b>	Muy importante.	

<b>Comentarios</b>	Una lección consiste básicamente en una explicación del tema actual y un ejercicio para evaluar la comprensión del tema explicado.
--------------------	--

<b>Nombre del caso de uso</b>	<b>SOLICITAR EXPLICACIÓN BÁSICA</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitirle al estudiante solicitar al sistema una explicación más extensa que la mostrada por defecto en la UI.	
<b>Descripción</b>	El estudiante necesita una explicación más amplia para comprender el tema de la lección actual y la solicita interactuando con alguno de los elementos de la UI que componen la lección.	
<b>Precondición</b>	La lección actual del curso debe estarse mostrando en pantalla.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El estudiante interactúa con alguno de los elementos de la UI que componen la lección para solicitar más información acerca de este elemento.
	2	El sistema muestra una ventana emergente con una explicación más detallada que la mostrada por defecto.
<b>Postcondición</b>	Se está mostrando una ventana emergente, con una explicación, en la UI.	
<b>Frecuencia esperada</b>	Muy frecuente.	
<b>Importancia</b>	Importante.	
<b>Comentarios</b>	Cada elemento de la UI que compone la lección podría representar un concepto distinto.	

<b>Nombre del caso de uso</b>	<b>SOLICITAR EXPLICACIÓN AVANZADA</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitirle al profesor solicitar al sistema una explicación más avanzada que la mostrada al estudiante en la UI.	
<b>Descripción</b>	El profesor necesita una explicación para guiar al estudiante y la solicita interactuando con alguno de los elementos de la UI que componen la lección.	
<b>Precondición</b>	La lección actual del curso debe estarse mostrando en pantalla.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor interactúa con alguno de los elementos de la UI que componen la lección para solicitar una explicación avanzada acerca de este elemento.
	2	El sistema muestra una ventana emergente con una explicación más detallada que la mostrada por defecto.
<b>Postcondición</b>	Se está mostrando una ventana emergente, con una explicación, en la UI.	
<b>Frecuencia esperada</b>	Frecuente.	
<b>Importancia</b>	Medianamente importante.	
<b>Comentarios</b>	Cada elemento de la UI que compone la lección podría representar un concepto distinto.	

<b>Nombre del caso de uso</b>	<b>REALIZAR EVALUACIÓN ACTUAL</b>	
<b>Versión</b>	1.0	
<b>Objetivos Asociados</b>	Permitirle al estudiante resolver la evaluación de la lección actual.	

<b>Descripción</b>	El estudiante intenta resolver la evaluación que se muestra en la interfaz, si lo logra se pasa a la siguiente lección, y si no lo logra recibe información sobre su fracaso.	
<b>Precondición</b>	La lección actual del curso debe estarse mostrando en pantalla.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El alumno interactúa con algunos elementos de la UI que componen la lección para resolver la evaluación pertinente al tema actual.
	2	El alumno logra resolver satisfactoriamente la evaluación.
	3	El sistema pasa a la siguiente lección.
<b>Postcondición</b>	Se está mostrando una ventana emergente, con una explicación, en la UI.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	2 - A	El alumno no logra resolver satisfactoriamente la evaluación.
	3 - A	El sistema muestra en la UI información sobre el fracaso del alumno.
	4 - A	El alumno vuelve al paso 1.
<b>Frecuencia esperada</b>	Frecuente.	
<b>Importancia</b>	Importante.	
<b>Comentarios</b>	-	

<b>Nombre del caso de uso</b>	<b>CONSULTAR PROGRESO DEL ESTUDIANTE</b>
<b>Versión</b>	1.0
<b>Objetivos Asociados</b>	Permitirle al profesor pedirle al sistema información sobre la situación de cursado de un estudiante determinado.

<b>Descripción</b>	El profesor interactúa con la UI, la cual muestra un listado de todos los alumnos registrados, para obtener información sobre el progreso de cursado de un alumno determinado en todos sus cursos.	
<b>Precondición</b>	Debe haber algún perfil de estudiante creado.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El profesor selecciona del listado de alumnos un alumno determinado para solicitar información.
	2	El sistema muestra una ventana emergente con los datos de estado de cursado del alumno en todos sus cursos.
<b>Postcondición</b>	Se está mostrando una ventana emergente, con los datos de estado de cursado del alumno, en la UI.	
<b>Curso alternativo</b>	<b>Paso</b>	<b>Acción</b>
	2 - A	El sistema registra un error al intentar recuperar la información de progreso del alumno.
	3 - A	Se muestra en la UI un mensaje de error.
<b>Frecuencia esperada</b>	Poco frecuente.	
<b>Importancia</b>	Importante.	
<b>Comentarios</b>	-	

### *Diagramas De Secuencia*

Los siguientes diagramas de secuencia representan el comportamiento del sistema de forma dinámica, y siguiendo los lineamientos del proceso unificado de desarrollo están basados en las descripciones de casos de uso. Todos los diagramas de secuencia se han realizado utilizando el programa Dia.

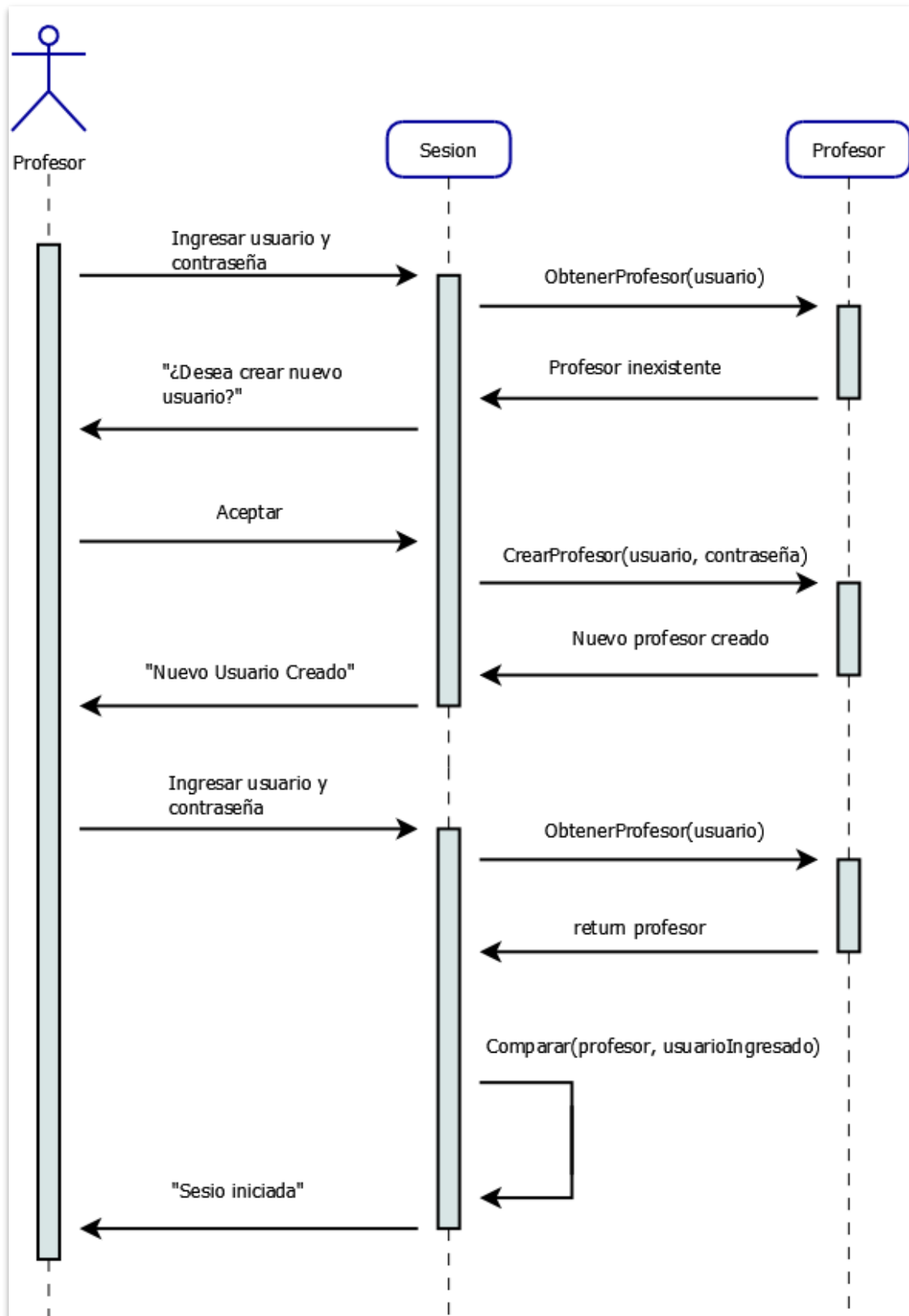


Diagrama de secuencia para el caso de uso *Iniciar Sesión*

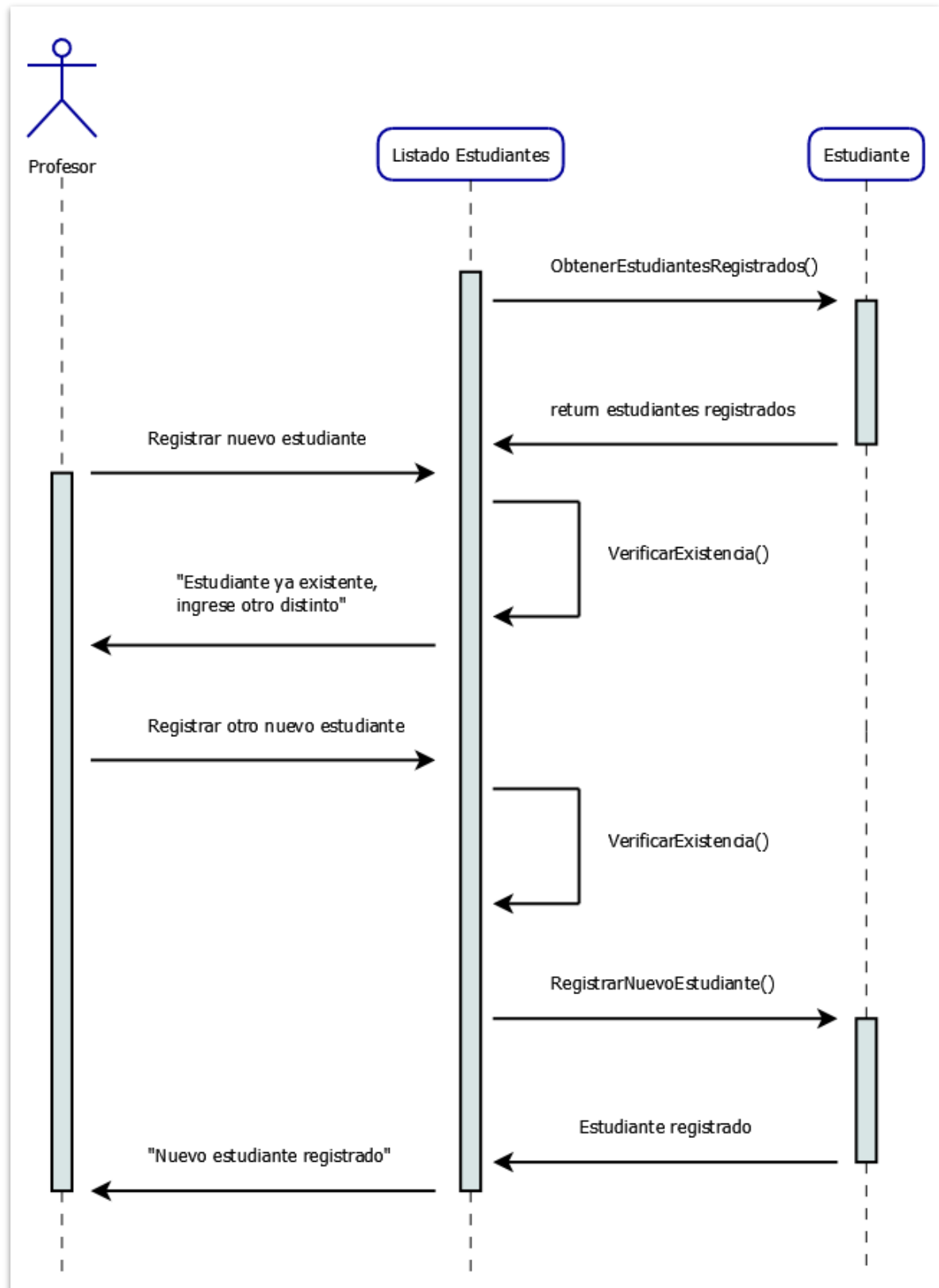


Diagrama de secuencia para el caso de uso **Registrar Nuevo Estudiante**

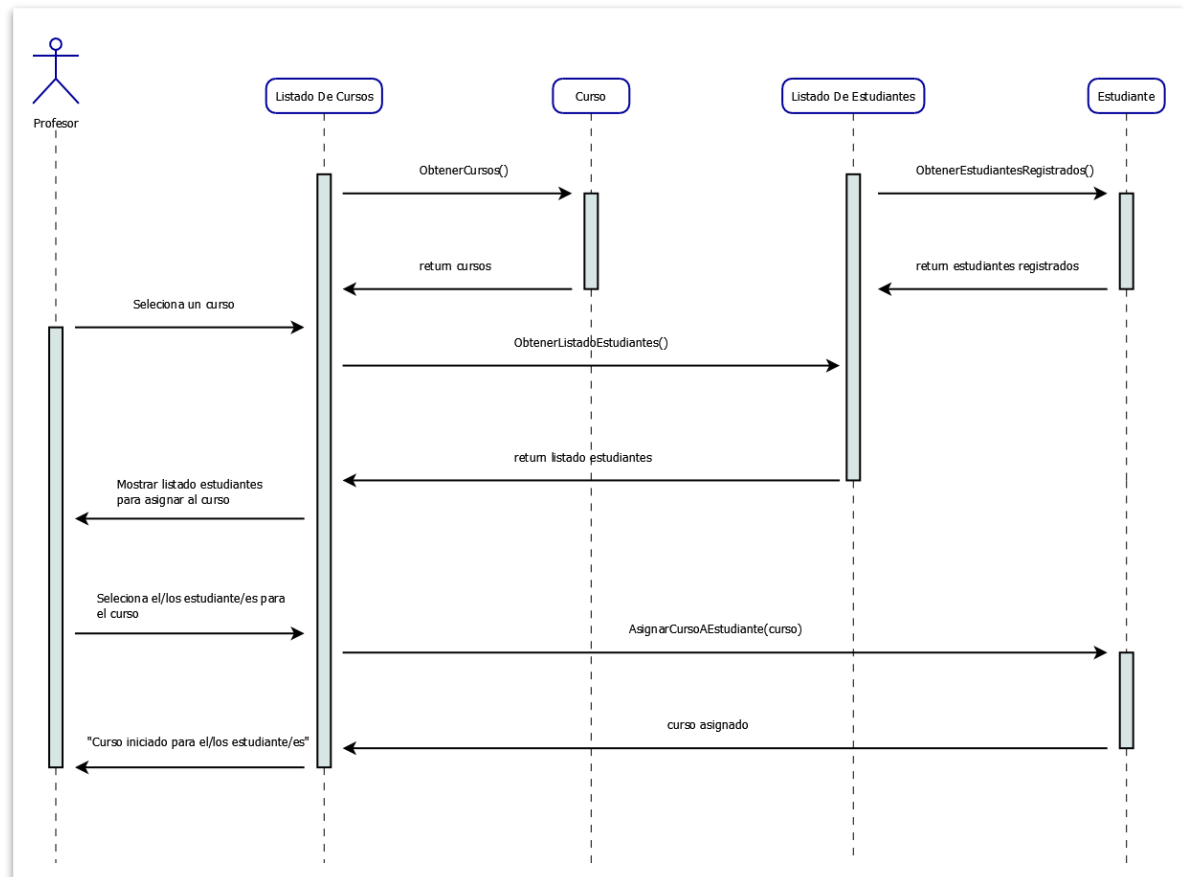
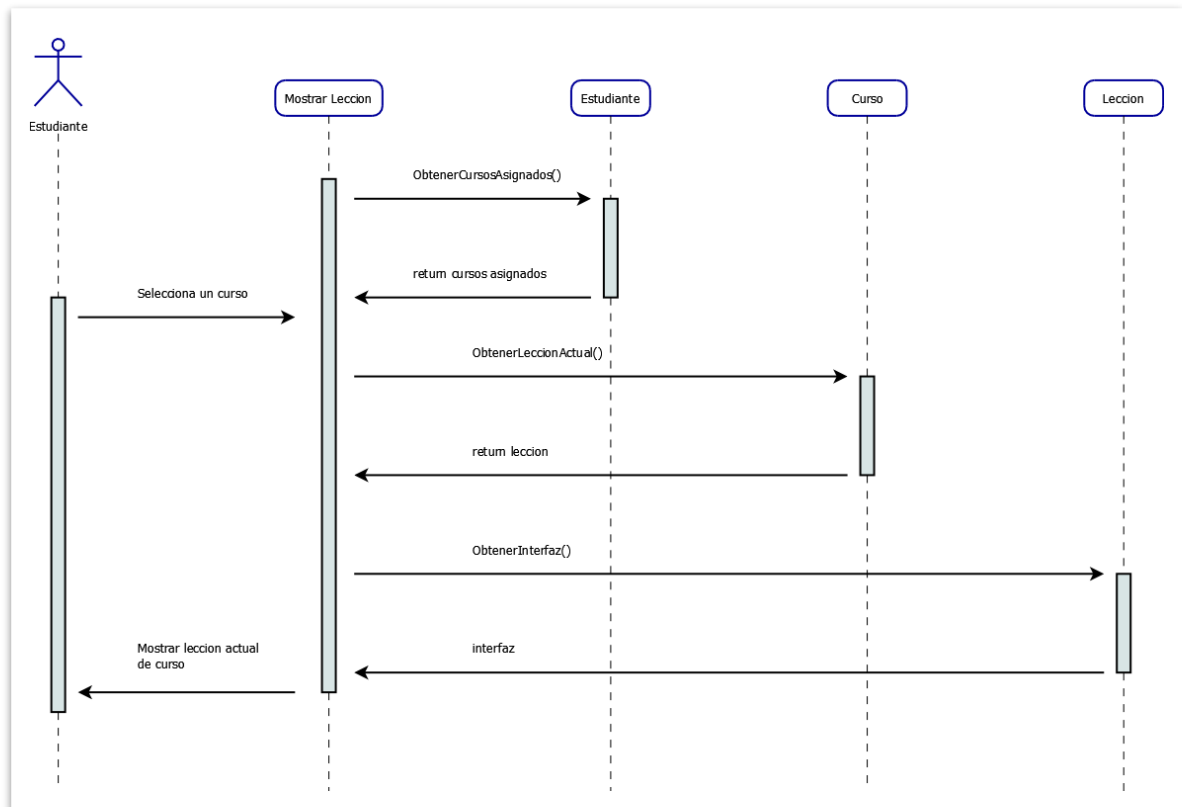
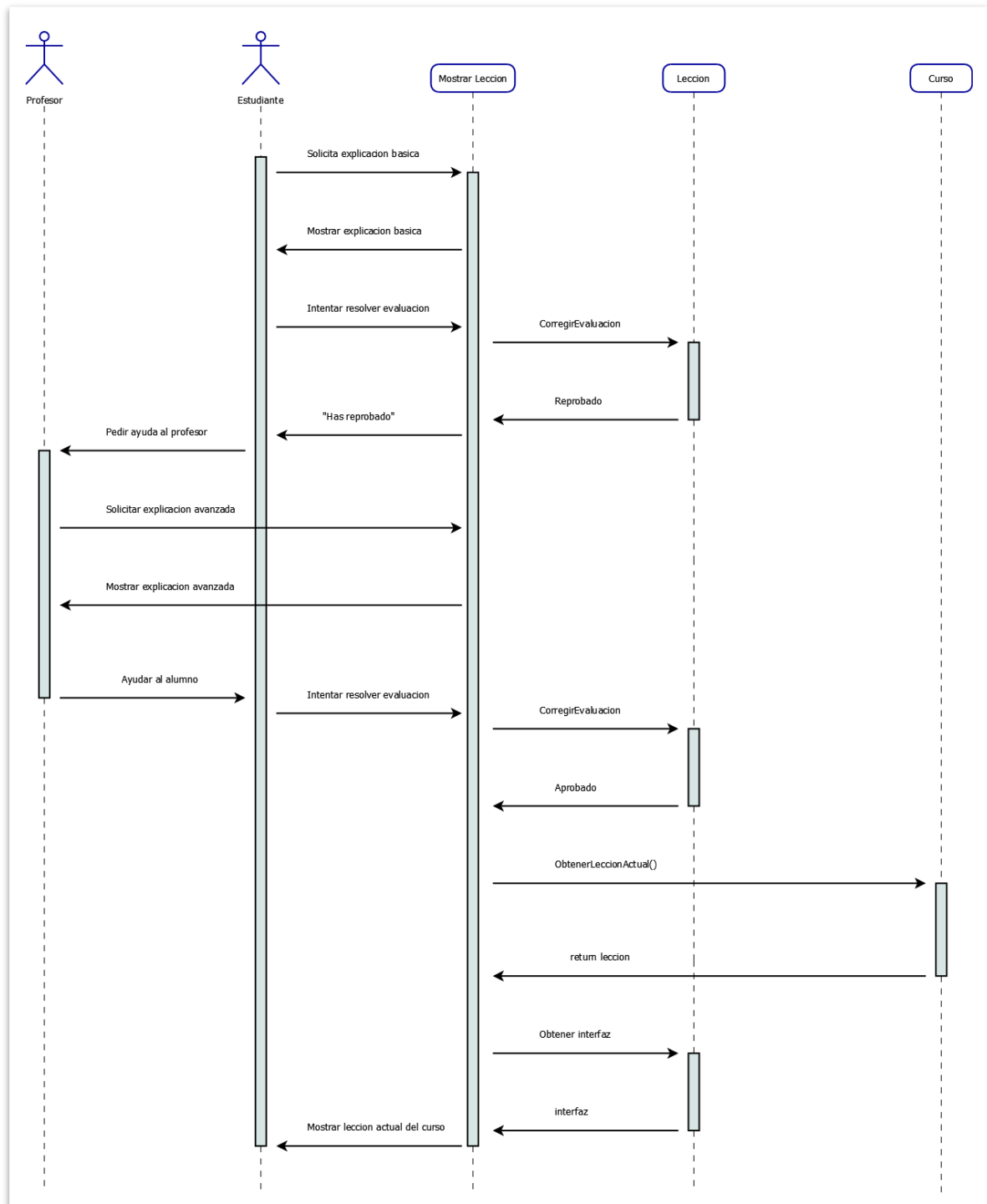


Diagrama de secuencia para el caso de uso **Iniciar Un Curso**





*Diagrama de secuencia para el caso de uso **Mostrar Lección Actual Del Curso***



*Diagrama de secuencia para los casos de uso Solicitar Explicación Básica, Solicitar Explicación Avanzada y Realizar Evaluación Actual.*

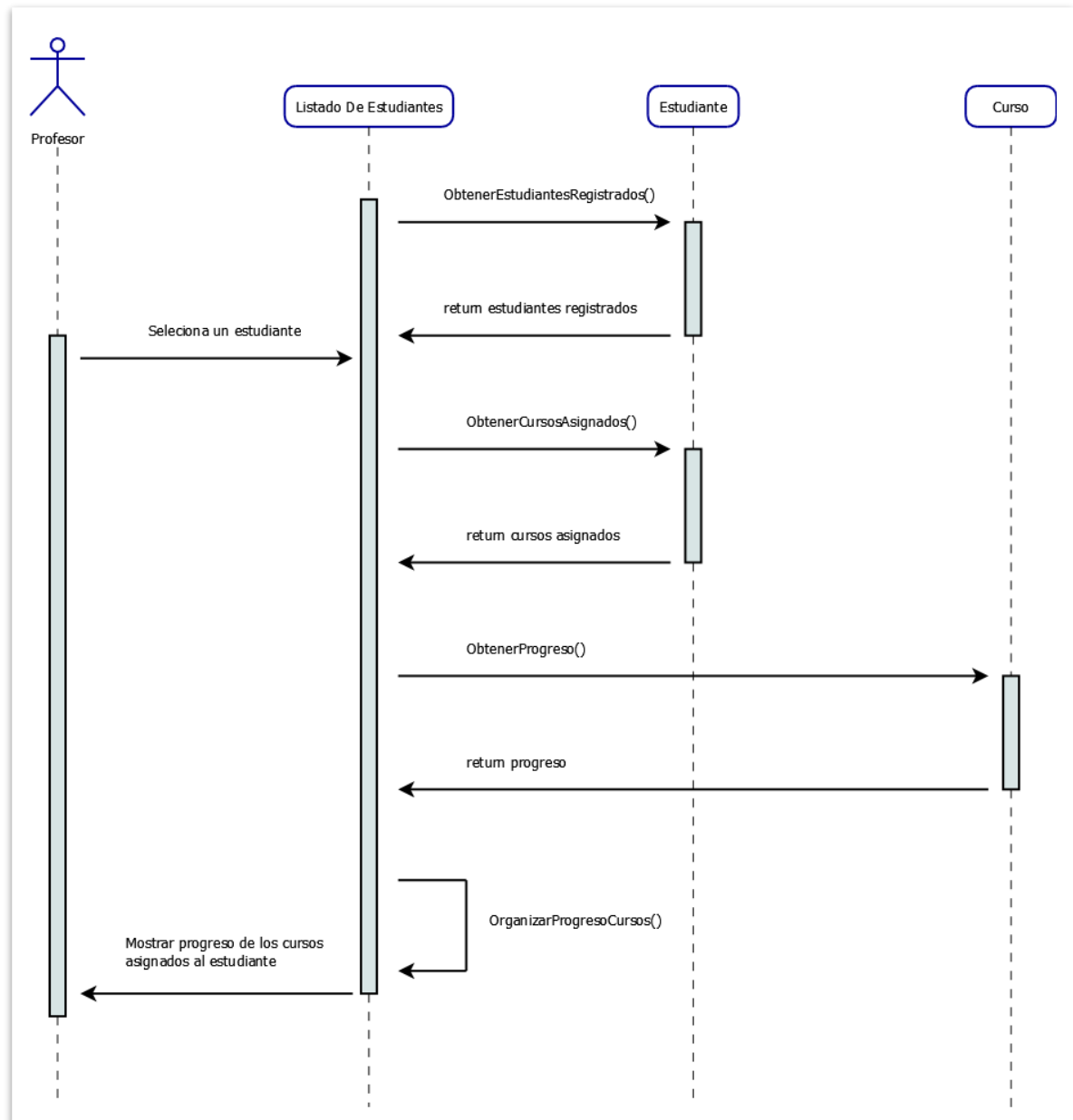


Diagrama de secuencia del caso de uso **Consultar Progreso Del Estudiante**

### Estructura De Datos

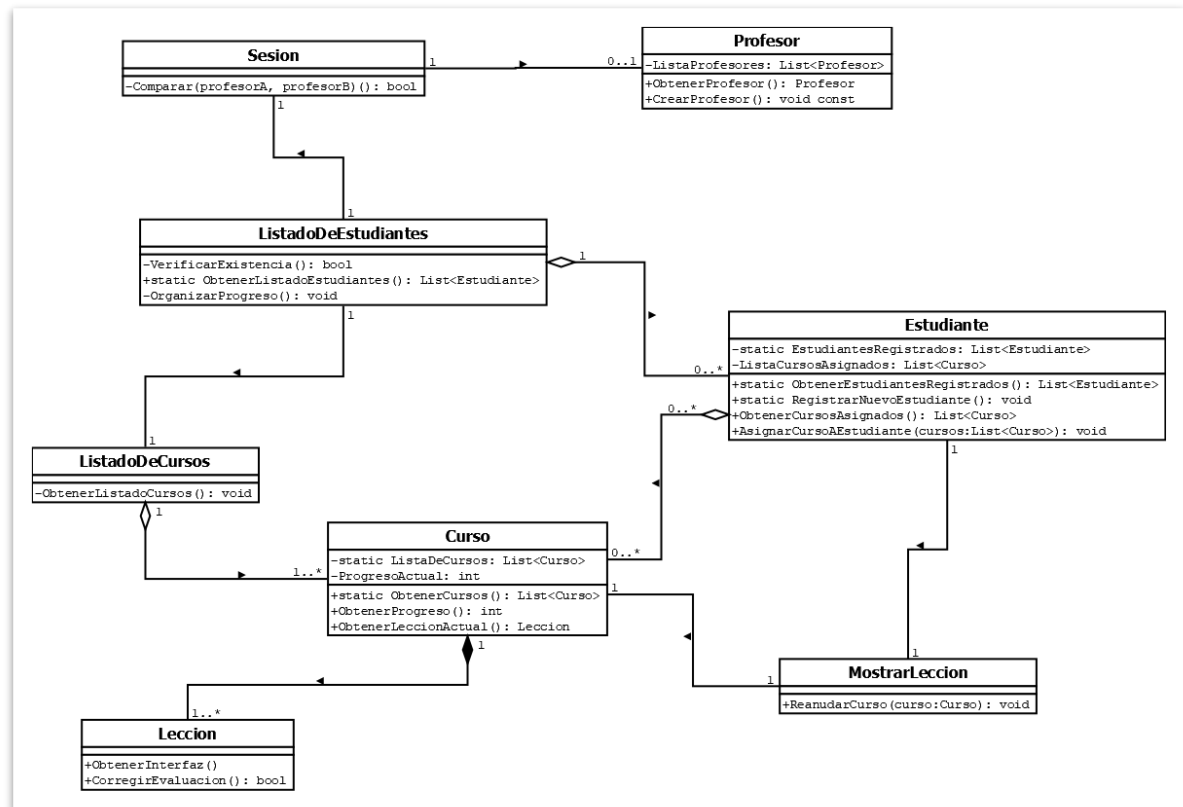


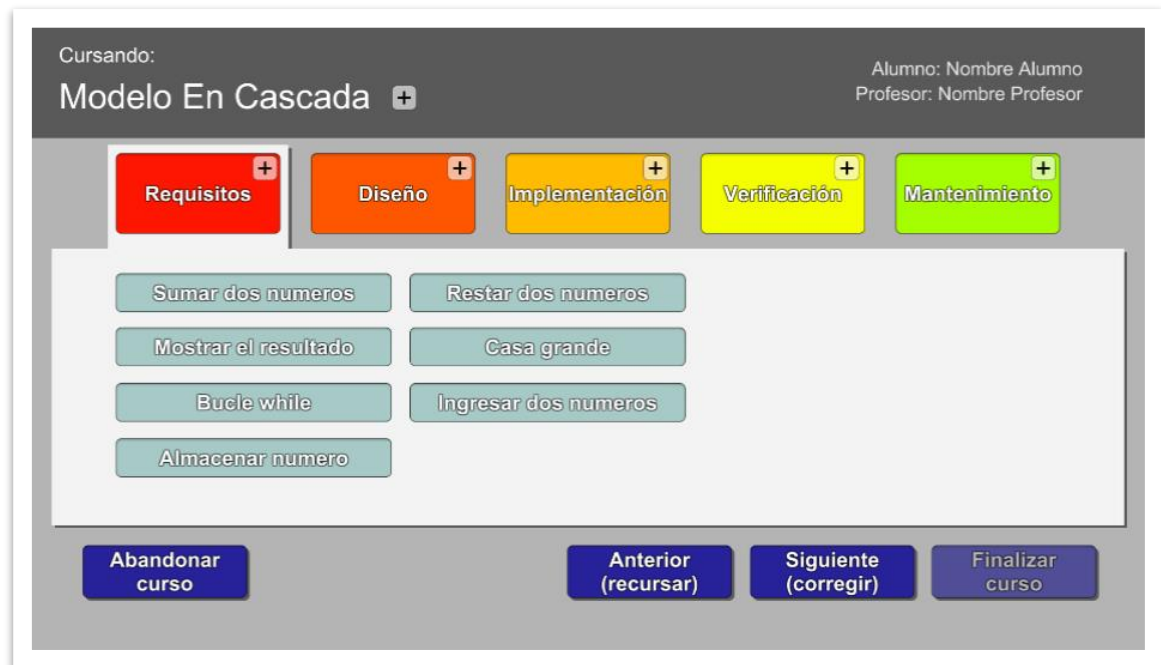
Diagrama de clases

### Prototipos De Interfaces De Pantallas

Los siguientes prototipos de interfaces son más simples que los que se usarán para el prototipo del sistema, es decir no implementan gráficos en tres dimensiones, por ejemplo, pero son útiles para ejemplificar parte del proceso de utilización normal del mismo y la configuración de los elementos de la interfaz, y cubren la realización de los siguientes casos de uso:

- Mostrar lección actual del curso.
- Solicitar explicación básica.
- Solicitar explicación avanzada.
- Realizar evaluación actual.

En este ejemplo, el profesor previamente inició el curso de Modelo en cascada y la primera lección se mostraría como en la **figura 1**, donde se comienza por la etapa de requisitos y se presenta una serie de ellos, alguno correctos y otros no.



*Figura 1*

Luego, en un caso hipotético, el estudiante está desorientado y no entiende que debe hacer en la etapa de requisitos, entonces hace clic en el símbolo + que se encuentra en el recuadro de requisitos y una ventana nueva muestra una explicación al respecto, **figura 2**.

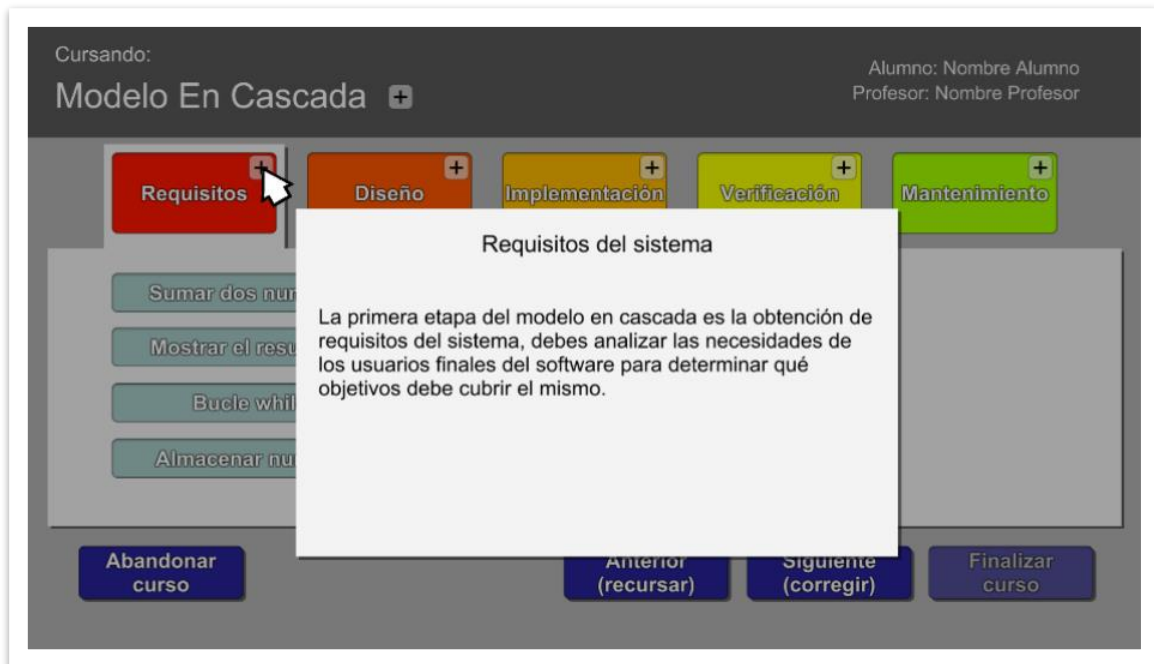


Figura 2

Ahora, el estudiante ya entiende lo que debe hacer y elige varios requisitos para su aplicación, **figura 3**.

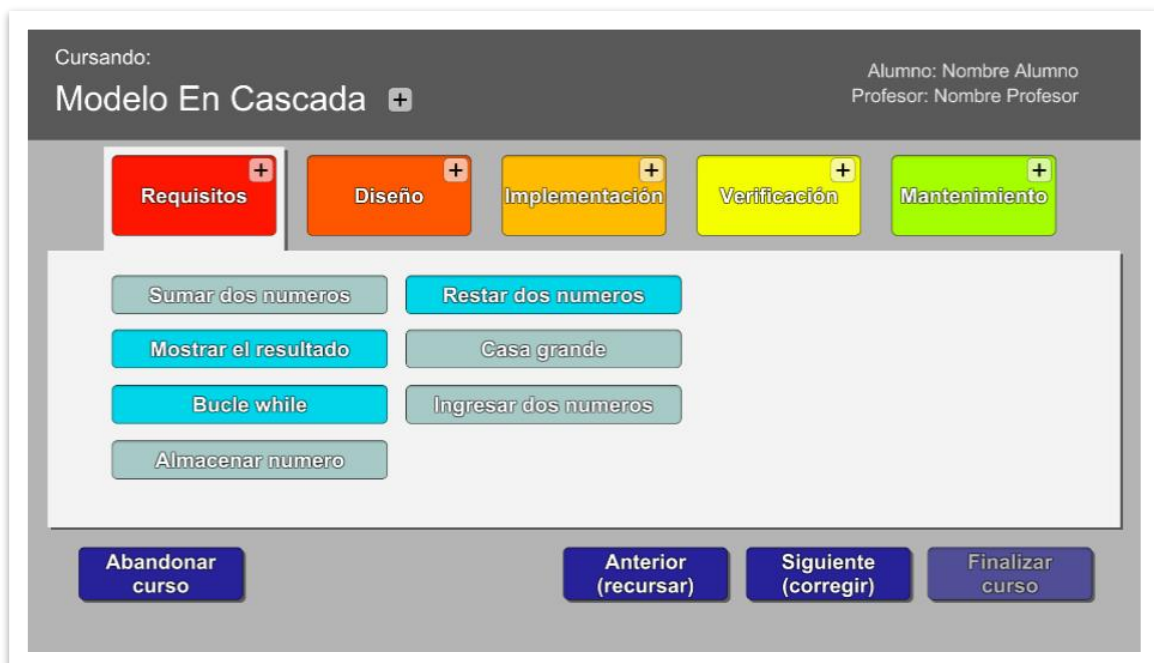


Figura 3

Luego, el estudiante hace clic en el botón **Siguiente** y recibe una corrección por parte del sistema: algunos requisitos están bien, pero se ha equivocado en otros, **figura 4**.

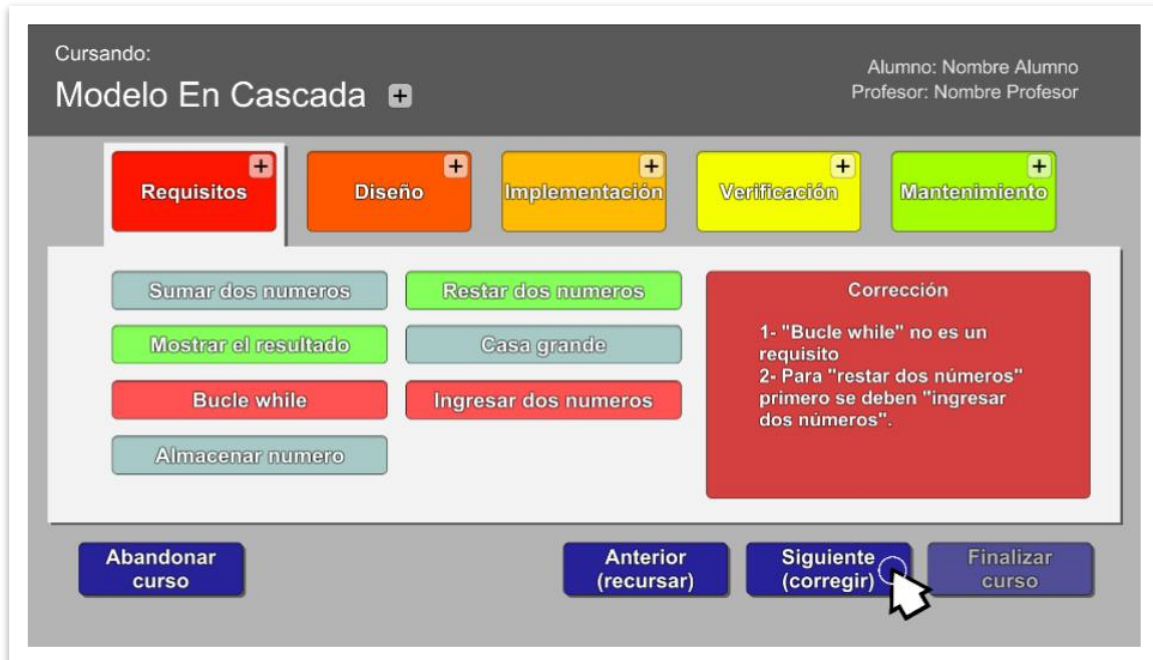


Figura 4

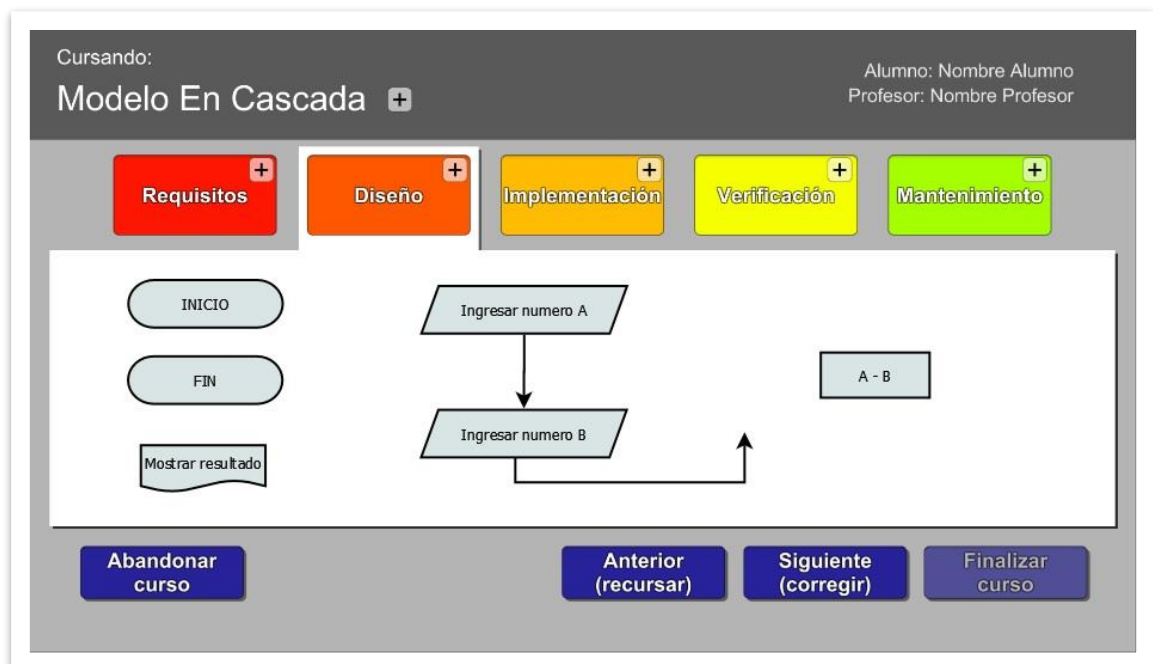


Figura 5

Finalmente, después de que el estudiante seleccionara unos requerimientos correctos e hiciera clic en el botón Siguiente se pasa a la etapa de requerimientos. En la **figura 5** se muestra como el estudiante está intentando resolver esta nueva etapa. Note que la etapa de diseño es generada a partir de los requerimientos seleccionados en la etapa anterior.

## Seguridad

### *Acceso A La Aplicación*

Para empezar, la aplicación se descargará en las computadoras personales de los usuarios y **no accederá a internet** para ninguna de sus funciones.

En el caso de una enseñanza en una institución educativa la aplicación será descargada, principalmente, en las computadoras personales ofrecidas por la propia institución.

En el caso de un aprendizaje autodidacta la aplicación será descargada en la propia computadora personal del estudiante autodidacta.

Para esta aplicación hay dos tipos de usuarios identificados:

- Administradores
- Usuario común

Los **administradores** son el profesor, en el caso de una educación tutorizada, y el estudiante autodidacta, en el caso de que el estudiante decida hacerse cargo de su aprendizaje.

Los **usuarios comunes** son los alumnos.

Los usuarios comunes pueden iniciar sesión y tienen acceso a la realización de los cursos, es decir, a las explicaciones y evaluaciones contenidas en ellos.

Además de todo el acceso de un usuario común los administradores tienen también el poder de registrarse en el sistema, registrar a los estudiantes, asignar cursos a



cada estudiante y consultar el progreso de cada estudiante en cada uno de sus cursos asignados.

Para la creación de perfiles de usuario tanto de administrador como de usuario común se deben cumplir una serie de restricciones:

- El nombre de usuario debe contener más de tres caracteres.
- La contraseña debe contener más de seis caracteres alfanuméricos.

Además, para la elección del nombre de usuario y contraseña se les sugiere a los administradores no utilizar información que se considere privada, ya que no es necesario y pondría en riesgo dicha información. Es recomendable utilizar nombres de usuario ficticios y contraseñas generadas aleatoriamente.

Como la aplicación no accede a internet los nombres de usuario y contraseña se almacenan localmente, y encriptados utilizando librerías propias del lenguaje C# para tal propósito, en un archivo creado por la aplicación.

### *Política De Respaldo De Información*

En su primer uso la aplicación crea una carpeta donde almacenar la información en forma de archivos. El perfil de usuario administrador se almacena en un archivo único. El perfil de estudiante, sus cursos asignados y su progreso se almacenan en otro archivo, y se crea un archivo distinto por cada estudiante.

La aplicación guarda los datos localmente utilizando la serialización, disponible con C#, la cual permite convertir objetos en un flujo de *bytes* para, en este caso, ser convertidos en un archivo y ser almacenados en un dispositivo de almacenamiento permanente.

En el caso del almacenamiento de un perfil de usuario, primero se encriptan el nombre de usuario y la contraseña antes de la serialización. El proceso de encriptado se realizará utilizando la interfaz `IDataProtector` de C#.

En el caso del guardado del progreso del estudiante, este se realiza cada vez que el alumno realiza un progreso en el curso, es decir, cada vez que el estudiante aprueba una evaluación dentro de un curso.

## Análisis De Costos

Para calcular el costo total del proyecto se tomaron en cuenta los costos en recursos humanos y del hardware y software necesarios para el desarrollo, y el coste de la publicación de la aplicación en una tienda de software en línea.

Para tener una idea más precisa y realista de los costos se calcularon dos escenarios para los costos en recursos humanos, uno **optimista** y otro **pesimista**, ya que estos costos pueden ser muy diferentes de los calculados debido principalmente a tiempos de desarrollo, cantidad de personas necesaria y honorarios mensuales diferentes a los calculados. Para los costos de hardware y software necesarios para el desarrollo y el coste de la publicación de la aplicación se realizará un único cálculo ya que se han considerado más predecibles.

Todos los costos están expresados en pesos argentinos, pero debido a los altos niveles de depreciación de esta moneda con el paso del tiempo, se incluye aquí el valor de un dólar estadounidense para la venta según el Banco de la Nación Argentina al día 21/10/2022 que es 1 USD = 159 ARS.

A continuación se muestra el costo en recursos humanos teniendo en cuenta el rol que ocupará la persona, los honorarios mensuales y la cantidad de meses en que prestará su servicio.

En el escenario pesimista se supuso que para lograr un nivel de calidad aceptable es necesario un tiempo de desarrollo de seis meses y la contratación de personal de alto nivel profesional.

Costo en recursos humanos: escenario pesimista			
Cargo	Remuneración Mensual	Meses	Subtotal

Analista Programador Sr.	\$230.183	4	\$920.732
Jefe de proyectos	\$247.837	6	\$1.487.025
Diseñador UI	\$200.947	2	\$401.894
Analista QA	\$220.480	3	\$661.440
<b>Total</b>			<b>\$3.471.091</b>

En el escenario optimista se supuso que para lograr un nivel de calidad aceptable es necesario un tiempo de desarrollo de cuatro meses y la contratación de personal de nivel profesional medio.

<b>Costo en recursos humanos: escenario optimista</b>			
<b>Cargo</b>	<b>Remuneración Mensual</b>	<b>Meses</b>	<b>Subtotal</b>
Analista Jr.	\$172.630	2	\$345.260
Programador Aplicaciones Jr.	\$146.468	3	\$439.404
Jefe de proyectos	\$247.837	4	\$991.348
Analista QA	\$220.480	2	\$440.960
<b>Total</b>			<b>\$2.216.972</b>

Todos los datos de remuneración mensual han sido obtenidos de la página web del Consejo Profesional De Ciencias Informáticas de la provincia de Córdoba el día 21/10/2022 (CPCIPC, 2022).

Para el desarrollo de la aplicación se utilizarán los programas de uso gratuito Unity 2021 LTS, Visual Studio Community 2022, y el programa de pago Adobe XD. El hardware utilizado para el desarrollo fue elegido para que cumpla sobradamente con los requisitos de hardware recomendados para todos los programas utilizados en el desarrollo. El costo del sistema operativo no se incluye en la tabla ya que viene previamente instalado en los equipos e incluido en su precio final.

<b>Costos en hardware y software para el desarrollo</b>			
<b>Producto</b>	<b>Precio</b>	<b>Cantidad</b>	<b>Subtotal</b>
Adobe XD (por dos meses)	\$1.346	1	\$1.346
Lenovo ThinkCentre M75s 2da Gen	\$196.999	4	\$787.996
Monitor Led 24 pulgadas Samsung	\$51.299	4	\$205.196
<b>Total</b>			<b>\$994.538</b>

Todos los precios de hardware y software son del día 22/10/2022. El precio de Adobe XD fue extraído de [commerce.adobe.com](https://commerce.adobe.com). El precio de las computadoras Lenovo fue extraído de [lenovo.com](https://lenovo.com). El precio de los monitores Samsung fue extraído de [shop.samsung.com](https://shop.samsung.com).

Para finalizar, se incluye en el cálculo de los costos finales el costo de publicación de la aplicación en la tienda de Steam, que son 15.900 pesos, precio obtenido de [partner.steamgames.com](https://partner.steamgames.com).

<b>Costos finales en ambos escenarios</b>				
<b>Escenarios</b>	<b>Recursos humanos</b>	<b>Hardware y software</b>	<b>Publicación en tienda</b>	<b>Totales</b>
Pesimista	\$3.471.091	\$994.538	\$15.900	\$4.481.529
Optimista	\$2.216.972	\$994.538	\$15.900	\$3.227.410

## **Análisis De Riesgos**

En la siguiente tabla se muestra un listado de los riesgos del proyecto identificados con su probabilidad de ocurrencia aproximada, el momento del proyecto donde se espera que ocurran, y su nivel de impacto aproximado.

<b>Riesgos identificados del proyecto</b>					
<b>Id</b>	<b>Tipo</b>	<b>Riesgo</b>	<b>Probabilidad de ocurrencia</b>	<b>Momento de ocurrencia</b>	<b>Nivel de impacto</b>
1	Proyecto	El tiempo de desarrollo se extiende debido a la dificultad técnica del proyecto	Media	A la mitad o cerca del final del desarrollo	Alto
2	Proyecto	Se detectan muchos errores en la aplicación cerca del final del desarrollo	Media	Al final del desarrollo	Medio
3	Proyecto	El personal seleccionado se declara técnicamente incapaz de continuar con el desarrollo	Baja	A la mitad del desarrollo	Alto
4	Proyecto	Abandono del proyecto por parte de algunos miembros del personal	Baja	Durante todo el desarrollo	Alto
5	Producto	El producto final, la aplicación, no despierta el interés de los usuarios finales	Media	Luego de finalizado el desarrollo	Alto

6	Producto	El producto final, la aplicación, no cumple con las expectativas luego de ser utilizado por un largo tiempo	Media	Luego de finalizado el desarrollo	Alto
7	Proyecto	El costo real del proyecto es mucho mayor que el calculado debido a la depreciación de la moneda argentina	Alta	A la mitad o cerca del final del desarrollo	Alto

En base a los riesgos identificados se ha calculado el grado de exposición en base a la probabilidad de ocurrencia y al impacto esperado de los mismos. Además, se han ordenado los riesgos identificados desde el grado de exposición mayor al menor.

<b>Análisis cuantitativo y grado de exposición</b>			
<b>Riesgo</b>	<b>Probabilidad de ocurrencia</b>	<b>Efecto o impacto</b>	<b>Grado de exposición</b>
El costo real del proyecto es mucho mayor que el calculado debido a la depreciación de la moneda argentina	90%	5	4,5
El tiempo de desarrollo se extiende debido a la dificultad técnica del proyecto	70%	4	2,8
El producto final, la aplicación, no despierta el interés de los usuarios finales	50%	5	2,5

El producto final, la aplicación, no cumple con las expectativas luego de ser utilizado por un largo tiempo	60%	4	2,4
Se detectan muchos errores en la aplicación cerca del final del desarrollo	50%	3	1,5
Abandono del proyecto por parte de algunos miembros del personal	25%	4	1
El personal seleccionado se declara técnicamente incapaz de continuar con el desarrollo	15%	5	0,75

Los riesgos con un grado de exposición mayor son los que tienen mayores probabilidades de causar un gran daño durante el desarrollo del proyecto.

En la siguiente tabla se listan los riesgos con su correspondiente plan de contingencia para prevenirlos o para aplicar en caso de que se materialicen. Se ha mantenido el orden según su grado de exposición. Los riesgos con mayor grado de exposición son los que deben ser atendidos con mayor urgencia. El responsable de detectar la materialización de los riesgos y de dirigir la ejecución de los planes de contingencia será el jefe de proyectos.

<b>Plan de contingencia para los riesgos</b>	
<b>Riesgo</b>	<b>Plan de contingencia</b>
El costo real del proyecto es mucho mayor que el calculado debido a la depreciación de la moneda argentina	Recalcular lo antes posible el presupuesto teniendo en cuenta la depreciación de la moneda argentina en periodos anteriores
El tiempo de desarrollo se extiende debido a la dificultad técnica del proyecto	Desviar parte del presupuesto para contratar personal de muy alto nivel para actividades críticas del desarrollo, como el diseño y construcción del software.

El producto final, la aplicación, no despierta el interés de los usuarios finales	Obtener realimentación de un grupo de usuarios seleccionado a través de la presentación de versiones parciales de la aplicación creadas durante el desarrollo
El producto final, la aplicación, no cumple con las expectativas luego de ser utilizado por un largo tiempo	Mantener un seguimiento de las opiniones de los usuarios a través del tiempo y realizar actualizaciones periódicas a la aplicación
Se detectan muchos errores en la aplicación cerca del final del desarrollo	Retrasar ligeramente la fecha de entrega de la aplicación y luego continuar realizando actualizaciones inmediatamente después del lanzamiento de la aplicación para resolver los errores urgentemente
Abandono del proyecto por parte de algunos miembros del personal	Negociar previamente con personal capacitado para mantenerlos en espera y contratar en caso de que un puesto se libere
El personal seleccionado se declara técnicamente incapaz de continuar con el desarrollo	Negociar previamente con personal de mayor nivel técnico para mantenerlos en espera y para contratar en caso de que esto ocurra



## Conclusiones

En este trabajo se llevó a cabo un proyecto para el desarrollo de una aplicación interactiva para la enseñanza de metodologías de desarrollo de software a jóvenes. El motivo inicial para llevar a cabo este proyecto surgió al momento de notar que la educación hacia los jóvenes actualmente hace mucho énfasis en formar en el área de la programación y se dejan de lado las otras áreas igual de importantes en la práctica, y se llegó a la conclusión de que es necesaria una educación integral en desarrollo de software para aportar una visión más cercana a la realidad en el estudiante, y para lograrlo se propuso como objetivo el desarrollo de una aplicación interactiva donde se puedan adquirir ciertos conocimientos fundamentales de manera teórico y práctica.

Considero que los objetivos se cumplieron de forma aceptable. Con el análisis de las diferentes maneras mediante las cuales un estudiante adquiere conocimientos en el área del desarrollo de software se diseñó una única solución teniendo en cuenta sus costos monetarios y sus riesgos durante el desarrollo. Además, con el desarrollo de un prototipo se pusieron a prueba las funciones más importantes del sistema propuesto como solución. Sin embargo, los resultados de la utilización de la aplicación interactiva propuesta en la práctica en un entorno real darían el veredicto sobre su grado de efectividad y sobre si es recomendable utilizarla o no.

## Demo

Se ha desarrollado una versión del prototipo diseñado y se puede acceder a sus archivos a través del siguiente enlace:

[https://drive.google.com/drive/folders/1WiOY3esLrpEcl6rdcm8nR6FWq8gsnPTL?usp=share\\_link](https://drive.google.com/drive/folders/1WiOY3esLrpEcl6rdcm8nR6FWq8gsnPTL?usp=share_link)

## Referencias

- Pressman, Roger S. (2010). *Ingeniería Del Software, Un Enfoque Práctico*. McGraw-Hill Interamericana Editores
- Unity Technologies. (s.f.). *¿Por qué Unity?* Obtenido de <https://store.unity.com/es/products/unity-personal>
- Microsoft. (2022). *Paseo por el lenguaje C#*. Obtenido de <https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- Fundación Scratch. (s.f.). *Acerca de Scratch*. Obtenido de <https://scratch.mit.edu/about>
- Code.org. (s.f.). *About us*. Obtenido de <https://code.org/about>
- Kodable. (2022). *¿Por qué Kodable?* Obtenido de <https://www.kodable.com/benefits>
- Rumbaugh James, Jacobson Ivar, Booch Grady. (2007). *El Lenguaje Unificado De Modelado, Manual De Referencia*. Pearson Educación
- Consejo Profesional De Ciencias Informáticas de la provincia de Córdoba. (2022). *Honorarios Recomendados*. Obtenido de <https://www.cpcipc.org.ar/honorarios-recomendados/>