

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Indice

1 MANUAL DE PROCEDIMIENTOS DE PRODUCCIÓN DE SOFTWARE DE ON S.R.L.	3
1.1 Aclaraciones	3
2 FLUJOGRAMA DEL PROCESO DE PRODUCCIÓN DE SOFTWARE	4
2.1 Descripción del Flujograma	5
2.2 Alcance	5
2.3 Roles	6
3 DESCRIPCIÓN DE LOS EVENTOS INVOLUCRADOS EN EL PROCESO	7
3.1 ADQUISICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS	7
3.1.1 Entregables	7
3.1.1.1 Minuta de relevamiento	7
3.1.1.2 Modelo de casos de uso estructurado	9
3.1.1.3 Análisis de riesgo de requerimientos	13
3.2 ANÁLISIS	14
3.2.1 Elementos requeridos	15
3.2.2 Entregables	15
3.2.2.1 Modelo de análisis	15
3.3 DISEÑO	19
3.3.1 Elementos requeridos	19
3.3.2 Entregables	20
3.3.2.1 Modelo de diseño	20
3.3.2.2 Modelo de implementación	24
3.3.2.3 Planeamiento de testing	26
3.3.2.4 Análisis de riesgo de técnico	27
3.4 CONSTRUCCIÓN	27
3.4.1 Elementos requeridos	27
3.4.2 Entregables	28
3.4.2.1 Componentes en funcionamiento	28
3.4.2.2 Pruebas de unidad automáticas	28

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.4.2.3 Documentación del usuario.....	29
3.5 TESTING	31
3.5.1 Elementos requeridos	31
3.5.2 Entregables.....	31
3.5.2.1 Ejecución de testing	31
3.5.2.2 Evaluación de testing	31
3.6 IMPLEMENTACIÓN	33
3.6.1 Entregables.....	33
3.6.1.1 Plan de implementación.....	33
4 ANTEPROYECTO.....	35
5 MÉTRICAS.....	37
5.1 Estimación de costos para adquisición y especificación de requerimientos e implementación.....	37
5.2 Estimación de costos para análisis, diseño, construcción y testing.....	37
6 ANEXOS.....	38
6.1 Documentos de testing	38
6.2 Análisis de riesgo	39
6.3 Convenciones de codificación	42
6.3.1 Recomendaciones generales.....	42
6.3.2 Declaraciones (statements)	42
6.3.3 Indentado	42
6.3.4 Clases	43
6.3.5 Funciones y Métodos:.....	43
6.3.6 Variables.....	46
6.3.6.1 Casos particulares	47
6.3.7 Glosario.....	48
6.3.7.1 Scope.....	48
6.4 Herramientas de desarrollo.....	48

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

1 MANUAL DE PROCEDIMIENTOS DE PRODUCCIÓN DE SOFTWARE DE ON S.R.L.

1.1 **Aclaraciones**


Este apartado describe los entregables esperados al finalizar cada una de los flujos de trabajo de un ciclo de desarrollo de software.

Para la interpretación de las descripciones incluidas en este documento y la posterior confección de entregables deberá tenerse en cuenta que:

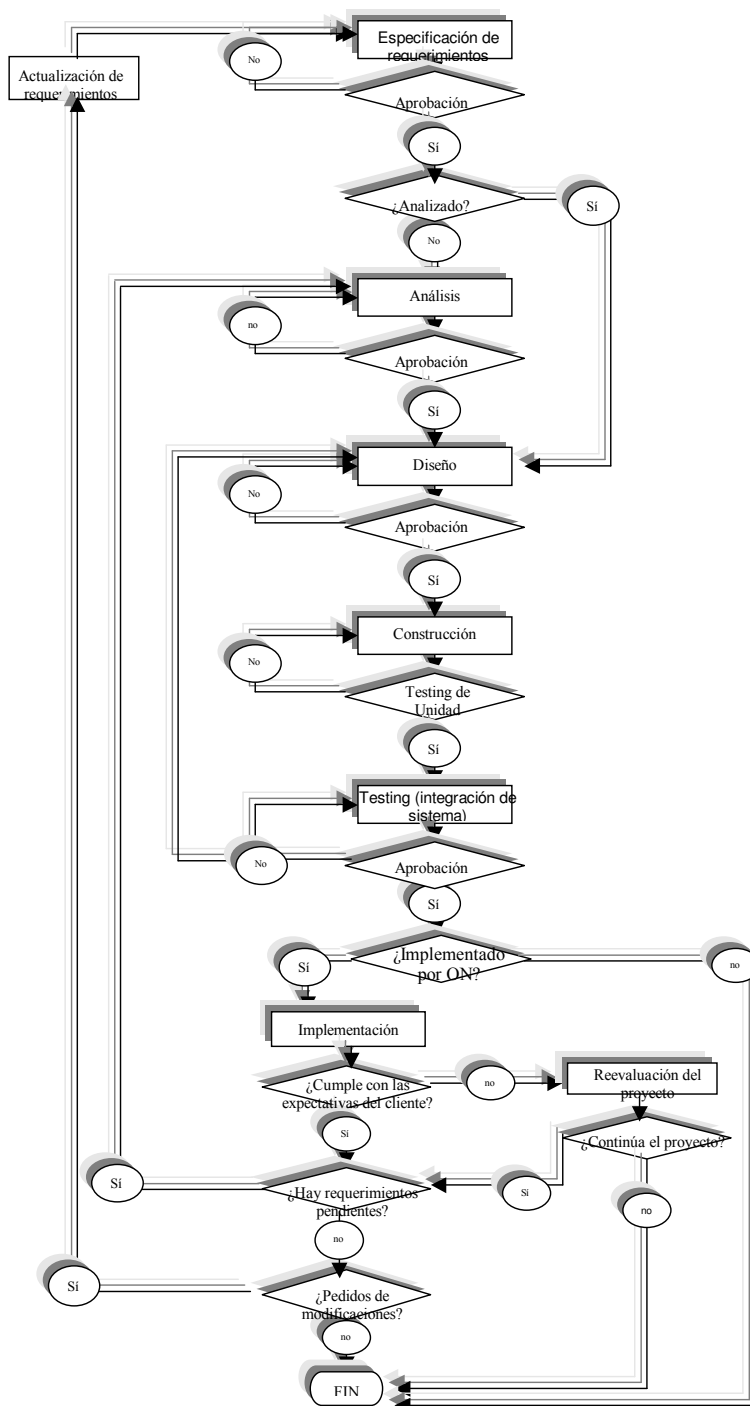
El lenguaje utilizado para la especificación de software es UML (Unified Modeling Language).


La plataforma de implementación se especificará y acordará para cada proyecto en forma particular, según la necesidad del cliente, los lenguajes pueden ser, .NET, JAVA, Visual Basic, Visual Fox Pro, etc.

La herramienta de codificación es GENEXUS.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción	
		13/07/07			
		Actualización			
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor	
1		1	13/07/07	R. Ortiz M. Ortiz	

2 FLUJOGRAMA DEL PROCESO DE PRODUCCIÓN DE SOFTWARE



	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

2.1 Descripción del Flujoograma


Considerando que el diagrama representa fielmente el proceso que sigue el desarrollo de un software, y que más adelante se describirá cada una de las actividades relacionadas en el proceso diagramado, sólo se hará mención de algunas aclaraciones:

- Cada una de las actividades \ eventos son un flujo de trabajo.
- Por cada fase del desarrollo de un sistema, la secuencia de flujos que plantea el diagrama puede ejecutarse una o más veces.
- Al finalizar cada una de las fases, los detalles descriptos en este documento deben ser evaluados para determinar si cumplen con el nivel de detalle que la fase finalizada requiere.
- El nivel de detalle que debe alcanzar cada documento en cada fase será acordado al realizar la planificación del proyecto.

2.2 Alcance

De los flujos de trabajo que se muestran en el Diagrama 1, el presente documento describe los siguientes eventos:

- Adquisición y especificación de requerimientos
- Análisis
- Diseño
- Construcción

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


- Testing
- Implementación

2.3 Roles

De acuerdo al proceso de desarrollo definido por la empresa, se establecen los siguientes roles a fin de cumplir con el mismo:

- Jefe de Desarrollo
- Encargado de relevamiento
- Desarrollador Senior de Genexus
- Implementación y Testing

Cabe destacar que una persona puede cumplir más de un rol en un mismo proyecto. Los roles deben ser definidos al comienzo de cada proyecto.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3 DESCRIPCIÓN DE LOS EVENTOS INVOLUCRADOS EN EL PROCESO

3.1 ADQUISICIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS

Durante este flujo de trabajo se recopilan y documentan los requerimientos del producto de software a desarrollar y se identifican los actores que harán uso del mismo.

Los requerimientos funcionales recopilados deben estar organizados en *casos de uso* y asociados en un *diagrama de casos de uso* a los actores que los ejecutarán.

Los requerimientos no funcionales se organizarán en notas que estarán visibles en el *diagrama de casos de uso*.

3.1.1 Entregables


3.1.1.1 Minuta de relevamiento

Objetivos

1. Documentar las reuniones de relevamiento realizadas.
2. Tener un documento que exprese el acuerdo entre los participantes de la reunión acerca de lo discutido en la misma.
3. Generar material que sirva de base para crear el modelo de casos de uso.

Contenido


El contenido de la minuta será el siguiente:

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- Nombre del proyecto.
- Número de versión.
- Módulo (si corresponde).
- Nro. de Minuta (secuencial para todo el proyecto).
- Fecha y hora.
- Lugar físico de la reunión.
- Lista de participantes.
- Objetivos y alcance.
- Historial de las modificaciones para cada versión.
- Descripción de lo discutido durante la reunión haciendo énfasis en los acuerdos alcanzados.
 - Lista de temas previstos en la agenda de la reunión que no se tocaron o se tocaron de forma incompleta.
 - Sección Anexos en donde se incluya material relacionado o presentados durante la reunión. (Se intentará que la sección Anexos sea lo más breve posible, separando en otros documentos los anexos muy extensos e incluyendo sólo la referencia al documento).

Entrega

Cada minuta deberá entregarse con la firma de cada participante que pruebe su acuerdo con el contenido de la misma. El original firmado se archivará en la carpeta del proyecto.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.1.1.2 Modelo de casos de uso estructurado

Objetivos

1. Centralizar toda la documentación de requerimientos.
2. Contener información suficiente para:
3. Diseñar las pruebas de sistema.
4. Diseñar las pruebas de aceptación.
5. Analizar, Diseñar y construir el sistema.

Contenido

Lista de Actores

Para cada uno de los *actores* que interactuará con el sistema se definirá un nombre y se redactará una descripción.


El nombre del *actor* se definirá utilizando el siguiente formato: **A{id} – {Nombre actor}**. Donde {id} es un número correlativo único para cada *actor* y {Nombre *actor*} es un nombre descriptivo único para cada *actor*.

La descripción del *actor* deberá estar claramente asociada al nombre y expresar de forma concisa el rol que cumple el *actor* en su relación con el sistema.

Descripción de Requerimientos

Contenido del documento:

- Nombre del proyecto.
- Versión.
- Fecha.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- Objetivo.
- Alcance.
- Consideraciones.
- Situación actual.
- Requerimientos funcionales:

RF{id} – {Descripción *requerimiento funcional*}

Donde: {id} es un número correlativo único para cada *requerimiento funcional*.

- Requerimientos no funcionales:

RN{id} – {Descripción *requerimiento no funcional* }

Donde: {id} es un número correlativo único para cada *requerimiento no funcional*.

- Historial de modificaciones.

Descripción de Casos de Uso

Los requerimientos funcionales identificados se organizarán en *casos de uso* y, para cada *caso de uso*, se incluirá:

- Un nombre descriptivo con el siguiente formato:

UC{id} – {Nombre *caso de uso*}


Donde:

{id} es un número correlativo único para cada *caso de uso*.

{Nombre *caso de uso*} es un nombre descriptivo único para cada *caso de uso*.

Debe especificarse:

- Grupo.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


- Escenario principal.
- Descripción breve.
- Actores.
- Pre-condiciones.
- Post-condiciones (curso normal).
- Curso normal y alternativo: una descripción paso a paso de la funcionalidad requerida expresada en términos de las actividades que realizan los usuarios frente al sistema. Debe especificarse, para cada paso, el curso normal y si correspondiera un curso alternativo.
- Observaciones.
- Al menos un *diagrama de actividad* en donde se especifiquen todos los pasos que sigue un *actor* al ejecutar este *caso de uso* en el sistema. En este diagrama deben plasmarse tanto el flujo principal como los alternativos indicando con claridad bajo qué condiciones se ejecutan los alternativos.
- Opcionalmente, uno o más, *diagramas de estados* que ayuden a comprender los estados por los que pasa el sistema a lo largo de la ejecución del caso de uso.

Diagramas de casos de uso

Se crearán relaciones entre los *actores* y los *casos de uso* que indiquen la posibilidad de ejecución de cada *actor* del *caso de uso* con el que está relacionado.

Se crearán relaciones de *extensión* e *inclusión* entre los *casos de uso* tendientes a identificar funcionalidad repetida entre los *casos de uso*.

Estas relaciones se plasmarán en uno o más *diagramas de casos de uso*.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Para cada *diagrama de casos de uso* se definirá un nombre y se redactará una descripción.

El nombre se definirá utilizando el siguiente formato:

UC{id} – {Nombre *diagrama de casos de uso*}

Donde:

{id} es un número correlativo único para cada *diagrama de caso de uso*.

{Nombre *diagrama de casos de uso*} es un nombre descriptivo único para cada *diagrama de casos de uso*.

{Descripción} La descripción del *diagrama de casos de uso* deberá estar claramente asociada al nombre y expresar de forma concisa cuál es el contenido del diagrama.


Glosario

El glosario contendrá la definición de los términos utilizados en el contexto del sistema, ordenados alfabéticamente.

Se prestará especial atención a aquellos términos que posean más de una acepción, especificando cuál es la que se utiliza en el contexto del sistema y aquellos términos que son inherentes al sistema que se está especificando.

Prototipos de interfaces de usuario

Se presentarán prototipos de interfaces de usuario indicando para qué casos de uso se utiliza cada una y cuál es la secuencia de interfaces para ejecutar cada caso de uso, tanto en su flujo principal como en los alternativos.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


3.1.1.3 Análisis de riesgo de requerimientos

Objetivos

1. Identificar los requerimientos que, por diferentes factores, representan un mayor riesgo para el proyecto.
2. Identificar a los actores que pueden significar un riesgo para el proyecto.
3. Identificar otros factores que puedan poner en riesgo al proyecto.
4. Cuantificar el riesgo que corre el proyecto en diferentes escenarios.
5. Planificar acciones preventivas y correctivas para los factores de riesgo identificados.

Contenido

El contenido de este entregable se confeccionará a partir de lo descrito en los Anexos en la sección Análisis de riesgo y lo particularmente acordado para el proyecto.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción	
		13/07/07			
		Actualización			
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor	
1		1	13/07/07	R. Ortiz M. Ortiz	

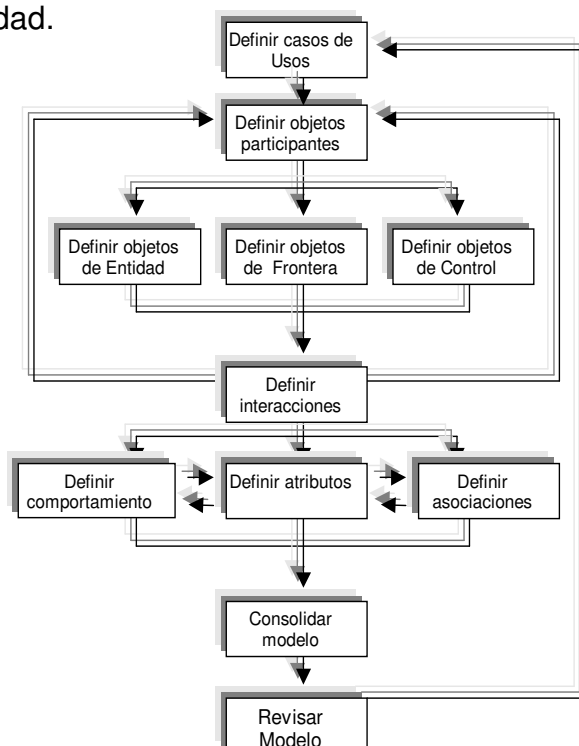
3.2 ANÁLISIS


Durante este flujo de trabajo se define y especifica la forma en que estará distribuida la funcionalidad del sistema independientemente de la plataforma de implementación. Para esto se utilizarán tres tipos de objetos: interfaz, entidad y control. A saber;

Interfaz: identifican puntos de comunicación del sistema con agentes externos, por ejemplo, otro sistema, un usuario o un periférico.

Entidad: representan conjuntos coherentes de datos de los que debe guardarse registro. Por ejemplo, un formulario o una factura.

Control: representan procesamiento de datos y control de flujo. Un uso típico es extraer los datos de los objetos de entidad, procesarlos y exponerlos a los objetos de interfaz, o, a la inversa, tomarlos de los objetos de interfaz, procesarlos y almacenarlos en los objetos de entidad.



	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.2.1 Elementos requeridos

Para iniciar este flujo de trabajo se requieren los siguientes elementos:

- Modelo de casos de uso estructurado

3.2.2 Entregables

3.2.2.1 Modelo de análisis

Objetivos

1. Organizar los requerimientos funcionales eliminando redundancias y contradicciones y pensando en la estructura que tendrá el sistema y cómo estará distribuida la funcionalidad.
2. Crear la primera aproximación a la arquitectura del sistema.
3. Comenzar a definir el sistema en términos de desarrollo de software y no en el lenguaje del usuario como sucede en el modelo de requerimientos.


Contenido

Paquetes de análisis

La funcionalidad especificada en cada caso de uso será asignada a algún *paquete de análisis*, pudiendo cada *paquete de análisis* contener la funcionalidad de más de un caso de uso. Se llamará a éstos *paquetes de análisis* de nivel 1.

Cada paquete de análisis contará con un nombre descriptivo y una descripción redactada en términos de responsabilidades asignadas al paquete.

Estas responsabilidades podrán ser distribuidas en *subpaquetes de análisis* recursivamente, no pudiendo superar más de dos niveles de desagregación. Es decir, se podrán crear *paquetes de análisis* hasta de nivel 3.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Objetos de análisis

Dentro de cada *paquete de análisis* del último nivel de desagregación disponible se definirán *objetos de análisis* de tipo interfaz, entidad y/o control entre los que se distribuirá las responsabilidades del *paquete de análisis* que los contiene.

Cada uno de estos objetos contará con un nombre descriptivo y una descripción en prosa de las responsabilidades que le fueron asignadas.

Diagramas de clases


Dentro de cada *paquete de análisis* del último nivel de desagregación disponible se creará al menos un diagrama de clases en el que se mostrarán las relaciones entre los *objetos de análisis* contenidos en el paquete.

Adicionalmente, podrá crearse un *diagrama de clases* general en el que se muestren las relaciones entre todos los *objetos de análisis* definidos.

Diagrama de Paquetes de Análisis

Cada diagrama de *paquetes de análisis* consiste de una agrupación de elementos que permiten subdividir el sistema en áreas manejables y que tengan competencias parecidas.

Este diagrama incluye en una descripción concisa del contenido del mismo más una representación gráfica de todos los paquetes de análisis identificados. Aquí se deberán indicar asociaciones de *dependencia* o de *generalización* entre estos.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Realización de casos de uso para análisis

Dentro de cada *paquete de análisis* de nivel 1 se creará un *caso de uso* con el estereotipo “use-case realization” (realización de caso de uso) por cada *caso de uso* del modelo de requerimientos cuya funcionalidad haya sido asignada al paquete.

Los *casos de uso* con el estereotipo “use-case realization” creados denominarán con el siguiente formato:

UCRA{id} – {Nombre caso de uso}


Donde:

{id} es el id del *caso de uso* del *modelo de requerimientos* que se está haciendo la realización.

{Nombre caso de uso} es el nombre del *caso de uso* del *modelo de requerimientos* del que se está haciendo la realización.

Para cada uno de estos *casos de uso* estereotipados “use-case realization” se creará un diagrama de secuencia o colaboración por cada flujo descrito en el *caso de uso* del *modelo de requerimientos* al que responde. En estos diagramas se mostrará cómo los *objetos de análisis* definidos dentro del paquete de análisis, o sus subpaquetes, se comunican entre sí para instanciar los flujos definidos en el *caso de uso* del *modelo de requerimientos*.

En la documentación de los *casos de uso* estereotipados “use-case realization” se incluirán requerimientos no funcionales relacionados con el *caso de uso* que se hayan detectado durante el análisis. Por ejemplo, “la ejecución del proceso no puede tardar más de 2 segundos”.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


Diagramas de trazabilidad

Como parte del *modelo de análisis* se creará uno o más *diagramas de clases* en donde se relacione cada *paquete de análisis* con el o los casos de uso cuya funcionalidad le ha sido asignada y cada *caso de uso* estereotipado “use-case realization” con el *caso de uso* del modelo de requerimientos que realiza.

En todos los casos, las relaciones creadas tendrán el estereotipo “trace” y estarán dirigidas con una flecha en el extremo correspondiente al *caso de uso* del *modelo de requerimientos*.

Cada uno de estos diagramas tendrá por nombre:

Trazabilidad análisis-requerimientos “n”, donde “n” es el número de diagrama de trazabilidad.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.3 DISEÑO

Durante este flujo de trabajo se define y especifica la plataforma que se utilizará para construir la solución y se distribuye la funcionalidad del sistema en subsistemas, respetando, en la medida de lo posible, la estructura definida durante la etapa de análisis y teniendo en cuenta las limitaciones de la plataforma seleccionada.


El objetivo central es generar especificación suficiente para comenzar la etapa de construcción

3.3.1 Elementos requeridos

Para iniciar este flujo de trabajo se requieren los siguientes elementos:

- Modelo de casos de uso estructurado
- Modelo de análisis

Dependiendo de la envergadura del proyecto, es posible que no se requiera la creación de un *modelo de análisis*. En ese caso, el *modelo de diseño* se desarrollará a partir del *modelo de caso de uso estructurado* utilizando las reglas para la creación de paquetes y delegación de responsabilidades desde el *modelo de casos de uso* definidas para el *modelo de análisis*.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.3.2 Entregables

3.3.2.1 Modelo de diseño

Objetivos

1. Definir la tecnología sobre la que se va a desarrollar el sistema de modo que permita cumplir con los requerimientos, tanto funcionales como no funcionales.
2. Teniendo en cuenta la estructura definida durante el análisis, estructurar el sistema en unidades funcionales que se ajusten a las características y limitaciones de la tecnología a utilizar.
3. Contener información suficiente para construir el sistema.


Contenido

Especificación estática

▪ *Arquitectura*

La arquitectura del sistema se especificará mediante uno o más *diagramas de clases* en donde cada subsistema de la aplicación se representará con un *paquete* que será nombrado *paquete de diseño*. Cada *paquete de diseño* tendrá un nombre descriptivo y una descripción en prosa en donde se especificarán las responsabilidades funcionales del subsistema y la tecnología con la que se lo implementará.

De ser necesario desagregar uno o más de los subsistemas para su especificación se podrá reproducir esta estructura dentro del subsistema a desagregar creando uno o más diagramas de clases y tantos *subpaquetes de diseño* como sea necesario.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Estos subpaquetes contarán también con un nombre descriptivo y una descripción en prosa en la que se especificará cuáles de las responsabilidades del paquete padre se delegan en el subpaquete, cuál es la tecnología con la que se lo implementará y por qué se la ha elegido.

El nombre tanto de los paquetes como de los subpaquetes deberá tener el siguiente formato:

PD{id} – {Nombre *paquete de diseño*}

Donde:

{id} es un número correlativo único para cada *paquete de diseño*.


{Nombre *paquete de diseño*} es el nombre descriptivo único para cada *paquete de diseño*.

Se trazarán en el diagrama las relaciones entre los paquetes y, siempre que sea posible, se definirán interfaces públicas para cada paquete a través de las cuales se accederá a la funcionalidad que éste implementa, indicando en la documentación de cada interfaz cuál es la funcionalidad a la que permite acceder.

Se incluirá para cada *paquete de diseño* un *diagrama de estados* que modele los estados por los que pasa el *paquete de diseño* como respuesta a los eventos que recibe a través de sus interfaces públicas.

- *Clases*

Basándose en los *objetos de análisis* de los *paquetes de análisis* cuya funcionalidad implementan, para cada *paquete de diseño* se definirán las clases que lo conforman. Documentando para cada una de las clases las responsabilidades que le son delegadas.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Para cada clase pública se incluirá la documentación de sus atributos y métodos públicos y protegidos, incluyendo la documentación de los parámetros y valores de retorno de los métodos.

Se creará al menos un diagrama de clases por paquete en donde se mostrarán las relaciones entre las clases que en él se encuentran definidas.

Se creará un diagrama de estados para cada una de las clases más importantes de cada paquete.

- *Base de datos*

Si se optara por la utilización de una base de datos relacional, se deberá confeccionar un diagrama entidad-relación, un diccionario de datos y un documento de configuración en donde se especifique la configuración de la base de datos y del motor.

- *Diagrama entidad-relación*

El diagrama entidad relación contendrá todas las tablas a ser utilizadas por el sistema, todos los campos de cada tabla (indicando cuáles son claves primarias), el tipo de dato de cada campo y las relaciones entre las tablas indicando la cardinalidad y los nombres de las claves foráneas.


De no haber requerimientos de performance que fueren lo contrario, el diagrama estará normalizado según la quinta forma normal.

- *Diccionario de datos*

En el diccionario de datos se documentará en prosa el contenido de cada tabla y cada atributo, haciendo énfasis en su valor semántico en el contexto del sistema.

Así mismo se especificará:

- Las restricciones entre los campos de una tabla.
- Las restricciones de los valores de un campo.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- La posibilidad de asignarle valores nulos a cada campo.

- Documento de configuración*

En este documentó se especificarán los índices a crear y el valor que debe tomar cualquier otro parámetro de configuración del motor de base de datos que tenga un impacto en el funcionamiento del sistema. Se hará énfasis en la seguridad, performance y el esquema de conexión entre la base de datos y los componentes del sistema que acceden a ella.

Especificación dinámica y trazabilidad

Realización de casos de uso para diseño

Por cada *caso de uso* del modelo de requerimientos se creará un *caso de uso* con el esterotipo “use-case realization” (realización de caso de uso) cuyo nombre responderá al siguiente formato:


UCRD{id} – {Nombre caso de uso}

Donde:

{id} es el id del *caso de uso* del *modelo de requerimientos* del que se está haciendo la realización.

{Nombre caso de uso} es el nombre del *caso de uso* del *modelo de requerimientos* del que se está haciendo la realización.

Para cada uno de estos casos de uso estereotipados “use-case realization” se creará un diagrama de secuencia o colaboración por cada flujo descrito en el caso de uso del modelo de requerimientos que realiza. En estos diagramas se mostrará cómo las

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

interfaces y clases públicas de los *paquetes de diseño* se comunican entre sí para instanciar los flujos definidos en el *caso de uso* del *modelo de requerimientos*.

- *Diagramas de trazabilidad*

Como parte del modelo de diseño se creará uno o más diagramas de clases en donde se relacione cada *paquete de diseño* con el o los *paquetes de análisis* cuya funcionalidad le ha sido asignada, cada clase con el o los objetos de análisis que implementa y cada caso de uso estereotipado “use-case realization” con el *caso de uso* del modelo de requerimientos que realiza.

En todos los casos, las relaciones creadas tendrán el estereotipo “trace” y estarán dirigidas con una flecha en el extremo correspondiente al *caso de uso* del modelo de requerimientos u objeto o paquete de análisis.


Cada uno de estos diagramas tendrá por nombre:

Trazabilidad diseño-análisis-requerimientos “n”, donde “n” es el número de diagrama de trazabilidad.

3.3.2.2 Modelo de implementación

Objetivos

1. Planificar los componentes a desarrollar.
2. Planificar la estructura de hardware.
3. Planificar la asignación de componentes a cada pieza de hardware, previniendo sobrecargas del hardware y sus consecuentes caídas de performance.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Contenido

Subsistemas de implementación

Por cada subsistema definido, especificado en un *paquete de diseño*, se creará un subsistema de implementación representado por un *paquete de implementación*.


Dentro de cada *paquete de implementación* se definirá un componente por cada archivo instalable que se genere a partir de un conjunto de clases, y se le asignará al componente el estereotipo que corresponda para identificar el tipo de archivo al que responde. Por ejemplo, en una implementación en Java se creará un componente por cada clase con el mismo nombre de la clase y el estereotipo <<class>>. En cambio, en una implementación en .NET, si lo que se está construyendo es una biblioteca de clases, se creará un solo componente para todas las clases del *subsistema de diseño* con el estereotipo <<dll>>.

Si uno o más *paquetes de implementación* se agruparan en un archivo para su implantación, se creará un componente por cada uno de estos archivos con el estereotipo que corresponda. Siguiendo con el ejemplo de una implementación en Java, si uno o más subsistemas se agruparan en un archivo .jar, por cada archivo .jar se creará un componente con el estereotipo <<jar>>.

Una vez creados los componentes se los relacionará con las clases a las que implementan de la manera en que lo posibilite la herramienta CASE que se esté utilizando.

Diagrama de implantación

Por cada servidor y cada tipo de computadora cliente que se utilice en el sistema se creará un nodo.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

A cada nodo se le asignará un nombre descriptivo y una descripción en prosa en la que se indicará el rol que cumple el nodo en el sistema y las características técnicas de la máquina.

Los nodos se comunicarán con links. Para cada link se especificará el tipo.

Para cada tipo de link, se indicará el ancho de banda y otras características relevantes.

Utilizando el mecanismo que brinde la herramienta CASE que se utilice, se asignarán los componentes definidos a los nodos en los que serán implantados.

3.3.2.3 Planeamiento de testing


Objetivos

1. Definir los objetivos de calidad del sistema.
2. Determinar los medios necesarios para verificar que se alcanzaron los objetivos de calidad.
3. Generar información suficiente para poder controlar la calidad del sistema una vez construido.

Contenido

Dentro del conjunto de documentos de testing acordados al comienzo del proyecto (ver en los Anexos, Documentos de Testing), se crearán los siguientes documentos:

- Test Plan
- Test Design Specification
- Test Case Specification

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- Test Procedure Specification

3.3.2.4 Análisis de riesgo de técnico

Objetivos

1. Identificar los factores técnicos que representan un mayor riesgo para el proyecto.
2. Cuantificar el riesgo que corre el proyecto en diferentes escenarios.
3. Planificar acciones preventivas y correctivas para los factores de riesgo identificados.

Contenido

El contenido de este entregable se confeccionará a partir de lo descrito en los Anexos en la sección Análisis de riesgo y lo particularmente acordado para el proyecto.

3.4 CONSTRUCCIÓN


Los objetivos centrales de este flujo de trabajo son codificar en GENEXUS el sistema según las especificaciones generadas en el flujo de trabajo de diseño y crear pruebas de unidad automáticas.

Adicionalmente, se realiza el planeamiento de la implementación de los componentes.

3.4.1 Elementos requeridos

Para iniciar este flujo de trabajo se requieren los siguientes elementos:

- Modelo de diseño

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- Modelo de implementación

3.4.2 Entregables

3.4.2.1 Componentes en funcionamiento

Binarios

Se considera un entregable a los componentes binarios que surjan de la compilación del código fuente para cada versión del sistema.

Código fuente

Se considera un entregable a todo el código fuente necesario para compilar cada versión del sistema.


El código fuente debe estar estructurado y documentado según las convenciones de codificación acordados al comenzar el proyecto.

Se incluyen en los Anexos unas Convenciones de codificación sugeridas.

3.4.2.2 Pruebas de unidad automáticas

En el caso en que se utilice un lenguaje de programación orientado a objetos, se considera la unidad de construcción del sistema a la clase. Si el lenguaje de programación no fuera orientado a objetos se definirá lo que se considera unidad antes de comenzar la codificación, posiblemente, al comienzo del proyecto.

Para cada *paquete de diseño* se creará un *paquete de testing de unidad* cuyo nombre se definirá con el siguiente formato:

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Test-PD{id del *paquete de diseño* al que prueba}-{nombre del *paquete de diseño* al que prueba}

Dentro de cada paquete de testing se creará para cada clase del paquete de diseño una prueba de unidad automática cuyo nombre tendrá el siguiente formato:

Test{nombre de la clase a la que prueba}

Se recomienda para la codificación de las pruebas de unidad la utilización de la implementación del framework XUnit indicada para el lenguaje que se esté utilizando.

De las pruebas automáticas se entregará el código fuente junto con el código fuente del resto del sistema y los binarios que resulten de la compilación de las pruebas automáticas se entregarán junto con los binarios del sistema.

3.4.2.3 Documentación del usuario

Objetivo


1. Brindar una referencia al usuario para despejar dudas durante la operación del sistema.
2. Tener una descripción formal y completa de las posibilidades que brinda el sistema orientada a usuarios no técnicos.

Contenido

El manual del usuario contendrá una introducción en la que se incluirá una descripción general del sistema, su propósito y la principal funcionalidad que brinda.

Si se tratara de una actualización, se incluirá en la introducción una descripción detallada de todo lo que ha cambiado en el sistema de la versión anterior a la versión que descripta en la documentación.


Se incluirá también una captura de cada pantalla del sistema junto a la que se documentará:

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- el propósito de la pantalla,
- la información que debe ser ingresada en cada campo si es que se trata de un formulario,
- las acciones del usuario que disparan procesos, por ejemplo, la presión de un botón,
- la descripción de cada uno de estos procesos disparados por el usuario
- la descripción de los posibles mensajes que puede recibir el usuario.

Si la pantalla estuviera relacionada con otras por ser parte de un proceso que requiere más de una pantalla, se agruparán las pantallas según van apareciendo en el sistema y se documentará su función en el contexto del proceso.

La documentación del usuario se entregará en una versión imprimible, con un layout orientado a papel, y una versión para consulta en línea orientada a ser leída y navegada en una pantalla. En esta segunda versión, se hará énfasis en la relación de la información mediante hipervínculos y en la facilidad de consulta, debiendo tener contemplado algún método de búsqueda de texto libre.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.5 TESTING

Durante este flujo de trabajo se ejecutan las pruebas planificadas en el flujo de trabajo de diseño. Su objetivo central es encontrar fallas en el sistema.

3.5.1 Elementos requeridos

Para iniciar este flujo de trabajo se requieren los siguientes elementos:

- Planeamiento de testing
- Componentes en funcionamiento

3.5.2 Entregables


3.5.2.1 Ejecución de testing

Este entregable incluye toda la documentación generada durante la ejecución de las pruebas incluyendo los siguientes documentos:


- Test Log
- Test Incident Report
- Test Item Transmittal Report

Si se hubiera automatizado algún procedimiento de prueba se entregará el código fuente del caso de prueba que alimenta al robot que realiza la prueba. Así mismo, se entregará documentación indicando las características técnicas del robot.

3.5.2.2 Evaluación de testing

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Este entregable consta de un solo documento (Test Summary Report) en el que se resume el resultado de la ejecución de las pruebas en cuanto a las fallas detectadas en el sistema probado y los inconvenientes encontrados en las pruebas en sí.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

3.6 IMPLEMENTACIÓN

Durante este flujo de trabajo se implanta el sistema desarrollado y se asiste a los usuarios en la incorporación del mismo a sus tareas cotidianas.

3.6.1 Entregables

3.6.1.1 Plan de implementación

Objetivos:

1. Construir un marco de referencia para acordar los pasos de la implementación con los clientes internos.
2. Prever los recursos necesarios para realizar la implementación.
3. Planificar las tareas necesarias para realizar la implementación y su asignación a los recursos.


Contenido

El plan de implementación contendrá la lista completa de recursos necesarios para completar la implementación, indicando, en el caso de ser estos recursos humanos, la capacitación y perfil requeridos.

Contendrá, además, la lista completa de tareas a realizar para completar la implementación detallando la dependencia entre tareas, el tiempo estimado para cada una y el recurso al que se la asigna.


Algunas de las tareas que deben ser contempladas como parte un plan de implementación son:

- Capacitación a usuarios

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

- Instalación del software
- Migración de datos
- Soporte a usuarios
- Captura de pedidos de mejora

La lista total de tareas se acordará para cada proyecto en particular.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

4 ANTEPROYECTO


El anteproyecto es el entregable en donde se describen las características particulares del proyecto.

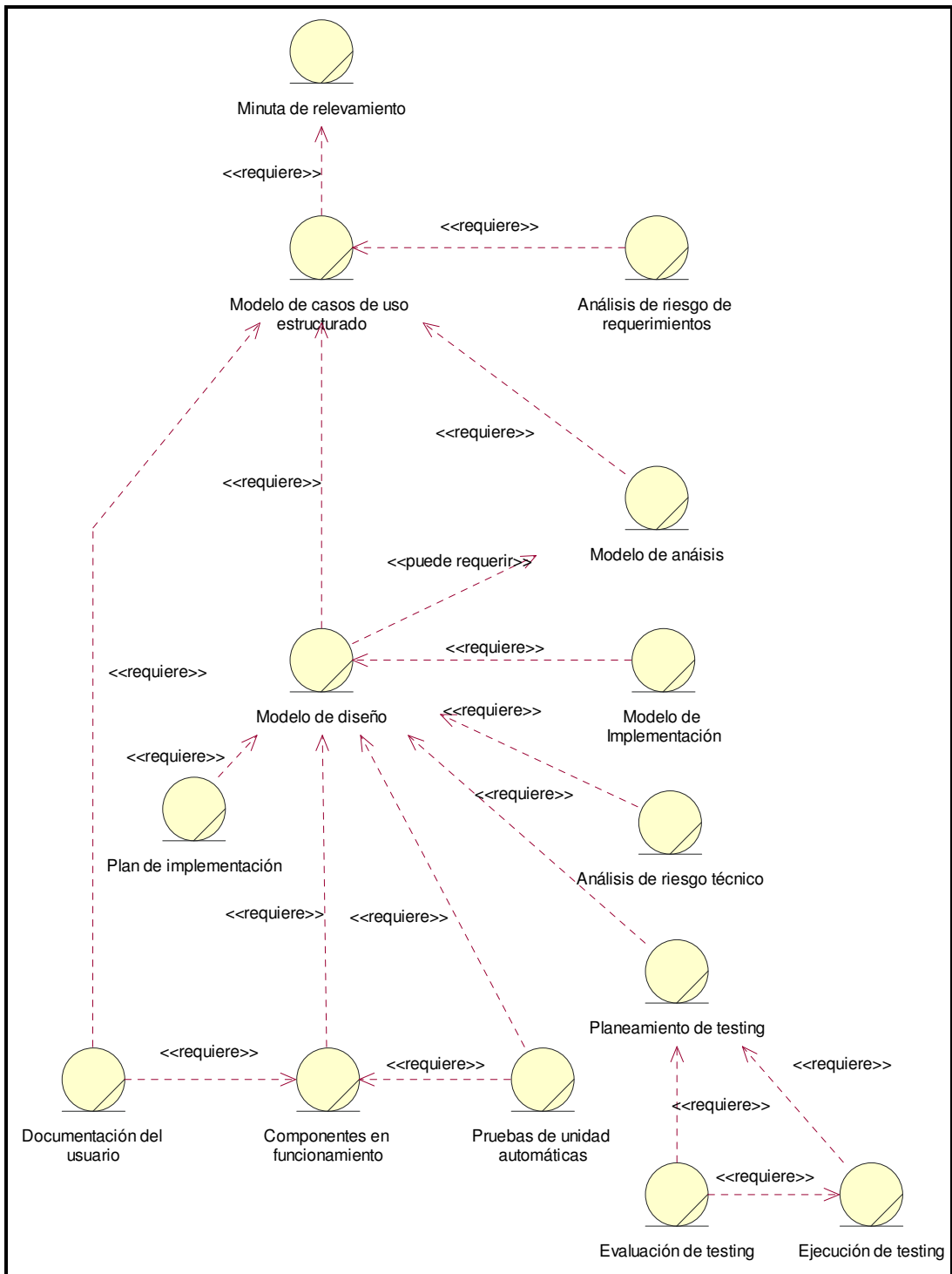
En particular, se describirán en este documento:


- El método de estimación de esfuerzo a utilizar para la Adquisición y especificación de requerimientos y la Implementación.
- El método de estimación de esfuerzo a utilizar para el Análisis, Diseño, Construcción y Testing.

Planificación inicial.

Los entregables que se crearán durante el proyecto y el nivel de detalle que alcanzará cada uno al finalizar cada fase del ciclo de desarrollo. Los entregables que se seleccionen deben respetar el siguiente diagrama de dependencias

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción	
		13/07/07			
		Actualización			
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor	
1		1	13/07/07	R. Ortiz M. Ortiz	



	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

5 MÉTRICAS


5.1 *Estimación de costos para adquisición y especificación de requerimientos e implementación*

Para realizar la estimación de esfuerzo y sus costos asociados para la Adquisición y especificación de requerimientos e Implementación se acordará un método al comenzar el proyecto.

5.2 *Estimación de costos para análisis, diseño, construcción y testing*

La estimación de costos para las etapas de análisis, diseño, construcción y testing se realizará estimando primero el tamaño del sistema a desarrollar, contando Puntos de función sobre los requerimientos relevados en la etapa de Adquisición y especificación de requerimientos.

Luego se acordará un método para extrapolar la estimación de tamaño a esfuerzo.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


6 ANEXOS

6.1 Documentos de testing

Para la documentación de la etapa de testing se utilizará el estándar IEEE-829 siguiendo los siguientes lineamientos:

- Cada caso de prueba de sistema será diseñado para probar la funcionalidad de un, y sólo un, caso de uso.
- Cada procedimiento de prueba será diseñado para ejecutar el o los casos de prueba asociados a través de un, y sólo un, paquete de diseño de nivel uno.
- Los componentes que automatizan uno o más procedimientos de prueba incluirán en su documentación el listado detallado de los procedimientos que automatizan.
- Los documentos del estándar a utilizar se acordarán al comienzo de cada proyecto y se definirán en el Test Plan.
- Los tipos y niveles de pruebas a realizar y la cobertura de cada una se acordarán al comienzo del proyecto.
- Para darle nombre a los documentos se utilizará la siguiente convención:

Documento	Nombre
Test Plan	tp-{nombre proyecto}
Test Design Specification	tds-{nombre proyecto}-n
Test Case Specification	Tcs-{nombre proyecto}-{código caso de uso}-n
Test Procedure Specification	tps-{nombre proyecto}-{código paquete de diseño}-n
Test Log	tl-{nombre proyecto}-{tester}-{fecha}

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


Test Incident Report	Tir-{nombre proyecto}-{código caso de pueba}-{fecha}
Test Summary Report	Tsr-{nombre proyecto}-n
Test Item Transmittal Report	titr-{nombre proyecto}-{código item}-{fecha}-n

Donde:

- {nombre proyecto} es el nombre asignado al proyecto
- {código caso de uso} es el código asignado al caso de uso cuyas pruebas especifica, ya sea este estereotipado “use-case realization” o no
- {código paquete de diseño} es el código del paquete de diseño de nivel 1 al que se aplica el procedimiento
- {tester} es el nombre de la persona que realizó las pruebas
- {fecha} es la fecha en la que se confeccionó el documento, especificada con el formato aaaa-mm-dd.
- {código item} es el código del item que se transmite mediante el documento
- “n” es un número entero correlativo que se utiliza para los casos en que la combinación de los parámetros anteriores del documento genere dos documentos de igual nombre. Para cada nuevo documento de igual nombre se incrementará el número en 1.

6.2 Análisis de riesgo

En el contexto de este documento, se entiende por Análisis de Riesgo a la tarea de recopilar y listar los sucesos o eventos que amenazan o eliminan la posibilidad de éxito de un proyecto; analizarlos y planificar acciones tendientes a evitarlos.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Los resultados de esta tarea se volcarán en un documento que contendrá, al menos, la siguiente información:

- Identificación de riesgos


Fuentes de riesgo

Las *fuentes de riesgo* son categorías de posibles eventos que amenazan o eliminan la posibilidad de éxito de un proyecto. Por ejemplo, acciones de los actores involucrados en el proyecto, estimaciones, problemas en el equipo de trabajo, errores de diseño, etc.

Por cada *fuentes de riesgo* identificada se identificarán y listarán los eventos que a ella pertenezcan detallando para cada uno:

- Probabilidad de ocurrencia: un número del 0 al 100 que indica qué tan probable es que el evento ocurra. 100 indica que el evento ocurrirá indefectiblemente y 0 que no ocurrirá de ninguna manera.
- Resultados posibles dado que ocurra
- Descripción en prosa de los efectos posibles en el proyecto si el evento ocurre.
- Etapa del proyecto en la que está previsto
- Etapa o rango de etapas del proyecto en las que está previsto que el evento ocurra.
- Síntomas

Descripción de características verificables que pueden indicar que el evento ha ocurrido y/o, más importante, que está próximo a ocurrir. Por ejemplo,

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

baja productividad en la programación puede indicar errores en el diseño y/o en la capacitación del personal.

- Frecuencia de ocurrencia esperada

Cantidad de ocurrencias de este evento que son esperables a durante el ciclo de vida del proyecto.

Respuestas posibles: posibles estrategias para evitar, mitigar o absorber la ocurrencia del evento.

Las respuestas se clasificarán en respuestas que:


“**Evitan**” la ocurrencia del evento, es decir, reducen su probabilidad de ocurrencia a 0.

”**Mitigan**” la ocurrencia del evento, es decir, disminuyen su probabilidad de ocurrencia.

”**Aceptan**” la ocurrencia del evento desarrollando un plan alternativo y/o procurando recursos para permitir que el proyecto siga adelante.

Cuantificación: analizando las fuentes de riesgo identificadas se describen los posibles resultados del proyecto en términos de su probabilidad de ocurrencia.

Evaluación: se realizará un resumen en donde el analista volcará su visión del proyecto en cuanto a los riesgos que se están afrontando.

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

6.3 Convenciones de codificación

6.3.1 Recomendaciones generales

Tener siempre en mente al generar el código que lo va a tener que entender otra persona.

El código debe ser claro y espaciado.

Se debe intentar dividir el código en bloques y cada bloque debe tener un comentario explicando su funcionalidad.

Los scripts que generen HTML deberán generarlo de forma que sea legible.

Al asignar nombres a las clases, funciones o variables se debe buscar nombres descriptivos y no cortar las palabras para hacer el nombre más corto. Por ejemplo, no usar bscDatosC() sino buscarDatosCliente().


Se intentará, en lo posible, no exceder los 80 caracteres por línea (contados estos con el caracter <TAB> igual a 4 espacios en blanco) de forma que la visualización del código no se vuelva confusa en monitores pequeños y/o interfaces en modo texto.

Las variables y constantes deben estar agrupadas por función y no aisladas.

6.3.2 Declaraciones (statements)

Independientemente del lenguaje que se esté utilizando, se organizarán las declaraciones de a uno por línea.

6.3.3 Indentado

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Independientemente del lenguaje que se esté utilizando, se mostrará la jerarquía de cada bloque de código con un caracter <TAB> al comienzo de cada línea del bloque, mostrando así la dependencia de este de la clase, función o estructura de control inmediata anterior.

6.3.4 Clases

1.Los nombres de las clases comenzarán siempre con mayúscula y serán redactados en castellano. Podrán utilizarse palabras en inglés en el caso en que sean de uso común en la jerga informática y su traducción complique su comprensión. Por ejemplo, palabras como *rollback*.


2.Si el nombre constara de dos o más palabras se las dividirá con mayúsculas. Es decir: PalabraPalabraPalabra.

3.Después de la declaración de la clase se incluirá una breve descripción de su utilidad y funcionamiento.

4.Se intentará agrupar los atributos y métodos de cada clase de la siguiente forma con el objetivo de que aquellos que componen la interfaz de la clase queden juntos:

- Atributos privados.
- Atributos protegidos.
- Atributos públicos.
- Métodos públicos.
- Métodos protegidos.
- Métodos privados.

6.3.5 Funciones y Métodos:

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz


1.Los nombres de las funciones y métodos comenzarán siempre con mayúscula y serán redactados en castellano. Podrán utilizarse palabras en inglés en el caso en que sean de uso común en la jerga informática y su traducción complique su comprensión. Por ejemplo, palabras como *rollback*.

2.Si el nombre constara de dos o más palabras se las dividirá con mayúsculas. Es decir: PalabraPalabraPalabra.

3.Antes de la declaración de la función o método se incluirá una breve descripción con la siguiente estructura:

Sección	Descripción de la sección
Propósito	Lo que hace el método o función (no cómo).
Presunciones	Lista de cada variable externa o global, archivo abierto, o cualquier otro elemento que no es obvio.
Efectos	Lista de cada variable externa o global, archivo abierto, o cualquier otro elemento que es modificado por este método o función.
Parámetros	Cada parámetro cuyo contenido no sea obvio, en particular si el método espera valores en un rango específico.
Retorno	Explicación de los valores de retorno.


(Ejemplo en la siguiente página)

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción	
		13/07/07			
		Actualización			
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor	
1		1	13/07/07	R. Ortiz M. Ortiz	

```

*****
*****
' Proposito:          Valida que los campos que se encuentran en
Fields, cuyo tipo
'                    de dato esta definido en colCampos, tengan un
formato valido.
' Presunciones:      Asume que existe un collection llamada Fields
que contiene
'                    los valores ingresados por el usuario con el
nombre del campo
'                    como clave y el valor ingresado por el usuario
como valor.
' Parametros:
'   colCampos:      Collection que asocia los nombre de los campos
del formulario
'                    con la tabla en la base de datos. Tiene el
siguiente formato:
'                    ..(0,0) = Cantidad de campos de datos
'                    ..(n,0) = Nombre del campo en la tabla
'                    ..(n,1) = Descripcion del campo en pantalla
'                    ..(n,2) = Tipo de Dato
'                    T :Texto
'                    F :Fecha
'                    C :CUIT o CUIL
'                    N          numerico real
'                    NE         numerico entero
'                    NEP        numerico          entero
positivo
'                    NEP0 numerico entero positivo sin
cero
'                    NRP          numerico real positivo
'                    NRP0 numerico real positivo sin
cero
'                    nnG          Se asigna al grabar
'                    ..(n,3) = Obligatorio "R" - No obligatorio
""
'
' Retorno:           Collection con el nombre del campo como clave y
el mensaje de
'                    error como valor.
*****
Function Common_validarCampos(colCampos)
End Function

```

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

6.3.6 Variables

1.Los nombres de las variables comenzarán siempre indicando el *scope* y el tipo de dato que la variable contiene y se redactarán en castellano. Podrán utilizarse palabras en inglés en el caso en que sean de uso común en la jerga informática y su traducción complique su comprensión. Por ejemplo, palabras como *rollback*.


2.El *scope* de una variable se indicará incluyendo al comienzo del nombre alguno de los siguientes prefijos:

Scope	Prefijo	Ejemplo
Global	G	gstrNombreUsuario
Módulo	M	mbInCalcInProgress
Local del método	Ninguno	dblVelocity

3.Si el nombre constara de dos o más palabras se las dividirá con mayúsculas. Es decir: [scope]tipoDeDatoPalabraPalabraPalabra.

Tipos de datos y prefijos:

Tipo de dato	Prefijo	Ejemplo
Boolean	BIn	blnFound
Byte	Byt	bytRasterData
Collection object	col	colWidgets
Currency	cur	curRevenue
Date (Time)	dtm	dtmStart

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Double	dbl	dblTolerance
Error	Err	errOrderNum
Integer	Int	intQuantity
Long	Lng	lngDistance
Object	obj	objCurrent
Single	sng	sngAverage
String	Str	strFName
User-defined type	udt	udtEmployee
Variant	Vnt	vntChecksum

6.3.6.1 Casos particulares

Índices de números enteros

Al utilizar índices en ciclos de la forma:

For indice = 1 To 10

Se utilizarán como nombres de los índices las variables:

i, j, k, y, z

en ese orden para cada nivel de anidamiento.

Por ejemplo:

For i = 1 To 10


For j = 1 To 10

For k = 1 To 10

Next 'k

Next 'j

Next 'i

	MANUAL DE PROCEDIMIENTOS DE ON S.R.L.	Fecha Emisión		Área de Producción
		13/07/07		
		Actualización		
Código de Documento	Desarrollo de Software	Revisión N°	Fecha Revisión	Autor
1		1	13/07/07	R. Ortiz M. Ortiz

Constantes

Los nombre de las constantes cumplirán las mismas reglas que los de las variables agregando la letra “c” al comienzo. Es decir:

[c][scope]tipoDeDatoPalabraPalabraPalabra

Por ejemplo, cgstrDbNombre.

6.3.7 Glosario

6.3.7.1 Scope

Porción de un programa en la cual una variable está definida.

6.4 Herramientas de desarrollo

En este apartado se incluye una lista de algunas de las herramientas utilizadas para el desarrollo de sistemas en la Dirección de Sistemas Informáticos de la empresa.

Genexus 6.1

Genexus 7.0

Genexus 7.5

Genexus 8.0

Genexus 9.0

Visual Studio 6.0

SQL Server 2000

SQL Server 7