

2 Recogida y documento de los requisitos

Al iniciar un proyecto software y al decidir que se emplea el Proceso Unificado, la primera iteración corresponde con la fase de Inicio. En ella se pretende *vislumbrar* el alcance del producto. El principal problema es si el personal involucrado está de acuerdo en la visión del proyecto, tanto desarrolladores como los posibles usuarios del producto final.

En esta fase se analiza la viabilidad del proyecto, aunque también podría incluir los primeros talleres de requisitos y la planificación de la primera iteración. La tabla adjunta muestra los artefactos comunes de la fase de inicio.

Tabla 2-1 Ejemplos de artefactos de la etapa de inicio

Artefactos	Comentario
Visión y análisis del negocio	Describe los objetivos y las restricciones de alto nivel, el análisis del negocio y proporciona un informe para la toma de decisiones.
Modelo de Casos de Uso	Cuenta los requisitos funcionales y en menor medida los no funcionales.
Especificación Complementaria	Documenta los requisitos relacionados con la calidad del SW.
Glosario	Terminología clave del dominio
Lista de Registros & Plan de Gestión del Riesgo darles respuestas.	Detalla los riesgos del negocio, técnicos, recursos, planificación y las ideas para mitigarlos.
Prototipos y pruebas de conceptos	Para clarificar la visión y validar las ideas técnicas.
Plan de Iteración	Describe qué hacer en la primera iteración de la elaboración.
Fase Plan de & Plan de Desarrollo de Software	Estimación de poca precisión de la duración y esfuerzo de la fase de elaboración. Herramientas, personas, formación y otros recursos.
Marco de Desarrollo	Una descripción de los pasos del UP y los artefactos adaptados para este proyecto. El UP siempre se debe adaptar al proyecto.

Estos artefactos se completan sólo parcialmente en esta fase. Se refinarán de manera iterativa en las siguientes iteraciones. El nombre en mayúsculas indica que es un artefacto UP con ese nombre oficial.

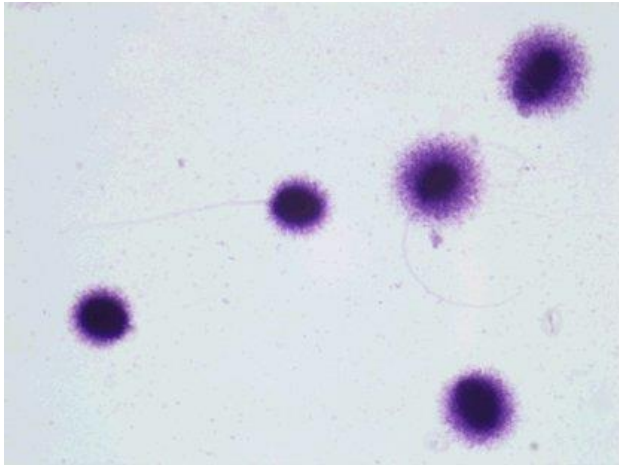
En esta etapa se puede dar la mayoría de los casos de uso, aunque sólo un 10% de ellos con detalles, sobre todo aquellos que son críticos. El resto sólo se enumeran.

También se puede lanzar algunas tareas de programación con la misión de crear prototipos de “pruebas de conceptos”. Éstos tratan de clarificar unos pocos requisitos mediante prototipos, aclarando la programación para cuestiones técnicas de carácter crítico.

La documentación se almacena digitalmente y en línea. Con el objeto de hacerla accesible a todo el personal involucrado, se puede emplear un servidor bien en una intranet o usar páginas privadas en un servidor *web*.

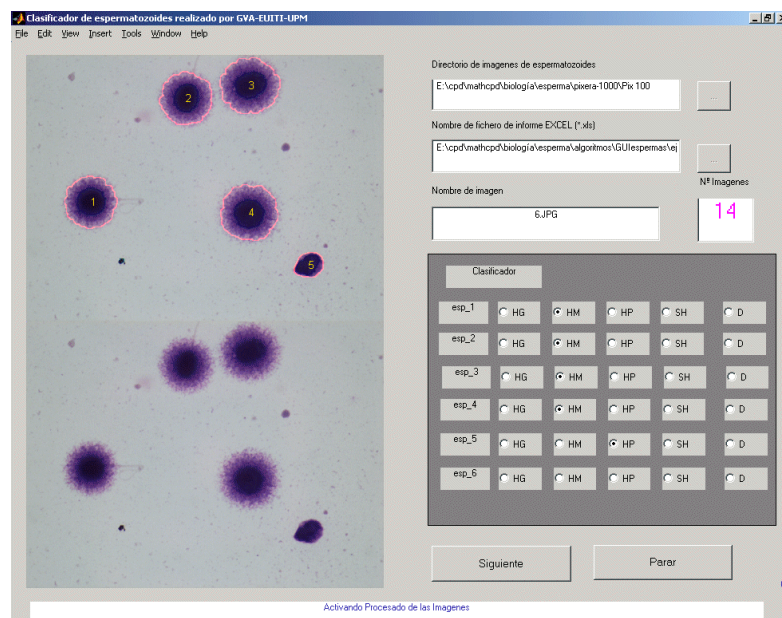
Ejemplo 2.1

Se pretende realizar un proyecto de clasificación de espermatozoides humanos mediante técnicas de análisis de imágenes digitales. Las imágenes son capturadas a través de un microscopio y el objetivo es detectar los espermatozoides y clasificarlos según su morfología.



La primera tarea fue capturar los requisitos de la aplicación. Se emplearon varias reuniones entre los biomédicos y los ingenieros para aclarar realmente qué es lo que se quería. De estos encuentros se rellenaron varios documentos (Visión y Alcance, Glosario, algunos casos de uso y Especificaciones Complementarias).

Para realimentar el conocimiento de los biomédicos en la aplicación, se desarrolló un prototipo empleando *Matlab*. Esta aplicación, exige poco tiempo y esfuerzo, sirve para reforzar algunas ideas que quedaron en el aire y para descartar otras. Muchas veces, la mentalidad de los desarrolladores va en una dirección, mientras los usuarios piensan en otros tipos de servicios.

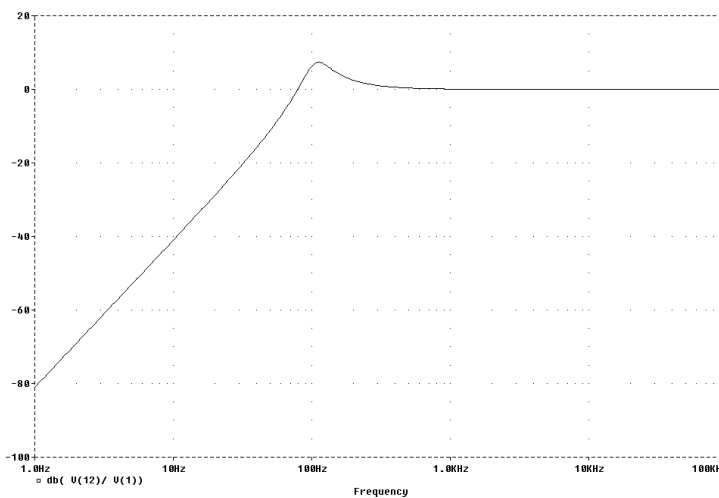
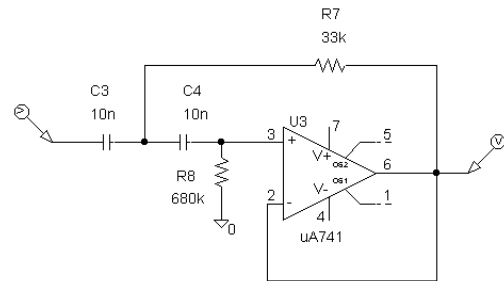


La presentación de un prototipo favorece la realimentación y acaba con actividades que no son necesarias. Además de acotar las singularidades del proyecto.

Ejemplo 2.2

Diseñar una aplicación que entregue la respuesta en frecuencia de los filtros analógicos lineales. Una vez capturado el circuito eléctrico, se pasará a determinar cual es la función de transferencia en el dominio de la frecuencia y se presentará la respuesta en frecuencia en un diagrama de Bode.

Este ejemplo sólo se empleará como hilo conductor de la materia de la asignatura, analizando las diferentes etapas que hay que cumplir, según el Proceso Unificado.



No obstante, el sistema pretende capturar un circuito analógico y mostrar su respuesta en frecuencias. En la actualidad hay aplicaciones que realizan este cometido, como por ejemplo PSPICE.

2.1 Comprensión de los requisitos

Los requisitos son capacidades y condiciones con las cuales debe estar conforme el sistema. El primer reto del trabajo de los requisitos es encontrar, comunicar y recordar lo que realmente se necesita, de manera que tenga un significado claro para el cliente y para los desarrolladores.

En un estudio sobre los costes reales de los proyectos SW, el 37% estaban relacionados con los requisitos.

Coste de los proyectos	%
Entradas de usuarios erróneas	13
Requisitos incompletos	12
Cambios en los requisitos	12
Habilidades técnicas pobres	7
Mala dirección	6
Otros	50

Los requisitos se clasifican de acuerdo con el modelo FURPS+ (*Functional, Usability, Reliability, Performance, Supportability*):

- Funcional (*Functional*): características funcionales de la aplicación, esto es, las utilidades que le dan al usuario.
- Facilidad de uso (*Usability*): factores humanos relacionados con su manejo, ayuda y documentación.
- Fiabilidad (*Reliability*): Capacidad de recuperación ante los fallos.
- Rendimiento (*Perfomance*): Tiempos de respuesta, precisión, eficacia.
- Soporte (*Supportability*): Adaptabilidad, facilidad de mantenimiento, internacionalización, configurabilidad.

El + indica requisitos adicionales tales como:

- Implementación: lenguajes y herramientas, soporte físico.
- Interfaz: interacción con otros sistemas externos.
- Operaciones: gestiones del sistema en su puesta en marcha.
- Empaquetamiento: instaladores, distribución.
- Legales: licencias, claves de instalación,...

Lo normal es dividir los requisitos en funcionales y no funcionales. Los requisitos funcionales se estudian y se recogen en los documentos de Casos de Uso. Los no funcionales aparecen en los informes de Especificaciones Complementarias.

Evidentemente, los requisitos influyen en la arquitectura del sistema. Se entiende como arquitectura los diferentes paquetes que constituyen el proyecto. Es el nivel más elevado de abstracción del proyecto y define los componentes que van a participar en la elaboración del sistema.

2.2 La secuencia de captura de los requisitos

Una secuencia recomendada en la captura de los requisitos puede ser:

1. Escribir un primer borrador breve de la visión del proyecto.
2. Identificar los objetivos de usuario y los casos de uso de apoyo.
3. Escribir algunos casos de uso y comenzar el documento de especificaciones complementarias.

4. Refinar la visión del proyecto, resumiendo la información a partir de de los documentos redactados.

2.3 Visión y Alcance del proyecto

El documento de Visión y Alcance del proyecto resume la visión del proyecto. Sirve para comunicar de manera concisa las grandes ideas de por qué se pospuso el proyecto, cuáles son los problemas a resolver, quienes son las personas involucradas, qué se necesita y cuál podría ser la apariencia de la solución propuesta. Puede servir como documento en la contratación entre la empresa receptora del servicio y los desarrolladores.

En este informe la granularidad que se refleja es alta, i.e. narra los aspectos más relevantes del proyecto, sin entrar en los detalles. El documento de Visión y Alcance resulta complementario con los documentos de Casos de Uso y Especificaciones Complementarias, los cuales se verán más adelante.

En este documento debe de aparecer cuáles son los objetivos esenciales y los problemas que hay que resolver. Pero ¿qué técnicas se pueden emplear para localizarlos?

Habrá que realizar trabajo de grupo. Existen distintas técnicas para el descubrimiento de los problemas. Con el objeto de descubrir los objetivos esenciales y para ayudar a la generación de ideas y su priorización se pueden utilizar: mapas mentales, diagramas causa-efecto, tormentas de ideas, votaciones,... Ante todo, son importantes las descripciones breves de alto nivel.

Una técnica para identificar los aspectos esenciales del proyecto es utilizar un esquema a dos niveles. Sucintamente se citará las características esenciales en una jerarquía de objetivos principales y objetivos secundarios.

Una guía para detectar las características del sistema es el siguiente test lingüístico:

El sistema debe de hacer la <característica X>

Si se supera la prueba, la característica se introducirá en el esquema. Con esta herramienta y con otras más, se confeccionará el esquema a dos niveles. Una vez realizado el trabajo de detectar las características del sistema, éstas deberán de reflejarse en el documento de Visión y Alcance.

Ejemplo 2.3

Determinar las características para la aplicación de clasificación de los espermatozoides enunciada en el ejemplo 2.1. Utilícese un esquema de dos niveles.

El producto residirá en los laboratorios biomédicos. Tendrá una interfaz sencilla. Proporcionará el servicio de automatizar la clasificación de los espermatozoides:

1. El sistema debe de procesar las imágenes procedentes del microscopio
 - a. Adquirir las imágenes de ficheros de tipo JPG, BMP, TIF, ...
 - b. Segmentar cada espermatozoide
 - c. Diferenciar en cada espermatozoide el núcleo y el halo.
 - d. Eliminar manchas y elementos espurios que hubiese en la imagen
 - e. Sólo se tomarán espermatozoides que estén completo su halo y su núcleo.
 - f. No se considerarán aquellos espermatozoides que estén juntos.
2. El sistema debe de clasificar los espermatozoides
 - a. Clasificarlos en un rango numérico de 0 a 1.
 - b. Correspondencia entre el rango numérico con las cinco categorías morfológicas de los espermatozoides.
3. El sistema debe producir un informe de resultados
 - a. Generar histograma continuo de las frecuencias de los espermatozoides del 0 al 1.
 - b. Tanto por ciento de espermatozoide del total en cada categoría. Se asociarán intervalos del rango del 0 al 1 con las categorías morfológicas de HG, HM, HP, SH y D.
 - c. Datos auxiliares: Fecha y hora, número de espermatozoides total del análisis, datos del cliente (Nombre, dirección,...).
 - d. Informe imprimible.

Ejemplo 2.4

Determinar las características para la aplicación de Respuesta en Frecuencia enunciada en el ejemplo 2.2. Utilícese un esquema de dos niveles.

1. El sistema debe capturar un circuito electrónico
 - a. Interacción con el usuario para determinar el circuito analógico
 - b. Determinar la FDT del circuito lineal.
2. El sistema debe realizar análisis en frecuencia
 - a. Parámetros de la respuesta en frecuencia (Rango de frecuencia, intervalo en el cálculo, lineal o en décadas)
 - b. Presentación gráfica del diagrama de Bode.

Otra sugerencia, resulta que a veces, las características del sistema son básicamente equivalentes a los nombres de los Casos de Uso. No hay que olvidar que los casos de uso, como se verá más adelante, hacen referencia a requisitos. Por tanto, los documentos de Visión y de Casos de Uso son complementarios. En este sentido, se debe evitar la repetición de conceptos, para mantener fluidez en los documentos y facilitar la actualización. Una manera eficiente es realizar referencia cruzadas a los otros documentos. Por ejemplo, se puede mencionar una característica y ampliar sus aspectos en un caso de uso o en el documento de Especificaciones Complementarias.

En la página *Web* de la asignatura está las plantillas de los documentos de recogida de los requisitos. El documento de Visión y Alcance está formado por la siguiente tabla de contenidos:

1 REQUERIMIENTOS DEL NEGOCIO

- 1.1 BACKGROUND
- 1.2 OPORTUNIDAD DEL NEGOCIO
- 1.3 OBJETIVOS ESENCIALES Y CRITERIOS DE ÉXITO
- 1.4 NECESIDADES DE LOS USUARIOS
- 1.5 RIESGOS DEL NEGOCIO

2. VISIÓN DE LA SOLUCIÓN

- 2.1 VISIÓN GLOBAL.....
- 2.2 PRINCIPALES CARACTERÍSTICAS.....
- 2.3 HIPÓTESIS Y DEPENDENCIAS.....

3. ALCANCE Y RESTRICCIONES

- 3.1. CARACTERÍSTICAS PRINCIPALES EN LA PRIMERA VERSIÓN
- 3.2. MEJORAS EN LAS SIGUIENTES VERSIONES
- 3.3. LIMITACIONES Y EXCLUSIONES

4. CONTEXTO DEL NEGOCIO

- 4.1 PERFILES DE POSIBLES CLIENTES EN EL PROYECTO.....
- 4.2 PRIORIDADES DEL PROYECTO
- 4.3 AMBIENTE DE OPERACIÓN

En cada apartado se orienta al desarrollador a que responda una serie de preguntas con el objeto de identificar los aspectos más relevantes del proyecto. El documento está formado en cuatro apartados: 1) hace referencia a la oportunidad de este proyecto, 2) analiza las características de la nueva aplicación, 3) tiene en cuenta cuáles van a ser sus limitaciones y 4) analizar las condiciones en el que se va a desarrollar el proyecto.

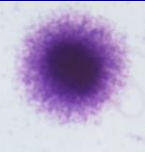
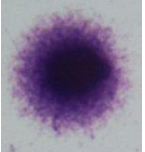
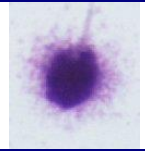
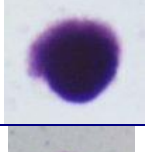

2.4 Glosario

El glosario almacena los términos y las definiciones del proyecto. Puede jugar el rol de diccionario de los datos relevantes. Es muy sorprendente que, a veces, el personal involucrado en el proyecto esté utilizando distintos nombres para la misma cosa.

El glosario podría almacenarse en una tabla de base de datos. Los campos que se puede utilizar son: nombre del término, la definición del campo, algún tipo de alias o cualquier otro campo que pueda ser considerado.

Ejemplo 2.5

Para la aplicación del análisis de los espermatozoides, el glosario de términos dado por los biomédicos fue:

Término	Definición e Información	Alias	Otros
Cromatina	Material nuclear (ADN + proteínas) que se tiñe de manera diferencial con ciertos colorantes (en nuestro caso: TODO LO QUE SE COLOREA, excepto la pieza intermedia y la cola)		
Tinción	Colorante que retiene la cromatina del espermatozoide. Una mayor densidad de color se traduce en una mayor masa de cromatina.		
Nucleoide	Núcleo celular parcialmente desproteinizado. Se compone de un "core" central y de un halo periférico.		
Halo	Superficie de la cromatina que se dispersa y que emana del core. Los halos tienen forma de corona circular o elíptica.		
Core:	Cuerpo central del nucleoide, que corresponde a la silueta del núcleo del espermatozoide. Los cores tienen forma circular o elíptica.	Núcleo	
Espermatozoide con Halo grande	Espermatozoide cuyo halo es igual o mayor que el diámetro menor del core.	HG	
Espermatozoide con Halo mediano	El grosor del halo está comprendido entre: mayor que 1/3 del diámetro menor del core y menor que el diámetro menor del core	HM	
Espermatozoide con Halo pequeño	El grosor del halo es igual o menor que 1/3 del diámetro menor del core	HP	
Espermatozoide Sin Halo	Espermatozoide que carece de halo de dispersión de la cromatina	SH	
Degradada(D):	Aquellos que sin mostrar halo, presentan la cabeza fragmentada en gránulos o muestran una tinción muy débil.	D	

2.5 Casos de uso

Los casos de uso son un mecanismo para ayudar a entender y describir los requisitos de forma simple y entendible para todo el personal involucrado. Fue introducido por Jacobson. En UP, el Modelo de Casos de Uso es un artefacto de la disciplina de Requisitos.

Definiciones para entender los casos de uso:

- **Actor** es algo (persona o sistema informatizado) que interactúa con el sistema.
- Un **escenario** es una secuencia específica de acciones e interacciones entre los actores y el sistema. Un escenario recoge los pasos que

dinamizan al sistema, que pueden ser de tres tipos: a) Una interacción con el actor, b) Una validación (normalmente a cargo del sistema) y c) un cambio de estado realizado por el sistema.

- Un *caso de uso* es una colección de escenarios de éxitos y fallos relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo.

Lo que hay que desarrollar es cómo utilizando el sistema proporciona un valor añadido al usuario o cómo cumple con sus objetivos. No hay que pensar en los requisitos del sistema a modo de “lista de la compra” (que tenga capacidad de imprimir, la aplicación tiene acceso a Internet,...). Los desarrolladores son muy dados a trabajar sobre objetivos que nada interesa a los futuros usuarios del sistema.

¿Cómo debería cubrirse un caso de uso?. Las tareas pueden agruparse en distintos niveles de granularidad. Para un primer nivel de análisis de los requisitos de una aplicación, hay que centrarse en los casos de uso al nivel de procesos del negocio elementales (*EBP, Elementary Business Processes*).

Hay que añadir valor observable y cuantificable al negocio y llegar a un acuerdo para que el sistema y los datos se encuentren en un estado estable y consistente. Por ejemplo, en una aplicación sobre la Declaración de la Renta, que el usuario se identifique al sistema no es un *EBP*. En cambio, si es un *EBP* calcular si la declaración es positiva o negativa. Hay tareas como “Identificarse al sistema y ser validado” que no añade valor observable o cuantificable al negocio. Se pueden definir subobjetivos que den soporte al objetivo del usuario. Sin embargo, no suele ser útil escribir casos de uso para las subfunciones, sólo se hace cuando ésta se repite o es una precondition en muchos casos. Centrándose en la guía *EBP* y en los requisitos funcionales, éstos orientan a realizar las siguientes tareas:

1. Encontrar los objetivos del usuario.
2. Definir un caso de uso para cada uno de estos objetivos.

2.5.1 Descubrir los actores, los objetivos y los casos de usos

Los pasos a seguir son:

1. Elegir los límites del sistema.
2. Identificar los actores principales.
3. Para cada uno de ellos, identificar sus objetivos de usuario.
4. Definir los casos de uso que satisfagan los objetivos de usuario.

Los actores y los objetivos se identifican a la vez. Una batería de preguntas útiles para encontrar los actores principales y sus objetivos pueden ser:

¿Quién arranca y para el sistema?

¿Quién se encarga de la administración del sistema?

¿Quién gestiona a los usuarios y la seguridad?

¿Es un actor “el tiempo” de forma que haya que dar una respuesta en un tiempo determinado?

¿Existe un proceso de control que reinicie el sistema si falla?

¿Quién evalúa la actividad o el rendimiento del sistema?

Otra forma de localizarlos es haciendo una lista de actor-objetivos. Por ejemplo, las siguientes tablas muestran las listas de actor-objetivos para las aplicaciones de análisis de imágenes biomédicas y para el análisis en frecuencia de los filtros analógicos.

Tabla 2-2 Lista de actor-objetivo para el análisis de imágenes de espermatozoides

Actor	Objetivos
Biomédico	Clasificar los espermatozoides en cinco tipos distintos
Microscopio robotizado	Entregar al sistema las imágenes de espermatozoides adquiridas

Tabla2-3 Lista de actor-objetivo para la aplicación de respuesta en frecuencia

Actor	Objetivos
Ingeniero	Respuesta en frecuencia de circuitos analógicos lineales

Otro enfoque para ayudar en la búsqueda de los actores, objetivos y casos de uso es identificar los eventos externos que llegan al sistema, ¿Cuáles son?, ¿de dónde proceden y por qué?. Una lista del evento externo, quién lo produce y cual es el objetivo son presentados para las dos problemáticas abordadas.

Tabla 2-4 Lista evento-actor-objetivo para la aplicación de análisis de los espermatozoides

Evento Externo	Parte del Actor	Objetivo
introducir un directorio de imágenes de espermatozoides	Biomédico	clasificar los espermatozoides que hay en cada imagen
Adquirir las imágenes	Microscopio robotizado	Generar la entrada al sistema de clasificación

Tabla 2-5 Lista evento-actor-objetivo para la aplicación de respuesta en frecuencia

Evento Externo	Parte del Actor	Objetivo
introducir el esquema de un filtro analógico	Ingeniero electrónico	Determinar la respuesta en frecuencia

En general, se define un caso de uso, de nivel EBP, por cada objetivo que tiene un actor. Obviamente se han identificado algunos casos de uso y éstos ni están completos, ni son perfectos. Habrá que refinar en otra iteración.

Como norma, los casos de uso son nominados comenzando con un verbo.

En la elaboración de los casos de uso, no deben de existir intermediarios entre los usuarios finales y los programadores. Los usuarios deben tener dedicación completa al proyecto.

Los casos de uso se escriben no considerando la interfaz del usuario, hay que centrarse en la intención.

2.5.2 La plantilla de documento de Casos de Uso

Los casos de uso de caja negra son la forma más recomendada y común; no describen el funcionamiento interno del sistema, ni sus componentes o diseño, sino que especifica cuáles son sus responsabilidades; muy en la línea del análisis orientado a objetos.

A través de la definición de las responsabilidades del sistema con los casos de uso de caja negra, es posible especificar **qué** debe hacer el sistema (los requisitos funcionales) sin decidir **cómo** lo hará (el diseño).

Los casos de uso se escriben con diferentes formatos:

- Breve: resumen conciso, normalmente del escenario principal con éxito.
- Informal: Formato de párrafo en estilo informal.
- Completo: el más elaborado y el que se va a tratar a continuación.

2.5.2.1 Elementos fundamentales del informe completo

La plantilla dejada en la página web de la asignatura tiene el siguiente índice:

1.	<u>IDENTIFICACIÓN DEL CASO DE USO</u>
2.	<u>DEFINICIÓN DEL CASO DE USO</u>
2.1.	<u>ACTOR</u>
2.2.	<u>DESCRIPCIÓN</u>
2.3.	<u>PRECONDICIONES</u>
2.4.	<u>POSTCONDICIONES</u>
2.5.	<u>CURSO DE ÉXITO</u>
2.6.	<u>CURSO ALTERNATIVO</u>
2.7.	<u>EXCEPCIONES</u>
2.8.	<u>INCLUSIONES</u>
2.9.	<u>PRIORIDAD</u>
2.10.	<u>FRECUENCIA DE CASO DE USO</u>
2.11.	<u>REGLAS DEL NEGOCIO</u>

2.12.	REQUERIMIENTOS ESPECIALES
2.13.	HIPÓTESIS DE PARTIDA
2.14.	NOTAS Y DOCUMENTOS ABIERTOS

El primer apartado hace referencia a un identificador que se le da al caso de uso, para su reconocimiento particular. Sin duda, la parte importante está en el segundo apartado, en la definición del caso de uso.

Se empieza definiendo al actor. El actor principal es el que recurre a los servicios del sistema para cumplir un objetivo concreto. El caso de uso captura el comportamiento relacionado con la satisfacción de los intereses del personal indicado. Un estilo habitual es poner con mayúsculas los nombres de los actores para facilitar sus identificaciones.

Las precondiciones establecen lo que siempre debe cumplirse antes de comenzar un escenario del caso de uso. El escenario de caso de uso recoge la dinámica de éste. Su evolución temporal depende de los eventos generados. Las postcondiciones son las acciones que se prevén tener finalizadas al acabar el caso de uso.

El curso de éxito establece que debe de cumplirse cuando el caso de uso se completa con éxito. El curso debería satisfacer las necesidades de todo el personal involucrado. En él se establece el flujo básico, el cual describe el camino de éxito típico que satisface los intereses del personal involucrado. Normalmente no incluye ninguna condición o bifurcación. Hay que intentar posponer todas las sentencias condicionales y de bifurcación a las secciones externas. Se utilizará una enumeración del tipo 1.0, 2.0, ... para cada una de las acciones del flujo básico.

Los cursos alternativos que pueden ocurrir dentro de este caso el uso, describen cualquier diferencia en la secuencia de los pasos de éxito. Se enumera cada curso alternativo en la forma "X.Y", donde "X" identifica el paso del caso de uso de éxito y la "Y" es un número de serie para la alternativa y mayor que cero. Por ejemplo, "5.3" indicarían el tercer curso alternativo para la etapa número 5 del caso de uso.

Las excepciones son interrupciones generadas de forma inesperada, cuya causa es un mal funcionamiento en la ejecución del curso básico. Por ejemplo, hay una falta de memoria, no existe el fichero indicado, hay errores de desbordamiento matemático, etc. También se indicará qué se propone para resolverlo. Para cada excepción se enumerará en la forma "X.Y.E.Z", donde "X" identifica el paso del caso del uso, Y indica (0) normal o el curso alternativo (>0) durante los cuales esta excepción podría ocurrir, "E" indica una excepción, y "Z" es un número de serie para las excepciones. Por ejemplo "5.0.E.2" indicaría la segunda excepción para el curso normal para el caso número 5 del uso.

Entre el camino de éxito y las extensiones (alternativa y excepciones) debería cubrirse "casi" todos los intereses del personal involucrado.

Los requisitos especiales o no funcionales suelen ir en el documento de Especificaciones Complementarias.

Ejemplo 2.6

Rellenar el caso de uso de *ClasificarEspermatozoides* de la aplicación de análisis de las imágenes.

Actores:	Biomédico
Descripción:	Analizar las imágenes con espermatozoides humanos y clasificarlos
Precondiciones:	Las imágenes están disponibles en un único directorio
Postcondiciones:	Informe de resultados de la clasificación
Curso normal:	1.0. El biomédico carga las imágenes de la muestra en un directorio. 2.0. Inicializa la aplicación y entrega al sistema, mediante el GUI, el directorio donde se encuentran las imágenes a analizar. 3.0. El biomédico introduce el nombre del fichero donde desea que se escriba los resultados de la clasificación. 4.0. El sistema procesa las imágenes y clasifica los espermatozoides. 5.0. El biomédico espera a que el sistema clasifique los espermatozoides
Curso alternativo:	4.1. Interacción entre el sistema y el biomédico. El biomédico desea analizar y clasificar paso a paso cada una de las imágenes
Excepciones:	2.0.E.1 No hay imágenes en el directorio. 3.0.E.1 Es ilegal el nombre del fichero de resultados. 4.0.E.1 Hay errores o falla el módulo de análisis y de clasificación.
Inclusiones:	
Prioridad:	Máxima. Núcleo del sistema.
Frecuencia de uso:	Podría ser casi continuo.
Reglas de negocio:	
Requerimientos especiales:	Los tiempos de análisis de cada imagen por debajo de los 5 segundos. Internacionalización del lenguaje de texto que se muestra.
Suposiciones de partida:	Se sigue el protocolo de adquisición definido
Notas y documentos:	Documento de adquisición de las imágenes a definir

Ejemplo 2.7

Rellenar el caso de uso *RespuestaFrecuencia* de la aplicación de análisis de los filtros electrónicos.

Actores:	Ingeniero electrónico
Descripción:	El ingeniero elige el tipo de filtro lineal e introduce los valores de los componentes analógicos, luego elige los parámetros de la respuesta en frecuencia y la aplicación le devolverá la respuesta en frecuencia en un diagrama de Bode
Precondiciones:	El ingeniero conoce y sabe todos los parámetros del filtro lineal y de la respuesta
Postcondiciones:	Se presentará en diagrama de Bode la respuesta en frecuencia del filtro capturado
Curso normal:	1.0.El ingeniero introduce el circuito eléctrico (mirar caso de uso incluido de capturar el circuito) 2.0. La aplicación pide el rango de frecuencias y el intervalo aplicado en la determinación de la respuesta en frecuencia. Además se le solicitará si desea una presentación en decibelios o lineal. 3.0. La aplicación calcula la FDT del circuito y determina la respuesta en frecuencia. 4.0. Los resultados son presentados en un diagrama de Bode (módulo/argumento).

Curso alternativo:	1.1 El circuito es capturado desde un fichero de descripción de componentes electrónicos tipo * CIR
Excepciones:	3.0.E.1 Hay algún error de desbordamiento o de división por cero en el cálculo de la respuesta en frecuencia. Se le notificará del error matemático al usuario
Inclusiones:	Captura del circuito
Prioridad:	Máxima. Núcleo del sistema.
Frecuencia de uso:	Podría ser casi continuo.
Reglas de negocio:	
Requerimientos especiales:	
Suposiciones de partida:	
Notas y documentos:	Documento de adquisición de los circuitos

2.5.3 Los casos de uso y la notación UML

Los Diagramas de Casos de Uso, dados por UML, son secundarios. Los Casos de Uso son documentos. Dibujar un diagrama de casos de uso junto con la lista actor-objetivos es complementario al documento.

En UML, los actores son representados, generalmente, por un “muñeco”, pero puede haber variaciones dependiendo del tipo de actor. Dentro de los actores están los principales, los actores de apoyo (aquellos que proporcionan un servicio al sistema) y los actores pasivos (los que están interesados en el caso de uso, pero no son ni el principal ni de apoyo). Por ejemplo, en un aplicación financiera, el actor principal puede ser un empleado de banca, la impresora sería un actor de apoyo y la Agencia Tributaria un actor pasivo.

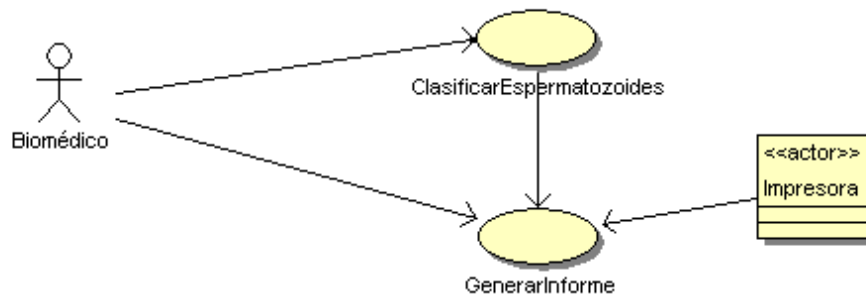
Los “muñecos” suelen emplearse para actores humanos. Cuando los actores son otros sistemas informáticos o equipos en general, se representa mediante un cuadrado, con el estereotipo¹ <<actor>>.

Los casos de uso se representan mediante una elipse con el nombre dado. Las relaciones entre los actores y los casos de uso se unen mediante líneas que pueden llevar un estereotipo.

Ejemplo 2.8

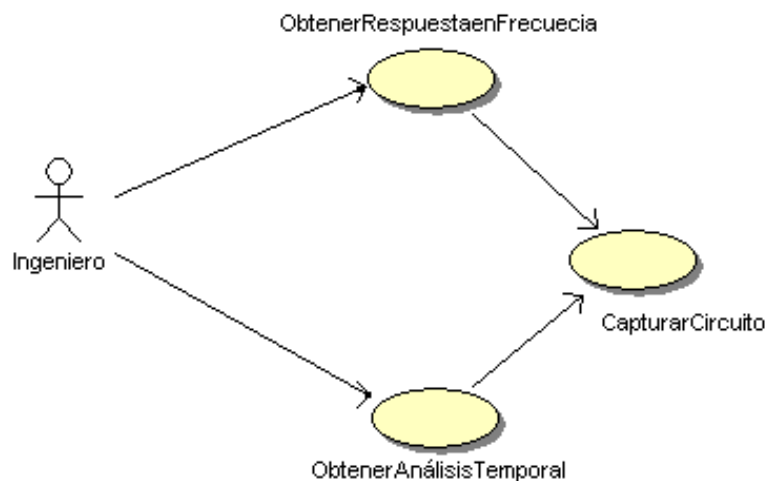
Diagrama del caso de uso de la aplicación de análisis de los espermatozoides

¹ Un estereotipo es una variante de un elemento de UML. Hay estereotipos estándares, definidos dentro del UML y hay otros que se pueden definir específicamente en un proyecto. Los estereotipos se identifican por una palabra clave entre los signos << y >>.



Ejemplo 2.9

Diagrama de caso de uso para la aplicación de respuesta en frecuencia



Los casos de uso no son orientados a objetos. Los casos de uso podrían ser utilizados dentro de los métodos estructurados. Es sólo una herramienta para capturar los requisitos. En UP serán la entrada para las actividades de AOO/D.

A los casos de uso se les puede acompañar con una lista de características de alto nivel del sistema. Aunque ésta es parte del documento de Visión y Alcance.

2.5.4 Casos de uso en el UP

Los casos de uso en el UP fomentan el desarrollo dirigido por los casos de uso. Éstos son una parte importante de la planificación iterativa. Los casos de uso dirigen el diseño.

UP diferencia entre los casos de uso del sistema (los vistos hasta ahora) con los casos de uso del negocio. Por ejemplo, en un restaurante, un caso de uso del negocio es

servir una comida. Esta actividad incluye no sólo la aplicación informática sino todo lo concerniente a este proceso: pedidos, colocación de la mesa, camareros,... *Rational Rose* al iniciar un proyecto, siguiendo UP, realiza el diagrama de UML que aparece en la figura; dando a entender una relación más amplia entre los casos de uso de la aplicación informática y su entorno ambiental.

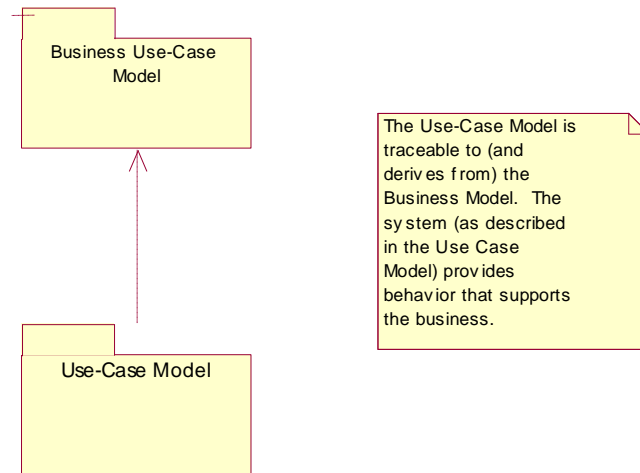


Figura 2. 1 Casos de uso de la aplicación y del negocio

Los casos de uso empiezan en la primera iteración, en el inicio, y son refinados en las siguientes iteraciones de la fase de elaboración. La tabla muestra un análisis de los requisitos de forma escalonada, donde se dirige por el alto riesgo y con la realimentación que incluye la evaluación del usuario, las pruebas y el conocimiento de lo que no se conocía hasta ese momento.

Tabla 2-6 Muestra del esfuerzo de los requisitos a lo largo de las primeras iteraciones; no es una receta

Comentarios y nivel de esfuerzo de los requisitos						
Disciplina	Artefacto	Inicio	Elab 1	Elab 2	Elab 3	Elab 4
		1 semana	4 semanas	4 semanas	3 semanas	3 semanas
Requisitos.	Modelo de Casos de Uso	2 días de taller de requisitos. Se identifican por el nombre de la mayoría de los casos de uso y éstos se resumen en un párrafo breve. Sólo el 10% se escribe en detalle.	Al iniciar esta iteración, tiene lugar un taller de requisitos de 2 días. Se obtiene un mejor entendimiento y retroalimentación a partir del trabajo de implementación, entonces se completa el 30% de los casos de uso en detalle.	Al iniciar esta iteración, tiene lugar un taller de requisitos de 2 días. Se obtiene una mejor comprensión y retroalimentación a partir del trabajo de implementación, entonces se completa el 50% del los casos de uso en detalle.	Repetir, se completa el 70% de todos los casos de uso en detalle.	Repetir con la intención de clarificar y escribir en detalle del 80-90% de los casos de uso. Sólo una pequeña parte de éstos se construyen durante la elaboración; el resto se aborda durante la construcción.
Diseño.	Modelo de Diseño.	Nada.	Diseño de un pequeño conjunto de requisitos de alto riesgo significativos desde el punto de vista de la arquitectura.	Repetir.	Repetir.	Repetir. Deberían ahora estabilizarse los aspectos de alto riesgo significativos para la arquitectura.
Implementación.	Modelo de implementación (código, etc.)	Nada.	Implementar esto.	Repetir. Se construye el 5% del sistema final	Repetir. Se construye el 10% del sistema final	Repetir. Se construye el 15% del sistema final.
Gestión del proyecto.	Plan de Desarrollo de SW.	Estimación muy imprecisa del esfuerzo total	La estimación comienza a tomar forma.	Un poco mejor ...	Un poco mejor ...	Ahora se pueden establecer racionalmente la duración global del proyecto, los hitos más importantes, estimación del coste y esfuerzo.

2.6 Especificaciones complementarias

No es suficiente con escribir los casos de uso y el documento de Visión y Alcance. Existen otros requisitos que no son necesarios de identificar con los usuarios, como los relacionados con la fiabilidad, el rendimiento, la documentación, el empaquetado, las licencias, etc. Éstos se recogen en el informe de las Especificaciones Complementarias, i.e. los que comprenden con los atributos o requisitos de calidad “URPS+”. Estos incluyen:

- Requisitos URPS+
- Informes
- Restricciones de HW y SW
- Idiomas, unidades, estándares
- Empaquetado
- ...
- y reglas del dominio o negocio

La calidad del SW no se refiere a los requisitos de funcionalidad, habla de la fiabilidad del sistema que influye en la arquitectura de la aplicación.

Las reglas del dominio dictan el modo en el que podrían operar un dominio o negocio. Con frecuencia, resulta útil identificar y registrar aquellas reglas del dominio que afectan a los requisitos, ya que pueden clarificar el contenido de un caso de uso ambiguo o incompleto.

Hay que evitar restricciones de diseño temprano, especialmente en la fase de inicio. Todavía se tiene poco conocimiento.

La plantilla empleada para este informe tiene la siguiente tabla:

1.INTRODUCCIÓN
1.1 PROPÓSITO.....
1.2 CONVENCIONES DEL DOCUMENTO.....
1.3 SUGERENCIAS PREVISTAS DE TIPO DE PERSONAL QUE VA A LEER ESTE DOCUMENTO.....
1.4 ALCANCE DEL PROYECTO.....
1.5 REFERENCIAS
2.DESCRIPCIÓN GENERAL
2.1 PERSPECTIVA DEL PRODUCTO.....
2.2 CARACTERÍSTICAS DEL PRODUCTO
2.3 CLASES DE USUARIOS Y CARACTERÍSTICA
2.4 AMBIENTE DE OPERACIÓN
2.5 RESTRICCIONES DE DISEÑO O DE IMPLEMENTACIÓN
2.6 DOCUMENTACIÓN DEL USUARIO
2.7 SUPOSICIONES Y DEPENDENCIAS.....
3.CARACTERÍSTICAS DEL SISTEMA.....
3.1 PRIMERA CARACTERÍSTICA DEL SISTEMA.....
3.2 SEGUNDA CARACTERÍSTICA DEL SISTEMA (Y SIGUIENTES)
4.REQUISITOS DE LAS INTERFASES EXTERNAS.....
4.1 INTERFASES CON LOS USUARIOS
4.2 INTERFASES DEL SOPORTE FÍSICO (HARDWARE)

4.3 INTERFASES DEL SOPORTE LÓGICO (SOFTWARE)
4.4 INTERFASES DE COMUNICACIONES
5. OTROS REQUISITOS NO FUNCIONALES.....
5.1 REQUISITOS DE RENDIMIENTO
5.2 REQUISITOS DE FIABILIDAD.....
5.3 REQUISITOS DE SEGURIDAD.....
5.4 ATRIBUTOS DE CALIDAD DEL SOFTWARE
6. OTROS REQUISITOS.....

Este documento debe de complementar a los de Visión y Alcance y a los Casos de Uso. Se evitará la duplicación de información entre todos los documentos de los requisitos, realizando conexiones y referencias a los otros documentos, cuando se consideren que deberían ser tratados con mayor profundidad en otros informes.

En este documento se hablará especialmente de los requisitos no funcionales y del tipo de arquitectura propuesta. Respecto a los requisitos no funcionales a tratar podrían ser, por ejemplo:

- Plataformas
- Herramientas de desarrollo
- Control de errores
- Instalación
- Multilenguajes
- ...

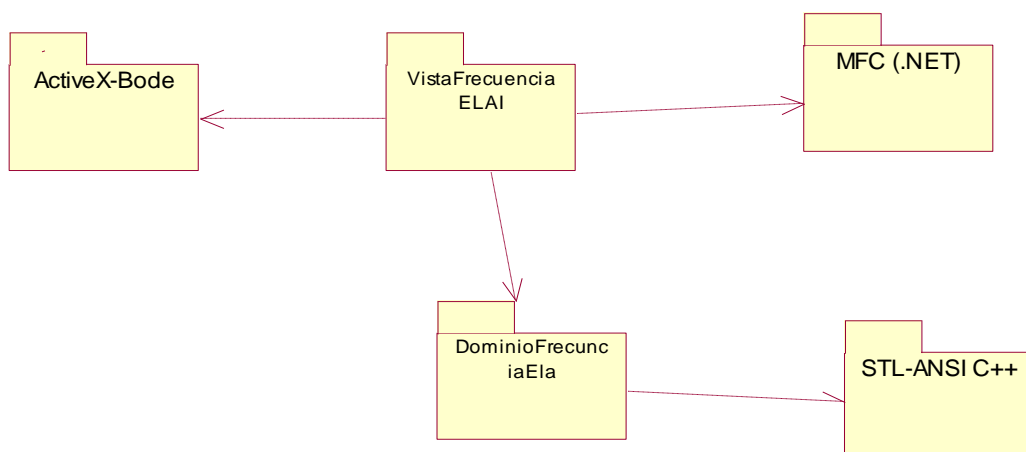
Ejemplo 2.10

Rellenar las especificaciones complementarias para la aplicación de respuesta en frecuencia

Este documento es complementario al de Visión y Alcance del producto FrecuenciaELAI. Se describe los requisitos no funcionales. En éste se hace referencia a la arquitectura y a los aspectos relacionados con:

- Procesamiento multihebra en las el procesamiento de la respuesta en frecuencia
- Multilenguaje, al menos español e inglés
- Lenguajes de programación y librerías de servicio: ANSI C++, STL y Qt
- Tipos de plataforma: Linux y Microsoft.
- Control de errores: excepciones con try-catch.
- Instalación:
- Documentación: del código mediante Doxygen, manuales de usuario en pdf.

La arquitectura propuesta es del tipo:



2.7 Problemas

1. Actividades a desarrollar en la fase de inicio dentro de UP.
2. Tipos de requisitos.
3. ¿Cómo capturar la lista de características principales?, ¿Dónde deben de aparecer?.
4. Técnicas para la identificación de actores, objetivos y casos de uso.
5. Estructura del documento de Casos de Uso.
6. Casos de Uso: documento, diagrama UML de Casos de Uso y lista de actores objetivos.
7. Evolución de las disciplinas con sus artefactos en la fase de inicio y de elaboración.
8. ¿Qué se recoge en el documento de Especificaciones Complementarias?.

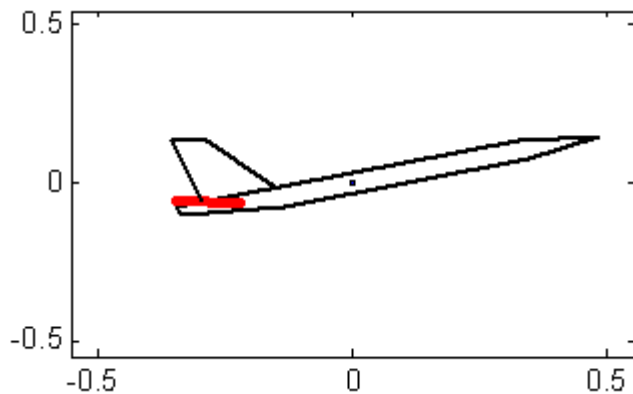
Problema 2.1

Los simuladores simplificados de los procesos monovariantes (SISO) se basan en tener modelos lineales e invariantes en el tiempo (LTI) para dichos sistemas. Éstos se caracterizan por tener, cada uno de ellos, una función de transferencia, FDT, en el dominio en Z:

$$M(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

Donde a_i y b_i son constantes y conocidas. De forma que ante una señal de excitación, $x(t)$, de tipo test (impulsional, escalón, rampa, parábola, senoidal, ...), se determina la salida haciendo la anti-transformada en Z, dando paso al siguiente algoritmo:

$$y_k = \frac{1}{a_0} \left((b_0 x_k + b_1 x_{k-1} + \dots + b_m x_{k-m}) - (a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_n y_{k-n}) \right)$$



El índice k da el valor de la muestra en el instante $t=kT$, siendo T el intervalo de tiempo empleado entre muestra y muestra. Para los sistemas físicos reales, los valores de k negativos son iguales a cero. Se pide, dado un proceso LTI-SISO y conocido los coeficientes a_i y b_i , realizar la recogida de requisitos:

1. Visión y Alcance. Lista de características principales (jerarquía a dos niveles)
2. Glosario
3. Lista de evento-actor-objetivos
4. Casos de uso
5. Especificaciones complementarias
 1. El sistema debe de simular la dinámica del control del avión ante una maniobra.
 - 1.a. El sistema debe tener la función de transferencia del modelo.
 - 1.b. Se debe definir el tipo de maniobra a realizar
 - 1.c. Opcionalmente se podrá introducir algunas perturbaciones
 2. El sistema visualizará la evolución temporal del proceso.
 - 2.a El sistema debe visualizar la orientación del avión en función del tiempo.
 - 2.b El sistema debe visualizar la posición del timón en función del tiempo.

Glosario: Función de transferencia, maniobra, perturbación, variable de salida, periodo de muestro, tiempo de simulación.

Evento	Actor	Objetivo
Definir función de transferencia	Ingeniero	Introducir modelo del proceso
Definir maniobra	Ingeniero	Definir la señal de entrada
Definir parámetro de la simulación	Ingeniero	Tiempo de evolución del proceso
Visualización de resultado	Maquina	Mostrar en gráficas los resultados de la simulación

Actores:	Ingeniero de Control
Descripción:	El ingeniero introduce la función de transferencia, la maniobra y opcionalmente las perturbaciones y el simulador debe de visualizar la dinámica del proceso ante las entradas dadas.
Precondiciones:	El ingeniero conoce la función de transferencia y las señales de mando y perturbaciones.
Poscondiciones:	El simulador visualiza la dinámica del proceso
Curso normal:	1.0. El ingeniero introduce la función de transferencia del proceso. 2.0. El ingeniero introduce el tipo de excitación y define opcionalmente las perturbaciones. 3.0. El ingeniero introduce los tiempos de simulación. 4.0. El sistema calcula la dinámica de simulación. 5.0. Visualizar la evolución temporal de la señal de salida.
Curso alternativo:	1.1 El ingeniero introduce el modelo a través de un fichero. 5.1 La salida es guardada en fichero.
Excepciones:	4.0.E.1 Hay algún problema en el cálculo de la dinámica.
Inclusiones:	
Prioridad:	Máxima. Núcleo del sistema.
Frecuencia de uso:	Podría ser casi continuo.

Reglas de negocio:	
Requerimientos especiales:	
Suposiciones de partida:	
Notas y documentos:	

Problema 2.2

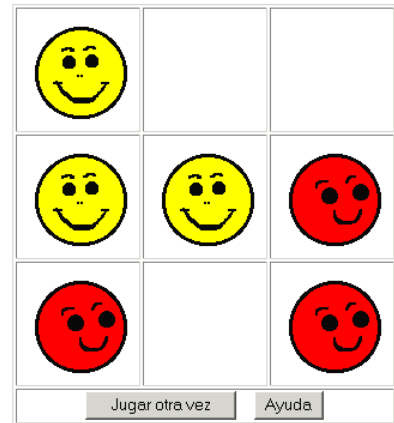
Recoger los requisitos más importantes de una aplicación para el alquiler automático de películas de cine en DVD. Realizar las siguientes tareas:

1. Lista de características principales (jerarquía a dos niveles).
2. Lista de evento-actor-objetivos.
3. Diagrama de Casos de Uso.

Problema 2.3

Rellenar los siguientes artefactos de la fase de Inicio para la aplicación del juego de las tres en raya:

1. Lista de características principales (jerarquía a dos niveles).
2. Glosario
3. Lista de evento-actor-objetivos
4. Casos de Uso.

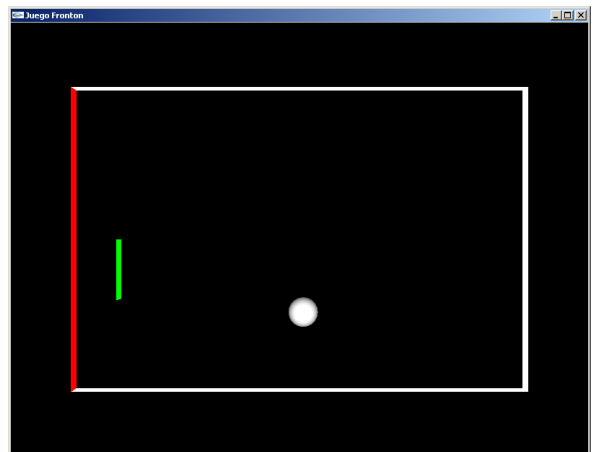


Problema 2.4

Realizar la recogida de requisitos para el juego del frontón.

En jerarquía a dos niveles se plantea las siguientes características:

1. Un jugador juega al frontón con una raqueta y una pelota con tres paredes
 - a. El sistema debe de permitir mover la raqueta según el deseo del usuario dentro de un área restringida
 - b. El sistema debe simular la interacción entre la pelota contra las paredes y la raqueta.
 - c. El sistema debe visualizar el escenario de paredes, pelota y raqueta.
 - d. El sistema debe de terminar la partida cuando la pelota cruce una determinada línea, al que se le llamará línea roja.



Los términos para el glosario serían:

Frontón, Pelota, Pared, Raqueta, LíneaRoja, ...

La lista de evento-actor-objetivo sería:

Evento	Actor	Objetivo
Mover raqueta	Jugador	Dar a la pelota
Iniciar saque	Jugador	Lanzar inicialmente la pelota
Interacción entre pelota-raqueta	Maquina	La pelota cambia la dirección de la velocidad
Interacción entre pelota-paredes	Maquina	La pelota cambia la dirección de la velocidad

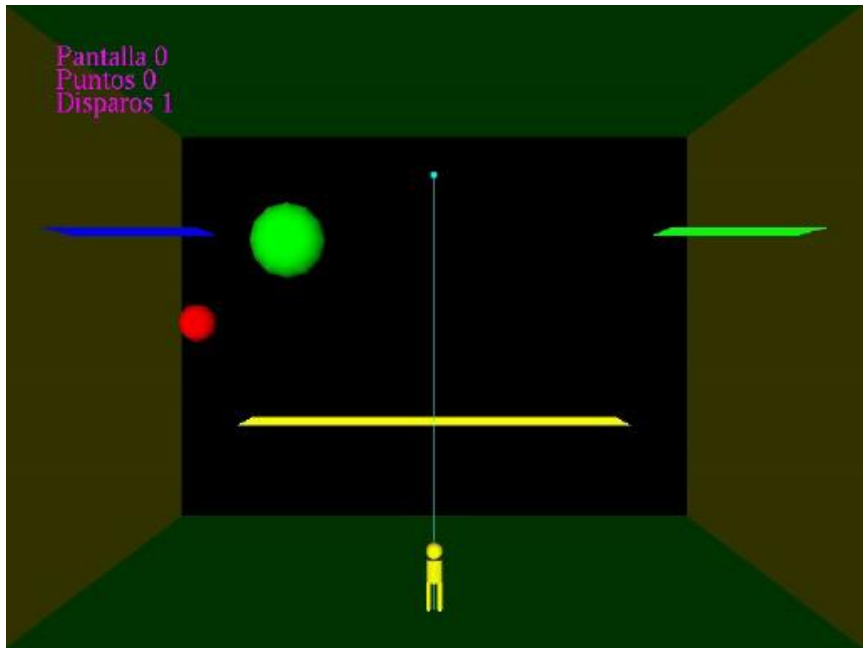
A este nivel y utilizando el criterio EBP se definirá el caso de uso *JugarAlFrontón*. Escribir el resumen de este caso de uso.

Actores:	Jugador
Descripción:	El jugador inicia una partida, con el saque de la pelota. El jugador pretenderá dar la pelota, antes de que ésta cruce la línea roja. La pelota rebota contra las paredes y contra la raqueta
Precondiciones:	El jugador conoce las reglas y sabe cuales son las teclas para mover la raqueta.
Poscondiciones:	El jugador ha terminado de jugar
Curso normal:	1.0. El jugador inicializa el saque de la pelota. 2.0. La pelota rebota contra las paredes. 3.0. El jugador mueve la raqueta. 4.0. El jugador acierta a dar a la pelota y entonces se va a 2.0.
Curso alternativo:	4.1 El jugador no acierta a dar a la pelota y se termina la partida.
Excepciones:	2.0.E.1 Hay algún problema en el choque entre pelota-pared. 4.0.E.1. Hay algún problema en el choque entre pelota-raqueta.
Inclusiones:	
Prioridad:	Máxima. Núcleo del sistema.
Frecuencia de uso:	Podría ser casi continuo.
Reglas de negocio:	
Requerimientos especiales:	
Suposiciones de partida:	
Notas y documentos:	

Problema 2.5

Rellenar los artefactos de la fase de Inicio para el juego del '*Pang*'. Éste trata de un escenario cerrado donde un jugador pretende eliminar las esferas existentes. El hombre dispara a las esferas, las cuales se dividen si son grandes, en caso contrario desaparecen. Además el jugador deberá de evitar que las esferas le toquen, ya que si le impacta perderá la partida. Se pide:

1. Lista de características principales (jerarquía a dos niveles).
2. Glosario
3. Lista de evento-actor-objetivos.
4. Casos de Uso.



Derecho de Autor © 2014 Carlos Platero Dueñas.

Permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; sin secciones invariantes, sin texto de la Cubierta Frontal, así como el texto de la Cubierta Posterior. Una copia de la licencia es incluida en la sección titulada "Licencia de Documentación Libre GNU".

La Licencia de documentación libre GNU (GNU Free Documentation License) es una licencia con [copyleft](http://www.gnu.org/copyleft/fdl.html) para [contenidos abiertos](http://www.gnu.org/copyleft/fdl.html). Todos los contenidos de estos apuntes están cubiertos por esta licencia. La versión 1.1 se encuentra en <http://www.gnu.org/copyleft/fdl.html>. La traducción (no oficial) al castellano de la versión 1.1 se encuentra en <http://www.es.gnu.org/Licencias/fdles.html>