# Introducción

El objetivo de este documento es presentar con detalle todo lo relacionado con la especificación de nuestro proyecto de Ingeniería del Software. Mediante este documento toda la funcionalidad, limitaciones de nuestro sistema, posibles ampliaciones y herramientas utilizadas, quedan reflejadas de manera formal para que puedan ser revisadas por el profesor que guía el proyecto.

# Índice

[1. Visión general del documento](#h.z7kdxlyj00e3)

[1.1. Ámbito del sistema](#h.gbuu1q3fypa7)

[1.2. Definiciones, Acrónimos y Abreviaturas](#h.ctqkchfwjl2l)

[1.2.1. Definiciones](#h.txncqxegy6zz)

[1.3. Referencias](#h.dumw2r4jhw35)

[2. Descripción general](#h.wcs7r728d1qd)

[2.1. Perspectiva del producto](#h.39dyl96x3pmp)

[2.2. Funciones del producto](#h.gxrirnqmy1gm)

[2.3. Características de los usuarios](#h.c35g45yy09p3)

[2.4. Restricciones](#h.j20r8trdqu3)

[2.5. Suposiciones y dependencias](#h.2a3czvng70c4)

[2.6. Requisitos futuros](#h.qsjf6868jxm8)

[3. Requisitos específicos](#h.kfnsxdme17ut)

[3.1. Interfaces externas de usuario](#h.55uzm3qjg3vc)

[3.2. Funciones](#h.kya28uv61rqi)

[3.2.1. Requisitos funcionales para un usuario invitado](#h.3f9p5rqxhox3)

[3.2.2. Requisitos funcionales para un usuario autenticado](#h.iibne9w0r7pq)

[3.2.3. Requisitos funcionales para un usuario administrador](#h.4s1fl8ebfm05)

[3.3. Requisitos de rendimiento](#h.7v2k9db2kdbh)

[3.4. Atributos del sistema](#h.2bbl3mzc8mbe)

[3.5.1. Seguridad](#h.5f8byp95vf4k)

[3.5.2. Portabilidad](#h.ntvbl7b6cr00)

[3.5.3. Mantenibilidad](#h.pefnpl3mfb49)

[3.5.4. Fiabilidad](#h.wj5px8ih7ysa)

[4. Apéndices](#h.xnlgx4wil428)

[4.1. Modelo RUP](#h.1jk0rebtww1q)

# 1. Visión general del documento

En esta sección vamos a dar una perspectiva general de la estructura del documento, el cual está dividido en cuatro secciones principales:

1. **Introducción**, en ella se comenta el propósito del sistema, su ámbito, la definición de los conceptos que se utilizan en él y los documentos externos que referenciamos.
2. **Descripción general,** en esta sección se explican todos los factores que afectan al producto y a sus requisitos.
3. **Requisitos específicos,** contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que los satisfaga.
4. **Apéndice,** contiene todo tipo de información relevante para la especificación pero que, propiamente, no forma parte de ésta.

## 1.1. Ámbito del sistema

El sistema va a ser un juego por navegador el cual hemos denominado Jugador Número 12. Éste pondrá al usuario en la piel de un hincha de fútbol, el cual tendrá la posibilidad nada más entrar de seleccionar su perfil dentro del juego, pudiendo elegir entre los siguientes:

* *El* ***ultra***, que con su espíritu conformará la primera línea de apoyo al equipo.
* *La* ***animadora de masas***,que intentará conseguir llenar los estadios.
* *El* ***empresario***,que representa las luchas en los despachos manejando grandes cantidades de dinero.

Al usuario se le dará la opción de unirse como aficionado a un equipo.

Una vez hecho esto el jugador podrá realizar acciones para preparar el partido, individuales o que hayan sido iniciadas por integrantes de su afición. Para poder realizarlas dispondrá de tres recursos:

* *El* ***ánimo***, o capacidad que tiene un hincha para animar.
* *La* ***influencia***, que refleja el renombre del hincha en la sociedad.
* *El* ***dinero***, que se irá incrementando con el tiempo y permitirá comprar diversos objetos.

Pero no sólo habrá acciones que preparen el partido, sino que el usuario podrá participar en el encuentro, es decir, se le dará la posibilidad de influir en la reacción del equipo durante el transcurso de éste haciendo que sea capaz de meter un gol o reponerse de un tanto del contrario. Para todo esto, el partido dispondrá de un sistema de turnos, y para poder participar en él, el usuario deberá comprar una entrada.

¿Qué pretendemos con este sistema? Tras la presentación del proyecto en mayo pretendemos refinar y modificar el proyecto para subirlo a la web con el fin de que toda la gente que le guste juegue y así nosotros poder sacar beneficio económico de ello. Pero no solo pensamos en este tipo de beneficio ya que realizar este proyecto nos aportará experiencia en el ámbito del desarrollo de aplicaciones web.

## 1.2. Definiciones, Acrónimos y Abreviaturas

En esta sección vamos a explicar las definiciones, acrónimos y abreviaturas utilizadas en el documento:

### 1.2.1. Definiciones

* **Hincha**: Un hincha es el partidario entusiasta de un equipo de fútbol o en general de cualquier deporte.
* **Rol o perfil**: Papel que desempeña el usuario dentro del juego, pudiendo elegir entre tres diferentes.

## 1.3. Referencias

1. Marina Bezares, “[Documento definición del juego”](https://docs.google.com/document/d/13eyi2Sa447NXmsxGQ2RQ6cNjF2xmwyemcddt7VxcZPA/edit#heading=h.bmol66efqo27)

# 2. Descripción general

## 2.1. Perspectiva del producto

Jugador Número 12 será un producto final e independiente de cualquier otro. Por ello no es necesario mostrar ninguna relación y bastará con la documentación restante del documento para su definición completa.

## 2.2. Funciones del producto

Mostraremos a continuación, en diferentes secciones, una visión a grandes rasgos de las funcionalidades que soportará el producto:

* **Mantenimiento de cuentas de usuario**: el sistema será capaz de almacenar y modificar los datos personales de los usuarios del mismo.
* **Comunicación interna**: permitirá a los usuarios y al propio sistema el envío, almacenamiento y recepción de mensajes privados.
* **Asignación del rol del usuario**: como ultra, empresario o movedora de masas, el sistema permitirá al usuario asumir dicho papel con sus correspondientes funcionalidades, acciones y atributos específicos.
* **Gestión de aficiones**: el producto admitirá la gestión de varias aficiones de fútbol por parte de los usuarios con diversas funcionalidades como añadir nuevos usuarios o crear eventos para los miembros asociados.
* **Facilitar información de contexto**: el sistema ofrecerá la capacidad de mostrar datos representativos para los usuarios como puede ser la clasificación de los equipos en un determinado momento.
* **Participación y resolución de partidos**: el producto permitirá a los usuarios participar mediante acciones en los diferentes encuentros y será capaz de generar el resultado de un partido virtual en función de los aportes durante y antes del mismo.

## 2.3. Características de los usuarios

Los requisitos mínimos que un usuario deberá cumplir para poder hacer uso del sistema “Jugador Número 12” son:

* **Nivel educacional**: el producto está destinado a un público general con un mínimo de educación básica y lingüística.
* **Experiencia**: los usuarios no necesitarán conocimientos previos acerca de juegos por navegador o estrategia para poder hacer uso del sistema. Se tratará de implementar la interfaz más sencilla posible para acercar el producto al mayor número de usuarios posible.
* **Experiencia técnica**: todo usuario del sistema deberá tener un mínimo de conocimientos sobre internet y navegadores web.

## 2.4. Restricciones

A continuación se presentará una lista de restricciones impuestas por los propios desarrolladores del producto:

* **Limitaciones de hardware**: inicialmente, el sistema estará pensado para funcionar en un servidor web de nivel básico/medio consumiendo el mínimo de recursos posibles para permitir el mayor número de usuarios simultáneos.
* **Operaciones paralelas**: el sistema debe ser capaz de administrar las transacciones concurrentes a la base de datos de forma correcta.
* **Funciones de auditoría:** se realizarán pruebas mínimas sobre el sistema para verificar la integridad y protección de los datos. Por ejemplo, comprobación de inyección SQL, ataques a la web, etc.
* **Funciones de control:** a cargo de varios supervisores estará la revisión de elementos como el repositorio, fechas de entrega, documentación, código, etc.
* **Lenguajes de programación**: se hará uso de las tecnologías de HTML, PHP, CSS, Javascript y MySQL. Todo ello se hará mediante el uso del framework Yii.
* **Requisitos de habilidad:** los miembros del grupo deberán tener unos conocimientos mínimos sobre el uso del framework y el repositorio que serán adquiridos durante una sesión didáctica.
* **Consideraciones acerca de la seguridad:** el sistema deberá incluir gran cantidad de pruebas sobre seguridad para prevenir la mayor cantidad de ataques posibles y mejorar la protección de datos de los usuarios.

## 2.5. Suposiciones y dependencias

Se han supuesto los requisitos para que el sistema trabaje correctamente con especificaciones estándar: resolución de pantalla media, navegadores actualizados en mayor o menor medida, etc. Es posible tratar de ampliar la capacidad de mercado optimizando la compatibilidad del sistema con navegadores antiguos aunque pueda suponer algún cambio en los requisitos.

## 2.6. Requisitos futuros

En un futuro se tratarán de implementar nuevas funcionalidades que requerirán una modificación de los requisitos iniciales:

* **Chat interno al juego**: para permitir la comunicación de los usuarios en tiempo real.
* **Aplicación móvil del sistema**: para aumentar la capacidad de negocio del mismo y permitir una mayor accesibilidad por parte de los usuarios.

# 3. Requisitos específicos

## 3.1. Interfaces externas de usuario

Las interfaces de usuario son las ventanas (formularios) con las que debe interactuar el usuario para realizar una operación determinada. Dicha manipulación será realizada por medio del teclado y el ratón.

Las interfaces de Jugador número 12 serán ventanas de navegador de Internet para ordenadores. Para poder jugar el usuario deberá de disponer de un navegador que soporte Javascript y CSS.  
Probaremos el sistema para que funcione correctamente en, al menos, los navegadores más comunes: Chrome y Firefox.

## 3.2. Funciones

En “Jugador Número 12” distinguiremos entre 3 tipos de usuarios: invitado, autenticado y administrador. Cada uno de ellos tendrá asociados distintos niveles de privilegios, definidos en la [sección 3.5.1](#id.seaa97ex7xtn), y por tanto, el comportamiento externo del sistema para cada uno de ellos será diferente.

### 3.2.1. Requisitos funcionales para un usuario invitado

Un usuario invitado es aquel que no está registrado en el juego, es decir, que no ha hecho login en el sistema con una cuenta previamente creada. Este tipo de usuario solo tendrá acceso a la página principal del juego y a los tutoriales. No podrá acceder a la zona de juego. La única función que puede llevar a cabo en el sistema es registrarse en él. Este requisito funcional forma parte de la arquitectura del sistema.

### 3.2.2. Requisitos funcionales para un usuario autenticado

Un usuario autenticado es aquel que ha hecho login en el sistema con una cuenta creada con anterioridad. Este tipo de usuario puede acceder a la zona de juego así como a sus datos personales.

Tendrá los siguientes requisitos funcionales, de **gestión de usuario**, a su disposición:

* **Consultar datos de usuario**. Los usuarios podrán consultar en una página llamada “mi perfil” sus datos de usuario introducidos. Éstos no serán compartidos, ni podrán ser vistos por ningún otro usuario. —Véase [seguridad](#id.aoyj9cf0q0p9).— Los datos que podrán consultar son: nombre, usuario, contraseña, e-mail registrado, personaje escogido y afición escogida. Este requisito funcional forma parte de la arquitectura.
* **Modificar datos de usuario.** Los usuarios tendrán la posibilidad de cambiar sus datos a través de la página de perfil. Podrán modificar aquellos datos indicados en el párrafo anterior, excepto el personaje. Si un usuario cambia la contraseña, se le pedirá que introduzca la contraseña actual antes de ingresar la nueva contraseña. Este requisito funcional no forma parte de la arquitectura.
* **Mensajería privada.** Un usuario registrado podrá usar el servicio de mensajería privada. Será capaz de mandar un mensaje a otro usuario, que solo lo podrán ver los jugadores implicados. También podrá recibir mensajes privados de otros usuarios. Para mandar un mensaje privado a otro usuario hace falta saber el nombre del destinatario. Podrá obtener un listado de los mensajes recibidos y enviados para consultarlos cuando quiera. Podrá también borrar un mensaje — o varios — del listado. Este requisito funcional no forma parte de la arquitectura.
* **Darse de baja.** Un usuario podrá, en su perfil, dar de baja su cuenta. Ésta se borrará y el usuario dejará de existir. Sus datos serán borrados de la base de datos. Este requisito forma parte de la arquitectura.

Tendrá los siguientes requisitos funcionales, de **gestión de hincha**, a su disposición:

* **Elegir el personaje**. Cada usuario podrá elegir entre 3 personajes[1](#id.6egka41eje35): animadora, empresario y ultra. Cada uno de ellos tendrá una forma de juego distinta. Al registrarse, el usuario deberá seleccionar qué personaje desea, y una vez elegido no lo podrá cambiar. Este requisito funcional forma parte de la arquitectura.
* **Consultar datos del personaje.** Los usuarios podrán consultar los datos relativos a su personaje, que serán el tipo de personaje elegido y recursos que posee. También podrán consultar su nivel y habilidades, pero lo hemos separado en otro epígrafe. Ésto se hará desde la página “mi perfil”. Este requisito funcional forma parte de la arquitectura.
* **Consultar habilidades, nivel y experiencia.** El personaje del jugador obtendrá experiencia. Cuando consiga una determinada cantidad de experiencia subirá de nivel. El usuario podrá consultar su nivel actual, la experiencia actual y cuánta experiencia le falta para llegar al siguiente nivel.  
  Cuando suba de nivel, el jugador podrá elegir diferentes habilidades[1](#id.6egka41eje35) para mejorar a su personaje. El usuario podrá consultar el árbol de habilidades total. Podrá saber qué habilidades cogidas del árbol tiene y cómo le afectan. Este requisito funcional forma parte de la arquitectura.
* **Elegir habilidades.** Cuando un jugador suba de nivel podrá elegir una nueva habilidad de árbol de habilidades. El jugador podrá saber qué habilidades tiene escogidas, cuales tiene bloqueadas y las que podrá escoger. Este requisito funcional forma parte de la arquitectura.
* **Ejecutar acciones individuales**. El jugador tendrá un abanico de acciones que puede ejecutar antes del partido. Éste, será más grande, cuanto más nivel tenga el personaje. El jugador podrá, gastando sus recursos, activar acciones disponibles de su abanico.   
  Las acciones que sean de efecto inmediato y permanente no se podrán volver a ejecutar, es decir, serán de un solo uso. Este requisito funcional forma parte de la arquitectura.

Tendrá las siguientes funcionalidades a nivel de **gestión de afición**:

* **Consultar datos generales de afición**. Los jugadores podrán visualizar los datos correspondientes a su afición como pueden ser la descripción y escudo del equipo asociado. Este requisito funcional forma parte de la arquitectura.
* **Unirse a una afición**. Los usuarios podrán unirse a una afición una vez creada su cuenta en el sistema y tras elegir el perfil de jugador deseado. Este requisito funcional forma parte de la arquitectura.
* **Cambiar de afición**. En caso de no estar satisfecho, a un usuario se le permitirá cambiar la afición a la que está asociado y pasar a colaborar con otra. Este requisito funcional no forma parte de la arquitectura.
* **Consultar miembros**. Los jugadores podrán consultar en todo momento los miembros asociados a una afición. Este requisito funcional forma parte de la arquitectura.
* **Consultar acciones de afición**. Un usuario tendrá la posibilidad de consultar las acciones activas de afición. Estas acciones, a diferencia de las individuales, serán creadas por un miembro de la afición y permitirán la participación del resto de componentes de la misma en ellas. Este requisito funcional forma parte de la arquitectura.
* **Participar en acciones de afición**. Una vez revisadas las acciones de afición, cada jugador podrá participar - haciendo uso de sus recursos - en aquellas que elija y hayan sido creadas por alguien de su misma afición. Este requisito funcional forma parte de la arquitectura.
* **Consultar clasificación**. Los usuarios tendrán la capacidad de consultar la clasificación de la liga para comprobar el estado de los equipos en cualquier instante. Este requisito funcional forma parte de la arquitectura.

A nivel de **gestión de partido** un usuario autenticado podrá:

* **Consultar información previa del partido**. Esto es, un jugador podrá visualizar datos como pueden ser el ambiente, nivel de equipos y aforo previsto para el encuentro antes de que éste tenga lugar.
* **Comprar entrada**. Los usuarios tendrán la posibilidad de comprar la entrada para asistir a un encuentro. En caso de no hacerlo, no podrán participar durante el mismo.
* **Asistir al partido**. Los jugadores podrán acudir al encuentro y seguir los sucesos del mismo.
* **Realizar acciones de partido**. Habiendo cumplido el requisito de comprar la entrada, los jugadores podrán participar durante un encuentro realizando diversas acciones. Esto influirá en el resultado del partido.

### 3.2.3. Requisitos funcionales para un usuario administrador

Un usuario administrador es aquel que tiene una mayor capacidad de acceso y modificación de datos del sistema a los cuales un usuario de menor rango no puede acceder con total libertad. Este tipo de usuario podrá hacer uso de todo lo mencionado con anterioridad y además poseerá ciertas funcionalidades extra.

* A nivel de **gestión de usuarios**: El administrador podrá crear, modificar y borrar los datos de cualquier usuario del sistema.
* A nivel de **gestión de hinchas**: El usuario tendrá la capacidad de crear, modificar o eliminar los datos del personaje asociado a cualquier usuario.
* A nivel de **gestión de aficiones**: Cualquier administrador podrá crear, modificar o borrar los datos de una afición como pueden ser las acciones activas o los miembros asociados.
* A nivel de **gestión de ligas**:

Administrar la liga. Los administradores tendrán la posibilidad de crear, modificar o eliminar cualquier dato referente a la liga. Esto es, por ejemplo, cambiar la posición de un determinado equipo en la clasificación.

Administrar los equipos. Este tipo de usuarios podrán crear, modificar y borrar cualquier equipo y los datos referentes al mismo.

## 3.3. Requisitos de rendimiento

A continuación se muestran algunos datos facilitados por la compañía creadora del juego de navegador Hattrick. Es información muy básica (ya que el resto está destinada sólo a administradores) pero suficiente para obtener un modelo de requisitos de rendimiento básicos.

En el caso de Hattrick cuentan con unos 600000 usuarios activos en el sistema y 660000 logins (285000 de los cuáles son únicos). En cuanto a la base de datos, reciben alrededor de 3500 escrituras por segundo y una media de 660 conexiones diarias (parte de las cuáles permanecen abiertas durante el día completo).

En base a los datos recopilados, y de cara al primer ciclo de desarrollo del sistema, se creará una aplicación que permita manejar como máximo unos 100 usuarios con 50 escrituras por segundo a la base de datos y una media de 25 conexiones diarias como mucho. Con esta versión se obtendrá un producto prototipo a pequeña escala para probar la estabilidad y funcionamiento del sistema. En futuros ciclos de desarrollo se mejorará el producto para soportar una cantidad mucho mayor de usuarios y conexiones.

## 3.4. Atributos del sistema

### 3.5.1. Seguridad

Cada usuario al registrarse deberá proporcionar un nombre de usuario único entre los ya registrados (login), una contraseña o password y su correo electrónico. Además deberá escoger una pregunta de una lista predefinida e insertar su respuesta a la misma. En caso de olvidar su contraseña, siempre podrá optar a cambiarla contestando correctamente a la pregunta escogida en el proceso de registro.

Se hará que aquellos datos que requieran un mayor grado de confidencialidad sean almacenados en la base de datos con su correspondiente encriptamiento (como puede ser el caso de las contraseñas) para evitar la sustracción de los mismos por algún ataque al sistema.

El sistema de login e identificación de usuarios permitirá también restringir el acceso de los mismos a ciertos datos. Por ejemplo, se impedirá que un usuario pueda modificar los datos personales de otro y se negarán capacidades administrativas, como modificar datos del sistema, a los clientes del mismo.

Se incluirá también una capa de seguridad a nivel de implementación para evitar el mayor número de ataques posible. Esto es, por ejemplo, revisar y evitar inyección SQL en todo momento, evitar guardar datos en cookies sin su correspondiente encriptación, etc.

### 3.5.2. Portabilidad

El sistema será inmediatamente portable entre distintos ordenadores ya que bastará con tener un navegador más o menos actualizado para poder hacer uso de él. Con respecto a otras plataformas, la que mayor viabilidad ofrece de cara a la portabilidad son los sistemas móviles ya que para ellos se puede realizar una adaptación del producto y permitir una correcta integración con los mismos.

### 3.5.3. Mantenibilidad

El sistema será altamente mantenible, es decir, la duración y el esfuerzo requeridos por las funciones de mantenimiento (de la base de datos y del servidor) tendrán un coste pequeño.

El mantenimiento del servidor se hará mediante una herramienta automatizada que nos informará del estado del mismo, o de forma manual. Para la parte software nos encargaremos de que se realicen todas las actualizaciones necesarias, de los análisis periódicos del antivirus y de la buena configuración del cortafuegos. Únicamente se dejarán activos los programas y servicios necesarios. Aseguraremos también la limpieza y el aislamiento del servidor.

Por otro lado, en cuanto al mantenimiento de la base de datos usaremos una herramienta que proporcione las funciones necesarias (ej: SQL Server Management Studio) para el funcionamiento óptimo de la misma. Para ellos, ejecutaremos las operaciones de mantenimiento (comprobación de la coherencia de la base de datos, actualización de estadísticas) periódicamente, y siempre nos aseguraremos de tener una copia de seguridad actualizada para poder restaurarla en caso de fallo inesperado.

En lo referente al producto, se tratará de hacer un software que permita sencillas modificaciones y ampliaciones para su posible inclusión en un futuro.

### 3.5.4. Fiabilidad

Se le garantiza al usuario un juego intuitivo, con una dinámica fácil de comprender y exento de errores excepto los causados por su conexión a internet. Para ello se contará con un completo plan de pruebas en cada una de las fases de desarrollo del producto.

# 4. Apéndices

## 4.1. Modelo RUP

El modelo RUP (Proceso Unificado de Desarrollo) es un proceso de desarrollo software que, junto con el Lenguaje Unificado de Modelado (UML) conforma la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

Dado que el RUP está dirigido por casos de uso, podemos resaltar la importancia de ciertas secciones de este documento como pueden ser los requisitos de funcionalidad. A partir de ellos se podrá extraer con gran facilidad la estructura de funcionalidades del sistema y asignar los casos de uso correspondientes a cada apartado.

Por otro lado, mencionar también que a través de la información de esta especificación presentada se podrá obtener un modelo de requisitos bastante sólido. Hacer uso del mismo, y de los casos de uso derivados, permitirá crear una estructura arquitectónica básica del sistema. Dicha base será ampliada y refinada mediante diversas iteraciones del producto hasta obtener la versión final.

Según el Proceso Unificado de Desarrollo, la vida de un sistema se define como una serie de ciclos, cada uno de los cuales consta de 4 fases (descritas a continuación). Dada la limitación de tiempo actual, se llevará a cabo tan sólo un ciclo de desarrollo durante el curso 2012-2013 y, una vez finalizado éste, se procederá a nuevas versiones y ciclos del producto.

El mencionado ciclo, al igual que el resto, constará de varias etapas bien delimitadas:

* Fase de inicio: en ella se especificará una visión general y suficiente del proyecto para garantizar su viabilidad y posterior fase de elaboración. Durante este periodo se discutirán a fondo los detalles de funcionalidad más básicos del producto. En general, se hará un brainstorming de ideas durante las reuniones iniciales del equipo para tomar las más apropiadas para este ciclo del sistema y se descartarán (momentáneamente) aquellas que no puedan llevarse a cabo por limitaciones de tiempo/recursos.
* Fase de elaboración: una vez finalizada la etapa de inicio, y con unas ideas del producto más o menos definidas, se pasará a esta nueva fase. En ella se comenzará a diseñar la arquitectura básica del sistema y sus requisitos en función de las decisiones tomadas con anterioridad. Asimismo, se repartirán los recursos y actividades disponibles para permitir el comienzo del desarrollo del producto.
* Fase de construcción: una vez el producto y arquitectura del mismo hayan sido definidos en su forma más básica, se procederá a su implementación. Para ello se llevarán a cabo diversas iteraciones basándose en las funcionalidades (casos de uso) que irán acumulando módulos sobre el sistema base hasta completarlo. Esta fase finalizará una vez el producto esté listo para ser distribuido, en su versión inicial, a la comunidad de usuarios.
* Fase de transición: cuando la construcción del sistema haya sido finalizada, el software se pondrá en manos de la comunidad de usuarios o, en nuestro caso, se iniciará un nuevo ciclo para obtener una versión mejorada y más estable del producto. Dicho ciclo comenzará al margen del presente curso de Ingeniería del Software.