



# La Conquista del Mundo

## Plan de gestión de proyecto software

**v02**

### ***Equipo de desarrollo***

Ángel Durán Izquierdo  
Antonio Gómez Poblete  
Antonio Martín Menor de Santos  
Daniel León Romero  
Jorge Colao Adán  
Laura Núñez Villa  
Ricardo Ruedas García

2 de noviembre de 2010

## Historial de cambios

<b>v01</b>	26-10-2010	Versión inicial del plan de gestión de proyecto software.
<b>v02</b>	02-11-2010	Añadido el plan de gestión de riesgos.

## Prefacio

Este documento pretende especificar el plan de proyecto para desarrollar la primera práctica de la asignatura de Ingeniería del Software II, de nombre *La Conquista del Mundo*, durante el curso 2010/11 en la Escuela Superior de Informática de la Universidad de Castilla – La Mancha.

Este plan está dirigido a los profesores de la asignatura, Ismael Caballero Muñoz-Reja y Macario Polo Usaola, así como al resto de integrantes del grupo de prácticas, Ángel Durán Izquierdo, Antonio Gómez Poblete, Antonio Martín Menor de Santos, Daniel León Romero, Jorge Colao Adán y Laura Núñez Villa.

Dada durante las clases de la asignatura la sugerencia de elegir un nombre para el grupo de prácticas, los integrantes de este grupo han decidido tomar el nombre de *Umbrella Corporation*, empresa ficticia que aparece en un conocido videojuego. Este grupo tomará ese nombre con su correspondiente logo como sus señas de identidad, y éstos aparecerán en toda la documentación que sea necesario generar.

## Índice general

<b>Historial de cambios</b>	<b>2</b>
<b>Prefacio</b>	<b>3</b>
<b>Índice general</b>	<b>5</b>
<b>1. Visión general</b>	<b>6</b>
1.1. Resumen del proyecto . . . . .	6
1.1.1. Propósito, alcance y objetivos . . . . .	6
1.1.2. Suposiciones y restricciones . . . . .	6
1.1.3. Entregables del proyecto . . . . .	7
1.1.4. Resumen de la planificación y el presupuesto . . . . .	7
1.2. Evolución del plan . . . . .	8
<b>2. Referencias</b>	<b>9</b>
<b>3. Definiciones</b>	<b>10</b>
<b>4. Organización del proyecto</b>	<b>11</b>
4.1. Interfaces externas . . . . .	11
4.2. Estructura interna . . . . .	11
4.3. Roles y responsabilidades . . . . .	11
<b>5. Planes de procesos de gestión</b>	<b>13</b>
5.1. Plan de inicio del proyecto . . . . .	13
5.1.1. Plan de estimación . . . . .	13
5.1.2. Plan de personal . . . . .	13

5.1.3.	Plan de adquisición de recursos . . . . .	13
5.1.4.	Plan de formación de personal . . . . .	14
5.2.	Plan de trabajo . . . . .	14
5.2.1.	Actividades . . . . .	14
5.2.2.	Planificación . . . . .	15
5.2.3.	Asignación de recursos . . . . .	17
5.3.	Plan de gestión de riesgos . . . . .	17
5.3.1.	Riesgos técnicos . . . . .	17
5.3.2.	Riesgos de recursos . . . . .	18
5.3.3.	Riesgos externos . . . . .	18
<b>6.</b>	<b>Planes de procesos técnicos</b>	<b>19</b>
6.1.	Modelo de procesos . . . . .	19
6.2.	Métodos, herramientas y técnicas . . . . .	19
6.3.	Plan de infraestructura . . . . .	20
6.4.	Plan de aceptación del producto . . . . .	20
<b>7.</b>	<b>Planes de procesos de soporte</b>	<b>21</b>
7.1.	Plan de gestión de la configuración . . . . .	21
7.2.	Plan de verificación y validación . . . . .	21
7.3.	Plan de documentación . . . . .	22
7.4.	Plan de garantías de calidad . . . . .	22
7.5.	Plan de revisiones y auditorías . . . . .	23
7.6.	Plan de resolución de problemas . . . . .	23

## 1.1. Resumen del proyecto

### 1.1.1. Propósito, alcance y objetivos

*La Conquista del Mundo* es un juego de estrategia en tiempo real que es jugado sobre un tablero con regiones que representan el mundo. Este juego sigue una arquitectura cliente-servidor y hace uso de un motor de comunicaciones de red para conseguirlo.

El propósito de este proyecto es pues realizar todo el ciclo de desarrollo hasta conseguir implementar un cliente para ordenador de *La Conquista del Mundo*.

Se consideran dentro del alcance del proyecto no sólo las actividades de desarrollo del producto, sino también las relacionadas con conseguir un entorno de desarrollo y ejecución adecuados.

El objetivo principal del proyecto será la realización de la práctica de manera satisfactoria antes de la fecha de entrega. Además habrá otros objetivos secundarios, en su mayor parte formativos, como el conocimiento y puesta en práctica del Proceso Unificado de Desarrollo o el uso de repositorios software en proyectos con varios integrantes.

### 1.1.2. Suposiciones y restricciones

Para el desarrollo de este proyecto no será necesario ningún tipo de hardware especial; los ordenadores personales de los propios desarrolladores serán suficiente.

Las herramientas software necesarias estarán a disposición de los desarrolladores a coste cero, bien porque sean herramientas libres o porque la universidad proporcione las licencias correspondientes.

Se entiende que los desarrolladores son estudiantes, y que por tanto el tiempo que podrán dedicar al proyecto es limitado. Este hecho debe estar

reflejado en la carga semanal de trabajo.

Por el carácter formativo del proyecto, cada desarrollador deberá adoptar al menos una vez durante el desarrollo del proyecto cada uno de los siguientes roles: analista, diseñador, programador y probador.

Este proyecto depende del proyecto encargado del servidor de la aplicación. Las comunicaciones con el servidor se realizarán mediante la tecnología RMI y la interfaz de comunicaciones será definida por el equipo del servidor.

Se deberá entregar a los destinatarios de este proyecto una máquina virtual para VirtualBox que contendrá todo el desarrollo del proyecto así como un entorno donde ejecutar la aplicación. El sistema operativo de esa máquina virtual queda a elección del grupo.

El número de componentes del equipo de desarrollo estaba limitado a un máximo de ocho personas, quedando el equipo finalmente formado por siete personas.

El tiempo para desarrollar la aplicación vence el día 31 de enero de 2011.

### **1.1.3. Entregables del proyecto**

El producto final del desarrollo será una máquina virtual para VirtualBox. Será entregado en formato DVD-R y contendrá los siguientes elementos:

- La aplicación final y todas las dependencias necesarias para su correcta ejecución.
- Manual de usuario.
- El proyecto de desarrollo con el código fuente completo de la aplicación.
- La documentación del proyecto compuesta por:
  - Plan de gestión de proyecto software
  - Especificación de requisitos software
  - Diagramas de análisis y diseño de clases
  - Plan de pruebas software

### **1.1.4. Resumen de la planificación y el presupuesto**

La fecha final del proyecto está establecida el 31 de enero de 2011. Adicionalmente, habrá entregas parciales para evaluar el avance del proyecto en fechas aún por determinar.



Debido al carácter académico del proyecto, éste carece de presupuesto. Los integrantes del grupo no trabajarán de manera remunerada sino para la obtención de créditos en la asignatura de Ingeniería del Software II.

### 1.2. Evolución del plan

Este plan podrá sufrir actualizaciones a lo largo del desarrollo del proyecto. Estas actualizaciones se realizarán de manera no planificada según vayan surgiendo y los integrantes del proyecto serán notificados de la nueva versión a través de la lista de correo asociada al proyecto.

En un repositorio GIT se podrá encontrar la última versión así como el resto de versiones anteriores.



## Referencias

El documento que ha dado inicio al proyecto se puede consultar en la siguiente dirección:

<https://campusvirtual.uclm.es/file.php/11303/PRG1P/PrIS02-1Pv1.1.pdf>

Para la elaboración de este documento se ha seguido ampliamente el siguiente estándar:

IEEE Std 1058-1998, *Standard for Software Project Management Plans*. The Institute of Electrical and Electronics Engineers, Inc. 1998.

Además, se han tomado como ejemplo a seguir los siguientes documentos:

[http://www.buckley-golder.com/papers/mbg\\_SPMP\\_ProjectManagement.pdf](http://www.buckley-golder.com/papers/mbg_SPMP_ProjectManagement.pdf)

<http://sourcefrog.net/projects/newuserfs/doc/plan/plan.pdf>

**Definiciones**

**GIT** Sistema de control de versiones distribuido.

**RMI** Remote Method Invocation. Interfaz de comunicaciones de red para el lenguaje de programación Java.

## Organización del proyecto

### 4.1. Interfaces externas

Los profesores de la asignatura harán las partes de clientes y serán los que, al final del desarrollo, evalúen el proyecto. Es pues a ellos a quien se deben dirigir las dudas o problemas que surgan durante el desarrollo.

Paralelamente, otro grupo de la asignatura se encargará del desarrollo de la parte servidor de la aplicación. Los grupos de trabajo encargados de los clientes, entre los que se incluye el nuestro, deberán ponerse de acuerdo con ellos y establecer una interfaz de comunicaciones común para todos.

### 4.2. Estructura interna

El equipo de desarrollo está formado por siete alumnos. Uno de ellos se encargará de las labores de gestión y organización en sus funciones de director de proyecto. El resto del equipo se encargará del desarrollo de la aplicación y trabajará bien de manera individual o en pequeños grupos que variarán de manera periódica.

### 4.3. Roles y responsabilidades

Este proyecto dispone de los siguientes roles y responsabilidades:

- **Director de proyecto:** Está encargado de la gestión, planificación y documentación del proyecto. Además deberá proveer al resto del equipo con las herramientas adecuadas para el desarrollo.
- **Especificador de caso de uso:** Realizará la especificación de requisitos software.
- **Analista - diseñador:** Participará en el análisis de los casos de uso y en el diseño de la aplicación. También estará a cargo del diseño de



los planes de prueba.

- **Programador:** Se encargará de la codificación de la aplicación y las pruebas.
- **Tester:** Llevará a cabo las pruebas del sistema y realizará los informes correspondientes.

## Planes de procesos de gestión

### 5.1. Plan de inicio del proyecto

#### 5.1.1. Plan de estimación

Para la estimación del proyecto, éste se ha estructurado en una serie de tareas y se ha estimado un esfuerzo para cada tarea basado en la experiencia personal del equipo de desarrollo en proyectos de características similares. Ese esfuerzo estará expresado en horas y representará el tiempo necesario por una persona para realizar dicha tarea.

Para la planificación del calendario del proyecto se ha tenido en cuenta un esfuerzo de una hora diaria de media a la semana por cada integrante del grupo, esto es, un total de 5 horas semanales.

#### 5.1.2. Plan de personal

El grupo está integrado por un jefe de proyecto y seis desarrolladores. Éstos forman parte del grupo desde su formación al inicio de la asignatura y permanecerán en él durante todo el curso.

#### 5.1.3. Plan de adquisición de recursos

En el desarrollo del proyecto se hará uso de dos aplicaciones fundamentales: Eclipse y Visual Paradigm. Ambas aplicaciones se pueden obtener libremente de sus respectivos sitios web, y sólo en el caso de Visual Paradigm hará falta una licencia, que será proporcionada por los profesores de la asignatura.

Se creará una lista de correo en Google Groups para las comunicaciones internas del grupo. Todos los integrantes del grupo, así como los profesores de la asignatura, recibirán una invitación a la lista tan pronto como sea creada.

El trabajo de desarrollo se llevará a cabo en dos repositorios Git, uno dedicado a la documentación del proyecto y otro al proyecto en sí. Ambos estarán hospedados en GitHub, donde deberán registrarse todos los integrantes del grupo.

#### 5.1.4. Plan de formación de personal

Los integrantes del grupo recibirán por parte de los profesores de la asignatura a lo largo del curso los conocimientos y la ayuda necesaria para llevar a cabo este proyecto.

### 5.2. Plan de trabajo

#### 5.2.1. Actividades

En la siguiente estructura de descomposición de trabajo se muestran las tareas que componen el proyecto así como el esfuerzo estimado para su realización.

Tabla 5.1: Estructura de descomposición de trabajo

Tarea	Esfuerzo	Descripción
1		<b>Inicio del proyecto</b>
1.1	30 h	Elaboración del plan de gestión de proyecto software.
1.2	30 h	Elaboración de la especificación de requisitos software.
	60 h	<b>Total</b>
2		<b>Análisis y diseño</b>
2.1	2 h	Caso de uso: Registrarse.
2.2	3 h	Caso de uso: Iniciar sesión.
2.3	2 h	Caso de uso: Cerrar sesión.
2.4	2 h	Caso de uso: Crear una nueva partida.
2.5	2 h	Caso de uso: Unirse a una partida.
2.6	4 h	Caso de uso: Conectarse a una partida.
2.7	2 h	Caso de uso: Desconectarse de una partida.
2.8	3 h	Caso de uso: Ver la lista de partidas.
2.9	2 h	Caso de uso: Recibir notificación de un nuevo jugador.
2.10	5 h	Caso de uso: Realizar un movimiento.
2.11	4 h	Caso de uso: Responder a un movimiento.
2.12	4 h	Caso de uso: Comprar refuerzos.
2.13	2 h	Caso de uso: Enviar petición de alianza.
2.14	2 h	Caso de uso: Recibir petición de alianza.
2.15	2 h	Caso de uso: Romper alianza.
2.16	2 h	Caso de uso: Enviar actualización de la partida.
2.17	2 h	Caso de uso: Recibir actualización de la partida.
	45 h	<b>Total</b>
3		<b>Implementación</b>
3.1	5 h	Caso de uso: Registrarse.
3.2	5 h	Caso de uso: Iniciar sesión.

Tabla 5.1: Estructura de descomposición de trabajo

Tarea	Esfuerzo	Descripción
3.3	2 h	Caso de uso: Cerrar sesión.
3.4	8 h	Caso de uso: Crear una nueva partida.
3.5	4 h	Caso de uso: Unirse a una partida.
3.6	25 h	Caso de uso: Conectarse a una partida.
3.7	2 h	Caso de uso: Desconectarse de una partida.
3.8	10 h	Caso de uso: Ver la lista de partidas.
3.9	2 h	Caso de uso: Recibir notificación de un nuevo jugador.
3.10	25 h	Caso de uso: Realizar un movimiento.
3.11	10 h	Caso de uso: Responder a un movimiento.
3.12	6 h	Caso de uso: Comprar refuerzos.
3.13	4 h	Caso de uso: Enviar petición de alianza.
3.14	4 h	Caso de uso: Recibir petición de alianza.
3.15	4 h	Caso de uso: Romper alianza.
3.16	4 h	Caso de uso: Enviar actualización de la partida.
3.17	4 h	Caso de uso: Recibir actualización de la partida.
	124 h	<b>Total</b>
4		<b>Pruebas</b>
4.1	10 h	Implementación del servidor de prueba.
4.2	20 h	Pruebas unitarias.
4.3	20 h	Pruebas de integración.
4.4	20 h	Pruebas de sistema.
	70 h	<b>Total</b>
	299 h	<b>Total global</b>

### 5.2.2. Planificación

A continuación se muestra el calendario que se seguirá para el desarrollo del proyecto. Cada iteración se corresponde con una semana de acuerdo a la siguiente lista de fechas:

- **Fase de inicio**

- Iteración 1: del 4 al 8 de octubre.

- Iteración 2: del 11 al 15 de octubre.

- Iteración 3: del 18 al 22 de octubre.

- **Fase de elaboración**

- Iteración 4: del 25 al 29 de octubre.

- Iteración 5: del 1 al 5 de noviembre.

- Iteración 6: del 8 al 12 de noviembre.

- **Fase de construcción**

- Iteración 7: del 15 al 19 de noviembre.

- Iteración 8: del 22 al 26 de noviembre.

Iteración 9: del 29 de noviembre al 3 de diciembre.

■ **Fase de transición**

Iteración 10: del 6 al 10 de diciembre.

Iteración 11: del 13 al 17 de diciembre.

Tabla 5.2: Planificación del proyecto

Tarea	It01	It02	It03	It04	It05	It06	It07	It08	It09	It10	It11
1.1											
1.2											
2.1											
2.2											
2.3											
2.4											
2.5											
2.6											
2.7											
2.8											
2.9											
2.10											
2.11											
2.12											
2.13											
2.14											
2.15											
2.16											
2.17											
3.1											
3.2											
3.3											
3.4											
3.5											
3.6											
3.7											
3.8											
3.9											
3.10											
3.11											
3.12											
3.13											
3.14											
3.15											
3.16											
3.17											
4.1											
4.2											
4.3											
4.4											





### 5.2.3. Asignación de recursos

Los desarrolladores serán asignados un rol en cada iteración para realizar las tareas que correspondan. Los distintos roles que podrán tomar son los siguientes:

- Recurso 1: Especificador de casos de uso.
- Recurso 2: Analista-diseñador.
- Recurso 3: Programador.
- Recurso 4: Tester.

A continuación se muestra la cantidad de recursos asignados en cada iteración.

Tabla 5.3: Asignación de recursos

	Recurso 1	Recurso 2	Recurso 3	Recurso 4
Iteración 1				
Iteración 2	4			
Iteración 3	4	2		
Iteración 4		3	2	
Iteración 5		3	3	
Iteración 6		2	3	1
Iteración 7			4	2
Iteración 8			4	2
Iteración 9			4	2
Iteración 10			2	4
Iteración 11				4

## 5.3. Plan de gestión de riesgos

El desarrollo del proyecto puede verse afectado por diversos riesgos.

### 5.3.1. Riesgos técnicos

Algunos de los desarrolladores pueden no haber usado hasta ahora ningún sistema de control de versiones. Además se hará uso de Git, que es un sistema bastante reciente y con más funcionalidades que sistemas anteriores como Subversion, lo que implica también mayor complejidad inicial. Se deberá proveer a los desarrolladores con la suficiente documentación para un correcto uso del sistema de control de versiones.

El equipo de desarrollo se enfrenta al diseño de una interfaz gráfica de



usuario. En esta proyecto se pretende aprender buenas prácticas de diseño pero es difícil seguir estas prácticas sin experiencia previa suficiente. Por eso es importante la creación de múltiples prototipos y ejemplos mínimos de código que realimenten a los analistas y diseñadores.

### 5.3.2. Riesgos de recursos

Es inevitable tener en cuenta que los desarrolladores no pueden dedicarse al proyecto a tiempo completo debido a que cursen otras asignaturas. Ésto provocará que en determinadas semanas se reduzca el tiempo disponible para el proyecto. Se ha realizado una planificación optimista del proyecto con una fecha de finalización hasta un mes anterior a la fecha de entrega final, de manera que se pequeños retrasos en las tareas no impliquen un impacto importante.

### 5.3.3. Riesgos externos

El proyecto no se realiza en solitario sino que interacciona con otros grupos de trabajo. Esta interacción debe formalizarse con un conjunto de funciones bien definidas entre los clientes y el servidor. Un retraso en la redacción de estas funciones puede provocar cambios internos en algunos componentes de la aplicación.

## Planes de procesos técnicos

### 6.1. Modelo de procesos

El proyecto sigue el proceso unificado de desarrollo, en el que se divide el proyecto en varias iteraciones en las que se refinan distintos elementos del proyecto.

El primer documento a desarrollar será la especificación de requisitos software, donde se plasmarán las funcionalidades que la aplicación debe cubrir. Este documento incluirá el modelo de casos de uso, que guiará el desarrollo de la aplicación en las sucesivas iteraciones.

Seguidamente los analistas-diseñadores desarrollarán cada caso de uso, describiendo su comportamiento mediante diagramas de secuencia y completando con cada nuevo caso de uso el diagrama de clases de diseño.

Los diagramas de análisis y diseño servirán a los programadores para realizar la implementación de la aplicación y a los testers para documentar los planes de pruebas.

### 6.2. Métodos, herramientas y técnicas

Para la redacción de la documentación del proyecto se usará el lenguaje de maquetación de documentos  $\text{\LaTeX}$ , y se recomendará (pero no obligará) el uso del programa Kile para su edición.

El modelado de la aplicación se realizará con la ayuda de la aplicación Visual Paradigm.

La aplicación estará escrita en el lenguaje de programación Java. Deberá tener interfaz gráfica de usuario, para lo que se usará la librería gráfica Swing. Además deberá disponer de una capa de comunicaciones con el servidor, para lo que se utilizará la tecnología RMI. Tanto Swing como RMI vienen incluidas en la librería estándar de Java. El desarrollo de la aplicación se realizará usando el IDE Eclipse.



Por último, todos los documentos serán gestionados por el sistema de control de versiones Git.

### 6.3. Plan de infraestructura

El desarrollo de la práctica se realizará usando los ordenadores personales de los propios desarrolladores ya que no existen requisitos de alto rendimiento o disponibilidad.

Estos equipos podrán disponer del sistema operativo que su usuario estime oportuno, siempre que sean capaces de ejecutar sin problemas las herramientas descritas en la sección anterior.

### 6.4. Plan de aceptación del producto

En las últimas semanas del desarrollo, el esfuerzo se centrará en la realización de pruebas de sistema y aceptación para asegurarse de que la aplicación realizada cumple con los requisitos descritos en el enunciado del problema, y matizados en las diferentes reuniones con los profesores de la asignatura.

## Planes de procesos de soporte

### 7.1. Plan de gestión de la configuración

La gestión de la configuración del proyecto se realizará usando el sistema de control de versiones Git, y se usarán los servicios gratuitos de GitHub para hospedar los repositorios, donde cada integrante del equipo tendrá una cuenta. La relación de cuentas se puede ver en la siguiente tabla.

Tabla 7.1: Relación de cuentas en GitHub

Desarrollador	Usuario
Ricardo Ruedas García (Jefe de proyecto)	ricki
Ángel Durán Izquierdo	Aduran
Antonio Gómez Poblete	pobleteag
Antonio Martín Menor de Santos	deejaytoni
Daniel León Romero	DaniLR
Jorge Colao Adán	JorgeCA
Laura Núñez Villa	LauraN

Se utilizará un flujo de trabajo distribuido, en el que cada integrante del grupo trabajará sobre su repositorio personal. La labor del jefe de proyecto será la de integrar todos los flujos de trabajo en su repositorio, que será siempre el repositorio de referencia para el resto del grupo.

### 7.2. Plan de verificación y validación

Para las pruebas unitarias, se utilizará la suite de pruebas JUnit disponible para Java en combinación con la herramienta EcEmma para Eclipse.

Para las pruebas exploratorias será necesario implementar un servidor sin funcionalidad para poder detectar cualquier problema de la aplicación en un entorno de ejecución lo más parecido posible al entorno final.



### 7.3. Plan de documentación

La documentación del proyecto estará siempre localizable en el sistema de control de versiones. Se puede visitar en formato web en la siguiente dirección:

<http://github.com/ricki/iso2010-11-doc>

El proyecto realizado con Visual Paradigm se encuentra en el directorio `uml`. El resto de documentación se encuentra en el directorio `latex`. Adicionalmente se añadirá una versión compilada en PDF de los documentos listos para distribución en el directorio `distrib`.

El código para identificar los documentos empezará por las letras UC, siglas de Umbrella Corporation. Le seguirán un número de dos cifras indicando el tipo de documento, un guión, y un número de tres cifras para identificar un documento de una categoría. Después se añadirá un punto y la versión del documento.

Los códigos de categoría usados son los siguientes:

- 01 - Documentación de la fase de inicio.
- 02 - Actas de reuniones.
- 03 - Planes de pruebas.

Por ejemplo, el código del presente documento será el siguiente:

UC 01-001.01

Es decir, la primera versión del primer documento de la fase de inicio.

### 7.4. Plan de garantías de calidad

Según se aproximen las últimas iteraciones del desarrollo, los recursos destinados a probar el sistema aumentarán con respecto al resto de flujos de trabajo. Serán las pruebas exploratorias de la aplicación ya en su entorno de ejecución las que determinen si la aplicación cumple con los requisitos del enunciado.

Se espera que para cuando empiecen las pruebas exploratorias, esté ya a disposición de todos los equipos de trabajo un servidor activo a todas horas sobre el que probar la aplicación.

## 7.5. Plan de revisiones y auditorías

Antes de la entrega final, se realizarán varias reuniones de grupo con los profesores de la asignatura que revisarán el progreso del proyecto e informarán al grupo sobre los aspectos a corregir.

A día de hoy hay planificadas las siguientes reuniones:

- 3 de noviembre
- 10 de noviembre
- 22 de diciembre

## 7.6. Plan de resolución de problemas

Debido al caracter cambiante de los roles que toman los integrantes, no habrá un recurso dedicado a la resolución de problemas. Para cada problema que surga, será la persona más capaz en el área de experiencia del problema la encargada de resolverlo.