



**Universidad Complutense**

Facultad de Informática



**eGorilla**

*Plan de Gestión de la Configuración*

*Asignatura: Ingeniería del Software*

*Curso: 2008/2009*

# Índice

1.	Introducción.....	3
1.1.	Propósito del plan.....	3
1.2.	Alcance del plan.....	3
1.3.	Definiciones y acrónimos .....	4
1.4.	Referencias .....	5
2.	Especificaciones de gestión.....	5
2.1.	Organización .....	5
2.2.	Responsabilidades .....	5
2.3.	Plan de implementación .....	5
3.	Políticas, directivas, procedimientos.....	6
3.1.	Niveles del software en un árbol jerárquico .....	6
3.2.	Nombrado de programas y módulos.....	6
3.3.	Identificación de productos software.....	6
3.4.	Identificación de documentación.....	6
3.5.	Identificación de medios y ficheros .....	6
4.	Actividades de GCS .....	7
4.1.	Identificación de la configuración.....	7
4.2.	Enumeración de las líneas base .....	7
4.3.	Bibliotecas y repositorios a utilizar .....	8
4.4.	Mecanismos para el control de versiones .....	8
	ANEXO A. Tablas ECS .....	10



# 1. Introducción

## 1.1. Propósito del plan

El objetivo de este documento es establecer las actividades de Gestión de la Configuración del Software que se van a realizar en el marco del proyecto eGorilla, cómo se van a realizar, quién las va a realizar, cuando y con qué recursos. Todo ello con el fin último de poder controlar en cada momento, a lo largo de todo el ciclo de vida, el estado del sistema que se está desarrollando y la evolución seguida.

Los destinatarios de este documento son (en principio) los miembros del equipo de desarrollo de la aplicación, ya que son ellos los interesados en determinar cómo se van a coordinar todos los elementos.

Se van a definir cuales serán los *elementos de configuración del software*, todas las piezas constitutivas de los productos finales que se van a entregar y cuales van a ser las *herramientas de desarrollo*. Ejemplos de esto último pueden ser los entornos de programación, sistemas de desarrollo de documentación y sistemas de control de versiones de dichos *elementos de configuración del software*.

Además se determinará que tipo de modelo de desarrollo se seguirá durante la implementación, y qué objetivos, revisiones, modificaciones, etc... han de hacerse para cumplir con lo esperado en cada una de las diferentes versiones del software y documentación. Esto hasta la entrega final, y después puede extenderse si hubiera un mantenimiento o extensión de la aplicación.

## 1.2. Alcance del plan

El ámbito de este Plan de Gestión de la Configuración abarca un único proyecto: el desarrollo de una aplicación p2p tipo eMule, a la que se ha bautizado como eGorilla, a realizar como práctica de la asignatura de Ingeniería del Software. Para unos detalles básicos acerca de su funcionalidad y características generales, se puede consultar el documento de *Especificación de Requisitos*, que también forma parte del desarrollo de este proyecto.

Nos basaremos para el desarrollo del producto en un *Modelo de Proceso en Espiral*, que en sucesivas iteraciones, nos permitirá ir pasando por todas las fases. Esto nos permite avanzar en paralelo en todos los frentes, hasta obtener el grado de completitud deseado en la entrega final. De cara a la gestión de la configuración, se identificarán inicialmente un conjunto de elementos de configuración del software, que se irán ampliando con otros nuevos a medida que se van completando iteraciones.

A continuación se presenta una lista con los *elementos de configuración del software* que se han identificado para esta primera iteración:

- Documento de *Especificación de Requisitos*.
- Documento de *Gestión de Riesgos*.
- Documento de *Casos de Uso*.
- Documento de *Tabla de Elementos de Configuración del Software*.
- Documento de *Plan de Gestión de Configuración* (este mismo documento también constituye un elemento de configuración del software).
- Documentos de *Planificación*.
- Paquete de software del *programa cliente*.
- Paquete de software del *programa servidor*.
- Paquete de software de las *interfaces del programa cliente*.





En principio no se va a incluir dentro de la gestión de configuración ningún otro software (de apoyo, etc) diferente del que se va a desarrollar.

Se va a aplicar un grado de formalidad medio sobre las actividades de gestión de la configuración del software, ya que no se trata de un proyecto excesivamente grande o complejo y se enmarca dentro de la realización de la práctica de una asignatura. Estas actividades cubrirán el ciclo de vida del producto desde la concepción hasta su puesta en producción (entrega final). Después podría extenderse si hubiera un mantenimiento o extensión de la aplicación.

Las limitaciones a tener en cuenta en este proyecto son de índole temporal, de hardware y de personal:

La primera viene determinada por tener como tiempo máximo el correspondiente a un curso académico (nueve meses). Los objetivos de desarrollo final deben poder ajustarse dentro de ese plazo.

La segunda viene determinada por el software/hardware disponible en los laboratorios de la facultad. Si bien el software será desarrollado en Java, y por tanto es multiplataforma, la aplicación debe poder funcionar y utilizar la red de computadores existente en el laboratorio. Además la disponibilidad de herramientas software es limitada.

La tercera es relativa al número de personas y tiempo de dedicación al proyecto. Se debe tener en cuenta que dicha dedicación no es total y que el equipo está formado por **trece** personas, a la hora de ajustar la planificación temporal.

Para este proyecto, partimos de dos tipos de supuestos :

- **Supuestos de usuario:** Para el manejo de la aplicación, se presuponen unos conocimientos mínimos de informática a nivel de usuario, para el correcto manejo del programa cliente. Éstos son relativos básicamente al manejo intuitivo de interfaces gráficas.

Sin embargo, para el programa servidor, se presuponen conocimientos de informática más avanzados: manejo de consolas de comandos, configuraciones de red, y familiaridad con plataformas basadas no sólo en sistemas Windows, sino también tipo Unix(Linux) y Mac.

El usuario no va a participar en las actividades de gestión de la configuración.

- **Supuestos de desarrollador:** El equipo de desarrollo deberá estar familiarizado (o familiarizarse) con el lenguaje de programación Java, entornos de desarrollo para el mismo (tipo NetBeans), conocimientos de redes a nivel suficiente como para comprender el diseño de la aplicación, etc... De cara a la gestión de la configuración tendrá que familiarizarse con la herramienta de control de versiones SVN.

### 1.3. Definiciones y acrónimos

A continuación se presenta una lista con las abreviaturas que se van a utilizar durante el presente documento y en todas aquellas otras partes en las que sea necesario:

- **ECS:** Elemento de Configuración del Software
- **ERS:** Especificación de Requisitos del Software
- **CU:** Caso (o Casos) de Uso
- **GR:** Gestión de Riesgos
- **GCS:** Gestión de Configuración del Software
- **TECS:** Tabla de Elementos de Configuración del Software (para controlar el estado de la configuración – ver apartado 5, Identificación de la configuración)





- **IDE:** Entorno de desarrollo integrado (en inglés Integrated Development Enviroment)
- **SCV:** Sistema de Control de Versiones
- **IGU:** Interfaz Gráfica de Usuario
- **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos (en inglés Institute of Electrical and Electronics Engineers)
- **SVN:** Subversión, software libre de control de versiones.

## 1.4. Referencias

Los principales documentos que usamos como apoyo para la elaboración del presente documento son :

1. Standard IEEE 828 – rev. 2005.
2. Documentación de Subversion.
3. Transparencias vistas en clase.

# 2. Especificaciones de gestión

## 2.1. Organización

Como ya se ha comentado anteriormente, el desarrollo de este proyecto se enmarca en el contexto de la realización de una práctica para la asignatura de Ingeniería del Software. Solamente los miembros del equipo de desarrollo van a participar en o ser responsables de las actividades de gestión de la configuración. Para esta primera iteración, la estructura de trabajo que se está siguiendo obedece a un esquema horizontal. Los miembros de equipo colaborarán en todas las partes que comprende el desarrollo. Las aportaciones concretas a cada parte pueden consultarse en el *ECS de Planificación*, y en el *ECS de control del estado de la configuración* *Tabla de Elementos de Configuración del Software*.

## 2.2. Responsabilidades

La responsabilidad realizar las actividades de gestión de configuración recaerá solidariamente sobre los miembros del equipo de desarrollo. Al trabajar en un contexto de colaboración completa, no se han establecido responsabilidades jerárquicas ni organizativas. No hay miembros “dueños” de ninguna parte de la estructura de desarrollo, ni de los productos que se van generando. Todos pueden colaborar en todo.

## 2.3. Plan de implementación

Consultar el subapartado de *enumeración de las líneas base* del apartado 5, y el documento de *planificación* (de este último sobre todo el *plan de fase*) para concebir una idea general de cómo se va a ir implementando el proyecto, y en qué momento se establecen los hitos importantes de la elaboración, pruebas y entregas.





### 3. Políticas, directivas, procedimientos

Como consecuencia del desarrollo que se ha seguido en la realización del proyecto, este apartado no ha sido aplicable en su totalidad y se tendrá en cuenta en posteriores iteraciones.

#### 3.1. *Niveles del software en un árbol jerárquico*

La estructura y los componentes software que van a conformar este proyecto no se ha establecido todavía en el momento de redactar este documento. Este apartado se rellenará más adelante, previsiblemente al final de la fase de elaboración, cuando ya se tenga una visión de la arquitectura general del sistema.

#### 3.2. *Nombrado de programas y módulos*

Se ha determinado que los nombres de programas, módulos, paquetes y clases no contengan acentos y, si se trata de nombres compuestos, utilizar el guión bajo '\_' como separador. Además los paquetes se han de nombrar exclusivamente en minúsculas y las clases deben empezar con la primera letra en mayúscula. Queda pendiente por determinar debido a la falta de consenso entre los miembros del grupo si se va seguir el convenio de utilizar nombres en español o en inglés.

En cuanto al código fuente del proyecto, los convenios anteriores también aplicarán a la nomenclatura de los atributos, métodos, etc. Los atributos siempre comenzarán por el carácter '\_'. Además de esto, todas las funciones/métodos han de incluir una documentación mínima (descripción del método, de sus atributos, etc) necesaria para generar fácilmente ayuda en formato HTML mediante javadoc. Designación de versiones

#### 3.3. *Identificación de productos software*

Hasta el momento se han distinguido dos productos principales: el cliente P2P y servidor P2P.

#### 3.4. *Identificación de documentación*

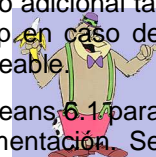
Toda la documentación generada ha de ser subida al repositorio, siguiendo los mismos criterios anteriores de nombrado para programas y módulos, es decir, nombres de los documentos sin acentos y con el carácter '\_' para separar palabras. Se ha decidido así para evitar problemas que pueden acarrear algunos clientes de svn.

Por otra parte, los documentos han de seguir todos un formato homogéneo, en apartados como el encabezado y pie de página, fuente de texto, tamaño, estilo, etc.

#### 3.5. *Identificación de medios y ficheros*

Por el momento el único medio de almacenamiento 'oficial' es el repositorio de código eGorilla, ubicado en la url <http://code.google.com/p/egorilla/> (dentro del servicio googlecode). Pero se está valorando la posibilidad de disponer de un medio de almacenamiento adicional tal como una memoria usb o 1 disco externo, con vistas a disponer de un backup en caso de surgir problemas con googlecode, con la red o algún otro tipo de problema no deseable.

En cuanto a las herramientas de desarrollo, finalmente se ha optado por NetBeans 6.1 para el desarrollo de software y diseño UML, y Microsoft Word 2003 para la documentación. Se





consideró la utilización de Eclipse, pero se ha descartado.

Pero a fecha actual (28-01-09) se está considerando cambiar Microsoft Word por OpenOffice Writer y demás programas de esa suite, pues ofrecen mayor compatibilidad. Cualquier decisión final que está por llegar, será consignada entre otros sitios en este documento.

## 4. Actividades de GCS

### 4.1. Identificación de la configuración

Una lista detallada de los *ECS* identificados hasta el momento puede verse en el subapartado *Alcance del proyecto* del apartado *Introducción*. Solamente van a ponerse bajo control de la configuración productos obtenidos del proceso de desarrollo (no otros elementos del entorno de desarrollo). De momento no es necesario descomponer dichos *ECS* en otros más sencillos. Complicaría la gestión de la configuración, ya que a estas alturas serían demasiados objetos a tener en cuenta a la vez para la magnitud de la entrega.

Para poder controlar de manera eficiente y rápida el *estado de la configuración*, vamos a utilizar una *Tabla de ECS (TECS* – ver acrónimos en la introducción), que será un documento aparte, donde guardaremos información de cada elemento, obedeciendo al siguiente esquema:

#### Esquema de identificación de los ECS :

- Código ECS
- Número de versión
- Fecha de creación
- Autor/es
- Número de copias
- Localización

De momento, las relaciones que se establecen entre los *ECS*, son *relaciones de sucesión*, es decir, tenemos ya nuevas versiones de *ECS* que han sido revisados. Esto viene reflejado explícitamente en el número de versión de cada *ECS* (ver *TECS*, Anexo A).

### 4.2. Enumeración de las líneas base

Con cada entrega parcial que se realice se generará una nueva línea base, que estará compuesta por los *ECS* especificados por el profesor. Se ha decidido incluir una línea base, en principio en torno al mes de Mayo, para replantear las cosas que sean necesarias con objetivo de alcanzar las metas del proyecto.

Se ha establecido y cumplido ya una primera *línea base*, con fecha 12 de Noviembre de 2008, constituida por los siguientes *ECS* :

- Documento de Especificación de Requisitos.
- Documento de Casos de Uso.
- Documento de Gestión de Riesgos.

Esta entrega no ha sido una iteración completa, ya que sólo ha consistido en una primera versión de los documentos necesarios para poder abordar el inicio del desarrollo del software.

La siguiente *línea base* está fijada con fecha 18 de Diciembre de 2008. Se van a incluir los siguientes *ECS* :







- Documento de Especificación de Requisitos (primera revisión).
- Documento de Casos de Uso (primera revisión).
- Documento de Gestión de Riesgos (primera revisión).
- Plan de Gestión de Configuración del Software (presente documento, que supervisa la coordinación del proyecto hasta la primera iteración completa).
- Tabla de Elementos de Configuración del Software.
- Documento/s de planificación/es.
- Programa cliente con funcionalidad reducida.
- Programa servidor con funcionalidad reducida.
- Transparencias de presentación.

Esta entrega constituye ya la primera iteración del *Modelo de Proceso en Espiral* que se está siguiendo.

La siguiente entrega está establecida para el día 30 de Enero de 2009. En ella se persigue definir la *arquitectura del sistema*, a través de las *vistas estáticas de descomposición en módulos*, las *vistas dinámicas de componentes y conectores*, y las *vistas de despliegue y asignación de trabajos*. Así mismo, quedarán también definidos los requisitos no funcionales del sistema, (atributos de calidad del software), a través de los escenarios de calidad.

Por tanto, los ECS a incluir en esta línea base, son los siguientes :

- Documento de Especificación de Requisitos (segunda revisión)
- Documento de Casos de Uso (primera revisión).
- Plan de Gestión de Configuración del Software (presente documento, que supervisa la coordinación del proyecto hasta la segunda iteración completa – primera revisión).
- Documento/s de planificación/es (primera revisión)
- Programa cliente con funcionalidad reducida.
- Programa servidor con funcionalidad reducida.
- Escenarios de Calidad del Software
- Vista de Asignación de Trabajo
- Vista de Despliegue
- Vista de Implementación
- Vista de Protocolo Cliente Servidor
- Vista de Interfaz Gráfica del Cliente
- Vista de Interfaz Gráfica del Módulo Estadísticas
- Vista de Comunicación de Estadísticas con resto de módulos

#### 4.3. *Bibliotecas y repositorios a utilizar*

Para el control automático de las versiones de los ECS a lo largo de todo el proyecto, vamos a utilizar un repositorio de *Google Code* basado en el sistema de control de versiones *Subversion*. La url del repositorio es <http://code.google.com/p/egorilla/>, accesible en modo sólo lectura para cualquiera que desee consultar los ECS. Los miembros del equipo de desarrollo tienen acceso completo a dicho repositorio.



Para gestionar este sistema de versiones, se usará tanto el cliente de *Subversion* que tiene integrado el *IDE NetBeans*, como los clientes *svn* (línea de comandos) y *RapidSvn* (interfaz





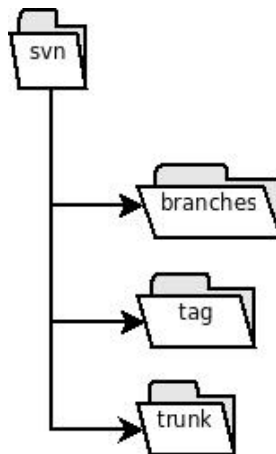
gráfica) de Linux, *Tortoise* para Windows, y sus equivalentes para Mac. La recuperación, modificación y almacenamiento de los ECS se realizará mediante los comandos ofrecidos por el cliente de *Subversion*.

Así contaremos con las siguientes bibliotecas de desarrollo:

- **Biblioteca de trabajo:** Constituida por la copia local del proyecto, que tendrá cada miembro del grupo.
- **Biblioteca de integración:** Constituida por la parte del repositorio destinada a la implementación de la aplicación (directorios *branches* y *trunk* – ver siguiente subapartado).
- **Biblioteca de proyecto:** Constituida por la parte del repositorio destinada a almacenar las líneas base aprobadas (es decir, los ECS que constituyen los hitos importantes, en nuestro caso las entregas). Es el directorio *tag* (ver siguiente subapartado)

#### 4.4. Mecanismos para el control de versiones

La estructura de directorios del repositorio, es la siguiente:



Dentro de cada directorio (*branches*, *tag*, *trunk*) se encuentra la siguiente disposición:

- **Directorio cliente:** Donde se guarda la implementación del *programa cliente* y sus correspondientes *IGUs*.
- **Directorio doc:** Donde se guarda la documentación asociada al proyecto. Esta directorio además contiene una estructura interna que permite clasificar la documentación adecuadamente :
  - **Directorio Generales :** Donde se guarda toda la documentación que no tenga que ver directamente con la arquitectura del sistema: planificaciones, gestión de la configuración, etc...
  - **Directorio Arquitectura del Sistema :** Actualmente clasificado en dos subdirectorios :
    - **Vistas :** Donde se guardan de momento los diferentes *tipos de vistas de la arquitectura*, y más adelante contendrá también los correspondientes diagramas de clases, etc...
    - **Escenarios de Calidad :** Donde se guardan los documentos con los que identifiquemos los *atributos de calidad del software* (*escenarios de calidad*).
- **Directorio servidor:** Donde se guarda la implementación del *programa servidor*.





Atendiendo al subapartado anterior, se puede identificar dónde se van a guardar los diferentes productos que se van a ir generando durante el desarrollo del proyecto. La diferencia entre el directorio branches y el directorio trunk, es que aunque ambos formen parte de la biblioteca de integración del proyecto, el primero sirve para almacenar modificaciones de la implementación que no constituyan una versión aprobada por el grupo de momento. Son propuestas de desarrollo pendientes de aprobar o rechazar, que avanzan en paralelo con la evolución común y aprobada del producto (directorio trunk).

Cada vez que algún miembro del equipo baja a un directorio local de alguna máquina, debe proceder a realizar un *checkout* (exploración que le bajará la última versión subida de los diferentes archivos de la configuración) o un *update* de su copia local. Esto es indispensable para trabajar siempre con la última configuración existente.

Para la identificación del número de versión, optaremos de momento, por elegir el número de revisión otorgado por subversion, para así determinar rápidamente cuál debería ser la configuración más reciente que todos los miembros del equipo deberían tener.

## ANEXO A. Tablas ECS

<b>Especificación de requisitos</b>	
<b>Código ECS</b>	ER
<b>Número de versión</b>	3
<b>Número de versión (SVN)</b>	??
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/Generales/





<b>Casos de Uso</b>	
<b>Código ECS</b>	CU
<b>Número de versión</b>	3
<b>Número de versión (SVN)</b>	??
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/Generales/

<b>Gestión de Configuración del Software</b>	
<b>Código ECS</b>	GCS
<b>Número de versión</b>	2
<b>Número de versión (SVN)</b>	r159
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/Generales

<b>Planificaciones</b>	
<b>Código ECS</b>	PLF
<b>Número de versión</b>	2
<b>Número de versión (SVN)</b>	??
<b>Localización</b>	svn/trunk/doc/Generales





<b>Programa Cliente</b>	
<b>Código ECS</b>	CEG
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r136
<b>Número de copias</b>	1
<b>Localización</b>	svn/trunk/cliente/

<b>Programa Servidor</b>	
<b>Código ECS</b>	SEG
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r136
<b>Localización</b>	svn/trunk/servidor/

<b>Escenarios de Calidad del Software</b>	
<b>Código ECS</b>	ESCA
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r153
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/EscenariosDeCalidad/





<b><i>Vista de Asignación de Trabajo</i></b>	
<b>Código ECS</b>	VAT
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r145
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/Asignacion/

<b><i>Vista de Despliegue</i></b>	
<b>Código ECS</b>	VD
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r145
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/Asignacion/

<b><i>Vista de Implementación</i></b>	
<b>Código ECS</b>	VI
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r145
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/Asignacion/





<b><i>Vista de Protocolo Cliente- Servidor</i></b>	
<b>Código ECS</b>	VPCS
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r152
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/C omponentesYConectores/

<b><i>Vista de Interfaz Gráfica del Módulo Cliente</i></b>	
<b>Código ECS</b>	VIGMC
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r150
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema//Vistas/C omponentesYConectores/





<b><i>Vista de Interfaz Gráfica del Módulo Estadísticas</i></b>	
<b>Código ECS</b>	VIGME
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	r149
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/ComponentesYConectores/

<b><i>Vista del Módulo Estadísticas - Resto</i></b>	
<b>Código ECS</b>	VMER
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	??
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/ComponentesYConectores/

<b><i>Vista de Descomposición</i></b>	
<b>Código ECS</b>	VDC
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	145
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/Modulos/







<b><i>Vista de Descomposición de Protocolo P2P</i></b>	
<b>Código ECS</b>	VDCP
<b>Número de versión</b>	1
<b>Número de versión (SVN)</b>	152
<b>Número de copias</b>	2
<b>Localización</b>	svn/trunk/doc/ArquitecturaDelSistema/Vistas/M odulos/

Para más información acerca de los miembros del equipo asignados a la realización de cada ECS, consultar la vista de asignación de trabajo.

