

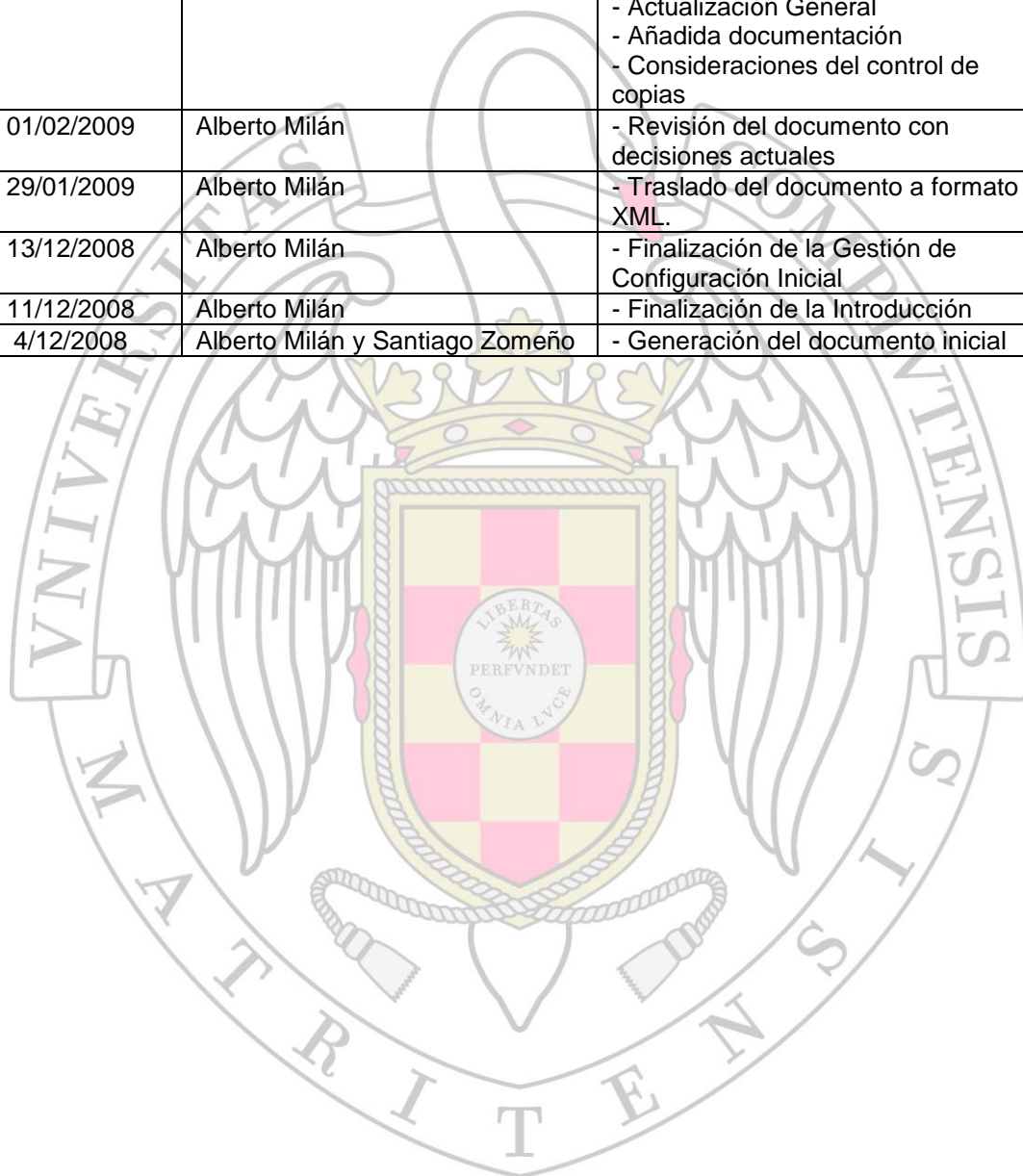
Gestión de Configuración



Proyecto Ingeniería del Software 2008/09
Casino Online

Tabla de Cambios

Fecha	Autor/Autores	Comentario
31/03/2009	Alberto Milán	- Añadido soporte a la gestión de diseño UML
24/02/2009	Alberto Milán	- Índice numerado - Corregido Diagrama del Sistema - Actualización General - Añadida documentación - Consideraciones del control de copias
01/02/2009	Alberto Milán	- Revisión del documento con decisiones actuales
29/01/2009	Alberto Milán	- Traslado del documento a formato XML.
13/12/2008	Alberto Milán	- Finalización de la Gestión de Configuración Inicial
11/12/2008	Alberto Milán	- Finalización de la Introducción
4/12/2008	Alberto Milán y Santiago Zomeño	- Generación del documento inicial



Índice

1. Definiciones y Acrónimos	4
1.1 Definiciones.....	4
1.2 Acrónimos	4
2. Plan de Gestión de Configuración Software	5
2.1 Introducción.....	5
2.2 Gestión de GCS	7
2.3 Actividades de GCS	11
3. Bibliografía y enlaces	13



1. Definiciones y Acrónimos

1.1 Definiciones

Versiones: Más comúnmente denominado revisión, dentro del ámbito de control de versiones. Marca el estado de evolución del proyecto

Variantes: Versión de producto alternativa, desarrollada para realizar alguna corrección o mejora de un modo paralelo, sin alterar la versión estable del producto.

Punto de control: Punto en el que requisitos específicos son aplicados a los elementos de configuración desarrollados.

Release: Versión particular de un elemento de configuración que está disponible para un fin específico.

1.2 Acrónimos

EC: Elemento de Configuración.

GCS: Gestión de Configuración de Software.

SVN: Subversion. Herramienta de control y gestión de versiones.

UML: Unified Modeling Language. Lenguaje de modelado de sistemas software.

2. Plan de Gestión de Configuración Software

2.1 Introducción

2.1.1 Descripción

El plan de configuración se enmarca en el proyecto Casino Online, desarrollado como proyecto de la asignatura Ingeniería del Software. El objetivo de la aplicación es la implementación de dos herramientas: un cliente y un servidor. El servidor mantendrá información de los usuarios y organizará el desarrollo y sincronización de los juegos, y el cliente, por su parte, se limitará a notificar al servidor las opciones seleccionadas por el usuario y representar las acciones que el servidor le comunique, por parte de otros usuarios.

Para el mantenimiento de los EC se ha elegido utilizar la herramienta Subversion. Subversion es una herramienta de libre distribución que permite el control de versiones de elementos software. Dicho sistema facilita el control de versiones de los distintos elementos software seleccionados por el equipo del proyecto. Está especialmente orientado a elementos conformados por texto, sobre los que permite realizar un buen manejo, control y gestión de sus modificaciones. Dicha herramienta, en su modo servidor/repositorio, estará alojada en un servicio que Google ofrece, denominado GoogleCode. Se permite, por parte de Google, un tamaño de repositorio de 1 GB (1024 MB).

El hecho de alojar el proyecto en un repositorio libre ofrecido por Google obliga a la elección de una licencia libre de cara a la exposición de todo elemento software del proyecto. Por tanto, y dadas las obligaciones que ciertas licencias libres llevan a cabo, se ha decidido liberar todo el proyecto software con una licencia *BSD* o derivada de la misma.

Ello implica que toda derivación del proyecto puede ser expuesta de forma libre, al igual que el original, o de forma privativa, según los intereses del agente ejecutor del cambio. Es decisión del equipo de desarrollo la **no**-obligación de ninguna cláusula que pudiera concluir en la limitación de libertad de este software, siendo el objetivo final el desarrollo de un software completamente libre.

2.1.2 Identificación de EC

El plan de Gestión de Configuración se aplicará, principalmente, a los dos contenidos básicos del proyecto: *código* y *documentación*. Ambos se expondrán de forma más minuciosa en el apartado 3.3.1 *Identificación de la Configuración*. Si se seleccionan nuevos elementos susceptibles de pertenecer al plan de gestión de configuración se añadirán tanto a este apartado, como al apartado antes mencionado.

2.1.3 Identificación de software ajeno incluido como parte de la GCS

Existe software ajeno al desarrollado por parte del proyecto que estará incluido dentro de la GCS. El software, a expensas de posibles cambios y/o actualizaciones, es el siguiente:

- Java JDK 1.6
- Eclipse 3.3+
- NetBeans 6.x
- ORACLE 10g Express
- Apache 2.x
- Rational Rose (versión por especificar y concluir)

Dicho software afecta a la Gestión de Configuración, dado que, si existiese cualquier cambio o problema con los mismos, afectaría de forma trascendental a otros documentos de especificaciones, que sí están incluidos en la GCS original.

2.1.4 Grado de formalismo, profundidad de control y parte del ciclo de vida del software al que se aplicará la GCS

La GCS se adecúa al nivel básico y elemental necesario para que el mantenimiento del mismo no delegue en un perjuicio en otras partes del desarrollo que han de realizarse de forma obligatoria. Intentará, en la medida de lo posible, coordinar los accesos a los distintos EC por parte de todos los componentes del grupo de desarrollo. Dicho control no especifica quién tiene acceso a determinados EC, sino que marcará dónde y cómo han de estar situados y configurados todos los EC. El control del acceso personal, quién puede tocar qué, no se marcará de forma documental ni automatizada, sino de forma organizativa entre los miembros del proyecto.

2.1.5 Limitaciones que se aplican a la GCS

La principal limitación de la GCS es el espacio temporal que pudiera conllevar. No todos los elementos pueden, o deben, ser considerados EC, ya que ello podría acarrear una pérdida de tiempo desproporcionada. Por este motivo, se han seleccionado los EC que son susceptibles de someterse a cambios, subjetivamente considerables, y aquellos componentes que requieren cierto control de versiones, con el fin de mejorar la calidad final del producto. Otra limitación, a priori secundaria, podría ser el límite físico del tamaño de repositorio obtenido. Al poseer 1024 MB de espacio para gestión de versiones, esta limitación se mitiga, pues no es objetivo el pensar que dicho espacio físico podría ser totalmente ocupado en un proyecto de esta duración.

2.1.6 Posibles impactos en el coste, planificación o desarrollo de las actividades de la GCS

El uso inadecuado, por parte de algún desarrollador, componente del grupo y/o cliente, de los EC o el deterioro o pérdida de algún elemento puede conllevar un impacto grave en la planificación y el desarrollo. Este tipo de impactos serán abordados desde la Gestión de Riesgos interna del proyecto, y, por lo tanto, es susceptible de ser configurado.

Los costes que el proyecto se ha enmarcado son indistinguibles del valor nulo, al no existir fin económico del mismo, sino poseer un valor didáctico.

2.2 Gestión de GCS

La gestión de GCS marca las responsabilidades y autoridades, tanto colectivamente como individualmente, que tendrán las actividades y su gestión.

2.2.1 Organización

La organización de la GCS viene marcada, de forma casi absoluta, por la organización del sistema de repositorio que se posee.

Dicha organización y estructura del repositorio no es mantenida por una persona en concreto, sino que es el conjunto del grupo el que, basada su coordinación, ha de mantener la coherencia y ordenación de los datos.

Esto significa que todos los componentes del grupo tienen acceso y permisos totales sobre cualquier EC, siendo la organización personal y la planificación de las tareas las que establezcan estos límites *virtuales* sobre dichos elementos.

Esta decisión está sometida a GCS en sí misma, con lo que podría verse modificada a lo largo del desarrollo del proyecto.

2.2.2 Responsabilidades

Como se ha definido en el contexto organizativo, las responsabilidades de la gestión de EC son individuales de cada componente del grupo.

Por el momento, se desconocen librerías externas al proyecto en uso, así pues el rol de *bibliotecario* no está definido. Al ser la organización de confianza mutua, se optará a que el usuario del módulo que requiera del uso de una librería documente dicho uso, mantenga la librería y ofrezca todas las facilidades al resto de componentes.

Esta decisión también está sometida a GCS, según las necesidades y cambios que, probablemente, sucedan en el desarrollo del producto.

2.2.3 Políticas, Directivas y Procedimientos

Niveles del Software en un árbol jerárquico

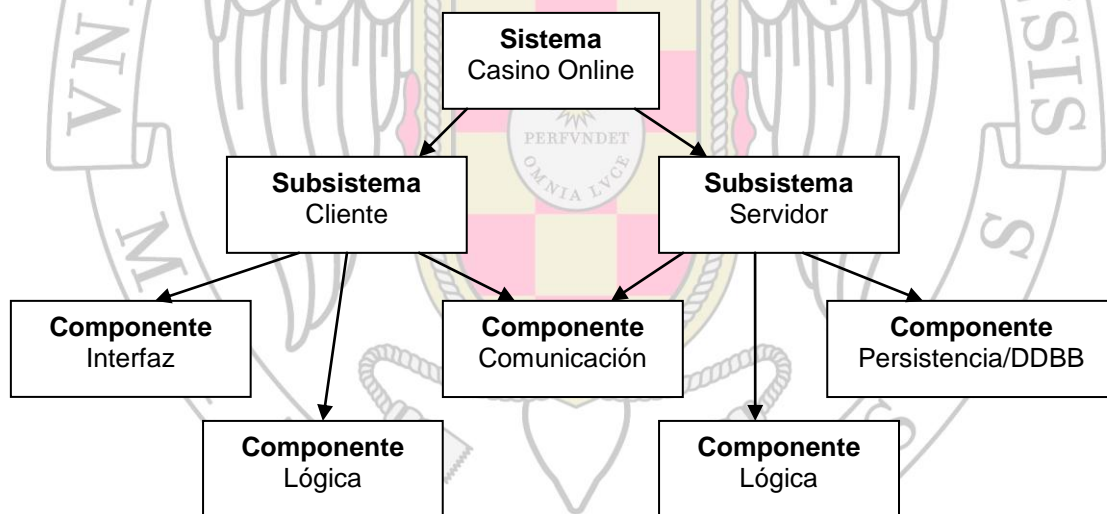
El producto Casino Online es, en sí, un *Sistema* con el que se relacionan, inferiormente de forma jerárquica, dos *Subsistemas*: el Cliente-Casino Online y el Servidor-Casino Online. Dichos subsistemas contendrán una serie de *Componentes*, que en el marco del desarrollo actual no están claramente definidos al encontrarse en desarrollo del prototipo del mismo. El Subsistema Servidor-Casino Online contendrá, al menos, los siguientes componentes:

- Comunicación
- Lógica de Negocio/Administración de eventos disparados por clientes
- Gestión de Persistencia de datos/Acceso y Gestión de BBDD

El Subsistema Cliente-Casino Online contendrá, al menos, los siguientes componentes:

- Interfaz de usuario/Presentación
- Comunicación
- Lógica de Negocio/Gestión de eventos disparados por clientes

Cada interfaz, clase y/o conjunto limitado de clases conformarán *Unidades*, con un significado y propósito atómico definido.



Nombrado de programas y módulos

Los programas que conformarán el proyecto final serán los definidos en los subsistemas previos y los módulos serán aquellos componentes con una finalidad diferenciada, que permita una organización/estructura lógica y clara del producto.

Los nombrados finales, tanto de los programas, como de los módulos, quedan sujetos a la definición de los mismos por parte de los desarrolladores, al desconocerse, actualmente, dicha nomenclatura.

Designación de versiones

La designación de versiones será la siguiente:

X.Y.Z

- X contendrá la versión del producto en su forma más genérica. Esto significa que una versión final, o release, será aquella que tenga como versión en X el valor 1.
- Y contendrá la versión del producto en su forma más particular. Ello significa que se tendrá en cuenta el factor de estabilidad del sistema. Si la última versión estable obtuvo como versión en Y el valor 14, el siguiente desarrollo nombrará en Y el valor 15, hasta que se consiga una cierta estabilidad y robustez en la aplicación, que pasará a denominarse valor 16 en Y. Esto significa que un valor par en la versión Y denota una versión estable y un valor impar una versión posiblemente inestable, o en proceso de estabilizarse.
- Z contendrá la versión referente a la revisión del producto en el control de versiones. Es decir, almacenará la revisión que Subversion contenga almacenada para el código en proceso.

Identificación de productos software

El producto software final estará formado por dos aplicaciones:

- El Servidor-Casino Online, que funcionará de forma estática como servicio ofrecido por el donante del mismo. Adicionalmente, se ha añadido un servicio Web de cara al usuario que permita interactuar en cuestiones básicas, definidas puntualmente en otros documentos, de gestión de cuenta personal.
- El Cliente-Casino Online, que funcionará en cada uno de los equipos de los usuarios que deseen acceder al servicio ofrecido por el donante.
- El entorno de diseño, modelado en UML, será el originado para la realización del producto final, implementando patrones de diseño y decisiones óptimas para todas las funcionalidades del sistema.

Identificación de documentación

Existen diversos documentos sobre los que se realiza GCS. En los mismos, se ha de identificar cada EC contenido en cada elemento mediante un convenio de identificación previo a su nombre. Véase:

- Gestión de Configuración Software: GCS
- Especificación de Requisitos Software: ERS
- Gestión de Riesgos: GR
- Casos de Uso: CU
- Planificación del Proyecto: PP
- Arquitectura del Sistema: AS

Esto significa que, por ejemplo, el caso de uso 0001 debe anteponer a su propio nombre, en su definición, el código CU, *siendo su resultado final CU_0001*.

Los nombres de los documentos, contenidos actualmente, han de ser los siguientes:

- Gestión de Configuración: GCS - Gestión de Configuración Software
- Especificación de Requisitos Software: ERS - Especificación de Requisitos Software
- Gestión de Riesgos: GR - Gestión de Riesgos
- Casos de Uso: CU - Casos de Uso
- Planificación del Proyecto: PP – Planificación del Proyecto
- Arquitectura del Sistema: AS – Arquitectura del Sistema

Todos los documentos se almacenan en formato XML. Se ha descartado el uso de documentación en formato Wiki, el anteriormente usado, porque su utilidad como gestor de documentación es cercana a la nulidad.

Por este motivo se ha elegido el formato XML, para que, dentro de la suite ofimática Microsoft Office, se puedan realizar tareas más elaboradas, como podría ser la inclusión de imágenes, diseños, esquemas...

Pudiera haberse elegido el formato DOC, que la suite antes mencionada admite. Se ha desestimado esa elección pues el formato DOC no es texto, sino un formato binario, y, por el contrario, el formato XML permite realizar un seguimiento incremental y gradual de sus cambios al estar codificado, de formato completa, por texto plano.

Identificación de medios y ficheros

Todos los EC han de estar implementados/desarrollados en medios que permitan su gestión de configuración.

Ello significa que no se permite otra configuración de fichero que no sea la de texto. Para mayor compatibilidad, se establece el siguiente formato de codificación de texto: **UTF-8** en todos los ficheros generados. Con este formato se intenta paliar las distintas configuraciones de guardado de texto que los distintos sistemas poseen (Windows, Linux y MACOS X, principalmente).

Si un desarrollador o componente del grupo genera algún tipo de deficiencia para el resto de compañeros debido a la no-configuración de su sistema del modo aquí detallado se compromete a corregir dicha deficiencia para el buen uso del proyecto.

Los ficheros de código también se atienen a la especificación aquí detallada, y la organización de los mismos en el repositorio se establece por consenso colectivo de los desarrolladores.

Proceso de liberación de documentos

Los documentos se liberarán y, por lo tanto, se los considerará en un estado estable y coherente al final de cada iteración.

Existe la posibilidad de que un documento no posea coherencia con el estado en código de la aplicación desarrollada, pues se permite la implementación previa a la documentación para evitar documentaciones *optimistas*, en el caso que pudieran existir.

En todo momento, y a excepción de decisiones consensuadas por componentes del grupo, cada componente ha de documentar los procesos por él mismo establecidos y de comunicar al resto de componentes dichos cambios para el conocimiento global del estado del proyecto.

La entrega y liberación de documentos se realizará siguiendo el siguiente protocolo. Se almacenará una copia del fichero *bruto* en la carpeta de entrega añadiendo a su nombre la revisión de la que procede el mismo. Así mismo, se añadirá una conversión de dicho fichero al formato *PDF* para facilitar la revisión y lectura por parte de usuarios, desarrolladores y clientes.

NOTA: Es obligación, por parte del componente que realice un cambio, la inserción de una fila en la tabla inicial del documento. En la misma se especificará el autor del cambio, la fecha del mismo y una breve descripción del cambio o contenido añadido desarrollado. Si los cambios son referentes a correcciones mínimas o sin significado semántico, dentro del proyecto, se acepta la no inserción de dicho nuevo dato en la tabla.

Proceso de liberación de productos software

Los productos software se liberarán siempre y cuando cumplan la condición inicial de encontrarse en su modo de versión Y, definida previamente, como estable.

Una vez liberados se procederá a su instalación, tanto el subsistema servidor como el subsistema cliente en los diversos equipos que se poseen para pruebas. Tras la instalación se realizarán las pruebas pertinentes, que deberán durar un tiempo mínimo de 3 días, para determinar que dicha versión es candidata a liberar.

Procesamiento de informes de incidencias, solicitudes de cambio, órdenes de cambio,...

Los informes de incidencia serán emitidos dentro de la herramienta GoogleCode, en el apartado *Issues*. Todo usuario y desarrollador está obligado a emitir cualquier incidencia que encuentra, aunque no pertenezca al ámbito de su módulo/sección. Gracias a dicha herramienta se podrá realizar un seguimiento del estado de la incidencia, hasta su posible, y deseable, cierre.

Las solicitudes de cambio, visto el modo organizativo del grupo, son inexistentes. Ello significa que cualquier desarrollador es libre de realizar un cambio, gracias a que es posible revertirlo, sin necesidad de emitir solicitud alguna.

2.3 Actividades de GCS

2.3.1 Identificación de la Configuración

A continuación se detallarán los EC seleccionados para su gestión de configuración:

- Diseño del sistema Casino Online (UML)
- Código Fuente
- Código Ejecutable
- Documentación interna al proyecto y Documentación de Usuario
- Elementos relacionados:
 - Java JDK 1.6
 - Sistemas Operativos (Windows 2000/XP, Linux 2.6+ y MACOS X 10.4.1+)
 - IDE de desarrollo Java: Eclipse 3.3+ y NetBeans 6.0+
 - Gestor de versiones Subversion

Todos estos elementos se han seleccionado, vista la posibilidad del cambio en los mismos durante el desarrollo, o vista la criticidad de los mismos en el desarrollo del software.

El repositorio de gestión de versiones contendrá el siguiente árbol (decorado) de carpetas:

svn/

- Servidor/
 - branches/ { Espacio de desarrollo de variantes/ramas de código de la aplicación Servidor }
 - tags/ { Espacio de almacenado de versiones ejecutables **liberadas** de la aplicación Servidor }
 - trunk/ { Espacio de desarrollo principal de código de la aplicación Servidor }
- Cliente/
 - branches/ { Espacio de desarrollo de variantes/ramas de código de la aplicación Cliente }
 - tags/ { Espacio de almacenado de versiones ejecutables **liberadas** de la aplicación Cliente }
 - trunk/ { Espacio de desarrollo principal de código de la aplicación Cliente }
- doc/ { Espacio de guardado y control de documentación del proyecto }
 - YYYYMMDD/ { Entrega realizada el año YYYY, mes MM y día DD }

- UML/ { Espacio de desarrollo del diseño del sistema en el lenguaje UML }

Las líneas base serán las siguientes:

Funcional	Análisis de Requisitos de la Aplicación	Definición final de la aplicación, requisitos finales y especificación de casos de uso
Distribución de funciones	Análisis de Requisitos Software	ERS de la aplicación Servidor y de la aplicación Cliente
Diseño	Diseño y Arquitectura Software	Implementación Arquitectura final y paso de pruebas establecido
Producto	Pruebas	Programas binarios e informes por parte del usuario positivos
Operación	Implantación del software	Desarrollo de la documentación de usuario: manual de usuario, manual de instalación y manual de operación

Los repositorios a utilizar están situados en la herramienta GoogleCode, que proporciona de forma gratuita bajo la obligación de liberar dicho sistema software bajo licencia abierta. La estructura de los mismos ya ha sido definida y su estimación en tamaño es no-superior a la ofrecida por la herramienta utilizada.

2.3.2 Control de la Configuración

Por decisión colectiva se ha determinado que no existe control establecido ni puntual de la gestión de elementos de configuración.

Esto conlleva que todo componente del grupo tendrá derechos y acceso a realizar cualquier tarea que estime necesaria. Es decir, se asume una **confianza mutua** por parte de todos los componentes hacia el resto, haciendo que cualquiera pueda gestionar el contenido del proyecto sin la necesidad de esperas provocadas por los permisos pertinentes.

Es obvio que este sistema no es óptimo y puede desajustar el equilibrio y seguridad del proyecto, pero estimamos más necesaria la rapidez de funcionamiento que el control de permisos. Todo esto está motivado más aún por el hecho de que los componentes del grupo raramente coincidirán en el desarrollo y, por lo tanto, es inviable que unos dependan de otros en su evolución de desarrollo.

Si existiera algún problema, o algún comportamiento erróneo por parte de algún componente del grupo, el mismo deberá corregirlo restaurando el estado previo del repositorio. Esa persona, y sólo esa persona, es responsable del problema, luego la ayuda de otros componentes está únicamente motivada por la ayuda personal al que generó el problema y, nunca, bajo obligación alguna.

Los cambios deberán ser realizados mediante actualizaciones al repositorio, en cuyo caso únicamente se solicitan dos requisitos:

- La aplicación ha de ser funcional, tal y como era antes de la actualización, si el cambio no afecta de forma drástica al diseño.
- La actualización ha de tener un comentario, tan breve y completo como se puedan conjugar ambos términos, añadido a la misma. Si no se realiza, es obligación por parte del desarrollador realizar una nueva actualización añadiendo a su nuevo comentario el comentario previo e informando que la actualización es de *Sincronización por falta de comentario*.

3. Bibliografía y enlaces

IEEE Standard de Gestión de Configuración

http://standards.ieee.org/reading/ieee/std_public/new_desc/se/828-2005.html

Descripción del Sistema de Control de Versiones Subversion

<http://es.wikipedia.org/wiki/Subversion> (Versión en Español)

[http://en.wikipedia.org/wiki/Subversion_\(software\)](http://en.wikipedia.org/wiki/Subversion_(software)) (Versión en Inglés)

Descripción del formato XML utilizado en Office 2003 para la documentación del proyecto

http://en.wikipedia.org/wiki/Microsoft_Office_XML_formats



Esta página se ha dejado en blanco de forma deliberada.

