

Plan de gestión de configuración

GESTIÓN DE PERSONAL UCM

Miguel Pascual Domínguez, Javier Pellejero Ortega, Isabel Pérez Pereda, Iván Prada Cazalla, Jesús Recio Herranz, Álvaro Rodríguez García

Gestor personal UCM

Contenido

1. Versiones	1
2. Identificación de las herramientas de trabajo a usar para microplanificación	2
3. Identificación de las herramientas de trabajo a usar para edición de documentos	3
4. Identificación de las herramientas de trabajo a usar como repositorio	5
5. Estructura de los archivos	7
6. Identificación de las herramientas para la comunicación de grupo	8

1. Versiones

En la tabla de a continuación se puede hacer un seguimiento de las versiones y personas que han realizado los cambios correspondientes.

Autor	Fecha	Versión	Descripción
Iván Prada Cazalla	16/12/2015	1.0	Creación del documento
Álvaro Rodríguez García	22/12/2015	1.1	Añadidos menores. Corrección de erratas y cambios de estilo.
Iván Prada Cazalla	04/01/2016	1.2	Cambios menores. Corrección de donde están las discusiones.
Iván Prada Cazalla	08/01/2016	1.3	Añadido IBM RSA

2. Identificación de las herramientas de trabajo a usar para microplanificación

Pivotal tracker

Pivotal Tracker es un producto para la gestión de proyectos ágiles y en los que es necesaria una coordinación de equipo. Permite descomponer según los requisitos del proyecto, por prioridades, etcétera. El equipo trabaja en distintos hilos y los va completando, además de permitir que los clientes vayan aceptando opciones y retroalimentando el proceso. La herramienta incluye el intercambio de archivos, la gestión de tareas, el seguimiento de estas, la velocidad y planificación de iteraciones, la liberación de marcadores y la creación de gráficos.

Mediante esta herramienta llevamos toda la planificación de tiempos. Con ella repartimos las tareas, les establecemos el tiempo real en el que hay que hacer las entregas, y el tiempo esperado que estimamos que pueda durar cada una.

3. Identificación de las herramientas de trabajo a usar para edición de documentos

Microsoft Office (Word y Excel):

Utilizaremos el paquete de aplicaciones ofimáticas Microsoft Office, y en concreto Word y Excel, para la creación y modificación de los documentos de texto con estilo asociado. Por tanto, la herramienta principal será Microsoft Word, con lo que el formato de archivo que utilizaremos será *.docx*. El uso de Microsoft Excel se limitará a la edición de tablas (en formato *.xlsx*) que luego se trasladarán al archivo de Word correspondiente.

Google Docs:

En caso de necesitar trabajar de forma colaborativa (simultánea) en algún tipo de documento de texto o tabla, se utilizará Google Docs. Nótese que en ningún momento Google Docs sustituirá a nuestro repositorio de desarrollo, pues cuando acabe la sesión de trabajo simultánea el archivo será exportado a formato *.docx* y añadido o modificado en el repositorio.

GanttProject:

Utilizaremos el programa GanttProject para crear y modificar los diagramas de Gantt asociados a nuestra planificación de tareas. Esto se guardarán normalmente con extensión *.gan* y serán exportados a imagen (*.jpg*) o pdf (*.pdf*) para facilitar su visualización individualmente y su inclusión en otros documentos, normalmente de texto.

PDF Creator:

Esta herramienta simple y útil cumple nos facilita la conversión de archivos de edición de texto a *.pdf* que facilita la lectura del mismo.

IBM Rational Software Architect

IBM Rational Software Architect es una herramienta completa de diseño, modelado y desarrollo para la entrega global de software. Utiliza el lenguaje de modelado unificado (UML) para diseñar servicios web y aplicaciones Java de empresa. Rational Software Architect está basado en la infraestructura de software de código abierto Eclipse y se puede ampliar con varios plug-ins de Eclipse. También puede mejorar la funcionalidad adecuándola a sus requisitos específicos con ampliaciones de Rational adquiridas por separado.

Rational Software Architect le ayuda a mantener un mejor control de los resultados de la entrega y la arquitectura con estas ventajas:

- **Soporte de modelado basado en UML** y herramientas de desarrollo controlado por modelos (MDD) que ayudan a racionalizar la creación de servicios y aplicaciones de Java y Web 2.0.
- **La potente orientación de herramientas y procesos** ayuda a reducir la complejidad y a disponer de mayor calidad y eficacia.
- **El acceso a servicios en nube** le permite sacar partido de servicios de infraestructura escalables.
- **Una plataforma flexible y ampliable** le ayuda a ofrecer software de alta calidad con un rendimiento de la inversión más rápido.

Se usará esta aplicación para el desarrollo de los diagramas de casos de uso y diagrama de clases necesarios para el desarrollo de la aplicación.

4. Identificación de las herramientas de trabajo a usar como repositorio

Github

Enlace de acceso al repositorio: <https://github.com/alvarorgtr/ProyectoIS>

Nuestro repositorio principal estará alojado en GitHub. Esta es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones git. Este repositorio nos permite llevar por tanto un seguimiento de las versiones de nuestros artefactos de desarrollo, completo con el responsable y el momento de los cambios y la posibilidad de volver a versiones anteriores. Si bien está más orientado a los archivos de código, este tipo de características lo hacen superior a alternativas como Google Drive.

Por tanto, usaremos GitHub para gestionar todos nuestros artefactos de desarrollo, es decir, como repositorio en el que mantener sincronizados todos estos artefactos.

Además, usamos una herramienta muy útil de esta plataforma, GitHub Desktop. Esta herramienta de escritorio nos permite trabajar de una forma mucho más cómoda, pudiendo alojar los archivos del proyecto en nuestro propio disco local y subiéndolos a nuestro repositorio en GitHub cuando queramos.

En el repositorio, se seguirá una estructura de carpeta por componente de programa (esto quiere decir, que cada una de las carpetas contendrá cada una de las partes que forman la entrega, y que realizan estudios de áreas diferentes de nuestro proyecto).

Para actualizar el repositorio se harán una serie de commits. Mantendrán la siguiente estructura según la importancia de los cambios que incorporen:

- En el caso de cambios menores o subidas provisionales mientras se está trabajando en un archivo, el nombre del commit será una breve descripción de los cambios realizados.
- En el caso de cambios de versión en uno de los archivos, el nombre del commit será: *Version <Número de la versión> <Nombre_del_archivo>*. El número de la versión se especificará según el siguiente esquema:
 - Podrá llevar un identificador de versión de como máximo de dos dígitos.
 - El primer dígito se cambiará en modificaciones importantes que den lugar a versiones estables.

- El segundo dígito indicara cambios menores en el documento.

Cada vez que se suba una nueva versión de un archivo, el creador de la nueva versión es responsable de añadir esta versión a la tabla del propio archivo. No obstante, el versionado queda guardado en el repositorio por si es necesario una consulta posterior.

Dado que vamos a trabajar con documentos de Word (.docx), pero las entregas se van a realizar en .pdf, es también responsabilidad del creador de una versión actualizar el .pdf correspondiente. Se entenderá que el .pdf está actualizado a la versión más reciente del archivo Word asociado.

Para la estructura de archivos y carpetas se seguirá el siguiente nombrado:

- No se incluirán acentos en nombres de carpetas o archivos.
- Si el nombre del archivo lleva varias palabras ha de separarse por '_ '.
- Si el nombre es de un directorio, se separará por espacios en blanco.
- Los nombres de archivos y carpetas han de empezar por mayúsculas.
- No se incluirá la versión en el nombre del archivo (irá dentro de cada archivo, como se indicó anteriormente).

5. Estructura de los archivos

Los archivos de texto tendrán todos la misma estructura. Se basarán en una plantilla de Microsoft Word (véase el ejemplo en este archivo), e incluirán la siguiente estructura:

- Portada del documento.
- Índice de contenido.
- Tabla de versiones, en la que se indicará la versión, la persona que lo ha realizado, la fecha en la que se creo una breve descripción.
- Contenido del documento, separado por apartados con la siguiente estructura:
 - *<Número.>. <Nombre de la sección>*

6. Identificación de las herramientas para la comunicación de grupo

Además de las reuniones presenciales del grupo, se utilizan las siguientes herramientas para la comunicación de los participantes en el proyecto:

Google Groups

En él tenemos un grupo con una dirección de correo asociada para poder mandar mensajes informativos grupales, y dejarlos registrados ellos.

Nota: Para decisiones más importantes, se tendrá una carpeta llamada “Histórico de decisiones” **en el repositorio** en la que se tendrá un documento de texto de Microsoft Word (con la estructura fijada anteriormente) en el que se recogerán las discusiones y las conclusiones finales sobre la aplicación.

Skype

Mediante esta plataforma realizaremos reuniones grupales tanto entre los pequeños grupos en los que nos dividimos, como del equipo completo (pudiendo trabajar simultáneamente desde casa, aprovechando las prestaciones del programa, como la opción de compartir escritorio, entre otras.

Llamadas telefónicas

Solo se usarán en caso de necesidad de ponerse en contacto de forma urgente con algún componente del grupo.