Manejo de las configuraciones

El presente documento muestra el plan de administración de las configuraciones para el proyecto denominado “Walkers of the city” que se realiza como trabajo práctico integrador de las materias Ingeniería de Software y Programación Concurrente. Se realiza con el fin de definir y dar a conocer un repositorio de versiones, la forma de acceso al código fuente y los documentos ubicados dentro del mismo, definir el equipo y las políticas generales de trabajo.

Indice

A. Control de versiones

B. Esquema de directorios

C. Política de etiquetado

D. Plan del esquema de ramas.

E. Entrega de releases.

F. Integrantes y Roles.

G. Herramienta de seguimiento de bugs.

**A. Control de versiones**

Se utilizó el servicio de Project Hosting que ofrece Google Code para proyectos de software libre. El hosting se configuró para que trabaje con repositorios del sistema de control de versiones Subversion (svn).

El proyecto se puede acceder a través siguiente link:

<http://code.google.com/p/juego-walkers-of-the-city>/

Para acceder al servicio de google se utiliza el cliente TortoiseSVN. Se sincroniza el proyecto con los ordenadores de cada miembro haciendo un checkout desde el TortoiseSVN con la siguiente URL de repositorio: <https://juego-walkers-of-the-city.googlecode.com/svn/trunk/.> A partir de este punto cada miembro puede crear sus propias ramas del proyecto (branch) a partir de la rama principal (trunk).

**B. Esquema de directorios**

Los repositorios Subversion (svn) respetan un esquema de directorios determinado que organiza el proyecto para optimizar el desarrollo en paralelo del proyecto. El esquema de directorios de nuestro proyecto es el siguiente:

• svn/

documentacion/

branches/

7170/

markdiaz/

icaio/

tags/

trunk/

En la carpeta documentación se incluyen los documentos de diseño y referidos al código.

En branches se encuentran las carpetas que va a utilizar cada usuario para desarrollar nuevas ramas del proyecto (ver punto d).

En la carpeta tags están aquellos avances del proyecto principal que se consideren estables y se decidan etiquetar.

En trunk va el desarrollo de la rama principal del proyecto.

**C. Política de etiquetado**

Para el etiquetado vamos a utilizar el siguiente sistema:

tags/

internos/

i1.0.0

i1.0.1

releases/

Beta/

1.0.0/

1.0.1/

1.1.0/

Final/

1.0.2/

1.1.1/

En el subdirectorio “internos” colocaremos las versiones del proyecto estables para el uso interno en el proyecto, no orientadas a una entrega. Estas versiones no serán entregables al cliente puesto que aún no cumplen con todos los requerimientos exigidos por el mismo, pero cumplen con ciertas funcionalidades que los programadores desean fijar. Agregamos una i antes del número, por ejemplo: i1.0.0

El subdirectorio “releases” contiene las versiones entregables al cliente, que cumplen la gran mayoría o todos los requerimientos exigidos por el mismo para esa version del desarrollo. Este directorio se divide en dos subdirectorios más:

* Final en donde se encontrarán las versiones entregables que cumplen absolutamente todos los requerimientos del cliente y además están testeadas.
* Beta en donde se encontrarán las versiones entregables que cumplen algunos o todos los requerimientos del cliente y no están completamente testeadas. Las versiones ubicadas en este subdirectorio tienen el fin de mostrarle funcionalidades al cliente y son candidatas a convertirse en una versión final.

**D. Plan de Esquemas de ramas y Políticas de Fusión**

Para el presente trabajo hemos seleccionado como plan de esquemas de ramas, el conocido como UCM (unified configuration management) este método se basa en “rebase before deliver”.

La principal característica de este esquema es que antes de realizar un commit de un branch determinado en la carpeta trunks\ primero se verifica si se ha modifica el trunks del cual se realizó el branch en un primer momento. Si la versión del trunks fue modificada, el commit no puede realizarse directo, sino que primero debe realizarse un “rebase”, este consiste en actualizar nuestro branch y fusionarlo con el nuevo trunks, y luego realizar el commit.

Otra de las características de este esquema es que cada usuario posee su propio branch, de tal manera que cada uno trabaja en su propia rama.

**E. Entrega de releases.**

Se le entregará al cliente un ejecutable de los distintos releases de las versiones del proyecto a medida que se va desarrollando el mismo. Estas pueden ser tanto versiones finales como betas. Los códigos fuente de los releases entregables se encontrarán en el directorio /svn/tags/releases, tanto en el subdirectorio Final como en el Beta.

Además junto con cada release se le hará entrega de manuales de usuario en formato pdf tanto para el uso, como para la instalación y la configuración. Los archivos de configuración necesarios para ese release tambien seran entregados.

Para la entrega de versiones finales, todo lo mencionado anteriormente se grabará dentro de un DVD y se le entregará al cliente, ofreciendo adicionalmente servicio de soporte para la instalacion y configuracion del sistema. Para los fallos encontrados o reportados sobre una cierta versión se ofrecerán parches descargables desde internet.

Para la entrega de versiones betas, todo lo mencionado anteriormente se comprimirá y se subirá a un sitio web en donde sea descargable por el cliente, ofreciendo adicionalmente servicio de soporte online para la instalacion y configuracion del sistema.

La instalación de cada release no dependerá nunca de instalaciones de versiones anteriores, para que el cliente pueda obtener y utilizar cualquier versión independientemente de la que tenga.

**F. Integrantes y Roles**

Los integrantes del grupo son:

* Altman Quaranta, Augusto D.
* Diaz, Marcos
* Fretes, Ricardo G.

En caso de necesitar una comunicación entre los miembros estas se realizarán vía chat o video-llamada brindada por la pagina www.facebook.com , asi como también el uso de telefonia celular.

Tendremos dos cargos destacados, los cuales son:

* CCB (change control Board) estara a cargo del señor Altman Quaranta, Augusto y Fretes, Ricardo
* Presidente de la CCB estará a cargo del señor Diaz, Marcos

Estos cargos están divididos por su responsabilidad, la responsabilidad de cada uno esta dado por:

* CCB: Toma las decisiones si los cambios a un proyecto deben ser implementados o no. Las decisiones tomadas por el change control board son tomadas como definitivas y obligatorias
* Presidente de la CCB: Establece dias y horarios de cuando se realizarán las reuniones correspondientes y dirige las mismas.

Las reuniones se realizarán 2 veces por semana los dias jueves y viernes en el horario de 15 hs, definido por el presidente de la CCB.

**G. Herramienta de seguimiento de bugs**

Para el seguimiento de errores e inconvenientes vamos a utilizar la herramienta que nos proporciona el repositorio de Google Code, para reportarlos y posteriormente hacer un seguimiento de los mismos (ver el estado, quien lo reporto, si fue resuelto, por quien, etc.).

La direccion del repositorio de issues es la siguiente:

<http://code.google.com/p/juego-walkers-of-the-city/issues/list>