Interpool

Manejo del Ambiente Controlado

Versión 2.0

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 02/09/2010 | 2.0 | Versión inicial | Vicente Acosta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contenido

[1. Definición del Ambiente Controlado 3](#_Toc271235884)

[1.1. Estructura 3](#_Toc271235885)

[1.1.1. Documentación 3](#_Toc271235886)

[1.1.2. Código 3](#_Toc271235887)

[2. Descripción del uso del Ambiente Controlado 3](#_Toc271235888)

[2.1. Clientes 3](#_Toc271235889)

[2.1.1. Cliente para la documentación 3](#_Toc271235890)

[2.1.2. Cliente para el código 3](#_Toc271235891)

[2.2. Comandos esenciales para el uso del repositorio 3](#_Toc271235892)

[2.2.1. Checkout 4](#_Toc271235893)

[2.2.2. Update 4](#_Toc271235894)

[2.2.3. Commit 4](#_Toc271235895)

[2.2.4. Add 4](#_Toc271235896)

[2.2.5. Locks 4](#_Toc271235897)

[2.3. Resolución de conflictos 5](#_Toc271235898)

[2.4. Autenticación 5](#_Toc271235899)

[3. Responsabilidades 6](#_Toc271235900)

1. Definición del Ambiente Controlado

El Ambiente Controlado, en lo que refiere a código y documentación, será implementado usando Subversion (SVN) como herramienta para el manejo del repositorio. El servidor usado será el que provee **googlecode.com**. En lo que refiere al registro de horas y documentos de carácter general, se usará Google Documents. Y por último, para agendar reuniones con el cliente y del grupo, será utilizado Google Calendar.

* 1. Estructura

Se creó un proyecto de nombre **interpool** en el sistema para el intercambio de archivos provisto por Google, y la url destinada al intercambio de los mismos es <https://proyectointerpool.googlecode.com/svn/>.

La documentación y el código tendrán carpetas distintas, con sus respectivas estructuras.

* + 1. Documentación

La documentación a entregar en las revisiones semanales se almacenará en la carpeta “Documentación” situada en el root del repositorio (<https://proyectointerpool.googlecode.com/svn/Documentacion>). Su estructura es similar a la utilizada para describir el modelo de proceso MUM.

* + 1. Código

El código se almacenará en la carpeta “trunk”. En la misma se almacenará tanto el código correspondiente al Proyecto así como ejemplos de utilidad para el grupo.

1. Descripción del uso del Ambiente Controlado
   1. Clientes
      1. Cliente para la documentación

El cliente a emplear para el repositorio SVN será Tortoise SVN. Se encuentra disponible de descarga en <http://tortoisesvn.net/downloads>. Un tutorial del mismo se puede encontrar en <http://tortoisesvn.net/support>. Dadas las restricciones del tiempo del proyecto, no es estrictamente necesario que todo el grupo lo lea, pero sí que se tenga al menos una idea general del funcionamiento del Tortoise SVN, y en caso de duda se debe recurrir al link antes dicho. El Tortoise SVN será usado preferentemente para documentación, pero los desarrolladores pueden optar usarlo si les es más sencillo.

* + 1. Cliente para el código

Opcionalmente, los desarrolladores pueden escoger un cliente SVN integrado con el Visual Studio para el intercambio de código. El cliente a emplear será el AnkhSVN, se puede descargar de <http://ankhsvn.open.collab.net/downloads>. Dicho cliente hace que no sea necesario dirigirse al lugar mismo donde se encuentra el proyecto, y que las mismas opciones que ofrece el Tortoise SVN estén disponibles desde el Visual Studio.

* 1. Comandos esenciales para el uso del repositorio
     1. Checkout

Lo primero que debe hacerse para trabajar sobre un conjunto de archivos existentes en el repositorio es hacer un *Checkout*. Lo que hace esto es crear una copia local de los archivos indicados que se encuentran en el repositorio, para poder trabajar sobre los mismos sin afectar el trabajo que otros pudieran realizar sobre el mismo archivo.

Los pasos para realizar un *Checkout* son:

* Crear el directorio donde se va a trabajar.
* En el menú contextual del directorio seleccionar *SVN Checkout*.
* Ingresar la URL del repositorio seguida de la carpeta sobre la cual se desea trabajar. Por ejemplo: para trabajar sobre la documentación la URL es: <https://proyectointerpool.googlecode.com/svn/Documentacion>.
* Presionar OK.
  + 1. Update

El comando *Update* permite actualizar el contenido de un directorio de trabajo. Es recomendable realizar *Updates* periódicamente, y es obligatorio realizarlo antes de empezar a modificar un archivo, para saber que se está trabajando sobre la última versión del mismo.

Para realizar un *Update* simplemente hay que situarse en el directorio que se desea actualizar, y en el menú contextual seleccionar *SVN Update*.

* + 1. Commit

El propósito del comando *Commit* es actualizar el repositorio con cambios realizados localmente. Se realiza cuando se hicieron cambios al proyecto, y sólo si el proyecto se encuentra en un estado consistente.

Es importante remarcar que los *Commit* deben ser de proyectos consistentes, es decir, no se pueden dejar cambios por la mitad en los documentos, y para código fuente es fundamental que compile correctamente.

Antes de realizar un *Commit* se debe realizar un *Update* y verificar que cambios ya realizados en el repositorio no estén en conflicto con cambios locales. Si existen conflictos, los mismos deben ser resueltos antes de realizar el *Commit*.

Es muy importante que al realizar un *Commit* se describan cuáles son los cambios realizados en el mensaje adjunto. Los pasos a seguir son:

* En el menú contextual del directorio de trabajo seleccionar *SVN Commit.*
* Escribir los cambios realizados en la sección *Message.* Es conveniente revisar los archivos listados como modificados para no olvidar ningún cambio y ningún mensaje.
* Presionar OK.
  + 1. Add

Si se agregan nuevos archivos a un proyecto en el directorio de trabajo éstos deben ser puestos en Control de Versión. Para esto se debe seleccionar *TortoiseSVN/Add* en el menú contextual del archivo o el directorio.

* + 1. Locks

Cuando se desee trabajar sobre determinados archivos, y es de gran importancia que los mismos no puedan ser modificados por otros, se debe usar un *Lock*. Este comando es particularmente útil para archivos difíciles de “mergear”, tales como los documentos.

Para obtener un *Lock* se debe seleccionar *TortoiseSVN/Get Lock* en elmenú contextual de un archivo. Un *Lock* debe ser liberado luego de haber completado las modificaciones y realizado el Commit. Para liberarlo se debe seleccionar la opción *TortoiseSVN/Release Lock*.

* 1. Resolución de conflictos

Cuando se realiza un *Update* y el Tortoise SVN detecta que un archivo está en conflicto con su correspondiente en el repositorio, ésto es indicado al usuario y el archivo es marcado como en conflicto. Tortoise SVN crea una copia del archivo con las modificaciones locales, otra del archivo con las modificaciones en el repositorio, y otra que tiene la última versión común al archivo local y al del repositorio.

Los pasos a seguir para resolver un conflicto son:

* En el menú contextual del archivo en conflicto seleccionar *TortoiseSVN/Edit Conflicts*. Esto abre la aplicación TortoiseMerge.
* Iterar por los cambios en los archivos eligiendo dejar los cambios del repositorio o los locales (usar menú contextual de los cambios marcados en rojo). En caso de no poder resolver un conflicto con el contenido de los archivos dejar sin modificar.
* Si algún conflicto no pudo resolverse con el TortoiseMerge se hace lo siguiente:
* Se edita manualmente el archivo. Los conflictos están marcados de la siguiente forma:

<<<<<<< nombre de archivo

cambios locales

=======

cambios del repositorio

>>>>>>> revisión

* Se marca el archivo como Resolved seleccionando en el menú contextual la opción *TortoiseSVN/Resolved*
  1. Autenticación

Debido a que Google Code es una herramienta utilizada para código abierto, cualquiera que conozca el nombre del proyecto puede descargar los archivos del mismo. Sin embargo, no sucede lo mismo con los archivos a ser subidos al proyecto.

Para poder hacer *Commit* es necesario que cada uno de los integrantes del grupo tenga un usuario y una contraseña. El encargado de SCM ingresará la cuenta de mail de cada uno de los integrantes, para que todos ellos puedan subir archivos al repositorio. Cuando se desee hacer un *Commit* debemos autenticarnos.

El usuario para autenticarse es la cuenta de correo (por ejemplo [vicenteacosta3@gmail.com](mailto:vicenteacosta3@gmail.com)), mientras que la contraseña debe obtenerse de <http://code.google.com>. Una vez ingresada la url, se debe buscar el proyecto de nombre **proyectointerpool** y en el tab Source del mismo, existe un link que proporciona la contraseña para el usuario actual. Si el usuario no se encontrara logueado a Gmail al momento de ingresar la url de Google Code, deberá autenticarse tal como debe hacerlo cuando se conecta a su cuenta de Gmail.

Afortunadamente, estos pasos de autenticación (el tener que ingresar usuario y contraseña) deben ser realizados una única vez, pues los mismos son guardados (opcionalmente) por el cliente SVN para futuros *Commits*.

1. Responsabilidades

El responsable SCM debe exigir a los integrantes los procedimientos descritos, para mantener un ambiente libre de inconsistencias. Es fundamental que todos los cambios sobre el repositorio sean hechos a conciencia.

Cada uno de los integrantes del grupo debe colocar los distintos entregables en la Fase e Iteración correspondiente.

En caso de que existan inconsistencias en algún archivo o módulo, se intentará solucionar el problema recuperando de algún otro medio (computadoras de trabajo local de los usuarios o backups del repositorio), y en el peor de los casos, se volverá a la versión que se tiene más reciente dentro de la Línea Base más nueva.

En caso de encontrar una inconsistencia que pueda trancar el avance del proyecto debe ser solucionada lo antes posible, interviniendo en dicho esfuerzo el/los responsables del conflicto y el encargado de SCM.