

Subversion

Infraestructura de despliegue de los clientes y el servidor de Subversión.

Tras la clase de Subversión que se dio en la universidad en la asignatura Iso3, se decidió usar Google Code como SVN (code.google.com). Una de las razones por las que se realizó esa elección fue debido a la facilidad de instalación y uso, mucho soporte no oficial (guías para aprender a usarlo, etc.), compatibilidad con Eclipse (mediante Subclipse) y otras herramientas e integración con las cuentas de Google (gmail, gtalk...).

Google code, en su modalidad de hosting de proyectos, ofrece los siguientes servicios:

- Control de versiones (obvio).
- Seguimiento de errores o bugs.
- Wiki de documentación.
- Gestor de descarga de ficheros.

En el caso del proyecto que nos concierne, únicamente se ha hecho uso de la primera de las funcionalidades.

Por consiguiente, no se tuvo nada más que abrir una cuenta en Google code y, posteriormente, crear un proyecto (llamado en nuestro caso *iso3-dst*), cuya dirección se encuentra en <https://code.google.com/p/iso3-dst/>.



Imagen 1. Home del proyecto de ISO3.

En cuanto a los clientes, se han usado dos tipos.

Por una parte se ha optado por TortoiseSVN. Este software fue propuesto en clase y así se ha puesto a su uso. Sirve, y mucho, para gestionar el SVN desde el explorador de Windows (las ventanas donde se gestionan las carpetas).

Además, dentro del IDE de Eclipse se ha instalado el plugin llamado Subclipse. Éste es más útil, si cabe y desde el punto de vista del programador en Eclipse, que el TortoiseSVN:

- Gestión y configuración del SVN desde Eclipse.
- Fácil instalación.
- Posibilidad de realizar todas las tareas que ofrece TortoiseSVN, pero desde Eclipse.

Gestión de repositorios y usuarios

El grupo que hemos formado para el proyecto de ISO3 está compuesto por tres personas. El repositorio que se ha usado ha sido el mismo para las 3 personas. Los privilegios o roles que se han concedido han sido los siguientes:

- Owner: Jon Inazio Sánchez.
- Committer: Lucía Lezana.
- Committer: Eztizen Garai.

Los permisos que venían por defecto no se han visto modificados, debido a que la complejidad del proyecto no lo requería.

Una vez que se creó el repositorio, se comenzó a introducir ficheros y carpetas.

Estructura del proyecto en el repositorio

El repositorio del proyecto tiene la siguiente estructura (se comenta los directorios que se han añadido y que no eran necesarios):

- **svn**
 - **.settings**
 - **branches**: no se ha usado.
 - **src**: no se ha usado.
 - **tags**: no se ha usado.
 - **trunk**
 - **.settings**
 - **bat**: contiene los scripts (de Windows) para automatizar tareas: un script para arrancar Tomcat, y otro script para realizar el despliegue del proyecto en Tomcat con Ant (desde limpiar ficheros, pasando por compilación nueva de binarios hasta el despliegue en tomcat).
 - **bd**: copia de seguridad de base de datos.
 - **conf**: ficheros de configuración.
 - **css**
 - **dist**
 - **doc**: documentos DOC y enunciados PDF del proyecto.
 - **html**
 - **jsp**
 - **lib**
 - **sql**: scripts para la introducción de la BD y de sus datos.
 - **src**
 - **iso3**
 - **pt**
 - **action**

- **dao**
- **model**
- **service**
- **wiki**: no se ha usado.

Tipo de uso que se le ha dado (copia de seguridad, coordinación entre varios usuarios, control de versiones monousuario, etc.)

Debido a que en el grupo somos 3 los integrantes, se ha decantado por muchas razones usar el SVN. Para empezar, se puede trabajar desde cualquier ordenador (siempre y cuando se suba y se “commiteen” los últimos cambios), ya estés a la mañana en casa y a la tarde en la universidad con ordenadores diferentes.

Por otra parte, sirve a modo de backup. En caso de haber perdido el ordenador por robo, que se hubiese estropeado, etc., no hubiera habido problemas. Restaurar a la versión de HEAD y listo (siempre y cuando se hubiera subido al SVN la revisión que se estaba realizando).

Por último, cobraba gran importancia tener una herramienta mediante la cual pudiéramos coordinarnos a la hora del desarrollo del proyecto. Además, en control de versiones es algo que le va a costar salirse fuera de nuestras vidas de programadores.

Historial de cambios (incluyendo capturas de pantalla donde se muestren las distintas revisiones del proyecto y las fechas)

Prácticamente ha habido 3 grandes bloques de revisiones (además de una última):

- 1) Desde la primera revisión hasta que Hibernate funcionó y con la capa DAO funcionando
- 2) Primera parte de Struts2: profesor.
- 3) Segunda parte de Struts2: estudiante.
- 4) Todo lo de Struts2, hibernate y la capa DAO funcionando. Además, todos los ficheros DOC puestos en orden.
























Committed Changes				23 - 1 of 23
Rev	Scores	Commit log message	Date	Author
 r23		Versión final	Dec 11 (6 days ago)	maya.takameru@gmail...
 r22		Esta versión incluye la parte del estudiante completamente hecha, funcional y sin errores. La capa PTDAO ha sido corregida, ya que contenía varios fallos (sobre todo de iniciarSesion() y de matricularse y desmatricularse).	Nov 30, 2012	captain06@gmail.com
 r21		Parte de estudiante hecha y funcionando correctamente: - Mostrar todas las asignaturas matriculadas de un estudiante. - Mostrar todas las notas de un estudiante. - Mostrar las notas de un estudiante en una asignatura. - Mostrar la lista de ...	Nov 27, 2012	captain06@gmail.com
 r20		[No log message]	Nov 19, 2012	maya.takameru@gmail...
 r19		[No log message]	Nov 19, 2012	captain06@gmail.com
 r18		- Avanzado un poco en la parte de struts2 de estudiante. [IMPORTANTE BUG] - Encontrado un bug en la capa DAO: NO SE DEBE DE CERRAR LA SESIÓN DE HIBERNATE, sino, los datos no van a estar disponibles. Solo se cierra la sesi...	Nov 15, 2012	captain06@gmail.com
 r17		- Login real hecho. - Tiene errores.	Nov 14, 2012	captain06@gmail.com
 r16		- Hecho login virtual. El real, no captura bien los datos del formulario. - Añadidos dos scripts: ejecutar tomcat y ejecutar todos los comandos del ant de un tirón.	Nov 13, 2012	captain06@gmail.com
 r15		Versión con Struts2 integrado. No funciona el login, pero falta poco para que sí lo haga.	Nov 11, 2012	captain06@gmail.com
 r14		Versión 1.0 - Persistencia - Faltaría probar los dos paquetes al 100% con las clases Test de ambos (os dejo la tarea a Lucía y Ezitzen :p). - Extraer los dos ficheros .zip de la carpeta /lib al directorio C:/ (para evitamos problemas con director...	Oct 19, 2012	captain06@gmail.com
 r13		[No log message]	Oct 15, 2012	eztizengarai4@gmail.com
 r12		Parte en la que se ha empezado con la capa DAO (dao/PTDAO.java), aunque faltan bastantes métodos para terminarla.	Oct 10, 2012	captain06@gmail.com
 r11		Primera versión base. - Hecho hibernate. - Falta la capa DAO (se ha empezado).	Oct 9, 2012	captain06@gmail.com
 r10		En teoría debería de funcionar bien. Hasta este punto, se supone que hibernate está completo. Se ha empezado a implementar la capa DAO.	Oct 9, 2012	captain06@gmail.com
 r9		[No log message]	Oct 8, 2012	captain06@gmail.com
 r8		Correcciones en ficheros hibernate y añadido el test. Mañana se probarán.	Oct 5, 2012	captain06@gmail.com
 r7		No llega a versión base, pero contiene todos los ficheros HBM.XML creados (para cada clase que se va a mapear). Falta comprobar mediante la clase Test que funciona bien hibernate (que está bien configurado).	Oct 4, 2012	captain06@gmail.com
 r6		Primera versión en la que funciona hibernate. Test con el profesor correcto. Se ha añadido un jar necesario (hibernate-jpa-2.0-api-1.0.1.Final.jar). Modificado el fichero hibernate.cfg.xml y añadidos los DTD de la configuración y el mapeo de cla...	Oct 4, 2012	captain06@gmail.com
 r5		Añadido en el build.xml en la línea del target build, en el javac: includeantruntime="false". Sirve para evitar que salga el mensaje "[javac] build.xml:29: warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false f...	Oct 4, 2012	captain06@gmail.com
 r4		Actualizados los jars de javassist y slf4j con las versiones más actuales (a las versiones que se facilitan junto a la versión d hibernate que se usaba). Quizás no funcionaba hibernate por usar versiones anteriores de javassist y slf4j a la de hib...	Oct 4, 2012	captain06@gmail.com
 r3		Añadidos varios jars (no sé si son todos necesarios), salvo los que existían (hibernate3.jar y mysql-connector-java-5.1.13-bin.jar). Varios cambios en ficheros hibernate.	Oct 3, 2012	captain06@gmail.com
 r2		Primeros ficheros subidos. Todavía queda mucho trabajo.	Oct 3, 2012	captain06@gmail.com
 r1		Initial directory structure.	Oct 3, 2012	---
				23 - 1 of 23

Imagen 2. Todas las revisiones del proyecto.

Conclusiones sobre su uso (ventajas, problemas y otros aspectos que el alumno quiera destacar)

En definitiva, el uso de una herramienta como ésta no acarrea nada más que ventajas, si bien es cierto que al principio cuesta un poco hacerse a Subclipse o TortoiseSVN.

El hecho de disponer de una copia de seguridad en la nube de google, donde se puede trabajar en equipo con los miembros del grupo y, además, controlar las versiones y bugs, te permite desarrollar un proyecto con mucha tranquilidad.

Aunque, si bien es cierto que tu proyecto lo puede consultar toda aquella persona que lo encuentre.

Finalizando, no hemos tenido ningún problema con el uso de Google Code, Subclipse o TortoiseSVN. Es más, nos ha facilitado mucho el desarrollo y, cosa que es de agradecer, se ha simplificado la forma con la que se pueden compartir proyectos como este. Antes de conocer SVN, nuestro grupo había probado a usar Dropbox para compartir proyectos, pero resultó ser un incordio y no dio nada más que problemas.

En conclusión, un SVN facilita la vida del desarrollador y, una vez hecho a la plataforma, todo son ventajas.