

Presentación

Índice

1	Guía de laboratorio.....	2
1.1	Ordenadores del aula de MAC de la Biblioteca General.....	2
1.2	Apuntes del Especialista.....	2
1.3	Moodle.....	3
1.4	Proyectos y espacios de trabajo en Eclipse.....	5
1.5	Repositorios SVN.....	12
1.6	Como realizar la entrega de los ejercicios.....	20
1.7	Ciclo de trabajo.....	23
1.8	Entrega de ejercicios.....	24
2	Guía de laboratorio - Ejercicios.....	24
2.1	Proyecto Java.....	24
2.2	Subversion.....	24
2.3	Proyecto JavaME.....	24
2.4	Emuladores de Android.....	25
2.5	Proyecto Android.....	25
2.6	Entrega de ejercicios.....	25
2.7	Proyecto iOS.....	25

1. Guía de laboratorio

En esta sesión de ejercicios vamos a describir los aspectos más importantes de las distintas plataformas, utilidades y sitios web necesarios para el desarrollo de las prácticas del Especialista.

Los elementos más importantes son los siguientes.

- **Ordenadores.** Las prácticas se desarrollarán en el aula de ordenadores Mac. Las diferentes plataformas que utilizaremos están instaladas. Todas ellas serían usables en otros sistemas operativos salvo la de desarrollo en iOS, para la que necesitamos los ordenadores Mac.
- **Repositorio Subversion privado.** Un repositorio subversion privado para cada alumno permitirá guardar todos los proyectos Java en un servidor del Especialista. De esta forma, los proyectos realizados podrán recuperarse aún en el caso de que se pierda la máquina virtual VMware, proporcionándose un grado adicional de seguridad. En algunos casos el repositorio SVN se utilizará también para realizar entregas de trabajos y prácticas.
- **Apuntes y materiales docentes.** Todos los apuntes, transparencias y materiales docentes están disponibles en una zona restringida de la web del Especialista.
- **Moodle.** Se utilizará un servidor Moodle como plataforma de interacción educativa on-line. Usaremos principalmente sus funcionalidades de foros y de entregas de ejercicios.

A continuación vamos a detallar el uso de estos elementos y algunos otros también necesarios para el desarrollo del Especialista.

1.1. Ordenadores del aula de MAC de la Biblioteca General

Los laboratorios en los que se realizarán las prácticas están gestionados por el CPD. Para acceder a ellos se utilizará el usuario `alumno`. No tendrás permisos de administración pero todo el software necesario para el curso está instalado.

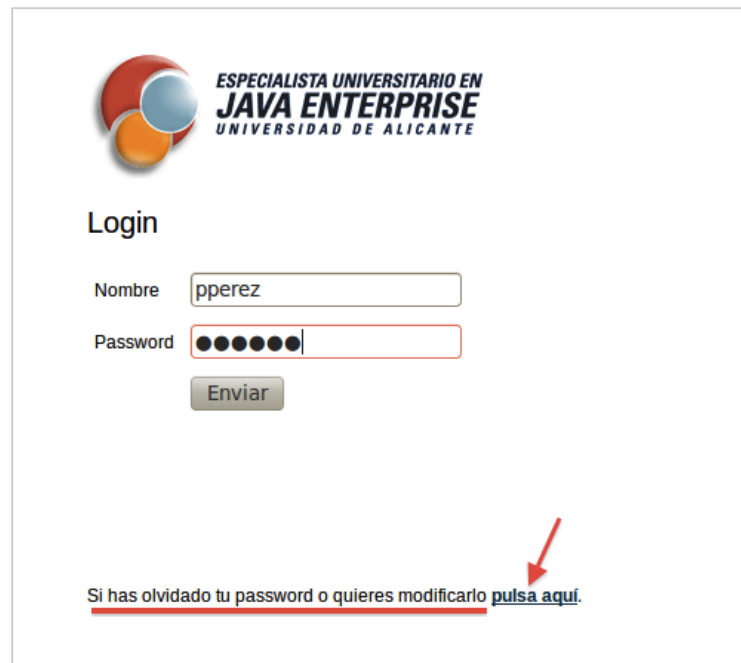
¡Cuidado con tus datos! No debes confiar en que lo que dejes en el disco duro del ordenador se encuentre en la siguiente sesión. Los discos duros de los ordenadores se reestablecen en cada arranque. Es muy importante que hagas un uso regular del repositorio Subversion que hemos instalado para el curso.

1.2. Apuntes del Especialista

Los apuntes, transparencias, plantillas para los ejercicios y demás material docente se encuentran en un sitio web restringido a los alumnos del Especialista. Puedes acceder a esta zona pinchando en el recuadro correspondiente (azul) en la web pública del Especialista (<http://www.jtech.ua.es/>) o escribiendo directamente la URL

<http://www.jtech.ua.es/dadm/restringido/>. Una vez introduzcas tu usuario del Especialista y tu contraseña podrás acceder a esta web.

El login en este sitio web es las iniciales de tu nombre y tu primer apellido. La contraseña es el número de DNI, sin la letra de control. Por ejemplo, si el nombre del alumno es Pedro Pérez, su login es pperez



ESPECIALISTA UNIVERSITARIO EN
JAVA ENTERPRISE
UNIVERSIDAD DE ALICANTE

Login

Nombre

Password

Si has olvidado tu password o quieres modificarlo [pulsar aquí](#)

Si deseas cambiar tu contraseña (o no la recuerdas), puedes pulsar en el enlace correspondiente. Se enviará un mensaje a tu dirección de correo con un enlace con el que podrás modificar la contraseña.

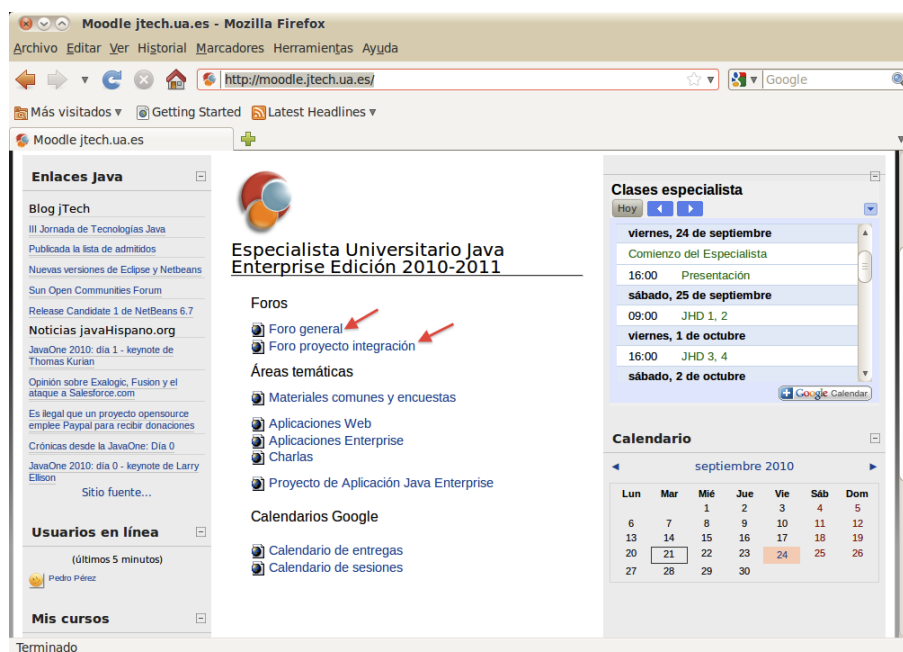
1.3. Moodle

Vamos a utilizar Moodle como plataforma de trabajo colaborativo. La usaremos para gestionar los foros, las entregas de ejercicios y alguna que otra encuesta que iremos presentando. Se accede pinchando en el recuadro correspondiente (amarillo) en la web del Especialista o escribiendo directamente la URL <http://moodle.jtech.ua.es>.

Al igual que el resto de recursos de acceso restringido, el usuario de acceso será las iniciales del nombre y tu primer apellido. La contraseña inicial será el DNI, y la primera vez que te registres te pedirá una nueva contraseña.

Uno de los elementos principales que utilizaremos de Moodle son los foros, que utilizaremos para resolver dudas y para las sesiones de consulta on-line del proyecto de integración. Podemos acceder a la plataforma desde la url <http://moodle.jtech.ua.es/>.

Desde la página principal de Moodle se puede acceder directamente a los foros. Se llaman *Foro general* y *Foro proyecto integración*.



Para poder utilizar correctamente el foro es muy importante que actualices tu foto en Moodle. Para eso, debes pinchar en tu nombre (en la zona de *Usuarios en línea* o abajo a la derecha, donde pone *Usted se ha autenticado como*) y seleccionar la pestaña *Editar información*. Allí puedes colocar tu foto.

Cuando todos tenemos la foto es mucho más sencillo contestar e interactuar en el foro. La siguiente imagen muestra un ejemplo del foro general en la edición 2009-2010 del especialista de Java Enterprise:

Topic	User	Replies	Last Post
el problema del upload.php sigue pasando al entregar	Ulises Córdoba Ferri	5	mar, 9 de feb de 2010, 12:03
Entrega de la integración de servidores de aplicaciones	Domingo Gallardo	0	mar, 9 de feb de 2010, 09:41
Instalación de "Guest Additions" en la máquina virtual	Domingo Gallardo	0	sáb, 6 de feb de 2010, 08:50
[SAPP] Subentrega sesion 1 y 2	Sergio Sánchez Martínez	16	sáb, 6 de feb de 2010, 08:45
[SAPP] Como entregar sesion 3 y 4?	Juan Antonio Osuna Ramírez	16	vie, 5 de feb de 2010, 09:35
Fecha limite de entrega JPA	Rafael López Pérez	5	vie, 5 de feb de 2010, 09:29
[Integración SAPP] - Crear proyecto desde CVS	Sergio Sánchez Martínez	2	jue, 4 de feb de 2010, 14:19
[SAPP] sesion 3	Sergio Sánchez Martínez	4	jue, 4 de feb de 2010, 10:53
[JPA] Sesion 3 Herencia	Álvaro Pérez De Pieri	2	jue, 4 de feb de 2010, 00:03
[JPA] Enumerados en JPQL	Manuel Mestres Baeza	5	mié, 27 de ene de 2010, 19:01

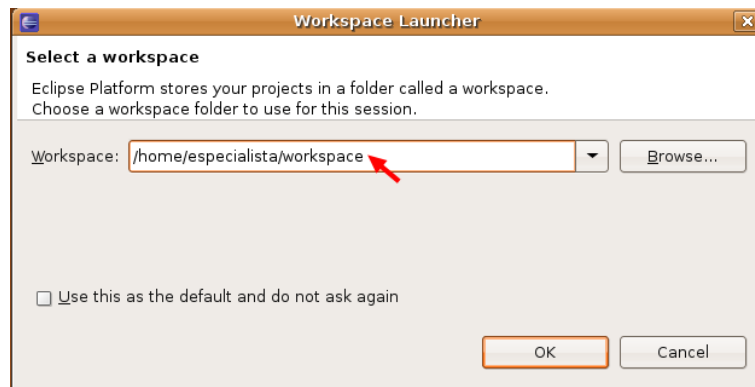
1.4. Proyectos y espacios de trabajo en Eclipse

Eclipse es el entorno de programación que se va a usar en todas los módulos del curso. El trabajo en Eclipse se organiza en proyectos que se guardan en carpetas que Eclipse denomina espacios de trabajo (*Workspaces*). Un espacio de trabajo se corresponde con un directorio físico del disco duro y los proyectos se corresponden con subdirectorios suyos. Para organizar mejor los trabajos que se irán realizando, trabajaremos con dos espacios de trabajo. Uno para los proyectos realizados en los distintos módulos del especialista y otro para los distintos proyectos de integración.

Nota:

Eclipse guarda la configuración del espacio de trabajo y de los proyectos en ficheros ocultos que residen en los mismos directorios que los espacios de trabajo y los proyectos. Los ficheros más importantes son `.metadata` que se guarda en el directorio raíz del espacio de trabajo, `.project` que se guarda en el directorio raíz de cada proyecto y otro fichero `.classpath` en cada proyecto con información necesaria para la compilación en Java de las clases del proyecto.

Cuando arrancamos Eclipse por primera vez aparece un cuadro de diálogo en el que nos pide el espacio de trabajo en el que vamos a trabajar.



Por defecto, Eclipse sugiere el directorio llamado *workspace* en el *home* del usuario. En el caso de la MV el directorio que sugiere Eclipse es `/home/especialista/workspace`. Si aceptamos ese espacio de trabajo, Eclipse crea el directorio y comenzaremos a trabajar en él.

Podemos abrir otro espacio de trabajo pulsando el botón *Browse* o escribiendo su ruta en el campo de texto. Eclipse abrirá el directorio correspondiente, o creará un nuevo directorio si no existe.

Un espacio de trabajo puede contener distintos proyectos. Cada proyecto puede contener (entre otras cosas) un conjunto de clases Java y un conjunto de librerías. Los proyectos pueden estar relacionados, de forma que uno utilice clases o librerías Java que residen en otro. Veremos todo esto con detalle en el módulo *Java y Herramientas de Desarrollo*. En la sesión de hoy estudiaremos los aspectos más sencillos relacionados con los proyectos. Entre otras cosas, cómo crearlos, importarlos, guardar una copia de seguridad o subirlos a un repositorio SVN. Por último, explicaremos cómo realizar una entrega de los ejercicios realizados utilizando el repositorio SVN.

1.4.1. Creación e importación de proyectos

En este pequeño ejercicio vamos a comprobar lo sencillo que es crear nuevos proyectos en el espacio de trabajo Eclipse. Durante el curso, vamos a trabajar con dos espacios de trabajo:

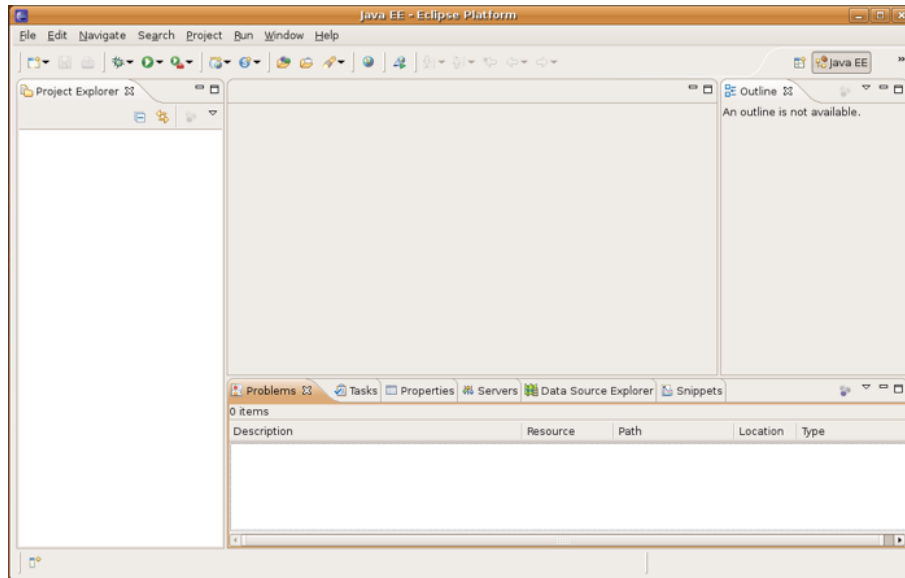
- El espacio de trabajo por defecto (*workspace*) en el que guardaremos los proyectos de los módulos
- El espacio de trabajo del proyecto de integración, que llamaremos (*proyint*)

Vamos a empezar creando un pequeño proyecto Java. Para ello seguiremos los siguientes pasos:

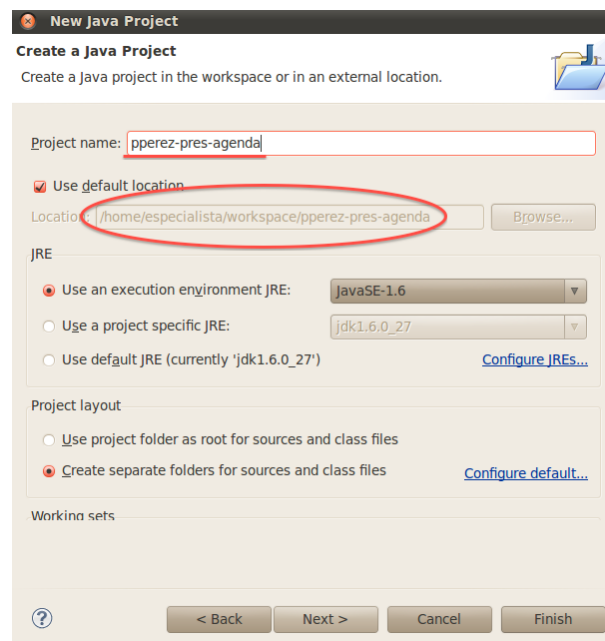
1. Abrimos Eclipse con el espacio de trabajo *workspace*.
2. Creamos un proyecto (*pperez-pres-agenda*) y algún fichero dentro del proyecto.
3. Importamos otro proyecto en el espacio de trabajo.

Vamos a continuación a realizar estos pasos de forma detallada.

Arrancamos Eclipse con el espacio de trabajo `workspace` definido. Tendrá el siguiente aspecto:



2. El panel vertical izquierdo es el destinado a contener los proyectos, ahora está vacío. Para crear un proyecto podemos escoger la opción *File > New > Java Project...* o pulsar el botón derecho sobre el panel de proyectos y escoger la opción *New > Java Project...*. En cualquiera de estos casos, aparecerá un asistente que nos guiará para crear el tipo de proyecto que deseemos. En este caso vamos a crear nuestro primer proyecto Java. Escribimos como nombre del proyecto `pres-agenda`, precedido por nuestro nombre de usuario: `pperez-pres-agenda`. Todos los nombres de proyecto vamos a escribirlos utilizando este formato: nuestro nombre de usuario, el nombre corto del módulo en que estemos haciendo los ejercicios y el nombre del proyecto. De esta forma podremos almacenarlos todos en un mismo directorio o repositorio (el espacio de trabajo `workspace` de Eclipse y el repositorio SVN). La utilización de tu login en el prefijo hará posible publicar, compartir o corregir los mismos proyectos realizados por distintos alumnos.



Cuando pulsamos *Finish* Eclipse crea el proyecto, creando un directorio nuevo en el espacio de trabajo. Aparece como una carpeta en el panel de proyectos. Una vez creado el proyecto, vamos a crear en él un pequeño programa Java, que nos va a servir para demostrar algunas de las características de Eclipse. En el programa vamos a definir una clase *Tarjeta* con los datos de una tarjeta de contactos: *id* (int), *nombre* (String) y *eMail* (String) y una clase *Main* con el programa principal de la aplicación, que crea un par de tarjetas, las guarda en un array y lo recorre para mostrarlas por la salida estándar.

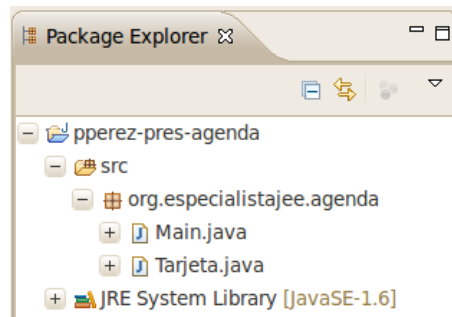
Todas las clases deben estar en el paquete `org.especialistajee.agenda`. Es una costumbre habitual utilizar como prefijo en el nombre del paquete la URL de la empresa u organización que desarrolla el código en orden inverso. De esta forma se garantiza que el nombre completo de la clase es único y no coincide con otras clases desarrolladas por otras organizaciones. En nuestro caso la URL <http://especialistajee.org> está redirigida al frontal del Especialista: <http://web.ua.es/especialistajava>.

Para crear una clase seleccionamos *New > Class* y rellenamos el cuadro de diálogo con el nombre de la clase y el paquete. Comenzamos con la clase *Tarjeta*:

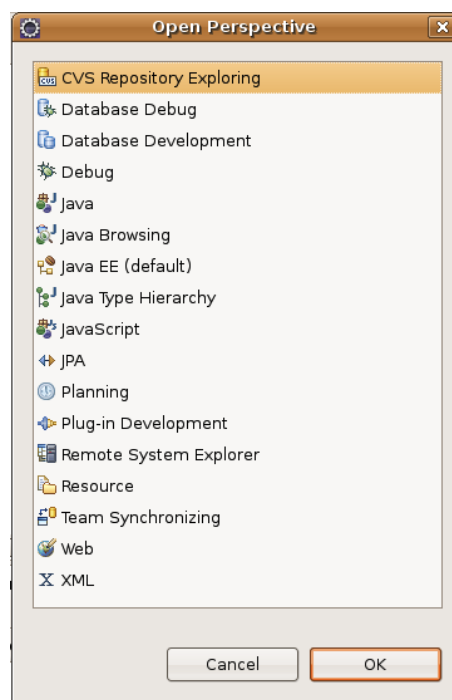
Escribimos la clase *Tarjeta* con los campos indicados, los *getters* y *setters* correspondiente y un constructor que cree una tarjeta nueva a partir de un identificador. Podemos probar a utilizar la ayuda de Eclipse para generar código con la opción *Source > Generate Getters and Setters ...* o *Source > Generate Constructor using fields ...*. Generamos también de forma automática el método `toString`.

Creamos una clase *Main* en la que declaramos un array de tarjetas, creamos y añadimos un par de tarjetas y recorremos el array imprimiendo sus datos en la salida estándar. La

estructura de clases que aparece en el proyecto es esta:

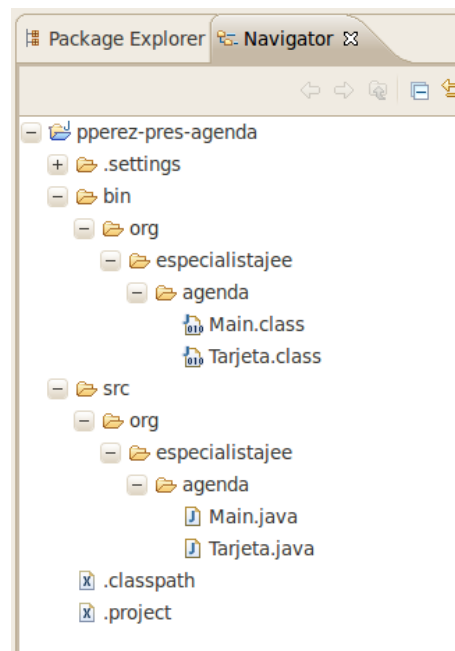


Antes de lanzar la aplicación vamos a probar a modificar la configuración de los paneles en la perspectiva actual. Una perspectiva es un conjunto de vistas en unas posiciones determinadas. Una vista es un panel especializado de Eclipse que permite interactuar sobre una determinada característica del proyecto. Eclipse contiene una enorme cantidad de vistas que permiten gestionar múltiples aspectos del desarrollo: listas de tareas por hacer, baterías de pruebas, despliegue, edición y compilación, etc. Todas estas vistas se organizan en un conjunto de perspectivas predefinidas.



Las perspectivas que vamos a utilizar principalmente en el curso son Java y Java EE.

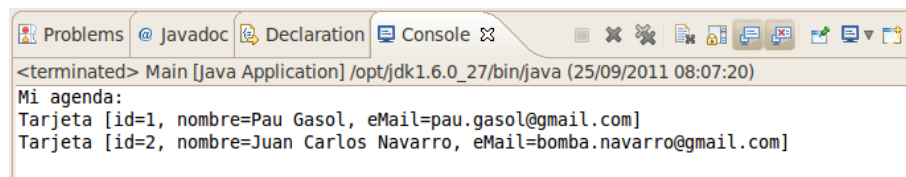
Añadimos la vista *Navigator* que nos permite explorar la estructura de archivos creados en el disco duro. La podemos colocar junto a la vista *Package Explorer*:



Vemos que Eclipse realiza automáticamente todo el trabajo de compilar las clases .java a ficheros .class y colocarlos en los directorios correspondientes a sus paquetes. También podemos ver los ficheros .settings, .classpath y .project donde guarda distintas características del proyecto. Es recomendable dejar esta vista *Navigator* porque la utilizaremos más de una vez para visualizar la estructura real del sistema de archivos.

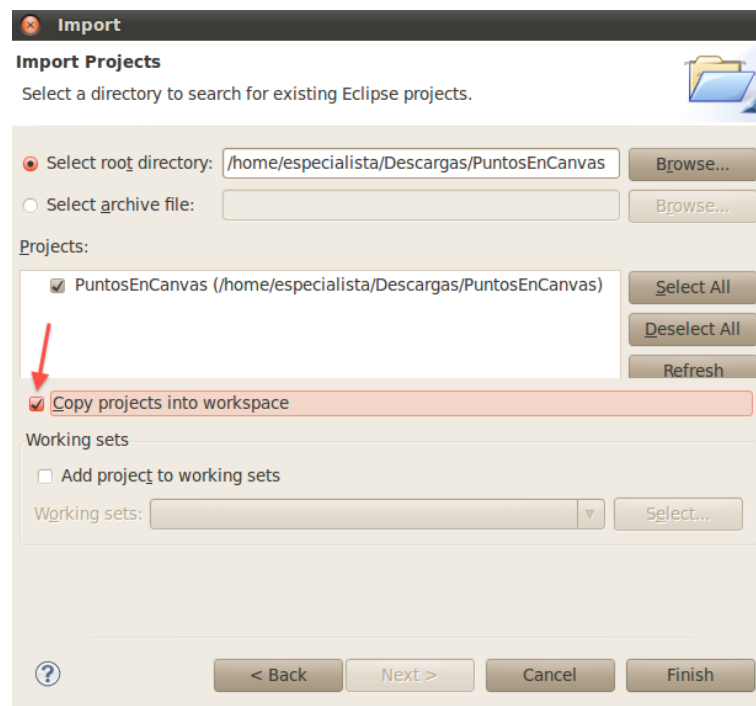
En algunos casos, puede ser que al mover las distintas vistas dejemos bastante cambiada una perspectiva. Si queremos volver a la configuración por defecto debemos seleccionar *Window > Reset Perspective..*

3. Ejecutamos la aplicación con la opción *Run > Run As > Java Application*. Una forma rápida de acceder a esta opción es con el icono de *play* en la barra superior de eclipse.

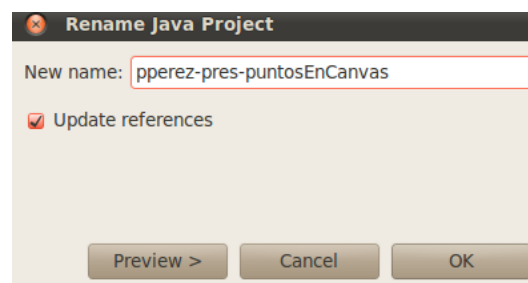


4. Por último vamos a probar la importación de proyectos Eclipse. Vamos a hacer la prueba con un proyecto que hemos creado para esta sesión. Se encuentra comprimido en el fichero *sesion01-ejercicios.zip*, en la página principal de los apuntes de este módulo (lo puedes descargar directamente desde [este enlace](#)). Guarda el fichero ZIP en la carpeta de descargas, por ejemplo, y descomprímelo. Verás el proyecto *PuntosEnCanvas*. Se trata de una sencilla aplicación Java en la que se abre una ventana y se dibujan algunos puntos en ella.

Importa el proyecto con la opción *File > Import...* (o pulsando el botón derecho sobre el panel y seleccionando *Import > Import...*). Selecciona *General > Existing Project into Workspace*. Selecciona *Select root directory* y escoge el directorio del proyecto que acabas de descargar. Puedes ver cómo aparece marcada la opción *Copy projects into workspace*, indicando que Eclipse copiará físicamente el proyecto al espacio de trabajo. Si se dejara desmarcada Eclipse trabajaría los ficheros del proyecto en su localización original. La dejamos marcada:



Una vez importado, puedes consultar el código fuente del proyecto y también ejecutarlo. Después, para seguir el convenio de que todos los proyectos de un mismo módulo comiencen su nombre con el módulo, cambia el nombre del proyecto a *pperez-pres-puntosEnCanvas* con la opción *Refactor > Rename...*:



Una vez creados e importados los proyectos, vamos a ver en el siguiente apartado cómo subirlos al repositorio SVN.

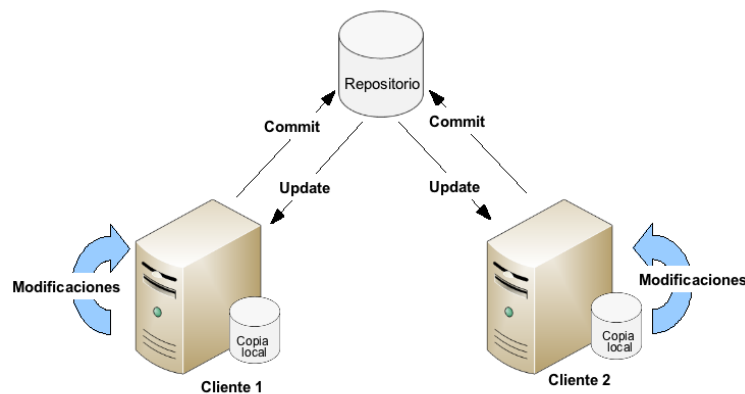
Nota:

Hay que hacer notar la gran portabilidad de los proyectos de Eclipse (debido sobre todo a que Eclipse está desarrollado en Java). Es posible comenzar a trabajar en Windows y seguir trabajando en cualquier otro sistema operativo (Linux o Mac OS X). El único detalle a tener en cuenta es el de asegurarse de usar la misma **codificación de texto** (*text encoding*) en todos los sistemas. Por defecto, Eclipse usa la codificación de texto del sistema operativo, lo que puede dar lugar a problemas de compatibilidad al abrir ficheros de texto creados en otros sistemas operativos. Lo recomendable cuando vamos a trabajar con distintos SOs es escoger una única codificación (ISO-8859-1, por ejemplo) y un único delimitador de fin de línea (el de Unix, por ejemplo). Para escoger la codificación en la que va a trabajar Eclipse hay que seleccionar la opción *Window > Preferences > General > Workspace*.

1.5. Repositorios SVN

Los sistemas de control de versiones permiten mantener una historia de los cambios que se realizan en los ficheros que componen un proyecto. En el caso de los sistemas distribuidos, como CVS o SVN, permiten además que distintos desarrolladores puedan trabajar simultáneamente en proyectos compartidos. El sistema registra los cambios de cada desarrollador y guarda las versiones en un repositorio común accesible a todos.

Uno de los sistemas más veteranos de control de versiones es CVS. Está muy extendido en múltiples comunidades de desarrolladores y existen bastantes herramientas gráficas y clientes que lo soportan. [Apache Subversion](#) (SVN) se desarrolla para mejorar algunos aspectos de CVS, pero manteniendo su filosofía y funcionamiento.



Tanto CVS como SVN proporcionan un sistema cliente-servidor de control de versiones y permiten que varios desarrolladores trabajen simultáneamente en un mismo proyecto. Permiten sincronizar un único repositorio remoto (donde se encuentra el proyecto en el que se está trabajando) con múltiples clientes locales (desarrolladores), de forma que todos los clientes comparten el mismo proyecto, pueden subir sus cambios y mantener sus proyectos sincronizados. Existe una gran variedad de sistemas de control de versiones

como Subversion, Mercurial o Git. Hemos escogido CVS/SVN por motivos históricos (es uno de los sistemas más antiguos y extendidos) y por la robustez y la integración del cliente de Eclipse. En esta sección vamos a comprobar cómo funciona el sistema y cómo será el ciclo de trabajo habitual de desarrollo.

Un escenario común de uso de SVN en los equipos de desarrollo de software es el que se representa en la imagen anterior. El servidor SVN guarda la última versión y la historia del proyecto en desarrollo. Un desarrollador trabaja en su copia local hasta haber codificado (¡y probado!) algunos cambios en algunos ficheros del proyecto. Una vez se ha asegurado de que los cambios son correctos, se realiza un *commit* (confirmación) de los ficheros, subiéndolos al servidor. Allí se sustituyen los ficheros antiguos por los nuevos, anotándose la historia de cambios. Los otros desarrolladores que están trabajando simultáneamente también subirán sus cambios (normalmente, en ficheros diferentes). En la siguiente sesión de desarrollo, cada desarrollador debe realizar un *update* para actualizar sus proyectos locales antes de proceder de nuevo a su modificación. Si más de un desarrollador modifica el mismo fichero, SVN lo detecta y lanza un error, para que se solucionen manualmente los conflictos.

Los clientes SVN proporcionan un interfaz gráfico que simplifica la conexión al repositorio y la realización de todos estos *commit* y *update*. Eclipse contiene un cliente SVN que utilizaremos para subir todos los proyectos desarrollados.

El ciclo de trabajo recomendable para trabajar con SVN es el siguiente:

1. Realizar un *update* del proyecto para actualizar la versión local con los posibles cambios de los compañeros.
2. Trabajar en el código del proyecto.
3. Hacer un *update* para comprobar si algún cambio subido al repositorio entra en conflicto con lo que hemos desarrollado.
4. Finalmente hacer un *commit* para subir nuestros cambios al repositorio.

Cada alumno tiene disponible en el servidor del especialista (*server.jtech.ua.es*) un repositorio privado en el que puede mantener de forma remota los proyectos en los que está trabajando. El nombre del repositorio coincide con el login del alumno. El login es el mismo que con el que se accede al servidor de apuntes: iniciales del nombre + primer apellido, sin espacios y en minúscula. Por ejemplo, el login de Ana Isabel Pérez Carrascosa sería *aiperez*. La contraseña es el número del DNI, sin la letra de control.

Hay creados dos directorios en el repositorio de cada alumno. Uno para subir los proyectos de los módulos y otro para subir los del proyecto de integración:

- URL del repositorio SVN de los proyectos de los módulos:
`svn+ssh://server.jtech.ua.es/home/svn/pperez/modulos/trunk`
- URL del repositorio SVN del proyecto de integración:
`svn+ssh://server.jtech.ua.es/home/svn/pperez/proyint/trunk`

El directorio `trunk` es el directorio principal en el que se guardarán los proyectos que subamos al repositorio. Además, existe el convenio de crear los directorios `tags` y

branches en los que se pueden guardar versiones y ramas de los proyectos. Utilizaremos más adelante las *tags* para marcar las entregas de los módulos.

1.5.1. Cómo subir un proyecto al repositorio SVN

Eclipse permite publicar en SVN los proyectos de forma individual mediante un asistente que proporciona una interfaz a los comandos `svn` con los que se interactúa realmente con el repositorio. Vamos a probarlo. Comencemos guardando el proyecto `pperez-pres-agenda` creado en esta sesión.

Nota:

Antes de continuar, una breve nota sobre los nombres de los proyectos. Es necesario que todos los proyectos desarrollados en el especialista tengan nombres distintos, ya que todos van a residir en el mismo directorio del repositorio SVN. Para evitar coincidencias en el nombre de los proyectos, seguiremos la norma de que las primeras letras del proyecto será siempre el nombre corto del módulo en el que estamos trabajando (`pres` para el módulo *Presentación*, `jhd` para el módulo *Java y Herramientas de Desarrollo*, etc.). Además, para facilitar la corrección y permitir en algún momento la publicación junto con otros proyectos de distintos alumnos, precederemos el nombre del proyecto con el login.

Para subir el proyecto al repositorio SVN lo seleccionamos, pulsamos el botón derecho y escogemos la opción *Team > Share Project.. > SVN*.

Aparecerá una ventana en la que hay que proporcionar los datos del repositorio SVN. Estos datos se grabarán en el directorio de trabajo de Eclipse y no tendrás que introducirlos de nuevo. Si marcamos la opción para grabar el usuario y contraseña Eclipse pedirá adicionalmente una contraseña maestra para encriptar el login y contraseña introducidos.

New Repository Location

Enter Repository Location Information

Define the SVN repository location information. You can specify additional settings for proxy and svn+ssh, https connections.

General Advanced SSH Settings SSL Settings

URL: Browse...

Label

☒ Use the repository URL as the label

☐ Use a custom label:

Authentication

User:

Password:

☐ Save authentication (could trigger secure storage login)

To manage your security data, please see "[Secure Storage](#)"

Show Credentials For:

☒ Validate Repository Location on finish

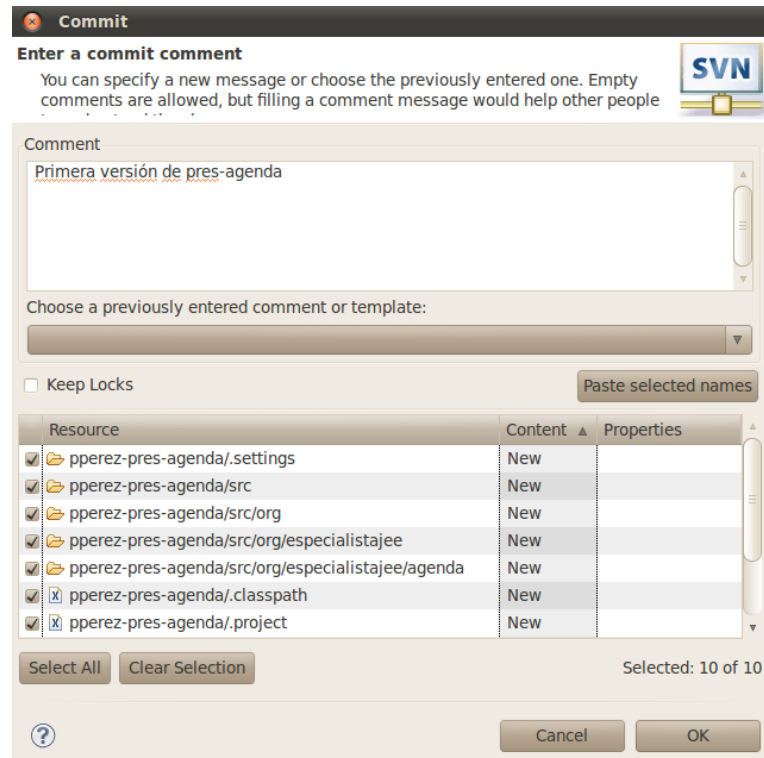
Reset Changes

Cancel Finish

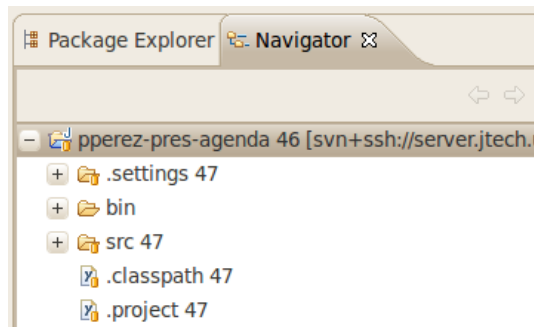
Eclipse detecta que el directorio `trunk` es un directorio especial (en él se guarda la rama principal del proyecto) y nos pide si queremos normalizar la URL. Aceptamos su sugerencia y decimos que sí.

A continuación debemos indicar qué directorios se van a crear en el repositorio SVN. El asistente nos pide la localización del proyecto en el repositorio. Debemos marcar la opción *Advanced Mode*, *Use project name* y *Use Repository Layout*. En el repositorio se creará bajo el directorio `trunk` un directorio con el mismo nombre que el mismo nombre que el proyecto:

Pulsando el botón *Next* nos pide un comentario inicial para la subida del proyecto. Podemos aceptar el que ofrece por defecto. Pulsando el botón *Next>* aparecerá la siguiente pantalla en la que podrás revisar qué ficheros se van a guardar en el directorio recién creado. No debería aparecer el directorio `bin` ni ningún fichero compilado `.class`. Son ficheros que no deben residir en el repositorio común, porque se generan cada vez en el espacio de trabajo local.



Repasamos que todo está correcto, ponemos el comentario indicado y pulsamos *OK*. Una vez guardados los ficheros en el repositorio, el proyecto quedará enlazado al repositorio SVN. En el panel de proyectos de Eclipse aparecerá el proyecto con una etiqueta indicando la URL del repositorio y un icono en los ficheros y directorios que están enlazados con el repositorio:

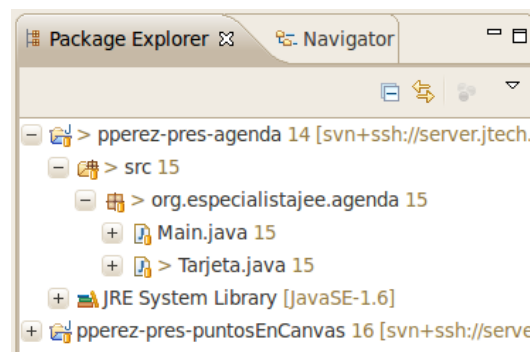


Nota:

La conexión entre los proyectos locales y repositorio SVN queda grabada en el disco duro en forma de unos directorios llamados `.SVN` que se crean en los directorios enlazados. Los datos del repositorio SVN quedan también grabados en el directorio local del proyecto. De esta forma, al copiar el proyecto también se copia la conexión con el repositorio SVN.

1.5.2. Cómo subir al repositorio cambios en el proyecto local

Una vez creado o descargado un repositorio en un proyecto haremos modificaciones en el proyecto local: modificaremos el contenido de algún fichero, crearemos nuevos ficheros y directorios o borraremos ficheros existentes. Eclipse marca todos aquellos recursos que se han modificado con un símbolo especial (">"). Para comprobarlo, cambiamos el método `toString` de la clase `Tarjeta`:



Para subir los cambios, seleccionamos el directorio raíz del proyecto y escogemos la opción *Team > Commit...*. Al igual que cuando hemos subido el proyecto por primera vez, aparecerá la ventana *Commit* en la que podrás escribir un comentario sobre los cambios realizados. Una vez que aceptamos el commit, Eclipse se conecta con el repositorio y actualiza los cambios. Veremos que desaparece el signo '>' que marcaba las diferencias con el repositorio.

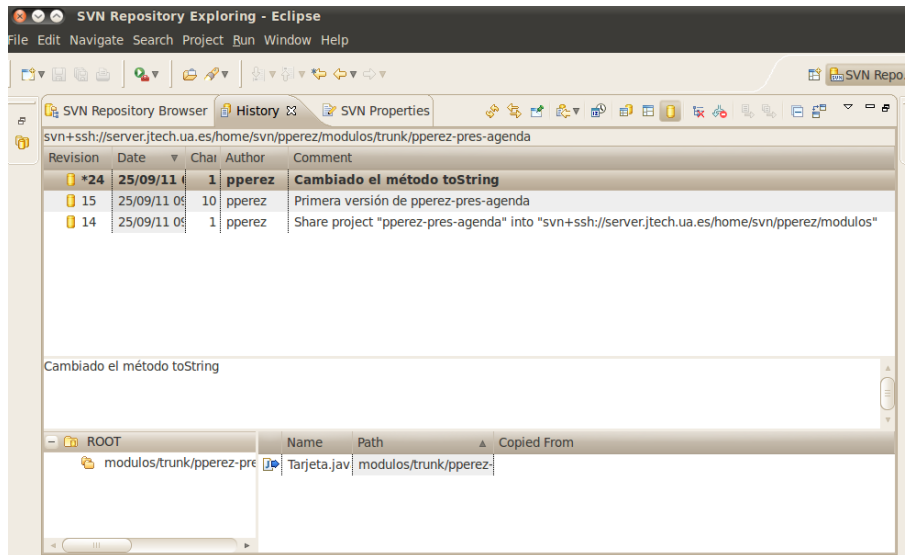
Aviso:

Al final de cada sesión de ejercicios debes subir los proyectos con los que has estado trabajando al repositorio SVN. También lo puedes hacer si trabajas en casa y tienes conexión a Internet. De esta forma, el repositorio SVN estará siempre actualizado, te será más cómodo realizar la entrega de los ejercicios y siempre tendrás una copia de seguridad, aparte de la propia copia de la máquina virtual.

1.5.3. Cómo explorar versiones de un fichero

Una vez que se han subido distintas versiones de un mismo fichero es posible consultar su historia de cambios y comparar distintas versiones subidas al repositorio SVN.

Para ver la historia de cambios de un fichero o un directorio, lo seleccionamos con el botón derecho y pulsamos la opción *Team > Show History*. Aparece el panel *History* en la parte inferior de la ventana de Eclipse. Si lo hacemos con todo el proyecto `pperez-pres-agenda` debe aparecer algo como esto:



Este panel es bastante interesante y tiene muchas funcionalidades. No tenemos demasiado tiempo para probarlas, pero sería interesante que las investigaras por tu propia cuenta. Investiga la historia de algún fichero concreto (por ejemplo, `Tarjeta.java`) y prueba a hacer lo siguiente:

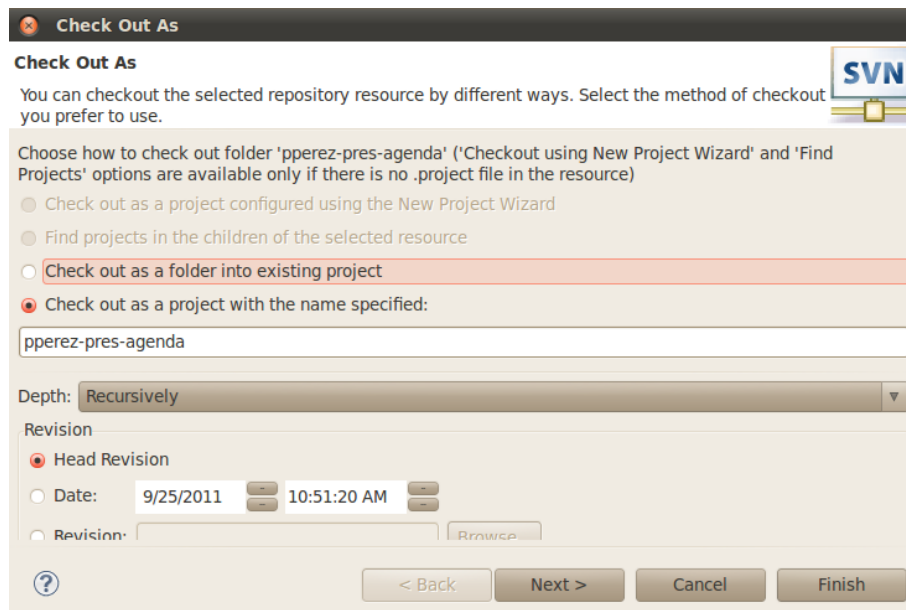
- Haciendo un doble click en cualquiera de las versiones se abre en el panel del editor su contenido.
- Es posible comparar distintas versiones del documento pulsando el botón derecho sobre la versión que queremos comparar y escogiendo la opción *Compare Current with 1.X*.

1.5.4. Cómo bajar proyectos del repositorio SVN

Supongamos que la máquina virtual se ha estropeado y que queremos recuperar algún proyecto subido al servidor. ¿Cómo lo hacemos?. Empezamos borrando el proyecto `pperez-pres-agenda` del espacio de trabajo. Para ello seleccionamos el proyecto y escogemos con el botón derecho la opción *Delete*. Activamos también la opción *Delete project contents on disk* para que borre físicamente el directorio. Si no lo hiciéramos así, Eclipse borraría el proyecto del espacio de trabajo, pero no del disco duro.

Una vez borrado, vamos a recuperarlo del repositorio. Para ello seleccionamos con el botón derecho sobre la vista de proyectos la opción *Import > Import... > SVN > Project from SVN...*. Seleccionamos la localización del repositorio SVN en la que está el proyecto y *Next*.

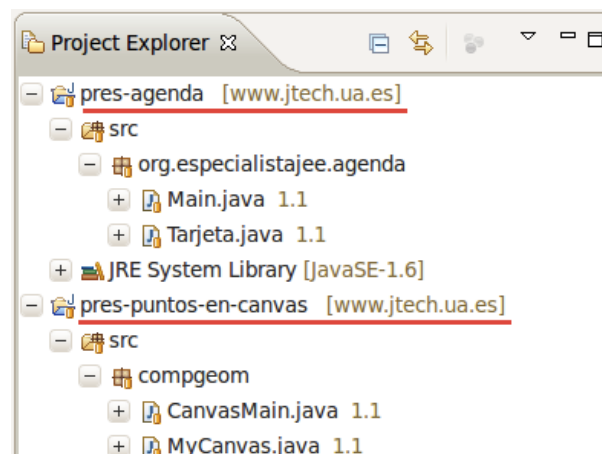
Pinchamos ahora en el botón *Browse* para explorar el repositorio y seleccionar el módulo que queremos descargar (`modulos > trunk > pperez-pres-agenda`). Pulsamos *OK* y *Finish* y aparecerá una última pantalla en la que Eclipse nos pregunta cómo debe descargar el proyecto. Escogemos que lo importe como un proyecto con el nombre indicado:



Podríamos incluso descargar versiones anteriores del proyecto. Pulsamos *Finish* y el proyecto volverá a aparecer en el espacio de trabajo.

1.6. Como realizar la entrega de los ejercicios

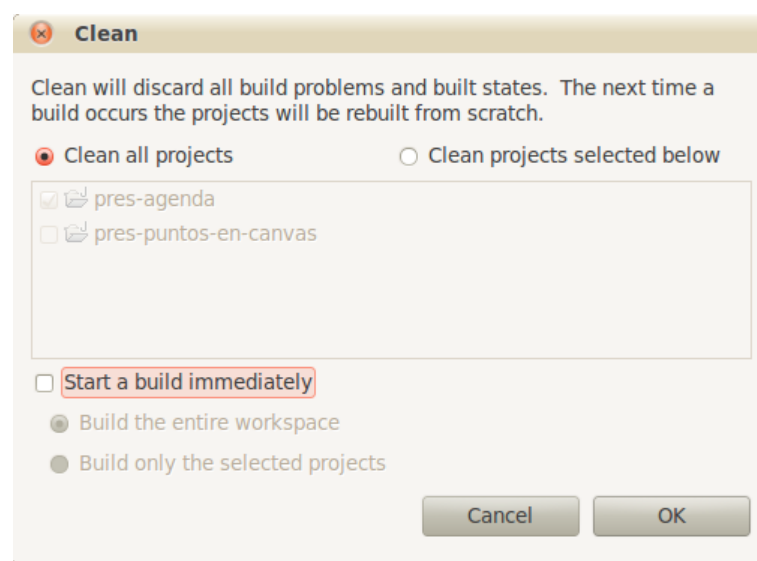
Vamos a terminar realizando una entrega de los ejercicios. Antes sube el proyecto *pres-puntos-en-canvas* a tu repositorio SVN. En el panel de proyectos deben aparecer los dos proyectos conectados al repositorio.



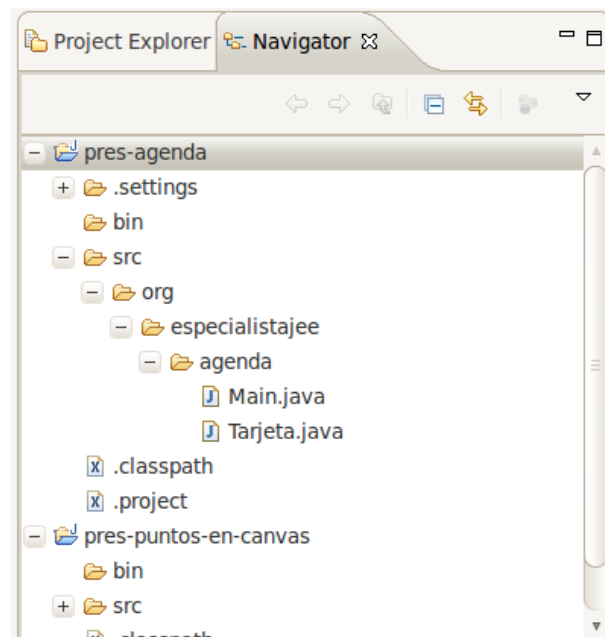
Supongamos que ya hemos terminado todos los proyectos del módulo y que tenemos que entregar los ejercicios. Tenemos que comprimir todos los proyectos en un único fichero que es el que subiremos a Moodle. Vamos a verlo paso a paso.

1. En primer lugar, tenemos que **desconectar los proyectos** locales del repositorio SVN con la opción *Team > Disconnect...*. Los proyectos seguirán estando en el repositorio SVN (y podremos recuperarlos de allí cuando queramos). Al desconectar los proyectos debemos borrar también la información SVN. De esta forma sólo entregaremos el código fuente de los proyectos y no los datos de la conexión SVN.

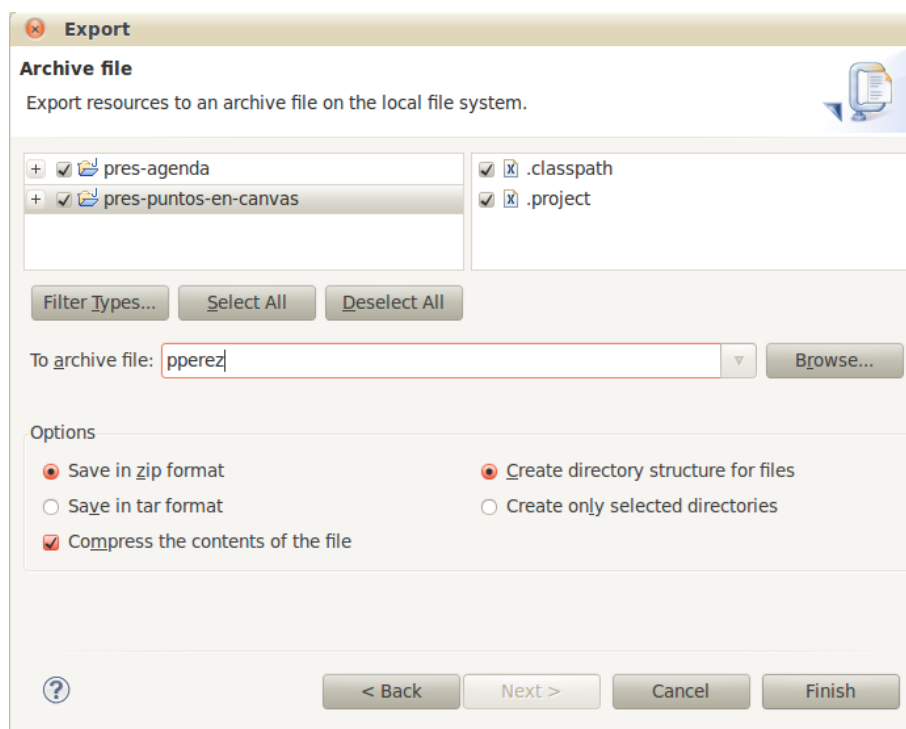
2. Después debemos **limpiar los proyectos**, eliminando todo el código compilado (ficheros `.class`) y dejando sólo el código fuente. Para ello desactivamos la opción de construcción automática *Project > Build Automatically* y seleccionamos la opción de limpiar: *Project > Clean...* :



Con la vista *Navigator* comprobamos que los directorios `bin` en los que Eclipse deja los ficheros `.class` están vacíos.

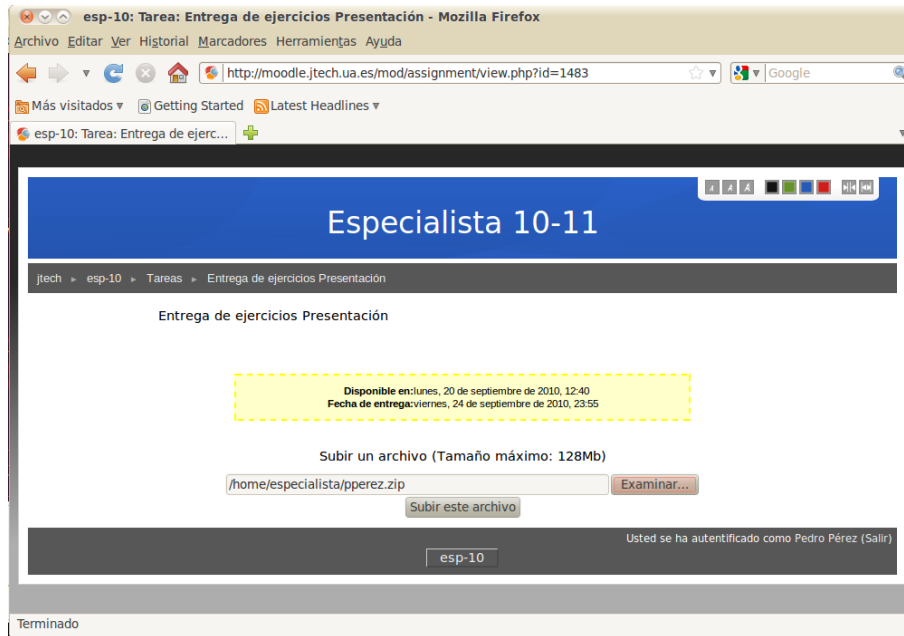


3. Por último exportamos los proyectos a un fichero .ZIP utilizando la opción *Export > General > Archive File* y poniendo como nombre de fichero **tu nombre de usuario**:



Y sólo falta entregar en Moodle, en la tarea correspondiente al módulo que se ha

terminado, el fichero `pperez.zip` generado:



1.7. Ciclo de trabajo

El ciclo de trabajo de una sesión de ejercicios será el siguiente. Supongamos que estamos en el módulo *Java y Herramientas de Desarrollo* (nombre corto: *jhd*):

1. **Crear el espacio de trabajo.** Si es la primera sesión de ejercicios, deberás crear el espacio de trabajo `jhd`
2. **Crear o importar los proyectos.** Crea los proyectos que se indiquen en la sesión de ejercicios y trabaja con ellos. Todos los nombres de proyectos tendrán como prefijo `jhd-`. Algunas veces se proporcionará una versión inicial de los proyectos en un fichero ZIP.
3. **Publicar en SVN.** Sube los proyectos que has desarrollado a tu repositorio SVN privado. Asegúrate que al final de la sesión y antes de cerrar Eclipse, todos los proyectos estén sincronizados.
4. **Entrega de ejercicios.** Limpia y exporta todos los proyectos en formato ZIP usando tu nombre de usuario (`pperez.zip`). Entrega el fichero en la tarea correspondiente de Moodle.

Nota:

Una idea fundamental del ciclo de trabajo es la de las **integraciones continuas** (se explicará en el módulo *Metodologías de desarrollo*): al final de cada sesión el espacio de trabajo debería estar completo y listo para entregar.

1.8. Entrega de ejercicios

Exporta los proyectos del módulo a un único fichero `login.zip`, donde "login" es tu identificador de usuario en el SVN. Entrégalo en la tarea de Moodle que está en el curso *Materiales comunes y encuestas*.

2. Guía de laboratorio - Ejercicios

2.1. Proyecto Java

Descarga las plantillas de la sesión. En ellas tenemos un proyecto llamado `PuntosEnCanvas`. Impórtalo desde Eclipse y ejecútalo como aplicación Java.

Podrás hacer click con cualquier botón sobre un canvas en el que aparecerán puntos. ¿Cómo podríamos modificar el código para que con el botón izquierdo aparezca un punto, y con el botón derecho se elimine? ¿Podría ser implementando un método `mouseRightClicked()` o similar? Para saberlo, utiliza el menú "Source/Override or implement methods" de Eclipse, para ver los métodos que puedes sobrecargar o implementar. Si no encuentras el método adecuado, busca la forma de saber qué botón se ha pulsado desde dentro del propio método `mouseClicked()` que tenemos implementado.

2.2. Subversion

Cambia el nombre del proyecto del ejercicio anterior a `intro-PuntosEnCanvas` para saber que es un ejercicio perteneciente a la sesión de introducción. Súbelo al Subversion utilizando el plugin de Eclipse, como se indica en los apuntes. La ruta sería la siguiente:

```
svn+ssh://server.jtech.ua.es/home/svn/pperez/modulos/trunk
```

Ahora realiza algún cambio y antes de hacer Commit observa las diferencias entre tu archivo local y el archivo del repositorio, con la vista del Diff Editor.

Sube ahora el cambio que has realizado y a continuación elimina tu proyecto local.

Abre la vista SVN Repository Exploring y localiza tu proyecto. ¿Está el código compilado también subido al repositorio? Si es así, elimina del repositorio la carpeta que contiene código compilado: la carpeta `bin`. Ahora abre las preferencias generales de Eclipse y en "Team/Ignored resources" añade el patrón `bin`.

Finalmente importa el proyecto desde el repositorio a tu workspace local. En la vista Navigator de Eclipse observarás que la carpeta `bin` ha vuelto a ser generada por Eclipse, pero esta vez si realizas algún Commit la carpeta no subirá a Subversion.

2.3. Proyecto JavaME

En las plantillas de la sesión hay dos juegos JavaME: el Tapper y el Cochedrilo. Impórtalos en tu workspace de Eclipse y ejecútalos en distintos emuladores JavaME.

Utiliza la opción Refactor de Eclipse para cambiar el nombre de los paquetes del proyecto Tapper. En lugar de que empiecen por `es.ua.jtech.ajdm.tapper`, que empiecen por `es.ua.jtech.intro.tapper`. Comprueba que el cambio es coherente con el código y que el proyecto sigue ejecutándose correctamente.

2.4. Emuladores de Android

Configura dos emuladores de Android con versiones diferentes y tamaño de pantalla diferentes. Arranca los dos. ¿En qué puerto está cada uno de ellos? Envía un sms de uno al otro utilizando dicho puerto como si fuera un número de teléfono.

Abre un navegador web en el emulador. Si intentaras acceder a `http://localhost/`, ¿estarías intentando acceder al puerto 80 del ordenador (el host), o del emulador? Necesitarás acceder al puerto 80 del ordenador si tienes un servidor web en marcha en ese puerto. Busca en internet qué IP podrías utilizar para acceder al ordenador host en el que se ejecuta tu emulador.

2.5. Proyecto Android

En las plantillas de la sesión hay un proyecto Android llamado `EntityCloud`. Impórtalo en tu workspace y ejecútalo.

La aplicación hace uso del portapapeles del teléfono. Prueba a abrir una página web de noticias (desde el emulador) y seleccionar y copiar texto a la aplicación. No podrás copiar texto desde fuera del emulador.

Si tienes a mano un terminal Android con un cable USB, conéctalo y comprueba que a la hora de ejecutar el proyecto, te deja elegir entre lanzarlo en los emuladores o en tu móvil. Para ello tendrás que tener activada la depuración USB de tu terminal Android.

2.6. Entrega de ejercicios

Desde Eclipse selecciona el proyecto Java del primer ejercicio y el proyecto Android del anterior ejercicio y expórtalos a un archivo Zip que tendrás que llamar `intro-login.zip`, donde el login tiene que ser el mismo que tu identificador en el Subversion.

Entrega a través de Moodle ese archivo Zip.

2.7. Proyecto iOS

Crea un nuevo proyecto iOS. En el asistente de creación que sea multiplataforma y ejecútalo en un iPhone y en un iPad.