

Prueba Periodo de Recuperación SVN

1. Crea un repositorio en sourceforge y dame acceso.

Entramos en sourceforge y creamos el repositorio examenSVN-JavierCampos

Una vez creado entramos en la pestaña admin y elegimos la opción user permission, le damos a add y añadimos a joadelvia

2. En un directorio de la misma máquina, crea un directorio que representará un proyecto de desarrollo y añade los ficheros que te ha proporcionado el profesor en el fichero ListaOrdenadaGenerica.zip

En el escritorio con el comando mkdir examenSVN creamos el directorio y para añadir los ficheros utilizamos el siguiente comando

```
javier@javier-GL65-9SEK:~/Descargas$ cp -r ListaOrdenadaGenerica  
../Escritorio/examenSVN/
```

3. Importa ese proyecto que acabas de crear en el punto 2 al repositorio que creaste en el punto 1. Procura que el repositorio tenga la estructura recomendada en svn.

Primero creamos una copia del repositorio con el comando.

```
javier@javier-GL65-9SEK:~/Escritorio$ svn checkout --username=javier-campos  
https://svn.code.sf.net/p/examensvn-javiercampos/code/ examenSVN-JavierCampos
```

Creamos la estructura recomendada con el comando : mkdir trunk branches tags

Añadimos las carpetas con svn add * y posteriormente hacemos svn commit -m "Añadida estructura inicial

importamos el proyecto que acabamos de crear con el comando: svn import examenSVN/
https://svn.code.sf.net/p/examensvn-javiercampos/code/trunk -m "importado el primer proyecto"

4. Obtén una copia de trabajo dentro del directorio listaOrdenada.

```
javier@javier-GL65-9SEK:~/Escritorio$ svn co https://svn.code.sf.net/p/examensvn-  
javiercampos/code/ listaOrdenada
```

5. Modifica el método compareTo de la clase Persona (líneas 89 a 93 del fichero Persona.java) y cámbialo por este:

```
@Override  
public int compareTo(Persona o) {  
    return Integer.compare(edad, o.edad)  
}
```

Entramos en el sublime y modificamos lo que nos pide

6. Añade la clase Contenedor.java que te ha proporcionado el profesor.

Lo copiamos desde descargas con el comando: `javier@javier-GL65-9SEK:~/Descargas$
svn cp Contenedor.java
../Escritorio/listaOrdenada/trunk/ListaOrdenadaGenerica/src/prueba`

7. Elimina los ficheros compilados (todos los ficheros dentro del directorio bin/prueba, son los ficheros .class)

Entramos al fichero y ejecutamos el comando: `svn rm ListaOrdenada.class Persona.class Principal.class`

8. Mira el estado de la copia de trabajo y comprueba los códigos asociados a los archivos modificados.

```
javier@javier-GL65-9SEK:~/Escritorio/listaOrdenada$ svn status
D   trunk/ListaOrdenadaGenerica/bin/prueba/ListaOrdenada.class
D   trunk/ListaOrdenadaGenerica/bin/prueba/Persona.class
D   trunk/ListaOrdenadaGenerica/bin/prueba/Principal.class
A   trunk/ListaOrdenadaGenerica/src/prueba/Contenedor.java
M   trunk/ListaOrdenadaGenerica/src/prueba/Persona.java
```

9. Añade al control de versiones los archivos nuevos y elimina los archivos borrados. Crea una nueva revisión con los cambios realizados.

```
javier@javier-GL65-9SEK:~/Escritorio/listaOrdenada$ svn commit -m "Modificado proyecto"
Eliminando   trunk/ListaOrdenadaGenerica/bin/prueba/ListaOrdenada.class
Eliminando   trunk/ListaOrdenadaGenerica/bin/prueba/Persona.class
Eliminando   trunk/ListaOrdenadaGenerica/bin/prueba/Principal.class
Añadiendo    trunk/ListaOrdenadaGenerica/src/prueba/Contenedor.java
Enviando     trunk/ListaOrdenadaGenerica/src/prueba/Persona.java
Transmitiendo contenido de archivos ..done
Committing transaction...
Commit de la revisión 3.
```

10. Mira en qué revisión se encuentra la copia de trabajo y mira todos los registros.

Con el paso anterior podemos ver la revision en la que nos encontramos pero para asegurarnos ejecutamos el comando `svn status -u`.

Para ver todos los registros hacemos `svn log`

11. Queremos que la clase Persona vuelva al comparador anterior.

```
javier@javier-GL65-9SEK:~/Escritorio/listaOrdenada$ svn cp https://svn.code.sf.net/p/examensvn-javiercampos/code/trunk/ListaOrdenadaGenerica/bin/prueba/Persona.class@2 Persona.class
```

Yo directamente lo he recuperado tanto en local como en el repositorio

12.Manda una nueva revisión al repositorio.

```
svn commit -m "recuperado el fichero Persona.class"
```

13.Repite el punto 10. ¿Qué ha ocurrido?.

Hacemos `svn status -u` y vemos que estamos en la revisión 4

hacemos `svn log` y vemos todos los commits que hemos hecho

14.Actualiza la copia de trabajo y repite el punto 10. ¿Qué ha ocurrido?.

Que seguimos en la revision 4

15.Recupera uno de los ficheros compilados que eliminaste en el punto 7, asegúrate que se recupere también en el repositorio.

```
javier@javier-GL65-9SEK:~/Escritorio/listaOrdenada$ svn cp  
https://svn.code.sf.net/p/examensvn-javiercampos/code/trunk/ListaOrdenadaGenerica/bin/  
prueba/Principal.class@2 Principal.class  
A Principal.class
```

Recuperamos el fichero Principal.class

16.Crea una rama privada de desarrollo sobre el directorio `branches` del proyecto. Llámala `canelaEnRama`.

```
javier@javier-GL65-9SEK:~/Escritorio/listaOrdenada$ svn copy  
https://svn.code.sf.net/p/examensvn-javiercampos/code/trunk https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama
```

Ademas hacemos un commit que le indicamos que creamos la rama canelaEnRama

17.Haz una copia de trabajo de dicha rama en el directorio `canela`.

```
svn co https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama canela
```

18.Usa `svn info` para tener claro que has hecho una copia de trabajo de la rama y pon una captura con el resultado.

Aqui vemos la raiz del repositorio que es esa

```
A canela/ListaOrdenadaGenerica/src/prueba/Principal.java  
A canela/ListaOrdenadaGenerica/.classpath  
A canela/ListaOrdenadaGenerica/.project  
Revisión obtenida: 5  
javier@javier-GL65-9SEK:~/Escritorio$ ^C  
javier@javier-GL65-9SEK:~/Escritorio$ cd canela  
javier@javier-GL65-9SEK:~/Escritorio/canela$ svn info  
Ruta: .  
Working Copy Root Path: /home/javier/Escritorio/canela  
URL: https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama  
Relative URL: ^/branches/canelaEnRama  
Raíz del repositorio: https://svn.code.sf.net/p/examensvn-javiercampos/code  
UUID del repositorio: df3dcaad-9a64-454b-a494-a70106037b00  
Revisión: 5  
Tipo de nodo: directorio  
Agendado: normal  
Autor del último cambio: javier-campos  
Revisión del último cambio: 5  
Fecha de último cambio: 2021-02-01 13:34:56 +0100 (lun 01 de feb de 2021)  
javier@javier-GL65-9SEK:~/Escritorio/canela$
```

19. Realiza una sesión de trabajo en la rama que has creado siguiendo los siguientes pasos:

1. Actualiza tu copia de trabajo.

Svn update

2. Modifica el método compareTo de la clase Persona como hicimos antes. Añade algún fichero, por ejemplo un Readme.md en la raíz del proyecto.

Entramos en el sublime y modificamos el método compareTo le ponemos edad1
añadimos un fichero vacío con touch ficheroPrueba.txt

3. En la rama trunk añade el atributo color de tipo String a la clase Contenedor. Para ello añade debajo de la línea 7 la siguiente línea:

```
private String color;
```

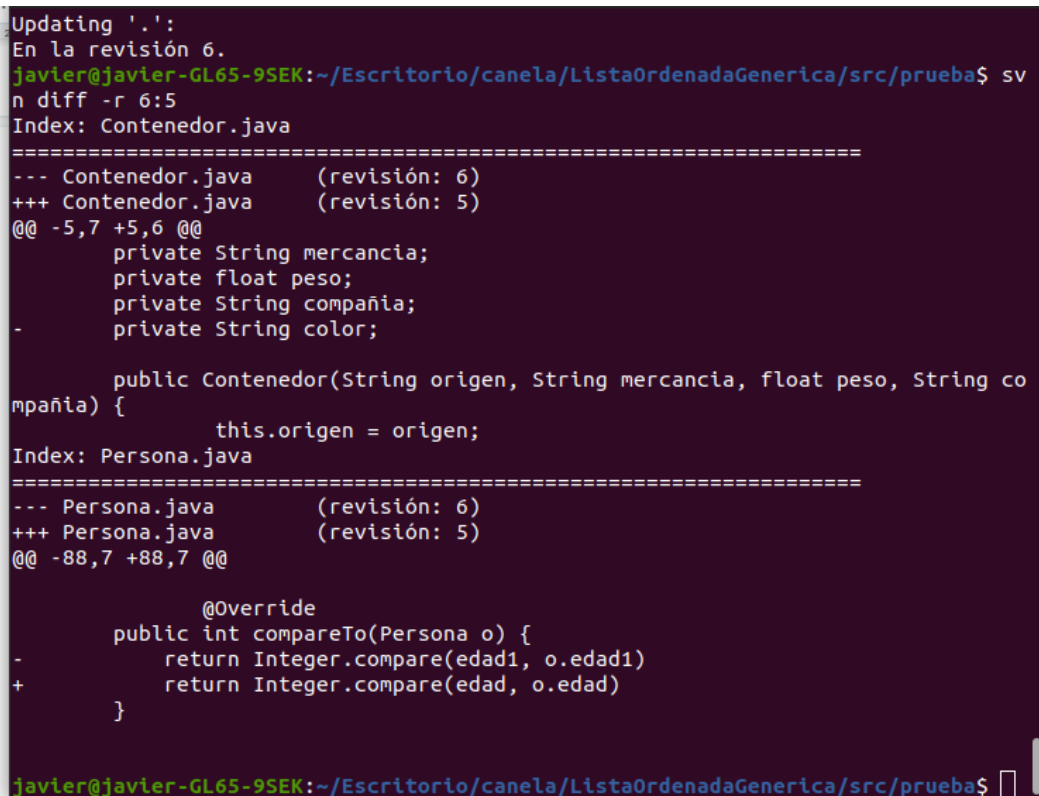
Sube los cambios al repositorio.

Entramos al sublime y añadimos lo que nos dice

```
svn commit -m "Modificado Persona.java y Contenedor.java"
```

4. En la rama canelaEnRama consulta el estado de tu copia de trabajo respecto a la última revisión que actualizaste y respecto a la última revisión que haya en el repositorio. Indica las diferencias y pon una captura de cada una de ellas.

Con svn status vemos los cambios y si hacemos svn diff 6:5 vemos las diferencias entre las últimas versiones actualizadas



```
Updating '.':
En la revisión 6.
javier@javier-GL65-9SEK:~/Escritorio/canela/ListaOrdenadaGenerica/src/prueba$ svn diff -r 6:5
Index: Contenedor.java
=====
--- Contenedor.java      (revisión: 6)
+++ Contenedor.java      (revisión: 5)
@@ -5,7 +5,6 @@
     private String mercancia;
     private float peso;
     private String compañía;
-    private String color;

     public Contenedor(String origen, String mercancia, float peso, String co
mpañía) {
         this.origen = origen;
Index: Persona.java
=====
--- Persona.java          (revisión: 6)
+++ Persona.java          (revisión: 5)
@@ -88,7 +88,7 @@
         @Override
         public int compareTo(Persona o) {
-            return Integer.compare(edad1, o.edad1)
+            return Integer.compare(edad, o.edad)
         }
     }

javier@javier-GL65-9SEK:~/Escritorio/canela/ListaOrdenadaGenerica/src/prueba$
```

17. Usa svn info para tener claro que has hecho una copia de trabajo de la rama.

5. Revisa los cambios realizados en los ficheros que has tocado respecto a la última revisión que actualizaste y respecto de la última revisión que haya en el repositorio. Para esto usamos el comando `svn diff 6:5` para ver las diferencias de actualización con la última revisión actualizada. Aquí vemos los cambios que hemos realizado en los ficheros `Persona` y `Contenedor`

6. Incorpora los cambios de la rama `trunk` a la rama `canelaEnRama`.

Lo hacemos con el comando `svn merge` <https://svn.code.sf.net/p/examensvn-javiercampos/code/trunk>

me sale un conflicto, así que tengo que resolverlo, elegimos la opción en la que tenemos el `private String color`;

nos sale la opción `r` para marcarlo como resuelto

7. Envía los cambios al repositorio creando una nueva revisión (*commit*).

`svn commit -m "Actualizada rama y resuelto conflicto"`

20. En la rama `trunk` modifica el método `compareTo` para que compare por nombre. Cambia las líneas 89 a 92 por estas:

```
@Override
public int compareTo(Persona o) {
    return Integer.compare(nombre, o.nombre);
}
```

Entramos al sublime desde el archivo `persona` de `trunk` y modificamos lo que nos pide.

21. Reintegra tu rama privada con el desarrollo principal del proyecto (`trunk`). El método `compareTo` de la clase `Persona` debe quedar comparando por edad.

Primero tenemos que subir los cambios anteriores al repositorio, para ello hacemos un `commit`, hacemos también `svn update` para asegurarnos que está todo actualizado y por último fusionamos la rama.

Para fusionar la rama tenemos que estar situados en `trunk`, posteriormente ejecutamos el comando: `svn merge --reintegrate` <https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama>

Efectivamente se ha fusionado la rama:

```
javier@javier-GL65-9SEK:~/Escritorio/examenSVN-JavierCampos/trunk$ svn merge --reintegrate
https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama
```

--- Fusionando las diferencias entre URLs del repositorio en '':

U ListaOrdenadaGenerica/src/prueba/Contenedor.java

A ListaOrdenadaGenerica/src/prueba/ficheroPrueba.txt

--- Recording mergeinfo for merge between repository URLs into '':

U

mp no ha surgido el conflicto porque nos surgió en el paso anterior y ya lo solucioné, ya que por medio me equivoque y modifiqué cosas en ficheros que no eran, pero resolví el conflincto eligiendo la opcion e de editar y posteriormente le di a la opcion r para marcar como resuelto.

Una vez hecho esto hacemos el commit.

Una vez fusionada la rama lo lógico sería eliminarla, ya que no vamos a poder volver a utilizarla, como no lo pide no la voy a borrar pero si voy a dejar indicado cómo sería.

svn delete <https://svn.code.sf.net/p/examensvn-javiercampos/code/branches/canelaEnRama> .