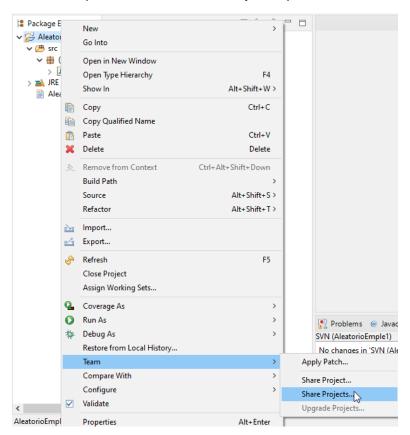
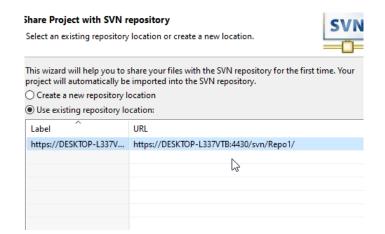
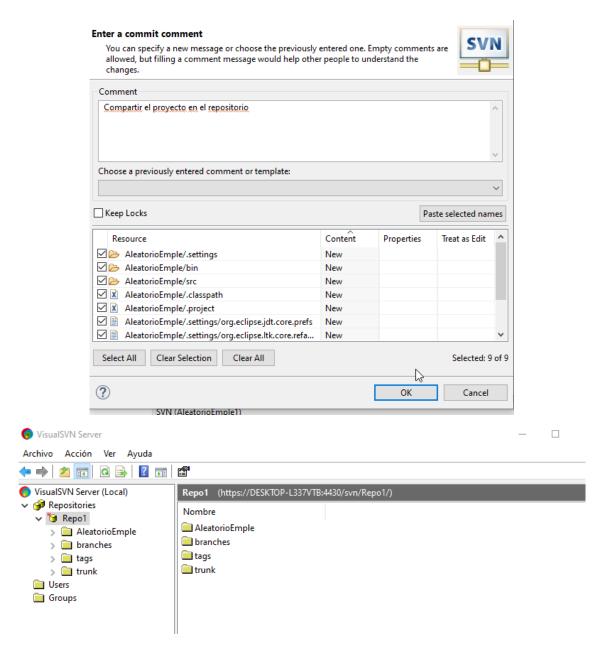
## Ejercicios Tema 4.2

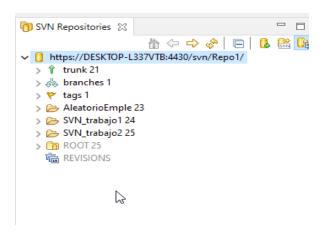
Empezamos subiendo el trabajo al repositorio:

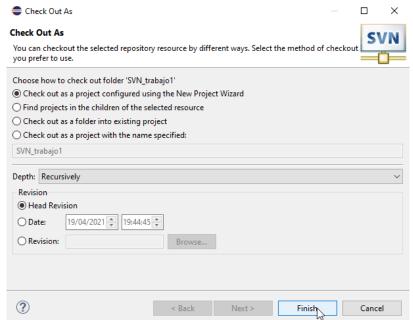


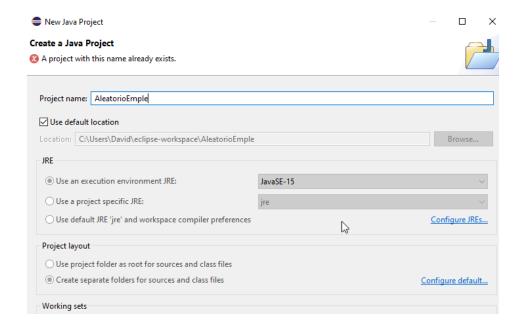




Con esto tenemos el proyecto dentro del repositorio. Vamos a crear la carpeta de trabajo SVN\_trabajo1, y le vamos a hacer un checkout para tener dos copias extras del proyecto AleatorioEmple:







```
✓ 

AleatorioEmple 22 [https://DESKTOP-L337VTB:4430/svn/Repo1/: AleatorioEmple]

  default package) 23
       Aleatorio Emple. java 23
  > M JRE System Library [jre]
    AleatorioEmple.dat 23

    AleatorioEmple1 24 [https://DESKTOP-L337VTB:4430/svn/Repo1/: SVN_trabajo1]

  JRE System Library [JavaSE-15]
  > AleatorioEmple.java 26
     > A module-info.java 26

✓ ReatorioEmple2 26 [https://DESKTOP-L337VTB:4430/svn/Repo1/: SVN_trabajo1]

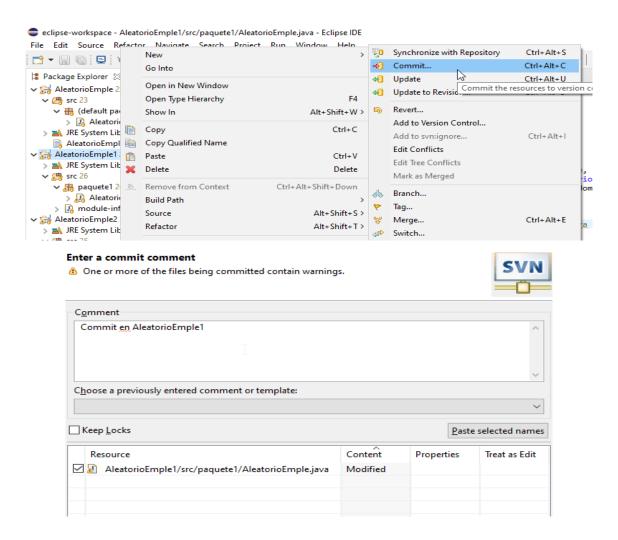
  > A JRE System Library [JavaSE-15]
  > 📠 paquete1 26
     > R module-info.java 26
> 👍 > Repo1 23 [https://DESKTOP-L337VTB:4430/svn/Repo1/: Repo1/s
```

Ahora vamos a realizar unas modificaciones en cada archivo .java de cada una de las copias de trabajo, para provocar un conflicto.

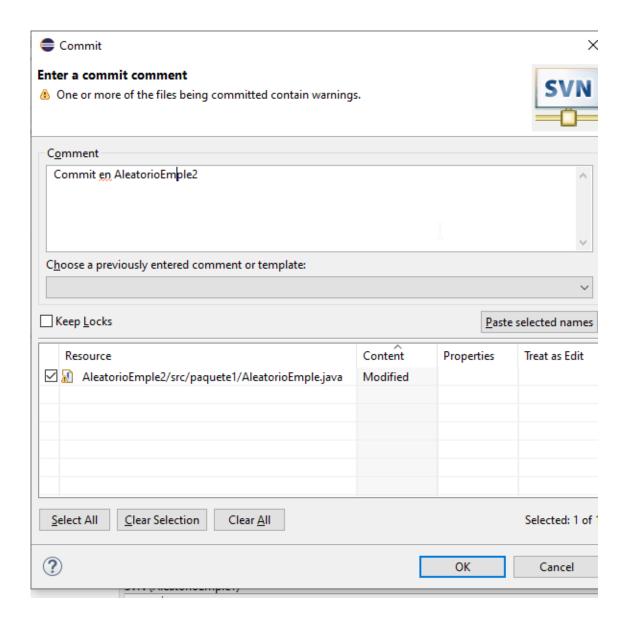
```
🛮 AleatorioEmple.java 💢
  package paquete1;
  3⊕ import java.io.File;
 8 public class AleatorioEmple {
 10 public static RandomAccessFile file;
 12@ public static void consultar(int num, double importe) throws IOException {
        File fichero = new File("AleatorioEmple.dat");
13
        RandomAccessFile file = new RandomAccessFile(fichero, "rw");
 15
        double salario, nuevosalario;
 16
        int pos,dep;
 17
        char aux;
 int dato = 0; // Declaración para AleatorioEmple1
        String nom="";
 19
        // 36 es la longitud del registro
 20
        if (file.length() <=0)
 21
             System.out.println(" EL FICHERO ESTÁ VACÍO.");
 22
 23
        else
        if (file.length()<(num*36))</pre>
 24
 25
         System.out.println(" EL NUMERO DE EMPLEADO NO EXISTE.");
 26
         else
```

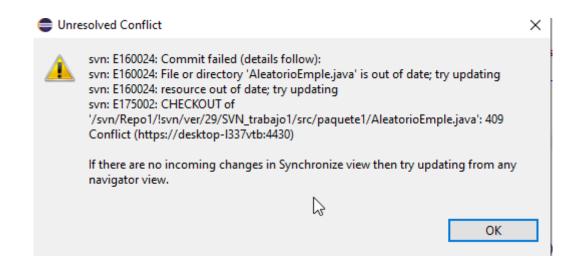
```
import java.io.File;
public class AleatorioEmple {
public static RandomAccessFile file;
public static void consultar(int num, double importe) throws IOException {
    File fichero = new File("AleatorioEmple.dat");
    RandomAccessFile file = new RandomAccessFile(fichero, "rw");
    double salario, nuevosalario;
    int pos,dep;
    char aux;
    String cad = "HOLA"; // Declaración en AleatorioEmple2
String nom="";
    // 36 es la longitud del registro
    if (file.length() <=0)</pre>
         System.out.println(" EL FICHERO ESTÁ VACÍO.");
    else
    if (file.length()<(num*36))</pre>
     System.out.println(" EL NUMERO DE EMPLEADO NO EXISTE.");
     else
      {
        pos=(num-1)*36; file.seek(pos);
        num=file.readInt(): // obtengo el den
```

Una vez realizadas las modificaciones, vamos a validar el archivo de AleatorioEmple1.

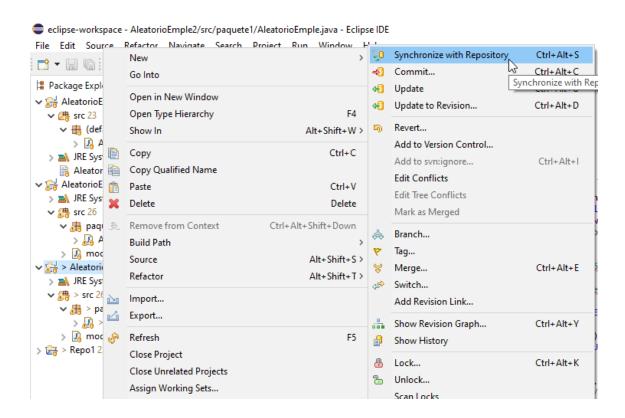


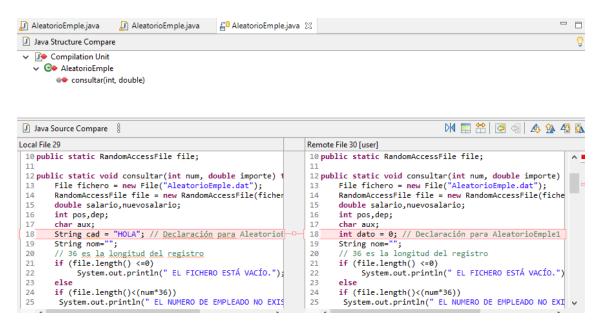
Después vamos a intentar validar Aleatorio Emple2, pero nos va a dar error porque hay un conflicto.



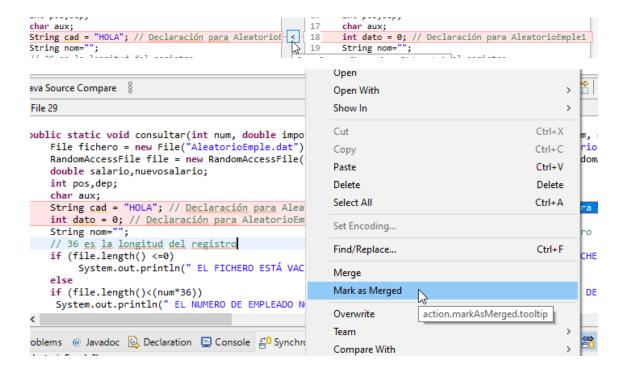


Para solucionar el problema, vamos a sincronizar Aleatorio Emple2 con el repositorio:

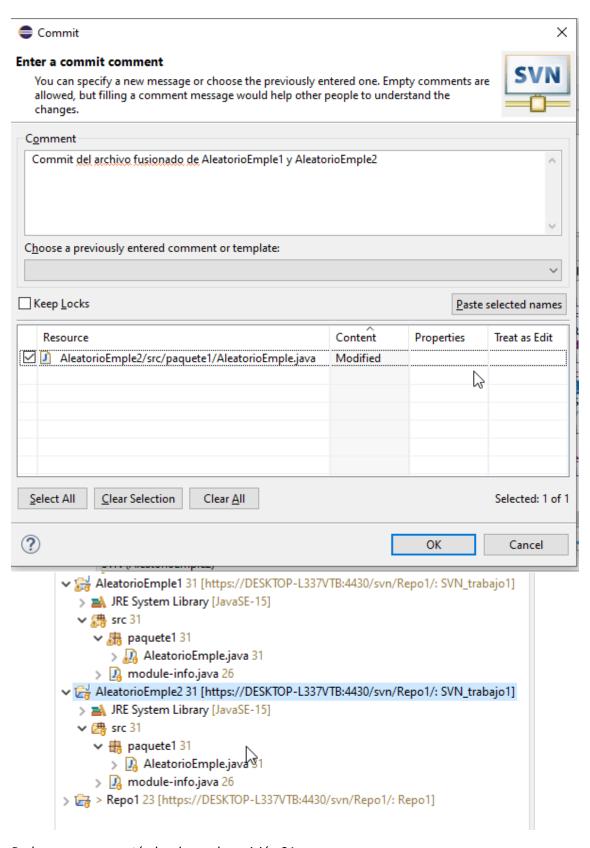




En la línea roja podemos ver el conflicto. Para solucionarlo, hacemos clic en el cuadro rojo que hay en el centro. Luego habrá que marcar el archivo como fusionado para que no vuelva a ocurrir el conflicto.



Después, podemos validar Aleatorio Emple2, y actualizar las dos copias de trabajo para que estén en la misma versión.



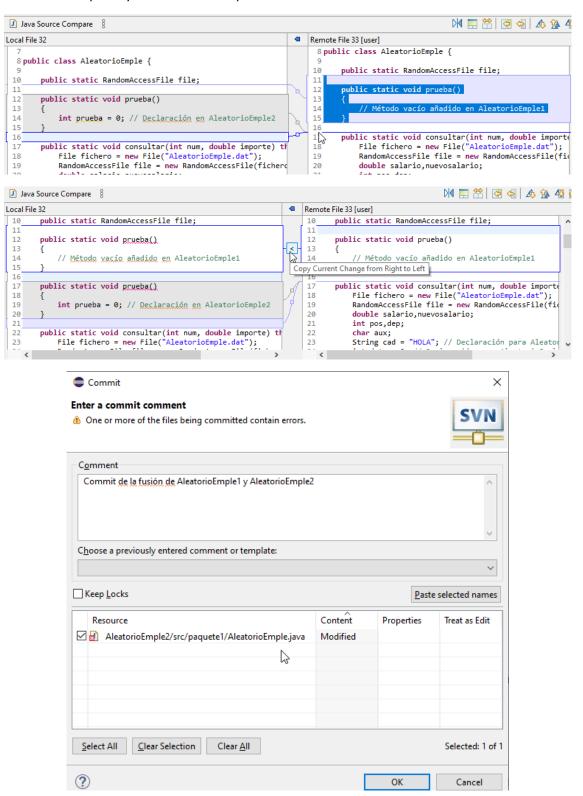
Podemos ver que están las dos en la revisión 31.

Ahora vamos a crear un método en Aleatorio Emple 1, vamos a validarlo, y vamos a crear un método diferente en Aleatorio Emple 2.

```
package paquetel;
        import java.io.File;
        import java.io.IOException;
        import java.io.RandomAccessFile;
        public class AleatorioEmple {
            public static RandomAccessFile file;
            public static void prueba()
                // Método vacío añadido en AleatorioEmple1
            public static void consultar(int num, double importe) throws 1
  ublic class AleatorioEmple {
      public static RandomAccessFile file;
      public static void prueba()
           int prueba = 0; // Declaración en AleatorioEmple2
      public static void consultar(int num, double importe) throws IOException
inter a commit comment
One or more of the files being committed contain warnings.
 Comment
  Commit de Aleatorio Emple 1
 Choose a previously entered comment or template:
                                                             B
 Keep Locks
                                                                     Paste selected names
   Resource
                                                  Content
                                                               Properties
                                                                           Treat as Edit
✓ M AleatorioEmple1/src/paquete1/AleatorioEmple.java
                                                  Modified
```

Como nos vuelve a surgir un conflicto, lo vamos a solucionar.

Esta vez, en vez de una línea roja, nos saldrán dos, una azul y otra negra. Estos bloques de colores hacen referencia a dónde están los cambios, si en la copia de trabajo local o en la del repositorio. La que queremos actualizar para fusionar, es la azul, así que hacemos clic en el cuadrito azul que hay en el centro. Después validaremos.



```
AleatorioEmple1 34 [https://DESKTOP-L337VTB:4430/svn/Repo1/: SVN_trabajo1]

> ■ JRE System Library [JavaSE-15]

* □ src 34

* □ paquete1 34

> □ MeleatorioEmple.java 34

> □ module-info.java 26

* □ AleatorioEmple2 34 [https://DESKTOP-L337VTB:4430/svn/Repo1/: SVN_trabajo1]

> ■ JRE System Library [JavaSE-15]

* □ src 34

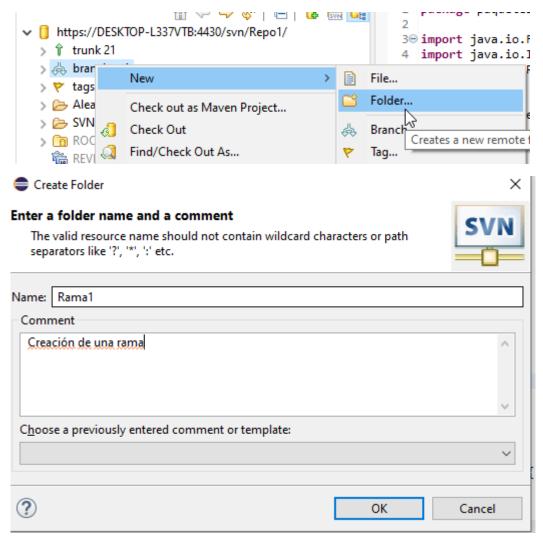
* □ paquete1 34

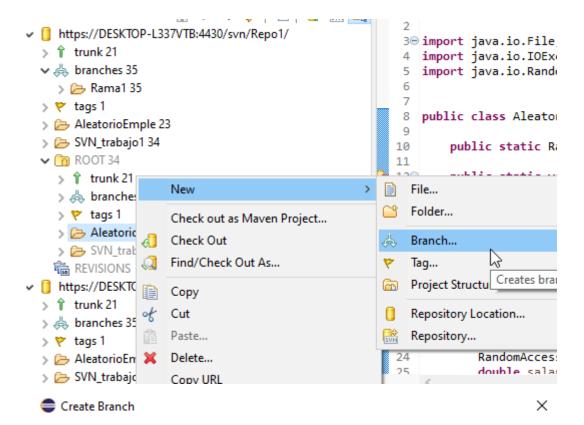
> □ AleatorioEmple.java 34

> □ module-info.java 26
```

Podemos comprobar que ambas copias están en la misma versión.

Ahora vamos a crear una rama, llamada Rama1. Para ello creamos una carpeta en "branches", con el nombre Rama1. Después iremos a "ROOT > AleatorioEmple", clic derecho y "New > Branch". Buscaremos la rama que hemos creado en la carpeta "Branches" y la seleccionamos.



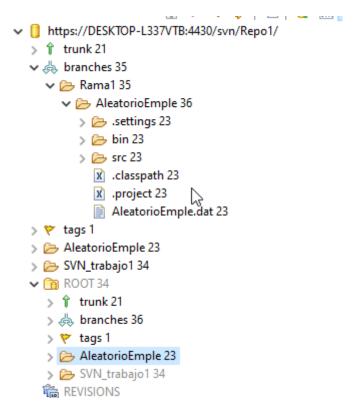


## Enter a branch location and a comment

The branch with the name 'Rama1' already exists.

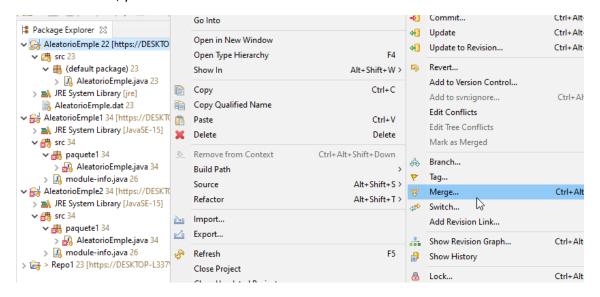


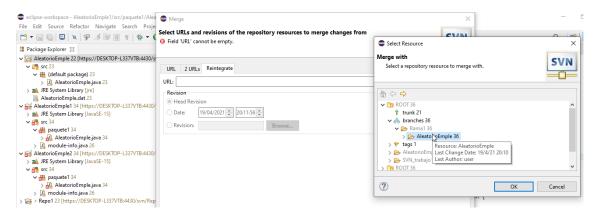
ranch: Ram	a1	~	Browse
Revision			
Head Rev	ision		
O Date:	19/04/2021 🗘 20:10:01 🕏		
Revision:	Browse		
Comment			
	el proyecto a la rama		^
	el <u>proyecto</u> a <u>la rama</u>		^
	el proyecto a la rama		^
Añadimos	el proyecto a la rama eviously entered comment or template:		^



Podemos comprobar que la carpeta Rama1 ahora contiene una rama de AleatorioEmple, que está en ROOT.

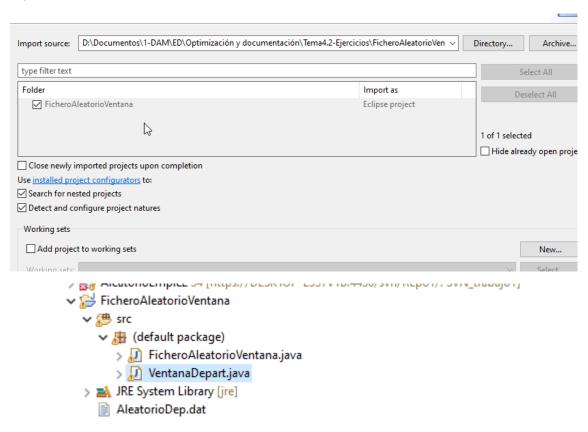
Ahora vamos a realizar una fusión entre la copia de trabajo de "trunk", y la copia de trabajo de Rama1. Para ello, tendremos que hacer clic derecho en la copia de trabajo de "trunk", e ir a "Team > Merge...", y seleccionaremos la pestaña "Reintegrate". En la URL, seleccionaremos la URL de la Rama1, y damos a continuar.



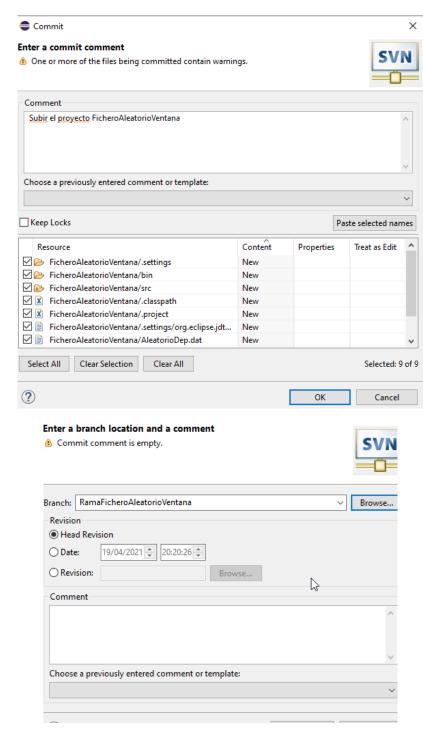


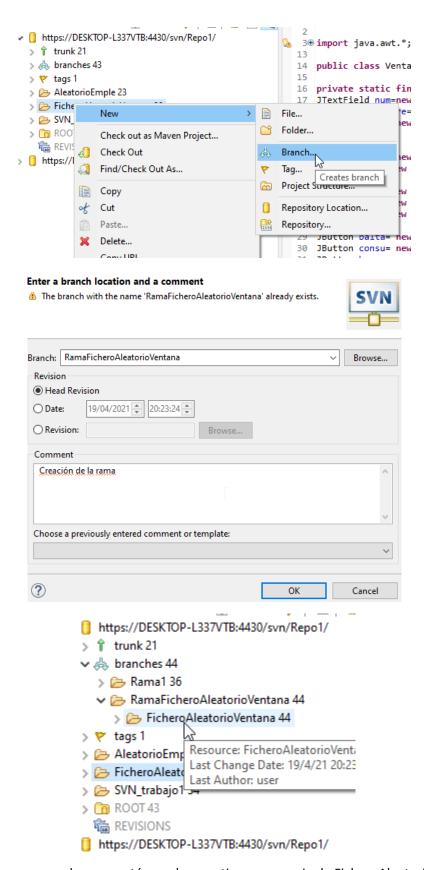
Con esto habremos fusionado la Rama1 y la copia de "trunk", luego procedemos a validar los cambios.

A continuación, importaremos el trabajo FicheroAleatorioVentana, y lo añadiremos al repositorio.



Validaremos la copia de trabajo en el repositorio. Después, crearemos una rama para el repositorio.





Aquí podemos ver que la rama está creada y contiene una copia de FicheroAleatorioVentana.

Procedemos a hacer una copia de trabajo de la rama con Checkout As.

You can checkout the selected repository resource by different ways. Select the method of checkout you prefer to use. Choose how to check out folder 'FicheroAleatorioVentana' ('Checkout using New Project Wizard' and 'Find Projects' options are available only if there is no .project file in the resource) Check out as a project configured using the New Project Wizard Find projects in the children of the selected resource Check out as a folder into existing project Check out as a project with the name specified: FicheroAleatorioVentana1 Depth: Recursively Revision Head Revision 19/04/2021 O Date: 20:25:50 \$ Revision: Browse... B FicheroAleatorioVentana 44 [https://DESKTOP-L337VTB:4430/svn/Repo1/, Branch: f ▼ FicheroAleatorioVentana1 44 [https://DESKTOP-L337VTB:4430/svn/Repo1/, Branch: √ № (default package) 38 > A FicheroAleatorioVentana.java 38 > A VentanaDepart.java 38 JRE System Library [jre] AleatorioDep.dat 38 D----1 22 [L44---//DECVTOD | 227/TD-4420/----/D----1/-D----11

FicheroAleatorioVentana es la copia de trabajo de "trunk", mientras que FicheroAleatorioVentana1 es la copia de trabajo de la rama.

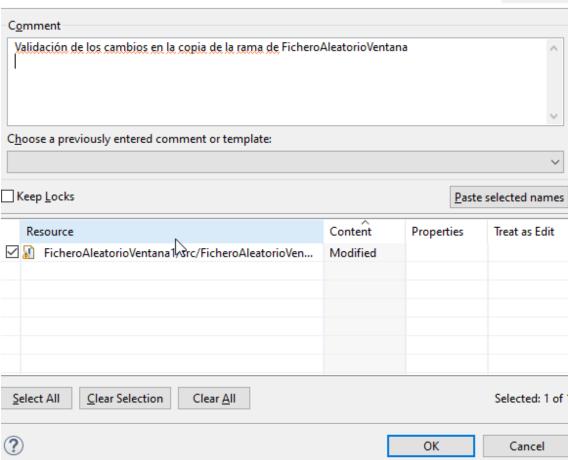
Vamos a crear un método en la copia de trabajo de la rama. Y luego lo validaremos.

```
🔃 FicheroAleatorioVentana.java 🛭 🔃 VentanaDepart.java
1⊕ import java.io.File;
 8 public class FicheroAleatorioVentana {
10⊝
         public static void main(String[] args) throws IOExc
             JFrame f = new JFrame("DEPARTAMENTOS.");
             // Nos assguramos que el fichero exista para ex
File fichero = new File("AleatorioDep.dat");
13
14
             RandomAccessFile file;
15
             file = new RandomAccessFile(fichero, "rw");
16
             file.close();
19
             VentanaDepart v = new VentanaDepart(f);
20
             v.setVisible(true);
21
         }//fin main
22
         public static void prueba() {
24⊝
25
             // No hace nada
26
28
 20 1//fin class
```

## Enter a commit comment

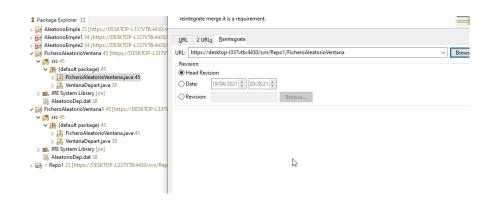
One or more of the files being committed contain warnings.





Ahora queremos hacer la fusión entre las dos copias, la de "trunk" y la de la rama.

Para ello, vamos a hacer clic derecho en la copia de "trunk", seleccionar "Team > Merge", y la pestaña "Reintegrate". Seleccionamos la URL de la rama, y seleccionamos continuar. Nos aparecerán algunos conflictos, pero se solucionarán de la misma manera que los últimos, seleccionando el cuadrado azul que aparece entre las dos copias. Después guardaremos el proyecto, y validaremos la copia de "trunk".



Después vamos a modificar el módulo que creamos anteriormente en la copia de la rama y vamos a validarla.

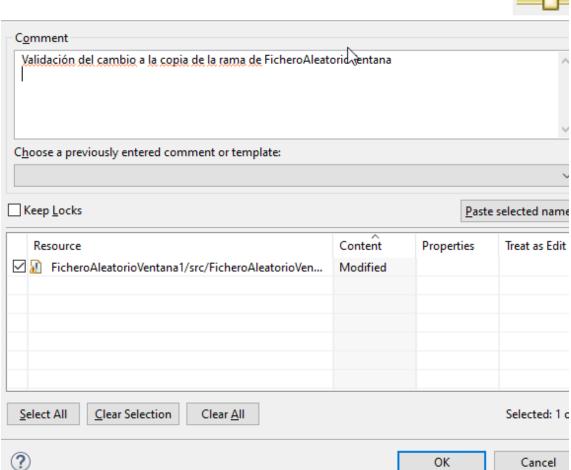
```
public static void mostrar(int a) {
    System.out.println(a);
    // Muestra por pantalla el número dado
}

/fin class
```

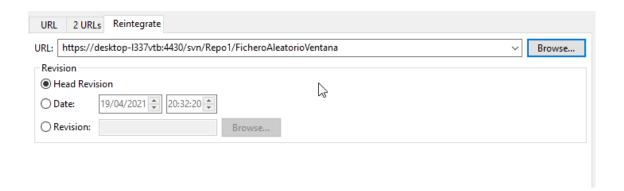
## Enter a commit comment

One or more of the files being committed contain warnings.





Podemos ver que tienen revisiones diferentes. Ahora vamos a fusionarlas. Seguimos los mismos pasos de antes.



```
FicheroAleatorioVentana.java [Rev:HEAD]
                                  ⊕ import java.io.File;
public class FicheroAleatorioVentana {
      public static void main(String[] args) throws IOException {
         JFrame f = new JFrame("DEPARTAMENTOS.");
          // Nos aseguramos que el fichero exista para evitar posibles errores
         File fichero = new File("AleatorioDep.dat");
         RandomAccessFile file;
         file = new RandomAccessFile(fichero, "rw");
         file.close();
         VentanaDepart v = new VentanaDepart(f);
         v.setVisible(true);
     }//fin main
     public static void mostrar(int a) {
         System.out.println(a);
         // Muestra por pantalla el número dado.
) }//fin class
```

Esta es la copia de "trunk", que tiene el mismo módulo después de la fusión.