

Eclipse pull up es uno de los refactorings más habituales de Eclipse cuando estamos trabajando con jerarquías de clases. ¿Para qué sirve este refactoring y su complementario pull down? . Sirven para mover métodos entre diferentes niveles de una jerarquía de clases . Supongamos que disponemos de tres clases (Persona , Empleado y Deportista) y existe una relación de herencia entre las clases. Empleado y Deportista heredan de Persona , el código sería el siguiente:

```
package com.arquitecturajava;

public class Persona {

    private String nombre;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

```
package com.arquitecturajava;

public class Empleado extends Persona{
```

```
private String trabajo;

public String getTrabajo() {
    return trabajo;
}

public void setTrabajo(String trabajo) {
    this.trabajo = trabajo;
}

public void andar() {
    System.out.println("la persona anda");
}
}
```

```
package com.arquitecturajava;

public class Deportista extends Persona {

    private String deporte;

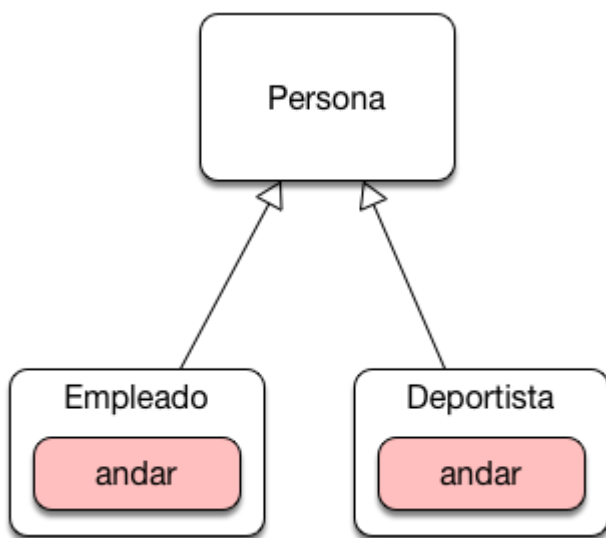
    public String getDeporte() {
        return deporte;
    }

    public void setDeporte(String deporte) {
        this.deporte = deporte;
    }

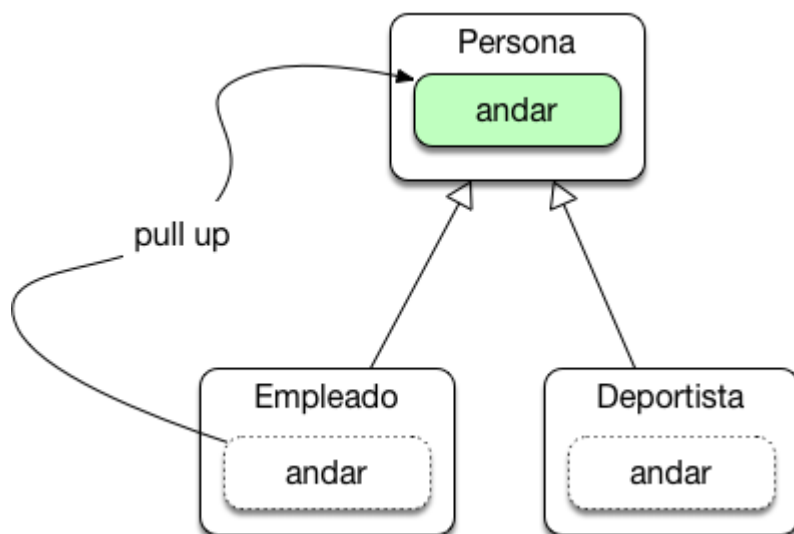
    public void andar() {
```

```
        System.out.println("la persona anda");  
    }  
}
```

El problema que tenemos con estas clases es que el método andar esta repetido en la jerarquía de clases y lo comparten tanto Empleado como Deportista:



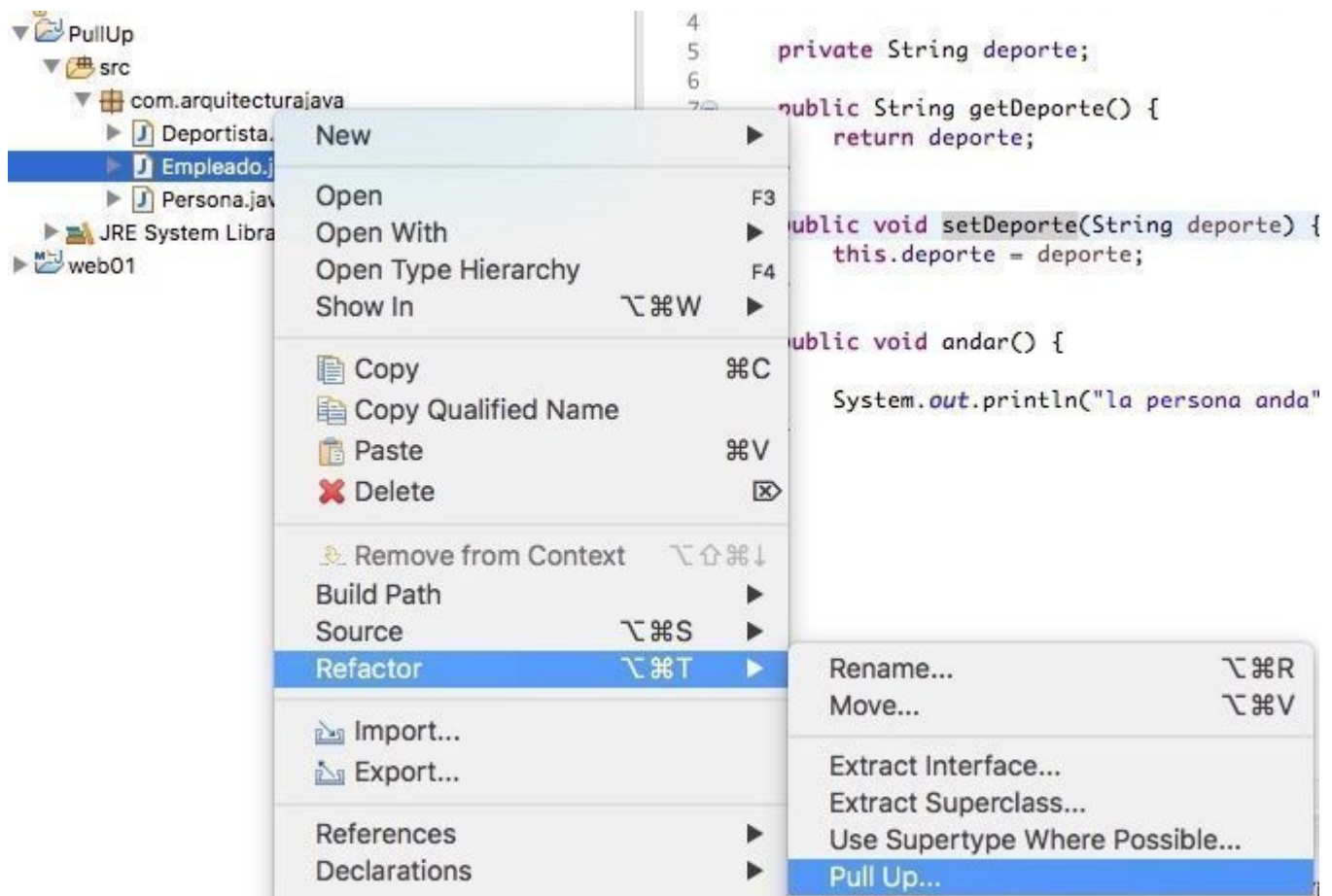
Para solventar este problema debemos modificar la jerarquía de clases y subir el método andar a la clase Persona .



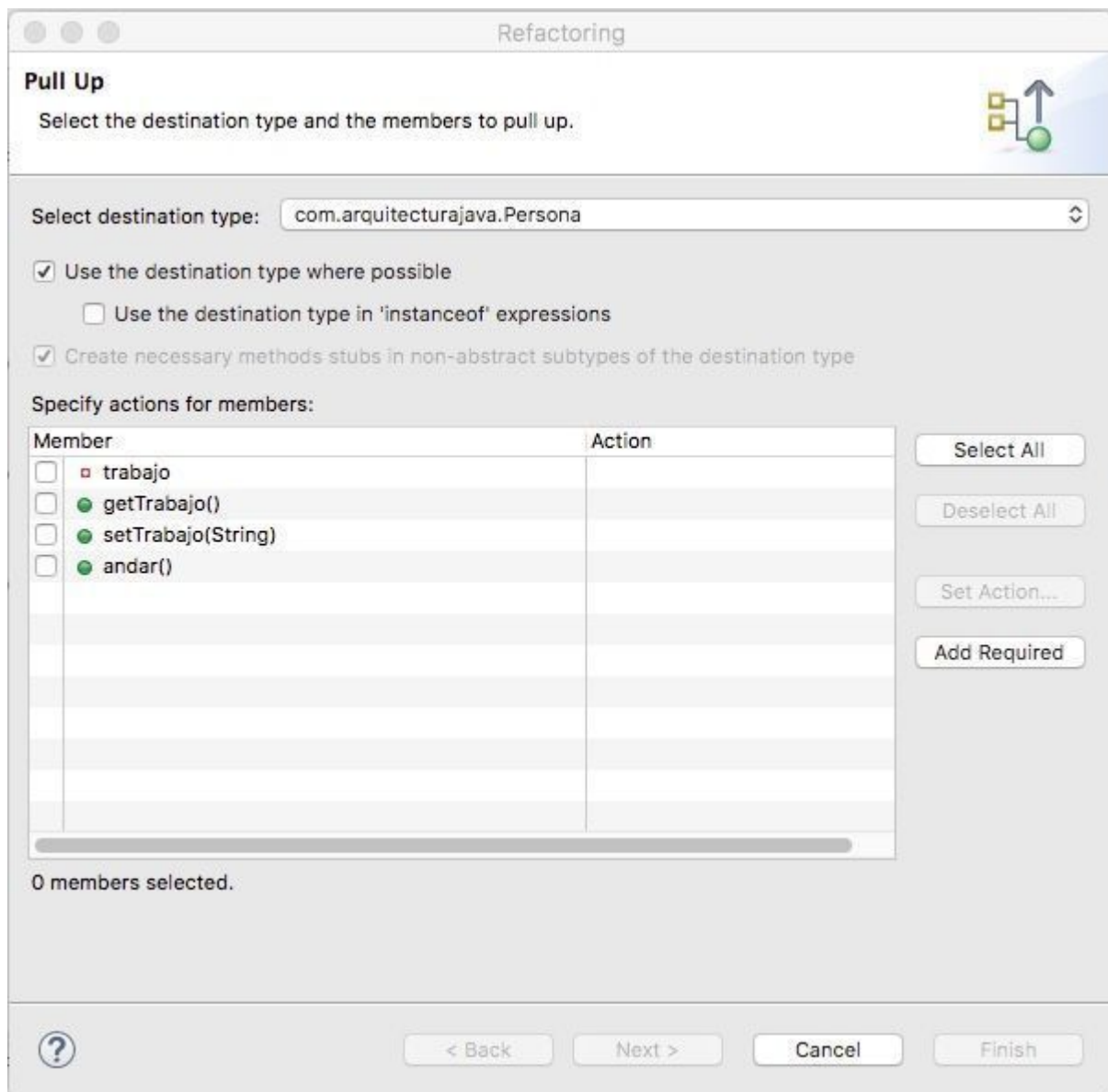
¿Cómo podemos realizar esta operación de forma automática con el Eclipse?. Vamos a usar el refactor de Eclipse Pull up para que Eclipse realice la operación por nosotros.

Eclipse Pull up

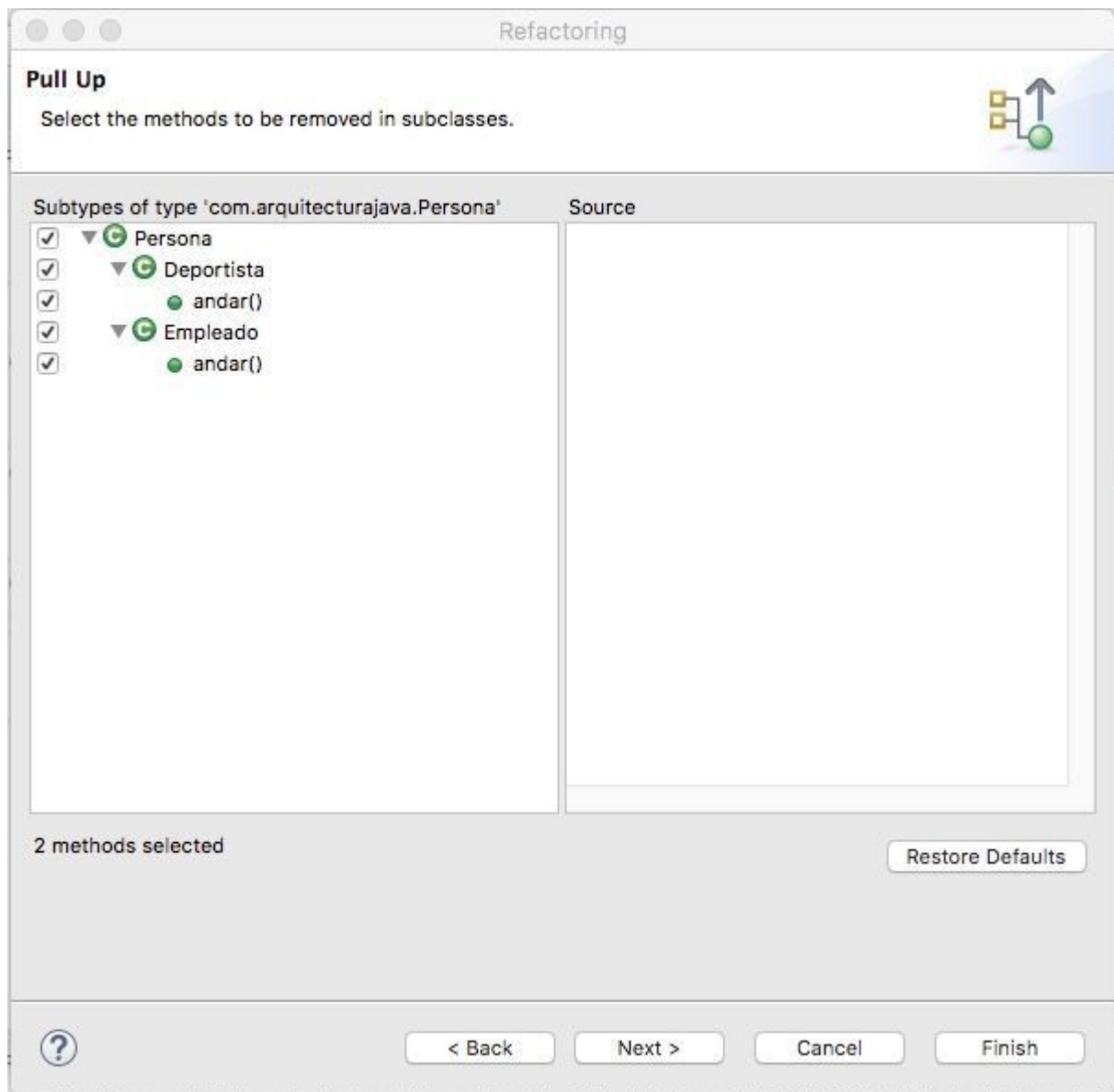
El primer paso es seleccionar la clase Empleado y pulsar botón derecho Refactor, nos aparecerán un conjunto de opciones, seleccionamos pull up:



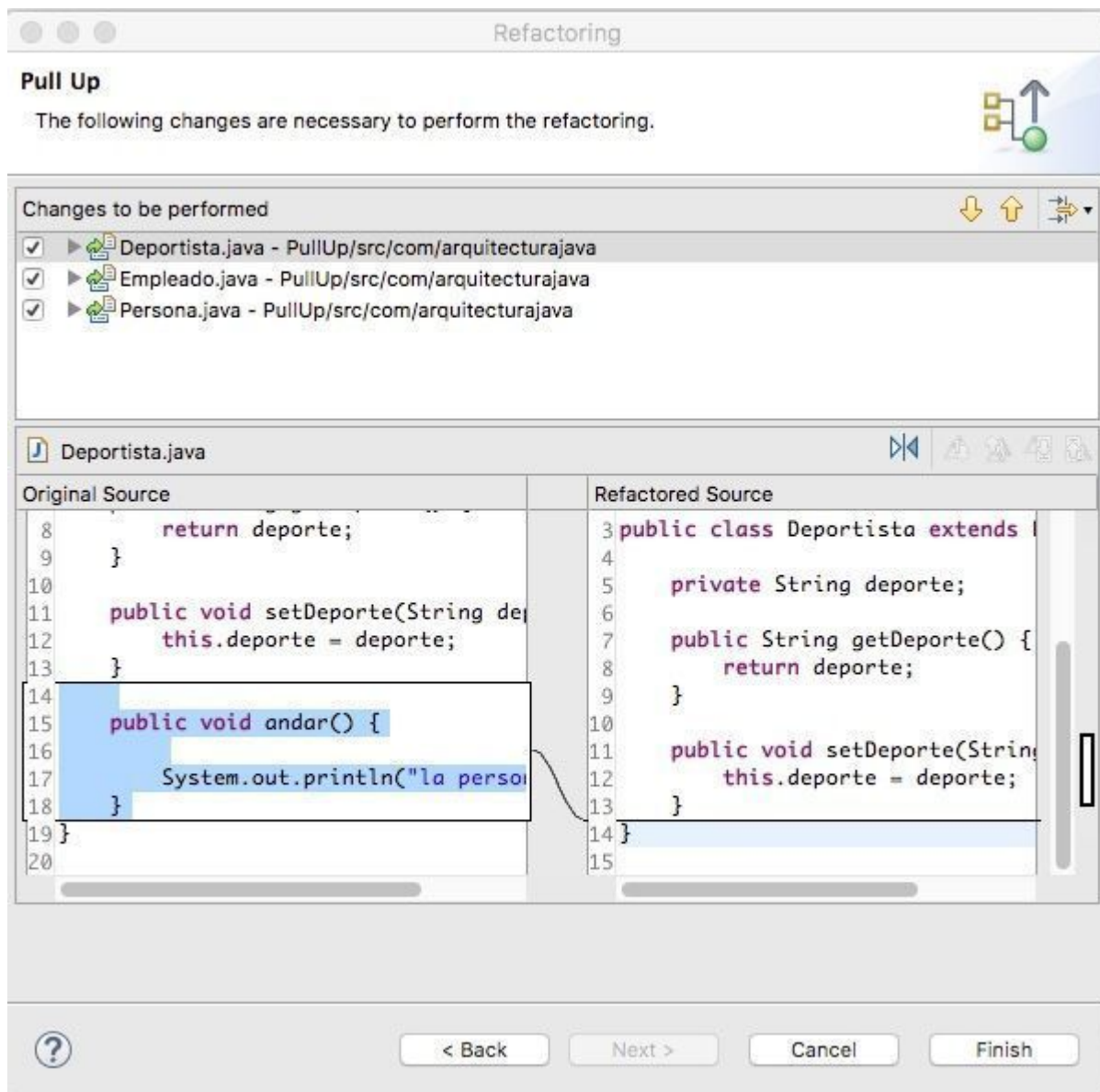
Hecho esto Eclipse presenta un menu donde podemos seleccionar los métodos que deseamos mover por la jerarquía de clases:



En esto seleccionamos el método andar() y pulsamos en siguiente:



Accederemos a un menú que nos permite seleccionar el método andar de ambas clases tanto Deportista como Empleado , marcamos ambas y pulsamos siguiente:



Eclipse nos muestra los cambios que se producirán en las diferentes clases. Si estamos conformes podemos pulsar sobre finalizar y Eclipse procederá a modificar el código fuente:

```
package com.arquitecturajava;
```

```
public class Persona {  
  
    private String nombre;  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public void andar() {  
        System.out.println("la persona anda");  
    }  
}
```

```
package com.arquitecturajava;
```

```
public class Empleado extends Persona{  
  
    private String trabajo;  
  
    public String getTrabajo() {  
        return trabajo;  
    }  
  
    public void setTrabajo(String trabajo) {  
        this.trabajo = trabajo;  
    }  
}
```

```
}
```

```
package com.arquitecturajava;
```

```
public class Deportista extends Persona {
```

```
    private String deporte;
```

```
    public String getDeporte() {  
        return deporte;  
    }
```

```
    public void setDeporte(String deporte) {  
        this.deporte = deporte;  
    }
```

```
}
```

Acabamos de subir un método dentro de la jerarquía de clases con Eclipse Pull Up. La operación de Pull Down sería muy parecida.

1. [Eclipse y el concepto de Delegación](#)
2. [Eclipse Kepler y maven](#)
3. [Organizandonos con Eclipse II \(Utility projects\)](#)