

## 89 - Métodos finales

[Listado completo de tutoriales](#)

En Java podemos sellar un método para que las subclases no puedan sobrescribirlo, para ello debemos agregar la palabra clave final previo al tipo de dato que devuelve:

```
public final void imprimir()
```

El método imprimir no podrá ser implementado en una subclase.

Un método que se lo declara como final es más eficiente, ya que el compilador puede vincular con el código definido en dicho método ya que las subclases no lo pueden implementar.

### Problema:

Confeccionar una clase Persona que tenga como atributos el nombre y la edad. Retornar si la persona es mayor de edad o no (definir dicho método como final)  
Plantear una segunda clase Empleado que herede de la clase Persona. Añadir un atributo sueldo y los métodos de cargar el sueldo e imprimir su sueldo.

Definir un objeto de la clase Persona y llamar a sus métodos. También crear un objeto de la clase Empleado y llamar a sus métodos.

### Clase: Persona

```
public class Persona {  
    protected String nombre;  
    protected int edad;  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    public void imprimirDatosPersonales() {  
        System.out.println("Nombre:" + nombre);  
        System.out.println("Edad:" + edad);  
    }  
  
    public final boolean esMayor() {  
        if (edad >= 18)  
            return true;  
        else  
            return false;  
    }  
}
```

```
}  
}
```

### Clase: Empleado

```
public class Empleado extends Persona {  
    protected int sueldo;  
  
    public Empleado(String nombre, int edad, int sueldo) {  
        super(nombre, edad);  
        this.sueldo = sueldo;  
    }  
  
    public void imprimirSueldo() {  
        System.out.println("El sueldo es:" + sueldo);  
    }  
}
```

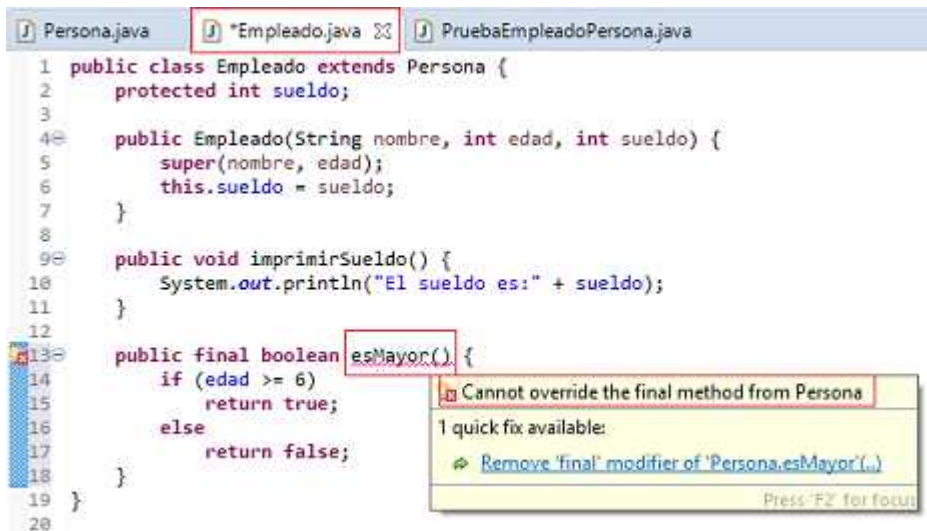
### Clase: PruebaEmpleadoPersona

```
public class PruebaEmpleadoPersona {  
    public static void main(String[] args) {  
        Persona persona1 = new Persona("Juan", 10);  
        persona1.imprimirDatosPersonales();  
        if (persona1.esMayor())  
            System.out.println("Es mayor de edad");  
        else  
            System.out.println("No es mayor de edad");  
        Empleado empleado1 = new Empleado("Carlos", 22, 10000);  
        empleado1.imprimirDatosPersonales();  
        empleado1.imprimirSueldo();  
        if (empleado1.esMayor())  
            System.out.println("Es mayor de edad");  
        else  
            System.out.println("No es mayor de edad");  
    }  
}
```

Hemos definido el método 'esMayor' de tipo final:

```
public final boolean esMayor() {  
    if (edad >= 18)  
        return true;  
    else  
        return false;  
}
```

Luego no se puede cambiar su significado en una subclase (se genera un error):



No podemos reescribir el método 'esMayor' para la clase Empleado indicando que 6 años es mayor de edad:

```
public final boolean esMayor() {  
    if (edad >= 6)  
        return true;  
    else  
        return false;  
}
```

Muchas veces por seguridad evitamos que las subclases puedan sobrescribir métodos definiendo los mismos como 'final'.

## Retornar