

# ED Refactorización

Alumno: Adrián Pisabarro García

---

1. Lo primero que he hecho es cambiar las variables y utilizar variables con nombres en formato joroba de camello y en varias líneas. Tipo:

```
Cuenta datos;  
double saldoActual;  
double saldoInteres;
```

2. He dejado espacios libres como estos y ajustarlos a la misma sangría para organizar todo el código de forma correcta:

```
try  
{  
    datos.retirar(230);  
} catch (Exception e)  
{  
    System.out.print("Fallo al retirar");  
}
```

3. He dejado espacios como estos en los operadores:

```
saldoActual = datos.getSaldo();  
saldoInteres = datos.getSaldo() + datos.getSaldo() * INTERES;  
System.out.println("\n");  
System.out.println("El saldo actual es :" + saldoActual );  
System.out.println("El saldo actual con interés es :" + saldoInteres);
```

4. Hay que saber diferenciar entre el . y , no es la misma la lengua española que el sistema inglés que es el que se utiliza en programación:

```
datos = new Cuenta("Ekaitz", "1234-5678-90-123456789", 1000.0, 0.02);
```

5. Los métodos getters y setters hay que identificarlos de la mejor forma. Aprovecho para decir que los métodos también utilizan la nomenclatura y el sistema joroba de camello.

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

6. Comentarios de una línea mejor de esta forma:

```
// Constructor vacío
```

7. Los constructores a la hora de llamar a los atributos utilizar this para acceder a ellos, no crear otros nombres.

```
// Constructor con argumentos
public Cuenta(String nombre, String idCuenta, double saldo, double
tipoInteres) {
    this.nombre = nombre;
    this.idCuenta = idCuenta;
    this.saldo = saldo;
    this.tipoInteres = tipoInteres;
}
```

8. Método corregido:

```
//Método para retirar cantidades en la cuenta. Modifica el saldo.
public void retirar (double cantidad) throws Exception {
    if (cantidad <= 0) {
        throw new Exception("No se puede retirar una cantidad negativa");
    }
    if (getSaldo() < cantidad) {
        throw new Exception("No se hay suficiente saldo");
    }
    saldo = saldo - cantidad;
}
```

9. He decidido utilizar variables globales como esta:

```
//Variables globales
final double INTERES = 0.015;

saldoInteres = datos.getSaldo() + datos.getSaldo() * INTERES;
```