



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

ACG - Auditoría Calidad y Gestión de Sistemas  
2024/2025

Francesco Lorenzoni  
PCA25403GU

## **Practica 3**

*Testing CoffeeMaker*



# Chapter 1

## Práctica 3

### 1.1 Errores en el código

#### 1.1.1 CoffeeMaker

```
@Test
public void testAddInventory() {
    // Comprueba los valores iniciales
    Inventory inventory = coffeeMaker.
        checkInventory();
    assertEquals(15, inventory.getCoffee());

    // Agrega inventario
    boolean added = coffeeMaker.addInventory
        (5, 3, 2, 1);

    assertTrue(added); // Falla
```

```
public boolean addInventory(int amtCoffee,
    int amtMilk, int amtSugar,
    int amtChocolate) {
    boolean canAddInventory = true;
    // ERROR!!! amtSugar > 0
    if(amtCoffee < 0 || amtMilk < 0 ||
        amtSugar > 0 || amtChocolate < 0) {
        canAddInventory = false;
    }
    else {
        ...
```

```
@Test
public void testMakeCoffee() {
    coffeeMaker.addRecipe(recipe1);

    Inventory initialInventory = coffeeMaker
        .checkInventory();
    int initialCoffee = initialInventory.
        getCoffee();

    int change = coffeeMaker.makeCoffee(
        recipe1, 100);

    assertEquals(50, change);

    Inventory finalInventory = coffeeMaker.
        checkInventory();
    int finalCoffee = finalInventory.
        getCoffee();

    // Falla porque se suma en lugar de
    // restar
    assertEquals(initialCoffee - 6,
        finalCoffee);
}
```

```
public int makeCoffee(Recipe r,
    int amtPaid) {
    boolean canMakeCoffee = true;
    if(amtPaid < r.getPrice()) {
        canMakeCoffee = false;
    }
    if(!inventory.enoughIngredients(r)) {
        canMakeCoffee = false;
    }
    if(canMakeCoffee) {
        // ERROR!!! Se anade en lugar de
        // restar
        inventory.setCoffee(inventory.
            getCoffee() + r.getAmtCoffee());
        ...
```

#### 1.1.2 Inventory

```
@Test
public void testSetChocolate() {
    inventory.setChocolate(5); // 15 + 5 =
    20
    assertEquals(20, inventory.getChocolate
        ());
```

Listing 1.1: Hay un += en lugar de =

```
public void setChocolate(int chocolate) {
    if(chocolate >= 0) {
        Inventory.chocolate += chocolate; //
        ERROR!!!
```

```

        inventory.setChocolate(-5);
        assertEquals(0, inventory.getChocolate());
    };
}

```

```

    }
    else {
        Inventory.chocolate = 0;
    }
}

```

```

@Test
public void testMultiplesInstancias() {
    // Crea una nueva instancia y verifica los valores
    Inventory inventory2 = new Inventory();

    assertEquals(15, inventory.getCoffee());
    assertEquals(15, inventory.getMilk());
    assertEquals(15, inventory.getSugar());
    assertEquals(15, inventory.getChocolate());
}

```

Listing 1.2: No tiene sentido que estos valores sean estaticos, hay un inventario por cada instancia

```

public class Inventory {
    private static int coffee;
    private static int milk;
    private static int sugar;
    private static int chocolate;
}

```

### 1.1.3 Recipe

```

@Test
public void testSetAmtSugar() {
    Recipe r = new Recipe();
    r.setAmtSugar(5);
    assertEquals(5, r.getAmtSugar());
    r.setAmtSugar(-5);
    assertEquals(0, r.getAmtSugar());
}

```

```

public void setAmtSugar(int amtSugar) {
    if(amtSugar >= 0) {
        // ERROR!!!
        this.amtMilk = amtSugar;
    }
    else {
        this.amtSugar = 0;
    }
}

```

## 1.2 Leer el fichero JSON

```

public class Inventory {
    ...
    private void loadFromJson(String filename) throws IOException {
        // Crea el ObjectMapper de Jackson
        ObjectMapper mapper = new ObjectMapper();

        // Lee el archivo JSON
        File jsonFile = new File(filename);
        JsonNode rootNode = mapper.readTree(jsonFile);

        // Obtiene el array de ingredientes
        JsonNode ingredientes = rootNode.get("ingredientes");

        // Procesa cada ingrediente
        for (JsonNode ingrediente : ingredientes) {
            String nombre = ingrediente.get("nombre").asText();
            int cantidad = ingrediente.get("cantidad").asInt();

            // Establece el valor segun el nombre del ingrediente
            switch (nombre) {
                case "coffee":
                    setCoffee(cantidad);
                    break;
                case "milk":
                    setMilk(cantidad);
                    break;
                case "sugar":
                    setSugar(cantidad);
                    break;
                case "chocolate":
                    setChocolate(cantidad);
                    break;
            }
        }
    }
}

```

|

}

}

}