

Guía de consignas – CafeteraInteligente

Requisitos generales

- Usar **JUnit 5**.
 - Crear la clase de pruebas **CafeteraInteligenteTest**.
 - Preparar una instancia común en **@BeforeEach** con: **marca "Acme"**, **modelo "X100"**, **capacidad 500 ml**.
 - Usar **import static** de **org.junit.jupiter.api.Assertions.***.
 - Nombrar los tests con **@DisplayName** descriptivos.
 - Cada test debe ser **independiente** (no compartir estado entre casos).
 - Mantener las **precondiciones explícitas** (por ejemplo, encender antes de operar si aplica).
-

1) Estado inicial coherente

Consigna:

Al construir la cafetera, verificar que: - **marca="Acme"**, **modelo="X100"**, **capacidadMaxima=500** - **contenidoActual=0**, **temperatura=20**, **encendida=false**, **cafesServidos=0**

Criterio de aceptación:

Usar **assertAll** para agrupar todas las aserciones del estado inicial.

2) Apagar apaga y reinicia contador de cafés (sin cambiar agua ni temperatura)

Consigna:

- Encender; cargar 300 ml; calentar hasta 100 °C; servir **al menos un** café. Guardar **aguaAntes** y **tempAntes**. - Llamar **apagar()**.

Esperado:

- **encendida=false**
- **cafesServidos=0**
- **contenidoActual == aguaAntes**
- **temperatura == tempAntes**

3) Cargar agua respeta capacidad máxima

Consigna:

- Encender. - Invocar **cargarAgua(600)** (más que la capacidad).

Esperado:

- Retorna `true`.
 - `contenidoActual=500` (se respeta el máximo).
-

4) Calentar estando apagada

Consigna:

- No encender. - Invocar `calentar()`.

Esperado:

- Retorna `false`.
 - `temperatura` permanece en 20.
-

5) Calentar sube de a 40 °C y no supera 100 °C

Consigna:

- Encender. - Llamar `calentar()` repetidas veces.

Esperado:

- Llamadas adicionales **no** superan 100 °C.
-

6) Servir café con cafetera apagada

Consigna:

- No encender. - Invocar `servirCafe()`.

Esperado:

- Retorna `false`.
-

7) Servir café exitoso (consume 100 ml y aumenta contador)

Consigna:

- Encender; `cargarAgua(300)`; calentar hasta 100 °C. - Servir 3 veces.

Esperado:

- 1ª llamada: retorna `true`; `contenidoActual=200`; `cafesServidos=1`
 - 2ª llamada: retorna `true`; `contenidoActual=100`; `cafesServidos=2`
 - 3ª llamada: retorna `true`; `contenidoActual=0`; `cafesServidos=3`
-