

## Guía para Realizar Pruebas Unitarias en JUnit

---

### 1. Preparar el Proyecto

#### 1. Configura el Proyecto en tu IDE:

- Asegúrate de que el proyecto esté configurado para soportar pruebas unitarias con JUnit.
- Incluye la dependencia de JUnit en el archivo pom.xml (si estás usando Maven).

#### 2. Crear la Clase de Prueba:

- En la carpeta src/test/java, crea una nueva clase de prueba llamada TestMiClase.

### 2. Configurar la Clase de Prueba

#### 1. Definir la Clase de Prueba:

- En la clase TestMiClase, importa JUnit y configura una instancia de MiClase que se utilizará para realizar las pruebas.

### 3. Escribir Pruebas para Cada Método

Para cada método de la clase MiClase, sigue estos pasos:

#### 1. Crear el Método de Prueba:

- Define un método de prueba para cada método de MiClase.
- Usa la anotación @Test antes de cada método de prueba.

#### 2. Definir Casos de Prueba:

- Para cada método de MiClase, debes escribir al menos tres pruebas que cubran diferentes casos (por ejemplo: casos válidos, límites y casos excepcionales).

#### 3. Usar Aserciones (Asserts):

- Usa aserciones para verificar que el resultado de cada método de prueba es el esperado.
- Ejemplos de aserciones:
  - assertEquals: Para comparar valores esperados y reales.
  - assertTrue / assertFalse: Para verificar condiciones booleanas.
  - assertNull / assertNotNull: Para verificar valores null o no null.
  - instanceof: Para comprobar el tipo de un objeto.

#### 4. Crear Casos de Prueba Específicos

Para cada método en MiClase, crea tres pruebas específicas. Usa los siguientes lineamientos:

- **Método booleano (esPar):**
  - Prueba con un número par.
  - Prueba con un número impar.
  - Prueba con cero.
- **Método que devuelve un entero (multiplicar):**
  - Prueba con dos números positivos.
  - Prueba con cero.
  - Prueba con un número positivo y uno negativo.
- **Método que devuelve un String (obtenerSaludo):**
  - Prueba con un nombre común.
  - Prueba con una cadena vacía.
  - Prueba con null y observa cómo manejarlo.
- **Método que devuelve null (obtenerValorNulo):**
  - Verifica que el método devuelva null sin condiciones adicionales.
- **Método que devuelve un objeto (crearPersona):**
  - Prueba creando una persona con un nombre y una edad válidos.
  - Verifica que el objeto no sea null.
  - Verifica que los atributos del objeto sean correctos.
- **Método que calcula el cuadrado de un número (calcularCuadrado):**
  - Prueba con un número positivo.
  - Prueba con cero.
  - Prueba con un número negativo.

- **Método booleano para cadena vacía (esCadenaVacía):**
    - Prueba con una cadena vacía.
    - Prueba con null.
    - Prueba con una cadena no vacía.
- 

## **5. Ejecutar y Verificar las Pruebas**

### **1. Ejecutar las Pruebas:**

- En el IDE, ejecuta la clase TestMiClase para ejecutar todas las pruebas.