

Uso de Asserts en Java (con Eclipse)

¿Qué es un assert?

Los asserts ayudan a detectar errores lógicos en tiempo de desarrollo y depuración. Un assert es una afirmación que el programador da por verdadera. Si la condición del assert no se cumple, el programa lanza una *AssertionError*.

Sintaxis básica:

- `assert condición;`
- `assert condición : "Mensaje de error";`

Ejemplo:

```
int edad = -5;  
assert edad >= 0 : "La edad no puede ser negativa";  
System.out.println("Edad: " + edad);
```

Si ejecutas este programa con las aserciones activadas, mostrará un error: `Exception in thread "main" java.lang.AssertionError: La edad no puede ser negativa`

¿Qué sucede si no las activas? Pruébalo.

Activar los asserts en Eclipse

Por defecto, los asserts están desactivados en Java. Para que funcionen en Eclipse:

1. Haz clic derecho sobre el archivo o clase que contiene el main.
2. Selecciona Run As → Run Configurations...
3. En la pestaña Arguments, dentro de VM arguments, escribe: `-ea`
4. Pulsa Run o Apply and Run.

Si quieres activar las assertions solo para un paquete concreto (ej. `com.miempresa.pruebas`), usa `-ea:com.miempresa.pruebas...`

Nota: Si no activas `-ea`, el programa ignorará todos los asserts.

Ejemplo completo

```
public class TestAssert {  
    public static void main(String[] args) {  
        int nota = 11; // Error lógico  
        assert nota >= 0 && nota <= 10 : "La nota debe estar entre 0 y 10";  
        System.out.println("Nota válida: " + nota);  
    }  
}
```

- Si ejecutas sin -ea, el programa se ejecuta normalmente (aunque el valor sea erróneo).
- Si ejecutas con -ea, se lanza una AssertionError.

Cuándo usar assert

✅ Durante el desarrollo o pruebas

- Para verificar precondiciones o postcondiciones.
- Para detectar estados imposibles o errores lógicos.

❌ No en código de producción

- No sustituye a las validaciones para el usuario.
- Los asserts pueden desactivarse en ejecución.

Buenas prácticas

1. Usa assert solo para condiciones que nunca deberían fallar.
2. Acompaña el assert de un mensaje claro.
3. No utilices assert para controlar entradas del usuario.
4. Recuerda probar con -ea activado antes de entregar el programa.

Actividad propuesta

1. Crea un proyecto en Eclipse llamado UD2_Asserts.
2. Escribe una clase ValidaEdad con un método:

```
static void comprobarEdad(int edad) {  
    assert edad >= 0 && edad <= 120 : "Edad no válida";  
}
```

3. Prueba con distintos valores.
4. Activa las aserciones en Eclipse y observa el resultado.