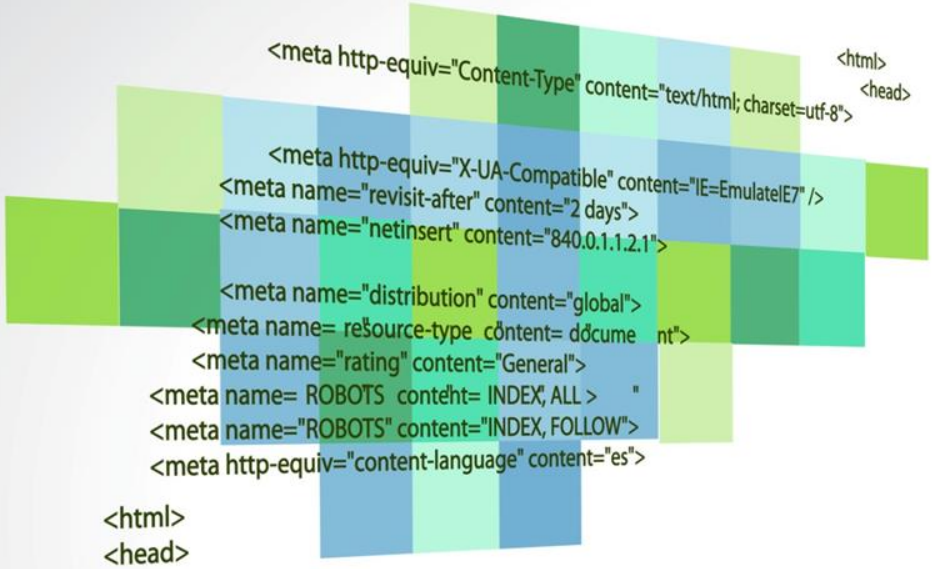

PROGRAMACIÓN ORIENTADA A OBJETOS

Material de trabajo autónomo

Pruebas Unitarias

UNIDAD 1

Indicaciones

Para un estudio eficaz, te recomendamos que sigas las siguientes pautas:

Lee con atención	Relaciona	Contrasta y complementa	Elabora	Realiza la tarea y participa en el foro

Logros de la sesión

Al finalizar esta sesión, estarás preparado para:

- ✓ Realizar pruebas unitarias

Agenda

1. Definición
2. Que beneficios tiene?
3. Como trabaja?
4. Ejemplo
5. Aserciones
6. Otros beneficios de JUNIT
7. MTA 1 Propuesto

1. Definición: Prueba Unitaria

Método para realizar pruebas a fragmentos de código de un programa, estos fragmentos deben ser **unidades** estructurales del software que se está desarrollando.

En el caso de un lenguaje orientado a objetos estos fragmentos se les denominan **métodos** de las clases de un sistema.

2. ¿Qué beneficios tiene?

- El objetivo de las pruebas unitarias es el aislamiento de las partes del código y la demostración de que estas partes no contienen errores.
- Garantiza su correcto funcionamiento y además indica el tiempo que se demoró en ejecutar la rutina probada.
- Permite un mejor trabajo en equipo por que se pueden compartir o intercambiar componentes ya probados, garantizados y listos para probalos nuevamente ante cualquier cambio.

3. ¿Cómo Trabaja?

Esta basado en aserciones, que son condiciones que debe cumplir el resultado de la ejecución del método para demostrar que su funcionamiento es correcto.

Existen herramientas, frameworks, que facilitan este trabajo, una de ellas es el framework JUNIT.

4. Ejemplo

Si un método “suma” de una clase debe calcular la suma de dos números que se les pasa como parámetros, entonces la forma de probarlo es:

```
resultado = objeto.suma(2,3);
objeto.assertEquals(5, resultado);
```

Código que al ejecutarlo debe dar como resultado una ejecución con luz “verde”, en caso contrario se prende una luz “roja” mostrando los errores.

5. Aserciones

Como en el ejemplo anterior, asseryEquals(), es un método que proveed el framework JUNIT, y sirve simplemente para comparar un resultado esperado versus el calculado por tu código (método).

assertEquals(resultado_esperado, resultado_calculado)

Existen otros métodos disponibles de JUNIT.

5. Aserciones

assertTrue **assertFalse**	Comprueba que la condición es cierta/falsa, sino lo es lanza una `AssertionFailedError` con el mensaje proporcionado (opcional)
assertEquals	Comprueba que dos objetos son iguales, sino son lanza una `AssertionFailedError` con el mensaje proporcionado (opcional)
assertNull **assertNotNull**	Comprueba que un objeto es/no es null, sino lo es lanza una `AssertionFailedError` con el mensaje proporcionado (opcional)
assertNotSame **assertSame**	Comprueba que dos objetos se refieren al mismo objeto, sino son lanza una `AssertionFailedError` con el mensaje proporcionado (opcional)
fail	Provoca que el test falle y muestra un mensaje (opcional)

6. Reglas de los test unitarios

Deben ser:

- **Rápidos**, deben ejecutarse en pocos segundos, a menudo todos a la vez.
- **Independientes**, una prueba no debe depender de otra.
- **Repetible**, el resultado del test debe dar el mismo resultado, para valores aleatorios inclusive.
- **Pequeños**, por que son más sencillas de modificar y comprender.
- **Transparentes**, el propósito de cada test debe ser claro.

6. Otros beneficios de JUNIT

- Las pruebas unitarias sirven también como punto de partida para realizar desarrollo de aplicaciones orientada a pruebas “TDD”, disciplina muy usada en metodologías “ágiles”.

- Por medio de otras herramientas las pruebas unitarias también sirven para inspeccionar el código automáticamente y detectar malas practicas de programación y de control de excepciones.

Conclusiones

- Junit es un framework que facilita la ejecución de las pruebas unitarias.
- Las pruebas unitarias son parte de los entregables que toda aplicación debería considerar.
- No es necesario imprimir los resultados en consola basta con utilizar aserciones que se encargarán de comparar los resultados obtenidos frente a los esperados y si falla muestra automáticamente los resultados obtenidos que están fallando.
- Las pruebas unitaria permite un mejor trabajo en equipo y es considerado un método ágil.