



GUSTAVO CORONEL
DESARROLLA SOFTWARE



ENTERPRISE JAVA DEVELOPER TALLER DE PROGRAMACIÓN CON JAVA

PRUEBAS UNITARIAS CON JUNIT

Eric Gustavo Coronel Castillo

I N S T R U C T O R

youtube.com/DesarrollaSoftware

gcoronelc@gmail.com

LOGRO ESPERADO

Al finalizar esta lección el participante estará en capacidad de poder realizar pruebas unitarias con Junit componentes de software.



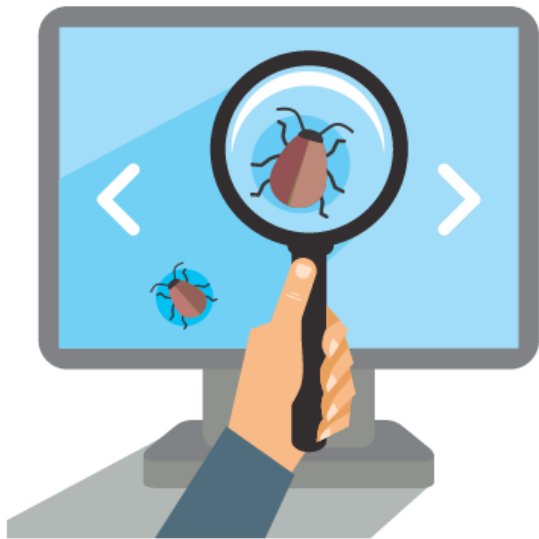
¿Por qué probamos el software?

- Para asegurar que el sistema es consistente con los requisitos y las necesidades de los usuarios y clientes.
- Para hacer un primer diagnóstico de la calidad del producto desarrollado.

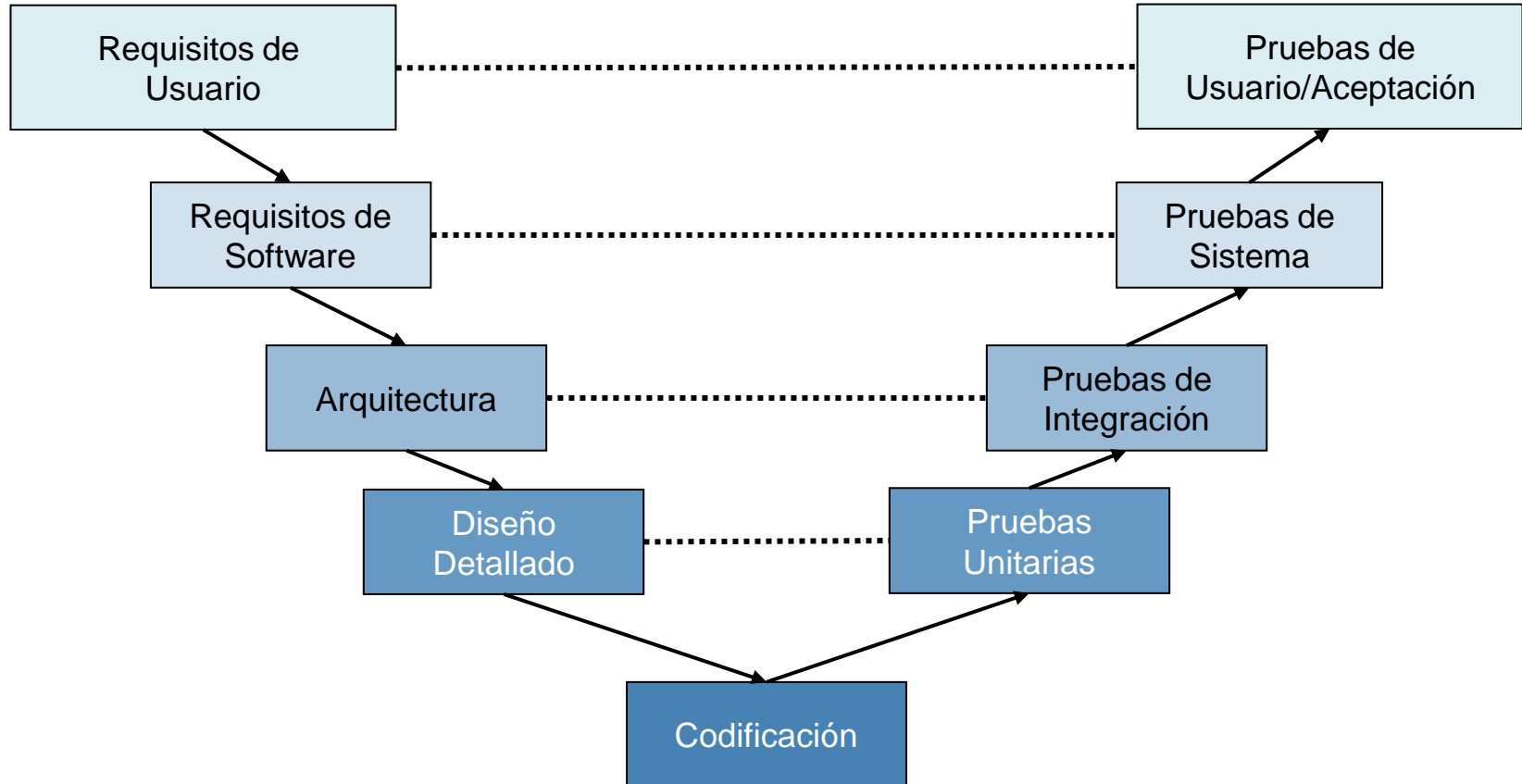


Tipos de pruebas

- Pruebas unitarias o de módulos (unit testing)
- Pruebas de integración (integration testing)
- Pruebas de sistema (system testing)
- Pruebas de Usuario/Aceptación (qualification)
- Pruebas de regresión (regression testing)



Modelo de ciclo de vida en V



Pruebas unitarias

- Propósito: Encontrar defectos en un módulo o función.
- Normalmente las pruebas unitarias son hechas por el programador del módulo.



Diseño de pruebas unitarias

- Las pruebas de cada función/método se especifican junto a la especificación de cada función.
- Debemos asegurarnos de incluir pruebas para:
 - Casos habituales y casos extremos de los parámetros.
 - Todas las excepciones.
 - Todos los tipos de efectos laterales.
- Un mismo caso de prueba puede mostrar una excepción y un efecto lateral.



unit testing

Problemas y limitaciones (1)

- “El Testing sólo puede mostrar la presencia de defectos, no su ausencia” (Dijkstra)
- Expectativas
 - pasó la prueba => ¿no tiene defectos?
- 1er Objetivo del Testing:
 - Encontrar defectos
 - para que se corrijan y con eso mejorar la calidad del producto

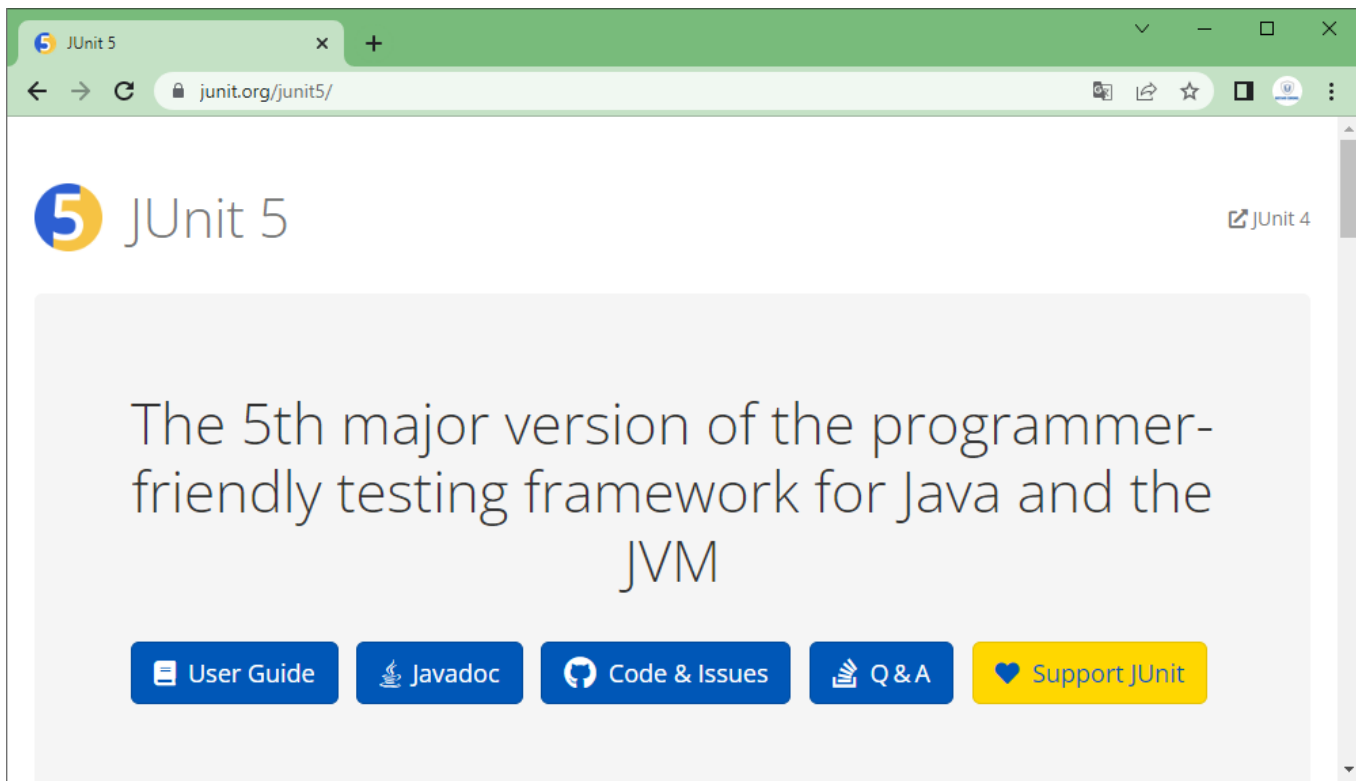
Problemas y limitaciones (2)

- Psicología de la prueba
 - Al probar nuestro producto cuál es el objetivo
 - ¿encontrar defectos o “mostrar” que funciona?
 - Importancia de la independencia
- “Cuanto más defectos se detectan en un componente, aumenta la probabilidad de que tenga más defectos”
- 2do. Objetivo del testing:
 - Evaluar la calidad de productos y componentes

JUnit

- JUnit es un “framework” para realizar pruebas de código Java;
- Consta de un conjunto de clases que el programador puede utilizar para construir sus casos de prueba y ejecutarlos automáticamente
- Los casos de prueba son realmente programas Java. Quedan archivados y pueden ser ejecutados tantas veces como sea necesario.

JUnit



Taller de pruebas unitarias

```
@DisplayName("Dado un usuario que queremos crear esperamos que cuando "
            + "llamamos al metodo 'crearUsuario' el usuario sea creado.")
@Test
public void testCrearUsuario1() {
    System.out.println("Prueba 1 del método: crearUsuario");
    Long id = 100L;
    String nombre = "Gustavo";
    UsuarioDto esperado = new UsuarioDto(id, nombre);
    UsuarioDto resultado = usuarioService.crearUsuario(id, nombre);
    assertEquals(esperado, resultado);
}
```

GRACIAS
TOTALES



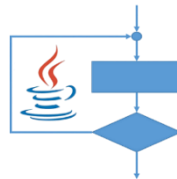
Gustavo Coronel
gcoronelc.github.io



gcoronelc.github.io

youtube.com/DesarrollaSoftware

facebook.com/groups/desarrollasoftware



FUNDAMENTOS DE PROGRAMACIÓN CON JAVA

Inicia tu aprendizaje, utilizando las mejores prácticas de programación



CURSO PROFESIONAL DE JAVA ORIENTADA A OBJETOS

Aprende programación en capas, patrones y buenas prácticas



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON PL/SQL

Aprende a obtener el mejor rendimiento de tú base de datos



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JDBC

Aprende a programar correctamente con JDBC