



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

«Misión
TIC2022»

«Misión
TIC2022»

SEMANA 5

INICIAMOS 8:05PM



UNIVERSIDAD
DE ANTIOQUIA

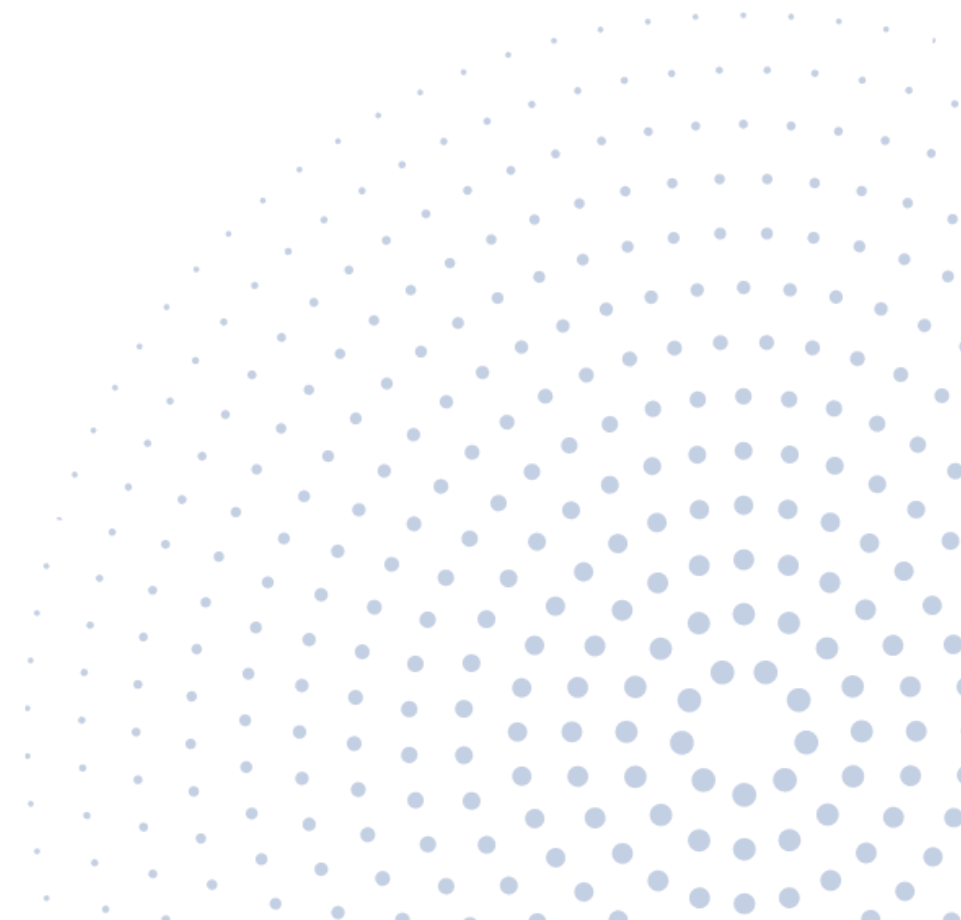
Facultad de Ingeniería

Luisa Fernanda Restrepo.



Agenda

- Introducción
- Pruebas unitarias
- JUnit





El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería



¿Qué es una prueba de software?



¿Qué son las pruebas de software?

- Incluyen planeación y control, diseño, verificación de resultados.
- Con el objetivo de:
 - Encontrar defectos
 - Prevenir defectos



Acción humana que
produce un resultado
incorrecto

Error

Fallo

Manifestación física o
funcional de un defecto.
Si un defecto es
encontrado durante la
ejecución de una
aplicación puede
producir un fallo..

Defecto

Desperfecto en un
componente o sistema
que puede causar que el
componente o sistema
falle en desempeñar las
funciones requeridas.

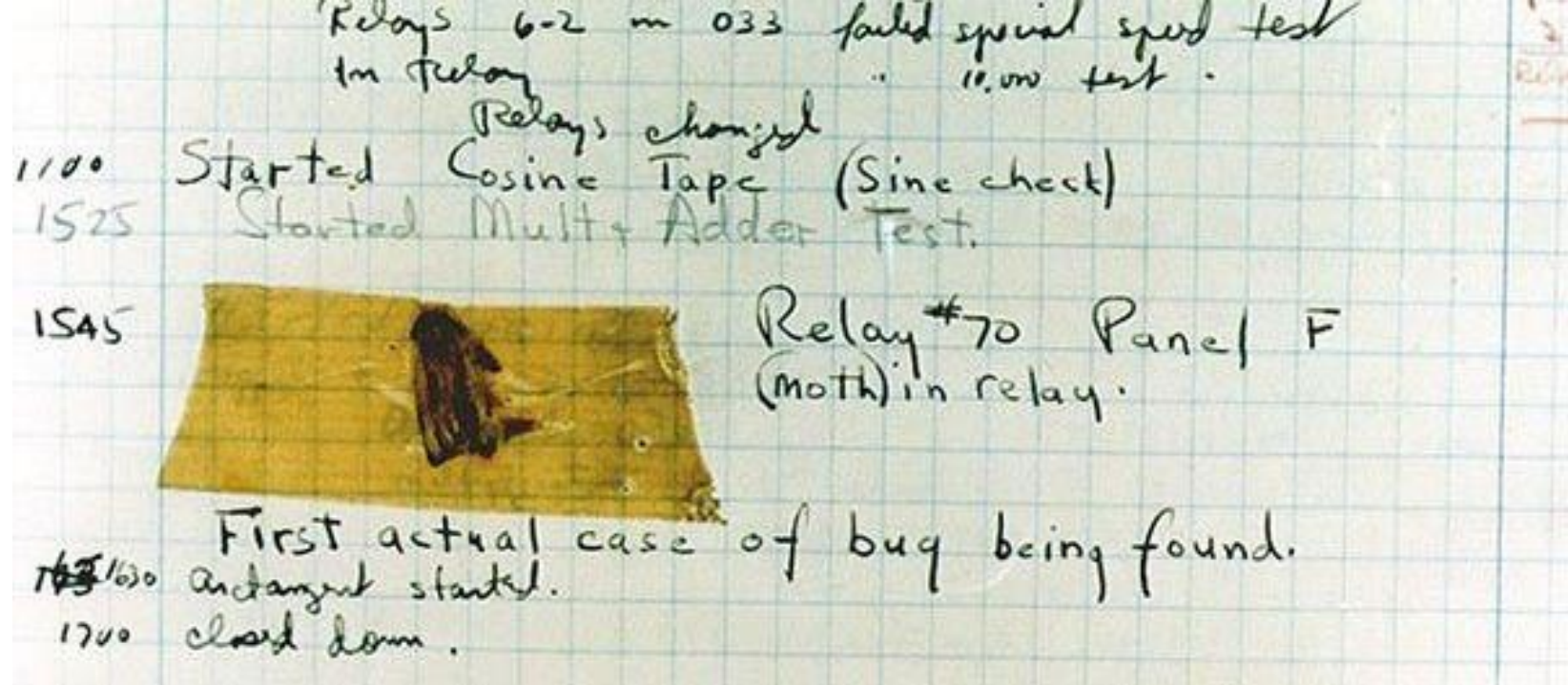
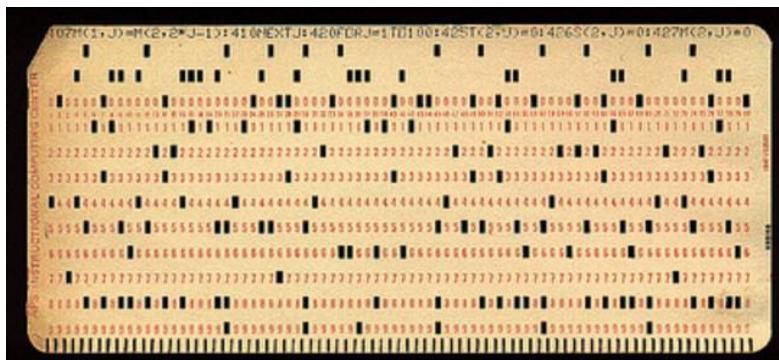
BUG



El futuro digital
es de todos

MinTIC

**1945. Mark II de la
universidad de
Harvard encontró
una polilla
enganchada a uno
de los relés del
sistema**





¿Por qué son necesarias las pruebas?

- La importancia económica del software
 - El funcionamiento de maquinaria y equipamiento depende en gran medida de un producto software.
 - No es posible imaginar grandes sistemas, en el ámbito de las finanzas ni el control de tráfico automotor, entre otros, funcionando sin un software
- Calidad software
 - Cada vez más, la calidad software se ha convertido en un factor determinante del éxito de sistemas técnicos o comerciales y productos.
- Pruebas para la mejora de la calidad
 - Las pruebas y revisiones aseguran la mejora de la calidad de productos software así como de la calidad del proceso de desarrollo en sí.



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería



Calidad

Grado en el cual un componente, sistema o proceso satisface requisitos especificados y/o necesidades y expectativas del usuario/cliente. .



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA
Facultad de Ingeniería

‘Mision
<TIC2022>’





El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería

‘Mision
<TIC2022>’



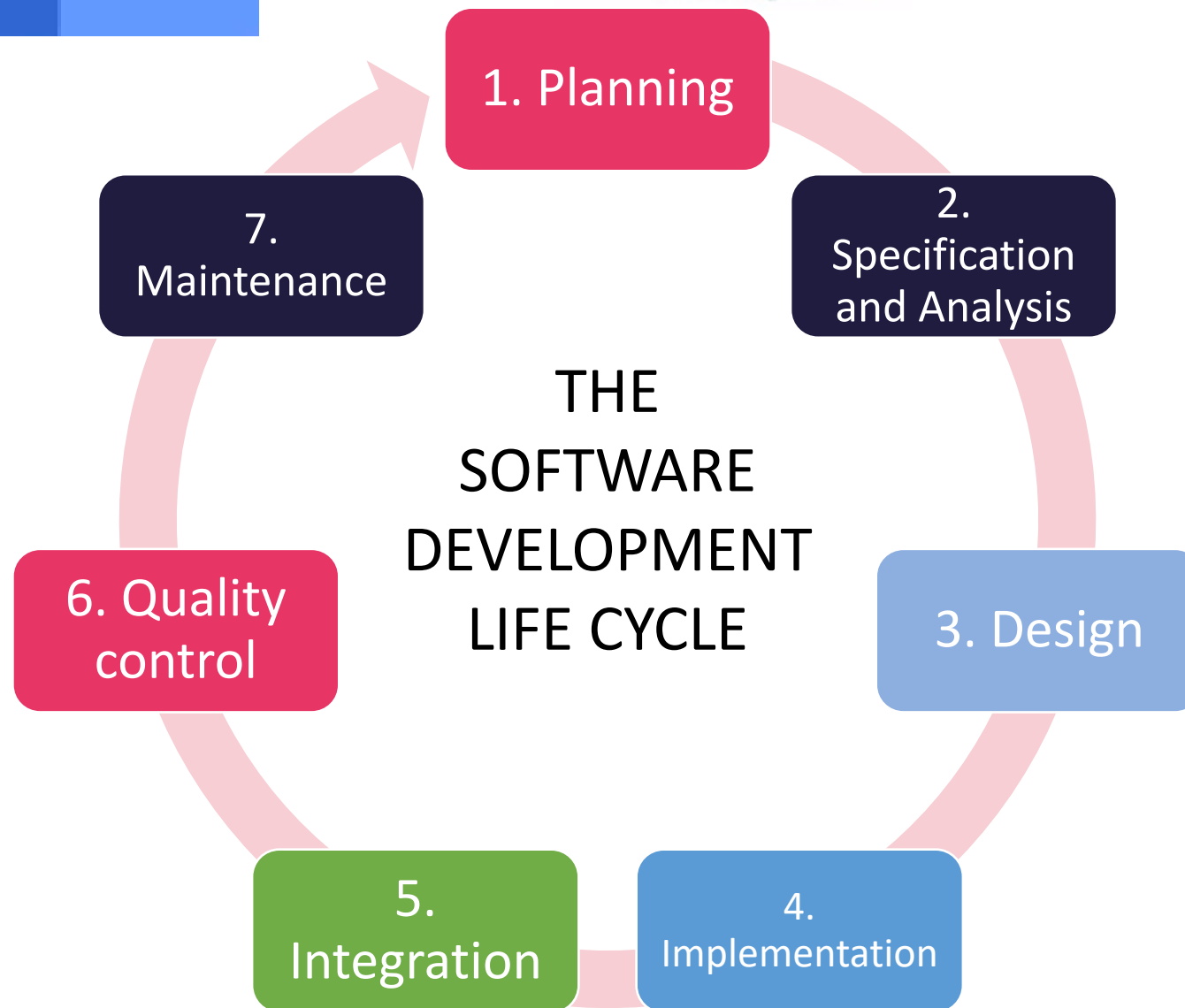


Pruebas de Software

- Las Pruebas de Software es un proceso realizado concurrentemente en las diferentes etapas del ciclo de vida de desarrollo de software que crea, utiliza y mantiene el testware y cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación con el correcto Costo/Beneficio, aumentando la confianza de los aplicativos en operación y permitiendo generar, además, información para la mejora del proceso de desarrollo de software.

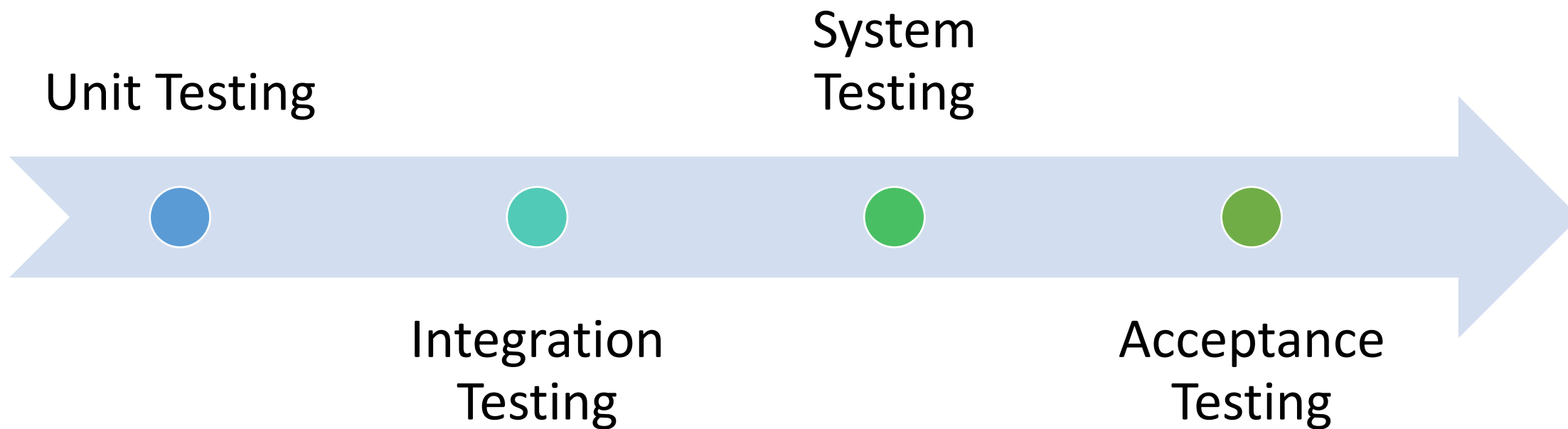


El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso.





Tipos de pruebas





Pruebas Manuales	Pruebas Automatizadas
La ejecución manual de casos de prueba sin el soporte de ninguna herramienta se conoce como prueba manual.	Tomar el soporte de la herramienta y ejecutar los casos de prueba mediante el uso de una herramienta de automatización se conoce como automatización de pruebas.
Consume mucho tiempo y es tedioso : dado que los casos de prueba los ejecutan recursos humanos, es muy lento y tedioso.	Rápido : la automatización ejecuta casos de prueba significativamente más rápido que los recursos humanos.
Gran inversión en recursos humanos : dado que los casos de prueba deben ejecutarse manualmente, se requieren más probadores en las pruebas manuales.	Menos inversión en recursos humanos : los casos de prueba se ejecutan utilizando herramientas de automatización, por lo que se requiere menos cantidad de probadores en las pruebas de automatización.
Menos confiable : las pruebas manuales son menos confiables, ya que deben tener en cuenta los errores humanos.	Más confiable : las pruebas de automatización son precisas y confiables.
No programable : no se puede realizar ninguna programación para escribir pruebas sofisticadas para obtener información oculta.	Programable : los probadores pueden programar pruebas sofisticadas para revelar información oculta.



El futuro digital
es de todos

MinTIC



Pruebas Unitarias



Prueba Unitaria

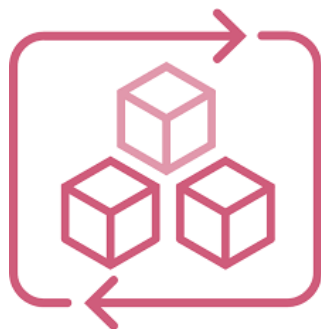
- Pruebas unitarias es el primer nivel de prueba de software donde se prueban unidades/componentes individuales de software.
- Una unidad puede ser un programa, una función, un procedimiento, un método en OO.





¿Cuales son las características?

- ▶ Unitaria.
- ▶ Completa
- ▶ Independiente.
- ▶ Objetiva
- ▶ Repetible y reutilizable.
- ▶ Automatizables.
- ▶ Profesional
- ▶ Rápidas de crear.





Ventajas

- ▶ Detecta errores temprano/
Facilita los cambios
- ▶ Calidad del código
- ▶ Facilita y favorece la integración/
Reduce el costo.
- ▶ Mejoras en el diseño
- ▶ Entendimiento del código
Procesos de debug

Desventajas

- ▶ No permite determinar problemas
de integración o desempeño.
- ▶ No es trivial anticipar todos los casos
especiales de entradas.
- ▶ Las pruebas unitarias determinan la
presencia de defectos, no la
ausencia de éstos.



Herramientas

JUnit



PHPUnit



TestNG



El futuro digital
es de todos

MinTIC



JUnit



¿Qué es JUnit?

- JUnit es un marco de pruebas unitarias para el lenguaje de programación Java. Desempeña un papel crucial en el desarrollo impulsado por pruebas y es una familia de marcos de pruebas unitarias conocidos colectivamente como xUnit.
- JUnit promueve la idea de "primero probar y luego codificar", que enfatiza la configuración de los datos de prueba para un fragmento de código que se puede probar primero y luego implementar.
- Aumenta la productividad del programador y la estabilidad del código del programa, lo que a su vez reduce el estrés del programador y el tiempo dedicado a la depuración.



Método	Definición
assertEquals(Object, Object)	El método reporta “success” si los dos objetos son los mismos usando el método equals.
assertTrue(booleanExpression)	El método reporta “success” si la expresión booleana evalúa true.
assertNull(Object)	El método reporta “success” si la referencia del objeto pasado es null.
fail(String)	EL método causa que la prueba falle e imprime el string resultante.



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA
Facultad de Ingeniería



Descargar JUnit4

JUnit 4:

<https://github.com/downloads/junit-team/junit/junit-4.10.jar>



El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

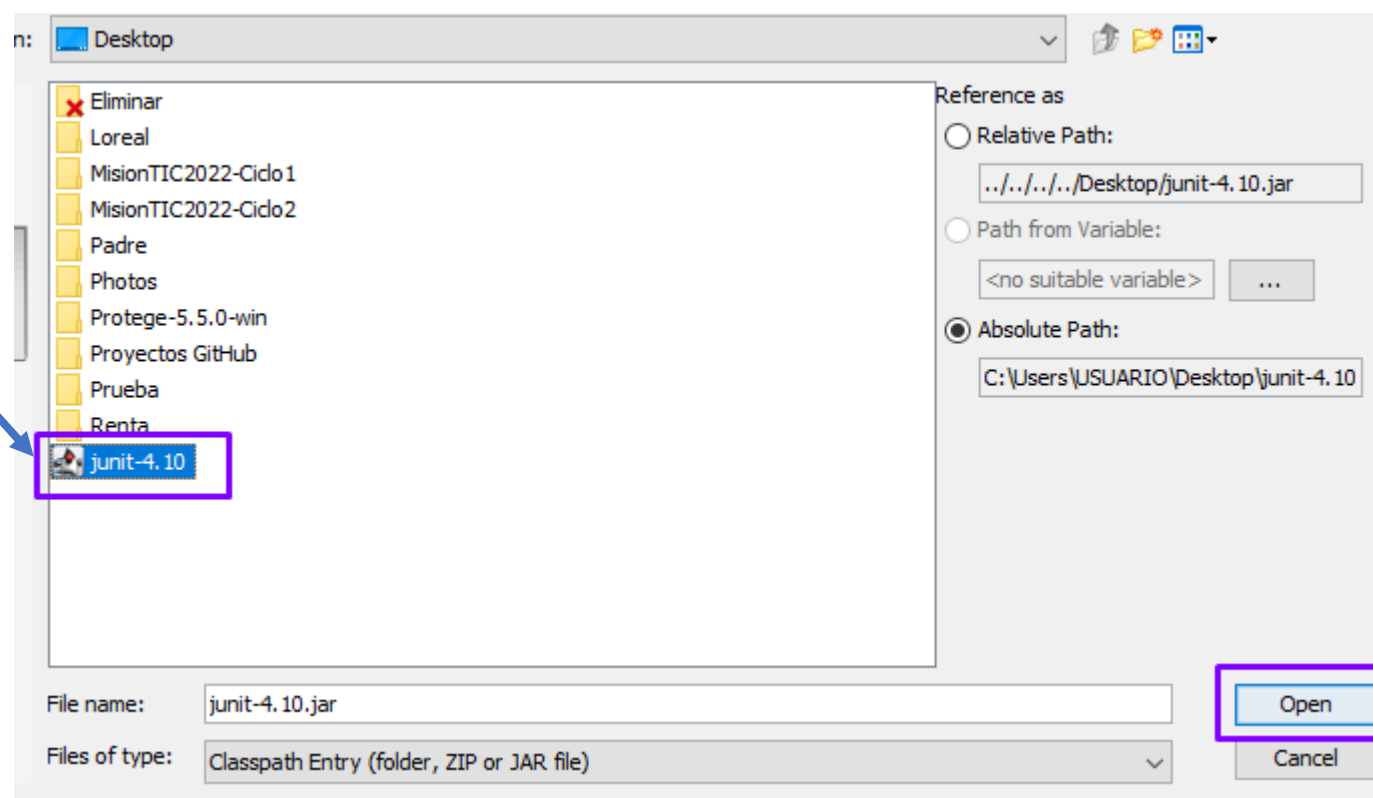
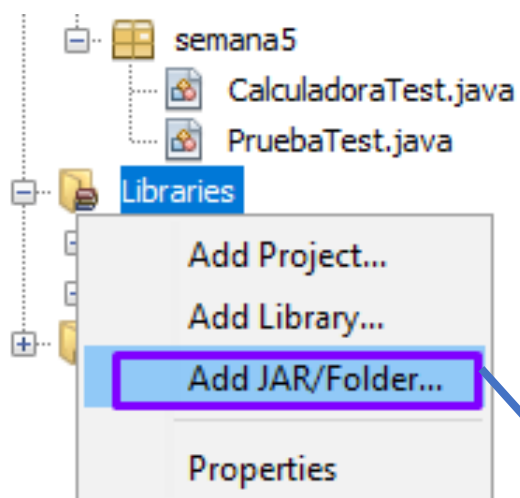
Facultad de Ingeniería

‘Mision
<TIC2022>’



<https://netbeans.apache.org/>

NetBeans





El futuro digital
es de todos

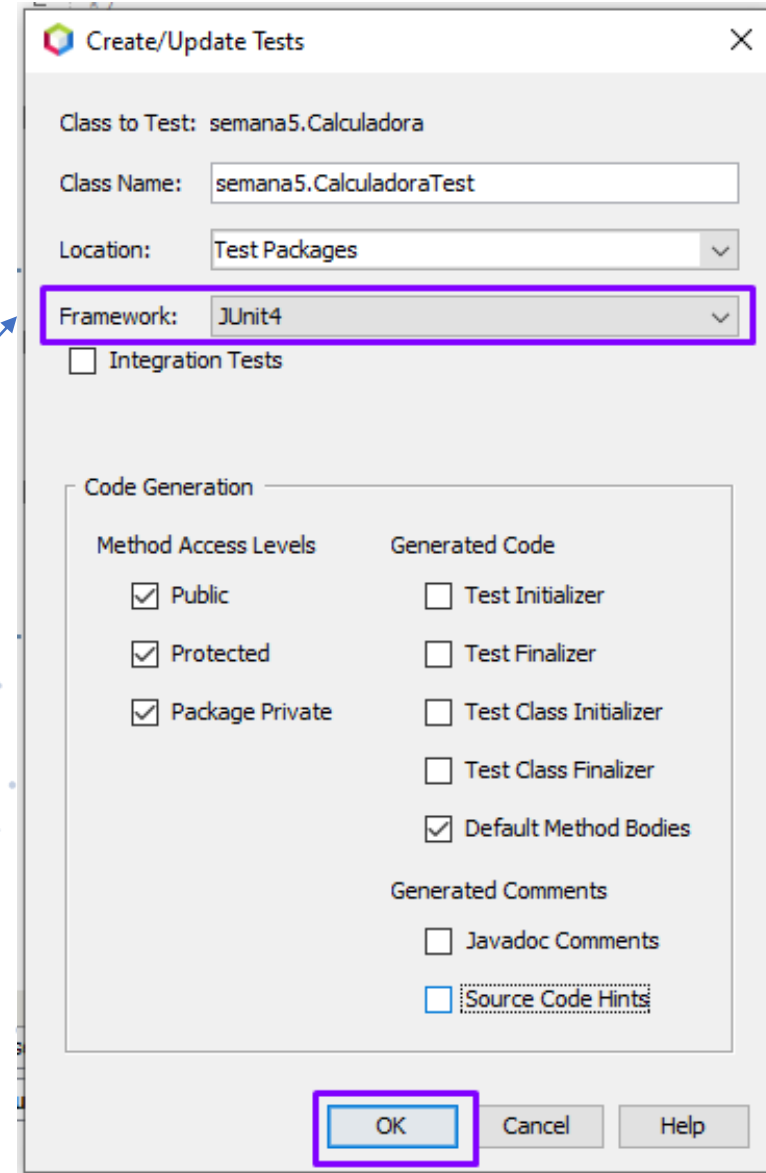
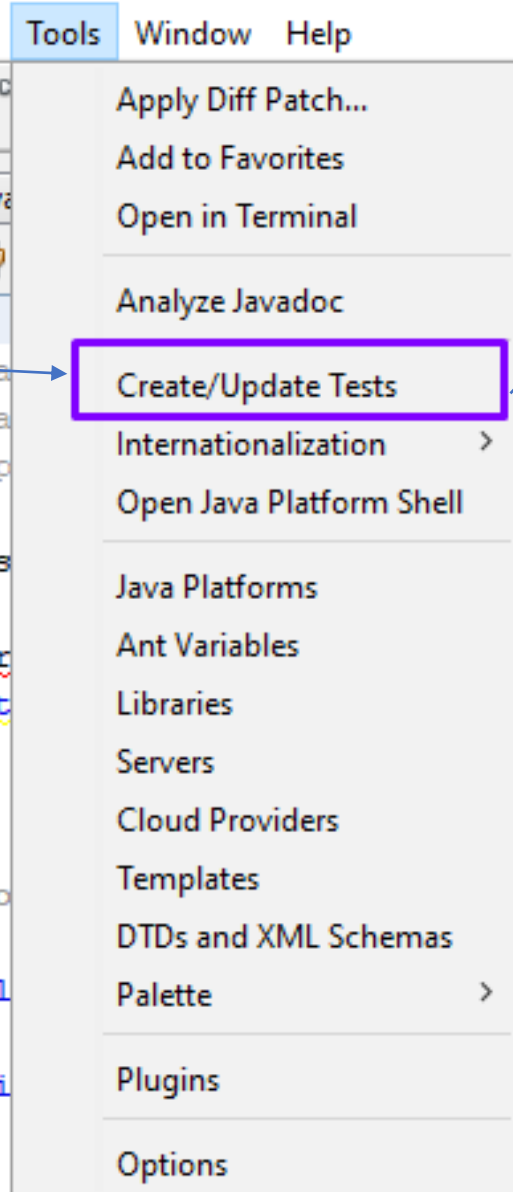
MinTIC



OPCION 1



```
public class Calculadora {  
  
    public double getSuma(double a, double b) {  
        return a + b;  
    }  
  
    public double getResta(double a, double b) {  
        return a - b;  
    }  
  
}
```





```
20 public class CalculadoraTest {
21
22     public static Calculadora instance;
23
24     public CalculadoraTest() {
25     }
26
27     @Test
28     public void testGetResta() {
29         assertEquals(1, 1, 0);
30     }
31 }
```

Test Results × Output - CodigosPruebasNetbeans (test)

semana5.CalculadoraTest ×

Tests passed: 100.00 %

The test passed. (0.089 s)

- semana5.CalculadoraTest passed
 - testGetResta passed (0.001s)

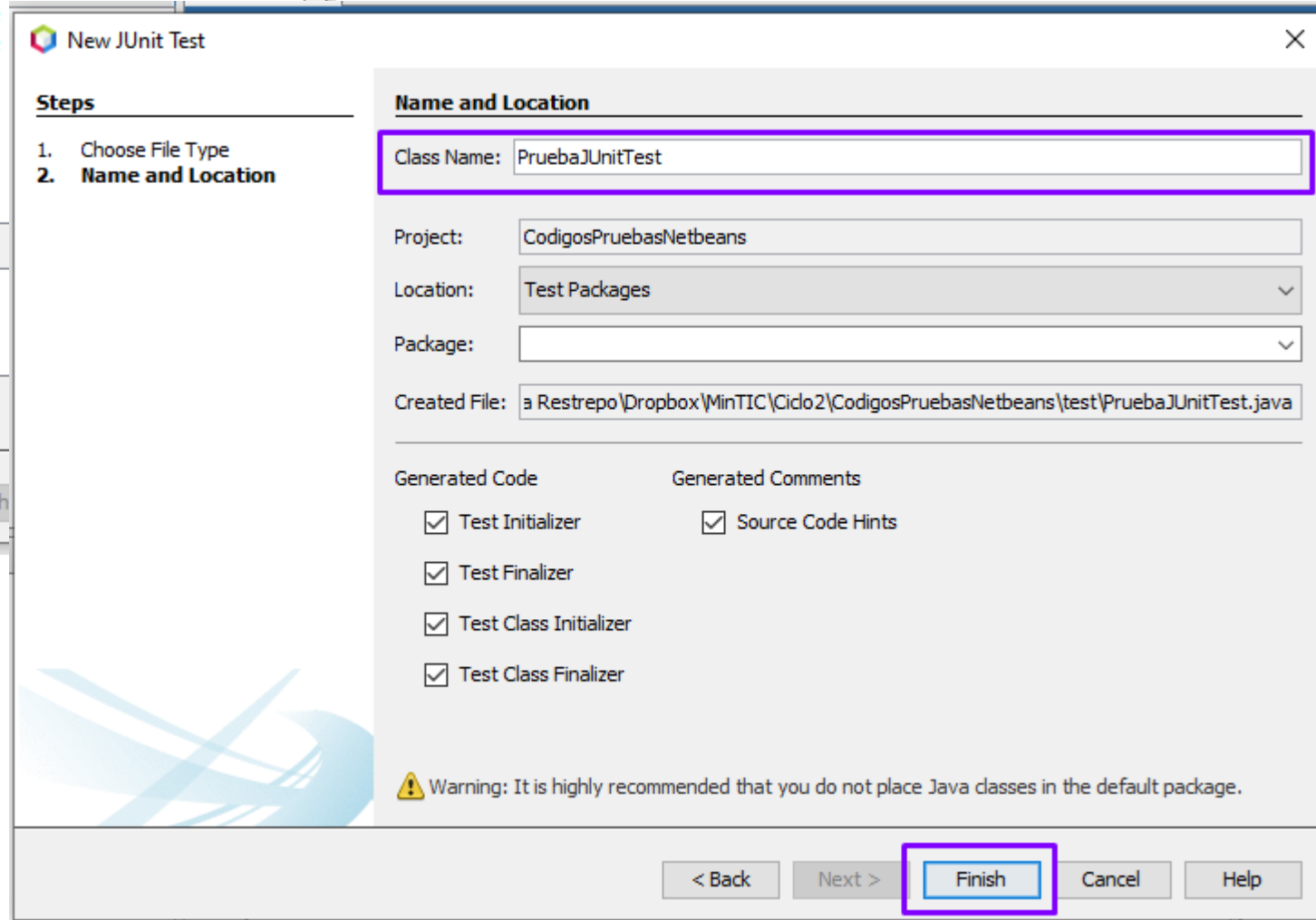
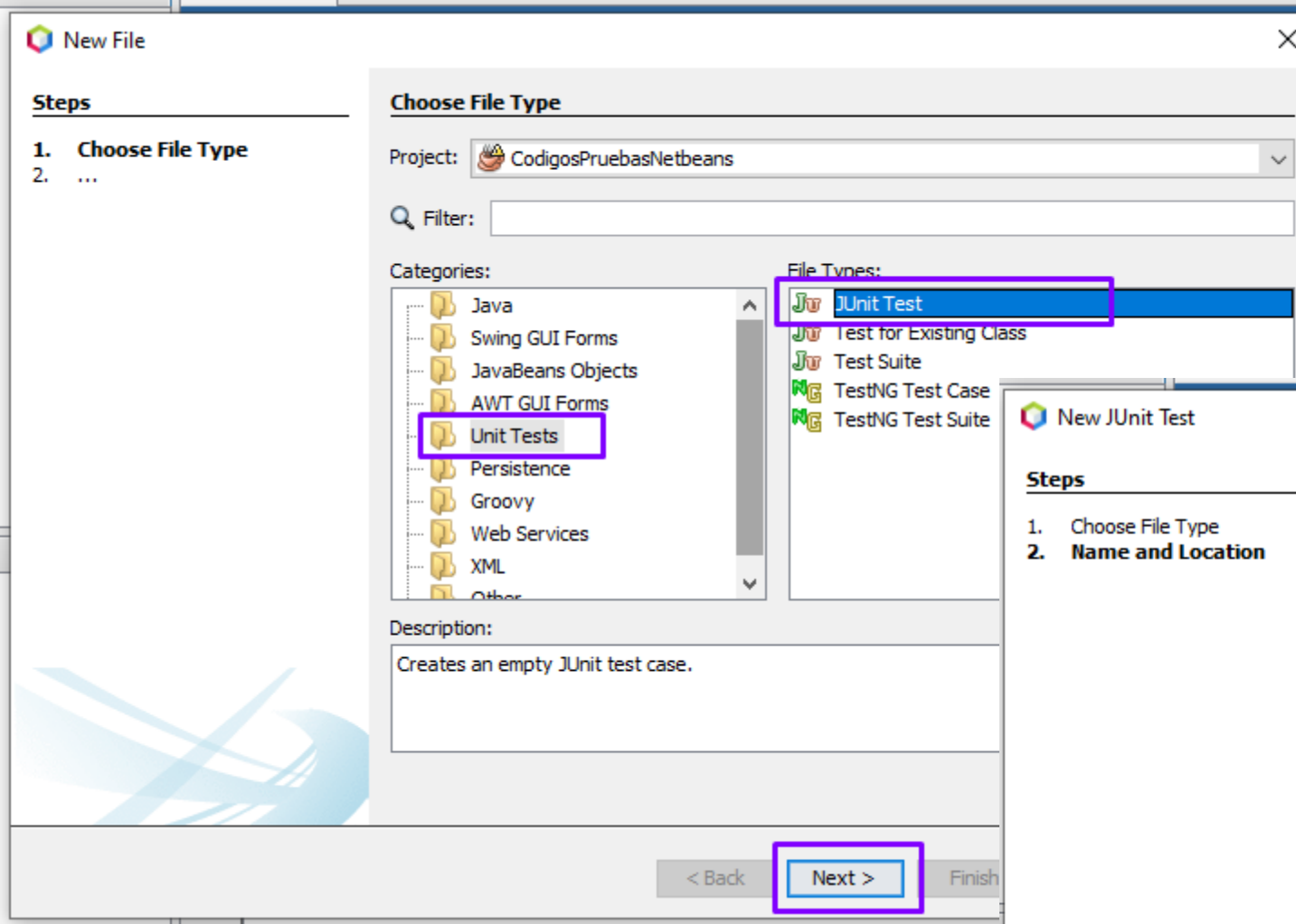


El futuro digital
es de todos

MinTIC



OPCION 2





El futuro digital
es de todos

MinTIC



UNIVERSIDAD
DE ANTIOQUIA

Facultad de Ingeniería



New Test for Existing Class

Steps

1. Choose File Type
2. Existing Class To Test

Existing Class To Test

Class to Test: semana4.BibliotecaArray Browse...

Created Test Class: semana4.BibliotecaArrayTest

Project: CodigosPruebasNetbeans

Location: Test Packages

Created File: MinTIC\Cido2\CodigosPruebasNetbeans\test\semana4\BibliotecaArrayTest.java

Method Access Levels	Generated Code	Generated Comments
<input checked="" type="checkbox"/> Public	<input type="checkbox"/> Test Initializer	<input checked="" type="checkbox"/> Javadoc Comments
<input checked="" type="checkbox"/> Protected	<input type="checkbox"/> Test Finalizer	<input checked="" type="checkbox"/> Source Code Hints
<input checked="" type="checkbox"/> Package Private	<input checked="" type="checkbox"/> Test Class Initializer	
	<input checked="" type="checkbox"/> Test Class Finalizer	
	<input checked="" type="checkbox"/> Default Method Bodies	

< Back Next > Finish Cancel Help

New JUnit Test

Steps

1. Choose File Type
2. Name and Location

Name and Location

Class Name: Prueba

Project: CodigosPruebasNetbeans

Location: Test Packages

Package:

Created File: C:\Users\USUARIO\Dropbox\MinTIC\Cido2\Codigos

Generated Code

☐ Test Initializer

☐ Test Finalizer

☐ Test Class Initializer

☐ Test Class Finalizer

Generated Comments

☒ Source Code Hints

Warning: It is highly recommended that you do not place Jav

< Back

Next >

Referencias

ISQTB. (2018). Foundation Level. International Software Testing Qualifications Board.