

Nombre Alumno / DNI	Ignacio Jesús Medrano Jadraque / 06635576F
Título del Programa	1ºPHE CIBERSECURITY & DIGITAL INTELLIGENCE
Nº Unidad y Título	UNIT 1- PROGRAMMING & CODING
Año académico	2023-2024
Profesor de la unidad	Gabriela García
Título del Assignment	AB FINAL PROGRAMMING & CODING
Día de emisión	18/10/2023
Día de entrega	31/01/2024
Nombre IV y fecha	
Declaración del estudiante	<p>Certifico que la presentación del assignment es completamente mi propio trabajo y entiendo completamente las consecuencias del plagio<sup>1</sup>. Entiendo que hacer una declaración falsa es una forma de mala práctica.</p> <p>Fecha: 31/01/2024</p> <p>Firma del alumno:</p> 

<sup>1</sup> Plagio

El plagio es una forma particular de hacer trampa. El plagio debe evitarse a toda costa y los alumnos que infrinjan las reglas, aunque sea inocentemente, pueden ser sancionados. Es su responsabilidad asegurarse de comprender las prácticas de referencia correctas. Como alumno de nivel universitario, se espera que utilice las referencias adecuadas en todo momento y mantenga notas cuidadosamente detalladas de todas sus fuentes de materiales para el material que ha utilizado en su trabajo, incluido cualquier material descargado de Internet. Consulte al profesor de la unidad correspondiente o al tutor del curso si necesita más consejos.

# MSMK

## AB FINAL PROGRAMMING & CODING

1ºPHE CYBERSECURITY & DIGITAL INTELLIGENCE  
NACHO M.

## **ÍNDICE:**

<b>Introducción.</b> .....	3
<b>Conceptos Básicos.</b> .....	4
Diferencias entre testing y pruebas de código. ....	4
Objetivos y beneficios de realizar pruebas. ....	4
<b>Tipos de pruebas.</b> .....	4
<b>Técnicas de Testing</b> .....	6
<b>Automatización de pruebas</b> .....	7
<b>Bibliografía</b> .....	8

## Introducción.

En el desarrollo de software, el testing y las pruebas de código son fundamentales para asegurar la calidad y fiabilidad del producto. Estas buscan errores para solucionarlos, afianzar la correcta funcionalidad y mejorar la experiencia del usuario. Además, la importancia de estas actividades se debe a que contribuyen a la pronta detección de defectos, facilitan la mantenibilidad del código y ofrecen confianza a los desarrolladores y usuarios.

## Conceptos Básicos.

### Diferencias entre testing y pruebas de código.

El testing es un proceso amplio que consiste en la evaluación del software en su totalidad, mientras que las pruebas de código se centran específicamente en la revisión de fragmentos individuales de código fuente. Las pruebas de código buscan comprobar que cada parte del código funciona como se espera, mientras que el testing se concentra en la operabilidad y la funcionalidad del software en su conjunto.

### Objetivos y beneficios de realizar pruebas.

Los objetivos principales de realizar pruebas son detectar y corregir errores, asegurar la calidad del software, mejorar la documentación del código y facilitar la colaboración entre desarrolladores. Los beneficios incluyen la reducción de costes a largo plazo, la mejora de la satisfacción del usuario y la creación de un código más robusto y mantenible.

## Tipos de pruebas.

Existen distintos tipos de pruebas de software, cada una de ellas tienen como objetivo asegurarse que el código funcione correctamente mediante distintos métodos.

**Pruebas unitarias:** Probar partes individuales del código como pueden ser las funciones o los métodos. Son fáciles de automatizar utilizando herramientas específicas como por ejemplo JUnit para Java.

**Pruebas de integración:** Probar como interactúan entre sí distintas partes del sistema. Ejemplos de este tipo de prueba pueden ser hacer pruebas de API o interactuar con la base de datos que haya conectada. Ejemplos de herramientas comunes son Postman y RestAssured para pruebas de API.

**Pruebas de sistema:** Comprobación de que el sistema cumple con los requisitos requeridos. Selenium es una herramienta muy utilizada para pruebas de interfaz de usuario.

**Pruebas de aceptación:** Este tipo de prueba verifica que el sistema o software cumple con los criterios del usuario final. Cucumber y SpecFlow son comunes en el desarrollo guiado por comportamiento.

**Pruebas de carga y estrés:** Evalúan el rendimiento que tiene el software en condiciones extremas, por ejemplo, un gran número de interacciones simultáneas. JMeter es una herramienta destacada para pruebas de carga.

**Pruebas de rendimiento:** Estas pruebas tienen por objetivo evaluar el rendimiento del software con una carga de trabajo concreta. Una prueba de rendimiento puede comprobar el tiempo de respuesta al procesar muchas solicitudes al mismo tiempo. JMeter además de pruebas de carga permite analizar el rendimiento.

(atlassian, 2024)

(Trans-Ti, 2024)

## Técnicas de Testing

### **TDD:**

Desarrollo Dirigido por Pruebas (Test-Driven Development), técnica de desarrollo de software trata de escribir pruebas automatizadas antes de escribir el código. Su objetivo es asegurar que el código cumple con las especificaciones requeridas y que cualquier cambio posterior no introduce errores no deseados.

### **BDD:**

Desarrollo Dirigido por Comportamiento (Behavior-Driven Development), se centra en el comportamiento del software desde la perspectiva del usuario. Cucumber y Behave son herramientas que facilitan la colaboración entre desarrolladores y no desarrolladores en la definición de comportamientos.

(imaginaformacion, 2024)

## Automatización de pruebas

La automatización consiste en la utilización de herramientas para automatizar la revisión y validación de software. Agiliza el proceso de pruebas, aumenta la cobertura y facilita la detección de defectos. Herramientas como Selenium, JUnit y Appium son usadas para la automatización de pruebas.

(atlassian, 2024)

## Casos de uso

Es una técnica para la identificación de posibles requisitos en la creación de un nuevo sistema o en la actualización de software. Cada caso de uso ofrece uno o más escenarios que la forma en que el sistema debe interactuar con el usuario o con otros sistemas para lograr objetivos específicos.

(juntadeandalucia, 2024)

## Conclusión

El testing y las pruebas de código son elementos fundamentales en el desarrollo de software. Contribuyen a la entrega de productos de calidad, reducen costes a largo plazo y generan confianza en desarrolladores y usuarios. El uso de técnicas como TDD, BDD y la automatización mejora la eficacia del proceso. En un entorno donde la calidad del software es esencial, el testing y las pruebas de código son indispensables para asegurar el éxito del desarrollo de software.



## **Bibliografía**

atlassian, 2024. *atlassian*. [En línea]

Available at: <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>

[Último acceso: 2024].

imaginaformacion, 2024. *imaginaformacion*. [En línea]

Available at: <https://imaginaformacion.com/tutoriales/que-es-testing-tdd-y-bdd>

[Último acceso: 2024].

juntadeandalucia, 2024. *juntadeandalucia*. [En línea]

Available at: <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>

[Último acceso: 2024].

Trans-Ti, 2024. *Trans-Ti*. [En línea]

Available at: <https://trans-ti.com/2020/11/23/conoce-las-tecnicas-de-software-testing/>

[Último acceso: 2024].