


Assignment brief A.B.

PORTADA

Nombre Alumno / DNI	Daniel Bautista Suarez 09075134R
Título del Programa	Cybersecurity
Nº Unidad y Título	UNIT 2- Programing & coding
Año académico	2023-2024
Profesor de la unidad	Gabriela García
Título del Assignment	AB FINAL
Día de emisión	20/01/2024
Día de entrega	30/01/2024
Nombre IV y fecha	x
Declaración del estudiante	<p>Certifico que la presentación del assignment es completamente mi propio trabajo y entiendo completamente las consecuencias del plagio. Entiendo que hacer una declaración falsa es una forma de mala práctica.</p> <p>Fecha: 30/01/2024</p>  <p>Firma del alumno:</p>

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.

Informe de Testing y Pruebas de Código

Introducción:

- El testing es importante a un alto nivel, ya que la pruebas de software son necesarias para detectar los héroes de dicho software y probar si el software cumple con los requisitos del cliente.

Conceptos Básicos:

- **Testing:** Es el método que se utiliza para verificar si al diseñar un producto digital este cumple con los requisitos esperados y se encuentra libre de errores.
- **Pruebas de código:** Estos sirven para verificar la calidad del software y encontrar errores de dicho código, como que vaya más fluido.
- El "testing" en el desarrollo de software es un término amplio que incluye diversas formas de evaluación como pruebas de usabilidad, seguridad, compatibilidad y rendimiento. Por otro lado, las "pruebas de código" son un aspecto específico del testing que se enfoca en la verificación del código fuente, asegurándose de que funcione correctamente a través de pruebas unitarias, de integración y de sistema. Mientras las pruebas de código se centran en aspectos técnicos y son realizadas generalmente por desarrolladores, el testing abarca una gama más amplia de evaluaciones para garantizar la calidad total del software.
- **Beneficios:**
 - Ahorra dinero y muchos dolores de cabeza.
 - Previene emergencias corporativas catastróficas.
 - Inspira confianza del cliente.
 - Mantiene una gran experiencia de usuario.

- Trae más ganancias.
- Aumenta la satisfacción del cliente.

Tipos de Pruebas:

- **Pruebas Unitarias:** Es el instrumento utilizado para validar un fragmento de código fuente. Los desarrolladores aíslan una línea del lenguaje codificado para saber si el sistema está operando correctamente en una función. Se utilizan herramientas como Live Unit Testing e IntelliTest.
- **Pruebas de Integración:** Estas sirven para verificar que los distintos módulos o servicios utilizados por tu aplicación funcionan bien en conjunto. Se utilizan herramientas como Jenkins, Travis CI y AWS.
- **Pruebas de sistema:** Es un tipo de prueba de software que realiza comprobaciones del sistema en su conjunto. Consiste en integrar todos los módulos y componentes individuales del software que has desarrollado, para comprobar si el sistema funciona conjuntamente como se esperaba. Se utilizan herramientas como TestFlight Beta Testing y Crashlytics Beta.
- **Pruebas de aceptación:** Consiste en descubrir si el software que ha creado no es lo que el cliente y los usuarios esperaban. Se utilizan herramientas como Canoo y Concordion
- **Pruebas de carga:** Se pueden realizar internamente utilizando hardware y software dentro de su propia red o se pueden realizar externamente con sistemas de prueba de terceros. Se utilizan herramientas como Performance Analyzer.
- **Pruebas de estrés:** es un tipo de prueba que se utiliza para determinar los límites del sistema, app o web, sometiéndose a un alto nivel de concurrencia de usuarios en simultáneo, con el objetivo de encontrar su punto de inflexión, ya sea a nivel de aplicación como de su infraestructura. Se utilizan herramientas como Selenium.

Técnicas de testing:

- **TDD:** Es un enfoque de programación que se utiliza durante el desarrollo de software en el que se realizan pruebas unitarias antes de escribir el código.
- **BDD:** Es una metodología ágil de desarrollo software en la que una aplicación se documenta y diseña en función del comportamiento que los usuarios esperan experimentar al interactuar con ella.
- **ATDD:** Es una metodología de desarrollo de software, a menudo asociada a las metodologías ágiles, que fomenta la colaboración entre los desarrolladores, los probadores y las partes interesadas del negocio.

Automatización de pruebas:

- Las pruebas automatizadas consisten en la aplicación de herramientas de software para automatizar el proceso manual de revisión y validación de un producto de software que lleva a cabo una persona.
- Algunas herramientas que se utilizan para la automatización Serenity(Java) Cypress(JavaScript), Robot Framework(Python), Galen Framework (JavaScript/Java), Gauge(Java/Ruby/C#), Citrus Framework(Java), Karate-DSL(Java).

Casos de Uso y Ejemplos:

- En el desarrollo de una aplicación móvil para gestionar finanzas, se utilizan pruebas unitarias para comprobar funciones individuales como el cálculo del saldo total o la validación de transacciones. Estas pruebas aseguran que cada componente del código funcione correctamente de forma aislada.
- Para un sistema de e-commerce, las pruebas de integración son fundamentales. Estas pruebas se centran en la interacción entre el carrito de compras y el sistema de pagos, verificando que los datos de las transacciones se manejan adecuadamente entre los distintos módulos del sistema.

Conclusión:

- Las pruebas y el testing son esenciales para garantizar la calidad y confiabilidad del software, asegurando que los productos no solo funcionen correctamente, sino que también cumplan con las expectativas y necesidades de los usuarios finales.

Referencias:

- <https://es.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/>
- <https://www.nimblework.com/es/agile/pruebas-de-aceptacion/#:~:text=Las%20pruebas%20de%20aceptaci%C3%B3n%20consisten,que%20se%20especificaban%20los%20requisitos.>
- <https://www.loadview-testing.com/es/pruebas-de-carga/#:~:text=Las%20pruebas%20de%20carga%20se,la%20demanda%20en%20un%20sistema.>
- <https://www.loadview-testing.com/es/pruebas-de-carga/#:~:text=Las%20pruebas%20de%20carga%20se,la%20demanda%20en%20un%20sistema.>
- <https://www.pragma.co/es/blog/conoce-cinco-herramientas-para-integracion-y-entrega-continua-con-devops>
- <https://hiberus.com/crecemos-contigo/como-automatizar-pruebas-funcionales-con-selenium/#:~:text=Selenium%20es%20una%20herramienta%20de,posibilidades%20de%20uso%20est%C3%A1n%20abiertas.>
- [https://www.arsys.es/blog/behavior-driven-development#:~:text=Behavior%20Driven%20Development%20\(BDD\)%2C,experimentar%20al%20interactuar%20con%20ella](https://www.arsys.es/blog/behavior-driven-development#:~:text=Behavior%20Driven%20Development%20(BDD)%2C,experimentar%20al%20interactuar%20con%20ella)
- [https://www.arsys.es/blog/behavior-driven-development#:~:text=Behavior%20Driven%20Development%20\(BDD\)%2C,experimentar%20al%20interactuar%20con%20ella](https://www.arsys.es/blog/behavior-driven-development#:~:text=Behavior%20Driven%20Development%20(BDD)%2C,experimentar%20al%20interactuar%20con%20ella)
- <https://josepablosarco.wordpress.com/2018/05/02/top-7-frameworks-de-pruebas-automatizadas/>
- <https://www.hiberus.com/crecemos-contigo/testing-fase-de-testeo-de-software/#:~:text=Por%20qu%C3%A9%20el%20testing%20es,un%20producto%20de%20buena%20calidad.>
- <https://www.hiberus.com/crecemos-contigo/testing-fase-de-testeo-de-software/#:~:text=Por%20qu%C3%A9%20el%20testing%20es,un%20producto%20de%20buena%20calidad.>
- <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/#:~:text=El%20testing%20o%20prueba%20de,un%20determinado%20programa%20o%20aplicaci%C3%B3n.>
- <https://www.encora.com/es/blog/7-beneficios-de-hacer-software-testing-y-pruebas-de-calidad-qa>
- <https://www.testingit.com.mx/blog/pruebas-unitarias-de-software#:~:text=Una%20prueba%20unitaria%20de%20software,funci%C3%B3n%2C%20proceso%20o%20actividad%20espec%C3%ADfica.>

- <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing#:~:text=Las%20pruebas%20de%20integraci%C3%B3n%20verifican,conjunto%20y%20seg%C3%BAn%20lo%20esperado.>
-