



Testing de código con Postman (15%)

Autor: Jimmis J. Simanca

Tecnología en Desarrollo de Software, Corporación Universitaria Uniremington

Asignatura: Lenguaje de Programación 3

Director: Milton Javier Mateus Hernández

22 de noviembre de 2024

Tabla de contenido

Diligenciar la tabla Comparativa de los conceptos básicos (Valor 5%)	3
Tabla 1.....	3
Referencias	7

Diligenciar la tabla Comparativa de los conceptos básicos (Valor 5%)

Decir la diferencia entra las pruebas Funcionales y las pruebas de Software No funcionales

llenar la tabla con las ventajas, desventajas y las herramientas o aplicaciones utilizadas para realizar pruebas funcionales de software

Tabla 1

Tipos de prueba FUNCIONALES	Ventajas	Desventajas	Herramientas o aplicaciones
Pruebas unitarias	De acuerdo con (AWS, s.f.) Son pequeñas pruebas de funcionalidad que se realiza al código de manera individual, esto nos da la ventaja de detectar de manera eficiente y eficaz detección de errores, esto nos permite solucionar fallos de manera más rápida	No detectan todas las fallas de a la aplicación o el software, requieren mucho tiempo ya que al ser pruebas unitarias se debe probar cada método o función que tiene el código de manera manual, no se prueba la aceptación.	<ul style="list-style-type: none"> • JUnit • Cactus • EasyMock • Mockito • MockEjb • Spring Test • Jetty • Dumbster
Pruebas de integración	Estas pruebas nos permiten probar si los	Una de sus desventajas es que es muy costos de	Selenium JIRA

	componentes individuales del sistema funcionan bien cuando son combinados dándonos como ventaja que podamos verificar que todas las partes del sistema tienen funcionalidad, detección de fallas en etapas tempranas en el desarrollo, asegura que los componentes se integren y mejora la calidad y confiabilidad del software.	implementar se requiere de habilidades muy específicas se debe documentar la información y son complicadas de realizar.	Postman: TestComplete: Jenkins:
Pruebas de sistema	Según una de sus ventajas es que no se necesitan conocimientos en programación, revela fallos o errores que las pruebas unitarias no pueden encontrar	Probar el sistema lleva mucho tiempo. Requiere probadores altamente cualificados. Es un reto para los proyectos grandes y complejos. Los probadores no tienen acceso al código fuente del software.	Selenium HP Quality Center (Micro Focus ALM): IBM Rational Functional Tester TestComplete Zephyr
Pruebas de aceptación	<ul style="list-style-type: none"> • Se conocen todas las funciones y características que se probarán. 	<ul style="list-style-type: none"> • Requiere recursos significativos y planificación. 	TestRail: Zephyr: Selenium:

	<ul style="list-style-type: none"> • Se conocen los detalles de las pruebas y se pueden medir. • Las pruebas se pueden automatizar permitiendo las pruebas de regresión. • El progreso de las pruebas se puede medir y monitorear. • El criterio de aceptabilidad es conocido. (Library, s.f.) 	<ul style="list-style-type: none"> • Las pruebas pueden ser una reimplementación de las pruebas del sistema. • Puede que no se encuentren los defectos subjetivos, sino solo los que se están buscando. (Library, s.f.) 	<p>QTP (Quick Test Professional):</p> <p>JIRA:</p>
Pruebas de regresión	<p>Garantizan funciones correctas:</p> <p>Reducen riesgos:</p> <p>Mantienen calidad</p> <p>Aceleran lanzamientos:</p> <p>defectos:</p>	<p>Requieren mucho tiempo: Las pruebas manuales pueden ser lentas y necesitan probadores bien capacitados.</p> <p>Cuantificación difícil:</p> <p>Necesitan pruebas exhaustivas, lo que puede causar retrasos.</p>	<p>Selenium</p>

		Identificación de problemas: A veces, es difícil encontrar casos de prueba que revelen problemas.	
		Errores adicionales: El conjunto de pruebas puede encontrar más errores que los introducidos.	
		Riesgo de actualizaciones: Si no se encuentran defectos, puede retrasar actualizaciones o causar quejas de clientes.	

Enlace video explicativo: <https://youtu.be/DViRNmhbh-M?si=TTV6geeyl4yQp2n4>

Proyecto en github: <https://github.com/JimsimroDev/productodApi.git>

Referencias

AWS. (s.f.). AWS. Obtenido de AWS: <https://aws.amazon.com/es/what-is/unit-testing/>

Library. (s.f.). *Library*. Obtenido de Library: <https://1library.co/article/pruebas-aceptaci%C3%B3n-pruebas-despliegue.y8gd23m5>

