

SENA

Centro De Electricidad Electrónica Y Telecomunicaciones (CEET)

Julio ,2021

BOGOTA D.C

Ficha:2057746

Guisli Nataly Ramírez Sanabria

Tecnólogo Análisis Y Desarrollo De Sistemas De Información (ADSI)

Realizar la lectura del documento “Pruebas de Software”. Luego, crear un mapa mental acerca de las diferentes pruebas de software existentes y del concepto de calidad de software. Consultar que es un mapa mental y elaborarlo, una vez finalizado el mapa mental exportar una imagen o generar un enlace público donde se pueda consultar.



3.2 – Actividades de transferencia del conocimiento (Conceptualización y teorización) Evidencia: Documento con normas APA Materiales: Word

1. ¿A Qué Hace Referencia El Término Calidad, En Desarrollo De Software?, Argumente.

Rta: E cuando un sistema, componente o proceso cumple con los requisitos planteados y con las expectativas del cliente el usuario.

2. ¿Cuáles Son Las Dimensiones De Calidad Del Software?, Explique Cada Dimensión.

Rta:

Calidad del desempeño. Presenta, las funciones y las características especificadas en los requerimientos.

Calidad de las características. Satisfacción del cliente en cuanto a lo requerido

Confiabilidad. Está disponible sin errores y sin fallas.

Conformidad. Es acorde con los estándares locales e internacionales.

Durabilidad. facilidad el mantenimiento y corrección.

Servicio. El mantenimiento y corrección se pueden hacer en un tiempo breve.

Estética. aceptable por los usuarios en general.

Percepción. Recibe buenos comentarios por parte de los usuarios.

3. ¿Cuáles Son Los Factores De Calidad Del Software?, Explique Cada Factor.

Rta: **Revisión Del Producto:** En cuanto facilidad de recibir mantenimiento, Flexibilidad

Transición Del Producto: en los siguientes factores Funcionalidad Portabilidad
Reusabilidad

Operación Del Producto: en Confiabilidad, Usabilidad, Eficiencia, Corrección, Integridad

4. ¿Qué Aspectos Se Deben Tener En Cuenta Para Implementar Una Prueba De Software Y Por Qué Es Importante Tenerlos En Cuenta?

Rta:

- Analizar los requerimientos de desarrollo de **software**
- Identificar las funcionalidades nuevas a probar
- Identificar las funcionalidades de sistemas existentes que deben probarse.
- Definir la estrategia de **pruebas**.
- Definir los criterios de inicio, aceptación y suspensión de pruebas
- Identificar los entornos (ambientes) requeridos
- Determinar necesidades de personal y entrenamiento
- Establecer la metodología y procedimientos de prueba
- Elaborar la planificación de las pruebas
- Identificar los riesgos y definir planes de respuesta

5. ¿Cuáles Son Las Etapas Para Implementar Una Prueba De Software?

Rta:

Pruebas De Componentes: También llamadas pruebas unitarias es la primera etapa donde ya se cuenta con los componentes del sistema desarrollado donde se prueba que los componentes de acuerdo con lo esperado los cuales dependen del lenguaje utilizado ya puede ser clases, módulos y unidades normalmente para este tipo de pruebas se utilizan herramientas de depuración donde se revisa línea a línea el comportamiento del código.

Pruebas De Integración: Se realizan una vez concluidas las pruebas de componentes se realizan pruebas con los componentes integrados para verificar que su funcionamiento de manera conjunta a diferencia de las pruebas unitarias está requieren una mejor estructura y organización requiere al menos un plan de pruebas.

Pruebas De Sistema: Sirve para validar que todas las funciones y componentes del sistema funcionen de manera conjunta y deseada. Se ejecuta después de las pruebas de componentes y de integración se diferencia de las dos anteriores por que esta se realiza desde el punto de vista de los usuarios donde se visualizan la implementación de los requerimientos e incluso aquellos pequeños que afectan su funcionamiento.

Pruebas De Aceptación: donde el usuario final valida el producto tiene como objetivo validar que el desarrollador entendió e implemento lo que el usuario solicito. Además de la aceptación del contrato, del usuario, operativa y de campo

Pruebas Funcionales: valida que los requerimientos funcionales, la funcionalidad del sistema, configuración, instalación / desinstalación, caída y recuperación, seguridad, control de acceso y datos se cumplan y funciones de acuerdo con lo esperado.

Pruebas De Desempeño: validan que se cumpla todos los requerimientos de rendimiento del aplicativo sometiendo a pruebas de concurrencia, estrés, volumen y resistencia con herramientas especiales que simulen estas cargas poniendo a prueba su funcionamiento.

Pruebas De Usabilidad O Interfaz De Usuario: validan que las interfaces proporcionan al usuario el confort y la navegación necesario para todas las funcionalidades de la aplicación además de evitar que cometa errores por ejemplo que ingrese letras en campos donde solo se puede aceptar números.

6. Complete La Siguiete Tabla Con Los Tipos De Pruebas De Software. Tipo De Prueba Descripción Ventajas Y Desventajas Pruebas De Caja Blanca Pruebas De Caja Negra Pruebas Unitarias Pruebas De Stress Pruebas De Regresión Pruebas De Humo Pruebas De Seguridad Y Control De Acceso Pruebas De Gui Pruebas Tdd Pruebas Bdd

Nombre	Descripción	Ventajas	Desventajas
Pruebas De Caja Blanca	se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente	Optimización de código al encontrar errores ocultos. También podemos automatizar fácilmente los casos de prueba para el cuadro Blanco. La prueba es más exhaustiva, porque generalmente verificamos todas las rutas de código. Las pruebas pueden comenzar temprano en el proyecto, incluso si la GUI no está disponible.	Las pruebas de caja blanca pueden ser complejas y costosas. Los desarrolladores que suelen ejecutar casos de prueba de caja blanca lo detestan. Por lo tanto, los desarrolladores no prueban la caja blanca en detalle, lo que puede provocar errores de producción. Las pruebas de caja blanca también requieren herramientas profesionales y conocimientos profundos de programación e implementación. La prueba de caja blanca lleva mucho tiempo, lleva tiempo probar completamente las aplicaciones más grandes.
Pruebas De Caja Negra	es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación	A diferencia de la prueba de la caja blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba 'caja negra' tiende a ser aplicada en	El problema con las pruebas de caja negra no suele estar en el número defunciones proporcionadas por el módulo (que siempre es un

	o escenarios de ejecución internos en el software	posteriores fases de prueba. Ya que la prueba de la caja negra intencionadamente ignora la estructura de control, concentra su atención En el dominio de la información.	número muy limitado); sino en los datos que pasan a estas funciones. El conjunto de datos posibles suele ser muy amplio (por ejemplo, un entero)
Pruebas Unitarias	es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo, en diseño estructurado o en diseño funcional una función o un procedimiento	Sin duplicación de código en las pruebas Puede combinar diferentes constructores para pruebas complejas Interfaz limpia Los gráficos de objetos grandes de los constructores pueden influir en el resultado de la prueba	Los constructores deben estar escritos y mantenidos El código del constructor también debe probarse El código del constructor necesitará un cuidado y una atención constantes
Pruebas De Estrés	consiste en probar los límites que un sistema puede soportar.	la prueba de estrés puede medir la capacidad de nuestro software y si es efectivo para lo que necesita el cliente	es un proceso muy estructurado y debe hacerse con organización y un plan previo
Pruebas De Regresión	son cualquier tipo de pruebas de software con el objeto de descubrir errores, carencias de funcionalidad, o divergencias funcionales con	Nos ayuda a asegurarnos de que cualquier cambio, como la corrección de errores o cualquier mejora en el módulo o la aplicación, no haya	Si la prueba de regresión se realiza sin usar herramientas automatizadas, puede ser muy tedioso y lento porque aquí

	<p>respecto al comportamiento esperado del software.</p>	<p>afectado el código probado existente. Asegura que los errores encontrados anteriormente NO sean creables. La prueba de regresión se puede hacer usando las herramientas de automatización Ayuda a mejorar la calidad del producto.</p>	<p>ejecutamos el mismo conjunto de casos de prueba una y otra vez. Se requiere una prueba de regresión incluso cuando se realiza un cambio muy pequeño en el código porque esta pequeña modificación puede traer problemas inesperados en la funcionalidad existente.</p>
Pruebas De Humo	<p>Son una revisión rápida de un producto de software para comprobar que funciona y no tiene defectos evidentes que interrumpan la operación básica del mismo.</p>	<p>Ayudan a detectar los errores en las primeras etapas de la prueba. Permiten verificar que los problemas corregidos en la versión anterior NO estén impactando en las principales funcionalidades de la aplicación. Se requiere un número muy limitado de casos de prueba para llevarla adelante. Las pruebas de humo pueden llevarse a cabo en poco tiempo.</p>	<p>No realiza una prueba detallada de la aplicación. Al ser una prueba no exhaustiva, no seremos capaces de encontrar todos los otros problemas críticos en la aplicación</p>
Pruebas De Seguridad Y Control De Acceso	<p>las pruebas de seguridad son diversas y se orientan a varios</p>	<p>Control de Entradas y Salidas Mayor Seguridad y Control del Público</p>	<p>La seguridad absoluta es imposible y la seguridad</p>

	<p>ámbitos, especialmente en lo relativo a asegurar el funcionamiento y disponibilidad de los servicios web y contenidos publicados. En general nos referimos a Sitios Web cuyo contenido es público y que no incluye información estratégica o de seguridad institucional, por este motivo, la correcta configuración de servidores y servicios debe incluir verificación</p>	<p>Ahorro en Costos de Personal Capacidad de Diferir Pagos del Costos del Proyecto Rápido Retorno de la Inversión Disminución en Tiempo de Registro Mejoramiento en la Productividad del Personal Permitir/Restringir la Apertura de Puertas Valorización Monetaria de la Edificación Valor Agregado en Modernización</p>	<p>informática es un conjunto de técnicas encaminadas a obtener altos niveles de seguridad en los sistemas informáticos. En los equipos de cómputo más desactualizados un antivirus realmente efectivo puede ser muy pesado, puede hacerlos más lentos y ocupar mucho espacio en memoria. Los requisitos para la creación de contraseñas son cada vez más complejos, la mayoría de los sitios web requieren inicio de sesión, y el cambio de contraseñas con frecuencia se ha vuelto obligatorio en muchos lugares de trabajo, recordarlas en ocasiones es muy difícil.</p>
Pruebas GUI	<p>Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo</p>	<p>la ventaja es que es agradable a la vista además que debe hacerse de manera que se pueda entender fácilmente</p>	<p>Utilizando una interfaz implica mover un puntero a un icono o algún otro elemento de control, haciendo</p>

	para encontrar fallas y vulnerabilidades en aplicaciones web	ventajas Relativamente fáciles de aprender y utilizar.	clic y luego esperando que algo suceda Control Recursos informáticos Falta de capacidades de secuencias de comandos
Pruebas TDD	El propósito del desarrollo guiado por pruebas es lograr un código limpio que funcione. La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.	1.- Mayor calidad 2.- Diseño enfocado en las necesidades 3.- Mayor simplicidad en el diseño 4.- El diseño se va adaptando al entendimiento del problema 5.- Mayor productividad 6.- Menos tiempo invertido en debugging de errores	1.- Interfaz de usuario 2.- La Base de datos 3.- Errores no identificados 4.- Perder la visión general 5.- Pronunciada curva de aprendizaje
Pruebas BDD	behavior-driven development o desarrollo guiado por el comportamiento , es un proceso de desarrollo de software que surgió a partir del desarrollo guiado por pruebas .123 El desarrollo guiado por el comportamiento combina las	Ideal para principiantes gracias al lenguaje ubicuo, que no requiere conocimientos previos. Las especificaciones mal redactadas dificultan el trabajo de los desarrolladores. Mejor comunicación entre desarrolladores, stakeholders y gestores de la calidad.	Los casos de prueba son una documentación viviente que puede adaptarse fácilmente. La conversión a un flujo de trabajo BDD supone un esfuerzo añadido si se usan códigos heredados. La prioridad es el confort del usuario al manejar el software.

	técnicas generales y los principios del DGP, junto con ideas del diseño guiado por el dominio y el análisis y diseño orientado a objetos para proveer al desarrollo de software	Como integra a diversas partes interesadas, el proceso de desarrollo se alarga	
--	---	--	--

Rta:

7. Consulte Acerca De Los Estándares Para La Documentación De Las Pruebas De Software.

Rta: Documentos requeridos por

IEEE 829: La documentación de la prueba de software es el elemento vital que eleva las actividades experimentales al nivel de una prueba de software. 1 organizaciones internacionales como IEEE e ISO han publicado estándares para la documentación de pruebas de software. Donde se recopilan documentos de suma importancia.

Los documentos son:

- Plan de prueba maestro (MTP)
- Plan de prueba de nivel (LTP)
- Diseño de prueba de nivel (LTD)
- Caso de prueba de nivel (LTC)
- Procedimiento de prueba de nivel (LTP)
- Registro de prueba de nivel (LTL)
- Informe de anomalías (AR)
- Informe de estado de prueba intermedia de nivel (LITSR)
- Informe de prueba de nivel (LTR)
- Informe de prueba maestro (MTR)

8. ¿Consulte Acerca De Las Diferentes Herramientas Para Realizar Pruebas De Software?

Rta:

Selenio

Selenio es como un nombre familiar para las pruebas automatizadas. Es una herramienta portátil de código abierto y proporciona pruebas funcionales de varios módulos de aplicaciones web en plataformas y navegadores. Admite la ejecución de texto en paralelo, lo que ahorra mucho tiempo de prueba. Selenium admite muchos lenguajes como Java, Ruby, C #, Python, etc. Selenium ejecuta una herramienta de reproducción para registrar pruebas funcionales, por lo que no es necesario que aprenda el lenguaje de programación de pruebas. El selenio es muy fácil de configurar; tienes que agregar el complemento específico del navegador. Automatiza los navegadores web proporcionando una velocidad y un nivel de eficiencia diferentes que no es posible mediante pruebas manuales.

Estudio de prueba

Telerik Test Studio facilita las pruebas automatizadas. Incluye código y capacidades sin código para probar funcional, API y carga. Telerik tiene muchas herramientas de interfaz de usuario, y estoy seguro de que habrías oído hablar de Fiddler, un software de depuración.

Testim

Testim es una herramienta de prueba de software automatizada inteligente que utiliza el aprendizaje automático para acelerar el diseño, la ejecución y el mantenimiento de casos de prueba automatizados. Los casos de prueba se pueden ejecutar en múltiples plataformas, incluidos dispositivos móviles. usa anotaciones para encontrar inconsistencias y errores en el sistema. Los errores que se registran se pueden reproducir automáticamente simplemente haciendo clic en la prueba nuevamente. El rastreador de errores Testim se usa para compartir capturas de pantalla anotadas y ver detalles de errores.

Herramientas de prueba de carga

Las herramientas de prueba de carga ayudan a determinar el tiempo necesario para cargar un sitio, una aplicación o páginas. Estas herramientas evalúan la rendimiento del sitio web y ayudar a garantizar un acceso sin problemas de ida y vuelta al sitio web.

LoadRunner

LoadRunner es adecuado para evaluar el rendimiento de un sitio web o una aplicación web en condiciones de alta carga. Puede simular miles de usuarios que utilizan el software para registrar y analizar el rendimiento de la aplicación. Proporciona análisis y seguimiento fáciles de usar. Funciona en todos los entornos empresariales más importantes y admite diferentes tipos de aplicaciones y protocolos. LoadRunner puede ejecutarse de forma independiente o agrupada por varios usuarios a través de la versión empresarial. Tiene un panel interactivo a través del cual los usuarios se pueden administrar fácilmente.

WebLoad

Herramienta muy poderosa con capacidades de scripting, WebLoad puede probar escenarios de casos de prueba complejos. Se puede utilizar para probar aplicaciones independientes y en la nube, y es compatible con todos los principales protocolos web, aplicaciones empresariales y tecnologías. Con Webload, es fácil crear y cargar escenarios de prueba, ya que la herramienta proporciona más de 80 tipos de gráficos e informes para identificar y trabajar en problemas de rendimiento en el sistema. También permite que la instalación ejecute casos de prueba de rendimiento desde la nube utilizando Amazon EC2 para generar carga. Webload tiene un componente llamado PMM, que puede recopilar estadísticas del lado del servidor durante las ejecuciones de prueba, lo que proporciona datos adicionales para realizar análisis de causa raíz.

Blazemeter

Blazemeter se utiliza para ejecutar casos de prueba de rendimiento en aplicaciones móviles, API o aplicaciones web para evaluar el rendimiento en cualquier etapa del desarrollo de la aplicación. Asegura la entrega de software de alto rendimiento al final. Puede crear una nueva configuración de prueba obteniendo nuevos datos de Google Analytics e integrándolos.

De la misma manera, puede probar múltiples ubicaciones geográficas e incorporarlas en un solo informe. La herramienta es 100% compatible con código abierto y lista para empresas, proporciona automatización de pruebas inteligente y es ideal para un entorno de prueba continuo.

JMeter

JMeter de Apache es una herramienta de prueba de código abierto que se utiliza para probar el rendimiento de sitios web y aplicaciones web dinámicas. Una persona con muy

pocos conocimientos técnicos también puede interpretar los resultados proporcionados por JMeter.

Jmeter

Funciona en varios tipos de servidores como SOAP, FTP, LDAP, JDBC, TCP, etc., para pruebas de carga y rendimiento. Los planes de prueba se almacenan en formato XML y se generan mediante un editor de texto. JMeter también se utiliza para pruebas funcionales y automatizadas.

También puedes echarle un vistazo herramientas de prueba de carga basadas en la nube para realizar pruebas de estrés y capacidad.

Herramientas de gestión de pruebas

Las herramientas de gestión de pruebas ayudan a gestionar todo el ciclo de pruebas de un producto. Una herramienta de gestión de pruebas útil debería poder integrarse con otros marcos de automatización y Integración CI.

Radiografía

Radiografía es una de las herramientas de gestión de pruebas preferidas para pruebas manuales y automatizadas. Proporciona la estructura adecuada para organizar y clasificar conjuntos de pruebas y proporciona resultados de prueba eficientes en menos tiempo. Se integra perfectamente con marcos de prueba como Jira, Selenium, Junit, etc. Puede establecer condiciones previas personalizadas que se pueden reutilizar y asociar con diferentes pruebas. Xray facilita las integraciones de CI con Bamboo, Jenkins y proporciona una trazabilidad perfecta y una correspondencia entre requisitos, pruebas, errores y ejecución. También tiene la posibilidad de configurar entornos de prueba, planes de prueba y la generación de informes integrados.

9. ¿Qué Es Jmeter?, Explique Sus Principales Características Y Funcionalidades.

Rta: JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios, con énfasis en aplicaciones web

Características generales de Apache JMeter

Una GUI amigable.

Plataforma independiente.

Multi-threading.

El resultado de la prueba se puede visualizar en un formato diferente como gráfico, tabla, árbol y archivo de registro.

Altamente extensible. ...

Estrategia de pruebas múltiples.

Bibliografías

- https://es.m.wikipedia.org/wiki/Pruebas_de_software
- https://qdata.com.co/servicios/capitacion-en-pruebas-de-software/?gclid=CjwKCAjwr56IBhAvEiwA1fuqGn4Z5stDmpb2vjJ19xNqKCmLSiDSh1PUD6rQgBvMEaPPQZe3FZJiZhoCA5oQAvD_BwE
- <http://consejosdelguru.blogspot.com/2020/06/ventajas-y-desventajas-de-las-pruebas.html?m=1>