

SENA

Centro De Electricidad Electrónica Y Telecomunicaciones (CEET)

agosto ,2021

BOGOTA D.C

Ficha:2057746

Diego Alejandro Caceres Slazar

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:** Es 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:** 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 siguiente 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

Rta:

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	A diferencia de la prueba de la caja	El problema con las pruebas de caja

	<p>funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software</p>	<p>blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba 'caja negra' tiende a ser aplicada en 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.</p>	<p>negra no suele estar en el número defunciones proporcionadas por el módulo (que siempre es un 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)</p>
Pruebas Unitarias	<p>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</p>	<p>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</p>	<p>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</p>
Pruebas De Estrés	<p>consiste en probar los límites que un sistema puede soportar.</p>	<p>la prueba de estrés puede medir la capacidad de nuestro software y si es efectivo para</p>	<p>es un proceso muy estructurado y debe hacerse con organización y un plan previo</p>

		lo que necesita el cliente	
Pruebas De Regresión	son cualquier tipo de pruebas de software con el objeto de descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software.	Nos ayuda a asegurarnos de que cualquier cambio, como la corrección de errores o cualquier mejora en el módulo o la aplicación, no haya 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.	Si la prueba de regresión se realiza sin usar herramientas automatizadas, puede ser muy tedioso y lento porque aquí 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.
Pruebas De Humo	Son una revisión rápida de un producto de software para comprobar que funciona y no tiene defectos evidentes que	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	No realiza una prueba detallada de la aplicación. Al ser una prueba no exhaustiva, no seremos capaces de encontrar todos los otros

	<p>interrumpan la operación básica del mismo.</p>	<p>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>problemas críticos en la aplicación</p>
<p>Pruebas De Seguridad Y Control De Acceso</p>	<p>las pruebas de seguridad son diversas y se orientan a varios 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</p>	<p>Control de Entradas y Salidas Mayor Seguridad y Control del Público 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>La seguridad absoluta es imposible y la seguridad 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</p>



	servidores y servicios debe incluir verificación		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.
Pruebas GUI	Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web	la ventaja es que es agradable a la vista además que debe hacerse de manera que se pueda entender fácilmente ventajas Relativamente fáciles de aprender y utilizar.	Utilizando una interfaz implica mover un puntero a un icono o algún otro elemento de control, haciendo 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	1.- Mayor calidad 2.- Diseño enfocado en las necesidades 3.- Mayor simplicidad en el diseño 4.- El diseño se va adaptando al	1.- Interfaz de usuario 2.- La Base de datos 3.- Errores no identificados 4.- Perder la visión general

	pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.	entendimiento del problema 5.- Mayor productividad 6.- Menos tiempo invertido en debugging de errores	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 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	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. Como integra a diversas partes interesadas, el proceso de desarrollo se alarga	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.

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

**Rta:**

**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.

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:** TestRail

TestRail es una herramienta de gestión de casos de prueba basada en la web que se puede configurar y utilizar fácilmente con la nube o la configuración local. Es altamente escalable y personalizable. Puede ver información en tiempo real sobre el progreso de las pruebas a través de paneles interactivos, métricas, informes de actividad, etc. Los casos de prueba automáticos y manuales se pueden administrar y

documentar fácilmente mediante capturas de pantalla, comparación de resultados esperados versus reales. Existe una amplia gama de plantillas integradas y los usuarios también pueden crear plantillas personalizadas. Los equipos pueden establecer hitos, objetivos personales, listas de verificación y notificaciones por correo electrónico para mejorar la productividad. TestRail se puede integrar fácilmente con herramientas CI como Jenkins, JIRA, TFS, etc., y es compatible con contenedores Docker.

### Testpad

Testpad utiliza planes de prueba inspirados en listas de verificación para pruebas ágiles, pruebas exploratorias, gestión de casos de prueba tradicional, BDD resaltado por sintaxis y mucho más. Es una herramienta liviana con un editor controlado por teclado y tiene una interfaz de usuario altamente receptiva impulsada por JavaScript. Se puede invitar a varias personas a probar la aplicación a través de invitaciones por correo electrónico, y estos probadores invitados no necesitan cuentas ni acceso de inicio de sesión. Puede agregar casos de prueba en cualquier momento durante la prueba. Testpad proporciona una integración perfecta con JIRA. Herramientas de seguimiento de defectos Las herramientas de prueba de software de seguimiento de defectos ayudan en el flujo de trabajo de un extremo a otro y en la gestión de errores. Estas herramientas rastrean el progreso del equipo y registran quién está trabajando en qué módulo.

### Jira

Jira se usa popularmente para Gestión ágil de proyectos además del seguimiento y registro de defectos. Está integrado con el entorno de desarrollo para fines avanzados de seguimiento, registro y generación de informes. JIRA es fácil de usar y tiene muchas funciones. Por ejemplo, los filtros se pueden crear fácilmente con un clic con el propio lenguaje de consulta de JIRA. Los evaluadores pueden crear flujos de trabajo personalizados de extremo a extremo para crear y probar software. Puede utilizar una amplia gama de complementos para adaptarse a diferentes tipos de casos de prueba.

### Mantis

Mantis es una herramienta de seguimiento de errores de código abierto simple pero potente. Puede colaborar con los miembros del equipo sin problemas y personalizar completamente la herramienta en diferentes niveles, como creación de problemas, notificaciones de flujo de trabajo, etc. Obtiene un panel interactivo que muestra elementos pendientes, elementos asignados a un usuario en particular y filtros para ver solo elementos seleccionados.

mantis2large

Puede ver varias versiones del proyecto, lo que permite una mejor gestión de versiones. Mantis tiene su propia función de seguimiento del tiempo; por ejemplo, los usuarios pueden registrar el tiempo dedicado a los problemas. La herramienta crea automáticamente un informe del tiempo total dedicado a diferentes problemas. Mantis es compatible con JetBrains y Eclipse y proporciona integración para varios complementos como TestRail, Slack, HipChat, etc. También tiene una función de correo electrónico, a través de la cual todos los compañeros de equipo pueden conocer los cambios y actualizaciones realizados por otros usuarios.

Herramientas de prueba de seguridad

Estas herramientas prueban la presencia de vulnerabilidades o la posibilidad de ataques de seguridad en la aplicación o el software.

Netparker

Netparker puede escanear automáticamente sitios web, aplicaciones web y servicios web en busca de lagunas y vulnerabilidades de seguridad. Puede identificar problemas y proporcionar pruebas de explotación sin falsos positivos. Es un escáner en línea totalmente configurable que no requiere mantenimiento, no presenta complicaciones, es escalable y admite la colaboración a nivel empresarial. Puede escanear cualquier tipo de aplicación web independientemente de la plataforma o el idioma en el que estén construidas. Puede integrar Netsparker durante el ciclo de vida del desarrollo y rastrear las vulnerabilidades a través de Jira, GitHub, etc.

Herramientas de prueba API

Como su nombre indica, estas herramientas ayudan a probar protocolos como REST, SOAP, etc.

## SoapUI

SoapUI funciona para servicios SOAP, REST, JMS y AMF y se ha creado en Java con Swing para la interfaz de usuario. Es una herramienta multiplataforma de código abierto que se utiliza tanto para pruebas funcionales como de carga en API. También puede realizar llamadas HTTP y JDBC. Características como la cobertura y refactorización WSDL, el inspector de esquemas, el inspector XML, los asistentes XPath, el descubrimiento de API y la interfaz gráfica de usuario fácil de usar, hacen de SOAPUI la herramienta más popular y ampliamente utilizada. Puede obtener protección contra virus y piratas informáticos a través de las funciones de prueba de seguridad. También obtiene un análisis de prueba exhaustivo utilizando la función de informes. apoya Prueba de inyección SQL, que utiliza métodos SQL para identificar áreas de aplicaciones potencialmente débiles y vulnerables. Explora más Herramientas de compilación y prueba de API.

## Herramientas de prueba entre navegadores

Estas herramientas prueban la funcionalidad o aplicación de un sitio web en diferentes navegadores como IE, Firefox, Chrome, Safari, etc.

## LambdaTest

LambdaTest es una herramienta de prueba entre navegadores basada en la nube. Es escalable y se procesa sin problemas en todos los navegadores en dispositivos móviles y de escritorio. Admite la ejecución de casos de prueba en paralelo y realiza pruebas de navegador en vivo de manera eficiente. Por lo tanto, un sitio web se puede probar localmente (antes de la puesta en funcionamiento) utilizando un túnel basado en SSH para conectar la máquina local y los servidores en la nube. proporciona un cambio sencillo de las pruebas receptivas a las pruebas entre navegadores en tiempo real. Puede personalizar el tamaño del monitor para tener la misma ventana gráfica incluso cuando cambia el tamaño de la pantalla. La herramienta tiene una función de desplazamiento inteligente para ignorar los desplazamientos en iframes.

SauceLabs admite más de 700 plataformas de navegador, sistemas operativos y combinaciones de dispositivos, lo que brinda una infraestructura completa para pruebas manuales y automatizadas de aplicaciones móviles y de escritorio utilizando Selenium y otros marcos de prueba. Cubre una amplia gama de plataformas y es ideal para pruebas de JavaScript, pruebas entre navegadores, pruebas manuales y pruebas web funcionales. También es compatible con la integración continua y las pruebas. Herramientas de prueba móviles Estas herramientas automatizan la prueba de aplicaciones en dispositivos Android e iOS.

#### 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.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo.

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://es.m.wikipedia.org/wiki/Pruebas_de_software)

- [https://qdata.com.co/servicios/capacitacion-en-pruebas-de-software/?gclid=CjwKCAjwr56lBhAvEiwA1fuqGn4Z5stDmpb2vjJ19xNqKCmLSiDSh1PUD6rQgBvMEaPPQZe3FZJiZhoCA5oQAvD\\_BwE](https://qdata.com.co/servicios/capacitacion-en-pruebas-de-software/?gclid=CjwKCAjwr56lBhAvEiwA1fuqGn4Z5stDmpb2vjJ19xNqKCmLSiDSh1PUD6rQgBvMEaPPQZe3FZJiZhoCA5oQAvD_BwE)
- <http://consejosdelguru.blogspot.com/2020/06/ventajas-y-desventajas-de-las-pruebas.html?m=1>