**ACTIVIDAD FASE DE PLANEACION Y ANALISIS**

**CURSO DISEÑOS DE SITIOS WEB - COD. 301122**

**FORMATO GUION SITIO WEB DEL OVI**

**Documentación y técnicas de pruebas de software**

**204027**

**Diseñado Por: Juan Pablo Escobar**

A continuación se presenta el formato de Guion para el desarrollo de la actividad de la Fase de Planeación y Análisis, revise muy bien las instrucciones para que realice un correcto diligenciamiento del mismo.

Éxitos!!!

Objetivos del OVI (describa mediante el registro de 1 objetivo general y tres específicos para que se construye este OVI)

**Objetivo general:**

Las pruebas de software es un elemento crítico para la garantía del correcto funcionamiento y usabilidad del [software](https://www.ecured.cu/Software) a utilizar, sea de industria (empresa) o de versionamiento libre (cualquier persona la puede utilizar).

**Objetivo específico 1:**

Detectar defectos en el software.

Verificar la integración adecuada de los componentes.

Verificar que todos los requisitos se han implementado correctamente

**Objetivo específico 2:**

Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente y garantizar que en producción el software desarrollado funcione sin ningún problema

**Objetivo específico 3:**

Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

1. **Contenido informativo del OVI por secciones (**Replique el siguiente cuadro de acuerdo al número de secciones que vaya a crear en el OVI**).**

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: Conceptos básicos de las pruebas de software** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Hablaremos sobre el contexto de la prueba de software. En esta ocasión, nos concentraremos en definir algunas definiciones relacionadas con la prueba de software, así como su justificación y alcance. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| En los recursos a utilizar se integraran textos explicativos los cuales se conformaron de títulos y párrafos en donde se da a entender la temática expuesta para el tema 1.  **El tipo de letra a utilizar será Arial tamaño 12.**  Los videos son un método de estudio muy eficiente ya que incluyen lectura, audio y comprensión visual. Para esto no ayudaremos de la plataforma YouTube para obtener información relevante.  Las imágenes son otra atractivo para adquirí conocimiento ya que de ellas nos vamos para expresar un idea concreta del tema expuesto.  **Las imágenes a utilizar serán en formato.; JPG GIF y PNG**  Para finalizar se utilizara un mapa conceptual para explicar el proceso de las pruebas de software. |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| **Conceptos básicos**  **Proceso de pruebas**  La forma más común de organizar las actividades relacionadas al proceso de pruebas de software son:  **Planeación**, fija las metas y una estrategia general de pruebas  **Preparación**, se describe el procedimiento general de pruebas y se generan los casos de prueba específicos  **Ejecución**, incluye la observación y medición del comportamiento del producto.    **Análisis**, incluye verificación y análisis de resultados para determinar si se observaron fallas  **Seguimiento**, si se detectaron fallas, se inicia un monitoreo para asegurar que se remueva el origen de éstas  **Casos de prueba y Criterios de prueba**  Generar casos de prueba efectivos que revelen la presencia de fallas es fundamental para el éxito del proceso de pruebas (etapa de preparación).  Idealmente, se debería determinar un conjunto de casos de prueba tales que su ejecución exitosa implique que no hay errores en el software desarrollado.  Comúnmente este objetivo ideal no se puede lograr debido a las limitaciones prácticas y teóricas.  Cada caso de prueba cuesta dinero: esfuerzo para generarlo, tiempo de cómputo para ejecutarlo, esfuerzo para evaluar los resultados, Por lo tanto, el número de casos de prueba necesarios para detectar los errores debe ser minimizado para reducir costos  **Objetivo del proceso de pruebas**  Los dos objetivos principales del proceso de pruebas:   * Maximizar el número de errores detectados (cobertura) * Reducir al mínimo el número de casos de prueba (costo)   Como con frecuencia son contradictorios, el problema de seleccionar el conjunto de casos de prueba con el que un programa debe ser probado se vuelve una tarea muy compleja  **Niveles de pruebas**  Generalmente se comienza probando las partes más pequeñas y se continúa con las más grandes.  Para el software convencional:   * El módulo (componente) se prueba primero * Se continua con la integración de módulos   Para el software orientado a objetos Se prueba primero una clase (atributos, métodos, colaboración) |

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: Introducción a los tipos de pruebas y su documentación.** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Especificar y comprender los tipos de pruebas que son utilizadas día a día en la industria del software, a su vez se indicara que tipos de documentos, entregables y artefactos son los desarrollados en el área el testing (QA). |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| En los recursos a utilizar se integraran textos explicativos los cuales se conformaron de títulos y párrafos en donde se da a entender la temática expuesta para el tema 1.  **El tipo de letra a utilizar será Arial tamaño 12.**  Los videos son un método de estudio muy eficiente ya que incluyen lectura, audio y comprensión visual. Para esto no ayudaremos de la plataforma YouTube para obtener información relevante.  Las imágenes son otra atractivo para adquirí conocimiento ya que de ellas nos vamos para expresar un idea concreta del tema expuesto.  **Las imágenes a utilizar serán en formato.; Jpg gif y png**  Para finalizar se utilizara un mapa conceptual para explicar el proceso de las pruebas de software. |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| **Tipos de pruebas de software**  Un conjunto de actividades de **pruebas** suele orientase a comprobar determinados aspectos de un sistema software (o de una parte del mismo). Continuando así con nuestro anterior artículo sobre el [modelo de cebolla para los Niveles de Pruebas Software](https://www.panel.es/blog/pruebas-software-modelo-capas-cebolla/), y siguiendo las directrices del [ISTQB](http://www.istqb.org/), acotaremos los **Tipos de Pruebas Software** en función del objetivo en que se centran. PRUEBAS UNITARIAS  |  |  | | --- | --- | | Objetivo de la Prueba: | Se focaliza en ejecutar cada módulo (o unidad mínima a ser probada, ej = una clase) lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores.  Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido. | | Descripción de la Prueba: | * Particionar los módulos en pruebas en unidades lógicas fáciles de probar. * Por cada unidad hay que definir los casos de prueba (pruebas de caja blanca). * Para esto los casos de prueba deben diseñarse de forma tal que se recorran todos los caminos de ejecución posibles dentro del código bajo prueba; por lo tanto el diseñador debe construirlos con acceso al código fuente de la unidad a probar. * Los aspectos a considerar son los siguientes: Rutinas de excepción, Rutinas de error, Manejo de parámetros, Validaciones, Valores válidos, Valores límites, Rangos, Mensajes posibles. | | Técnica: | * Comparar el resultado esperado con el resultado obtenido. * Si existen errores, reportarlos. | | Criterio de Completitud: | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. | | Consideraciones Especiales: | Para la elaboración de pruebas unitarias en java se puede utilizar el JUNIT y CACTUS. |  PRUEBAS DE INTEGRACIÓN  |  |  | | --- | --- | | Objetivo de la Prueba: | Identificar errores introducidos por la combinación de programas probados unitariamente.  Determina cómo la base de datos de prueba será cargada.  Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.  Verificar que las especificaciones de diseño sean alcanzadas.  Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente. | | Descripción de la Prueba: | ·Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente.  · Determina cómo la base de datos de prueba será cargada.  · Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.  · Decide qué acciones tomar cuando se descubren problemas.  Por cada Caso de Prueba ejecutado: | | Técnica: | * Utilizar la técnica top-down. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos. * Utilizar la técnica down-top. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos. | | Criterio de Completitud: | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. | | Consideraciones Especiales: | Ninguna |  Prueba de Regresión  |  |  | | --- | --- | | Objetivo de la Prueba: | Determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes. | | Descripción de la Prueba: | En esta prueba se vuelve a probar el sistema a la luz de los cambios realizados durante el debugging, mantenimiento o desarrollo de la nueva versión del sistema buscando efectos adversos en otras partes. | | Técnica: | * La prueba de regresión es una nueva corrida de casos de prueba previos. * Se requiere de políticas para ser creada la prueba de regresión y decidir qué casos de prueba incluir, para probar eficientemente. * La prueba de regresión es un buen candidato para automatización. Desde que estas pruebas se repiten una y otra vez, las herramientas para minimizar el esfuerzo del trabajo son útiles. * La prueba de viejas funcionalidades es más importante que la de nuevas funcionalidades. * Aquellos casos de uso (y los casos de prueba asociados) que descubren defectos tempranamente deben ser incluidos en la prueba de regresión. | | Criterio de Completitud: | * Todas las pruebas planeadas han sido ejecutadas. * Todos los defectos que se identificaron han sido tenidos en cuenta. | | Consideraciones Especiales: | Ninguna |  Pruebas de Humo (Smoke Testing o Ad Hoc)  |  |  | | --- | --- | | Objetivo de la Prueba: | Los objetivos son los siguientes:   * Detectar los errores en realeases tempranos y de manera fácil * Probar el sistema constantemente * Garantizar poco esfuerzo en la integración final del sistema * Asegurar los resultados de las pruebas unitarias * Se reducen los riesgos y a baja calidad. | | Descripción de la Prueba: | Toma éste nombre debido a que su objetivo es probar el sistema constantemente buscando que saque “humo” o falle. En algunos proyectos este tipo de prueba va junto con las pruebas funcionales. Permite detectar problemas que por lo regular no son detectados en las pruebas normales. Algunas veces, si las Pruebas ocurren tarde en el ciclo de desarrollo está será una forma de garantizar el buen desarrollo.  Las pruebas de humo NO SON exhaustivas, pero van de extremo a extremo de la aplicación. | | Técnica: | 1. Realizar una integración de todo el sistema cada cierto periodo (se recomienda un día, máximo una semana)  2. Realizar los casos de prueba asignados a los casos de uso finalizados ese día más los realizados en días anteriores  3. Buscar eficientemente errores | | Criterio de Completitud: | · Todas las pruebas planeadas han sido ejecutadas.  · Todos los defectos que se identificaron han sido tenidos en cuenta. | | Consideraciones Especiales: | Cuando se encuentre un error en el release correspondiente al periodo elegido para hacer las integraciones del sistema, se detiene el desarrollo hasta que el error es corregido.  Este tipo de pruebas es útil en la programación extrema (extremme programming) y de sistemas complejos.  Es útil el uso de programas de prueba automáticas que se encarguen de probar os casos de prueba ya ejecutados en realeases anteriores. |  PRUEBAS DEL SISTEMA  |  |  | | --- | --- | | Objetivo de la Prueba: | Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación. | | Descripción de la Prueba: | Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Este tipo de pruebas se basan en técnicas de caja negra, esto es, verificar el sistema (y sus procesos internos), la interacción con las aplicaciones que lo usan vía GUI y analizar las salidas o resultados.  En esta prueba se determina qué pruebas de Sistema (usabilidad, volumen, desempeño, etc.) asegurarán que la aplicación alcanzará sus objetivos de negocio.  La prueba de Sistema incluye:  Prueba funcionalidad  Prueba de Usabilidad  Prueba de Performance  Prueba de Documentación y Procedimientos  Prueba de Seguridad y Controles  Prueba de Volumen  Prueba de Esfuerzo (Stress)  Prueba de recuperación  Prueba de múltiples sitios  Para sistemas web se recomienda especialmente realizar mínimo el siguiente grupo de pruebas de sistema:  · Humo.  · Usabilidad  · Performance  · Funcionalidad  Para capitalizar el trabajo hasta ahora completado, los casos de prueba de las pruebas previas realizadas pueden frecuentemente ser reorganizados y rehusados durante la prueba de sistema. No obstante, deben ser desarrollados casos de prueba adicionales para aquellos aspectos del sistema, tales como documentación, procedimientos y desempeño que no han sido probados durante la prueba unitaria y de integración. |  Pruebas de Carga  |  |  | | --- | --- | | Objetivo de la Prueba: | Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga. | | Descripción de la Prueba: | Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga.  La meta de las pruebas de carga es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada. Adicionalmente, las pruebas de carga evalúan las características de desempeño (tiempos de respuesta, tasas de transacciones y otros aspectos sensibles al tiempo). | | Técnica: | * Use los scripts desarrolladas para Pruebas del Negocio. * Modifique archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre). | | Criterio de Completitud: | * Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado. | | Consideraciones Especiales: | * Las pruebas de carga deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas. * La Base de datos utilizada para pruebas de desempeño debe ser de un tamaño real o proporcionalmente más grande que la diseñada. | |

|  |
| --- |
| **Nombre de la sección que se creara en el OVI: Documentación y certificación para las pruebas de software** |
| **2.1 Objetivo de la sección: (**Registre a continuación el objetivo que tiene esta sección**)** |
| Conocer e implementar los tipos de artefactos que están involucrados en el transcurso del desarrollo de software para el área de calidad. |
| **2.2 Recursos de consulta que usara en la sección: (**coloque el nombre del material que usara para crear los contenidos de la sección y el enlace de descarga de los mismos sean estos Texto, Imágenes, Audios o Vídeos**)** |
| En los recursos a utilizar se integraran textos explicativos los cuales se conformaron de títulos y párrafos en donde se da a entender la temática expuesta para el tema 1.  **El tipo de letra a utilizar será Arial tamaño 12.**  Los videos son un método de estudio muy eficiente ya que incluyen lectura, audio y comprensión visual. Para esto no ayudaremos de la plataforma YouTube para obtener información relevante.  Las imágenes son otra atractivo para adquirí conocimiento ya que de ellas nos vamos para expresar un idea concreta del tema expuesto.  **Las imágenes a utilizar serán en formato.; Jpg gif y png**  Para finalizar se utilizara un mapa conceptual para explicar el proceso de las pruebas de software. |
| **2.3 Redacte un borrador del contenido de lectura en formato de texto que tendrá la sección: (**Sea este la presentación de la sección, el contenido o ambos; redacte un borrador del texto que publicara como contenido en la sección coloque un subtítulo para identificar si corresponde a la presentación de la sección o el contenido de lectura de la sección**)** |
| Documentos En un proyecto de desarrollo de software existe un conjunto de documentos asociados a cada una de las fases del ciclo de vida: planificación, análisis, diseño, construcción,... Podemos considerar el proceso de testing como un proyecto que se ejecuta en paralelo con el desarrollo y en el que se pueden distinguir tres grandes etapas:  Preparación de las pruebas.  Ejecución de las pruebas.  Finalización de las pruebas. En cada una de estas fases hay que generar la documentación apropiada, lo cual puede ser complicado si no se tiene una referencia adecuada. Para proporcionar una base estándar para la documentación del proceso de testing se creó la norma IEEE 829.IEEE 829 propone una serie de documentos que encajan en las etapas de testing de la siguiente forma:  Preparación de las pruebas.  Plan de pruebas.  Especificación de diseño de pruebas.  Especificación de casos de prueba.  Especificación de procedimientos de prueba.  Informe de transferencia de elementos de prueba.  Ejecución de las pruebas.  Registro de pruebas.  Informe de incidentes.  Finalización de las pruebas.  Informe de resumen de pruebas. Aunque el estándar hace referencia a documentos distintos, en la práctica no tienen por qué ser documentos físicos separados. Incluso en muchas ocasiones, gran parte de la información no residirá en documentos, sino en herramientas orientadas a soportar el proceso de testing. Por otro lado, no todos los proyectos requieren el mismo grado de formalidad en la documentación del proceso de testing: seguramente no tendrá el mismo rigor documental un proyecto de un software médico que uno sóbrela construcción de un sencillo web site. Otros factores como la cultura y política de la empresa pueden influir en la formalidad de la documentación. |