

Actividad:

taller sobre construcción del prototipo del *software* de acuerdo al análisis de las características funcionales y de calidad

GA5-220501095-AA1-EV01.

Aprendiz:

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Con base en los requerimientos funcionales del sistema y de calidad, en esta actividad se proponen aspectos clave que se deben tener en cuenta para la construcción de prototipos, también el aprendiz deberá utilizar algunas herramientas para apoyarse en la elaboración del esquema de la funcionalidad. Adicionalmente, se introduce el tema de calidad para lo cual deberá realizar una investigación corta respondiendo algunos lineamientos establecidos en esta actividad.

Elementos para tener en el desarrollo del taller:

Sección 1 - Taller

Teniendo en cuenta las siguientes funcionalidades, elaborar el diagrama de la funcionalidad, establecer colores y componentes a utilizar:

- Se desea elaborar una pantalla que permita validar la autenticidad de un usuario.

RTA: A continuación, se presentan algunos componentes comunes que podrían estar involucrados en un proceso de autenticación de usuario en pantalla:

1. **Pantalla de inicio de sesión:** Una pantalla en la que el usuario puede ingresar sus credenciales, como nombre de usuario y contraseña.
2. **Base de datos de usuarios:** Una base de datos que almacena la información de los usuarios, incluidos los nombres de usuario y las contraseñas.
3. **Verificación de credenciales:** Un proceso que verifica si las credenciales ingresadas por el usuario son correctas y corresponden a un usuario registrado en la base de datos.
4. **Autenticación exitosa:** Si las credenciales son correctas, la pantalla mostrará un mensaje de autenticación exitosa y permitirá al usuario acceder al sistema.
5. **Autenticación fallida:** Si las credenciales no son correctas, la pantalla mostrará un mensaje de autenticación fallida y pedirá al usuario que vuelva a ingresar sus credenciales.
6. **Registro de errores:** El sistema registrará los intentos de inicio de sesión fallidos, lo que puede ayudar a detectar intentos de acceso no autorizado o ataques de fuerza bruta.

Diagrama

Inicio

|
V

Pantalla de inicio de sesión: Mostrar la pantalla de inicio de sesión para que el usuario pueda ingresar sus credenciales.

|
V

Verificación de credenciales: Verificar si las credenciales ingresadas por el usuario son correctas y corresponden a un usuario registrado en la base de datos de usuarios.

|
V

¿Las credenciales son correctas?

|
V

Autenticación exitosa: Mostrar un mensaje de autenticación exitosa y permitir al usuario acceder al sistema.

|
V

¿Las credenciales son incorrectas?

|
V

Autenticación fallida: Mostrar un mensaje de autenticación fallida y pedir al usuario que vuelva a ingresar sus credenciales.

|
V

¿Se han realizado demasiados intentos de inicio de sesión fallidos?

|
V

Proceso de seguridad adicional: Realizar un proceso de seguridad adicional, como enviar un código de verificación al correo electrónico o número de teléfono del usuario.

|
V

Fin

- Se desea elaborar una pantalla que permita ingresar nombres, apellidos, cédula, fecha de nacimiento de un usuario.

RTA: Este diagrama muestra que la pantalla de ingreso de información personal se mostrará primero, y luego se validarán los datos ingresados por el usuario. Si los datos son correctos, la información se guardará en la base de datos de usuarios. Si se requiere información adicional, se mostrará una pantalla adicional para que el usuario pueda ingresar la información adicional necesaria. Finalmente, el proceso de ingreso de información personal se completará.

Diagrama

Inicio

|
V

Pantalla de ingreso de información personal: Mostrar la pantalla de ingreso de información personal para que el usuario pueda ingresar su nombre, apellidos, cédula y fecha de nacimiento.

|
V

Validación de datos: Validar que los datos ingresados por el usuario sean correctos y cumplan con los requisitos necesarios (por ejemplo, la cédula tenga el formato correcto).

|
V

¿Los datos son correctos?

|
V

Guardar información: Si los datos son correctos, guardar la información en la base de datos de usuarios.

|
V

¿Se requiere información adicional?

|
V

Proceso de ingreso de información adicional: Si se requiere información adicional (por ejemplo, dirección, número de teléfono), mostrar una pantalla adicional para que el usuario pueda ingresar la información.

|
V

Fin

- Se desea dibujar una pantalla en donde se informe de posibles errores a un usuario.

RTA: Este diagrama muestra que después de realizar el proceso correspondiente, se verificará si se ha detectado algún error. Si se detecta algún error, se mostrará una pantalla que informe al usuario sobre el error detectado. Si el usuario puede corregir el error, se proporcionarán las herramientas necesarias para que el usuario pueda corregir el error. Una vez que se haya corregido el error, se volverá a realizar el proceso correspondiente con los datos corregidos.

Diagrama

Inicio

|
V

Proceso: Realizar el proceso correspondiente (por ejemplo, envío de un formulario, validación de datos, etc.).

|

V

¿Se detectó algún error?

|
V

Pantalla de error: Mostrar una pantalla que informe al usuario sobre el error detectado.

|
V

¿El usuario puede corregir el error?

|
V

Corrección del error: Proporcionar al usuario las herramientas necesarias para corregir el error (por ejemplo, un campo de texto para que el usuario pueda volver a ingresar los datos).

|
V

¿El usuario corrigió el error?

|
V

Proceso: Volver a realizar el proceso correspondiente con los datos corregidos.

|
V

Fin

- Proponer la paleta de colores de los componentes anteriores utilizando como máximo la combinación de tres colores.

RTA: Este diagrama muestra que en primer lugar se deben seleccionar los componentes que se utilizarán en el diseño, y luego definir la paleta de colores que se utilizará en el diseño. Después de seleccionar la paleta de colores, se visualizará el diseño utilizando esta paleta de colores y se verificará si la combinación de colores es coherente. Si la combinación de colores es coherente, se presentará la propuesta de paleta de colores al equipo de diseño o al cliente para su aprobación o sugerencias de cambios. Si se aprueba la propuesta, el proceso se da por terminado.

Diagrama

Inicio

|
V

Selección de los componentes: Seleccionar los componentes que se utilizarán en el diseño (por ejemplo, botones, campos de texto, menús desplegables, etc.).

|

V

Definición de la paleta de colores: Definir la paleta de colores que se utilizará en el diseño. Se recomienda utilizar una combinación de tres colores para mantener una apariencia coherente en todo el diseño.

|
V

¿La paleta de colores es coherente?

|
V

Visualización del diseño: Visualizar el diseño utilizando la paleta de colores seleccionada y verificar si la combinación de colores es coherente.

|
V

¿La combinación de colores es coherente?

|
V

Propuesta de la paleta de colores: Presentar la propuesta de paleta de colores al equipo de diseño o al cliente para su aprobación o sugerencias de cambios.

|
V

¿Se aprueba la propuesta?

|
V

Fin

Sección 2 - Taller

Elabore una investigación corta usando los materiales disponibles en la biblioteca o internet respecto a los fundamentos de calidad de *software* seleccionando, al menos, tres fuentes que le permitan resolver los siguientes conceptos:

- Definición de calidad de *software*.

RTA: La calidad del software se refiere a la medida en que un programa de computadora cumple con los requisitos y expectativas del usuario, así como con los estándares de la industria en términos de funcionalidad, fiabilidad, facilidad de uso, seguridad, mantenibilidad y eficiencia.

La calidad del software se puede medir en términos de su capacidad para satisfacer las necesidades del usuario, la corrección de los resultados y la consistencia en la ejecución de sus funciones.

Además, se considera que un software de alta calidad es fácil de usar, fácil de aprender y mantener, y es seguro y confiable en su ejecución.

La calidad del software también se puede evaluar mediante pruebas exhaustivas y técnicas de aseguramiento de la calidad, tales como pruebas de unidad, pruebas de integración, pruebas de sistema y pruebas de aceptación. Estas técnicas ayudan a detectar errores y defectos en el software y garantizan que el software cumpla con los estándares y requisitos establecidos.

● ¿Cuál es el concepto de usabilidad en *software*?

RTA: La usabilidad en el software se refiere a la medida en que un programa es fácil de usar y comprender por los usuarios finales. En otras palabras, se trata de la capacidad de un software para permitir a los usuarios realizar tareas de manera eficiente, efectiva y satisfactoria.

La usabilidad se mide en términos de varios factores, como la facilidad de aprendizaje, la eficiencia de uso, la capacidad de recuperación de errores, la satisfacción del usuario y la accesibilidad. Los factores de usabilidad están diseñados para garantizar que el software sea intuitivo y fácil de navegar para los usuarios, independientemente de su nivel de experiencia técnica.

Un software con buena usabilidad se diseña pensando en las necesidades de los usuarios y ofrece una interfaz gráfica intuitiva, organizada y clara. Además, un software con buena usabilidad es capaz de proporcionar retroalimentación útil y ayuda en línea para el usuario.

La usabilidad es un aspecto crítico del diseño de software, ya que tiene un gran impacto en la satisfacción del usuario y la productividad. Los diseñadores de software deben trabajar para crear interfaces de usuario intuitivas y fáciles de usar que permitan a los usuarios alcanzar sus objetivos de manera rápida y eficiente.

● ¿Cuáles son los estándares de calidad de *software*?

RTA: Existen varios estándares de calidad de software que se utilizan en la industria para evaluar y medir la calidad de los programas de computadora. Algunos de los estándares más comunes son:

1. ISO 25000: Estándar de calidad de software que proporciona un conjunto de modelos de calidad y requisitos para evaluar y mejorar la calidad del software.
2. ISO 9001: Estándar de calidad general que se aplica a cualquier tipo de organización y se enfoca en la mejora continua de los procesos y la satisfacción del cliente.
3. CMMI: Modelo de madurez de capacidad de software que proporciona un marco para la mejora del proceso de software y la evaluación de la madurez del proceso de software en una organización.

4. SPICE: Marco de referencia de proceso de software que proporciona un conjunto de prácticas recomendadas para la gestión de procesos de software y la evaluación de la madurez del proceso de software en una organización.
5. IEEE 1061: Estándar que proporciona directrices para la evaluación de la calidad del software y la selección de métricas de calidad adecuadas.

Estos estándares proporcionan un conjunto de directrices y modelos para garantizar que el software cumpla con los requisitos de calidad establecidos y se entregue de manera eficiente y efectiva. Cada uno de estos estándares puede adaptarse a las necesidades específicas de una organización y a los requisitos del proyecto en particular.

•¿Qué tecnologías existen para implementar calidad?

RTA: Existen varias tecnologías y herramientas disponibles para implementar y asegurar la calidad del software en un proyecto de desarrollo. Algunas de las tecnologías más populares son:

1. Pruebas automatizadas: Las pruebas automatizadas utilizan herramientas y scripts para simular acciones de usuario y verificar que el software funcione como se espera. Las pruebas automatizadas son rápidas y eficientes, lo que las convierte en una herramienta valiosa para garantizar la calidad del software.
2. Control de versiones: Las herramientas de control de versiones permiten a los desarrolladores rastrear cambios en el código fuente y colaborar en equipo de manera efectiva. El control de versiones también ayuda a garantizar que las versiones de software sean coherentes y se puedan rastrear los cambios a lo largo del tiempo.
3. Integración continua: La integración continua es una práctica que implica compilar y probar el software de manera regular y automatizada. Esto permite detectar problemas de calidad más temprano en el ciclo de desarrollo y permite a los equipos de desarrollo responder rápidamente a los problemas que surjan.
4. Análisis estático de código: Las herramientas de análisis estático de código examinan el código fuente del software en busca de problemas potenciales, como errores de sintaxis, vulnerabilidades de seguridad y posibles problemas de rendimiento.
5. Pruebas de rendimiento: Las pruebas de rendimiento permiten evaluar la capacidad del software para manejar una carga de trabajo determinada y proporcionan información sobre el rendimiento y la escalabilidad del software.

Estas tecnologías y herramientas pueden ayudar a garantizar la calidad del software y a detectar problemas de calidad antes de que el software se publique. Es importante seleccionar las herramientas y tecnologías adecuadas para el proyecto y las necesidades específicas del equipo de desarrollo.