



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

Aprendiz:

Maria Camila Muñoz Gomez

**Servicio Nacional de Aprendizaje SENA – Centro de Comercio, Industria y Turismo Regional
Córdoba**

Programa: Tecnología En Análisis y Desarrollo De Software (2758357)

Evidencia: Taller sobre construcción del prototipo del software de acuerdo al análisis de las características funcionales y de calidad - GA5-220501095-AA1-EV01

Instructor: Arnaldo Alfonso Montiel Brun

Fecha: 30/09/2024

Colombia

Sesión 1 – Taller

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

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

BOOKTOPIA ≡ MENÚ Q Catálogos Digitales Catálogos En Físico Contactanos :) Términos y Condiciones

Ingresa tu Email

Email

Contraseña

Continuar

O puedes ingresar con:

Ingresa con Google

Ingresa con Facebook

Beneficios de comprar en Booktopia.com

- Múltiples medios de pago**
Paga con tarjeta de crédito o débito.
- Garantía de devolución**
Si tu producto falla, puedes devolverlo y te devolveremos el dinero o reemplazaremos tu producto.
- Envíos a toda Colombia**
Booktopia llega a la puerta de tu casa.

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

BOOKTOPIA ¿Ya tienes una cuenta? INICIAR SESIÓN

Registrarse en Librería Booktopia
Regístrate utilizando tus redes sociales

Correo electrónico
nombre@correo.com

Teléfono Celular
Teléfono Celular

Nombre(s)
Nombre(s)

Contraseña
Contraseña

Apellido(s)
Apellido(s)

Confirmar Contraseña
Confirmar Contraseña

Registrarme

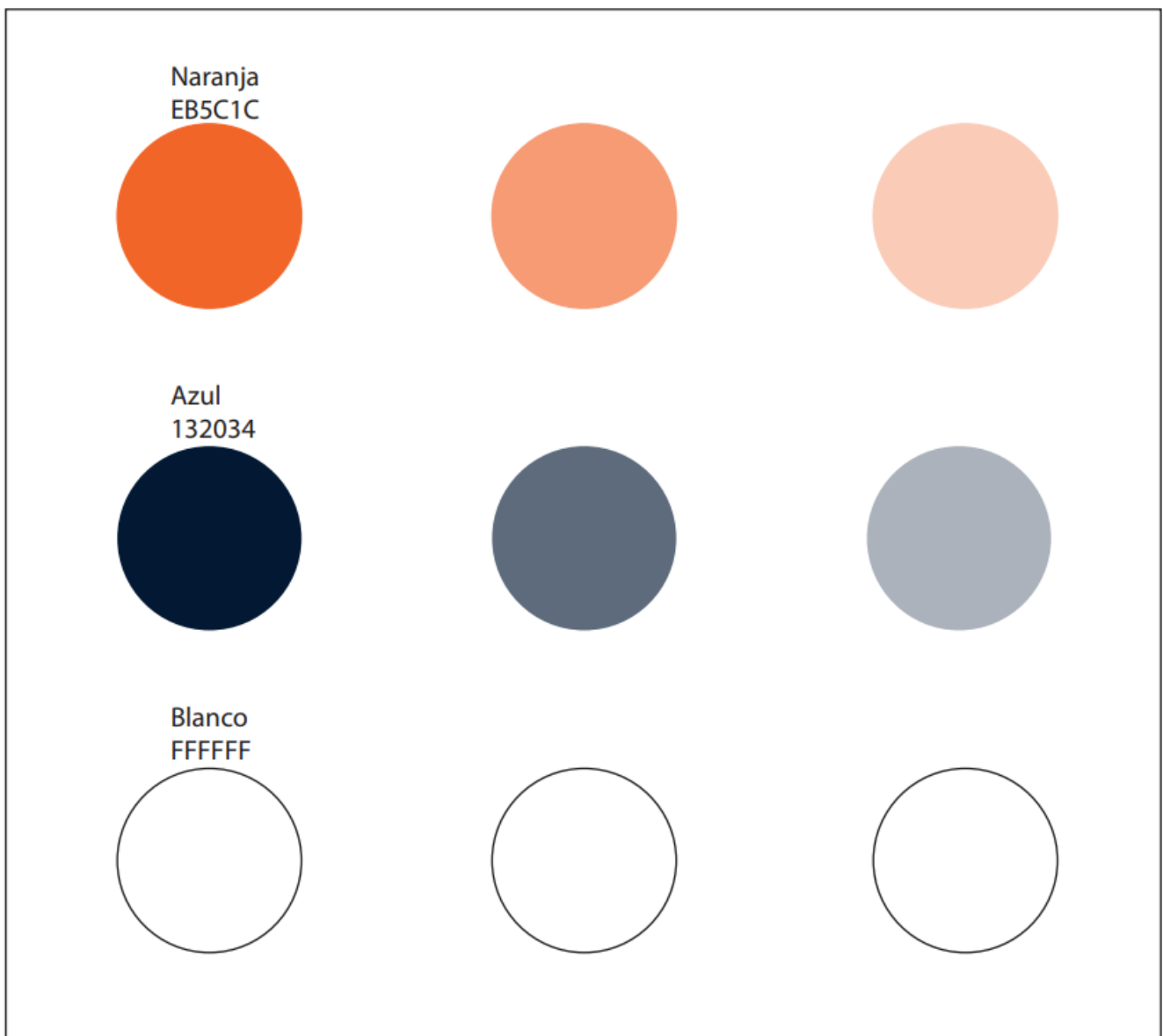
¡Más de 200.000 títulos!
Ahora puedes comprar tus libros por internet desde la comodidad de tu casa u oficina a través de nuestra Tienda Virtual.

Síguenos en redes sociales

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

The screenshot shows the Booktopia website interface. The header includes the Booktopia logo, a menu button, a search bar, and links to digital and physical catalogs. The main content area features a login form with the title 'Ingresa tu Email'. The email field contains 'camilamg071@gmail.com' and the password field contains a masked password. A message below the fields states: 'Tu email o contraseña son incorrectos, intentalo nuevamente'. An orange 'Continuar' button is at the bottom of the form. To the right, a sidebar titled 'Beneficios de comprar en Booktopia.com' lists three benefits: 'Múltiples medios de pago' (Paga con tarjeta de crédito o débito), 'Garantía de devolución' (Si tu producto falla, puedes devolverlo y te devolveremos el dinero o reemplazaremos tu producto), and 'Envíos a toda Colombia' (Booktopia llega a la puerta de tu casa).

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



Sección 2 – Taller

Fundamentos de Calidad de Software

1. Definición de Calidad de Software

La calidad de software se refiere al grado en que un sistema, componente o proceso de software cumple con los requisitos funcionales y no funcionales establecidos y satisface las expectativas del usuario final. Este concepto se puede desglosar en dos grandes categorías: calidad interna y calidad externa. La calidad interna está relacionada con la estructura del software, como el código fuente y los procesos de desarrollo. Esto incluye aspectos como la modularidad, la coherencia y la capacidad de mantenimiento del código. La calidad externa, por otro lado, se refiere a cómo el software se comporta desde la perspectiva del usuario. Esto incluye su funcionalidad, confiabilidad, rendimiento y usabilidad.

Una de las definiciones más formales de calidad de software es proporcionada por la norma ISO/IEC 25010, que establece un modelo de calidad compuesto por ocho características:

1. Adecuación funcional, que mide si el software cumple con las funciones y tareas esperadas.
2. Eficiencia del rendimiento, que evalúa la capacidad del software para utilizar recursos de manera óptima.
3. Compatibilidad, que garantiza que el software funcione bien en distintos entornos o con otras aplicaciones.
4. Usabilidad, que asegura que el software sea fácil de entender y usar.
5. Fiabilidad, que mide la capacidad del software para mantener su nivel de rendimiento bajo diferentes condiciones.
6. Seguridad, que garantiza la protección de la información y los datos del usuario.
7. Mantenibilidad, que permite modificar el software con facilidad para corregir errores o adaptarlo a nuevas necesidades.
8. Portabilidad, que evalúa la capacidad del software para ser transferido a diferentes entornos de hardware o software.

Por lo tanto, la calidad de software no es simplemente un término abstracto; está fundamentado en múltiples características técnicas que permiten evaluar y mejorar el desempeño del software a lo largo de su ciclo de vida.

2. ¿Cuál es el concepto de usabilidad en software?

El concepto de usabilidad en software es clave para garantizar que los usuarios finales puedan interactuar con el sistema de manera eficiente, efectiva y satisfactoria. Se trata de una medida de la facilidad de uso de una aplicación y, en términos más formales, incluye cinco dimensiones importantes:

1. **Facilidad de aprendizaje:** Los usuarios deben poder aprender a usar el software de manera rápida, incluso si es su primera vez utilizando la aplicación. Esto implica una curva de aprendizaje baja y una interfaz intuitiva.
2. **Eficiencia:** Una vez que los usuarios han aprendido a usar el sistema, deben ser capaces de realizar sus tareas de manera rápida y eficiente.
3. **Memorabilidad:** Los usuarios ocasionales deben poder volver al sistema después de un tiempo sin tener que reaprender cómo funciona. Esto implica que el diseño sea consistente y lógico.
4. **Prevención de errores:** Un software bien diseñado minimiza los errores de los usuarios y, si ocurren, permite corregirlos fácilmente.
5. **Satisfacción:** El sistema debe proporcionar una experiencia agradable y libre de frustraciones, lo que lleva a una satisfacción general positiva por parte del usuario.

La usabilidad es una característica crítica porque no importa qué tan funcional o avanzado sea un software si los usuarios no pueden utilizarlo de manera efectiva. Un software de alta usabilidad no solo reduce la necesidad de formación y soporte técnico, sino que también mejora la productividad y la satisfacción del usuario, lo que es fundamental para el éxito de cualquier sistema.

El diseño centrado en el usuario y las pruebas de usabilidad son dos metodologías clave que se utilizan para garantizar la usabilidad en el software. Estas metodologías involucran a los usuarios desde las primeras etapas de desarrollo, asegurando que el producto final sea coherente con las expectativas y necesidades reales de los usuarios.

3. ¿Cuáles son los estándares de calidad de software?

Los estándares de calidad de software son fundamentales para garantizar que los productos desarrollados cumplan con las expectativas de los usuarios, los requisitos legales y las mejores prácticas del sector. Estos estándares son desarrollados por organismos internacionales y son utilizados como referencia por las organizaciones de desarrollo de software en todo el mundo. Algunos de los estándares más relevantes son:

1. ISO/IEC 25010 (SQuaRE): Este estándar define un modelo de calidad que evalúa el software desde dos perspectivas principales: características internas y externas. Abarca aspectos como la seguridad, la eficiencia, la mantenibilidad y la usabilidad, proporcionando un marco integral para medir la calidad de un software.
2. ISO/IEC 12207: Este estándar establece los procesos del ciclo de vida del software, que incluye desde la adquisición y el suministro del software hasta su desarrollo, operación y mantenimiento. Está diseñado para proporcionar una estructura coherente y estandarizada que las organizaciones puedan seguir para gestionar el desarrollo de software de manera eficiente.
3. CMMI (Capability Maturity Model Integration): Este modelo está diseñado para ayudar a las organizaciones a mejorar sus procesos y capacidades de desarrollo de software. El CMMI tiene cinco niveles de madurez que reflejan la capacidad de una organización para producir software de alta calidad de manera consistente. Este estándar se utiliza ampliamente para mejorar la eficiencia de los procesos y asegurar la calidad del software a lo largo de su ciclo de vida.
4. IEEE 730-2014: Este estándar se enfoca en la planificación del aseguramiento de la calidad del software. Proporciona directrices sobre cómo crear planes de aseguramiento de calidad (SQA) efectivos, que incluyen prácticas como la revisión del código, las pruebas de software y la auditoría de procesos.
5. ISO/IEC 90003: Este estándar proporciona pautas para la aplicación de la norma ISO 9001 en el desarrollo, suministro y mantenimiento de software. Es ampliamente utilizado en organizaciones que buscan certificar sus procesos de desarrollo de software bajo la norma ISO 9001.

La adopción de estos estándares ayuda a las organizaciones a estructurar y mejorar sus procesos de desarrollo de software, asegurando que el producto final cumpla con los niveles de calidad esperados.

4. ¿Qué tecnologías existen para implementar calidad?

Existen múltiples tecnologías que facilitan la implementación y el aseguramiento de la calidad en el desarrollo de software. Estas tecnologías permiten automatizar procesos, detectar problemas en fases tempranas y garantizar que el software cumpla con los estándares de calidad establecidos. Algunas de las tecnologías más utilizadas son:

1. **Herramientas de Pruebas Automatizadas:** Las pruebas automatizadas son esenciales para garantizar la calidad del software, especialmente en proyectos grandes y complejos donde las pruebas manuales no son suficientes. Herramientas como Selenium, JUnit, TestNG, y Cypress permiten realizar pruebas funcionales, de integración y de regresión de manera automatizada, asegurando que el software funcione correctamente después de cada cambio.

2. **Integración Continua y Despliegue Continuo (CI/CD):** Las plataformas de CI/CD, como Jenkins, GitLab CI, Travis CI, y CircleCI, permiten automatizar el proceso de integración y despliegue del software. Estas tecnologías facilitan la entrega continua de nuevas funcionalidades y correcciones, al tiempo que garantizan que el software se pruebe automáticamente en cada cambio de código. Esto reduce significativamente el riesgo de introducir errores en el software y mejora la velocidad de entrega.
3. **Análisis Estático de Código:** Herramientas como SonarQube, Checkmarx, y Pylint analizan el código fuente para detectar errores, vulnerabilidades de seguridad, y problemas de calidad, como duplicación de código o adherencia a las mejores prácticas de programación. Este análisis se realiza sin ejecutar el software, lo que permite detectar y corregir problemas antes de que afecten la funcionalidad del sistema.
4. **Monitorización y Registro en Tiempo Real:** Tecnologías como Prometheus, Grafana, Elastic Stack (ELK), y Splunk permiten monitorear el rendimiento del software en tiempo real, detectando posibles cuellos de botella, errores o problemas de seguridad que puedan surgir durante la operación del software. Estas herramientas también permiten el análisis de registros (logs), lo que facilita la identificación y solución de problemas en producción.
5. **Metodologías Ágiles:** Aunque no es una tecnología per se, las metodologías ágiles como Scrum y Kanban permiten una gestión más efectiva de los proyectos de software, con un enfoque en la entrega continua de valor y la mejora de la calidad mediante ciclos de retroalimentación rápidos y constantes.

El uso combinado de estas tecnologías permite que los equipos de desarrollo mejoren la calidad del software desde las primeras etapas de desarrollo hasta la implementación y operación en producción. Estas tecnologías no solo aseguran que el software funcione correctamente, sino que también mejoran la eficiencia del equipo, reducen costos a largo plazo y aumentan la satisfacción del cliente.

Conclusión

La calidad de software es un aspecto multifacético que abarca desde la definición de requisitos y su cumplimiento, hasta la usabilidad, el cumplimiento de estándares internacionales y el uso de tecnologías avanzadas para asegurar que el software sea robusto, eficiente y seguro. Al comprender y aplicar estos fundamentos, las organizaciones pueden no solo satisfacer las necesidades de los usuarios, sino también mantenerse competitivas en un mercado tecnológico en constante evolución.