DOCUMENTO CON LAS VERIFICACIONES DE CONDICIONES DE CALIDAD DEL PRODUCTO DE SOFTWARE AJUSTADO

GA11-220501098-AA3-EV02

JAVIER ENRIQUE CAMPOS TORRES

FICHA 2627060

INSTRUCTOR

ANDRÉS RUBIANO CUCARÍAN

ANÁLISIS Y DESARROLLO DE SOFTWARE

CENTRO DE SERVICIOS FINANCIEROS

2024

VERIFICACIÓN DE CONDICIONES DE CALIDAD DEL PRODUCTO DE SOFTWARE: SISTEMA DE INFORMACIÓN DE LA PANADERÍA "PAN DE DIOS"

TABLA DE CONTENIDO

1. INTRODUCCIÓN

En el desarrollo de software, la calidad del producto es un aspecto fundamental que determina su éxito y aceptación por parte de los usuarios finales. La verificación de condiciones de calidad del producto de software es un proceso sistemático y riguroso que asegura que el software cumple con los requisitos especificados y las expectativas de los usuarios. Este proceso se enfoca en evaluar diversos atributos de calidad definidos por estándares internacionales como la norma ISO/IEC 25010, que incluyen características como la funcionalidad, eficiencia, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad del software. La verificación de calidad abarca una serie de actividades y prácticas, desde la revisión y prueba del código hasta la validación del desempeño del sistema en entornos reales. Estas actividades son esenciales para identificar y corregir defectos, mejorar el rendimiento y garantizar que el software sea robusto, seguro y fácil de mantener. Entre las técnicas más comunes se encuentran las pruebas unitarias, de integración, de sistema y de aceptación, cada una diseñada para evaluar diferentes aspectos del software en diversas etapas de su ciclo de vida. La importancia de la verificación de calidad radica en su capacidad para prevenir problemas costosos y mejorar la satisfacción del cliente. Al detectar y resolver errores tempranamente, se minimizan los riesgos asociados con el despliegue de software defectuoso, se optimizan los recursos y se asegura un producto final de alta calidad. Además, este proceso fomenta la adopción de prácticas de desarrollo de software más disciplinadas y meticulosas, contribuyendo a la creación de soluciones tecnológicas más confiables y efectivas.

2. DESCRIPCIÓN DEL PRODUCTO

- **Nombre del Producto**:

  Sistema de Información de la Panadería "Pan de Dios"

- **Descripción General**:

  El sistema permite la gestión de productos, clientes y pedidos para una panadería.

- **Módulos Incluidos**:

  - Módulo de Productos: Gestión de inventario, precios y categorías.
  - Módulo de Clientes: Registro y mantenimiento de datos de clientes.
  - Módulo de Pedidos: Creación, seguimiento y gestión de pedidos.

3. OBJETIVOS DE LA VERIFICACIÓN DE CALIDAD

- **Objetivos Generales**:

  Asegurar que el sistema cumple con los requisitos especificados y ofrece un rendimiento, seguridad y usabilidad adecuados.

- **Objetivos Específicos**:
  - Verificar la funcionalidad de cada módulo.
  - Evaluar la eficiencia y rendimiento del sistema.
  - Comprobar la seguridad y protección de datos.
  - Asegurar la usabilidad y experiencia de usuario.

4. METODOLOGÍA DE VERIFICACIÓN

- **Enfoque de Verificación**:

  Utilización de pruebas automatizadas y manuales.

- **Herramientas Utilizadas**:
  - Selenium para pruebas automatizadas.
  - JIRA para seguimiento de defectos.
  - JMeter para pruebas de rendimiento.
- **Proceso de Verificación**:
  - Preparación del entorno de pruebas.
  - Ejecución de casos de prueba.
  - Documentación y análisis de resultados.

5. CRITERIOS DE CALIDAD

- **Funcionalidad**:

  Cada módulo debe cumplir con los requerimientos funcionales especificados.

- **Rendimiento**:

  El sistema debe responder dentro de los tiempos establecidos bajo condiciones de carga esperadas.

- **Seguridad**:

  Los datos de clientes y pedidos deben estar protegidos contra accesos no autorizados.

- **Usabilidad**:

  La interfaz debe ser intuitiva y fácil de usar para los usuarios finales.

6. CASOS DE PRUEBA

6.1. MÓDULO DE PRODUCTOS

- **Caso de Prueba 1**: Adición de un nuevo producto

    - **Descripción**:

      Verificar que se puede añadir un nuevo producto con todos los detalles necesarios.

    - **Pasos**:

        1. Navegar al módulo de productos.
        2. Seleccionar "Añadir nuevo producto".
        3. Completar los campos requeridos (nombre, precio, categoría).
        4. Guardar y verificar que el producto aparece en la lista de productos.

    - **Resultado Esperado**:

      El producto se añade correctamente y aparece en la lista.

- **Caso de Prueba 2**: Modificación de un producto existente

    - **Descripción**:

      Verificar que se puede modificar los detalles de un producto existente.

- **Pasos**:

    1. Seleccionar un producto de la lista.
    2. Cambiar los detalles (nombre, precio, categoría).
    3. Guardar los cambios y verificar que las modificaciones se reflejan.

- **Resultado Esperado**:

  Los cambios se guardan y se reflejan correctamente.

6.2. MÓDULO DE CLIENTES

- **Caso de Prueba 1**: Registro de un nuevo cliente

    - **Descripción**:

      Verificar que se puede registrar un nuevo cliente con todos los detalles necesarios.

    - **Pasos**:

        1. Navegar al módulo de clientes.
        2. Seleccionar "Añadir nuevo cliente".
        3. Completar los campos requeridos (nombre, dirección, teléfono, correo electrónico).
        4. Guardar y verificar que el cliente aparece en la lista de clientes.

- **Resultado Esperado**:

  El cliente se añade correctamente y aparece en la lista.

- **Caso de Prueba 2**: Actualización de los detalles de un cliente
  - **Descripción**:

    Verificar que se pueden actualizar los detalles de un cliente existente.

  - **Pasos**:

    1. Seleccionar un cliente de la lista.
    2. Cambiar los detalles (nombre, dirección, teléfono, correo electrónico).
    3. Guardar los cambios y verificar que las modificaciones se reflejan.

  - **Resultado Esperado**:

    Los cambios se guardan y se reflejan correctamente.

6.3. MÓDULO DE PEDIDOS

- **Caso de Prueba 1**: Creación de un nuevo pedido
  - **Descripción**:

    Verificar que se puede crear un nuevo pedido con productos seleccionados y un cliente asignado.

- **Pasos**:

    1. Navegar al módulo de pedidos.
    2. Seleccionar "Crear nuevo pedido".
    3. Seleccionar productos y asignar un cliente.
    4. Guardar y verificar que el pedido aparece en la lista de pedidos.

- **Resultado Esperado**:

  El pedido se crea correctamente y aparece en la lista.

- **Caso de Prueba 2**: Seguimiento y actualización del estado de un pedido

    - **Descripción**:

      Verificar que se puede seguir y actualizar el estado de un pedido existente.

    - **Pasos**:

        1. Seleccionar un pedido de la lista.
        2. Actualizar el estado del pedido (ej., de "pendiente" a "completado").
        3. Guardar los cambios y verificar que el estado se refleja correctamente.

    - **Resultado Esperado**:

      El estado del pedido se actualiza y se refleja correctamente.

7. RESULTADOS DE LAS PRUEBAS

- **Resumen de Resultados**:

  Resumen general de los resultados obtenidos en las pruebas, incluyendo número de casos de prueba ejecutados, número de casos exitosos y número de fallos.

- **Análisis de Resultados**:

  Análisis detallado de los resultados, incluyendo la identificación de defectos y áreas que requieren mejora.

- **Gráficos y Tablas**:

  Visualización de los resultados a través de gráficos y tablas.

8. CONCLUSIONES Y RECOMENDACIONES

- **CONCLUSIONES**:

1. Garantía de Cumplimiento de Requisitos:

La verificación de calidad garantiza que el software cumple con los requisitos funcionales y no funcionales establecidos. Esto incluye no solo la funcionalidad básica del sistema, sino también atributos críticos como el desempeño, la usabilidad, la seguridad y la mantenibilidad.

2. Mejora de la Satisfacción del Cliente:

Al asegurar que el software funcione correctamente y satisfaga las necesidades del usuario, se incrementa la satisfacción del cliente. Un producto que opera de manera confiable y eficiente genera confianza y lealtad, elementos clave para el éxito comercial.

3. Reducción de Costos y Tiempos de Desarrollo:

Detectar y corregir errores en etapas tempranas del desarrollo reduce significativamente los costos y tiempos asociados con la corrección de defectos en fases posteriores. La verificación de calidad contribuye a un ciclo de desarrollo más eficiente y económico.

4. Prevención de Problemas Críticos:

Las pruebas exhaustivas y la verificación continua permiten identificar y resolver problemas potenciales antes de que se conviertan en fallos críticos. Esto no solo mejora la estabilidad del software, sino que también previene incidentes que podrían afectar gravemente la operación y reputación de la organización.

5. Promoción de Mejores Prácticas de Desarrollo:

La implementación de un riguroso proceso de verificación de calidad fomenta la adopción de mejores prácticas en el desarrollo de software. Esto incluye el uso de metodologías ágiles, la integración continua y el desarrollo orientado a pruebas, que en conjunto contribuyen a la creación de software de alta calidad.

6. Seguridad y Confianza:

La verificación de calidad incluye evaluaciones de seguridad que aseguran que el software está protegido contra amenazas y vulnerabilidades. Esto es esencial para proteger la información y los datos de los usuarios, así como para cumplir con las normativas y estándares de la industria.

7. Facilidad de Mantenimiento:

Un software bien verificado es más fácil de mantener y actualizar. La claridad en el código y la documentación, junto con la identificación y corrección temprana de defectos, simplifican las tareas de mantenimiento y aseguran la longevidad del producto.

- **RECOMENDACIONES**:

La verificación de condiciones de calidad del producto de software es un pilar esencial en el desarrollo de aplicaciones robustas, seguras y eficientes. A través de un proceso sistemático y riguroso, se asegura que el software cumpla con los requisitos especificados y las expectativas de los usuarios, minimizando riesgos y optimizando recursos. Recomendaciones para mejorar el sistema basado en los resultados y análisis de las pruebas.

10. GLOSARIO

A

API (Interfaz de Programación de Aplicaciones):

Conjunto de protocolos y herramientas que permiten a diferentes aplicaciones interactuar entre sí.

Automatización de Pruebas:

Proceso de usar software para ejecutar pruebas predefinidas sobre el código del software, en lugar de realizar pruebas manuales.

B

BPMN (Business Process Model and Notation):

Notación gráfica para especificar procesos de negocio en un diagrama de flujo.

C

CI/CD (Integración Continua/Despliegue Continuo):

Prácticas de desarrollo que permiten a los equipos de desarrollo integrar y desplegar cambios de código de manera frecuente y automatizada.

Compatibilidad:

Capacidad del software para funcionar en diferentes entornos y con otros sistemas.

Componente:

Parte modular de un sistema de software que encapsula una funcionalidad específica.

D

Desempeño Eficiente:

Capacidad del software para realizar sus funciones con el mínimo uso de recursos computacionales.

Diagrama de Actividad:

Diagrama UML que muestra el flujo de control de una actividad a otra.

Diagrama de Clases:

Diagrama UML que describe la estructura de un sistema mostrando sus clases, atributos, operaciones y las relaciones entre los objetos.

Diagrama de Secuencia:

Diagrama UML que muestra cómo los objetos interactúan en un orden temporal.

E

ERD (Diagrama Entidad-Relación):

Modelo de datos que describe la estructura lógica de las bases de datos, representando entidades y sus relaciones.

Especificaciones SMART:

Criterios para definir objetivos específicos, medibles, alcanzables, relevantes y temporales.

F

Fiabilidad:

Capacidad del software para funcionar sin fallos bajo condiciones especificadas.

G

GoF (Gang of Four):

Cuatro autores que escribieron el libro "Design Patterns: Elements of Reusable Object-Oriented Software", que detalla patrones de diseño de software ampliamente utilizados.

H

Historias de Usuario:

Descripción corta y simple de una funcionalidad del software escrita desde la perspectiva del usuario final.

I

ISO/IEC 25010:

Norma internacional que define un modelo de calidad para evaluar las características del software y su uso.

M

Mockup:

Prototipo visual de una interfaz de usuario, que muestra la estructura y diseño de la interfaz.

Modelo Relacional:

Modelo de datos basado en la teoría de conjuntos y las relaciones matemáticas entre datos.

N

Normalización (Base de Datos):

Proceso de organización de datos en una base de datos para reducir la redundancia y mejorar la integridad de los datos, comúnmente alcanzado hasta la tercera forma normal (3FN).

P

Patrón de Diseño:

Solución reutilizable a problemas comunes en el diseño de software.

Portabilidad:

Capacidad del software para ser transferido de un entorno a otro.

R

Requerimientos Funcionales:

Descripción de las funciones que el sistema debe ser capaz de realizar.

Requerimientos No Funcionales:

Descripción de las cualidades que el sistema debe tener, como desempeño, usabilidad y fiabilidad.

S

Seguridad:

Capacidad del software para proteger la información y datos contra acceso no autorizado y daños.

Sketch:

Boceto o esquema simple de una interfaz de usuario, utilizado en las etapas iniciales del diseño.

SMART:

Acrónimo para Specific (Específico), Measurable (Medible), Achievable (Alcanzable), Relevant (Relevante), Time-bound (Temporal).

Stress Testing:

Prueba que evalúa cómo se comporta el sistema bajo condiciones extremas de carga.

T

Técnicas de Recolección de Información:

Métodos utilizados para obtener datos y conocimientos necesarios para el análisis y diseño de sistemas, como entrevistas, encuestas y observación.

U

Usabilidad:

Facilidad con la que los usuarios pueden aprender a utilizar y operar el sistema.

UML (Lenguaje Unificado de Modelado):

Lenguaje de modelado estándar utilizado para especificar, visualizar y documentar los artefactos de un sistema de software.

W

Wireframe:

Esquema visual de una página web o aplicación que muestra la estructura básica y los elementos clave sin detalles de diseño visual.

Este documento establece una guía estructurada para la verificación de condiciones de calidad del sistema de información de la panadería "Pan de Dios", asegurando que cumple con los requisitos especificados y ofrece un rendimiento, seguridad y usabilidad adecuados.