Universidad Autónoma De Chiapas

Licenciatura en ingeniería en desarrollo y tecnologías de software

Definición de los Requerimientos de la Arquitectura

Taller De Desarrollo 4

Dr. Luis Gutiérrez Alfaro

Alumno: Manuel Antonio Nuñez Sánchez

Matricula: A200884 "6°M"

INTRODUCCIÓN:

En esta investigación veremos los Requerimientos de la Arquitectura de Software y nos dicen que son un conjunto de especificaciones y directrices que describen cómo debe estar estructurado y organizado un sistema de software. Estos requerimientos capturan las decisiones fundamentales sobre la estructura, componentes, interacciones y comportamientos del sistema, y actúan como la base para el diseño y desarrollo del software. Son esenciales para establecer una visión clara de la arquitectura del software desde el principio y para garantizar que el sistema cumpla con los objetivos y necesidades del proyecto.

Estos requerimientos suelen derivarse de los Requerimientos Funcionales, que describen las funciones y tareas que el software debe realizar, así como de los Requerimientos No Funcionales, que abordan aspectos de rendimiento, seguridad, usabilidad y otros atributos de calidad del sistema.

REQUERIMIENTOS FUNCIONALES:

Estos requerimientos son esenciales para guiar el diseño y desarrollo de la arquitectura del software y asegurarse de que el sistema cumpla con las expectativas y necesidades de los usuarios.

Los Requerimientos Funcionales de la arquitectura de software son un conjunto de especificaciones que describen las funciones, capacidades y comportamientos específicos que el sistema de software debe proporcionar para cumplir con las necesidades y objetivos del proyecto. Estos requerimientos se centran en lo que el sistema debe hacer en términos de interacciones y comportamientos funcionales. Aquí hay algunos ejemplos de Requerimientos Funcionales de la arquitectura de software:

Registro de usuarios: El sistema debe permitir a los usuarios registrarse proporcionando su nombre, dirección de correo electrónico y contraseña. Debe verificar la unicidad de las direcciones de correo electrónico y enviar un correo electrónico de confirmación.

Iniciar sesión: Los usuarios registrados deben poder iniciar sesión en el sistema utilizando su dirección de correo electrónico y contraseña.

Gestión de productos: El sistema debe permitir a los usuarios ver, buscar, agregar, editar y eliminar productos de su carrito de compras.

Procesamiento de pagos: El sistema debe integrarse con una pasarela de pago para permitir a los usuarios realizar transacciones seguras al comprar productos.

Generación de informes: El sistema debe generar informes mensuales de ventas, que incluyan la cantidad total de ventas, los productos más vendidos y los ingresos generados.

Comunicación de errores: Cuando ocurra un error, el sistema debe mostrar mensajes de error claros y descriptivos para ayudar a los usuarios a comprender y solucionar el problema.

Notificaciones por correo electrónico: El sistema debe enviar notificaciones por correo electrónico a los usuarios cuando se realicen cambios en sus pedidos, como confirmaciones de compra y actualizaciones de envío.

Control de acceso: El sistema debe tener diferentes niveles de acceso para los usuarios, como usuarios normales y administradores, cada uno con diferentes privilegios y capacidades.

Búsqueda avanzada: Los usuarios deben poder realizar búsquedas avanzadas utilizando filtros, palabras clave y otros criterios para encontrar productos específicos.

Integración de redes sociales: El sistema debe permitir a los usuarios compartir productos en sus redes sociales y conectarse con sus perfiles en plataformas sociales. Estos ejemplos ilustran cómo los Requerimientos Funcionales de la arquitectura de software se centran en las interacciones y comportamientos específicos que el sistema debe proporcionar a los usuarios y cómo deben funcionar esas interacciones.

REQUERIMIENTOS NO FUNCIONALES: Los Requerimientos No Funcionales de la arquitectura de software son un conjunto de especificaciones que describen los atributos de calidad, restricciones y características no funcionales que el sistema de software debe cumplir. A diferencia de los Requerimientos Funcionales, que se centran en las funciones y comportamientos del sistema, los RNFAS se centran en aspectos como el rendimiento, la seguridad, la usabilidad y otros atributos que afectan la calidad global del sistema. Aquí hay algunos ejemplos de Requerimientos No Funcionales de la arquitectura de software:

*Rendimiento: El sistema debe responder a las solicitudes de los usuarios en menos de 2 segundos en promedio y manejar al menos 1000 solicitudes concurrentes.

*Escalabilidad: El sistema debe ser capaz de manejar un aumento del 50% en la carga de trabajo sin degradación significativa en el rendimiento.

*Disponibilidad: El sistema debe estar disponible para los usuarios al menos el 99.9% del tiempo, excluyendo ventanas de mantenimiento programado.

*Seguridad: Los datos confidenciales de los usuarios deben estar encriptados tanto en tránsito como en reposo, y el acceso a funciones críticas debe requerir autenticación de dos factores.
*Usabilidad: La interfaz de usuario debe seguir los principios de diseño centrado en el usuario y cumplir con las pautas de accesibilidad WCAG 2.0.

*Mantenibilidad: El código fuente debe estar bien documentado y seguir estándares de codificación para facilitar futuras actualizaciones y modificaciones.

*Compatibilidad: El sistema debe ser compatible con los navegadores web más comunes, como Chrome, Firefox y Safari, en sus versiones más recientes.

*Interoperabilidad: El sistema debe poder comunicarse e intercambiar datos con otros sistemas mediante APIs estándar.

*Eficiencia: Las operaciones del sistema deben utilizar los recursos de manera eficiente para minimizar los costos operativos.

*Cumplimiento normativo: El sistema debe cumplir con las regulaciones y estándares de seguridad de datos pertinentes, como GDPR o HIPAA.

Al igual que con los Requerimientos Funcionales, los Requerimientos No Funcionales también son esenciales para el diseño y desarrollo del software, ya que influyen en la calidad general y la experiencia del usuario. Estos requerimientos ayudan a establecer estándares claros y objetivos medibles para aspectos clave del sistema que no están directamente relacionados con las funciones visibles para los usuarios finales.

CASOS DE USO: Los casos de uso de interacción con la arquitectura describen cómo los actores interactúan con los diferentes elementos y componentes de la arquitectura de software. Estos casos de uso son útiles para comprender cómo se utilizan y se benefician los diversos aspectos de la arquitectura en el contexto de la interacción del usuario y el sistema.

los ejemplos de casos de uso de interacción con la arquitectura muestran cómo diferentes usuarios o sistemas externos interactúan con los componentes de la arquitectura de software para realizar diversas acciones y lograr objetivos específicos. Estos casos de uso ayudan a comprender cómo se utilizan y se aprovechan los elementos arquitectónicos para proporcionar funcionalidades y servicios a los usuarios.

Realizar una búsqueda en la interfaz de usuario: Este caso de uso podría describir cómo un usuario interactúa con la interfaz de búsqueda del sistema para encontrar productos, información o cualquier otro contenido relevante. Puede involucrar solicitudes a la capa de presentación que, a su vez, interactúa con la capa de servicios para obtener y presentar los resultados.

Procesar una transacción de pago: Describe cómo un usuario realiza una transacción de pago en la plataforma, lo que implica la interacción con la pasarela de pago y la comunicación entre diferentes componentes de la arquitectura para garantizar la seguridad y la integridad de la transacción.

Acceder a datos a través de una API: Ilustra cómo sistemas externos pueden interactuar con la arquitectura a través de ApIs, cómo una aplicación móvil obtiene información actualizada a partir de un servicio REST o cómo una aplicación web utiliza una API para autenticar usuarios.

Gestionar el perfil de usuario: Describe cómo un usuario accede, edita y actualiza su perfil en la aplicación, lo que podría implicar interacciones con la base de datos y componentes de autenticación.

Recibir notificaciones por correo electrónico: Muestra cómo el sistema interactúa con un servicio de envío de correos electrónicos para notificar a los usuarios sobre cambios en su cuenta, pedidos o actualizaciones importantes.

Explorar productos recomendados: Describe cómo se generan y presentan recomendaciones de productos a los usuarios, lo que podría implicar interacciones con motores de recomendación y análisis de datos.

Solicitar asistencia en tiempo real: Ilustra cómo los usuarios pueden solicitar asistencia en tiempo real, como un chat en vivo, y cómo esta interacción se maneja dentro de la arquitectura, potencialmente involucrando la comunicación en tiempo real entre el usuario y los agentes de soporte.

Subir y descargar archivos: Describe cómo los usuarios pueden cargar archivos al sistema y cómo otros usuarios pueden descargarlos posteriormente, lo que podría implicar la interacción con un sistema de almacenamiento y gestión de archivos.

Bibliografías

Moodle USP: e-Disciplinas

Arquitectura del software (umich.mx)

https://repository.ucatolica.edu.co/server/api/core/bitstreams/c6da0ef9-0141-496e-978e-75fde62443b4/content