

CICLO DE VIDA DE DESARROLLO DE SOFTWARE (DISEÑO).

Etapa del Diseño en el Desarrollo de Software

- Es la fase en la que se transforma un conjunto de requisitos en una arquitectura y diseño detallado del sistema.
- **Objetivo Principal:** Garantizar que el software sea funcional, mantenible y escalable.
- **Subfases:**
 - Diseño arquitectónico (estructura general del sistema).
 - Diseño detallado (definición específica de módulos y componentes).

Diseño Detallado de Software

- **Definición:** Desarrolla las especificaciones técnicas de cada componente.
- **Incluye:**
 - Identificación de módulos.
 - Definición de interfaces entre componentes.
 - Estructura de datos requerida.
- **Beneficio:** Reduce errores y asegura que el sistema sea fácil de implementar y mantener.

Identificación de los Componentes y Módulos

- **Componentes:** Partes funcionales reutilizables del software (ej. autenticación de usuarios).
- **Módulos:** Agrupaciones lógicas de funcionalidades relacionadas.
- **Técnica:** Divide el sistema en módulos pequeños y manejables para facilitar la implementación y pruebas.
- **Ejemplo:**
 - Sistema de Reservas:
 - Módulos: Gestión de clientes, Gestión de reservas, Gestión de pagos.

Definición de Interfaces entre Componentes

- **Propósito:** Establecer cómo interactúan los módulos entre sí.
- **Incluye:**
 - Formatos de entrada y salida.
 - Protocolos de comunicación (API, servicios web).
- **Ejemplo:**
 - Un módulo de “Reservas” debe enviar datos al módulo de “Pagos” mediante una API REST.

Definición de Estructura de Datos

- **Qué es:** Especificar cómo los datos serán organizados, almacenados y manipulados.
- **Incluye:**
 - Tipos de datos utilizados (cadenas, enteros, listas, etc.).
 - Modelos de datos (bases de datos relacionales o no relacionales).
- **Ejemplo:**
 - Una base de datos de clientes podría incluir campos como ID, Nombre, Email, Teléfono.

Consideraciones Clave: Escalabilidad, Rendimiento y Seguridad

- **Escalabilidad:**

- Diseñar módulos que soporten aumento de usuarios o datos.
- Ejemplo: Uso de bases de datos distribuidas.

- **Rendimiento:**

- Optimizar algoritmos y minimizar tiempos de respuesta.
- Ejemplo: Implementar almacenamiento en caché.

- **Seguridad:**

- Proteger datos sensibles y prevenir accesos no autorizados.
- Ejemplo: Cifrado de datos y autenticación robusta.

Revisión del Diseño

- **Propósito:** Garantizar que el diseño cumple con los requisitos funcionales y no funcionales.
- **Métodos:**
 - Revisiones de pares (peer reviews).
 - Creación de prototipos para validar funcionalidades clave.
 - Verificación de consistencia con los requisitos iniciales.
- **Ejemplo:**
 - Validar que el módulo de pagos funciona con los métodos especificados en los requisitos.

Ejemplo Práctico: Sistema de E-commerce

- **Módulos Identificados:**
 - Gestión de productos.
 - Carrito de compras.
 - Procesamiento de pagos.
- **Interacción entre módulos:**
 - “Carrito de Compras” consulta “Gestión de Productos” para actualizar precios.
 - “Procesamiento de Pagos” valida los datos enviados por el “Carrito”.

Conclusión

- Un diseño detallado y bien estructurado es la base para un desarrollo exitoso.
- **Claves del éxito:**
 - Identificar claramente componentes y módulos.
 - Diseñar interfaces bien definidas.
 - Considerar escalabilidad, rendimiento y seguridad desde el inicio.
 - Revisar constantemente para cumplir los requisitos.