

Universidad privada Domingo Savio



Diseño de Sistemas

Estudiante: Laura Andrea Rivero Parada

Carrera: Ingeniería en sistema

Docente: Hugo Arnaldo Guzman Centellas

Santa Cruz – Bolivia

Diseño de sistemas

1. Concepto del diseño

El diseño es una fase clave en el ciclo de vida de desarrollo de software. Se centra en traducir los requisitos y especificaciones del sistema en una solución técnica viable. Esta etapa abarca la creación de representaciones de alto y bajo nivel que guiarán la implementación final. El objetivo es planificar cómo se logrará la funcionalidad deseada de manera efectiva, mientras se toman en cuenta aspectos como la escalabilidad, la mantenibilidad y la eficiencia.

- **Diseño lógico vs físico:** El diseño lógico se enfoca en cómo se organizará el sistema desde un punto de vista conceptual, mientras que el diseño físico detalla cómo será implementado técnicamente, incluyendo tecnologías, lenguajes de programación, y detalles sobre el hardware.

2. Diseño de la arquitectura

El **diseño de la arquitectura** de un sistema establece la estructura general del mismo. Define cómo los diferentes componentes interactúan y se comunican entre sí, las capas del sistema, y los estándares y patrones que se utilizarán. Se encarga de dividir el sistema en módulos o subsistemas y determinar sus interacciones para crear una solución robusta y flexible. La arquitectura debe soportar el crecimiento futuro y permitir modificaciones sin comprometer el sistema completo.

- **Patrones arquitectónicos:** Algunos de los patrones arquitectónicos comunes incluyen:
 - **Arquitectura en capas:** Separa la lógica de presentación, la lógica de negocio y el acceso a datos en capas independientes.
 - **Microservicios:** Descompone el sistema en servicios pequeños e independientes que se comunican entre sí, ideales para sistemas escalables y mantenibles.
 - **Cliente-servidor:** Divide el sistema en dos partes: un cliente que realiza peticiones y un servidor que responde a dichas peticiones.

El diseño arquitectónico también incluye la toma de decisiones sobre la seguridad, el manejo de errores y la tolerancia a fallos.

3. Diseño de base de datos

El **diseño de base de datos** se encarga de organizar los datos del sistema de forma que se pueda acceder a ellos de manera eficiente y precisa. Un diseño eficiente reduce la redundancia y mejora el rendimiento de las consultas. Se lleva a cabo en varias fases:

- **Modelo conceptual:** Se identifican las entidades principales y las relaciones entre ellas. Se suele utilizar diagramas de entidad-relación (ER) para visualizar el modelo.
- **Modelo lógico:** Se traduce el modelo conceptual en un modelo lógico que puede implementarse en un sistema de gestión de bases de datos (DBMS), definiendo tablas, columnas, tipos de datos, y relaciones.
- **Normalización:** Es el proceso de organizar los datos para minimizar la duplicidad y asegurar la integridad de los datos.

Además, un buen diseño de base de datos toma en cuenta:

- **Índices** para mejorar el rendimiento en las consultas.
- **Integridad referencial**, para asegurar que las relaciones entre las tablas se mantengan coherentes.
- **Seguridad de datos**, implementando permisos y restricciones en el acceso a los datos.

4. Diseño de interfaces de usuario

El **diseño de interfaces de usuario** (UI/UX) se enfoca en cómo los usuarios interactuarán con el sistema. Un diseño efectivo proporciona una experiencia fluida, intuitiva y centrada en las necesidades del usuario. Para lograr esto, el proceso de diseño debe involucrar:

- **Investigación de usuario:** Identificar quiénes son los usuarios del sistema y cuáles son sus necesidades y objetivos.
- **Prototipado:** Crear prototipos de baja y alta fidelidad que permitan probar y validar la interfaz antes de la implementación.
- **Pruebas de usabilidad:** Evaluar el prototipo con usuarios reales para detectar problemas de navegación o de interacción.

El diseño de interfaces debe enfocarse en:

- **Estética visual:** Diseño agradable y limpio que mejore la experiencia sin generar confusión.

- **Accesibilidad:** Asegurar que el sistema pueda ser utilizado por personas con diferentes capacidades, incluyendo el uso de lectores de pantalla y otras herramientas de accesibilidad.
- **Flujo lógico:** Los pasos que sigue el usuario para completar una tarea deben ser claros y predecibles.

5. Diseño de procesos y de programas

El **diseño de procesos y programas** se centra en la planificación detallada del flujo de control y las operaciones que ejecutarán los distintos módulos del sistema. En esta fase, se establecen los algoritmos y los detalles de implementación de cada funcionalidad.

- **Diseño de procesos:** Implica la identificación de los diferentes flujos de trabajo dentro del sistema y cómo se manejarán las entradas, los cálculos y las salidas. Se utilizan **diagramas de flujo, pseudocódigo y diagramas de actividad** para visualizar estos procesos.
- **Diseño de programas:** Se define la lógica interna de cada componente. Esto incluye los métodos que implementarán las funcionalidades, cómo los datos se moverán entre las diferentes capas del sistema, y cómo se manejarán los errores. Además, se decide la estructura del código y se seleccionan los patrones de diseño, como:
 - **Factory Method:** Para crear objetos sin especificar la clase exacta de los objetos que se crearán.
 - **Singleton:** Garantiza que una clase solo tenga una instancia y proporciona un punto de acceso global a ella.

En esta etapa, también se consideran las pruebas unitarias y de integración, ya que el diseño del programa debe facilitar la creación de pruebas automáticas para validar la funcionalidad.

Mapa Mental



Bibliografía

- Pressman, R. S., & Maxim, B. R. (2014). *Ingeniería del Software: Un enfoque práctico*. McGraw-Hill.
- Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley.