

**UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE**

**DEPARTAMENTO DE CIENCIAS DE LA COMUNICACIÓN**

**SOFTWARE**



**MATERIA:**

ANALISIS Y DISEÑO DE SOFTWARE

**Nombre:**  
Alisson Nicole Clavijo Gutiérrez

**NRC:**  
9864

**Docente:**  
Ing. Alexis Darío Estévez Salazar

## **1. Tema**

Diseño de Arquitectura de Software

## **2. Introducción**

El diseño de software es un proceso esencial en el desarrollo de aplicaciones y sistemas informáticos. Implica la creación de una estructura conceptual y funcional que guía la construcción y organización del código para lograr soluciones eficientes, mantenibles y escalables. En este informe, exploraremos tres enfoques de diseño: Diseño Estructurado, Diseño Orientado a Objetos y Diseño de Interfaz de Usuario, analizando sus principios fundamentales y su relevancia en la creación exitosa de software.

## **3. Objetivos**

- Tener una visión general de los principios, características y métodos de diseño del software.
- Comprender los conceptos fundamentales de la arquitectura de software y su importancia en el desarrollo de aplicaciones y sistemas.
- Explorar las ventajas y desventajas de cada estilo arquitectónico en términos de escalabilidad, mantenibilidad y eficiencia.
- Identificar los principios clave para diseñar una arquitectura sólida, incluyendo la separación de preocupaciones, modularidad y cohesión.
- Presentar ejemplos reales de arquitecturas de software exitosas en diversas industrias y aplicaciones.

## **4. Marco Teórico**

### **4.1 Diseño Estructurado:**

El Diseño Estructurado se basa en la división del software en módulos lógicos y funcionales, lo que permite una comprensión clara del flujo de control y la interacción entre estos módulos. Los objetivos del Diseño Estructurado son mejorar la claridad del código, promover la reusabilidad y simplificar la detección de errores.

#### **Factores de calidad:**

- Acoplamiento y Cohesión.
- Diseño de Calidad.
- Resultados del Diseño.
- Diagrama de Estructura.
- Estrategias de Diseño.

#### **Construcción del Diagrama de Estructura**

- Diagramas UML.

Cada módulo se acompaña de los procedimientos de acuerdo a los cuales debe llevar a cabo su tarea, y de las estructuras de datos que procesa.

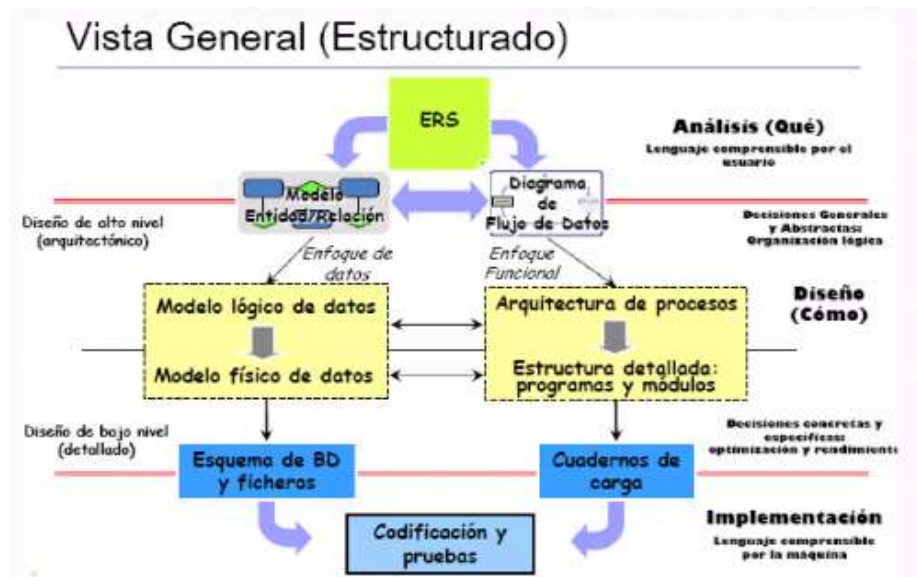


Figura 1. Visión General Diseño Estructurado.

#### Principios clave del Diseño Estructurado:

- **Modularidad:** División del sistema en módulos independientes, facilitando la comprensión y el mantenimiento.
- **Jerarquía:** Organización de módulos en una estructura jerárquica que refleja las dependencias funcionales.
- **Abstracción:** Ocultamiento de detalles internos de un módulo para facilitar su uso y comprensión.
- **Encapsulación:** Limitación del acceso a componentes internos de un módulo, reduciendo el impacto de cambios futuros.
- **Ejemplo de Aplicación:** Desarrollo de una plataforma educativa en línea con componentes diferenciados para registro de estudiantes, creación de cursos y seguimiento del progreso académico.

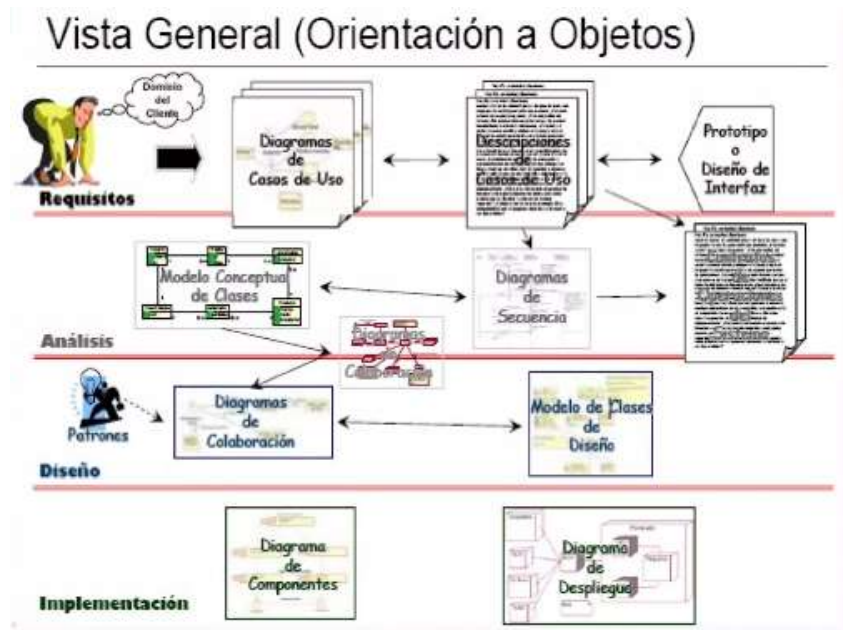
#### 4.2. Diseño Orientado a Objetos:

El Diseño Orientado a Objetos (OO) se basa en la creación de objetos que encapsulan datos y comportamientos relacionados. Promueve la reutilización, la flexibilidad y el modelado del mundo real en el diseño del software.

##### ventajas del diseño orientado a objetos

Son diversas las ventajas de su uso:

- Son fáciles de mantener. Los objetos vienen a representar entidades autocontenidas.
- Pudiera haber un mapeo obvio entre lo que son las entidades del mundo real y los objetos propios del sistema, esto para algunos de ellos.
- Los objetos vienen a ser propiamente componentes reutilizables.



**Figura 2. Visión General Diseño Orientado a Objetos.**

#### **Principios clave del Diseño Orientado a Objetos:**

- **Abstracción:** Representación de entidades del mundo real como objetos con atributos y métodos.
- **Encapsulación:** Protección de los detalles internos de un objeto y exposición de una interfaz controlada.
- **Herencia:** Creación de clases basadas en otras clases existentes para compartir atributos y comportamientos.
- **Polimorfismo:** Capacidad de objetos de diferentes clases para responder al mismo mensaje de manera única.
- **Ejemplo de Aplicación:** Implementación de un sistema de reservas y gestión de horarios para un centro de fitness, utilizando el enfoque orientado a objetos para crear clases como "Clase de Ejercicio", "Miembro" y "Reserva", con propiedades y funciones adaptadas a las necesidades de programación y seguimiento de la actividad física.

#### **4.3. Diseño de Interfaz de Usuario:**

El Diseño de Interfaz de Usuario (UI) se enfoca en crear una experiencia atractiva y funcional para los usuarios finales. Implica el diseño visual, la disposición de elementos y la interacción con los usuarios.

**¿Quién es el diseñador UI?** EL UI Designer como lo llaman en inglés, es el encargado de crear un medio de comunicación eficaz entre el software y el ser humano ¿cómo? por medio de unas reglas de diseño de la interfaz.

¿**Quién lo hace?** El encargado de diseñar la interfaz de usuario es el ingeniero del software que aplica a una serie de procesos y principios predefinidos según un diseño UI.

¿**Por qué es tan importante?** Si estamos ante una herramienta difícil de utilizar, forzamos al cliente a cometer errores para alcanzar una meta determinada, entonces, no le gustará.

¿**Cuál es el proceso del diseño de la interfaz de usuario?** El método se sustenta en base a una correlación de diferentes elementos que se necesitan mutuamente.

¿**Cuál es el producto final?** Se crea la coordinación visual por medio de herramientas para hacer mockup, organigramas, diagramas, etc. Aparece el prototipo de la interfaz que está interactivo y activo.

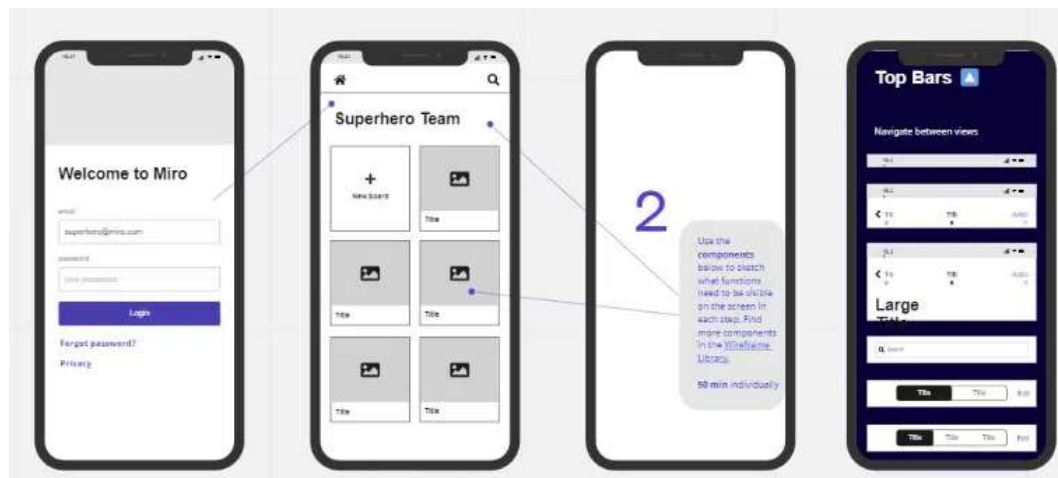


Figura 3. Interfaz de usuario.

### Principios clave del Diseño de UI:

- **Usabilidad:** Facilitar la interacción usuario-sistema con una interfaz intuitiva y fácil de usar.
- **Diseño Visual:** Uso de elementos gráficos, colores y tipografía para transmitir información y emociones.
- **Flujo de Usuario:** Diseño de rutas lógicas para las acciones del usuario y la navegación dentro de la interfaz.
- **Consistencia:** Mantenimiento de un estilo coherente en toda la interfaz para reducir la confusión del usuario.
- **Realimentación:** Proporcionar respuestas visuales o auditivas para las acciones del usuario.
- **Ejemplo de Aplicación:** Diseño de la interfaz de usuario de una aplicación de redes sociales, considerando la disposición de botones, colores y la navegación fluida entre perfiles y publicaciones.



Figura 4. Interfaz de usuario.

## 5. Conclusiones y Recomendaciones

### 5.1 Conclusiones

- El Diseño de Arquitectura de Software, el Diseño Estructurado, el Diseño Orientado a Objetos y el Diseño de Interfaz de Usuario son elementos cruciales en la creación de software efectivo y atractivo. Cada enfoque aborda aspectos específicos del desarrollo, desde la organización del código y la reutilización hasta la creación de experiencias de usuario agradables. Integrar estos enfoques en el proceso de diseño garantiza sistemas sólidos, fáciles de mantener y que satisfacen las necesidades y expectativas de los usuarios.

### 5.2 Recomendaciones

- Comprender los Requisitos: Antes de comenzar cualquier diseño, asegúrate de tener una comprensión clara y detallada de los requisitos del proyecto. Esto incluye tanto los requisitos funcionales como los no funcionales, para que el diseño se alinee con las necesidades del usuario y las expectativas del cliente.
- Enfoque Iterativo: Adopta un enfoque iterativo en el diseño. Crea versiones iniciales de tu diseño y solicita retroalimentación temprana de los stakeholders. Esto te permitirá realizar ajustes y mejoras a lo largo del proceso.
- Modularidad y Separación de Preocupaciones: Aplica el principio de modularidad, dividiendo el software en componentes cohesivos y acoplados débilmente. Cada módulo debe abordar una preocupación específica para facilitar el mantenimiento y la escalabilidad.
- Documentación Clara: Documenta el diseño de manera exhaustiva. Proporciona descripciones claras de cada módulo, su función, interfaces, relaciones y responsabilidades. Una documentación bien estructurada facilita la comprensión y colaboración entre miembros del equipo.

## 6. Bibliografía

- Diseño, E. (n.d.). Ucv.Cl. Retrieved August 8, 2023, from [http://zeus.inf.ucv.cl/~bcrawford/AULA\\_ICI\\_3242/0\\_2020\\_1\\_abc\\_del\\_dise\\_no.pdf](http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/0_2020_1_abc_del_dise_no.pdf)
- Ruiz, F., & González Harbour -Is, M. (n.d.). INGENIERÍA DEL SOFTWARE I. Unican.Es. Retrieved August 8, 2023, from <https://www.ctr.unican.es/asignaturas/is1/is1-t04-trans.pdf>
- Orientada, P., & Mestras, J. P. (n.d.). Patrones de diseño orientado a objetos. Ucm.Es. Retrieved August 8, 2023, from <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14pdoo.pdf>
- Cadavid, S. R., Osorio, A. S., Chiquito, J. L. S., Valencia, L. F. C., Marín, J. A. M., Arboleda, W. C., Galeano, J. M. A., & Garcia, E. M. L. (n.d.). Diseño de interfaz. Sucerman.com. Retrieved August 8, 2023, from <http://contenidos.sucerman.com/nivel4/desarrollo/unidad2/leccion4.html>
- Clase no. 1 unidad no. iii introduccion al analisis y diseño estructurado diseño estructurado. (n.d.). Slideshare.net. Retrieved August 8, 2023, from <https://es.slideshare.net/negroues/clase-no-1-unidad-no-iii-introduccion-al-analisis-y-diseo-estructurado-diseo-estructurado-4087770>