

Capítulo 2

Conceptos de Diseño

Mónica Elizabeth Páez Padilla

Resumen

En este capítulo, exploraremos los "Conceptos de Diseño", analizando detalles relevantes como las "Características y Principios de Diseño" que establecen bases para sistemas eficientes y flexibles, y los "Estándares de Diseño" que aseguran coherencia y calidad. Además, se profundizará en el "Diseño de Arquitectura de Software", considerando enfoques estructurados y orientados a objetos para sistemas adaptables. También se abordará el "Diseño de Interfaz de Usuario", resaltando su importancia en la experiencia del usuario y la creación de interfaces visuales e intuitivas. Por último, se explorarán los "Modelos de Interfaz de Usuario", incluyendo interfaces de línea de comandos, texto, gráficas y de voz. En resumen, este capítulo será una inmersión completa en los fundamentos del diseño en análisis y diseño de sistemas, estableciendo una base sólida para soluciones efectivas.

Palabras claves: principios de diseño, interfaz usuario, arquitectura de software.

Páez Padilla, M.E. (2023). Conceptos de Diseño. En M.E. Páez Padilla (ed). *Análisis y diseño de sistemas*. (pp. 86-121). Religación Press. <http://doi.org/10.46652/religacionpress.89.c83>



2.1 Concepto de Diseño

Dentro del campo de estudio relacionado con el análisis y la creación de sistemas., el concepto de diseño alude al procedimiento creativo y organizado que implica la formulación de soluciones tangibles y representaciones concretas para satisfacer las exigencias y necesidades que han sido identificadas durante la etapa de análisis. Este momento corresponde a la transformación de la información recopilada en un esquema detallado que delineará el procedimiento de construcción del sistema (Coronel, 2012).

En esta materia, el diseño involucra la generación de representaciones visuales y técnicas que delinean el funcionamiento proyectado del sistema, las interconexiones entre sus componentes, el enfoque de la interfaz para los usuarios y el planteamiento de soluciones a los problemas identificados. Esta tarea engloba la toma de decisiones relativas a la arquitectura del sistema, la organización de los datos, la circulación de la información, las interfaces de usuario, las reglas de operación y otros aspectos esenciales.

La ejecución del diseño se rige por los preceptos y principios reconocidos en la ingeniería de software, con el objetivo de asegurar que el sistema resultante se distinga por su eficacia, su mantenibilidad, su escalabilidad y su adhesión a los objetivos establecidos. Asimismo, el proceso de diseño debe abarcar factores tales como la usabilidad, la seguridad, el desempeño y la integración con otros sistemas.

En resumen, en el ámbito de análisis y diseño de sistemas, la fase de diseño conlleva la concepción detallada de la propuesta para la construcción del sistema, considerando de manera exhaustiva todas las implicaciones técnicas y funcionales necesarias para garantizar la materialización de un sistema eficaz y exitoso (Coronel, 2012).

2.1.1 Características y principios de diseño

Los atributos y fundamentos de diseño en el dominio del análisis y diseño de sistemas desempeñan un papel fundamental en garantizar que los sistemas resultantes sean eficaces, eficientes y capaces de satisfacer los requisitos y demandas de los usuarios. A continuación, se enumeran algunos de los atributos y fundamentos esenciales:

Características de Diseño:

- **Mantenibilidad:** El sistema debe ser diseñado de manera que permita futuras modificaciones y actualizaciones de manera fácil y sin generar impactos no deseados.
- **Flexibilidad:** El diseño debe ser lo suficientemente adaptable para acomodar cambios y evoluciones en los requisitos del sistema con el menor esfuerzo posible.

- **Usabilidad:** El diseño debe priorizar la experiencia del usuario, creando interfaces y flujos de trabajo intuitivos y fáciles de usar.
- **Seguridad:** La seguridad del sistema debe ser considerada en todas las etapas del diseño para proteger la información y prevenir posibles vulnerabilidades.
- **Escalabilidad:** El diseño debe permitir que el sistema pueda crecer y expandirse para manejar incrementos en la carga de trabajo sin comprometer su rendimiento.

Según Schumacher (2023), nos señala que los principios de diseño son:

Principios de Diseño:

Principio de Responsabilidad Única (SRP): Cada componente o clase debe tener una única responsabilidad, lo que facilita la comprensión, el mantenimiento y la reutilización.

Principio de Abierto/Cerrado (OCP): Las entidades del sistema (clases, módulos, etc.) deben estar abiertas a extensiones, pero cerradas a modificaciones, lo que promueve la extensibilidad sin afectar el código existente.

Principio de Sustitución de Liskov (LSP): Los objetos de una subclase deben ser capaces de reemplazar a los objetos de la clase padre sin afectar la integridad del programa.

Principio de Segregación de Interfaces (ISP): Los clientes no deben depender de interfaces que no utilizan. En lugar de interfaces monolíticas, se deben preferir interfaces más específicas.

Principio de Inversión de Dependencia (DIP): Los módulos de alto nivel no deben depender de los módulos de bajo nivel. Ambos deben depender de abstracciones. Además, los detalles deben depender de las abstracciones, no al revés.

Principio de Separación de Intereses (SoC): Los diferentes aspectos del sistema (como la lógica de negocio, la presentación y el almacenamiento) deben ser separados en módulos distintos para facilitar el mantenimiento y la evolución independiente.

Principio de Composición sobre Herencia: Se debe preferir la composición (combinación de objetos) en lugar de la herencia (extender clases existentes) para construir sistemas más flexibles y reutilizables.

Estas características y principios de diseño guían la creación de sistemas robustos, mantenibles y adaptativos en el campo del análisis y diseño de sistemas.

2.1.2 Estándares de diseño

En el contexto de análisis y diseño de sistemas, los estándares de diseño son pautas y directrices que se establecen para asegurar la coherencia, calidad y eficiencia en el proceso de diseño de sistemas. Estos estándares abarcan diversas áreas y aspectos

del diseño, desde la arquitectura del sistema hasta la interfaz de usuario.

Algunos ejemplos de estándares de diseño incluyen (Coronel, 2012).

Estándares de Diseño de Interfaz de Usuario: Establecen principios para la disposición de elementos, el uso de colores, tipografías, iconos y la organización de información en la interfaz para lograr una experiencia de usuario coherente y efectiva.

Estándares de Diseño de Arquitectura: Definen la estructura general del sistema, la organización de los componentes y la interacción entre ellos. Estos estándares pueden abordar la modularidad, escalabilidad y reutilización de componentes.

Estándares de Nomenclatura: Establecen convenciones para nombrar variables, funciones, clases y otros elementos del código, lo que facilita la comprensión y colaboración entre los miembros del equipo.

Estándares de Documentación: Establecen el formato y el contenido de la documentación técnica, como diagramas de flujo, diagramas de clases, descripciones de funciones y otros artefactos que documentan el diseño del sistema.

Estándares de Seguridad: Definen las medidas de seguridad que deben implementarse en el diseño para proteger la información y prevenir vulnerabilidades.

Estándares de Rendimiento: Establecen requisitos y metas relacionados con el rendimiento del sistema, como tiempos de respuesta y capacidad de manejo de cargas.

Estándares de Usabilidad: Definen criterios para asegurar que la interfaz de usuario sea intuitiva y fácil de usar, considerando la navegación, la interacción y la experiencia global del usuario.

Estándares de Codificación: Establecen directrices para el estilo de codificación, la indentación, el uso de comentarios y otras prácticas que hacen que el código sea más legible y mantenible.

Estándares de Pruebas: Definen enfoques para la planificación y ejecución de pruebas, incluyendo pruebas unitarias, de integración y de aceptación, con el objetivo de asegurar la calidad del sistema.

Estándares de Integración: Establecen prácticas para la integración de componentes y sistemas, asegurando que las partes funcionen correctamente juntas.

Los estándares de diseño son fundamentales para garantizar la consistencia y la calidad en el proceso de diseño de sistemas, así como para facilitar la colaboración entre los miembros del equipo y el mantenimiento a largo plazo de los sistemas desarrollados.

Autoevaluación 5

1. **¿Cuál es el propósito principal del diseño en el análisis y diseño de sistemas?**
 - a. Generar representaciones visuales y técnicas.
 - b. Identificar necesidades y exigencias.
 - c. Recopilar información detallada.
 - d. Definir la arquitectura del sistema.
2. **¿Cuál de las siguientes consideraciones NO es parte del proceso de diseño en ingeniería de software?**
 1. Usabilidad
 2. Desarrollo de código fuente
 3. Seguridad
 4. Integración con otros sistemas
3. **¿Qué engloba la fase de diseño en el análisis y diseño de sistemas?**
 - a. Toma de decisiones relativas a la arquitectura del sistema y organización de los datos.
 - b. Identificación de necesidades y exigencias.
 - c. Evaluación de la eficacia del sistema resultante.
 - d. Recopilación de información detallada sobre el sistema.
4. **¿Cuál de las siguientes características de diseño se refiere a la capacidad del sistema para utilizar eficientemente los recursos disponibles?**
 1. Mantenibilidad
 2. Flexibilidad
 3. Eficiencia
 4. Usabilidad

5. **¿Qué principio de diseño establece que las entidades del sistema deben estar abiertas a extensiones, pero cerradas a modificaciones?**
 - a. Principio de Responsabilidad Única (SRP)
 - b. Principio de Abierto/Cerrado (OCP)
 - c. Principio de Sustitución de Liskov (LSP)
 - d. Principio de Inversión de Dependencia (DIP)
6. **¿Cuál de los siguientes principios de diseño se enfoca en la separación de los diferentes aspectos del sistema en módulos distintos?**
 1. Principio de Separación de Intereses (SoC)
 2. Principio de Composición sobre Herencia
 3. Principio de Segregación de Interfaces (ISP)
 4. Principio de Responsabilidad Única (SRP)
7. **¿Cuál es el propósito del principio de Inversión de Dependencia (DIP) en el diseño de sistemas?**
 1. Los módulos de alto nivel deben depender de los módulos de bajo nivel.
 2. Los detalles deben depender de las abstracciones.
 3. Promover la extensibilidad sin afectar el código existente.
 4. Facilitar la comprensión y el mantenimiento del código.
8. **¿Qué característica de diseño se enfoca en la protección de la información y la prevención de vulnerabilidades en el sistema?**
 1. Escalabilidad
 2. Usabilidad
 3. Seguridad
 4. Flexibilidad

- 9. ¿Cuál es el propósito principal de los estándares de diseño en el análisis y diseño de sistemas?**
- Aumentar la complejidad del sistema.
 - Asegurar la coherencia, calidad y eficiencia en el proceso de diseño.
 - Establecer una interfaz de usuario atractiva.
 - Reducir la modularidad del sistema.
- 10. ¿Qué tipo de estándares se enfocan en la disposición de elementos, el uso de colores y la organización de información en la interfaz de usuario?**
- Estándares de Arquitectura
 - Estándares de Codificación
 - Estándares de Diseño de Interfaz de Usuario
 - Estándares de Pruebas
- 11. ¿Qué función cumplen los estándares de nomenclatura en el proceso de diseño?**
- Definir la estructura general del sistema.
 - Establecer requisitos de rendimiento.
 - Facilitar la comprensión y colaboración entre el equipo.
 - Planificar y ejecutar pruebas de aceptación.
- 12. ¿Por qué son importantes los estándares de seguridad en el diseño de sistemas?**
- Para definir la estructura del sistema.
 - Para mejorar el rendimiento del sistema.
 - Para proteger la información y prevenir vulnerabilidades.
 - Para determinar la eficiencia de la interfaz de usuario.

13. ¿Qué área abordan los estándares de documentación en el diseño de sistemas?

- a. Definición de nombres de variables.
- b. Estructura general del sistema.
- c. Formato y contenido de la documentación técnica.
- d. Requisitos de rendimiento del sistema.

14. ¿Cuál es el propósito de los estándares de rendimiento en el diseño de sistemas?

- a. Facilitar la colaboración entre el equipo.
- b. Definir la estructura del sistema.
- c. Establecer metas relacionadas con el rendimiento del sistema.
- d. Planificar y ejecutar pruebas de unidad.

15. ¿Para qué son esenciales los estándares de usabilidad en el diseño de sistemas?

1. Para definir la estructura del sistema.
2. Para mejorar el rendimiento del sistema.
3. Para asegurar que la interfaz de usuario sea intuitiva y fácil de usar.
4. Para establecer convenciones de nomenclatura.

2.2 Diseño de arquitectura de software

El proceso del “Diseño de Arquitectura de Software” representa una fase crucial en el ámbito de la asignatura de análisis y diseño de sistemas, donde se configura la estructura y disposición general del sistema de software que se busca crear. En esta etapa, se enfoca en la elaboración meticulosa de un plan exhaustivo que delineará la forma en que los distintos elementos del sistema interactuarán entre sí para cumplir con los preceptos y metas previamente identificados durante la fase de análisis (Laudon & Laudon, 2020).

Durante esta etapa, es posible emplear diferentes enfoques de diseño, tales como el diseño estructurado y el diseño orientado a objetos. El diseño estructurado busca segmentar el sistema en módulos y submódulos, precisando sus funciones y responsabilidades. Por otro lado, el diseño orientado a objetos traza el sistema mediante la utilización de objetos, clases y relaciones entre ellos, lo cual facilita representar de manera más fiel la realidad y las interacciones dentro del sistema.

Además, el proceso de diseño de arquitectura de software implica la toma de decisiones en relación con aspectos como la intercomunicación entre los componentes, la administración de datos, la escalabilidad, la seguridad y la eficiencia del sistema. Para visualizar la arquitectura, es posible recurrir a diagramas y modelos visuales, como los diagramas de componentes, de despliegue y de secuencia.

La creación de una arquitectura de software sólida es crucial para lograr sistemas resistentes, fáciles de mantener y adaptables. Además, facilita una comprensión más clara de cómo interactúan los componentes y cómo se logran los objetivos del sistema. En resumen, el proceso de diseño de la arquitectura de software se erige como un componente fundamental en el campo del análisis y diseño de sistemas, ya que establece las bases para el desarrollo exitoso de sistemas de software que sean funcionales y eficaces (Laudon & Laudon, 2020).

2.2.1 Diseño Estructurado

El enfoque “Diseño Estructurado” en el ámbito del análisis y diseño de sistemas representa una metodología fundamental cuyo objetivo es generar soluciones organizadas y eficaces mediante la descomposición de un sistema en elementos más pequeños y manejables. Esta técnica se centra en la subdivisión del sistema en módulos interconectados, cada uno de los cuales posee funciones y responsabilidades claramente definidas (James, 2007).

En el transcurso del proceso de diseño estructurado, se identifican las funciones primordiales del sistema y se organizan en módulos lógicos, lo que simplifica la comprensión y la futura mantención del sistema. Cada módulo desempeña una tarea específica y se relaciona con otros módulos a través de interfaces precisas. Esto impulsa la modularidad y favorece la reutilización de componentes en distintos proyectos.

Para representar gráficamente el diseño estructurado, se emplean diagramas como el “Diagrama de Estructura de Programa” (DEP), que representa la conexión entre los módulos y la jerarquía de llamadas entre ellos. Este enfoque asiste a los desarrolladores en la comprensión de cómo interactúan las partes del sistema y en la implementación de cambios con un impacto limitado en otras áreas

El diseño estructurado resulta particularmente valioso en sistemas complejos, al fragmentar la complejidad en elementos más manejables y al fomentar la colaboración entre los integrantes del equipo. No obstante, puede presentar limitaciones en la representación de relaciones entre objetos en sistemas más dinámicos. En su conjunto, el diseño estructurado conforma una herramienta esencial en el repertorio del análisis y diseño de sistemas, proporcionando un enfoque metódico para encarar proyectos de desarrollo de software (James, 2007).

2.2.2 Diseño Orientado a Objetos

El “Diseño Orientado a Objetos” en el contexto de análisis y diseño de sistemas constituye un enfoque crucial que se centra en la representación de sistemas mediante objetos y sus interacciones. Este enfoque se basa en la idea de modelar el mundo real a través de objetos con propiedades y comportamientos, creando así un sistema más cercano a la realidad y de mayor flexibilidad (Coronel & Morris, 2017).

Durante el proceso de diseño orientado a objetos, se identifican las entidades relevantes en el sistema y se modelan como clases, definiendo sus atributos y métodos. Las clases se organizan en jerarquías de herencia, lo que permite la reutilización de código y la captura de relaciones entre conceptos. Los objetos son instancias de estas clases y colaboran entre sí a través de mensajes, lo que simula la comunicación entre objetos en el mundo real.

El diseño orientado a objetos se apoya en diagramas como el “Diagrama de Clases”, que visualiza las clases, sus atributos y relaciones. Además, los “Diagramas de Secuencia” y “Diagramas de Colaboración” detallan las interacciones entre objetos en diferentes escenarios.

Este enfoque promueve la encapsulación, que protege la implementación interna de un objeto y expone solo su interfaz pública. También fomenta el polimorfismo, que permite que diferentes clases implementen métodos con el mismo nombre de manera específica. La herencia es otra característica clave, permitiendo que una clase herede atributos y métodos de otra.

El diseño orientado a objetos es altamente eficaz para modelar sistemas complejos y promover la reutilización de componentes. Sin embargo, su implementación exitosa requiere un profundo entendimiento de los principios orientados a objetos y una planificación cuidadosa para crear una arquitectura coherente y escalable. En resumen, el diseño orientado a objetos es un pilar esencial en la materia de análisis y diseño de sistemas, proporcionando un enfoque poderoso para crear sistemas flexibles y adaptativos que se asemejan a la realidad (Coronel & Morris, 2017).

Autoevaluación 6

1. ¿Qué representa el proceso de «Diseño de Arquitectura de Software» en el ámbito de la asignatura de análisis y diseño de sistemas?

1. Una fase opcional en el desarrollo de software.
2. Una etapa donde se configura la estructura general del sistema de software.
3. Un paso exclusivo del diseño orientado a objetos.

2. ¿Cuál es el enfoque del diseño estructurado en el proceso de diseño de arquitectura de software?

- a. Utilizar objetos y clases para modelar el sistema.
- b. Segmentar el sistema en módulos y submódulos.
- c. Representar la realidad del sistema de manera fiel.

3. ¿En qué se diferencia el diseño orientado a objetos del diseño estructurado en el proceso de diseño de arquitectura de software?

1. El diseño orientado a objetos utiliza objetos y clases, mientras que el diseño estructurado se enfoca en módulos.
2. El diseño estructurado es más adecuado para sistemas pequeños.
3. El diseño orientado a objetos no utiliza diagramas visuales.

4. ¿Cuáles son algunos de los aspectos clave que se deben considerar durante el proceso de diseño de arquitectura de software?

- a. La elección del lenguaje de programación.
- b. La ubicación geográfica de los desarrolladores.
- c. La intercomunicación entre componentes, la administración de datos, la escalabilidad y la seguridad.

5. ¿Por qué se considera esencial una arquitectura de software sólida en el desarrollo de sistemas de software?

1. Porque facilita la creación de documentación extensa.
2. Porque permite comprender cómo interactúan los elementos y alcanzar los objetivos del sistema.
3. Porque reduce el tiempo de desarrollo.

6. ¿Cuál es el objetivo principal del enfoque “Diseño Estructurado” en el análisis y diseño de sistemas?

- a. Generar soluciones caóticas.
- b. Descomponer un sistema en elementos más pequeños y manejables.
- c. Crear sistemas sin módulos interconectados.

7. ¿Qué se hace durante el proceso de diseño estructurado para simplificar la comprensión y la mantención del sistema?

1. Se eliminan todos los módulos interconectados.
2. Se organizan las funciones primordiales en módulos lógicos.
3. Se crean interfaces complejas entre los módulos.

8. ¿Cómo se fomenta la reutilización de componentes en el diseño estructurado?

- a. No se fomenta la reutilización de componentes.
- b. Cada módulo desempeña una tarea específica.
- c. No se relacionan los módulos entre sí.

9. ¿Qué tipo de diagrama se utiliza comúnmente para representar gráficamente el diseño estructurado?

1. Diagrama de flujo.
2. Diagrama de Gantt.
3. Diagrama de Estructura de Programa (DEP).

10. ¿Cuándo resulta especialmente valioso el diseño estructurado en el desarrollo de sistemas?

- a. En sistemas dinámicos.
- b. En sistemas sin colaboración entre equipos.
- c. En sistemas complejos, al fragmentar la complejidad en elementos más manejables y fomentar la colaboración entre los integrantes del equipo.

11. ¿Qué es el «Diseño Orientado a Objetos» en el contexto del análisis y diseño de sistemas?

1. Un enfoque que no utiliza objetos ni interacciones.
2. Un enfoque que se centra en la representación de sistemas mediante objetos y sus interacciones.
3. Un enfoque que se basa en la programación lineal.

12. ¿Cómo se modelan las entidades relevantes en el diseño orientado a objetos?

- a. Como funciones independientes.
- b. Como objetos con propiedades y comportamientos en forma de clases.
- c. Como módulos interconectados.

13. ¿Qué permite la herencia en el diseño orientado a objetos?

1. La creación de objetos.
2. La reutilización de código y la captura de relaciones entre conceptos.
3. La encapsulación de métodos.

14. ¿Qué tipo de diagramas se utilizan para visualizar las clases y sus relaciones en el diseño orientado a objetos?

1. Diagramas de flujo.
2. Diagramas de estructura.
3. Diagramas de clases.

15. ¿Qué características clave fomenta el diseño orientado a objetos, como la encapsulación, el polimorfismo y la herencia?

- a. La obsolescencia del código.
- b. La confusión entre objetos.
- c. La flexibilidad y la reutilización de componentes

2.3 Diseño Interfaz de usuario

El “Diseño de Interfaz de Usuario” juega un rol crucial en el ámbito de análisis y diseño de sistemas al focalizarse en forjar interacciones efectivas y atractivas entre los usuarios y el sistema. Esta etapa se dedica a concebir cómo los usuarios interactuarán con la aplicación, garantizando que la experiencia sea fácil, eficiente y gratificante (Tavera, 2018).

En el transcurso del proceso de diseño de interfaz de usuario, se toman en cuenta aspectos tales como la disposición de elementos en la pantalla, la elección de colores, fuentes, íconos y la navegación entre pantallas. El objetivo es crear una interfaz que permita a los usuarios cumplir sus tareas de manera fluida y sin dificultades, maximizando la usabilidad y minimizando la curva de aprendizaje.

La trascendencia de la interfaz de usuario radica en su rol como el principal punto de contacto entre los usuarios y el sistema, impactando directamente en su percepción y satisfacción. Un diseño que sea visualmente atractivo y funcional puede influir positivamente en la aceptación y adopción del sistema por parte de los usuarios.

Durante la etapa de diseño de interfaz de usuario, es posible emplear wireframes y mockups para visualizar la estructura y disposición de elementos en pantalla antes de llevarlo a cabo. Se busca alcanzar un equilibrio entre un diseño que sea visualmente agradable y una experiencia de usuario que sea intuitiva y eficiente (Cáceres, 2014).

Asimismo, el diseño de interfaz de usuario debe tener en consideración la accesibilidad, garantizando que el sistema sea usable por personas con discapacidades. Además, el diseño debe ser coherente con la identidad visual de la marca y en sintonía con las expectativas y necesidades de los usuarios.

En síntesis, el diseño de interfaz de usuario es un componente esencial en la materia de análisis y diseño de sistemas, ya que contribuye en gran medida a la usabilidad, aceptación y éxito del sistema (Paredes Hernández & Velasco Espitia, 1998).

2.3.1 La importancia de la interfaz de usuario

El diseño de la interfaz de usuario, que también se conoce como diseño de la interfaz o ingeniería de la interfaz, implica la actividad de establecer múltiples aspectos que afectan la apariencia y el funcionamiento externo de las interfaces de usuario en diversos tipos de sistemas, que van desde computadoras y dispositivos móviles hasta software y páginas web. Esta disciplina se alinea con el diseño industrial y busca optimizar la usabilidad y la experiencia del usuario. Su propósito es simplificar y agilizar la interacción entre el usuario y el sistema, manteniendo el enfoque en los objetivos del usuario (Domínguez Coutiño, 2012).

En esta perspectiva centrada en el usuario, el diseño gráfico y la tipografía se combinan para mejorar tanto la usabilidad como la estética, equilibrando la funcionalidad técnica con elementos visuales. El diseño de la interfaz de usuario abarca una

diversidad de disciplinas, como diseño gráfico, diseño industrial, diseño web y ergonomía, y su aplicación abarca desde sistemas informáticos hasta proyectos de envergadura como el desarrollo de aeronaves comerciales. El objetivo final es lograr una comunicación efectiva y eficiente, así como una adaptabilidad a las necesidades cambiantes de los usuarios, al tiempo que se mantenga una operación técnica óptima (James, 2007).

2.3.2 Diseño visual, funcional e intuitivo

En el desarrollo de software, el diseño abarca tres elementos esenciales: visual, funcional e intuitivo. En primer lugar, el diseño visual se enfoca en la estética y presentación gráfica de la interfaz de usuario, buscando crear una representación atractiva y coherente de la información. Por otro lado, el diseño funcional se concentra en asegurar el funcionamiento efectivo y sin problemas de todas las funciones y características, con el objetivo de cumplir con los requisitos y necesidades de los usuarios (Coronel, 2012).

Por último, el diseño intuitivo desempeña un papel crucial al facilitar la interacción del usuario con el sistema. Esta dimensión se propone reducir la curva de aprendizaje al crear una interfaz comprensible y fácil de usar, permitiendo que los usuarios realicen tareas de manera natural y sin confusiones. En conjunto, estos tres aspectos del diseño operan en armonía para desarrollar sistemas de software visualmente atractivos, funcionalmente eficientes y de fácil comprensión para los usuarios.

Autoevaluación 7

1. **¿Cuál es el objetivo principal del diseño de interfaz de usuario?**
 - a. Crear aplicaciones complejas.
 - b. Maximizar la curva de aprendizaje.
 - c. Facilitar interacciones efectivas y atractivas entre usuarios y el sistema.
 - d. Elegir colores y fuentes llamativos.
2. **¿Qué aspectos se consideran durante el proceso de diseño de interfaz de usuario?**
 - a. El costo del proyecto.
 - b. La elección de colores y fuentes.
 - c. La ubicación geográfica de los usuarios.
 - d. La disposición de elementos en la pantalla y la navegación entre pantallas.
3. **¿Por qué es importante la accesibilidad en el diseño de interfaz de usuario?**
 1. Porque solo afecta a un pequeño grupo de usuarios.
 2. Porque garantiza que el sistema sea usable por personas con discapacidades.
 3. Porque no tiene ningún impacto en la usabilidad.
 4. Porque aumenta la curva de aprendizaje.
4. **¿Cómo puede influir positivamente el diseño de interfaz de usuario en la adopción del sistema por parte de los usuarios?**
 1. Haciendo que el sistema sea más complejo.
 2. Aumentando la curva de aprendizaje.
 3. Siendo visualmente atractivo y funcional.
 4. No tiene ningún impacto en la adopción del sistema.

- 5. ¿Qué herramientas se pueden utilizar durante la etapa de diseño de interfaz de usuario para visualizar la estructura de la pantalla?**
- Diagramas de flujo.
 - Calculadoras.
 - Wireframes y mockups.
 - Hojas de cálculo.
- 6. ¿Cuál es el propósito principal del diseño de la interfaz de usuario?**
- Optimizar la estética.
 - Simplificar la comunicación técnica.
 - Mejorar la usabilidad y la experiencia del usuario.
 - Centrarse en los objetivos del sistema.
- 7. ¿Qué disciplinas se involucran en el diseño de la interfaz de usuario?**
- Solo diseño gráfico.
 - Diseño industrial y ergonomía.
 - Diseño web y desarrollo de software.
 - Todas las anteriores.
- 8. ¿Qué se busca equilibrar en el diseño de la interfaz de usuario?**
- La usabilidad y la estética.
 - La programación y la tipografía.
 - La velocidad y la seguridad.
 - La conectividad y la funcionalidad.
- 9. ¿Cuál es el objetivo final del diseño de la interfaz de usuario?**
- Mantener una operación técnica óptima.
 - Priorizar los objetivos del sistema.
 - Lograr una comunicación efectiva y eficiente.
 - Ignorar las necesidades cambiantes de los usuarios.

10. ¿En qué tipos de sistemas se aplica el diseño de la interfaz de usuario?

- a. Únicamente en sistemas informáticos.
- b. Exclusivamente en proyectos de diseño industrial.
- c. En una variedad de sistemas, incluyendo computadoras, dispositivos móviles, software y sitios web.
- d. Solo en proyectos relacionados con aeronaves comerciales.

11. ¿Cuáles son los tres elementos esenciales del diseño en el desarrollo de software mencionados en el texto?

1. Visual, interactivo y técnico.
2. Visual, funcional e intuitivo.
3. Estético, técnico y eficiente.
4. Atractivo, rápido y completo.

12. ¿En qué se enfoca principalmente el diseño visual en el desarrollo de software?

- a. En garantizar que todas las funciones funcionen sin problemas.
- b. En crear una interfaz fácil de usar.
- c. En la estética y la presentación gráfica de la interfaz de usuario.
- d. En reducir la curva de aprendizaje.

13. ¿Cuál es el objetivo principal del diseño funcional en el desarrollo de software?

- a. Crear una interfaz atractiva.
- b. Cumplir con los requisitos y necesidades de los usuarios.
- c. Reducir la curva de aprendizaje.
- d. Centrarse en la presentación gráfica.

14.¿Qué papel desempeña el diseño intuitivo en la interacción del usuario con el sistema?

1. Aumenta la curva de aprendizaje.
2. Complica la realización de tareas por parte de los usuarios.
3. Facilita la interacción del usuario al reducir la curva de aprendizaje.
4. No tiene impacto en la interacción del usuario.

15.¿Cómo trabajan en conjunto estos tres aspectos del diseño en el desarrollo de software?

1. Operan en discordia.
2. Operan de manera independiente.
3. Operan en armonía para desarrollar sistemas visualmente atractivos, funcionalmente eficientes y de fácil comprensión para los usuarios.
4. Solo el diseño visual y el diseño funcional trabajan juntos.

2.4 Los modelos más destacados de user interfaz

Implica un medio que simplifica el manejo de un software o hardware específico por parte de un usuario. En este sentido, la Interfaz de Usuario (UI) permite a un cliente llevar a cabo acciones e interactuar con las múltiples opciones proporcionadas por un dispositivo electrónico. Una interfaz de usuario efectiva se destaca por su simplicidad, facilidad de comprensión y alto nivel de usabilidad (Martins, 2023).

Además, también abarca las conexiones entre los usuarios y las aplicaciones web, creando un punto central de enfoque para las compañías que buscan mejorar globalmente la experiencia del usuario.

Según Martins (2023), dice que, para lograrlo, hace uso de una variedad de elementos y controles visuales que enriquecen la interacción de los clientes.

2.4.1 Interfaz de línea de comandos (CLI)

Es una de las interfaces más longevas que continúa en uso en la actualidad. Se basa en texto como medio para interactuar con la computadora, permitiendo la ejecución y administración de programas o archivos. Un ejemplo de esta interfaz de usuario es el sistema operativo MS-DOS, además del Shell de comandos que integra el sistema operativo Windows.

2.4.2 Interfaz de usuario de texto (TUI)

La Interfaz de Usuario de Texto, conocida como Text User Interface en inglés, es una forma de interfaz que se crea mediante el uso de caracteres. Para lograr esto, el sistema se enlaza con un dispositivo, como un teclado, y se concentra en establecer conexiones con el hardware de un dispositivo. Esta interfaz se emplea comúnmente en la configuración de sistemas operativos junto con sus terminales para acceder a diversos programas.

2.4.3 Interfaz gráfica de usuario (GUI)

Denominadas también como Interfaces Gráficas de Usuario (GUI, por sus siglas en inglés), estas representan el formato de interfaz más ampliamente empleado en la contemporaneidad. Este entorno se fundamenta en elementos gráficos y representaciones visuales que muestran tanto la información como las alternativas disponibles para la comunicación entre el usuario y el aparato. En efecto, este tipo de interfaz se diseña específicamente para su uso en dispositivos móviles.

2.4.4 Interfaz de usuario de voz (VUI)

La Interfaz de Usuario por Voz identifica y comprende patrones de habla para activar y llevar a cabo diversas funciones en un sistema. Como ejemplo destacado, mencionamos a Google Nest o Alexa, que, al recibir una orden, pueden reconocer lo

que el usuario dice y responder o llevar a cabo una acción. Estas capacidades proporcionan una mayor comodidad y flexibilidad al permitir que el usuario interactúe sin necesidad de contacto físico, especialmente cuando se está desplazando.

2.4.5 Interfaz de usuario natural (NUI)

Es una categoría de interfaz que proporciona una forma natural e intuitiva de interactuar con el usuario. Su base se encuentra en el uso de gestos y toques en dispositivos táctiles, como se observa en la tecnología Kinect de Xbox. En consecuencia, la Interfaz de Usuario Natural (NUI, por sus siglas en inglés) tiene la capacidad de detectar y comprender las acciones humanas derivadas de movimientos y expresiones faciales.

Autoevaluación 8

1. **¿Qué es una Interfaz de Usuario (UI) en el contexto de la informática?**
 1. Un dispositivo electrónico
 2. Un software específico
 3. Un medio que simplifica la interacción entre un usuario y un dispositivo o software
 4. Un punto central de enfoque para las compañías
2. **¿Cuál es una característica importante de una interfaz de usuario efectiva?**
 1. Complejidad
 2. Facilidad de comprensión
 3. Dificultad de uso
 4. Falta de opciones
3. **¿Qué objetivo busca lograr una interfaz de usuario efectiva?**
 - a. Complicar la interacción del cliente
 - b. Reducir la usabilidad
 - c. Simplificar la interacción del cliente
 - d. Aumentar la complejidad del software
4. **¿Qué elementos y controles visuales se utilizan en una interfaz de usuario para enriquecer la interacción del cliente?**
 - a. Reducir la interacción
 - b. Simplificar la experiencia
 - c. Añadir complejidad
 - d. Enriquecer la interacción

- 5. ¿Qué función cumple una interfaz de usuario en las aplicaciones web?**
1. Simplificar la experiencia del usuario
 2. Complicar la experiencia del usuario
 3. No tiene ninguna función en las aplicaciones web
 4. Facilitar la programación de aplicaciones web
- 6. ¿Por qué es importante para las compañías mejorar la experiencia del usuario a través de una interfaz de usuario efectiva?**
- a. Porque aumenta la complejidad del software
 - b. Porque reduce la usabilidad
 - c. Porque atrae a más usuarios y mejora la satisfacción del cliente
 - d. Porque complica la interacción del cliente
- 7. ¿Cuál es una característica destacada de la Interfaz de Línea de Comandos (CLI)?**
1. Utiliza elementos visuales e imágenes.
 2. Se basa en texto para interactuar con la computadora.
 3. Permite la interacción mediante gestos.
 4. Se utiliza principalmente en dispositivos móviles.
- 8. ¿Cuál es un ejemplo de sistema operativo que utiliza una Interfaz de Línea de Comandos (CLI)?**
- a. MacOS
 - b. Linux
 - c. Windows
 - d. Android

9. ¿Cuál es un ejemplo de un Shell de comandos que integra el sistema operativo Windows?

1. Terminal
2. PowerShell
3. Command Prompt
4. Terminal Emulator

10. ¿Qué significa la sigla TUI en inglés?

- a. Texto Unificado Internacional
- b. Texto Universal Integrado
- c. Text User Interface
- d. Tecnología de Usuarios Interactivos

11. ¿Cómo se crea una Interfaz de Usuario de Texto (TUI)?

1. Utilizando imágenes y elementos visuales.
2. Mediante el uso de caracteres.
3. A través de gestos y movimientos.
4. Usando comandos de voz.

12. ¿En qué se concentra una TUI en términos de hardware?

1. En conectar dispositivos móviles.
2. En establecer conexiones con el hardware de un dispositivo.
3. En mejorar la calidad de imagen.
4. En aumentar la velocidad de procesamiento.

13. ¿Dónde se utiliza comúnmente la TUI?

- a. En dispositivos móviles.
- b. En la configuración de sistemas operativos y terminales.
- c. En la navegación web.
- d. En la edición de videojuegos.

14. ¿Qué significa la sigla GUI en inglés?

- a. General User Interface
- b. Graphics User Interface
- c. Gesture User Interface
- d. Good User Interaction

15. ¿En qué se basa la Interfaz Gráfica de Usuario (GUI) para la interacción entre el usuario y el dispositivo?

1. En elementos auditivos.
2. En elementos visuales e imágenes.
3. En comandos de voz.
4. En gestos y toques.

16. ¿Qué tipo de acciones ejecuta una Voice User Interface (VUI) mediante el reconocimiento de patrones vocales?

1. Acciones visuales.
2. Acciones gestuales.
3. Acciones de texto.
4. Acciones de voz.

17. ¿Cuál es un ejemplo de dispositivo que utiliza una VUI según el texto?

- a. iPhone
- b. Xbox Kinect
- c. Sistema operativo Windows
- d. Dispositivos móviles

18. ¿Cuál es la base de la Interfaz de Usuario Natural (NUI) para la interacción con el usuario?

1. Uso de comandos de voz.
2. Uso de gestos y toques en dispositivos táctiles.
3. Uso de comandos de texto.
4. Uso de elementos visuales.

19. ¿Dónde se observa comúnmente el uso de gestos y toques en dispositivos táctiles como parte de una NUI?

- a. En la tecnología Kinect de Xbox.
- b. En la navegación web.
- c. En sistemas operativos de computadoras.
- d. En reproductores de música.

20. ¿Qué capacidad tiene una Interfaz de Usuario Natural (NUI) según el texto?

1. Detectar y comprender acciones de comandos de voz.
2. Detectar y comprender acciones humanas derivadas de movimientos y expresiones faciales.
3. Mostrar información visual.
4. Reproducir música de forma natural.

2.5 Conclusión

En conclusión, podemos decir que, en este capítulo, hemos explorado conceptos fundamentales del diseño en el desarrollo de software, discutiendo características, principios y estándares que guían la creación de soluciones efectivas y coherentes. Hemos analizado la importancia del diseño de la arquitectura de software, considerando enfoques estructurados y orientados a objetos para la construcción de sistemas sólidos y adaptables.

Destacamos la vitalidad de la Interfaz de Usuario (UI) en el diseño de software, reconociendo su rol crucial en mejorar la experiencia del usuario. Exploramos diversos tipos de interfaces, desde la tradicional Interfaz de Línea de Comandos (CLI) hasta la moderna Interfaz de Usuario de Texto (TUI) y la innovadora

Interfaz de Usuario Natural (NUI), cada una con un enfoque distintivo para interactuar con los sistemas.

En síntesis, abarcamos una amplia gama de temas en el diseño de software, desde fundamentos hasta modelos de interfaces destacados. Un diseño efectivo no solo influye en la apariencia visual, sino también en la funcionalidad, usabilidad y satisfacción del usuario en última instancia.

Referencias

- Cáceres, E. (2014). *Análisis y Diseño de Sistemas de Información*. <https://acortar.link/X2axIu>
- Coronel, C., Morris, S., y Rob, P. (2012). *Base de datos, diseño, implementación y administración*. CENGAGE Learning.
- James, O. (2007). *Diseño de Sistemas de Información Gerencial*. McGraw-Hill
- León Yacelga, A.R., Acosta Espinoza, J.L., & Díaz Vásquez, R.A. (2021). Aplicación de la metodología incremental en el desarrollo de sistemas de información. *Universidad Y Sociedad*, 13(5), 175-182. <https://rus.ucf.edu/cu/index.php/rus/article/view/2223>
- Martínez Canelo, M. (2020, 24 de junio). ¿Qué son los Patrones de Diseño de software? *Design Patterns. Profile Software Services*. <https://acortar.link/SDeRzu>
- Paredes Hernández, E., & Velasco Espitia, M. (1998). *Análisis y Diseño de Sistemas de Información*. Universidad de Pamplona

Design Concepts

Conceitos de design

Mónica Elizabeth Páez Padilla

Instituto de Educación Superior Nelson Torres | Cayambe | Ecuador

<https://orcid.org/0009-0006-1030-1394>

monica.paez@intsuperior.edu.ec

Abstract

In this chapter, we will explore the "Design Concepts", analyzing relevant details such as the "Design Characteristics and Principles" that establish the basis for efficient and flexible systems, and the "Design Standards" that ensure consistency and quality. In addition, "Software Architecture Design" will be discussed in depth, considering structured and object-oriented approaches for adaptive systems. "User Interface Design" will also be addressed, highlighting its importance in the user experience and the creation of visual and intuitive interfaces. Finally, "User Interface Models" will be explored, including command line, text, graphical and voice interfaces. In summary, this chapter will be a complete immersion into the fundamentals of design in systems analysis and design, establishing a solid foundation for effective solutions.

Keywords: design principles, user interface, software architecture.

Resumo

Neste capítulo, exploraremos os "Conceitos de design", analisando detalhes relevantes, como os "Recursos e princípios de design", que estabelecem a base para sistemas eficientes e flexíveis, e os "Padrões de design", que garantem a coerência e a qualidade. Além disso, o "Design de arquitetura de software" será discutido em profundidade, considerando abordagens estruturadas e orientadas a objetos para sistemas adaptativos. O "Design da interface do usuário" também será abordado, destacando sua importância para a experiência do usuário e a criação de interfaces visuais e intuitivas. Por fim, serão explorados os "Modelos de interface do usuário", incluindo interfaces de linha de comando, de texto, gráficas e de voz. Em resumo, este capítulo será uma imersão completa nos fundamentos do design na análise e no design de sistemas, estabelecendo uma base sólida para soluções eficazes.

Palavras-chave: princípios de design, interface do usuário, arquitetura de software.