



# Unidad de Trabajo 4

## Diseño

### Reutilización y

### Planificación

# ÍNDICE

<a href="#">1. Introducción a Diseño, Reutilización y Planificación.....</a>	<a href="#">3</a>
<a href="#">2. Principios del Diseño Eficiente.....</a>	<a href="#">3</a>
<a href="#">2.1 Importancia del Diseño en Proyectos de Software.....</a>	<a href="#">3</a>
<a href="#">2.2 Patrones de Diseño: Estructura y Comportamiento.....</a>	<a href="#">3</a>
<a href="#">2.3 Consideraciones Estéticas y Funcionales.....</a>	<a href="#">3</a>
<a href="#">3. Reutilización de Componentes.....</a>	<a href="#">4</a>
<a href="#">3.1 Definición y Beneficios de la Reutilización.....</a>	<a href="#">4</a>
<a href="#">3.2 Tipos de Componentes Reutilizables.....</a>	<a href="#">4</a>
<a href="#">3.3 Herramientas y Estrategias para Fomentar la Reutilización.....</a>	<a href="#">4</a>
<a href="#">4. Planificación de Proyectos.....</a>	<a href="#">5</a>
<a href="#">4.1 Técnicas y Herramientas para la Planificación.....</a>	<a href="#">5</a>
<a href="#">4.2 Estimación de Tiempos y Recursos.....</a>	<a href="#">5</a>
<a href="#">4.3 Identificación de Riesgos y Plan de Mitigación.....</a>	<a href="#">5</a>

# 1. Introducción a Diseño, Reutilización y Planificación

El **diseño**, la **reutilización** y la **planificación** son pilares fundamentales en el desarrollo de software moderno. Estos conceptos permiten optimizar recursos, mejorar la calidad del producto final y asegurar el éxito del proyecto mediante una gestión efectiva del tiempo y los costos.

- **Diseño:** Es el proceso de crear una representación estructurada del sistema, incluyendo su arquitectura, componentes y relaciones.
- **Reutilización:** Implica aprovechar componentes existentes para reducir el esfuerzo de desarrollo y garantizar consistencia y calidad.
- **Planificación:** Permite organizar y coordinar las actividades del proyecto, asegurando que los objetivos se cumplan dentro de los plazos y recursos establecidos.

Esta unidad se centra en explorar estos conceptos, destacando su importancia y proporcionando herramientas prácticas para su aplicación.

## 2. Principios del Diseño Eficiente

### 2.1 Importancia del Diseño en Proyectos de Software

El diseño es crucial para garantizar que el sistema sea funcional, escalable y fácil de mantener. Un buen diseño:

- Reduce errores en fases posteriores del desarrollo.
- Facilita la implementación y el mantenimiento.
- Mejora la experiencia del usuario final.

### 2.2 Patrones de Diseño: Estructura y Comportamiento

Los patrones de diseño son soluciones probadas para problemas comunes en el desarrollo de software. Se dividen en:

- **Patrones estructurales:** Se centran en cómo se organizan y relacionan los componentes del sistema (e.g., patrón de fachada, adaptador).
- **Patrones de comportamiento:** Abordan la interacción y responsabilidad entre los objetos (e.g., patrón observador, cadena de responsabilidades).

### 2.3 Consideraciones Estéticas y Funcionales

Un diseño efectivo debe equilibrar los aspectos estéticos y funcionales:

- **Estéticos:** Incluyen la interfaz de usuario y la experiencia visual.
- **Funcionales:** Abarcan el rendimiento, la usabilidad y la accesibilidad.

## 3. Reutilización de Componentes

### 3.1 Definición y Beneficios de la Reutilización

La reutilización consiste en aprovechar módulos, bibliotecas o fragmentos de código existentes para nuevos desarrollos. Sus beneficios incluyen:

- Eficiencia: Ahorra tiempo y recursos.
- Calidad: Los componentes reutilizables suelen ser más estables al estar probados previamente.
- Consistencia: Promueve estándares uniformes en los proyectos.

### 3.2 Tipos de Componentes Reutilizables

- Código Fuente: Clases y funciones compartidas.
- Librerías: Conjuntos de herramientas específicas (e.g., bibliotecas gráficas o de acceso a datos).
- APIs: Interfaz para interactuar con funcionalidades externas.
- Plantillas y Frameworks: Estructuras predefinidas para desarrollo rápido.

### 3.3 Herramientas y Estrategias para Fomentar la Reutilización

- Uso de repositorios compartidos (e.g., GitHub, Bitbucket).
- Creación de documentación clara para facilitar el uso de componentes reutilizables.
- Promoción de cultura de reutilización dentro del equipo de desarrollo.

## 4. Planificación de Proyectos

### 4.1 Técnicas y Herramientas para la Planificación

La planificación efectiva requiere técnicas y herramientas que permitan organizar y priorizar tareas:

Técnicas:

- Método SMART: Establecer objetivos específicos, medibles, alcanzables, relevantes y con un tiempo definido.
- Diagrama de Gantt: Visualización de tareas y su cronograma.

Herramientas:

- Jira: Gestión de tareas y seguimiento en metodologías ágiles.
- Microsoft Project: Planificación detallada y gestión de recursos.

### 4.2 Estimación de Tiempos y Recursos

- Tiempos: Uso de métodos como Planning Poker para estimar el esfuerzo de desarrollo en entornos ágiles.
- Recursos: Asignación de recursos humanos y materiales basados en la complejidad de las tareas.

### 4.3 Identificación de Riesgos y Plan de Mitigación

- Riesgos comunes: Cambios en los requisitos, falta de recursos, retrasos en entregas.
- Plan de mitigación: Estrategias preventivas y correctivas, como realizar pruebas continuas o tener recursos de respaldo.