

## **Definición de Generalización**

La generalización es una técnica utilizada para gestionar la complejidad agrupando entidades similares en clases más generales. Permite organizar y simplificar la información al identificar características comunes dentro de un conjunto de elementos, lo que facilita su manejo, reutilización y mantenimiento en el modelado de sistemas

## **Características de la Generalización**

1. Jerarquización: Organiza elementos en una estructura de clases con relaciones de herencia.
2. Abstracción: Permite centrarse en las características comunes, evitando detalles específicos.
3. Reutilización: Los atributos y operaciones de una clase general pueden ser heredados por sus subclases.
4. Mantenimiento simplificado: Los cambios en la superclase se reflejan automáticamente en las subclases.
5. Uso en lenguajes orientados a objetos: Se implementa mediante la herencia en lenguajes como Java, C++, Python, etc.
6. Representación en UML: Se ilustra mediante una flecha que apunta a la clase más general.

## **Ejemplo de Generalización**

En un sistema de gestión médica, los diferentes tipos de médicos pueden agruparse en una jerarquía de generalización:

Superclase:

- Médico (atributos: nombre, número telefónico, operaciones: registrar, dar de baja).

Subclases:

- Médico de cabecera (atributos adicionales: nombre del consultorio, dirección).
- Médico de hospital (atributos adicionales: número de personal, departamento).
- Consultor (médico experimentado con plenas responsabilidades).
- Médico de equipo (trabaja sin supervisión en un equipo de consultores).
- Médico practicante (recién graduado, requiere supervisión).

Este modelo permite representar de manera eficiente la estructura organizativa de los médicos y facilita la gestión de información dentro del sistema.

### **Ejemplo de Generalización: Vehículos**

En un sistema de gestión de transporte, podemos generalizar los distintos tipos de vehículos en una jerarquía de clases:

**Superclase:**

- **Vehículo** (atributos: marca, modelo, año, velocidad máxima, operaciones: acelerar, frenar, encender).

**Subclases:**

- **Automóvil** (atributos adicionales: número de puertas, tipo de combustible).
- **Motocicleta** (atributos adicionales: tipo de motor, tiene sidecar o no).
- **Camión** (atributos adicionales: capacidad de carga, número de ejes).

Cada tipo de vehículo hereda los atributos y operaciones de la clase Vehículo, pero también tiene características específicas. Gracias a esta generalización, podemos gestionar distintos vehículos de manera eficiente sin duplicar información en el sistema.

Proceso Unificado es una metodología de desarrollo de software basada en la orientación a objetos, diseñada para guiar a los equipos en la construcción de sistemas de software de manera iterativa e incremental. Se centra en minimizar riesgos y mejorar la calidad del producto final.

Características principales:

Ciclo de vida iterativo e incremental: Se divide en múltiples iteraciones, cada una de las cuales produce un incremento del sistema con funcionalidad añadida.

Centrada en la arquitectura: La estructura del sistema se define y estabiliza tempranamente en el proceso.

Basada en el modelado: Usa UML (Lenguaje Unificado de Modelado) para representar visualmente los elementos del software.

Gestionada por casos de uso: Se enfoca en la funcionalidad del sistema desde la perspectiva del usuario.

Fases del Proceso Unificado:

Inicio (Inception): Se establecen los objetivos del proyecto, alcance y principales riesgos.

Elaboración (Elaboration): Se define la arquitectura del sistema y se eliminan riesgos técnicos críticos.

Construcción (Construction): Se desarrollan y prueban los componentes del sistema.

Transición (Transition): Se entrega el producto final y se realizan ajustes según las necesidades del usuario. Ejemplo: Sistema de Reservas de Consultas Médicas

Un pequeño sistema para que los pacientes reserven citas médicas en línea.

## 1. Inicio (Inception)

Objetivo: Permitir que los pacientes reserven consultas médicas a través de una app.

Requisitos básicos:

Registro e inicio de sesión de pacientes.

Selección de médico y horario disponible.

Confirmación de cita.

Riesgos: Seguridad en el acceso a datos médicos.

## 2. Elaboración (Elaboration)

Diseño de la arquitectura: Se definen los módulos principales (usuarios, citas, médicos).

Selección de tecnologías: Se elige una base de datos (MySQL) y el backend (Django).

Prototipo inicial: Se crea un formulario simple para la reserva de citas.

## 3. Construcción (Construction)

Iteración 1: Registro e inicio de sesión de pacientes.

Iteración 2: Implementación del calendario de citas.

Iteración 3: Confirmación de citas por correo.

## 4. Transición (Transition)

Se lanza la versión de prueba.

Se recopila feedback de usuarios.

Se realizan ajustes antes del lanzamiento final.