

# Clase 2

Orientación a objetos - Conceptos

# Responsabilidad y Colaboración

## **Responsabilidad**

La responsabilidad es la capacidad/funcionalidad de una clase para hacer una tarea específica.

Debe ser eficiente, acotado en funcionalidades y no interferir con otras clases.

## **Colaboración**

La colaboración es cómo interactúan las clases, se invocan unas a otras mediante funciones/métodos definidas en cada clase.

# Responsabilidad y Colaboración

## Ejemplo

Equipo de trabajo de pintura

- Clase Lijador
- Clase Pintor
- Clase Parchador
- Clase Coordinador

# Responsabilidad y Colaboración

Equipo de trabajo de pintura

- Clase Lijador
- Clase Pintor
- Clase Parchador
- Clase Coordinador

## **coordinador coordinando la pintura de una pared**

pintor.sePuedePintar(pared) → no

pintor.queFalta(pared) → lijar

lijador.sePuedeLijar(pared) → no

lijador.queFalta(pared) → parchar pared

parchador.parchar(pared) → listo

lijador.sePuedeLijar() → si

lijador.lijar(pared) → listo

pintor.sePuedePintar(pared) → si

pintor.pintar(pared) → listo

# Responsabilidad y Colaboración

## Resumen

- Objetos trabajando juntos para lograr objetivos
- Comunicación entre objetos mediante métodos
- Cada clase tiene su propia responsabilidad
- Las clases no deben interferir entre sí
- La colaboración mejora modularidad y reutilización
- La responsabilidad adecuada garantiza eficiencia.

# Servicios

Son componentes (clases) que proporcionan cierta funcionalidad que se pueda necesitar desde distintos puntos de nuestra aplicación tal como acceso a datos, conversión de datos, acceso e integración con servicios externos.

- Componente que proporciona funcionalidad a una aplicación Java.
- Se define como una clase que implementa una interfaz específica.
- Interfaz define los métodos que se pueden llamar.
- La clase implementa los métodos para proporcionar funcionalidad.
- Se utilizan en arquitecturas de microservicios o aplicaciones monolíticas.
- Proporcionan una forma flexible y escalable de funcionalidad.