

Ejemplo de Patrón *Composite*

DigitalHouse >
Coding School

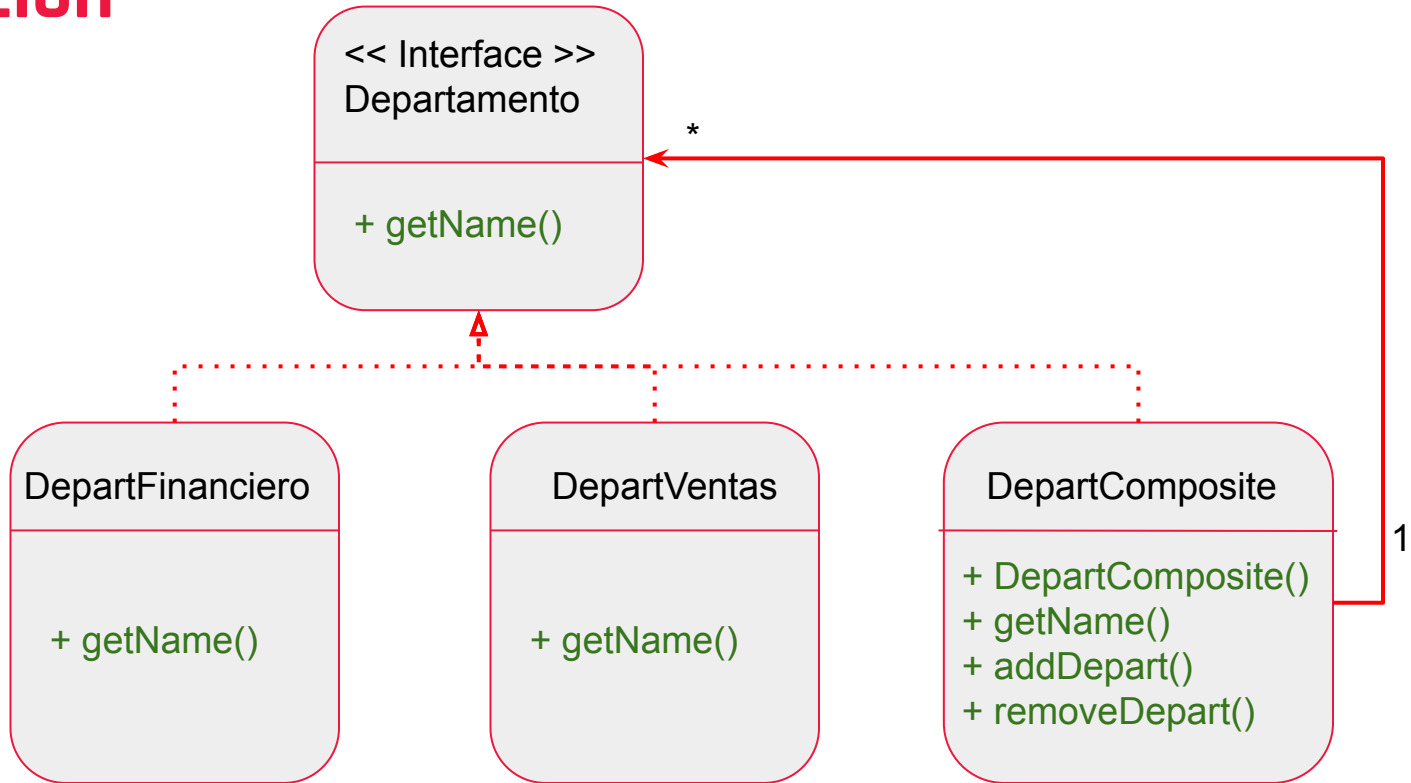


Certified Tech
Developer
The Ultimate Degree

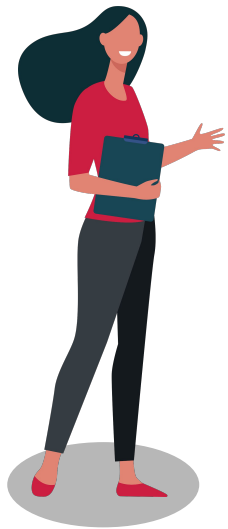
Ejemplo de Patrón *Composite*

Ahora, vamos a imaginar una implementación. Supongamos que deseamos construir una estructura jerárquica de **departamentos** en una empresa. Veamos el diagrama UML a seguir.

Solución



Implementación de las clases del diagrama UML



```
public interface Departamento {  
    void getName();  
}
```

Código de la
interface
Departamento

Implementación de las clases del diagrama UML

Para los componentes *hoja* vamos a definir Clases para los departamentos financiero y de ventas:



```
public class DepartFinanciero implements Departamento
{
    private int id;
    private String name;
    public void getName(){
        System.out.println(getClass().getSimpleName());
    }
}
```

Código de la
clase
DepartFinanciero

Implementación de las clases del diagrama UML

Para los componentes *hoja* vamos a definir clases para los departamentos financiero y de ventas:



```
public class DepartVentas implements Departamento {  
  
    private int id;  
  
    private String name;  
  
    public void getName(){  
  
        System.out.println(getClass().getSimpleName());  
  
    }  
  
}
```

Código de la
clase
DepartVentas

```
public class DepartComposite implements Departamento {
    private int id;
    private String name;

    private List<Departamento> childDepartments;

    public DepartComposite(int id, String name) {
        this.id = id;
        this.name = name;
        this.childDepartments = new ArrayList<>();

    }

    public void getName() {
        childDepartments.forEach(Departamento::getName);
    }

    public void addDepart(Departamento department) {
        childDepartments.add(department);
    }

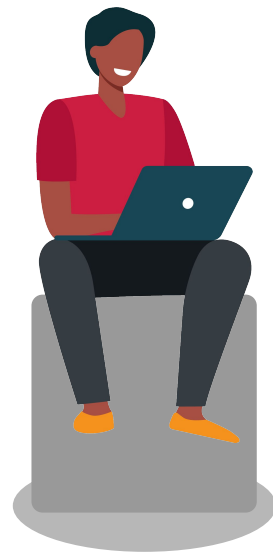
    public void removeDepart(Departamento department) {
        childDepartments.remove(department);
    }
}
```

Código de la
clase
DepartComposite

Ejemplo de Patrón *Composite*

Esta es una **clase compuesta** que contiene una colección de componentes de clase Departamento, y también métodos para adicionar o remover elementos de esta lista.

El método compuesto **getName()** está implementado iterando sobre la lista de elementos hoja e invocando el método apropiado para cada uno.



DigitalHouse>
Coding School