

Carpetas Modelo, Presentación y Persistencia en un Proyecto Java

 Estructura típica dentro de `src/`

Un proyecto Java bien organizado suele dividir su lógica en capas. Dentro de `src/`, es común encontrar esta estructura:

`src/` |— `modelo/` |— `persistencia/` |— `presentacion/`

1. Carpeta `modelo/` (Model)

Propósito

Contiene las clases que representan la lógica de negocio y los objetos del dominio (entidades).

Contenido típico

- Clases como `Usuario`, `Producto`, `Pedido`, etc.
- Atributos, constructores, getters/setters.
- Métodos que encapsulan reglas de negocio.

Ejemplo

```
public class Producto { private int id; private String nombre; private double precio; //  
Constructor, getters y setters }
```

2. Carpeta `persistencia/` (Persistence)

Propósito

Gestiona el acceso a datos: lectura/escritura en bases de datos, archivos, o servicios externos.

Contenido típico

- Clases DAO (Data Access Object) como `ProductoDAO`, `UsuarioDAO`.
- Conexiones JDBC, consultas SQL.
- Interfaces de repositorio si se usa JPA/Hibernate.

Ejemplo

```
public class ProductoDAO { public Producto obtenerPorId(int id) { // Código para consultar la  
base de datos } }
```

3. Carpeta **presentacion/** (Presentation)

Propósito

Contiene la interfaz de usuario y la interacción con el usuario.

Contenido típico

- Clases con interfaces gráficas (Swing, JavaFX) o consola.
- Controladores que reciben eventos del usuario.
- Vistas y formularios.

Ejemplo

```
public class VentanaPrincipal extends JFrame { public VentanaPrincipal() { // Configuración de la interfaz gráfica } }
```

Relación entre carpetas (MVC)

El patrón Modelo-Vista-Controlador (MVC) organiza la interacción entre capas:

presentacion ↓ controlador ↓ modelo ↔ persistencia

- **Presentación:** Interactúa con el usuario.
- **Modelo:** Contiene la lógica de negocio.
- **Persistencia:** Maneja el almacenamiento y recuperación de datos.