



**TECNOLÓGICO
NACIONAL DE MÉXICO®**



INSTITUTO TECNOLÓGICO DE MEXICALI

Carrera: Sistemas computacionales

Materia: Desarrollo de Aplicaciones Móviles

Tema: Modelos de Arquitectura de Software

Alumno: Angel Jair Favila Sanchez

Profesor(a): José Ramon Bogarin Valenzuela

Fecha: 17/02/2025

Modelo Vista-Controlador (MVC)

```

└─ proyecto_mvc
  └─ lib
    └─ controllers
      ├── producto_controller.dart
      └─ usuario_controller.dart
    └─ models
      ├── producto.dart
      └─ usuario.dart
    └─ views
      ├── dashboard_page.dart
      ├── home_page.dart
      ├── login_page.dart
      └─ main.dart
  └─ pubspec.yaml
```

-En carpeta 'controllers' están los controladores que conecta con la vista en donde están los archivos de controlador de usuario y de producto.

-En la carpeta 'models' es donde está la lógica del negocio y están las clases de los usuarios y los productos.

-En la carpeta 'views' está la IU, como la página home y el login

-El main se encuentra en 'lib' y es punto de entrada de la app

MVC separa la **lógica de negocio (Modelo)**, la **interfaz (Vista)** y la **lógica de control (Controlador)** para mantener el código organizado.

Modelo Vista-Presentador (MVP)

```

└─ proyecto_mvp
  └─ lib
    └─ models
      └─ usuario.dart
    └─ presenters
      └─ login_presenter.dart
    └─ views
      ├── home_page.dart
      └─ login_page.dart
  └─ main.dart
  └─ pubspec.yaml
```

-En la carpeta 'models' es donde están los datos y la lógica del negocio, ahí mismo está usuario.dart que representa al usuario en la app.

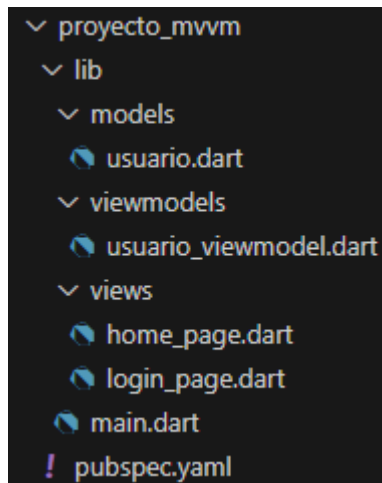
-En carpeta 'presenters' es un medidor entre el modelo y la vista y contiene la lógica del login.

-En la carpeta 'views' está la IU, como la página home y el login.

-El main se encuentra en 'lib' y es punto de entrada de la app.

El **Presentador (P)** maneja la lógica de negocio y le comunica a la **Vista (V)** lo que debe mostrar. Esto **desacopla la UI** de la lógica de negocio.

Modelo Vista-Vista Modelo (MVVM)



-En la carpeta 'models' se definen las clases de datos y con el archivo 'usuario.dart' se representa al usuario.

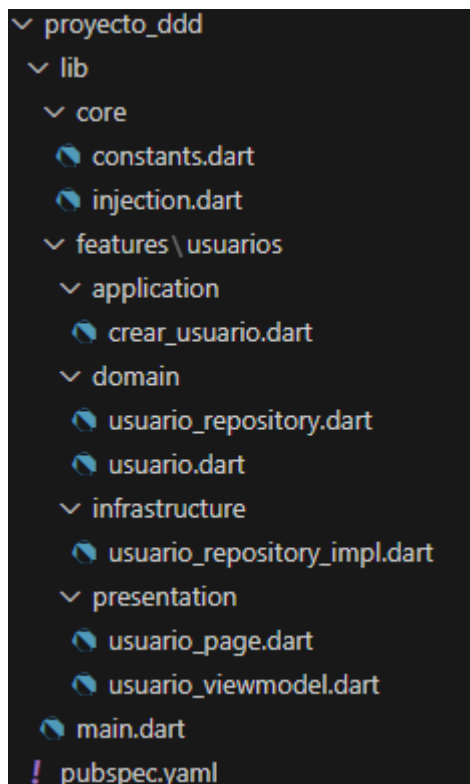
-En carpeta 'viewmodels' se gestiona el estado y la lógica del IU, también se controlan los datos de usuario y notifica cambios.

-En la carpeta 'views' está la IU, como la página principal y la página del login.

-El main se encuentra en 'lib' y es punto de entrada de la app.

MVVM separa la **UI (Vista - V)** de la **lógica de negocio (Modelo - M)** mediante un **ViewModel (VM)** que gestiona el estado y notifica cambios.

Domain-Driven Design (DDD)



-En 'core' están las configuraciones y utilidades generales, como los constantes globales y la inyección de dependencias.

-En 'usuarios' está todo el modelo de dominio de los usuarios.

-En la carpeta 'domain' están las reglas del negocio, junto la entidad del usuario y la interfaz del repositorio.

-En 'application' están los casos de uso o la lógica de la aplicación, como la lógica de crear un usuario.

-En 'presentation' está la IU y los widgets, como la pantalla del usuario y el viewmodel.

-El main se encuentra en 'lib' y es punto de entrada de la app.

Separa claramente la **lógica de negocio, casos de uso, infraestructura y presentación**. **Escalable y mantenible**, ideal para **proyectos grandes**.