

Documentación estructura del proyecto

Víctor García Fernández

December 3, 2023

Introducción

La estructura del proyecto refleja una organización modular y jerárquica que facilita la aplicación de los principios de Clean Architecture y MVVM en la capa `ui`.

Se usan también los paquetes Dagger y Hilt para la inyección de dependencias.

```

composables
  Header.kt
  layouts
    Default.kt
  Loader.kt
core
  navigation
    Navigator.kt
  network
    Retrofit.kt
  security
    HashModule.kt
login
  data
    LoginRepository.kt
  network
    LoginClient.kt
    LoginRequest.kt
    LoginResponse.kt
    LoginService.kt
  domain
    LoginHadErrors.kt
    LoginUseCase.kt
  ui
    LoginScreen.kt
    LoginViewModel.kt
MainActivity.kt
register
  data
    network
      RegisterClient.kt
      RegisterRequest.kt
      RegisterResponse.kt
      RegisterService.kt
      RegisterRepository.kt
    domain
      RegisterUseCase.kt
  ui
    RegisterScreen.kt
    RegisterViewModel.kt
restaurant
  data
    network
      DishResponse.kt
      RestaurantClient.kt
      RestaurantDishRelationResponse.kt
      RestaurantResponse.kt
      RestaurantService.kt
      RestaurantRepository.kt
    domain
      Dish.kt
      GetDishUseCase.kt
      GetRestaurantsWithDishesUseCase.kt
      Restaurant.kt
  ui
    composables

```

```
        DishCard.kt
    DishScreen.kt
    DishViewModel.kt
    RestaurantScreen.kt
    RestaurantViewModel.kt
ui
    theme
        Color.kt
        Theme.kt
        Type.kt
```

Clean Architecture

Composables

En la carpeta **composables**, se encuentran componentes de interfaz de usuario reutilizables.

Core

core es para helpers y módulos de Hilt.

Login

En el módulo de inicio de sesión, **data** maneja el acceso a los datos, **domain** encapsula las reglas de negocio y **ui** se encarga de la presentación.

Register

El módulo de registro sigue la misma estructura que el de inicio de sesión.

Restaurant

En el módulo de restaurantes sigo la misma estructura. Esta vez tengo 2 Screen ya que planteo Dish como un hijo de un Restaurant.

MVVM

ViewModels

Mis ViewModels están ubicados en la capa UI de cada funcionalidad.

El que interactúa con los casos de uso siempre es el ViewModel.

Screens

Cada funcionalidad tiene Composables con el nombre Screen.

Estas son las diferentes vistas de la aplicación.