



FIUBA – 75.52

Taller de programación II

Trabajo práctico grupal - “Jobify”

Documentación Técnica

2do cuatrimestre, 2016

Alumnos:

| Nombre | Padrón | Mail |
|--------------------------|--------|--------------------------|
| Levinas, Alejandro | 95719 | levinasale@gmail.com |
| Moccagatta, Nicolas | 94914 | nicomoccagatta@gmail.com |
| Palma, Leandro | 95841 | leandropalma0@gmail.com |
| Costamagna, Juan Ignacio | 95931 | costamagna.jic@gmail.com |

Arquitectura

La elección de distribuir la plataforma en un cliente implementado en android facilitó la adopción de un patrón que abstraiga la representación de los objetos en el ambiente remoto.

El patrón de arquitectura usado es Model-View-Presenter (MVP). Dicho patrón permite abstraer la complejidad de visualización, así como también la representación de los datos remotos.

Pero principalmente el componente Presenter del patrón permite abstraer la complejidad de la conexión con el servidor remoto y la asociación de los datos con la vista.

Finalmente, la implementación de la conexión y ruteo a los servicios remotos se encapsuló en una clase responsable de la implementación de los servicios del modelo de datos. Análogamente otro componente encapsula la complejidad de la autenticación de los usuarios.

La inspiración para la segregación de los componentes provino de la descripción de Clean Architecture relevada en [1].

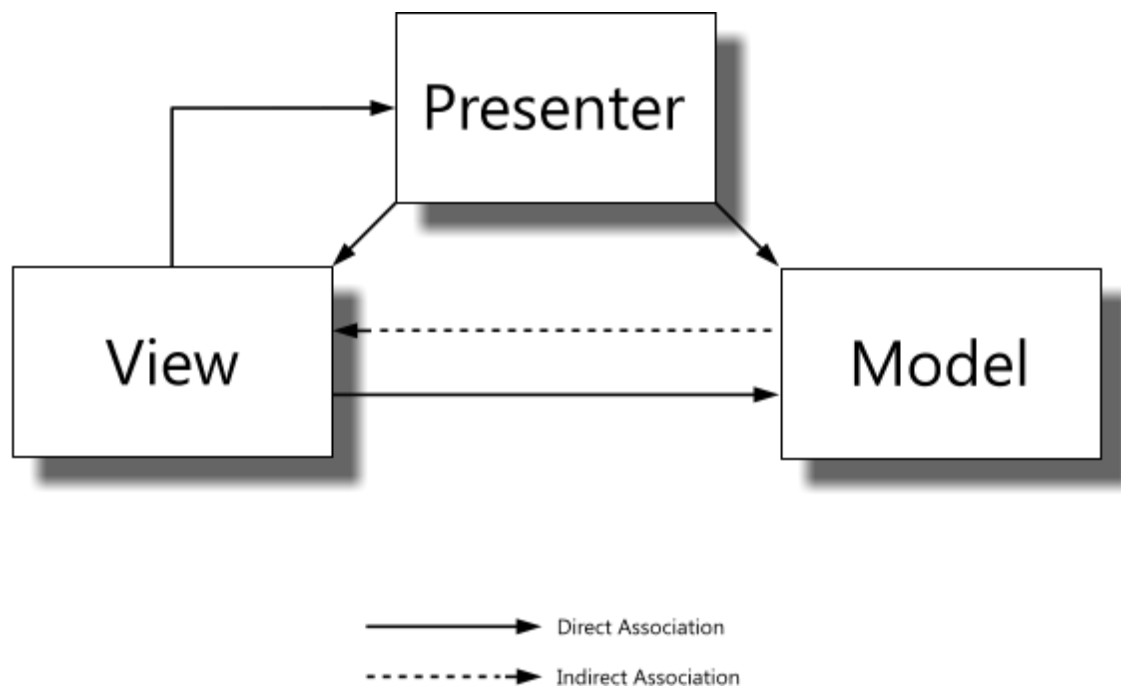


Figura 1: Representación del modelo MVP

¹ <https://github.com/ezeperez26/clean-architecture>