

Héctor Aristimuño

has.as.182@gmail.com

Capas y Responsabilidad de las clases

Capas de la aplicación:

- **Presentación:**
 - **MainActivity**: se encarga de inicializar y mostrar el menú de opciones.
 - **ListFragment**: clase que se encarga de mostrar el listado de películas por categoría.
 - **ContentFragment**: esta clase recibe un objeto Movie y se encarga de mostrar el detalle de la película.
- **Persistencia:**
 - **CacheStorageImp**: permite guardar, modificar, eliminar y obtener el listado de película de la aplicación.
 - **MoviesAdapter**: recibe la data correspondiente a las películas y construye el listado a mostrarse en la aplicación.
- **Dominio o negocio:**
 - **Movie**: clase que contiene todos los atributos de una película.
 - **ApiService**: realiza las solicitudes a los servicios web de The Movie DB.
 - **MoviesResponse**: recibe los resultados de las peticiones realizadas a los servicios web.
- **Red:**
 - **ApiClient**: abre la conexión a los servicios la página The Movie DB.

Principio de responsabilidad única y su propósito

La responsabilidad única es un principio de programación que indica que una clase, método debe tener una única función; esto con la finalidad de que esta clase o método tenga un único motivo para ser modificada, mejorando así la mantenibilidad de la misma.

Características de un código limpio

En mi opinión un código limpio debe ser fácil de entender, fácil de mantener y fácil de extender. Son un conjunto de principios que permiten al desarrollador organizar toda la codificación realizada de tal manera que las aplicaciones sean mantenibles con el paso del tiempo. Otros de los puntos que pretende el *clean code* es que otros desarrolladores puedan extender un programa realizado por otra persona.

Algunos de los consejos más relevantes que se tienen para cumplir con el objetivo del código limpio son los siguientes:

- **Nombres adecuados de métodos o variables:** ayudan a relacionarse con el funcionamiento del código.
- **Comentarios oportunos:** deben usarse cuando se necesita explicar el funcionamiento de un método, variable o módulo, pero no para darle sentido a nombres de funciones o variables los cuales deben ser autoexplicativos.
- **Formato:** la indentación ayuda a tener una mejor perspectiva de código y permite una mayor comprensión del mismo.