

## *Arquitectura y Diseño de Sistemas - PARCIAL*

Duración estimada: 60 minutos

### *Contexto*

Se entrega un proyecto Android escrito en Kotlin con una estructura de carpetas intencionalmente ambigua (A, B, C). La aplicación muestra vuelos sobre territorio nacional (utilizando OpenSky API). El proyecto viola explícitamente tres principios de SOLID y no sigue las prácticas de Clean Code. Está basado en un patrón arquitectónico a identificar (MVC o MVP).

*Consigna.* Sobre el proyecto provisto, realizar las siguientes tareas:

1. Identificación del patrón de arquitectura
  - Analizar el flujo de control y dependencias del proyecto.
  - Determinar si el proyecto sigue el patrón MVP o MVC.
  - Justificar la elección en base a responsabilidades, controladores/presentadores, y flujo de datos.
2. Refactoring
  - Renombrar las carpetas A, B, y C según el rol real de sus componentes.
  - Detallar las violaciones de SOLID .
  - Aplicar principios de Clean Code.
3. Testear el Presenter/Controller.
4. Crear en la raíz del proyecto un archivo **[Apellido][Nombre][LU].txt** con cualquier aclaración que se quiera hacer sobre la resolución de los ejercicios.
5. Entregar un zip del proyecto, renombrar el archivo con el formato **[Apellido][Nombre][LU].zip** y notificar al integrante de la cátedra presente.