

## *Simulación de Parcial - Arquitectura y Diseño de Sistemas*

Duración estimada: 2 horas

### *Contexto*

Se entrega un proyecto Android escrito en Kotlin con una estructura de carpetas intencionalmente ambigua (A, B, C). El proyecto muestra el precio de criptomonedas (ej: Bitcoin), accediendo a diferentes APIs externas según una configuración en tiempo de ejecución. 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.
  - Refactorizar el código corrigiendo estas violaciones de SOLID.
  - Aplicar principios de Clean Code.
3. Implementar una prueba unitaria que valide correctamente funcionalidades **de la vista.**

de otras clases.