



## Enunciado proyecto Android para postulantes PDS

### Objetivo del proyecto

Medir las capacidades analíticas de los postulantes al puesto Developer Android (Jr, SSr. o Sr.) dentro del marco de evaluaciones del proceso de selección.

### Pautas del trabajo

- El trabajo debe realizarse de forma remota por el postulante, con un tiempo máximo aproximado de 120 minutos.
- El trabajo consta en desarrollar una aplicación sencilla, con algunos features básicos que nos permitirán medir los skills técnicos del postulante.
- La aplicación debe desarrollarse en una computadora, que no será provista por PDS, utilizando Android Studio 4.x. Si el postulante desea hacer la aplicación en Kotlin, deberá tener el plugin correspondiente configurado previo a iniciar el examen.
- El comienzo de la evaluación queda iniciada al recibir este documento.
- Posterior a los 120 minutos de examen, o a la finalización del mismo (lo que ocurra primero), el postulante deberá ponerse en contacto con el evaluador que le entregó el link del examen, para poder defenderlo en una etapa de coloquio.
- Durante la etapa de coloquio, el postulante queda sujeto a defender el trabajo realizado, con posibilidad de repreguntas por parte de los evaluadores.



## Enunciado del trabajo

Se desea realizar una aplicación mobile (kotlin), partiendo de cero con algunos de los template base que ofrece Android Studio al iniciar un nuevo proyecto.

La aplicación debe contar con 3 features:

1. En la pantalla principal, debe haber un botón a la izquierda (centrado verticalmente) del Activity que permita calcular el factorial de un número. Ese número debe obtenerse de un EditText situado en la parte superior ocupando todo el layout.

El factorial de un número se define como la multiplicación de sí mismo por el número anterior, repitiendo este proceso hasta que el número anterior es 1. Por ejemplo el factorial de 3 es  $\rightarrow (3!) = 3 \times 2 \times 1 = 6$ . El de 5 es  $\rightarrow (5!) = 5 \times 4 \times 3 \times 2 \times 1 = 120$ .

2. En la misma pantalla, debe haber un botón a la derecha (centrado verticalmente) del Activity, que debe consumir una API pública (sugerimos <http://worldclockapi.com/api/json/est/now>). El resultado se debe mostrar por un **Toast**, pero el consumo debe hacerse por una **Coroutine** (Kotlin). Puede utilizar cualquier tecnología para el consumo de API (**Retrofit2**, Volley, o alguna otra que conozca).
3. En la parte inferior, ocupando todo el layout horizontal debe haber un **RecyclerView** que consuma una capa Dao conectado a **Room**, en donde se almacenarán los eventos de **Lifecycle** de la aplicación. Es decir, cada vez que la aplicación ejecute uno de los métodos propios del lifecycle de un Activity, deberá llenar una tabla vía DAO, que será gestionada por Room. Una vez insertados los eventos, deberán mostrarse en pantalla en el correspondiente RecyclerView en conjunto con el timestamp del evento.

El activity debe diseñarse con **ConstraintLayout**, con guías horizontales y verticales de modo de asegurar un "padding" de 5px desde todos los bordes de la pantalla

Cualquier otro detalle no definido en las líneas anteriores queda a discreción del postulante.