

## Final: Presentación

### Consignas para la entrega del final:

- La temática es de libre elección
- Debe realizarse en equipos de 2 integrantes
- Para la entrega del presente final, se deberá [crear un repositorio en Github](#).  
**Opcionalmente se puede usar el mismo repositorio ya usado (y es la opción recomendable).** El mismo deberá ser nombrado, respetando el siguiente formato para asignarle el nombre de repositorio:
  - **final-am-ac**[m|t|n]4[a|b|c|d]-[apellido\_alumn@ 1]-[apellido\_alumn@ 2]
    - Los “[”, “[” y “[” no deben permanecer, son a modo de separación
  - TODO EN MINÚSCULAS Y UTILIZANDO EL GUIÓN MEDIO COMO SEPARADOR.
  - LOS APELLIDOS DEBEN ESTAR EN ORDEN ALFABÉTICO.
  - Ejemplos:
    - final-am-acn4a-hernandez-perez
  - Se debe [agregar en el repositorio al usuario sergiomedinaio](#).
  - El [repositorio debe ser público](#)
  - ***En caso de incumplir con el formato del nombre, se restará automáticamente dos puntos, pudiendo desaprobado por este error el parcial.***
- La entrega será mediante **DvPanel**, justo a un documento incluyendo:
  - Dato del estudiante
  - Datos de cursada
  - Enlace hacia el repositorio
  - Enlace al informe
- **Fecha de entrega:**
  - Debe cargarse el documento por lo menos 24hs antes de la instancia de evaluación.

## **Desarrollo mínimo y obligatorio, nota 4**

Cualquier faltante de los requisitos que se presentan a continuación, hará que la entrega se considere desaprobada, aunque haya hecho cosas no solicitadas o del punto siguiente. Esta condición es mínima e innegociable.

- ☐ Debe contener varias pantallas, las cuales permiten una navegación entre las distintas características de la aplicación.
- ☐ Deberán contar con un informe que describa por cada pantalla a realizar con las funcionalidades esperadas, desarrollando el flujo de uso de la misma. La elección de la herramienta para el desarrollo del mock es libre.
- ☐ Debe implementar el uso de:
  - ConstraintLayouts
  - LinearLayouts (vertical / horizontal)
  - Uso de Button
  - Uso de TextView
  - Los contenidos de la aplicación deben ser REALES y estar acompañados correctamente por imágenes para enriquecer el producto.
- ☐ Tiene que hacer uso de eventos.
- ☐ Debe tener **comportamiento dinámico**:
  - al interactuar con los elementos responda modificando algo en relación al objetivo de la aplicación (**por ejemplo**: al hacer click en un botón que sea agregar X cosa, es incluya en una ScrollView, al agregar varios se puedan visualizar cada una).
  - Pueden guardarse los datos modificados en variables (**por ejemplo**: si se agregan/quitan elementos de una lista, se actualiza la variable para poder seguir viéndolo en la Activity).
- ☐ Pasaje de datos entre activities
- ☐ Implementación de firebase auth (al menos login)
- ☐ Implementación de firebase firestore trayendo al menos un documento (por ejemplo, los datos personales del usuario).

(sigue en la siguiente hoja)

## **Agregados para llegar a 10**

Entre las entregas que cumplan el punto anterior, incrementarán su nota los desarrollos que:

- ☐ Implementación de firebase auth (**login/registro**, verificación de email opcional)
- ☐ Implementación de firebase firestore (trayendo al menos una collection de documentos)
- ☐ Se valorará la elección de colores/diseños.
- ☐ Se evaluará la usabilidad de la aplicación.
- ☐ Uso apropiado de clases
- ☐ Descarga de imagenes desde url
- ☐ Agregado de contenido desde url (por medio de alguna biblioteca)
- ☐ El usuario deberá ser notificado en las operaciones (por medio de Toast)
- ☐ Deben organizarse apropiadamente:
  - ☐ Variables de String
  - ☐ Variables de Dimensiones
  - ☐ Variables de Colores
- ☐ Debe haber interacciones que vinculen datos en la base de datos (por ejemplo, que se pueda actualizar la información del usuario, se pueda agregar datos a la base de datos, etc)
- ☐ Uso de algún tipo de convención para los commit empleados. Sugerido: [conventional commits](#)
- ☐ Uso de extras para el envío de paquetes

## **Opcional (no resta para llegar al 10):**

- ☐ Implementación de algún [patrón de arquitectura](#) para organizar la lógica de la app
- ☐ Uso básico de [Google maps](#)
- ☐ Implementación de Base de Datos local

(sigue en la siguiente hoja)

### Instancia de final:

- Aun cuando se pueda haber desarrollado en equipos, cada integrante debe estar en la mesa evaluadora, pudiendo presentarse individualmente.
- Restricciones de entrega:
  - Los cambios deben ser aplicados directamente sobre el repositorio
  - Debe haber cambios significativos en github hecho por quien presenta el final.
- El envío de la resolución del final es condición necesaria, pero no suficiente para aprobación, pudiendo desaprobar en caso de (por ejemplo):
  - No presentarse a la mesa evaluadora
  - No alcanzar el **Desarrollo mínimo y obligatorio** esperado en la entrega
  - No poder responder preguntas sobre la autoría y desarrollo de la aplicación.
- Se realizará un coloquio donde se presentará lo desarrollado.
  - Se podrá exigir cámara y micrófono encendido
  - Se podrá consultar sobre el desarrollo en su totalidad
- Se recomienda el uso apropiado de github, al resultar beneficioso para versionar los avances. El uso de commit bien distribuidos en tareas y tiempo permiten evidenciar el proceso de desarrollo y sirviendo en gran medida para evacuar dudas razonables sobre la autoría y avance de la aplicación.