

Parcial 2: Presentación

Consignas para la entrega del parcial:

- La temática es de libre elección
- Debe realizarse en equipos de 2 integrantes
- Para la entrega del presente parcial, 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:

- **parcial-2-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:
 - parcial-2-am-acn4a-hernandez-perez
- Se debe agregar en el repositorio al usuario **sergiomedinaio**.
- ***En caso de incumplir con el formato del nombre, se restará automáticamente dos puntos, pudiendo desaprobar por este error el parcial.***
- La entrega será mediante **Campus**, en la actividad correspondiente que genere el docente, incluyendo:
 - Enlace hacia el repositorio
- **Fecha de entrega:**
 - La fecha de entrega será la que indique el docente en la actividad del **Campus**.

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.
- Deberán realizar un informe que describa **la pantalla** a realizar. 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.
- Al menos un elemento tiene que tener aplicado un evento
- La aplicación debe tener comportamiento dinámico, es decir, que 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.
- Los cambios deben ser aplicados directamente sobre el repositorio, no se admiten entregas con 1 o dos commits.
- Los cambios deben ser realizados por ambos miembros del equipo, y visualizarse en el github.

Agregados para llegar a 10

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

- Se valorará la elección de colores/diseño, y ajustes respecto a la pre-entrega anterior.
- Deben organizarse apropiadamente:
 - Variables de String
 - Variables de Dimensiones
 - Variables de Colores
- Mientras más interacciones/comportamientos se incluya, mayor será la valoración.
- Uso de algún tipo de convención para los commit empleados. Sugerido: [conventional commits](#)
- Pasaje de datos entre activities
- Uso de extras para el envío de paquetes
- Descarga de imagenes desde url
- Agregado de contenido desde url (por medio de alguna biblioteca)
- Implementación de firebase auth y/o firestore