

Requisitos de entrega del prototipo en Flutter

- ❖ Se tendrá tiempo hasta el **Domingo 1 de Diciembre para la entrega.** (Puede entregarse antes)
- ❖ El código deberá ser subido a GitHub, de manera pública o privada y estar correctamente documentado a través del archivo README.md. En caso de publicarse de manera privada deberá ser compartido con el docente a través de la casilla: sganan81@gmail.com
- ❖ NO es necesario que el prototipo realice peticiones http. Puede mostrarse información estática o incluir conjuntos de valores en el código (List, Map, etc.) para luego ser consultados o iterados. Pueden emplearse mocks (Ver repo base)
- ❖ El proyecto debe ser realizado con los mismos integrantes del grupo del TP1. (Con excepción de aquellos que no continuaron en la cursada)
- ❖ La aplicación debe contar con las siguientes pantallas básicas (Global).
 1. Screen Home
 2. Perfil del usuario con opción para el cambio de Tema (Dark Light)
 3. Drawer Menu
- ❖ Pantallas o Widgets individuales por cada integrante del grupo
 1. Screen Lista de registros (/api/v1/listado)
 2. Screen Visualización individual de un registro (/api/v1/listado/30)
 - Generar un Formulario que contemple:
 - TextFormField
 - Switch/Checkbox
 - Imagen principal (Solo si su api lo provee)
 3. Widget reutilizable (Ej: Card Basic, Card Image, Card Swiper, clase privada, etc.)
 - Contemplar el envío de parámetros al constructor requeridos y opcionales.

❖ Widgets que se recomiendan contemplar:

- Row, Column, Container, Textbox, Sizebox, IconButton, etc.
- StatelessWidget y StatefulWidget
 - Contemplar alguna pantalla que utilice el setState o Provider
- Scaffold (pueden utilizarse diferentes modelos de navegación. EJ: drawer, bottomNavigationBar, actions con IconButton's en AppBar, FloatingActionButton, etc.)
- ListView. (En alguna de sus variantes. ListView.builder, ListView.separated o crear una lista personalizada)
- Imágenes. Pueden utilizarse diferentes widgets para mostrarlas
 - FadeInImage, "BoxDecoration -> DecorationImage", Image.network, etc.

❖ Estructura

- Assets. Incluir desde pubspec.yaml recursos estáticos (Imágenes, gif, fuentes. etc.)
- Routes (Rutas): Dentro del archivo main deben configurarse las rutas para navegar y el initialRoute y también pueden contemplarse en el drawer o cualquier estructura de navegación que utilicen.
- Theme (Temas). Los estilos de la aplicación deben separarse del archivo main y colocarse en un archivo y carpeta separada.
- Utilizar archivos unificadores de exportación (Barrel files) dentro la carpeta donde almacenen sus pantallas (/screens/screen.dart, widgets/widgets.dart)
- Pueden utilizar paquetes externos para cubrir algunas funcionalidades.
- Material a compartir con el docente:
 - Repositorio en GitHub
 - N° de API de Android o S.O y Dimensiones del dispositivo con las que fue probado el prototipo a entregar. (Solo para desarrollos probados mediante virtualización con Android Studio)