**[ACTIVIDAD 2: APLICACIÓN APP MULTIPLATAFORMA](https://learn.universidadviu.com/webapps/assignment/uploadAssignment?content_id=_8272519_1&course_id=_191308_1&group_id=&mode=view)**

En esta segunda actividad se pretende desarrollar una pequeña aplicación multiplataforma usando el **FRAMEWORK REACT NATIVE** **o FLUTTER** en la que se apliquen los conocimientos teóricos y prácticos trabajados en esta primera parte. De forma paralela, al ser una actividad de elaboración en grupo se trabajará con herramientas y metodologías que favorezcan la gestión de las tareas y el seguimiento del desarrollo realizado.

La temática será la creación de una aplicación de presentación de eventos que se produzcan en una ciudad, pueblo o zona que conozcáis.

**OPCIÓN 1: REQUISITOS DE DISEÑO FLUTTER**

* El lenguaje base de desarrollo será framework FLUTTER
* Se programarán al menos 3 pantallas/intefaces
* En cada una de las 3 pantallas se usará los Widgets necesarios para presentar la información. Al menos deben aparecer:
  + Scaffold
  + AppBar
  + Text
  + ElevatedButton

TextField

DropdownButton

* Se utilizará el sistema Navigator para la navegación
* Se utilizará shared preferences para almacenar datos dentro de la app
* La información se recogerá o bien de un ApiRestFul abierto o bien creándolo con json\_server
* Una de las pantallas usadas deberá tener un pequeño formulario
* Al menos dos de las pantallas intercambiarán información para presentar, calcular o cambiar el estado de algún widget

**OPCIÓN 2: REQUISITOS DE DISEÑO REACT NATIVE**

* El lenguaje base de desarrollo será framework REACT NATIVE
* Se programarán al menos 3 pantallas/intefaces
* En cada una de las 3 pantallas se usará:
  + Estilos de diseño independientes utilizando el componente  StyleSheet  
    - Al menos 3 bloques de estilos
    - Flex para definir la colocación, posición y tamaño de cada elemento
* Se deberán usar los siguientes componentes como mínimo
  + View
  + SafeAreaView

Text

Image

TouchableOpacity O Pressable

ScrollView

FlatList

* Se construirá un sistema de navegación utilizando las librerías React Navigation visto en clase y los videos facilitados, usando al menos dos de los Navigators existentes
* La información que se presente en las Vistas puede ser creada y usada dentro del código. Como reto se puede acceder a esta información de forma externa, bien a través de API de terceros o un APIRestFul creado por el grupo
* Una de las pantallas usadas deberá tener un formulario en el que se permitiría generar un nuevo elemento en esos datos
* Al menos dos de las pantallas intercambiarán información para presentar, calcular o cambiar el estado de algún componente

**REQUISITOS DE CALIDAD**

* El nombre de las variables será significativo, tanto en los identificadores de componentes como en los objetos y variables, intentando utilizar nopmbrado camelCase.
* El nombre de los Componentes/Widgets será significativo y seguirá las pautas marcadas en los ejemplos de la asignatura
* Se introducirá comentarios significativos, en aquellas funciones o clases para describir su funcionamiento
* Se modularizará y crearán funciones para hacer más legible y reutilizable el código

**REQUISITOS DE TRABAJO EN EQUIPO**

* Se utilizará git y se almacenará el desarrollo en un repositorio remoto, realizando commits individales por cada miembro

**REQUISITOS DEL ENTREGABLE**

* El entregable consistirá en un único documento en PDF que contendrá los siguientes elementos
  + Portada con título de la asignatura, título de la aplicación, miembros y enlace al git de la aplicación (1 página)
  + Apartado con breve explicación de la aplicación
    - Bien a través de capturas ( máximo 2 páginas).
    - Bien a través de enlace a un vídeo (máximo 2 minutos)
* El repositorio será compartido con el usuario **pgomezarnalVIU**

Deseo crear esta aplicación

**Tema: App de Eventos Culturales de tu Ciudad**

**Pantallas recomendadas:**

1. **Pantalla principal (Home)**:
   * Lista de eventos usando FlatList
   * Imágenes de portada para cada evento
   * Opción de filtrado por categoría (usando ScrollView horizontal)
   * Cada evento sería un elemento tocable (TouchableOpacity)
2. **Pantalla de detalles del evento**:
   * Imagen grande del evento
   * Información detallada
   * Mapa o dirección
   * Botón para marcar como favorito o compartir
   * ScrollView para toda la información
3. **Pantalla de formulario**:
   * Formulario para añadir un nuevo evento
   * Campos para título, fecha, ubicación, descripción, categoría, etc.
   * Validación de campos
   * Botón para enviar

Esta estructura te permitirá implementar todos los componentes requeridos (View, SafeAreaView, Text, Image, TouchableOpacity, ScrollView, FlatList) de manera natural y útil.

La navegación entre pantallas sería intuitiva: de la lista de eventos al detalle, y un botón para ir al formulario. Podrías usar tanto Stack Navigator como Tab Navigator para cumplir con el requisito de usar al menos dos tipos de navegadores.

Los datos iniciales podrían ser creados dentro del código como sugeriste, y la funcionalidad del formulario permitiría añadir nuevos eventos a esta lista durante la ejecución de la app.

**REQUISISTOS**

* El lenguaje base de desarrollo será framework REACT NATIVE
* Se programarán al menos 3 pantallas/intefaces
* En cada una de las 3 pantallas se usará:
  + Estilos de diseño independientes utilizando el componente  StyleSheet
    - Al menos 3 bloques de estilos
    - Flex para definir la colocación, posición y tamaño de cada elemento
* Se deberán usar los siguientes componentes como mínimo
  + View
  + SafeAreaView

Text

Image

TouchableOpacity O Pressable

ScrollView

FlatList

* Se construirá un sistema de navegación utilizando las librerías React Navigation visto en clase y los videos facilitados, usando al menos dos de los Navigators existentes
* La información que se presente en las Vistas puede ser creada y usada dentro del código.
* Una de las pantallas usadas deberá tener un formulario en el que se permitiría generar un nuevo elemento en esos datos
* Al menos dos de las pantallas intercambiarán información para presentar, calcular o cambiar el estado de algún componente

**REQUISITOS DE CALIDAD**

* El nombre de las variables será significativo, tanto en los identificadores de componentes como en los objetos y variables, intentando utilizar nopmbrado camelCase.
* El nombre de los Componentes/Widgets será significativo y seguirá las pautas marcadas en los ejemplos de la asignatura
* Se introducirá comentarios significativos, en aquellas funciones o clases para describir su funcionamiento
* Se modularizará y crearán funciones para hacer más legible y reutilizable el código