

## TA 1

Objetivo: Crear un contador que aumente y disminuya al pulsar botones.

Tareas:

- Añadir dos botones: uno para incrementar y otro para disminuir.
- Mostrar el valor del contador en pantalla.
- Estilizar los botones y el texto.

## TA 2

Objetivo: Capturar y mostrar el texto ingresado por el usuario.

Tareas:

- Usar un TextInput para capturar el texto del usuario.
- Mostrar el texto debajo del campo de entrada.
- Añadir estilos al campo de texto y al resultado mostrado.

## TA 3

Objetivo: Crear una lista de tareas donde se pueda añadir y eliminar elementos.

Tareas:

- Usar un useState para manejar la lista de tareas.
- Añadir un TextInput para escribir nuevas tareas y un botón para añadirlas.
- Mostrar las tareas en una lista debajo del campo de entrada.
- Agregar un botón junto a cada tarea para eliminarla.

## TA 4

Objetivo: Cambiar una imagen al pulsar un botón.

Tareas:

- Usar el componente Image de React Native.
- Configurar un botón que cambie la imagen mostrada al pulsarlo.

## TA 5

Objetivo: Crear una galería de imágenes con desplazamiento vertical.

Tareas:

- Usar un FlatList o ScrollView para mostrar una lista de imágenes.
- Cada imagen debe tener una descripción debajo.
- Agregar un borde y un espacio entre las imágenes.

## TA 6

Objetivo: Implementar una lista de elementos con la opción de eliminar mediante un gesto de swipe (hacer una copia de lo hecho en la TA 3)

Tareas:

- Usar un componente como ReanimatedSwipeable ([https://docs.swmansion.com/react-native-gesture-handler/docs/components/reanimated\\_swipeable](https://docs.swmansion.com/react-native-gesture-handler/docs/components/reanimated_swipeable)) de React Native Gesture Handler.
- Configurar un gesto de deslizamiento para eliminar un elemento de la lista.
- Almacenar la lista de elementos en un estado.

## TA 7

Objetivo: Crear una aplicación que busque y muestre películas usando una API de películas.

API: Usa la API de <https://www.omdbapi.com/>.

Tareas:

- Crear un input para que el usuario ingrese el nombre de una película.
- Realizar una solicitud GET a la API de OMDb usando el nombre de la película.
- Mostrar el póster, el título y la sinopsis de la película encontrada.
- Mostrar un mensaje de error si no se encuentra la película.

## TA 8

Objetivo: Crear una aplicación que utilice la API Dimensions para ajustar su contenido dependiendo del tamaño del dispositivo

Tareas:

- Utilizar componentes tales como View, Text, etc., y aplicarles estilos usando Stylesheet
- Dichos estilos deben variar dependiendo del tamaño del dispositivo

## TA 9

Objetivo: Crear una aplicación que utilice la API Platform para ajustar su contenido dependiendo del sistema operativo que se encuentre corriendo el dispositivo

## TA 10

Objetivo: Crear una aplicación que utilice la librería expo-location para mostrar información acerca de la ubicación del dispositivo.

Restricciones:

- La aplicación debe pedir permiso al usuario a la hora de acceder a la localización del dispositivo. En caso de que el usuario no lo otorgue, mostrar en la pantalla un mensaje de error.

## TA 11

Objetivo: Crear una aplicación que utilice la librería expo-image-picker para acceder al álbum de fotos del usuario, subir una a la aplicación, y mostrarla en pantalla.

Restricciones:

- La aplicación debe pedir permiso al usuario a la hora de acceder al álbum del usuario. En caso de que el usuario no lo otorgue, mostrar en la pantalla un mensaje de error.

## TA 12

Objetivo: Crear una aplicación que utilice la librería expo-image-picker para acceder a la cámara del dispositivo, tomar una foto, subirla a la aplicación, y mostrarla en pantalla.

Restricciones:

- La aplicación debe pedir permiso al usuario a la hora de acceder a la cámara del dispositivo. En caso de que el usuario no lo otorgue, mostrar en la pantalla un mensaje de error.

## TA 13

Objetivo: Crear una aplicación que utilice la librería expo-camera para acceder a la cámara del dispositivo, tomar una foto, subirla a la aplicación, y mostrarla en pantalla.

Restricciones:

- La aplicación debe pedir permiso al usuario a la hora de acceder a la cámara del dispositivo. En caso de que el usuario no lo otorgue, mostrar en la pantalla un mensaje de error.