

CÓDIGO



FUNCIONES

ACCESO A LA CÁMARA REACT NATIVE

Agustina Angel y Micaela Cembal







X

NUESTRO PROYECTO

Realizar una app que nos permita para sacar una Foto y darnos la opción de mostrarla en un Image o almacenarla en la memoria interna.



CÁMARA



CREACIÓN DE UN PROYECTO







PASO 1

npx create-react-app camera



PASO 2

npm i expo-cli -g



PASO 3

npm i





LIBRERÍAS





01

EXPO Media Library

expo install expo-media-library



EXPO Camera

expo install expo-camera



03

React Navigation Native

npm install @react-navigation/native



React Navigation Stack

npm install @react-navigation/stack



APP. JS

X

Al ser App.js el componente padre de nuestra aplicación, armamos la navegación entre los componentes hijos:











APP. JS



```
export default function App() {
 const [imageGaleria, setImageGaleria] = useState("");
 return (
   <NavigationContainer>
     <Stack.Navigator initialRouteName="Home">
       <Stack.Screen name="Home">
         {(props) => (
           <HomeScreen {...props} setImageGaleria={setImageGaleria} />
       </Stack.Screen>
                                 <Stack.Screen name="Camera" component={CameraScreen} />
                                 <Stack.Screen name="ImageScreen">
                                    {(props) => <ImageScreen {...props} imageUri={imageGaleria} />}
                                 </Stack.Screen>
                                </Stack.Navigator>
                             </NavigationContainer>
```



HOME SCREEN







```
return (
  <View style={appStyles.container}>
    <Text style={appStyles.text}>;Bienvenido a la aplicación de cámara!</Text>
    <Button
     title="Abrir Cámara"
      onPress={() => navigation.navigate("Camera")}
    < Button
     title="Seleccionar Imagen"
      onPress={selectImage}
  </View>
```

HOME SCREEN



Seleccionar imagen

```
const selectImage = async () => {
  const { status } = await ImagePicker.requestMediaLibraryPermissionsAsync();
  if (status !== "granted") {
    console.log("Permisos insuficientes para acceder a la galería de imágenes.");
   return;
 try {
   const result = await ImagePicker.launchImageLibraryAsync({
      mediaTypes: ImagePicker.MediaTypeOptions.Images,
      allowsEditing: true,
      aspect: [1, 1],
      quality: 1,
   });
```



HOME SCREEN







```
console.log(result);
const path = result.assets[0].uri;
setImageGaleria(path);
console.log("Imagen seleccionada:", path);
navigation.navigate("ImageScreen", { imageUri: path });
```





X

HOME SCREEN

Seleccionar imagen



```
if (result.cancelled) {
  console.log("El usuario canceló la selección");
```



Permisos

```
useEffect(() => {
    (async () => {
        MediaLibrary.requestPermissionsAsync();
        const cameraStatus = await Camera.requestCameraPermissionsAsync();
        setHasCameraPermission(cameraStatus.status === "granted");
    })();
}, []);
```













Renderiza dependiendo si no hay una foto tomada

 Si no hay imagen, se renderiza el componente cámara que nos permite capturar fotos proveniente de: import { Camera } from "expo-camera";



Si hay una imagen, se renderiza el componente image de react Native, que muestra la imagen sacada.



</Camera>







```
{!image ? (
  <Camera
    style={appStyles.camera}
   type={type}
    ref={cameraRef}
    flashMode={flash}
    <View
      style={{
        flexDirection: "row",
        justifyContent: "space-between",
        paddingHorizontal: 30,
      }}
```

```
<Button
   title=""
   icon="retweet"
   onPress={() => {
      setType(
        type === Camera.Constants.Type.back
          ? Camera.Constants.Type.front
          : Camera.Constants.Type.back
      );
</View>
```

X

Renderiza la imagen si hay una foto tomada

```
) : (
    <Image source={{ uri: image }} style={appStyles.camera} />
)}
```















Funciones takePicture y savePicture

```
const takePicture = async () => {
 if (cameraRef) {
   try {
      const data = await cameraRef.current.takePictureAsync();
      console.log(data);
      setImage(data.uri);
    } catch (error) {
      console.log(error);
```



```
const savePicture = async () => {
  if (image) {
   try {
     const asset = await MediaLibrary.createAssetAsync(image);
      alert(";Foto guardada! [6");
      setImage(null);
      console.log("saved successfully");
    } catch (error) {
      console.log(error);
```

IMAGESCREEN





```
const imageUri = route?.params?.imageUri || null;
console.log('Image URI recibida:', imageUri);
return (
 <View style={appStyles.container}>
    {imageUri ? (
        <Image source={{ uri: imageUri }} style={appStyles.image} />
       <TouchableOpacity
         style={appStyles.goBackButton}
         onPress={() => navigation.goBack()}
          <MaterialIcons name="arrow-back" size={30} color="white" />
       </TouchableOpacity>
     </>
    ): (
     <Text>No se ha proporcionado una imagen para mostrar.</Text>
 </View>
```







iMUCHAS GRACIAS!



REACT JS

