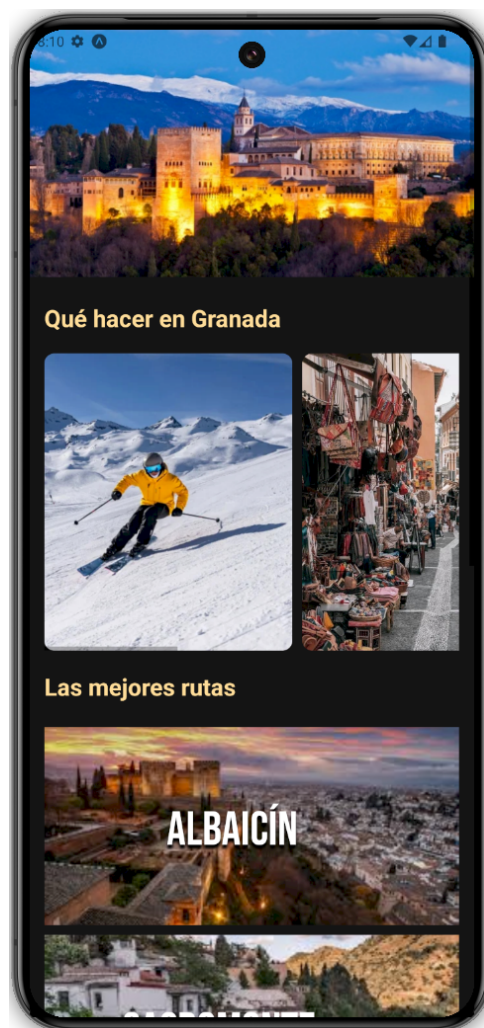


PROYECTO 2

QUÉ HACER EN GRANADA (VERSIÓN COMPONENTES + TypeScript)



- Número de personas: 1
- Tutoriales: 7, 9
- Lenguaje: JavaScript y TypeScript
- Tipo de proyecto: ejercicio guiado con pasos

• **¿Qué hace la aplicación?** Este proyecto consiste en adaptar el proyecto 1 para dividir la interfaz en componentes, facilitando así su estructura y mantenimiento y luego pasarlo a **TypeScript**. El resultado visual será el mismo que ya se tenía, y no se añadirá ninguna característica nueva. Por tanto, este proyecto es una **refactorización** (mejora de la estructura interna) del proyecto anterior.

1. Abre el proyecto 1 con Visual Studio Code y crea una rama llamada **proyecto2**, pasando a trabajar sobre ella.
2. En el proyecto, crea una carpeta llamada **components**
3. En dicha carpeta, crea un archivo llamado **Titulo.tsx** y en él teclea **rnfs+Intro**
4. Mueve el componente **Text** del primer título, junto con el estilo **titulo** al archivo **Titulo.tsx**, así:

```
1. export default function Titulo() {
2.   return (
3.     <Text style={[styles.titulo, { color: tema.colorTitulo }]}>
4.       ¿Qué hacer en Granada?
5.     </Text>
6.   )
7. }
8. const styles = StyleSheet.create({
9.   titulo: {
10.    fontWeight: "bold",
11.    fontSize: 24,
12.    marginVertical: 20,
13.  },
14. })
```

5. Haz que la función **Titulo** reciba estos dos props:
 - **texto**: Es el texto que se verá en el título
 - **tema**: Es el objeto que contiene los colores
6. Reemplaza “¿Qué hacer en Granada?” para que sea el prop **texto**
7. En **App.tsx**, añade un **Titulo** que tenga:
 - **texto**: ¿Qué hacer en Granada?
 - **tema**: El objeto con el tema seleccionado
8. Repite lo anterior con los demás **Text** de los títulos que aparecen en **App.tsx**, reemplazándolos por componentes **Titulo**

☑ **Punto de comprobación:** Comprueba que todos los títulos se ven correctamente

9. En **components** crea un archivo llamado **Carrusel.txt** y en él teclea **rnfs+Intro**
10. Mueve todo el **ScrollView** del carrusel de fotos al archivo **Carrusel.txt**, junto con el estilo **fotoCarrusel**, de la misma forma que hiciste en el punto 4 con el **Text** del título
11. En **App.tsx** inserta un componente **Carrusel**

☑ **Punto de comprobación:** Comprueba que el carrusel de fotos se ve correctamente

12. En **components** crea un archivo llamado **FotoRuta.tsx** y en él teclea **rnfs+Intro**
13. Mueve el primer componente **ImageBackground** junto con el **Text** que contiene y los estilos **fotoRuta** y **textoFoto** al archivo **FotoRuta.tsx**, de la misma forma que hiciste en el punto 4 con el primer título

14. Añade a la función **FotoRuta** estos dos props:
 - **texto:** Es el texto que se verá encima de la imagen de fondo
 - **imagen:** Es la imagen que se mostrará de fondo
 - **tema:** Es el objeto que contiene los colores
15. En **FotoRuta.tsx**, modifica los componentes **ImageBackground** y **Text** para que hagan uso de los props anteriores
16. En **App.tsx**, coloca un componente **FotoRuta** con estos props:
 - **texto:** Albaicin
 - **ruta:** require("../assets/mejores1.jpg")
 - **tema:** El objeto con el tema seleccionado
17. Reemplaza las demás fotos de las rutas con objetos **FotoRuta**
18. En **components** crea un archivo llamado **MejoresRutas.tsx** y en él teclea **rnfs+Intro**
19. Mueve a **MejoresRutas.tsx** todos los componentes **FotoRuta**, usando un **fragment** (las etiquetas **<>** y **</>**) para poder devolverlas en el return
20. Dentro de **MejoresRutas.tsx** cambia las rutas de las fotos que usan **require** para que en lugar de usar **./** usen **../**

☞ **Punto de explicación:** La función **require** toma como ruta base (**./**) la del archivo en el que estamos. De esa forma, si estamos en **App.tsx**, entonces **./** es la carpeta base del proyecto y podemos entrar directamente en **assets**. Si estamos en un archivo de la carpeta **components** (como **MejoresRutas.tsx**) entonces **./** es la carpeta **components**, y debemos salirnos de ella (**../**) para poder situarnos en la del proyecto, y poder entrar en **assets**

21. Añade a la función **MejoresRutas** el prop **tema**, como ya has hecho en los otros puntos
22. En **App.tsx** añade un componente **MejoresRutas.tsx** pasándole el prop **tema**

☑ **Punto de comprobación:** Comprueba que las fotos de las mejores rutas está correctas

23. En **components**, crea un archivo llamado **FotoAlojamiento.tsx** y en él teclea **rnfs+Intro**
24. Mueve el **View** de la primera foto de alojamiento, junto con el **Image** que contiene y los estilos **contenedorAlojamiento** y **fotoAlojamiento** a **FotoAlojamiento.tsx**
25. Añade a la función **FotoAlojamiento** este props:
 - **imagen:** Es la imagen del alojamiento
26. En **App.tsx**, inserta donde estaba el **View** que te has llevado, un componente **FotoAlojamiento** con este prop:
 - **image:** require("../assets/alojamiento1.jpg")
27. En **components**, crea un archivo llamado **MejoresAlojamientos.tsx** y en él teclea **rnfs+Intro**
28. Mueve todo el **View** de los mejores alojamientos, junto con los estilos **contenedorFotosAlojamiento**, **contenedorAlojamiento** y **fotoAlojamiento** a **MejoresAlojamientos.tsx**
29. Revisa las rutas de las fotos que usan **require**, cambiando **./** por **../**

☑ **Punto de comprobación:** Comprueba que la cuadrícula de alojamientos se ve bien

30. Ejecuta **git add .** (esto añadirá al siguiente commit los nuevos archivos creados)
31. Haz un commit titulado "finalizada la versión con componentes"

• **Nuevo objetivo:** Vamos a pasar a **TypeScript** todos los componentes generados

32. Abre **Temas.ts** y en él crea un tipo llamado **Tema**, que tendrá estos campos y tipos:
- **colorFondo:** string
 - **colorTitulo:** string
 - **colorTextoFoto:** string
 - **logo:** ImageSourcePropType (cuando lo importes, usa el de **react-native**)
33. En el archivo **Temas.ts** exporta el tipo **Tema**
34. Abre **Titulo.tsx** y en él crea un tipo llamado **TituloProps**, con estos tipos de datos
- **texto:** string
 - **tema:** Tema
35. Haz que el objeto (que contiene **texto**, **tema**) que recibe la función **Titulo** sea de tipo **TituloProps**

```
1. export default function Titulo({texto,tema}:TituloProps) {  
2. // resto omitido  
3. }
```

36. Abre **FotoRuta.tsx** y en él crea un tipo llamado **FotoRutaProps** con estos campos:
- **texto:** string
 - **imagen:** ImageSourcePropType
 - **tema:** Tema
37. Haz que el objeto que recibe la función **FotoRuta** sea de tipo **FotoRutaProps**
38. Abre **MejoresRutas.tsx** y crea un tipo llamado **MejoresRutasProps** que contenga:
- **tema:** Tema
39. Haz que el objeto que recibe la función **MejoresRutas** sea de tipo **MejoresRutasProps**
40. Abre **FotoAlojamiento** y crea un tipo llamado **FotoAlojamientoProps** que contenga:
- **imagen:** ImageSourcePropType
41. Haz que el objeto que recibe la función **FotoAlojamiento** sea de tipo **FotoAlojamientoProps**

☒ **Punto de comprobación:** Comprueba que todo el proyecto se ve correctamente y no se muestra ningún error en los archivos de código fuente

42. Haz un commit titulado “finalizada versión TypeScript”

• **Nuevo objetivo:** Vamos a mejorar **Carrusel**, **MejoresRutas** y **MejoresAlojamientos** para poder pasarles una lista de fotos arbitraria, para que **generen dinámicamente** los componentes **Image**, **FotoRuta** y **FotoAlojamiento** que necesiten, en lugar de definirlos “a mano” en el código.

43. Crea una carpeta llamada **model** y en ella un archivo llamado **Tipos.ts**
44. En **Tipos.tsx** crea (y exporta) un tipo llamado **Ruta** con estos campos
- **nombre:** Un string con el nombre de la ruta (por ejemplo, Albaicin)
 - **imagen:** Un ImageSourcePropType con la foto de la ruta
45. Vete a **MejoresRutas.tsx** y añade a **MejoresRutasProps** este campo:
- **listaRutas:** Array<Ruta> → Es la lista con las rutas que se van a mostrar

46. Coloca a la función **MejoresRutas** un prop llamado **listaRutas** (que como hemos definido antes, será una lista con objetos **Ruta**)
47. Borra todos los componentes **FotoRuta** que hay dentro de `<>` y `</>`
48. Entre las etiquetas `<>` y `</>`, abre un bloque de llaves `{ }`
49. Dentro del bloque de llaves, usa el método **map** de **listaRutas** para transformar la lista de objetos **Ruta** de **listaRutas** en una lista de etiquetas cuyo prop **texto** sea el nombre de la ruta, y el prop **imagen** sea la ruta de la imagen, así:

```

1. export default function MejoresRutas({ tema, listaRutas }:MejoresRutasProps) {
2.   return (
3.     <>
4.       {
5.         listaRutas.map( ruta => <FotoRuta texto={ruta.nombre} imagen={ruta.imagen} tema={tema}/> )
6.       }
7.     </>
8.   );
9. }

```

50. En la función **App**, antes del return, crea una lista con objetos **Ruta** que se van a mostrar en la sección de mejores rutas:

```

1. const rutas = [
2.   {
3.     nombre: "Albaicin",
4.     imagen: require("../assets/mejores1.jpg")
5.   },
6.   {
7.     nombre: "Sacromonte",
8.     imagen: require("../assets/mejores2.jpg")
9.   },
10.  {
11.    nombre: "El Centro",
12.    imagen: require("../assets/mejores3.jpg")
13.  },
14. ]

```

51. Pasa la lista anterior al prop **listaRutas** que ahora te pide la etiqueta **MejoresRutas**

☒ **Punto de comprobación:** Comprueba que todo vuelve a mostrarse correctamente

52. Añade a la carpeta **assets** una nueva foto de algún lugar de Granada y añade dicho lugar a la lista **rutas**

☒ **Punto de comprobación:** Comprueba que la ruta añadida se muestra correctamente y observa lo fácil que ha sido añadirla. Hemos ganado facilidad de mantenimiento

53. Repite todos los pasos para poder añadir una lista alojamientos a **MejoresAlojamientos**
54. Repite todos los pasos para poder añadir una lista de fotos a **Carrusel**
55. Realiza un commit titulado "mejorados los componentes"
56. Haz un push de la rama **proyecto2**
57. Pasa a trabajar en la rama **main**
58. Fusiona la rama **proyecto2** con la rama **main**
59. Haz un push de la rama **main**
60. Borra la rama **proyecto2**