

INSTRUCCIONES:**Objetivo del proyecto**

En la clase 96, creaste una pantalla inicio para mostrar una lista de historias disponibles. También revisaste conceptos y resolviste errores de proyectos anteriores. En este proyecto, tienes que usar los conceptos aprendidos para diseñar la pantalla inicio para la app Spectrogram.

Historia:

Jenny es una fotógrafa. Ella quiere compartir sus fotos con otros. Al mismo tiempo, ella quiere crear un espacio para que otros compartan su talento también. Ella ha decidido crear una app social. Ella ha pedido tu ayuda para crear una app. Guía a Jenny para hacer la estructura básica creada en el proyecto pasado más atractiva, añadiendo imágenes y descripciones cortas para la app Spectrogram.

Output esperado (para tu referencia):

***Esto es solo para tu referencia. Esperamos que apliques tu propia creatividad al proyecto.**

Primeros pasos:

1. Crea un nuevo proyecto react-native usando los siguientes pasos:
 - a. Abre el símbolo del sistema:
 - i. Para Windows OS, escribe **cmd** en el cuadro **ejecutar**.
 - ii. Para macOS, escribe **Terminal** en la barra de búsqueda.
 - b. Ve a donde se encuentra el proyecto que quieres crear.
 - c. Escribe **expo init <nombreapp>** para crear un nuevo proyecto en el símbolo del sistema.
 - d. Ve al nuevo proyecto creado y escribe el comando **cd <nombreapp>** en el símbolo del sistema.
2. Abre el archivo del proyecto en VS Code.
3. Ejecuta el código escribiendo **expo start** en el símbolo del sistema/Terminal.
4. Comienza a editar tu código.

Tareas específicas para completar el proyecto:

1. Crea un archivo **temp_posts.json**.
 - Este archivo JSON contendrá una lista de objetos.
 - Cada objeto tendrá el nombre del autor y el título.
2. En la FeedScreen - (pantalla inicio), usando el componente **Text**, añade el título de la app spectrogram. (Ve pista 1).
3. Crea otro archivo llamado **PostCard.js**. (Ve pista 3).
4. Crea una etiqueta que contendrá una imagen de perfil del autor y el nombre del título del autor.
5. Añade la imagen usando el componente **Image** y el título usando el componente **Text** el cual lo obtendrás del estado.
6. Importa el componente **PostCard** en la **FeedScreen**.
7. Usa el **FlatList** para mostrar PostCards.
8. En la función **renderItem**, usa el componente **PostCard** y pasa la publicación.
9. Ejecuta y prueba el código.

Pistas:

1. Código para **FeedScreen**:

```
import PostCard from "../PostCard";
import { FlatList } from "react-native-gesture-handler";

let posts = require("../temp_posts.json");

export default class Feed extends Component {

  renderItem = ({ item: post }) => {
    return <PostCard post={post} />;
  };

  keyExtractor = (item, index) => index.toString();

  render() {
    return (
      <View style={styles.container}>
        <SafeAreaView style={styles.droidSafeArea} />
        <View style={styles.appTitle}>
          <Text style={styles.appTitleText}>Spectagram</Text>
        </View>
        <View style={styles.cardContainer}>
          <FlatList
            keyExtractor={this.keyExtractor}
            data={posts}
            renderItem={this.renderItem}
          />
        </View>
      </View>
    );
  }
}
```

2. Styles para FeedScreen.

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "darkblue"
  },
  droidSafeArea: {
    marginTop: Platform.OS === "android" ? StatusBar.currentHeight : 35
  },
  appTitle: {
    flex: 0.07,
    padding: 5,
    flexDirection: "row",
    justifyContent: "center"
  },
  appTitleText: {
    color: "white",
    fontSize: 28,
  },
  cardContainer: {
    flex: 0.93
  }
});
```

3. Código para **PostCard**:

```
export default class PostCard extends Component {  
  render() {  
    return (  
      <View style={styles.container}>  
        <View style={styles.cardContainer}>  
          <View style={styles.authorNameContainer}>  
            <Text style={styles.authorNameText}>  
              {this.props.post.author}  
            </Text>  
          </View>  
          <Image  
            source={require('../assets/post.jpeg')}  
            style={styles.postImage}  
          />  
          <View style={styles.captionContainer}>  
            <Text style={styles.captionText}>  
              {this.props.post.caption}  
            </Text>  
          </View>  
        </View>  
      </View>  
    );  
  }  
}
```

4. Styles para PostCard:

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  cardContainer: {
    margin: 10,
    backgroundColor: "white",
    borderRadius: 10,
    padding: 15,
  },
  authorNameContainer: {
    flex: 0.85,
    justifyContent: 'center',
  },
  authorNameText: {
    fontSize: 20,
  },
  postImage: {
    marginTop: 10,
    resizeMode: 'contain',
    width: '100%',
    alignSelf: 'center',
    height: 275,
  },
  captionText: {
    paddingTop: 10,
  },
});
```

Entrega del proyecto:

1. Crea un nuevo repositorio en GitHub y nómbralo **Proyecto 96**.
2. **Sube** el código de tu proyecto a tu repositorio de GitHub.
3. Copia el link de tu repositorio de **GitHub** y pégalo en el **Panel de proyectos del alumno** -> sube tu proyecto en el panel del **Proyecto**.

RECUERDA... Haz tu mejor esfuerzo, eso es lo más importante.

Después de entregar tu proyecto, tu maestra te enviará comentarios sobre tu trabajo.

_____ xxx _____ xxx _____ xxx _____ xxx _____ xxx _____

