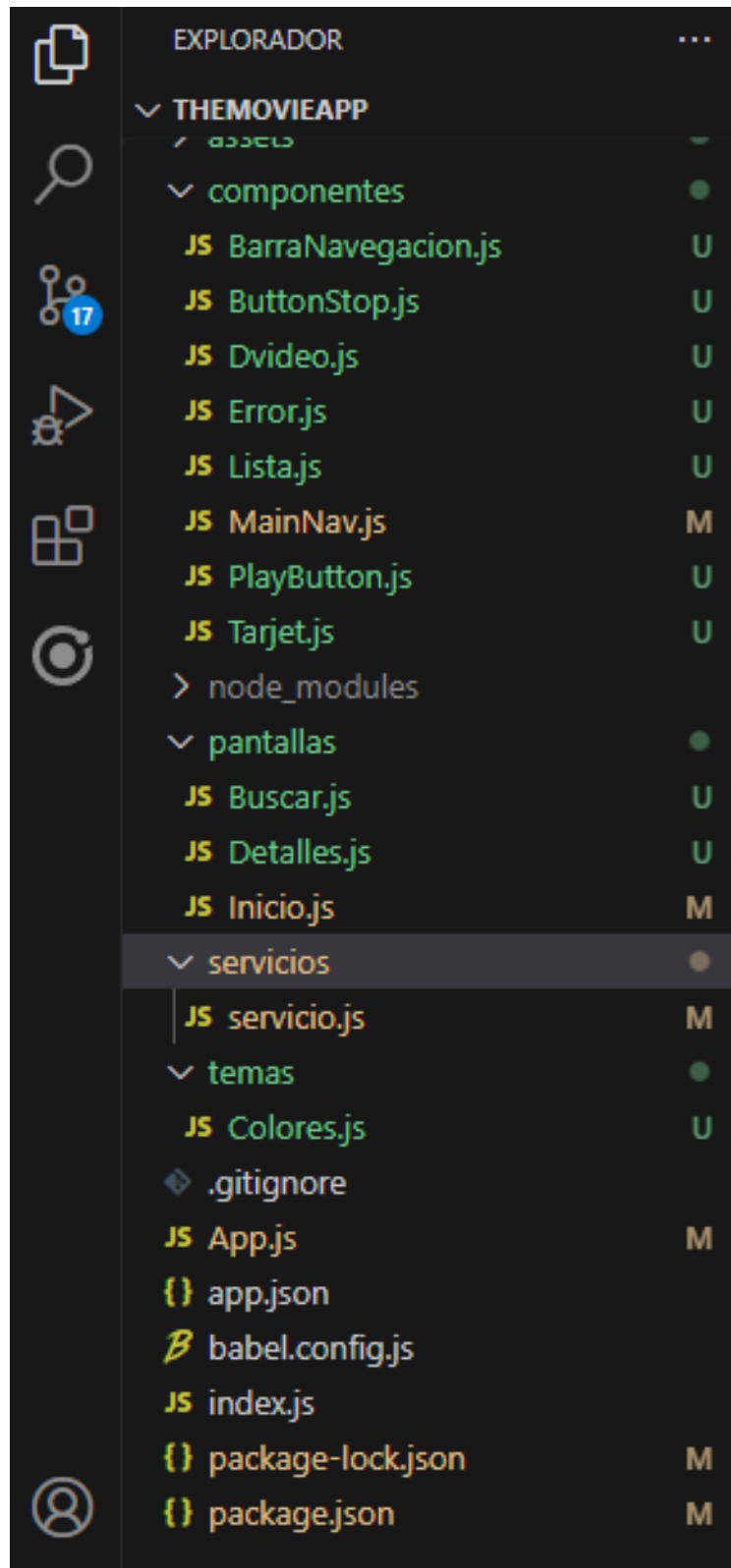


THE MOVIE APP

RAIZ DEL PROYECTO:



COMPONENTES

BARRA DE NAVEGACIÓN:

EXPLORADOR

THEMOVIEAPP

> .expo

> assets

▼ componentes

JS BarraNavegacion.js U

JS ButtonStop.js U

JS Dvideo.js U

JS Error.js U

JS Listas.js U

JS MainNav.js M

JS PlayButton.js U

JS Tarjet.js U

> node_modules

> pantallas

> servicios

> temas

◊ .gitignore

JS App.js M

() app.json

JS babel.config.js

JS index.js

() package-lock.json M

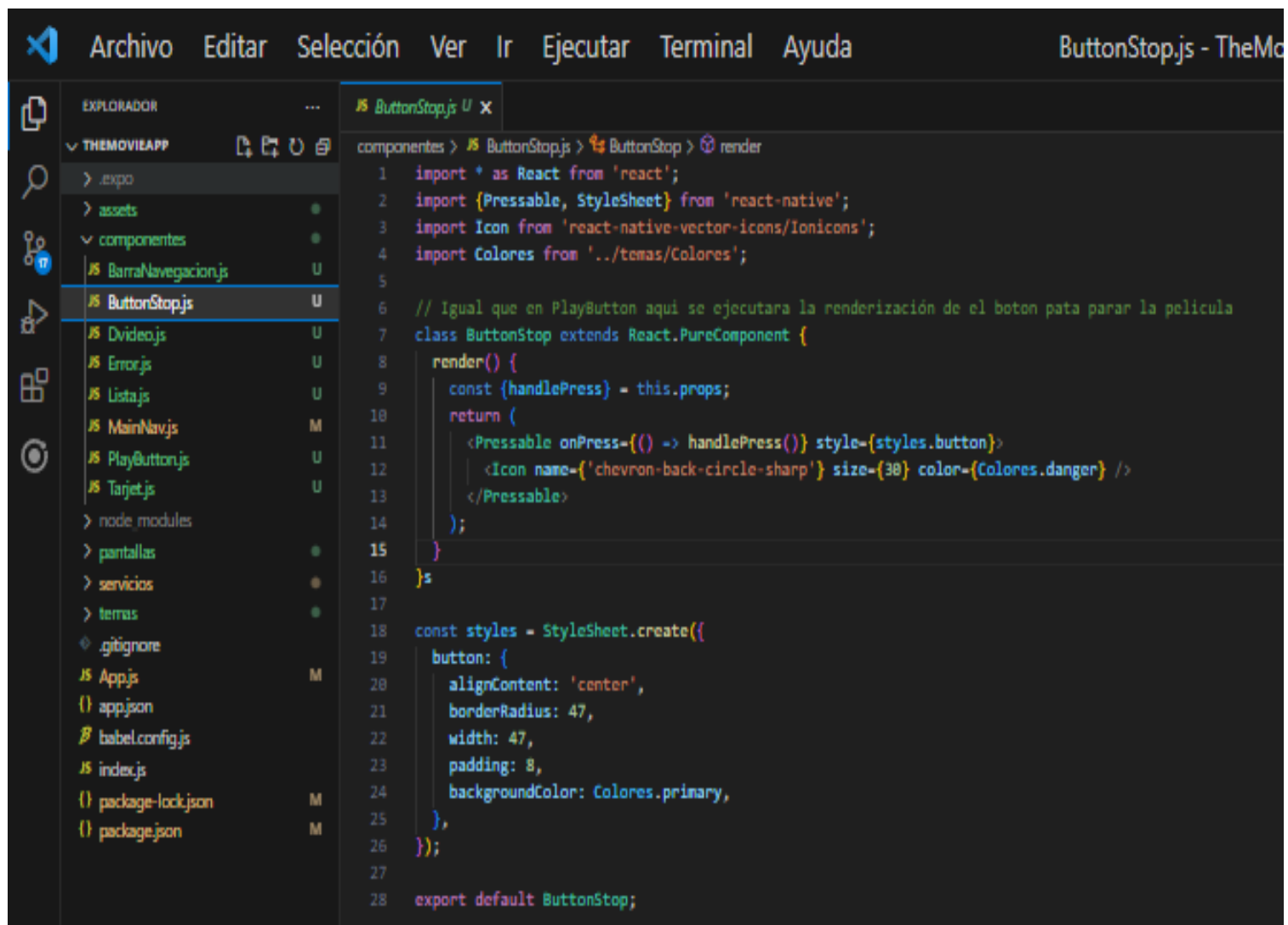
() package.json M

JS BarraNavegacion.js U X

componentes > JS BarraNavegacion.js > BarraNavegacion > render

```
1 import * as React from 'react';
2 import {View, SafeAreaView, TouchableOpacity, Image, StyleSheet} from 'react-native';
3 import Icon from 'react-native-vector-icons/Ionicons';
4 import PropTypes from 'prop-types';
5 import Colores from '../temas/Colores';
6
7 const propTypes={
8   main: PropTypes.bool,
9 }
10 const defaultProps = {
11   main: false,
12 }
13
14 // En esta clase se renderizara la navegacion del icono de busqueda.
15
16 class BarraNavegacion extends React.PureComponent{
17   state = {}
18   render () {
19     const {navigation, main} = this.props
20     return (
21       <SafeAreaView>
22         {main ? (
23           <View style = {styles.mainNav}>
24
25             <Image
26               style={style.logo}
27               source={require('../assets/favicon.png')}
28             />
29
30             <TouchableOpacity onPress={() => {
31               navigation.navigate('BUSQUEDA');
32             }}>
33               <Icon name = {'search-outline'} size={25} color={Colores.sucess}/>
34             </TouchableOpacity>
35           </View>
36         ) : (
37           <View>
38             <TouchableOpacity
39               onPress={()=>{
40                 navigation.goBack();
41               }}>
42               <Icon name = {'chevron.back'} size={35} color={Colores.white}/>
43             </TouchableOpacity>
44           </View>
45         )
46       </SafeAreaView>
47     )
48   }
49
50 BarraNavegacion.propTypes = propTypes;
51 BarraNavegacion.defaultProps = defaultProps;
52
53 export default BarraNavegacion;
```

BUTTONSTOP:



```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  ButtonStop.js - TheMo

EXPLORADOR  ...  JS ButtonStop.js U X
THEMOVIEAPP
  > .expo
  > assets
  > componentes
    JS BarraNavegacion.js U
    JS ButtonStop.js U
    JS Dvideo.js U
    JS Error.js U
    JS Listaj.js U
    JS MainNav.js M
    JS PlayButton.js U
    JS Tarjet.js U
  > node_modules
  > pantallas
  > servicios
  > temas
  > .gitignore
  JS App.js M
  {} app.json
  JS babel.config.js
  JS index.js
  {} package-lock.json M
  {} package.json M

componentes > JS ButtonStop.js > ButtonStop > render
1  import * as React from 'react';
2  import {Pressable, StyleSheet} from 'react-native';
3  import Icon from 'react-native-vector-icons/Ionicons';
4  import Colores from '../temas/Colores';
5
6  // Igual que en PlayButton aqui se ejecutara la renderización de el boton para para la película
7  class ButtonStop extends React.PureComponent {
8    render() {
9      const {handlePress} = this.props;
10     return (
11       <Pressable onPress={() => handlePress()} style={styles.button}>
12         <Icon name='chevron-back-circle-sharp' size={38} color={Colores.danger} />
13       </Pressable>
14     );
15   }
16 }
17
18 const styles = StyleSheet.create({
19   button: {
20     alignContent: 'center',
21     borderRadius: 47,
22     width: 47,
23     padding: 8,
24     backgroundColor: Colores.primary,
25   },
26 });
27
28 export default ButtonStop;
```

DVIDEO:

EXPLORADOR

THEMOVIEAPP

> expo

> assets

> componentes

JS BarraNavegacion.js

JS ButtonStop.js

JS Dvideo.js

JS Error.js

JS Lista.js

JS MainNav.js

JS PlayButton.js

JS Tarjet.js

> node_modules

> pantallas

> servicios

> temas

.gitignore

JS App.js

{ } app.json

.babel.config.js

JS index.js

{ } package-lock.json

{ } package.json

...

JS Dvideo.js U X

componentes > JS Dvideo.js > [M] Dvideo

```
1 import * as React from 'react';
2 import VideoPlayer from 'react-native-video-controls';
3 import Video from 'react-native-video';
4 import { StyleSheet } from 'react-native';
5
6 const Dvideo= ({onClose}) => {
7   return (
8     <Video
9       onBack={() => onClose()}
10      onEnd={() => onClose()}
11      fullscreenOrientation="all"
12      source={{uri: 'https://vjs.zencdn.net/v/oceans.mp4'}}
13      ref={(ref) => {
14        this.player = ref
15      }}
16      onBuffer={this.onBuffer}
17      onError={this.videoError}
18      style={styles.backgroundVideo}
19    />
20  );
21 };
22
23 const styles = StyleSheet.create({
24   backgroundVideo: {
25     position: 'absolute',
26     top: 0,
27     left: 0,
28     bottom: 0,
29     right: 0,
30   },
31 });
32
33 export default Dvideo;
```

ERROR:

EXPLORADOR

THEMOVIEAPP

> .expo

> assets

> componentes

JS BarraNavegacion.js

JS ButtonStop.js

JS Dvideo.js

JS Error.js

JS Lista.js

JS MainNav.js

JS PlayButton.js

JS Tarjet.js

> node_modules

> pantallas

> servicios

> temas

.gitignore

JS App.js

{ } app.json

babel.config.js

JS index.js

{ } package-lock.json

{ } package.json

JS Error.js

componentes > JS Error.js > propTypes

```
2 import {View,Text,StyleSheet} from 'react-native';
3 import PropTypes from 'prop-types';
4 import Colores from "../temas/Colores";
5
6 // Estas constantes tiene como función dar aviso en caso de que exista un error o haya una falla
7 // en la señal para abrir la pagina.
8 const propTypes={
9
10   errorText1: PropTypes.string,
11   errorText2: PropTypes.string,
12 }
13
14 const defaultProps = {
15   errorText1: 'Ok, Hubo Un Error',
16   errorText2: 'Verifica Tu Conexión o Refresca',
17 }
18 // Esta clase es para que cuando salga una de las alertas salga una forma ordenada.
19 class Error extends React.PureComponent{
20   render(){
21     const {errorText1, errorText2} = this.props;
22     return(
23       <View style= {styles.container}>
24         <Text style= {styles.text}>{errorText1}</Text>
25         <Text style= {styles.text}>{errorText2}</Text>
26       </View>
27     )
28   }
29 }
30
31
32
33 const styles = StyleSheet.create({
34   container: {
35     flex: 1,
36     justifyContent:'center',
37     alignItems: 'center',
38   },
39
40   text:{
41     fontWeight:'bold',
42     color: Colores.warning,
43   },
44 })
45
46 Error.PropTypes = propTypes;
47 Error.defaultProps = defaultProps;
48
49 export default Error;
```

LISTA:

EXPLORADOR

THEMOVIEAPP

> .expo

> assets

> componentes

JS BarraNavegacion.js

JS ButtonStop.js

JS Dvideo.js

JS Error.js

JS Listajs

JS MainNav.js

JS PlayButton.js

JS Tarjet.js

> node_modules

> pantallas

> servicios

> temas

.gitignore

JS App.js

{ } app.json

babel.config.js

JS index.js

{ } package-lock.json

{ } package.json

JS Listajs

componentes > JS Listajs > Lista > render

```
6  const propTypes={
7    title: PropTypes.string,
8    content: PropTypes.array,
9  };
10
11  class Lista extends React.PureComponent{
12    render(){
13      const{navigation, title, content} = this.props;
14      return(
15        <View style={styles.list}>
16          <View>
17            <Text style={styles.text}>{title}</Text>
18          </View>
19          <View>
20            <Flatlist
21              data={content}
22              horizontal={true}
23              renderItem={({item})=>{
24                <Tarjet navigation={navigation} item={item}/>
25              }}
26            />
27          </View>
28        </View>
29      );
30    }
31  }
32
33
34
35
36  const styles = StyleSheet.create({
37    list: {
38      marginTop: 25,
39    },
40
41    text:{
42      fontSize: 20,
43      fontWeight: 'bold',
44      padding: 10,
45      paddingBottom: 15,
46    },
47  });
48
49
50  Lista.propTypes=propTypes;
51  export default Lista;
```

MAINNAV:

EXPLORADOR

THEMOVIEAPP

> .expo

> assets

> componentes

> JS BarraNavegacion.js

> JS ButtonStop.js

> JS Dvideo.js

> JS Error.js

> JS Listas.js

> JS MainNav.js

> JS PlayButton.js

> JS Tarjet.js

> node_modules

> pantallas

> servicios

> temas

> .gitignore

> App.js

> app.json

> babel.config.js

> index.js

> package-lock.json

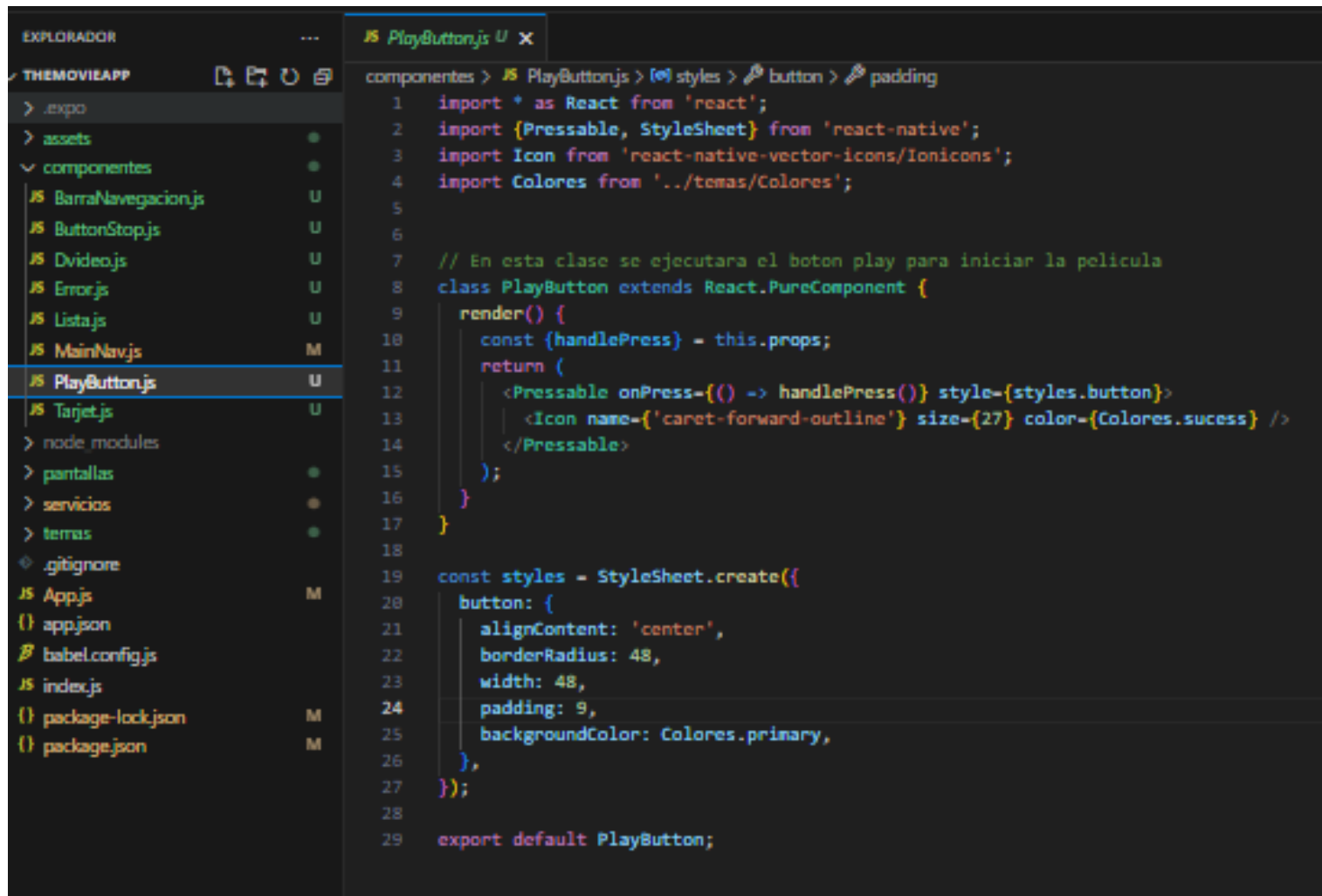
> package.json

JS MainNav.js

componentes > JS MainNav.js > MainNav > render

```
1 import * as React from 'react';
2 import { createStackNavigator } from '@react-navigation/stack';
3 import Inicio from '../Pantallas/Inicio';
4 import BarraNavegacion from './BarraNavegacion';
5 import Buscar from '../pantallas/Buscar';
6 import Detalles from '../pantallas/Detalles';
7
8 const Stack = createStackNavigator()
9 class MainNav extends React.PureComponent{
10   render(){
11     return(
12       <Stack.Navigator headerMode='screen'>
13         <Stack.Screen
14           name = "Inicio"
15           component={Inicio}
16           options={{
17             headerTransparent:true,
18             header:({navigation}) => (
19               <BarraNavegacion navigation={navigation} main={true} />
20             )
19           }}
21         />
22       <Stack.Screen
23         name = "Detalles"
24         component={Detalles}
25         options={{
26           headerTransparent:true,
27           header:({navigation}) => (
28             <BarraNavegacion navigation={navigation} main={false} />
29           )
30         }}
31       />
32       <Stack.Screen
33         name = "Búsqueda"
34         component={Buscar}
35         options={{
36           headerTransparent:true,
37           header:({navigation}) => (
38             <BarraNavegacion navigation={navigation} main={false} />
39           )
40         }}
41       />
42     </Stack.Navigator>
43   )
44 }
45
46 export default MainNav
```

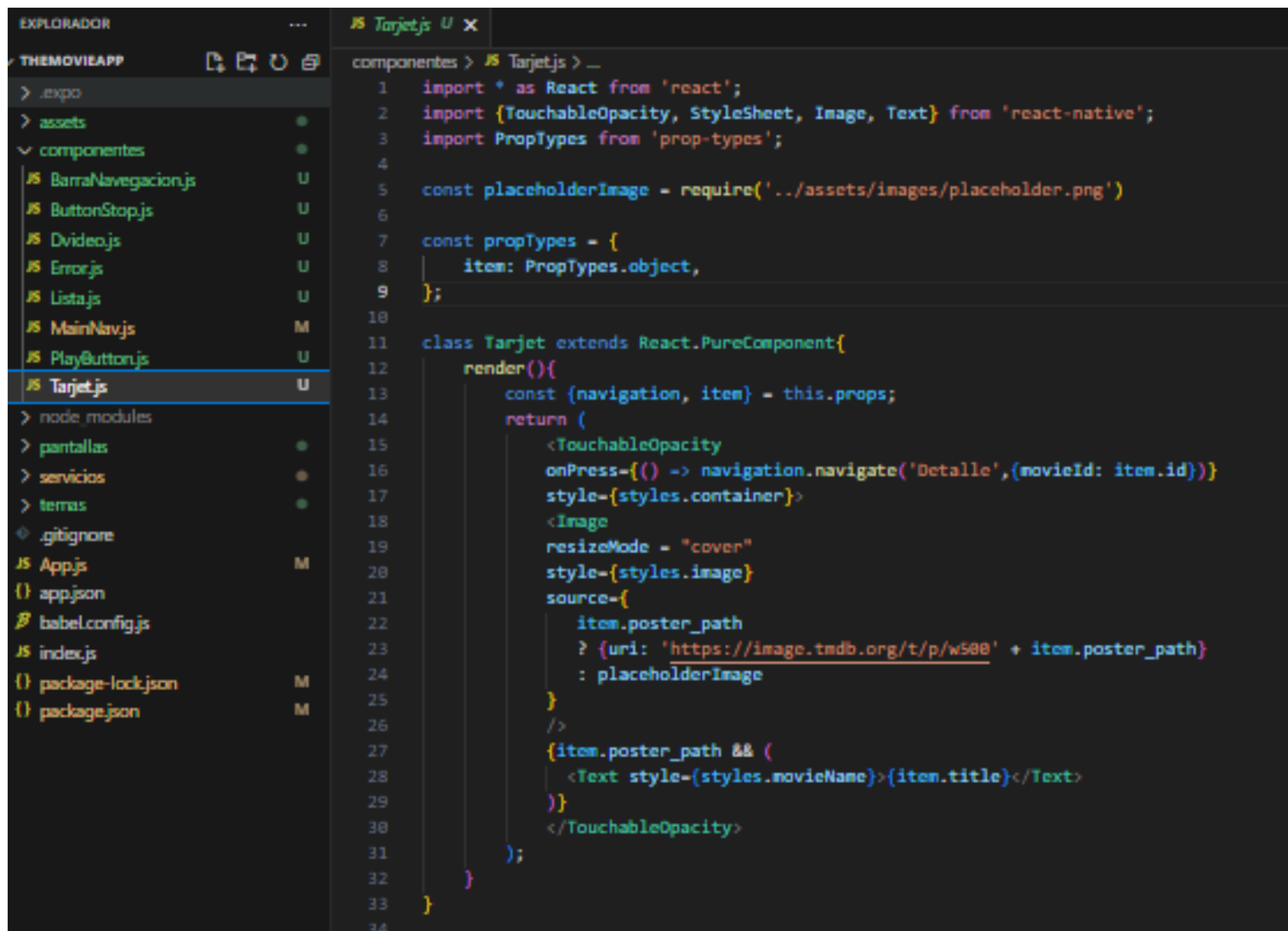
PLAYBUTTON:



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer, titled 'EXPLORADOR', shows a project named 'THEMOVIEAPP'. Under the 'componentes' folder, 'PlayButton.js' is selected. The code editor, titled 'JS PlayButton.js U X', shows the following code:

```
1  import * as React from 'react';
2  import {Pressable, StyleSheet} from 'react-native';
3  import Icon from 'react-native-vector-icons/Ionicons';
4  import Colores from '../temas/Colores';
5
6
7  // En esta clase se ejecutara el boton play para iniciar la pelicula
8  class PlayButton extends React.PureComponent {
9    render() {
10      const {handlePress} = this.props;
11      return (
12        <Pressable onPress={() => handlePress()} style={styles.button}>
13          <Icon name={'caret-forward-outline'} size={27} color={Colores.sucess} />
14        </Pressable>
15      );
16    }
17  }
18
19  const styles = StyleSheet.create({
20    button: {
21      alignContent: 'center',
22      borderRadius: 48,
23      width: 48,
24      padding: 9,
25      backgroundColor: Colores.primary,
26    },
27  });
28
29  export default PlayButton;
```


TARJET: (1)



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'THEMOVIEAPP' with various files and folders. The code editor shows the 'Tarjet.js' file, which is a React Native component. The code defines a class 'Tarjet' that extends 'React.PureComponent' and implements a 'render()' method. The 'render()' method uses 'TouchableOpacity' and 'Image' components to display movie information, including a poster image and the movie title. The code also includes imports for 'React', 'TouchableOpacity', 'StyleSheet', 'Image', 'Text', and 'PropTypes'.

```
EXPLORADOR
THEMOVIEAPP
  > .expo
  > assets
  > componentes
    JS BarraNavegacion.js U
    JS ButtonStop.js U
    JS Dvideo.js U
    JS Error.js U
    JS Lista.js U
    JS MainNav.js M
    JS PlayButton.js U
    JS Tarjet.js U
  > node_modules
  > pantallas
  > servicios
  > temas
  > .gitignore
  JS App.js M
  {} app.json
  JS babel.config.js
  JS index.js
  {} package-lock.json M
  {} package.json M

JS Tarjet.js U X
componentes > JS Tarjet.js > ...
1  import * as React from 'react';
2  import {TouchableOpacity, StyleSheet, Image, Text} from 'react-native';
3  import PropTypes from 'prop-types';
4
5  const placeholderImage = require('../assets/images/placeholder.png')
6
7  const propTypes = {
8    |   item: PropTypes.object,
9  };
10
11  class Tarjet extends React.PureComponent{
12    |   render(){
13    |     |   const {navigation, item} = this.props;
14    |     |   return (
15    |     |     <TouchableOpacity
16    |     |       onPress={() => navigation.navigate('Detalle',{movieId: item.id})}
17    |     |       style={styles.container}>
18    |     |       <Image
19    |     |         resizeMode = "cover"
20    |     |         style={styles.image}
21    |     |         source={
22    |     |           item.poster_path
23    |     |           ? {uri: 'https://image.tmdb.org/t/p/w500' + item.poster_path}
24    |     |           : placeholderImage
25    |     |         }
26    |     |       />
27    |     |       {item.poster_path && (
28    |     |         <Text style={styles.movieName}>{item.title}</Text>
29    |     |       )}
30    |     |     </TouchableOpacity>
31    |   )};
32  }
33
34
```

TARJET: (2)

```
34
35   const styles = StyleSheet.create({
36     container: {
37       padding: 4,
38       position: 'relative',
39       alignItems: 'center',
40       height: 198,
41       marginBottom: 7,
42     },
43
44     image: {
45       height: 199,
46       width: 118,
47       borderRadius: 18,
48     },
49
50     movieName: {
51       position: 'absolute',
52       width: 98,
53       top: 9,
54       textAlign: 'center'
55     },
56   });
57
58   Tarjet.propTypes = propTypes;
59   export default Tarjet;
```

PANTALLAS / VISTAS

BUSCAR: (1)

```
EXPLORADOR ...
THEMOVIEAPP
  > .expo
  > assets
  > componentes
  > node_modules
  ▼ pantallas
    JS Buscar.js U
    JS Detalles.js U
    JS Inicio.js M
  > servicios
  > temas
  .gitignore
  JS App.js M
  {} app.json
  babel.config.js
  JS index.js
  {} package-lock.json M
  {} package.json M

JS Buscar.js U
pantallas > JS Buscar.js > [0] Buscar
1  import React, {useState} from 'react';
2  import {SafeAreaView, View, TextInput, StyleSheet, TouchableOpacity, FlatList, Text,} from 'react-native';
3  import Icon from 'react-native-vector-icons/Ionicons';
4  import { BusPeliTv } from '../servicios/servicio';
5  import Tarjet from '../componentes/Tarjet';
6  import Error from '../componentes/Error';
7
8  const Buscar = ({navigation}) => {
9    const [text, onChangeText] = useState();
10   const [buscarResultados, setBuscarResultados] = useState();
11   const [error, setError] = useState(false);
12
13   const onSubmit = query => {
14     Promise.all([BusPeliTv(query, 'movie'), BusPeliTv(query, 'tv')])
15       .then(([movies, tv]) => {
16         const data = [...movies, ...tv];
17         setBuscarResultados(data);
18       })
19       .catch(() => {
20         setError(true);
21       });
22   };
23
24   return (
25     <React.Fragment>
26       <SafeAreaView>
27         <View style={styles.container}>
28           <View style={styles.form}>
29             <TextInput
30               style={styles.input}
31               placeholder='BUSCA UNA PELICULA O SHOW DE TV'
32               onChangeText={onChangeText}
33               value={text}
34             />
35           </View>
36           <TouchableOpacity
37             onPress={() => {
38               onSubmit(text);
39             }}>
40             <Icon name='search-outline' size={30} />
41           </TouchableOpacity>
42         </View>
43       </React.Fragment>
44     );
45   };
46
47   return <Buscar />
48 }
```

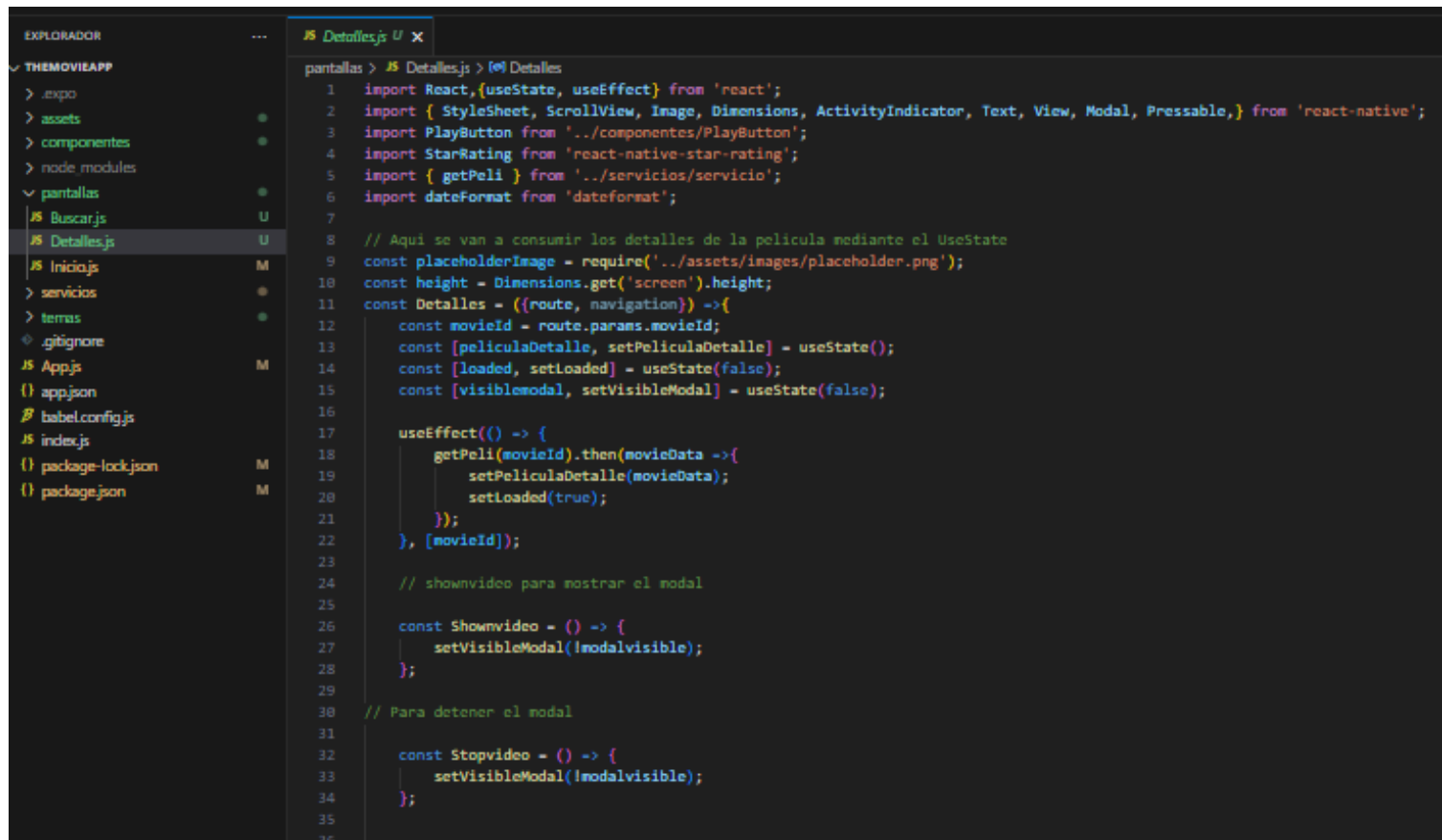
BUSCAR: (2)

```
1   <View style={styles.searchItems}>
2     {/* Para el resultado de la búsqueda */}
3     {buscarresultados && buscarresultados.length > 0 && (
4       <FlatList
5         numColumns={3}
6         data={buscarresultados}
7         renderItem={({item}) => (
8           <Tarjet navigation={navigation} item={item} />
9         )}
10        keyExtractor={item => item.id}
11      />
12    )}
13
14    {/* Esto es para Cuando no se encuentran resultados visibles */}
15    {buscarresultados && buscarresultados.length == 0 && (
16      <View style={styles.noResults}>
17        <Text>NO ES POSIBLE ENCONTRAR ESTE RESULTADO POR ESTE CRITERIO</Text>
18        <Text>DEBERIAS PROBAR CON OTRA PALABRA DIFERENTE.</Text>
19      </View>
20    )}
21
22    {/* Cuando no se ha realizado una búsqueda */}
23    {!buscarresultados && (
24      <View style={styles.empty}>
25        <Text>DEBES ESCRIBIR ALGO PARA INICIAR LA BUSQUEDA</Text>
26      </View>
27    )}
28
29    {/* Error */}
30    {error && <Error />}
31  </View>
32</SafeAreaView>
33</React.Fragment>
34);
35};
```

BUSCAR: (3)

```
79
80   const styles = StyleSheet.create({
81     input: {
82       borderRadius: 15,
83       borderWidth: 0.5,
84       height: 50,
85       padding: 8,
86     },
87     container: {
88       padding: 10,
89       paddingTop: 60,
90       flexDirection: 'row',
91       alignItems: 'center',
92     },
93     form: {
94       flexBasis: 'auto',
95       flexGrow: 1,
96       paddingRight: 8,
97     },
98
99     searchItems: {
100       padding: 5,
101     },
102
103     noResults: {
104       paddingTop: 20,
105     },
106   });
107
108   export default Buscar;
```

DETALLES: (1)



```
1 import React, {useState, useEffect} from 'react';
2 import { StyleSheet, ScrollView, Image, Dimensions, ActivityIndicator, Text, View, Modal, Pressable, } from 'react-native';
3 import PlayButton from '../componentes/PlayButton';
4 import StarRating from 'react-native-star-rating';
5 import { getPeli } from '../servicios/servicio';
6 import dateFormat from 'dateformat';
7
8 // Aqui se van a consumir los detalles de la pelicula mediante el UseState
9 const placeholderImage = require('../assets/images/placeholder.png');
10 const height = Dimensions.get('screen').height;
11 const Detalles = ({route, navigation}) => {
12   const movieId = route.params.movieId;
13   const [peliculaDetalle, setPeliculaDetalle] = useState();
14   const [loaded, setLoaded] = useState(false);
15   const [visibleModal, setVisibleModal] = useState(false);
16
17   useEffect(() => {
18     getPeli(movieId).then(movieData => {
19       setPeliculaDetalle(movieData);
20       setLoaded(true);
21     });
22   }, [movieId]);
23
24   // shownvideo para mostrar el modal
25
26   const Shownvideo = () => {
27     setVisibleModal(!modalvisible);
28   };
29
30   // Para detener el modal
31
32   const Stopvideo = () => {
33     setVisibleModal(!modalvisible);
34   };
35
36
```

DETALLES: (2)

```
//Aquí se retornara la ruta que se esta consumiendo del api thenovie.  
return (  
  <React.Fragment>  
    {loaded && (  
      <View>  
        <ScrollView>  
          <Image  
            resizeMode='cover'  
            style={styles.image}  
            source={peliculaDetalle.poster_path  
              ? {  
                uri: 'https://image.tmbd.org/t/p/w500' + peliculaDetalle.poster_path,  
              }  
              : placeholderImage  
            }  
          />  
          <View style = {styles.container}>  
            <View style = {styles.PlayButton}>  
              <PlayButton handlePress={Shownvideo}/>  
            </View>  
            <Text style = {styles.movieTitle}>{peliculaDetalle.title}</Text>  
            {peliculaDetalle.genres &&(  
              <View style = {styles.genresContainer}>  
                {peliculaDetalle.genres.map(genre =>{  
                  return(  
                    <Text style = {styles.genre} key={genre.id}>  
                      {genre.name}  
                    </Text>  
                  );  
                })}  
              )}  
            </View>  
          )}  
          <StarRating  
            disabled={true}  
            maxStars={5}  
            starSize={30}  
            rating={peliculaDetalle.vote_average / 2}  
            fullStarColor={'gold'}  
          />  
          <Text style={styles.overview}>{peliculaDetalle.overview}</Text>  
  
          <Text style={styles.release}>  
            {'Release date: ' +  
              dateFormat(peliculaDetalle.release_date, 'mmmm dS, yyyy')}  
          </Text>  
        </View>  
      </ScrollView>  
    )  
  )  
);
```

DETALLES: (3)

```
<Modal
  supportedOrientations={['portrait', 'landscape']}
  animationType="slide"
  visible={visiblemodal}>
  <View style={styles.videoModal}>

    {/ * <Video0 onClose={videoShown} /> */}
    {/ * <VideoPlayer
    source={{uri: 'https://vjs.zencdn.net/v/oceans.mp4'}} /> */}

    <Pressable>
      <Text> Debería Ser Un Modal Con El Reproductor Si Hubiese Acceso Al Video</Text>
      <ButtonStop onPress={Stopvideo} />
    </Pressable>

  </View>
</Modal>
</View>
)}
{!loaded && <ActivityIndicator size="large" />}
</React.Fragment>
```

```
);
```

```
};
```


DETALLES: (4)

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  image: {
    height: height / 2.5,
  },
  movieTitle: {
    fontSize: 24,
    fontWeight: 'bold',
    marginTop: 10,
    marginBottom: 10,
  },
  genresContainer: {
    flexDirection: 'row',
    alignContent: 'center',
    marginTop: 20,
    marginBottom: 20,
  },
  genre: {
    marginRight: 10,
    fontWeight: 'bold',
  },
  overview: {
    padding: 15,
  },
  release: {
    fontWeight: 'bold',
  },
  PlayButton: {
    position: 'absolute',
    top: -25,
    right: 20,
  },
  videoModal: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
});

export default Detalles;
```

INICIO: (1)

EXPLORADOR

THEMOVIEAPP

> expo

> assets

> componentes

> node_modules

▼ pantallas

JS Buscar.js U

JS Detalles.js U

JS Inicio.js M

> servicios

> temas

gitignore

JS App.js M

() app.json

babel.config.js

JS index.js

() package-lock.json M

() package.json M

ESQUEMA

JS Detalles.js U

JS Inicio.js M X

pantallas > JS Inicio.js > Inicio > [0] getData

1 import React, {useEffect,useState} from "react";

2 import { ActivityIndicator, View, StyleSheet, Dimensions, ScrollView, Slider } from "react-native";

3 import {getPeliPopulares, getProxPeli, getProPopulares, getFamPopulares, getDocPeli} from '../servicios/servicio'

4 import { SliderBox } from "react-native-image-slider-box";

5 import react from "react";

6 import Lista from '../componentes/Lista';

7 import Error from '../componentes/Error';

8

9 const dimensiones = Dimensions.get ('screen')

10

11 const Inicio = ({navigation}) =>{

12 /* Estos son los hooks para las diferentes categorias */

13 const [Imagenes_Peliculas, setimagenes_peliculas] = useState();

14 const [Peliculas_Populares, setpeliculas_populares] = useState();

15 const [Tv_Popular, settv_popular] = useState();

16 const [Peliculas_Familiares, setpeliculas_familiares] = useState();

17 const [Peliculas_Documentales, setpeliculas_documentales] = useState();

18

19 /* Estos son los hooks para los estados de carga y errores */

20 const [Error, seterror] = useState(false);

21 const [Loaded, setloaded] = useState(false);

22

23 const getData= ()=>{

24 return Promise.all([

25 getPeliPopulares(),

26 getProxPeli(),

27 getProPopulares(),

28 getFamPopulares(),

29 getDocPeli(),

30])

31 }

32

33 useEffect(() =>{

34 getData()

35 .then(

36 ([

37 PeliPopularesData,

38 ProxPeliData,

39 ProPopularesData,

40 FamPopularesData,

41 DocPeliData,

42

43]) => {

44 const Imagenes_PeliculasArray = []

45 ProxPeliData.forEach(movie =>{

46 Imagenes_PeliculasArray.push(

47 'https://image.tmdb.org/t/p/w500'+movie.poster_path,

48)

49 })

50

51 /*Aqui se modificaran los estados de las categorias */

52 setimagenes_peliculas(Imagenes_PeliculasArray);

53 setpeliculas_populares(PeliPopularesData);

54 settv_popular(ProPopularesData);

55 setpeliculas_familiares(FamPopularesData);

56 setpeliculas_documentales(DocPeliData);

57 }

58 })

INICIO: (2)

```
.catch(()=>{
    seterror(true)
})
.finally(()=>{
    setloaded(true)
})
},[])

return(
    <react.Fragment>
        { /* PROXIMAS PELICULAS */ }
        {loaded && !Error && (
            <ScrollView>
                {Imágenes_Películas && (
                    <View style={styles.sliderContainer}>
                        <SliderBox
                            images={Imágenes_Películas}
                            dotStyle={styles.sliderStyle}
                            SliderBoxHeight={dimensiones.height /1.5}
                            autoplay={true}
                            circleLoop={true}
                        />
                    </View>
                )}

                { /* PELICULAS POPULARES */ }
                {Películas_Populares && (
                    <View style={styles.carousel}>
                        <Lista
                            navigation={navigation}
                            title={"PELICULAS POPULARES ENTRE EL PUBLICO"}
                            content={Películas_Populares}
                        />
                    </View>
                )}

                { /* PROGRAMAS POPULARES DE TELEVISIÓN */ }
                {Tv_Popular && (
                    <View style={styles.carousel}>
                        <Lista
                            navigation={navigation}
                            title={"PROGRAMAS POPULARES DE TELEVISIÓN"}
                            content={Tv_Popular}
                        />
                    </View>
                )}
            </react.Fragment>
        )
    )
```

INICIO: (3)

```
        { /* PROGRAMAS POPULARES DE TELEVISIÓN */ }
        { Tv_Popular && {
            <View style={styles.carousel}>
                <Lista
                    navigation={navigation}
                    title={"PROGRAMAS POPULARES DE TELEVISIÓN"}
                    content={Tv_Popular}
                />
            </View>
        } }

        { /* PELICULAS FAMILIARES */ }
        { Peliculas_Familiares && {
            <View style={styles.carousel}>
                <Lista
                    navigation={navigation}
                    title={"PELICULAS FAMILIARES"}
                    content={Peliculas_Familiares}
                />
            </View>
        } }

        { /* PELICULAS DOCUMENTALES */ }
        { Peliculas_Documentales && {
            <View style={styles.carousel}>
                <Lista
                    navigation={navigation}
                    title={"DOCUMENTALES"}
                    content={Peliculas_Documentales}
                />
            </View>
        } }
    </ScrollView>

    )
    { !Loaded && <ActivityIndicator size='large' /> }
    { Error && <Error /> }
</react.Fragment>

)
```

INICIO: (4)

```

3
4
5
6
7
8 const styles = StyleSheet.create({
9   sliderContainer: {
10     flex: 1,
11     justifyContent: 'center',
12     alignItems: 'center',
13   },
14
15   sliderStyle:{
16     height: 8,
17   },
18
19   carousel:{
20     flex: 1,
21     justifyContent: 'center',
22     alignItems: 'center',
23   }
24 });
25 export default Inicio

```

SERVICIOS

SERVICIO:

EXPLORADOR	...	JS Detalles.js U	JS servicio.js M X
THEMOVIEAPP		servicios > JS servicio.js > ...	
> .expo		1	import axios from "axios";
> assets		2	
> componentes		3	//Apikey y Url del sitio el cual se va consumir los servicios
> node_modules		4	const apiUrl = 'https://api.themoviedb.org/3';
> pantallas		5	const apikey = 'api_key=46d9f44bb41a5348ed72c97c55581681';
> servicios		6	
JS servicio.js	M	7	// Con estos metodos GET se podran consumir cada una de las categorias
> temas		8	// de la api themoviedb.
> .gitignore		9	
JS App.js	M	10	//Obtener Categoria Peliculas Populares/GET
{ } app.json		11	export const getPeliPopulares = async () =>{
babel.config.js		12	const respuesta = await axios.get(`\${apiUrl}/movie/popular?\${apikey}`);
JS index.js		13	return respuesta.data.results;
{ } package-lock.json	M	14	};
{ } package.json	M	15	
		16	//Obtener Categoria Proximas Peliculas/GET
		17	export const getProxPeli = async () =>{
		18	const respuesta = await axios.get(`\${apiUrl}/movie/upcoming?\${apikey}`);
		19	return respuesta.data.results;
		20	};
		21	
		22	//Obtener Categoria Programas Populares De Televisión/GET
		23	export const getProPopulares = async () =>{
		24	const respuesta = await axios.get(`\${apiUrl}/tv/popular?\${apikey}`);
		25	return respuesta.data.results;
		26	};
		27	
		28	//Obtener Categoria Peliculas Familiar En Peliculas/GET
		29	export const getFamPopulares = async () =>{
		30	const respuesta = await axios.get(`\${apiUrl}/discover/movie?\${apikey}&with_genres=10751`,);
		31	return respuesta.data.results;
		32	};
		33	
		34	//Obtener Categoria Documentales En Peliculas/GET
		35	export const getDocPeli = async () =>{
		36	const respuesta = await axios.get(`\${apiUrl}/discover/movie?\${apikey}&with_genres=99`,);
		37	return respuesta.data.results;
		38	};
		39	
		40	//Renderización de otras funciones
		41	
		42	//Obtención De Peliculas Por ID
		43	export const getPeli = async id =>{
		44	const respuesta = await axios.get(`\${apiUrl}/movie/\${id}?\${apikey}`);
		45	return respuesta.data.results;
		46	};
		47	
		48	//Buscador de contenido por medio de caracter (letra o palabra)
		49	
		50	export const BusPeliTv = async (query, type) =>{
		51	const respuesta = await axios.get(`\${apiUrl}/search/\${type}?\${apikey}&query=\${query}`);
		52	return respuesta.data.results;
		53	};
		54	

COLORES

PREDETERMINADOS:

```
temas > JS Colores.js > [0] Colores
1  //Colores predeterminados para el proyecto.
2
3  const Colores = {
4    background: '#1E1E1E',
5    black: '#000000',
6    white: '#ffffff',
7    lightGray: '#ccc',
8    primary: '#4481FC',
9    danger: '#F5365C',
10    success: '#488543',
11    warning: '#ffc107',
12  };
13
14  export default Colores;
```

EJECUCIÓN WEB

The screenshot shows the Visual Studio Code interface with the 'TheMovieApp - Visual Studio Code' window. The Explorer sidebar on the left displays the project structure, including 'THEMOVIEAPP' with subfolders like '.expo', 'assets', and 'componentes'. The 'componentes' folder is expanded, showing files like 'BarraNavegacion.js', 'ButtonStop.js', 'Dvideo.js', 'Error.js', 'Listas.js', 'MainNav.js', 'PlayButton.js', 'Target.js', 'node_modules', 'pantallas', 'servicios', 'temas', '.gitignore', 'App.js', 'app.json', 'babel.config.js', 'index.js', 'package-lock.json', and 'package.json'. The 'BarraNavegacion.js' file is selected. The main editor area shows the code for 'BarraNavegacion.js' with a warning message: 'WARNING in ./node_modules/react-native-star-rating/StarRating.js:55:13 export \'ViewPropTypes\' (imported as \'ViewPropTypes\') was not found in \'react-native-web/dist/index\' (possible exports: AccessibilityInfo, ActivityIndicator, Alert, Animated, AppRegistry, AppState, Appearance, BackHandler, Button, CheckBox, Clipboard, DeviceEventEmitter, DeviceInfo, Dimensions, DrawerLayoutAndroid, Easing, FlatList, I18nManager, Image, ImageBackground, InputAccessoryView, InteractionManager, Keyboard, KeyboardAvoidingView, LayoutAnimation, Linking, LogBox, Modal, NativeEventEmitter, NativeModules, PanResponder, PermissionsAndroid, Picker, PixelRatio, Platform, Pressable, ProgressBar, RefreshControl, SafeAreaView, ScrollView, SectionList, Settings, Share, StatusBar, StyleSheet, Switch, Systrace, TVEventHandler, Text, TextInput, ToastAndroid, Touchable, TouchableHighlight, TouchableNativeFeedback, TouchableOpacity, TouchableWithoutFeedback, UIManager, Vibration, View, VirtualizedList, YellowBox, findNodeHandle, processColor, render, unmountComponentAtNode, unstable_createElement, useColorScheme, useLocaleContext, useWindowDimensions)'. The warning message is repeated twice. The bottom status bar shows 'web compiled with 31 warnings'.

Visual Studio Code interface showing the Explorer sidebar with the project structure of 'THEMOVIEAPP'. The main editor displays the code for 'BarraNavegacion.js' with a warning message: 'WARNING in ./node_modules/react-native-star-rating/StarRating.js:55:13 export \'ViewPropTypes\' (imported as \'ViewPropTypes\') was not found in \'react-native-web/dist/index\' (possible exports: AccessibilityInfo, ActivityIndicator, Alert, Animated, AppRegistry, AppState, Appearance, BackHandler, Button, CheckBox, Clipboard, DeviceEventEmitter, DeviceInfo, Dimensions, DrawerLayoutAndroid, Easing, FlatList, I18nManager, Image, ImageBackground, InputAccessoryView, InteractionManager, Keyboard, KeyboardAvoidingView, LayoutAnimation, Linking, LogBox, Modal, NativeEventEmitter, NativeModules, PanResponder, PermissionsAndroid, Picker, PixelRatio, Platform, Pressable, ProgressBar, RefreshControl, SafeAreaView, ScrollView, SectionList, Settings, Share, StatusBar, StyleSheet, Switch, Systrace, TVEventHandler, Text, TextInput, ToastAndroid, Touchable, TouchableHighlight, TouchableNativeFeedback, TouchableOpacity, TouchableWithoutFeedback, UIManager, Vibration, View, VirtualizedList, YellowBox, findNodeHandle, processColor, render, unmountComponentAtNode, unstable_createElement, useColorScheme, useLocaleContext, useWindowDimensions)'. The bottom status bar indicates 'web compiled with 31 warnings'.

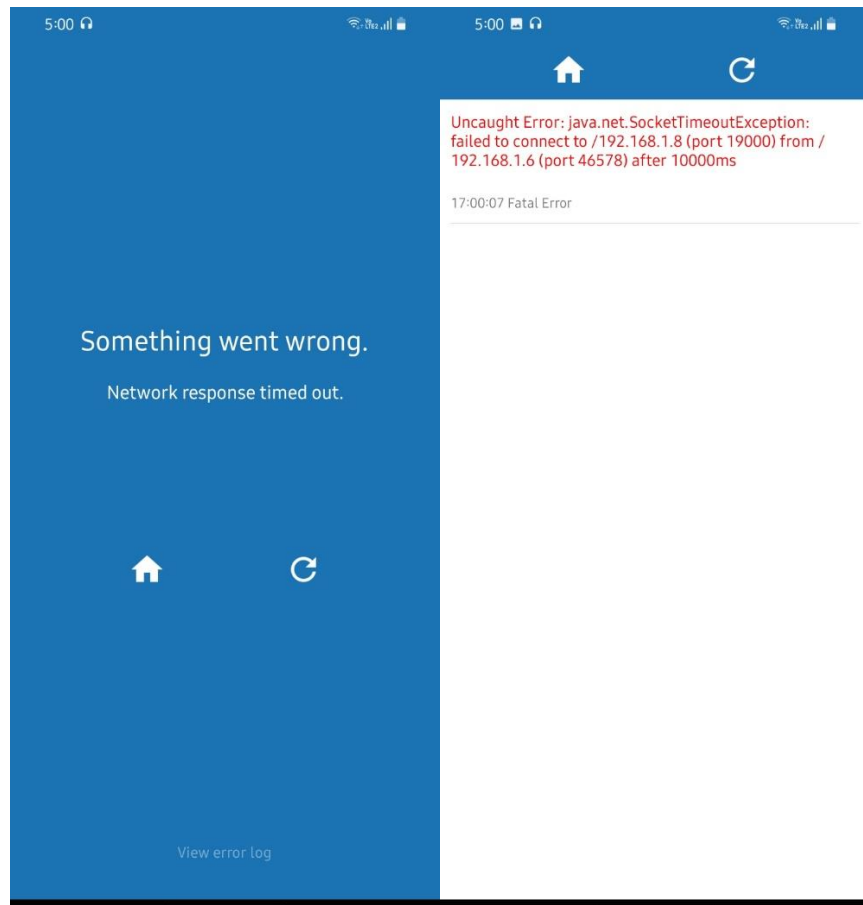
EJECUCIÓN MEDIANTE EXPO GO

```
Starting project at C:\Users\stive\Documents\PROYECTMOVIL\TheMovieApp
Some dependencies are incompatible with the installed expo version:
  react-native@0.71.3 - expected version: 0.71.8
  react-native-gesture-handler@2.10.1 - expected version: ~2.9.0
  react-native-reanimated@2.17.0 - expected version: ~2.14.4
  react-native-safe-area-context@4.5.3 - expected version: 4.5.0
Your project may not work correctly until you install the correct versions of the packages.
Install individual packages by running npx expo install react-native@0.71.8 react-native-gesture-handler@~2.9.0 react-native-reanimated@~2.14.4 react-native-safe-area-context@4.5.0
Starting Metro Bundler

> Metro waiting on exp://192.168.1.8:19000
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Press a | open Android
> Press w | open web
```

EJECUCIÓN EXPO GO: (05:00 PM)



EJECUCIÓN EXPO GO: (06:10 PM)

