

TRABAJO FINAL

Desarrollo móvil

Juan Esteban Tobón Duque

2023

EVIDENCIAS CÓDIGO

COMPONENTES:

CARD:

```
1  import React from 'react';
2  import { TouchableOpacity, StyleSheet, Image, Text } from 'react-native';
3
4  const placeholderImage = require('../assets/images/placeholder.png');
5
6  const Card = ({ navigation, item }) => {
7    return (
8      <TouchableOpacity
9        onPress={() => navigation.navigate('Detail', { movieId: item.id })}
10        style={styles.container}
11      >
12        <Image
13          resizeMode="cover"
14          style={styles.image}
15          source={
16            item.poster_path
17            ? { uri: 'https://image.tmdb.org/t/p/w500' + item.poster_path }
18            : placeholderImage
19          }
20        />
21        {!item.poster_path && <Text style={styles.movieName}>{item.title}</Text>}
22      </TouchableOpacity>
23    );
24  };
25
26  const styles = StyleSheet.create({
27    container: {
28      padding: 5,
29      position: 'relative',
30      alignItems: 'center',
31      height: 200,
32      marginBottom: 8,
33      backgroundColor: '#333',
34      borderRadius: 20,
35    },
36    image: {
37      height: 200,
38      width: 120,
39      borderRadius: 20,
40    },
41    movieName: {
42      position: 'absolute',
43      width: 100,
44      top: 10,
45      textAlign: 'center',
46      color: 'fff',
47    },
48  });
49
50  export default Card;
```

ERROR:

```
1  import React from 'react';
2  import { View, Text, StyleSheet } from 'react-native';
3  import PropTypes from 'prop-types';
4  import Colors from '../theme/Colors';
5
6  const propTypes = {
7    | errorText1: PropTypes.string,
8    | errorText2: PropTypes.string,
9  };
10
11  const defaultProps = {
12    | errorText1: 'Wow, parece que ocurrió un problema',
13    | errorText2: 'Verifique su conexión a internet o reinicie la app',
14  };
15
16  const Error = ({ errorText1, errorText2 }) => {
17    | return (
18    |   <View style={styles.container}>
19    |     <Text style={styles.text}>{errorText1}</Text>
20    |     <Text style={styles.text}>{errorText2}</Text>
21    |   </View>
22    | );
23  };
24
25  const styles = StyleSheet.create({
26    | container: {
27    |   flex: 1,
28    |   justifyContent: 'center',
29    |   alignItems: 'center',
30    |   backgroundColor: '#121212',
31    | },
32    | text: {
33    |   fontWeight: 'bold',
34    |   color: 'ffffff',
35    | },
36  });
37
38  Error.propTypes = propTypes;
39  Error.defaultProps = defaultProps;
40
41  export default Error;
```

LIST:

```
1 import * as React from 'react';
2 import {Text, View, StyleSheet, FlatList} from 'react-native';
3 import Card from './Card';
4 import PropTypes from 'prop-types';
5
6 const propTypes = {
7   title: PropTypes.string,
8   content: PropTypes.array,
9 };
10
11 class List 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               <Card navigation={navigation} item={item} />
25             )}
26           />
27         </View>
28       </View>
29     );
30   }
31 }
32
33 const styles = StyleSheet.create({
34   list: {
35     marginTop: 25,
36   },
37   text: {
38     fontSize: 20,
39     fontWeight: 'bold',
40     padding: 10,
41     paddingBottom: 15,
42   },
43 });
44
45 List.propTypes = propTypes;
46 export default List;
```

MAINNAVIGATION:

```
1  import * as React from 'react';
2  import {createStackNavigator} from '@react-navigation/stack';
3  import Home from '../screens/Home';
4  import Detail from '../screens/Detail';
5  import Navbar from './Navbar';
6  import Search from '../screens/Search';
7
8  const Stack = createStackNavigator();
9
10 class MainNavigation extends React.PureComponent {
11   render() {
12     return (
13       <Stack.Navigator headerMode='screen'>
14         <Stack.Screen
15           name="Home"
16           component={Home}
17           options={{
18             headerTransparent: true,
19             header: ({navigation}) => (
20               <Navbar navigation={navigation} main={true} />
21             ),
22           }}
23         />
24         <Stack.Screen
25           name="Detail"
26           component={Detail}
27           options={{
28             headerTransparent: true,
29             header: ({navigation}) => (
30               <Navbar main={false} navigation={navigation} />
31             ),
32           }}
33         />
34         <Stack.Screen
35           name="Search"
36           component={Search}
37           options={{
38             headerTransparent: true,
39             header: ({navigation}) => (
40               <Navbar main={false} navigation={navigation} />
41             ),
42           }}
43         />
44       </Stack.Navigator>
45     );
46   }
47 }
48
49 export default MainNavigation;
```

NAVBAR:

```
1 import 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
6 const propTypes = {
7   main: PropTypes.bool,
8 };
9
10 const defaultProps = {
11   main: false,
12 };
13
14 const Navbar = ({ navigation, main }) => {
15   return (
16     <SafeAreaView style={styles.container}>
17       {main ? (
18         <View style={styles.mainNav}>
19           <Image
20             style={styles.logo}
21             source={require('../assets/images/cine.png')}
22           />
23           <TouchableOpacity
24             onPress={() => {
25               navigation.navigate('Search');
26             }}
27             style={styles.searchButton}
28           >
29             <Icon name='search-outline' size={24} color="#ffffff" />
30           </TouchableOpacity>
31         </View>
32       ) : (
33         <View style={styles.secondaryNav}>
34           <TouchableOpacity
35             onPress={() => {
36               navigation.goBack();
37             }}
38             style={styles.backButton}
39           >
40             <Icon name='chevron-back' size={24} color="#cccccc" />
41           </TouchableOpacity>
42         </View>
43       )}
44     </SafeAreaView>
45   );
46 };
47
```

```
const styles = StyleSheet.create({
  container: {
    backgroundColor: '#000000',
  },
  mainNav: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    paddingVertical: 8,
    paddingHorizontal: 12,
  },
  secondaryNav: {
    flexDirection: 'row',
    alignItems: 'center',
    paddingVertical: 8,
    paddingHorizontal: 12,
  },
  logo: {
    width: 40,
    height: 40,
    marginRight: 6,
  },
  searchButton: {
    padding: 6,
    borderRadius: 16,
    backgroundColor: '#888888',
  },
  backButton: {
    padding: 4,
    borderRadius: 16,
    backgroundColor: '#888888',
  },
});

Navbar.propTypes = propTypes;
Navbar.defaultProps = defaultProps;

export default Navbar;
```

PLAYBUTTON:

```
1  import React from 'react';
2  import { Pressable, StyleSheet } from 'react-native';
3  import Icon from 'react-native-vector-icons/Ionicons';
4  import Colors from '../theme/Colors';
5
6  const PlayButton = ({ handlePress }) => {
7    return (
8      <Pressable onPress={handlePress} style={styles.button}>
9        <Icon name="caret-forward-outline" size={30} color={Colors.white} />
10     </Pressable>
11   );
12 };
13
14 const styles = StyleSheet.create({
15   button: {
16     justifyContent: 'center',
17     alignItems: 'center',
18     borderRadius: 50,
19     width: 60,
20     height: 60,
21     backgroundColor: Colors.primary,
22     elevation: 5,
23   },
24 });
25
26 export default PlayButton;
```

SCREENS:

DETAIL:

```
1 import React, { useState, useEffect } from 'react';
2 import { StyleSheet, ScrollView, Image, Dimensions, ActivityIndicator, Text, View, Modal } from 'react-native';
3 import PlayButton from '../components/PlayButton';
4 import StarRating from 'react-native-star-rating';
5 import { getMovie } from '../services/services';
6 import dateFormat from 'dateformat';
7 import VideoPlayer from 'react-native-video';
8
9 const placeholderImage = require('../assets/images/placeholder.png');
10 const height = Dimensions.get('screen').height;
11
12 const Detail = ({ route, navigation }) => {
13   const movieId = route.params.movieId;
14
15   const [movieDetail, setMovieDetail] = useState(null);
16   const [loaded, setLoaded] = useState(false);
17   const [error, setError] = useState(false);
18   const [modalVisible, setModalVisible] = useState(false);
19
20   useEffect(() => {
21     getMovie(movieId)
22       .then((movieData) => {
23         setMovieDetail(movieData);
24         setLoaded(true);
25       })
26       .catch((error) => {
27         console.log('Error al obtener los detalles de la película:', error);
28         setError(true);
29         setLoaded(true);
30       });
31   }, [movieId]);
32
33   const toggleModal = () => {
34     setModalVisible(!modalVisible);
35   };
36
37   return (
38     <React.Fragment>
39       {loaded && !error && (
40         <View style={styles.container}>
41           <ScrollView>
42             <Image
43               resizeMode="cover"
44               style={styles.image}
45               source={
46                 movieDetail.poster_path
47                 ? { uri: 'https://image.tmdb.org/t/p/w500' + movieDetail.poster_path }
48                 : placeholderImage
49               }
50             />

```

```
51       <View style={styles.content}>
52         <View style={styles.playButton}>
53           <PlayButton handlePress={toggleModal} />
54         </View>
55         <Text style={styles.movieTitle}>{movieDetail.title}</Text>
56         {movieDetail.genres && (
57           <View style={styles.genresContainer}>
58             {movieDetail.genres.map((genre) => (
59               <Text style={styles.genre} key={genre.id}>
60                 {genre.name}
61               </Text>
62             ))}
63           </View>
64         )}
65         <StarRating
66           disabled={true}
67           maxStars={5}
68           starSize={30}
69           rating={movieDetail.vote_average / 2}
70           fullStarColor="gold"
71         />
72         <Text style={styles.overview}>{movieDetail.overview}</Text>
73         <Text style={styles.release}>
74           { 'Release date: ' + dateFormat(movieDetail.release_date, 'mmm dS, yyyy') }
75         </Text>
76       </View>
77     </ScrollView>
78     <Modal
79       supportedOrientations={['portrait', 'landscape']}
80       animationType="slide"
81       visible={modalVisible}
82       onRequestClose={toggleModal}
83     >
84       <View style={styles.videoModal}>
85         <VideoPlayer
86           source={{ uri: 'https://example.com/video.mp4' }} // Reemplaza con la URL de tu video
87           style={styles.videoPlayer}
88           controls
89           resizeMode="contain"
90         />
91       </View>
92     </Modal>
93   </View>
94 )}
```



```

100     {loaded && !error && <ActivityIndicator size="large" />}
101   </React.Fragment>
102 );
103 };
104
105 const styles = StyleSheet.create({
106   container: {
107     flex: 1,
108     backgroundColor: '#111',
109   },
110   content: {
111     padding: 20,
112   },
113   image: {
114     height: height / 1.4,
115   },
116   movieTitle: {
117     fontSize: 24,
118     fontWeight: 'bold',
119     color: 'fff',
120     marginTop: 10,
121     marginBottom: 10,
122   },
123   genresContainer: {
124     flexDirection: 'row',
125     flexWrap: 'wrap',
126     marginTop: 20,
127     marginBottom: 20,
128   },
129   genre: {
130     marginRight: 10,
131     marginBottom: 5,
132     paddingHorizontal: 10,
133     paddingVertical: 5,
134     borderRadius: 20,
135     backgroundColor: '#555',
136     color: 'fff',
137   },
138   overview: {
139     padding: 15,
140     color: 'fff',
141   },
142   release: {
143     fontWeight: 'bold',
144     color: 'fff',
145   },

```

```

146   playButton: {
147     position: 'absolute',
148     top: -25,
149     right: 20,
150   },
151   videoModal: {
152     flex: 1,
153     justifyContent: 'center',
154     alignItems: 'center',
155     backgroundColor: '#000',
156   },
157   videoPlayer: {
158     flex: 1,
159     width: '100%',
160     height: '100%',
161   },
162   errorContainer: {
163     flex: 1,
164     justifyContent: 'center',
165     alignItems: 'center',
166   },
167   errorText: {
168     fontSize: 18,
169     color: 'red',
170     textAlign: 'center',
171   },
172 });
173
174 export default Detail;

```

HOME:

```
1 import React, { useEffect, useState } from 'react';
2 import { ActivityIndicator, View, StyleSheet, Dimensions, ScrollView, Text } from 'react-native';
3 import { SliderBox } from 'react-native-image-slider-box';
4 import { getTop_ratedTv, getAiring_todayTv, getNow_playingMovies, getPopularMovies, getUpcomingMovies, getPopularTv, getFamilyMovies, getDocumentaryMovies } from '../service';
5 import List from '../components/List';
6 import Error from '../components/Error';
7
8 const dimensions = Dimensions.get('screen');
9
10 const Home = ({ navigation }) => {
11   const [moviesImages, setMoviesImages] = useState([]);
12   const [Top_ratedTv, setTop_ratedTv] = useState([]);
13   const [Now_playingMovies, setNow_playingMovies] = useState([]);
14   const [popularMovies, setPopularMovies] = useState([]);
15   const [airing_today, setAiring_todayTv] = useState([]);
16   const [popularTv, setPopularTv] = useState([]);
17   const [familyMovies, setFamilyMovies] = useState([]);
18   const [documentaryMovies, setDocumentaryMovies] = useState([]);
19   const [error, setError] = useState(false);
20   const [loaded, setLoaded] = useState(false);
21
22   const getData = async () => {
23     try {
24       const [
25         Top_ratedTvData,
26         airing_todayData,
27         Now_playingMoviesData,
28         upcomingMoviesData,
29         popularMoviesData,
30         popularTvData,
31         familyMoviesData,
32         documentaryMoviesData,
33       ] = await Promise.all([
34         getTop_ratedTv(),
35         getAiring_todayTv(),
36         getNow_playingMovies(),
37         getUpcomingMovies(),
38         getPopularMovies(),
39         getPopularTv(),
40         getFamilyMovies(),
41         getDocumentaryMovies(),
42         getNow_playingMovies(),
43       ]);
44
45       const moviesImagesArray = upcomingMoviesData.map((movie) => 'https://image.tmdb.org/t/p/w500' + movie.poster_path);
46       setTop_ratedTv(Top_ratedTvData);
47       setMoviesImages(moviesImagesArray);
48       setNow_playingMovies(Now_playingMoviesData);
49       setPopularMovies(popularMoviesData);
50       setAiring_todayTv(airing_todayData);
51       setPopularTv(popularTvData);
52       setDocumentaryMovies(documentaryMoviesData);
53     } catch (error) {
54       setError(true);
55       console.log('Error al obtener los datos:', error);
56     } finally {
57       setLoaded(true);
58     }
59   };
60
61   useEffect(() => {
62     getData();
63   }, []);
64
65   return (
66     <View style={styles.container}>
67       {loaded && !error ? (
68         <ScrollView contentContainerStyle={styles.scrollViewContent}>
69           {moviesImages.length > 0 && (
70             <View style={styles.sliderContainer}>
71               <SliderBox
72                 images={moviesImages}
73                 dotStyle={styles.sliderStyle}
74                 sliderBoxHeight={dimensions.height / 1.5}
75                 autoplay
76                 circleLoop
77               />
78             </View>
79           )}
80
81           <View style={styles.carousel}>
82             <Text style={styles.title}>Películas populares entre el público</Text>
83             <List navigation={navigation} content={popularMovies} />
84           </View>
85
86           <View style={styles.carousel}>
87             <Text style={styles.title}>Películas actualmente en los cines</Text>
88             <List navigation={navigation} content={Now_playingMovies} />
89           </View>
90
91           <View style={styles.carousel}>
92             <Text style={styles.title}>Películas familiares</Text>
93             <List navigation={navigation} content={familyMovies} />
94           </View>
95
96           <View style={styles.carousel}>
97             <Text style={styles.title}>Programas de TV mejor clasificados</Text>
98             <List navigation={navigation} content={Top_ratedTv} />
99           </View>
100
101           <View style={styles.carousel}>
102             <Text style={styles.title}>Programas de TV populares</Text>
103             <List navigation={navigation} content={popularTv} />
104           </View>
105         </ScrollView>
106       ) : (
107         <Error />
108       )
109     </View>
110   );
111 }
```

```

102     <View style={styles.carousel}>
103       <Text style={styles.title}>Programas de TV populares</Text>
104       <List navigation={navigation} content={popularTv} />
105     </View>
106
107     <View style={styles.carousel}>
108       <Text style={styles.title}>Programas de TV que se transmiten hoy</Text>
109       <List navigation={navigation} content={airing_today} />
110     </View>
111
112     <View style={styles.carousel}>
113       <Text style={styles.title}>Documentales</Text>
114       <List navigation={navigation} content={documentaryMovies} />
115     </View>
116   </ScrollView>
117 ) : (
118   <View style={styles.loadingContainer}>
119     {loaded ? <Error /> : <ActivityIndicator size="large" />}
120   </View>
121 )}
122 </View>
123 );
124 };
125
126 const styles = StyleSheet.create({
127   container: {
128     flex: 1,
129     backgroundColor: '#111',
130   },
131   scrollViewContent: {
132     flexGrow: 1,
133     paddingTop: 70, // Ajustar el espacio entre las imágenes y las letras
134     paddingBottom: 40,
135   },
136   loadingContainer: {
137     flex: 1,
138     justifyContent: 'center',
139     alignItems: 'center',
140   },
141   sliderContainer: {
142     height: dimensions.height / 1.4,
143   },
144   sliderStyle: {
145     height: 0,
146   },
147   carousel: {
148     marginVertical: 25, // Ajustar el espacio entre las secciones
149     paddingHorizontal: 10,

```

```

144   sliderStyle: {
145     height: 0,
146   },
147   carousel: {
148     marginVertical: 25, // Ajustar el espacio entre las secciones
149     paddingHorizontal: 10,
150   },
151   title: {
152     color: '#fff',
153     fontSize: 20, // Ajustar el tamaño de las letras
154     fontWeight: 'bold',
155     marginBottom: 10,
156   },
157 });
158
159 export default Home;

```

SEARCH:

```
1 import React, { useState } from 'react';
2 import {
3   SafeAreaView,
4   View,
5   TextInput,
6   StyleSheet,
7   TouchableOpacity,
8   FlatList,
9   Text,
10 } from 'react-native';
11 import Icon from 'react-native-vector-icons/Ionicons';
12 import { searchMovieTv } from '../services/services';
13 import Card from '../components/Card';
14 import Error from '../components/Error';
15
16 const Search = ({ navigation }) => {
17   const [text, onChangeText] = useState('');
18   const [searchResults, setSearchResults] = useState([]);
19   const [error, setError] = useState(false);
20
21   const onSubmit = (query) => {
22     Promise.all([searchMovieTv(query, 'movie'), searchMovieTv(query, 'tv')])
23       .then(([movies, tv]) => {
24         const data = [...movies, ...tv];
25         setSearchResults(data);
26       })
27       .catch(() => {
28         setError(true);
29       });
30   };
31
32   return (
33     <SafeAreaView style={styles.container}>
34       <TouchableOpacity
35         onPress={() => navigation.goBack()}
36         style={styles.backButton}>
37         <Icon name="chevron-back-outline" size={24} color="#fff" />
38       </TouchableOpacity>
39
40       <View style={styles.searchContainer}>
41         <TextInput
42           style={styles.input}
43           placeholder="Busca una película o un show de TV"
44           onChangeText={onChangeText}
45           value={text}
46         />
47         <TouchableOpacity onPress={() => onSubmit(text)} style={styles.searchButton}>
48           <Icon name="search-outline" size={24} color="#fff" />
49         </TouchableOpacity>
50       </View>
51     </SafeAreaView>
52   );
53 }
```

```

53     <View style={styles.searchItems}>
54       {/* Searched items results */}
55       {searchResults.length > 0 ? (
56         <FlatList
57           numColumns={3}
58           data={searchResults}
59           renderItem={({ item }) => <Card navigation={navigation} item={item} />}
60           keyExtractor={(item) => item.id.toString()}
61         />
62       ) : searchResults.length === 0 ? (
63         <View style={styles.noResults}>
64           <Text style={styles.noResultsText}>No se encuentran resultados bajo ese criterio.</Text>
65           <Text style={styles.noResultsText}>Prueba con otras palabras.</Text>
66         </View>
67       ) : (
68         <View style={styles.empty}>
69           <Text>Escribe algo para iniciar tu búsqueda</Text>
70         </View>
71       )}
72
73       {/* Error */}
74       {error && <Error />}
75     </View>
76   </SafeAreaView>
77 );
78 };
79
80 const styles = StyleSheet.create({
81   container: {
82     flex: 1,
83     backgroundColor: '#000',
84   },
85   backButton: {
86     paddingHorizontal: 10,
87     paddingVertical: 15,
88   },
89   searchContainer: {
90     flexDirection: 'row',
91     alignItems: 'center',
92     padding: 10,
93     backgroundColor: '#000',
94     borderBottomWidth: 1,
95     borderBottomColor: 'ccc',
96     marginBottom: 20,
97   },
98   input: {
99     flex: 1,
100    height: 40,
101    borderRadius: 5,
102    paddingHorizontal: 10,
103    backgroundColor: 'fff',

```

```

107    searchButton: {
108      padding: 10,
109      borderRadius: 5,
110      backgroundColor: '#007bff',
111    },
112    searchItems: {
113      flex: 1,
114      padding: 10,
115    },
116    noResults: {
117      flex: 1,
118      justifyContent: 'center',
119      alignItems: 'center',
120      paddingVertical: 20,
121    },
122    noResultsText: {
123      marginBottom: 10,
124      textAlign: 'center',
125      color: 'fff',
126    },
127    empty: {
128      flex: 1,
129      justifyContent: 'center',
130      alignItems: 'center',
131    },
132  });
133
134  export default Search;

```

SERVICES:

```
1 import axios from 'axios';
2
3 const apiUrl = 'https://api.themoviedb.org/3';
4 const apiKey = 'api_key=0a31a928954da44adc698dcb8bfe3902';
5
6 // Obtener programas de tv mejor clasificados
7 export const getTop_ratedTv = async () => {
8   const resp = await axios.get(`${apiUrl}/tv/top_rated?${apiKey}`);
9   return resp.data.results;
10 };
11
12
13
14 // Obtener programas de tv que se emitirán hoy
15 export const getAiring_todayTv = async () => {
16   const resp = await axios.get(`${apiUrl}/tv/airing_today?${apiKey}`);
17   return resp.data.results;
18 };
19
20
21
22 // Obtener peliculas ahora mismo en cartelera
23 export const getNow_playingMovies = async () => {
24   const resp = await axios.get(`${apiUrl}/movie/now_playing?${apiKey}`);
25   return resp.data.results;
26 };
27
28 // Obtener peliculas populares
29 export const getPopularMovies = async () => {
30   const resp = await axios.get(`${apiUrl}/movie/popular?${apiKey}`);
31   return resp.data.results;
32 };
33
34 // Obtener próximas películas
35 export const getUpcomingMovies = async () => {
36   const resp = await axios.get(`${apiUrl}/movie/upcoming?${apiKey}`);
37   return resp.data.results;
38 };
39
40 // Obtener programas de tv populares
41 export const getPopularTv = async () => {
42   const resp = await axios.get(`${apiUrl}/tv/popular?${apiKey}`);
43   return resp.data.results;
44 };
45
46 // Obtener peliculas de familia
47 export const getFamilyMovies = async () => {
48   const resp = await axios.get(
49     `${apiUrl}/discover/movie?${apiKey}&with_genres=10751`,
50   );
51   return resp.data.results;
52 };
53
54 // Obtener peliculas de familia
55 export const getFamilyMovies = async () => {
56   const resp = await axios.get(
57     `${apiUrl}/discover/movie?${apiKey}&with_genres=10751`,
58   );
59   return resp.data.results;
60 };
61
62 // Obtener documentales
63 export const getDocumentaryMovies = async () => {
64   const resp = await axios.get(
65     `${apiUrl}/discover/movie?${apiKey}&with_genres=99`,
66   );
67   return resp.data.results;
68 };
69
70 export const getMovie = async id => {
71   const resp = await axios.get(`${apiUrl}/movie/${id}?${apiKey}`);
72   return resp.data;
73 };
74
75 // Búsqueda de películas o programas de televisión por palabra clave
76 export const searchMovieTv = async (query, type) => {
77   const resp = await axios.get(
78     `${apiUrl}/search/${type}?${apiKey}&query=${query}`,
79   );
80   return resp.data.results;
81 };
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

EVIDENCIAS FUNCIONAMIENTO:

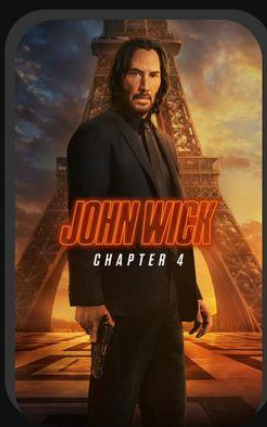
HOME:



99%



Películas populares entre el público



Películas actualmente en los cines



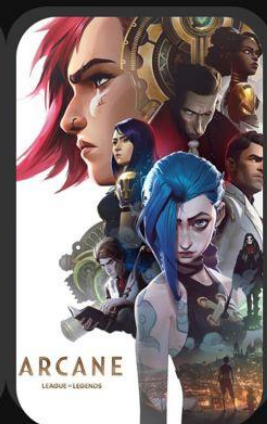
99%



Películas familiares



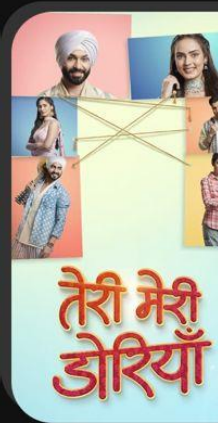
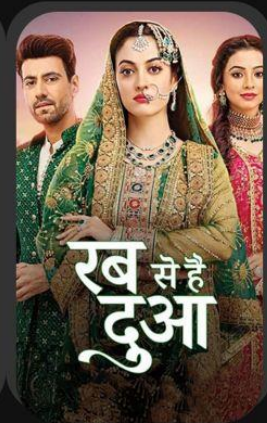
Programas de TV mejor clasificados



99%



Programas de TV populares



Programas de TV que se transmiten hoy



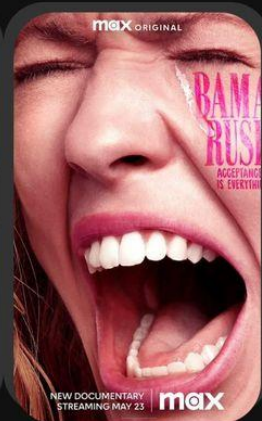
99%



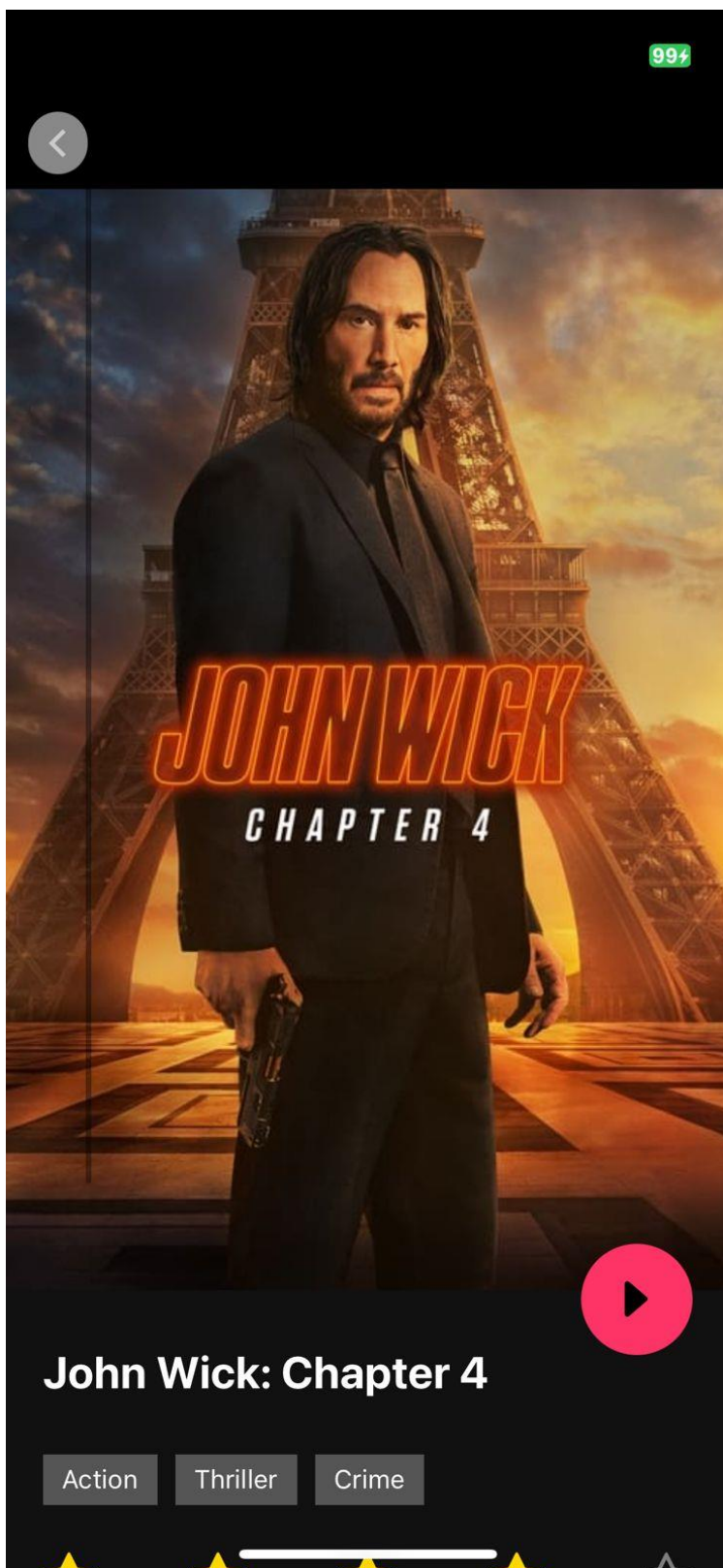
Programas de TV que se transmiten hoy



Documentales



DETALLES:



99%



John Wick: Chapter 4

Action

Thriller

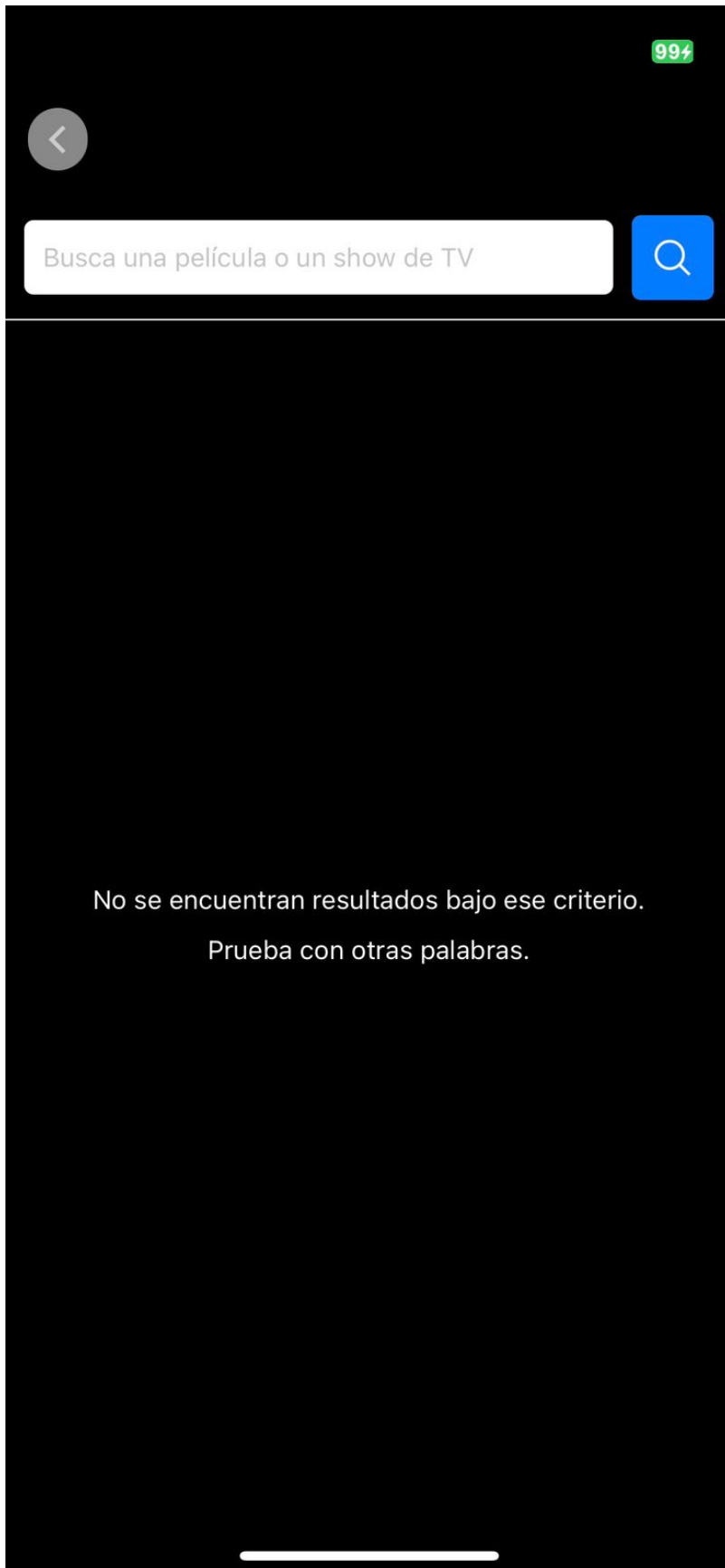
Crime



With the price on his head ever increasing, John Wick uncovers a path to defeating The High Table. But before he can earn his freedom, Wick must face off against a new enemy with powerful alliances across the globe and forces that turn old friends into foes.

Release date: March 21st, 2023

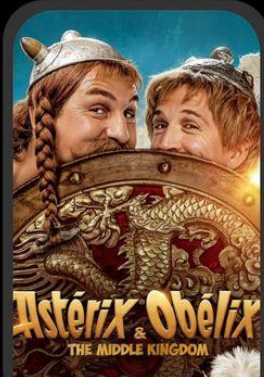
BUSQUEDA:



99%



A



99%



Ant-Man and the Wasp: Quantumania

Action

Adventure

Science Fiction



Super-Hero partners Scott Lang and Hope van Dyne, along with with Hope's parents, Janet van

ERROR:

