

Aplicación de Precios de Criptomonedas

Aplicación de Precios de Criptomonedas.....	1
Instalación de Expo	1
Recuperar datos api	5
Listado de componentes.....	7
Pasamos los datos coinItem.....	8
Agregamos imagenes	10
Le damos styles a la app.....	13
Formatos a los iconos.....	16
Filtrado del input.....	18
Código para refrescar app.....	21
Código completo de la app	23
Código App	23
Código de CoinItem.....	25

Aplicación de Precios de Criptomonedas

Aplicación native React Native & CoinGecko | Aplicación de Precios de Criptomonedas
<https://www.youtube.com/watch?v=k9ptn9zNHng&t=1337s>

Instalación de Expo

Es un marco y una plataforma para aplicaciones universales de React. Es un conjunto de herramientas y servicios construidos alrededor de React Native y plataformas nativas que lo ayudan a desarrollar, construir, implementar e iterar rápidamente en iOS, Android y aplicaciones web desde la misma base de código JavaScript / TypeScript.

Página con la documentación <https://docs.expo.io/> dentro de una consola en cmd escribimos

```
C:\Users\javeg>npm install --global expo-cli
```

```
npm install --global expo-cli
```

Solo es necesario instalarlo solo la primera vez.

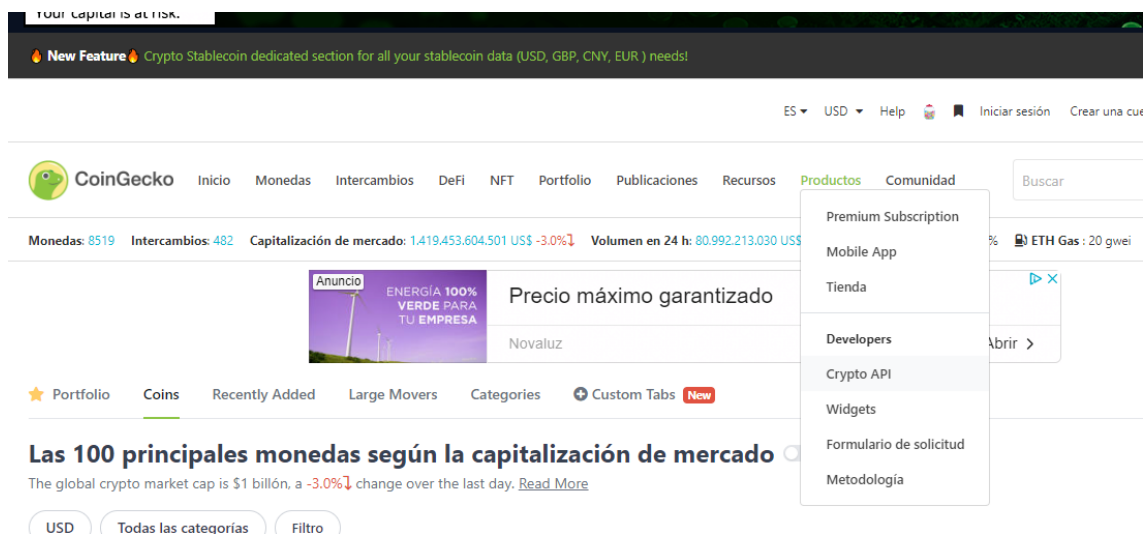
Creamos la aplicación dentro del escritorio

```
C:\Users\javeg\Desktop>expo init react-native-coingecko
```

Con unas preguntas en nuestro caso seleccionamos **black** un proyecto en blanco

```
C:\WINDOWS\system32\cmd.exe - "node" "C:\Users\javeg\AppData\Roaming\npm\node_modules\expo-cli\bin\expo.js" init react-native-coingecko
run `npm fund` for details
18 vulnerabilities (10 moderate, 8 high)
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
C:\Users\javeg>cd Desktop
C:\Users\javeg\Desktop>expo init react-native-coingecko
Choose a template: » - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
   blank (TypeScript) same as blank but with TypeScript configuration
   tabs (TypeScript) several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
   minimal bare and minimal, just the essentials to get you started
```

También vamos a necesitar una api en la página <https://www.coingecko.com/es>
seleccionamos Productos > Crypto API



Cuando estemos dentro seleccionamos /coins/markets y escribimos la moneda en nuestro caso usd

The screenshot shows the API documentation for the `/coins/markets` endpoint. It includes a 'GET' button and a description: 'List all supported coins price, market cap, volume, and market related data'. Below this, it says 'Use this to obtain all the coins market data (price, market cap, volume)'. The 'Parameters' section is a table with two columns: 'Name' and 'Description'. It lists three parameters: `vs_currency` (required, string, query), `ids` (string, query), and `category` (string, query). Each parameter has a corresponding input field. The `vs_currency` field contains 'usd'. The `ids` field contains 'ids'. The `category` field contains 'category'.

Name	Description
vs_currency * required string (query)	The target currency of market data (usd, eur, jpy, etc.) <input type="text" value="usd"/>
ids string (query)	The ids of the coin, comma separated cryptocurrency symbols (base). refers to <code>/coins/list</code> . When left empty, returns numbers the coins observing the params <code>limit</code> and <code>start</code> <input type="text" value="ids"/>
category string (query)	filter by coin category. Refer to <code>/coin/categories/list</code> <input type="text" value="category"/>

Pulsamos en ejecutar y copiamos la url

https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false

The screenshot shows a REST client interface. At the top, there are 'Execute' and 'Clear' buttons. Below them, the 'Responses' section shows the 'Response content type' as 'application/json'. The 'Curl' section shows the command: `curl -X 'GET' \ 'https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false' \ -H 'accept: application/json'`. The 'Request URL' section shows the URL: `https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false`. The 'Server response' section shows a 'Code' of 200 and a 'Response body' containing a JSON object: `{ "id": "bitcoin", "symbol": "btc", "name": "Bitcoin", ... }`.

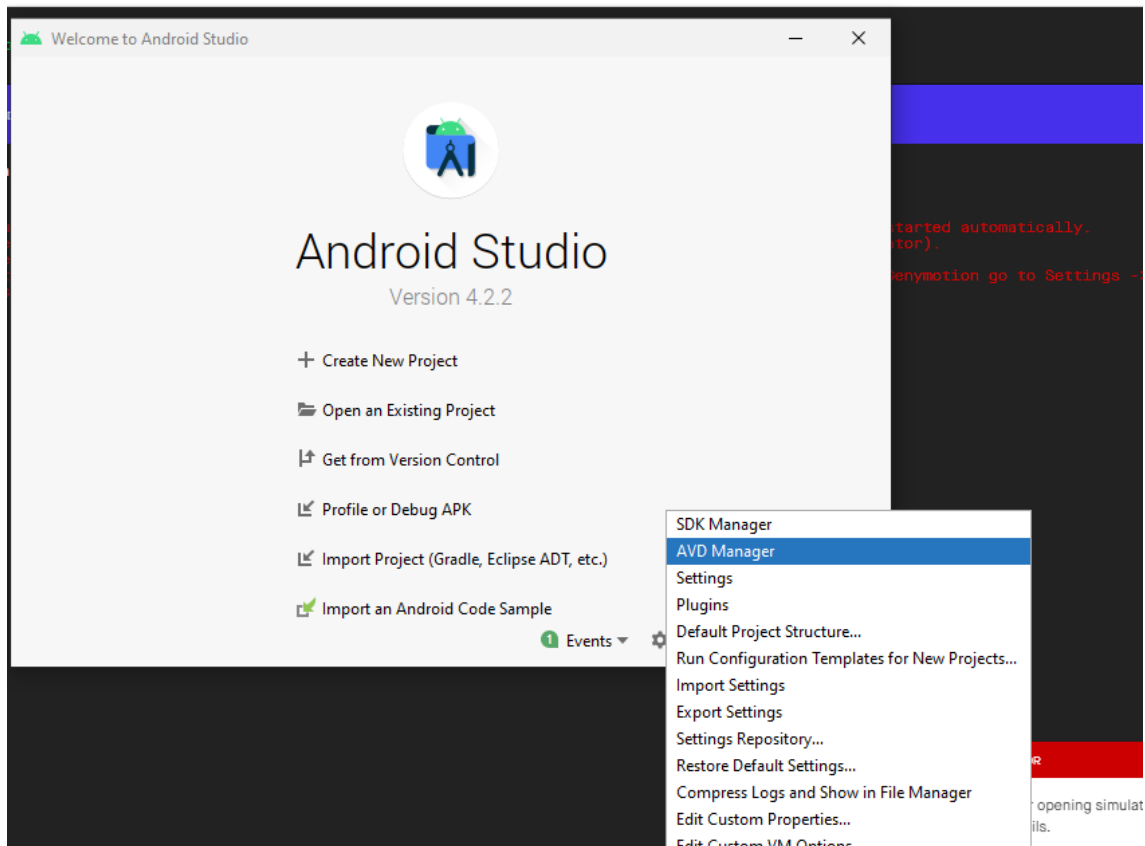
Vamos al cmd y buscamos el archivo creado y abrimos con code .

`C:\Users\javeg\Desktop>cd react-native-coingecko`

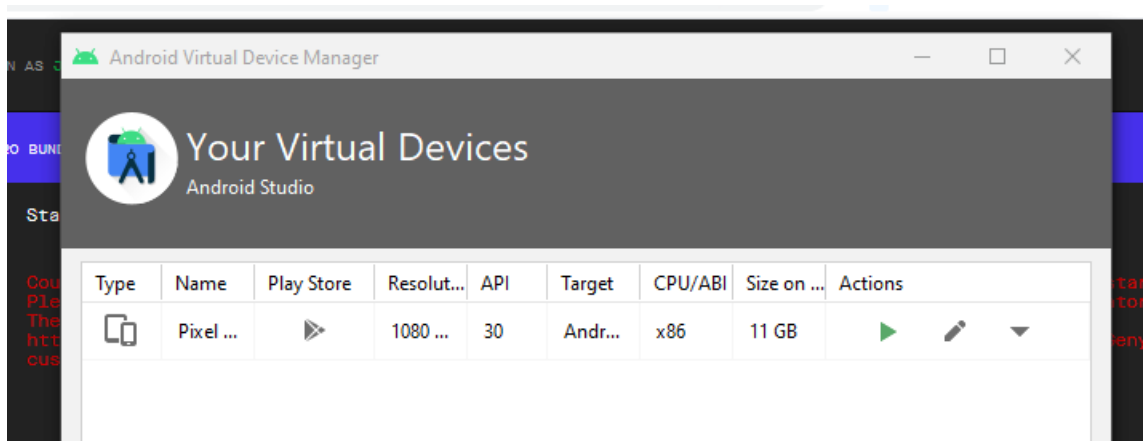
Una vez abierto visual studio code escribimos en el terminal

`PS C:\Users\javeg\Desktop\react-native-coingecko> expo start`

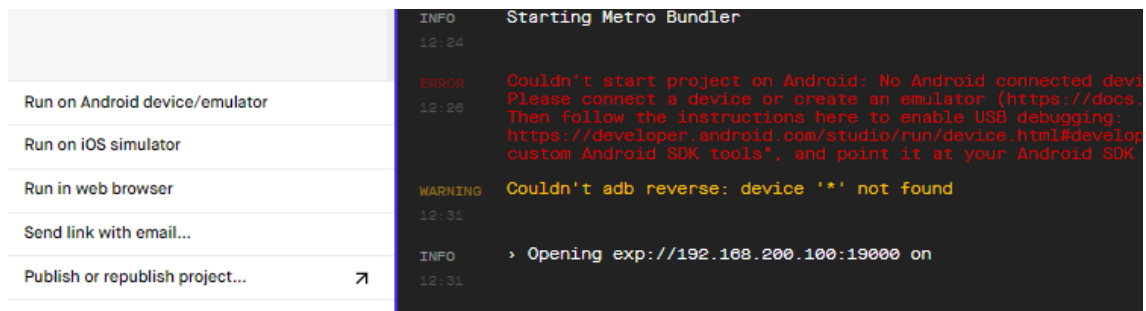
Tenemos que abrir un emulador en mi caso android studio. Seleccionamos configuración ->AVD Manager



Cuando esté abierto pulsamos play



Marcamos un run es nuestro caso el primero y esperamos que este en verde

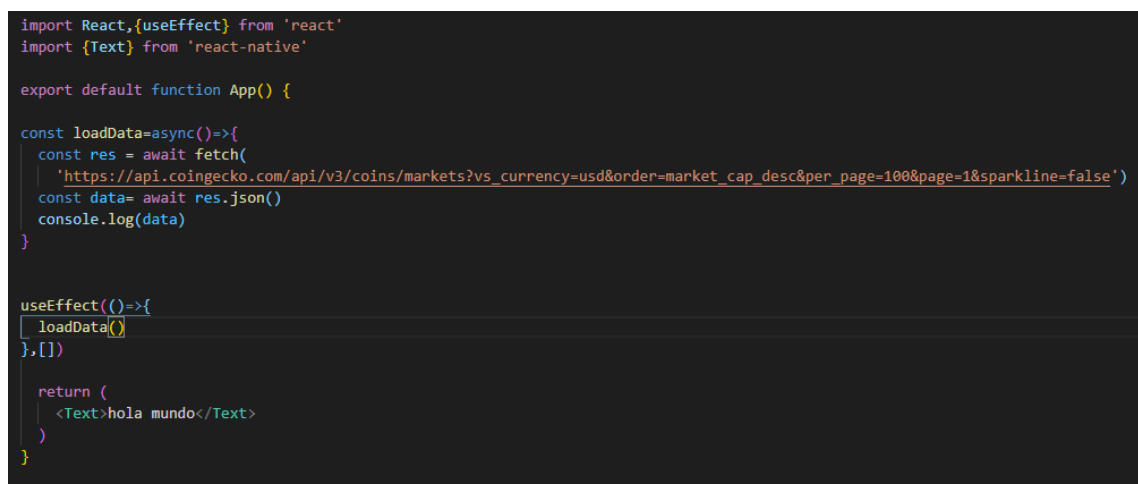


Cuando todo este correcto instalamos el plugging



Recuperar datos api

Para recuperar los datos de la API y mostrar por consola con fetch



Guardamos los datos de la API en un useState y lo mostramos por consola

```
import React,{useEffect, useState} from 'react'
import {Text} from 'react-native'

export default function App() {

  const [coins, setCoins]= useState([])

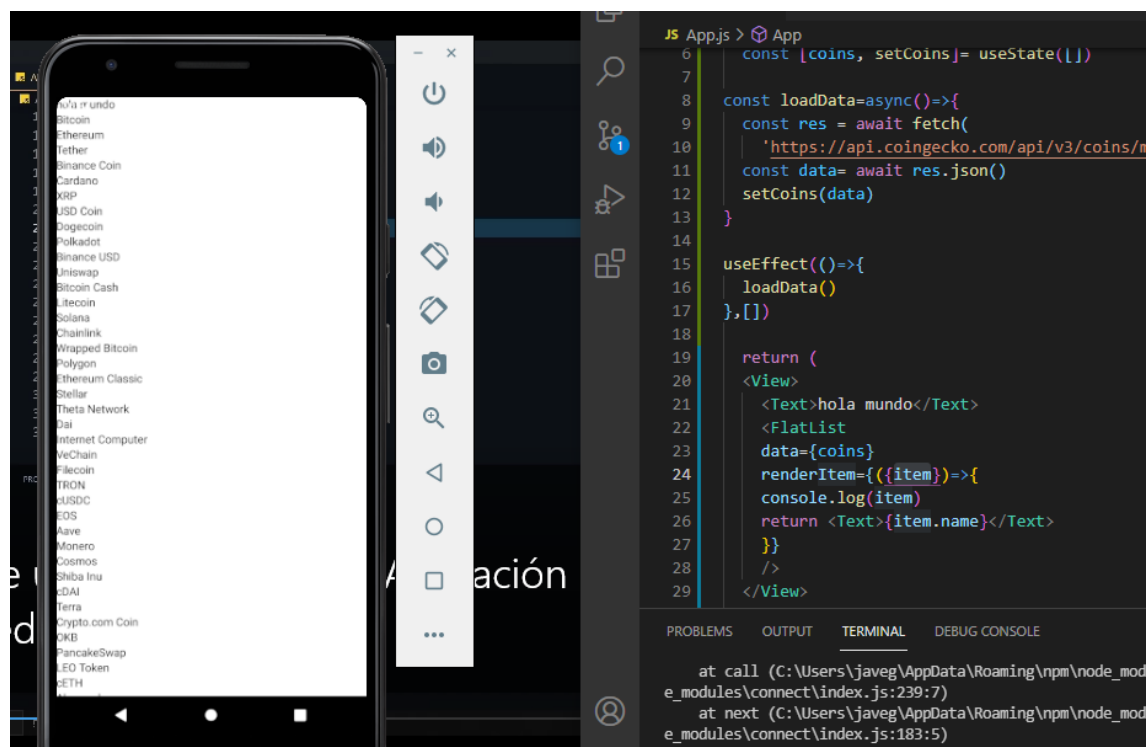
  const loadData=async()=>{
    const res = await fetch(
      'https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false')
    const data= await res.json()
    setCoins(data)
  }

  console.log(coins)

  useEffect(()=>{
    loadData()
  },[])

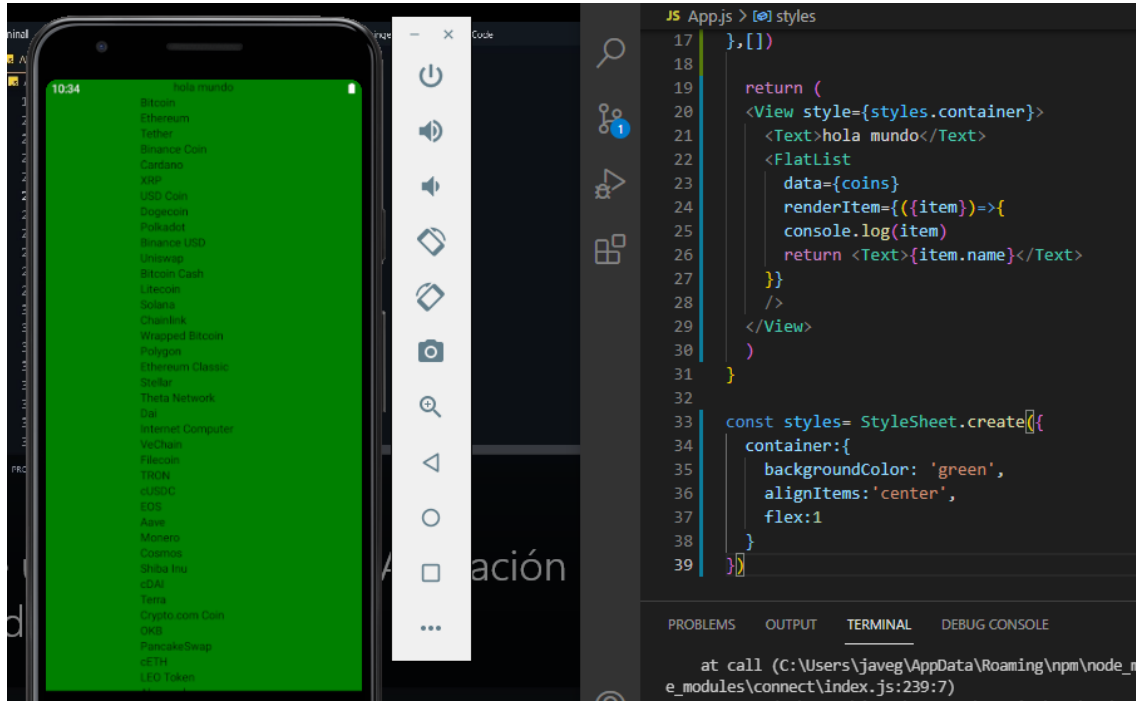
  return (
    <Text>hola mundo</Text>
  )
}
```

Con el FlatList podemos mostrar los datos del API

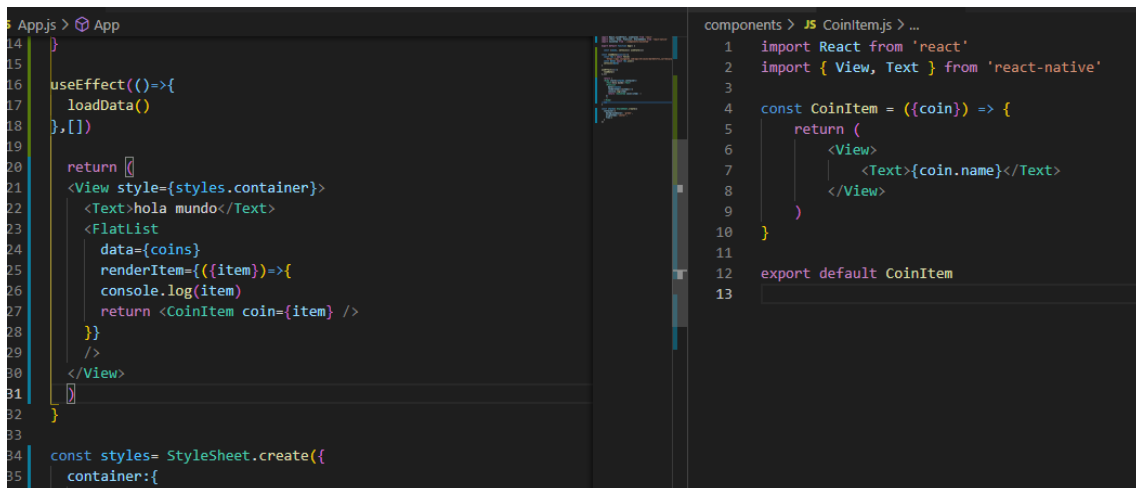


Listado de componentes

Le damos los primeros estilos con **StyleSheet**



Creamos un archivo nuevo con el nombre CoinItem.js dentro de la carpeta components
Y le pasamos los ítem en la variable coin



Pasamos los datos coinItem

Código App

```
import React, { useEffect, useState } from "react";
import { View, Text, FlatList, StyleSheet } from "react-native";
import CoinItem from "../components/CoinItem";
const App = () => {
  const [coins, setCoins] = useState([]);

  // console.log(coins)

  const loadData = async () => {
    const res = await fetch(
      "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false"
    );
    const data = await res.json();
    setCoins(data);
  };

  useEffect(() => {
    loadData();
  }, []);

  return (
    <View style={styles.container}>
      <Text>nuevo de hola</Text>
      <FlatList
        data={coins}
        renderItem={({ item }) => {
          return <CoinItem coin={item} />;
        }}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#141414",
    alignItems: "center",
    flex: 1,
  },
});

export default App;
```


Código CoinItem.js

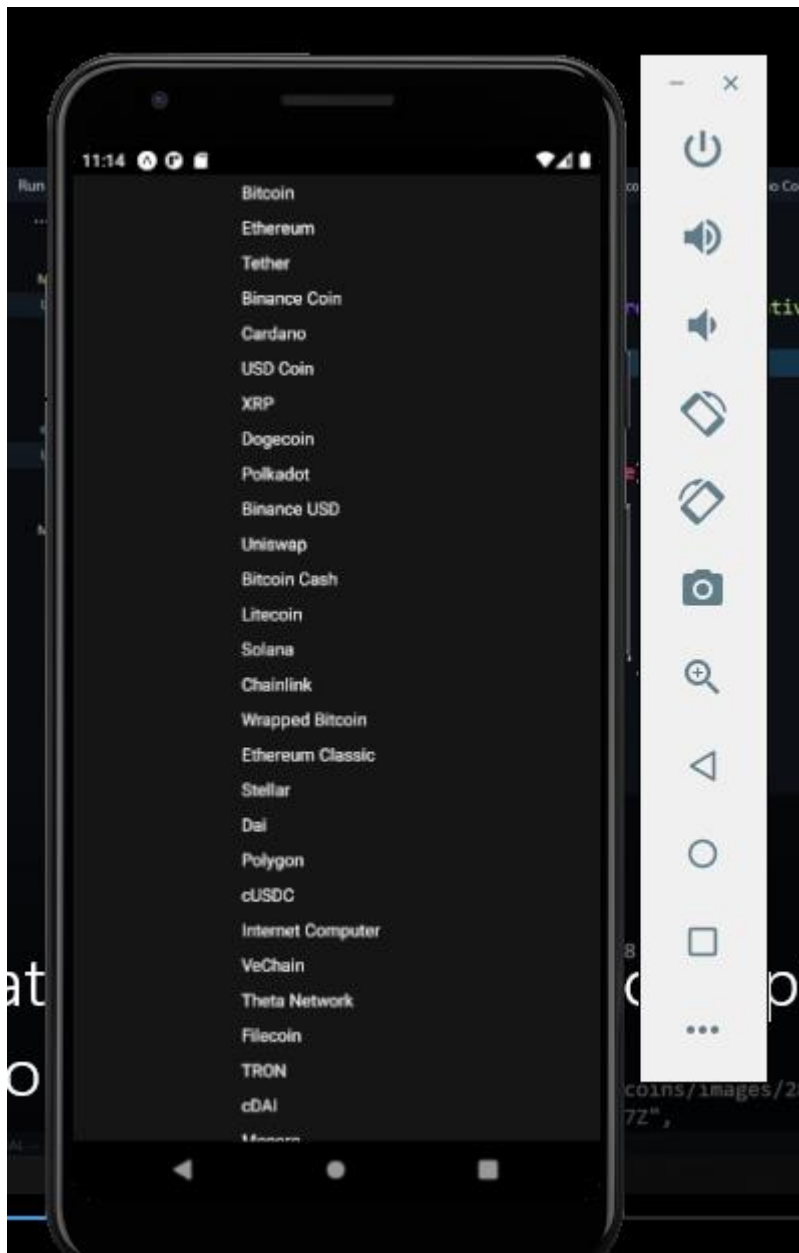
```
import React from "react";
import { View, Text, StyleSheet } from "react-native";

const CoinItem = ({ coin }) => {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>{coin.name}</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#121212",
    paddingTop: 10,
  },
  text: {
    color: "#fff",
  },
});

export default CoinItem;
```

Resultado:



Agregamos imagenes

Código CoinItem para dar formato y agregar imagenes

```
import React from "react";
import { View, Text, StyleSheet, Image } from "react-native";

const CoinItem = ({ coin }) => {
  return (
    <View style={styles.container}>
      <View style={styles.coinName}>
```

```

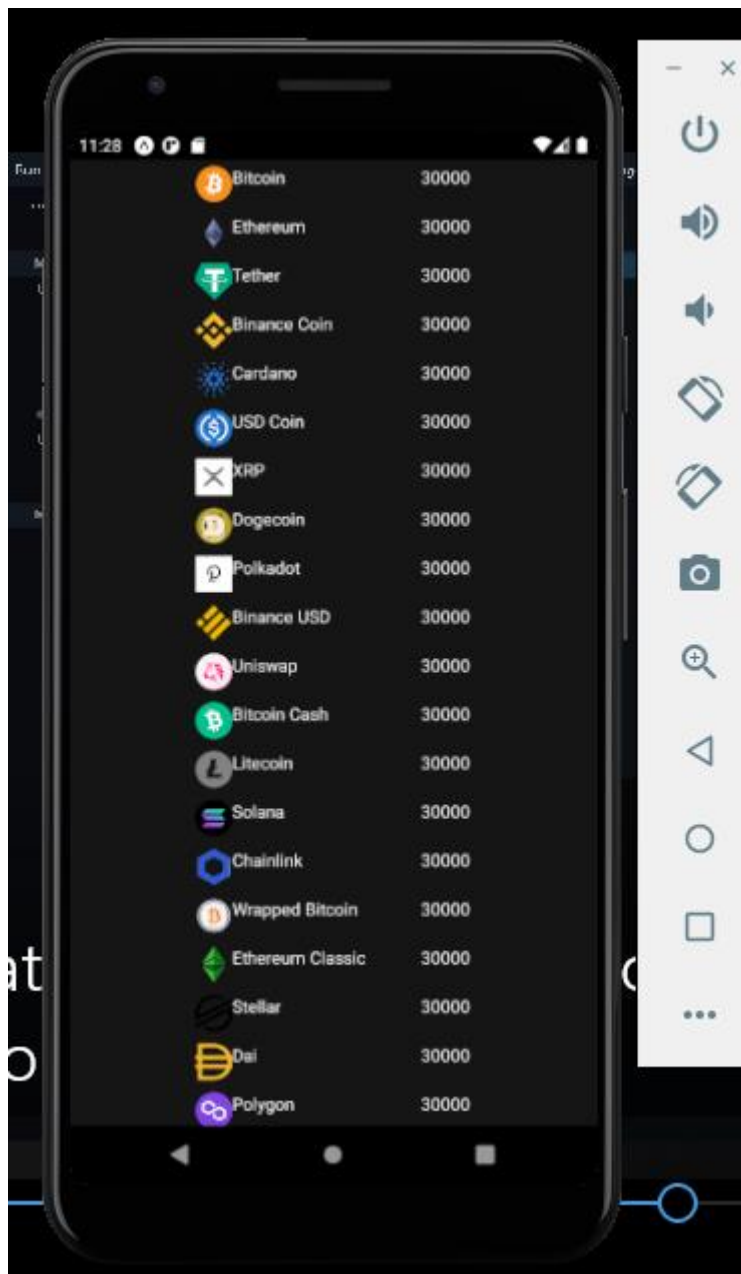
        <Image style={styles.image} source={{ uri: coin.image }} />
        <Text style={styles.text}>{coin.name}</Text>
      </View>
      <Text style={styles.text}>30000</Text>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#121212",
    paddingTop: 10,
    flexDirection: "row",
    justifyContent: "space-between",
  },
  coinName: {
    flexDirection: "row",
  },
  image: {
    width: 30,
    height: 30,
  },
  text: {
    color: "#fff",
  },
});

export default CoinItem;

```

resultado



Le damos estilos a la app

Código con formato App

```
import React, { useEffect, useState } from "react";
import {
  View as View,
  Text,
  FlatList,
  StyleSheet,
  TextInput,
  StatusBar,
} from "react-native";
import CoinItem from "../components/CoinItem";
const App = () => {
  const [coins, setCoins] = useState([]);

  // console.log(coins)

  const loadData = async () => {
    const res = await fetch(
      "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false"
    );
    const data = await res.json();
    setCoins(data);
  };

  useEffect(() => {
    loadData();
  }, []);

  return (
    <View style={styles.container}>
      <StatusBar backgroundColor="red" />
      <View style={styles.header}>
        <Text style={styles.title}>CryptoMarket</Text>
        <TextInput style={styles.searchInput} />
      </View>
      <FlatList
        style={styles.list}
        data={coins}
        renderItem={({ item }) => {
          return <CoinItem coin={item} />;
        }}
        showsVerticalScrollIndicator={false}
      />
    </View>
  );
};
```

```

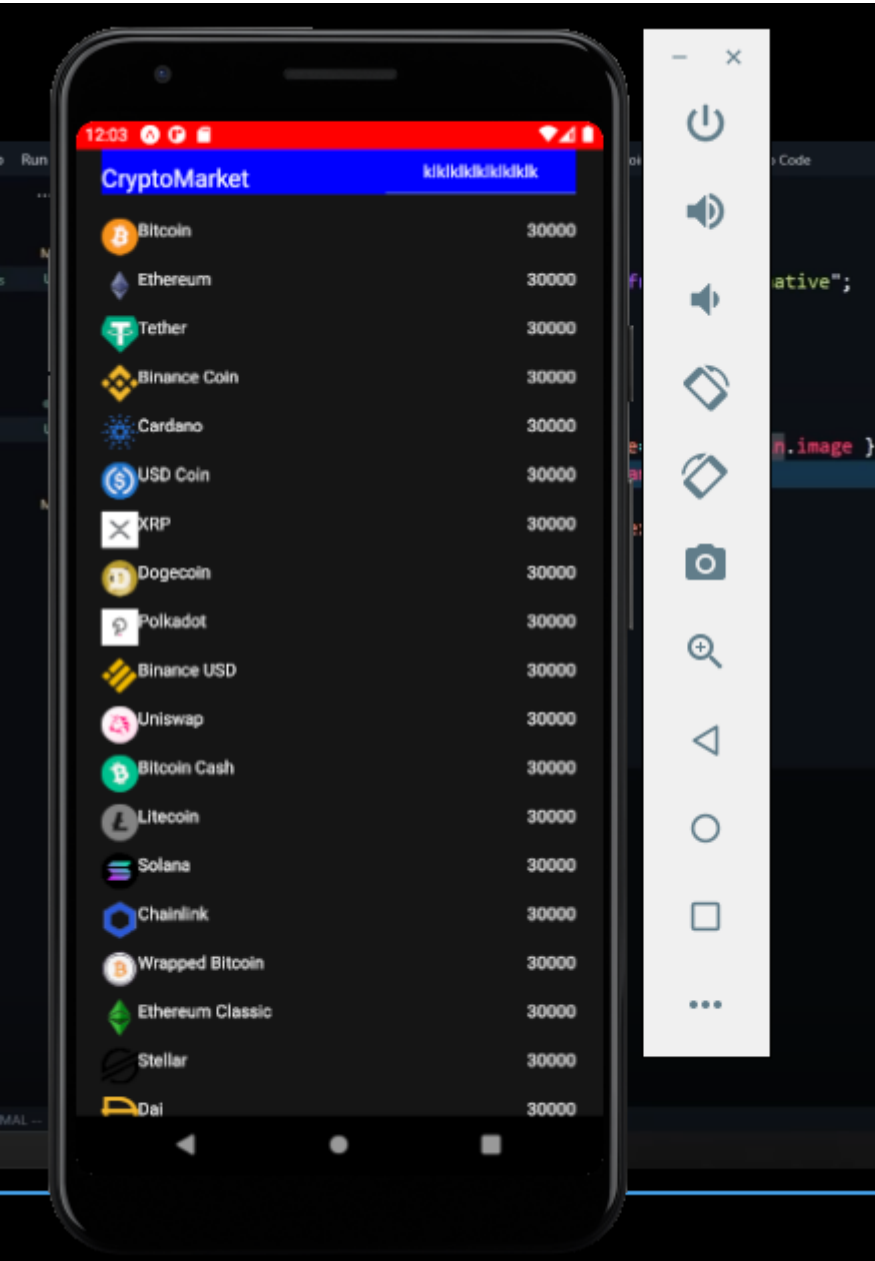
});

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#141414",
    alignItems: "center",
    flex: 1,
  },
  title: {
    color: "#fff",
    marginTop: 10,
    fontSize: 20,
  },
  list: {
    width: "90%",
  },
  header: {
    flexDirection: "row",
    backgroundColor: "blue",
    justifyContent: "space-between",
    width: "90%",
    marginBottom: 10,
  },
  searchInput: {
    color: "#fff",
    borderBottomWidth: 1,
    borderBottomColor: "#4657ce",
    width: "40%",
    textAlign: "center",
  },
});

export default App;

```

Resultado:



Formatos a los iconos

Añadimos a la derecha los iconos y le damos estilos

```
import React from "react";
import { View, Text, StyleSheet, Image } from "react-native";

const CoinItem = ({ coin }) => {
  return (
    <View style={styles.container}>
      <View style={styles.coinName}>
        <Image style={styles.image} source={{ uri: coin.image }} />

        <View style={styles.containerNames}>
          <Text style={styles.text}>{coin.name}</Text>
          <Text style={styles.textSymbol}>{coin.symbol}</Text>
        </View>
      </View>

      <View>
        <Text style={styles.textPrice}>$ {coin.current_price}</Text>
        <Text
          style={[
            styles.pricePercentage,
            coin.price_change_percentage_24h > 0
              ? styles.priceUp
              : styles.priceDown,
          ]>
          >
          {coin.price_change_percentage_24h}
        </Text>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#121212",
    paddingTop: 10,
    flexDirection: "row",
    justifyContent: "space-between",
  },
  containerNames: {
    marginLeft: 10,
  },
  coinName: {
    flexDirection: "row",
```



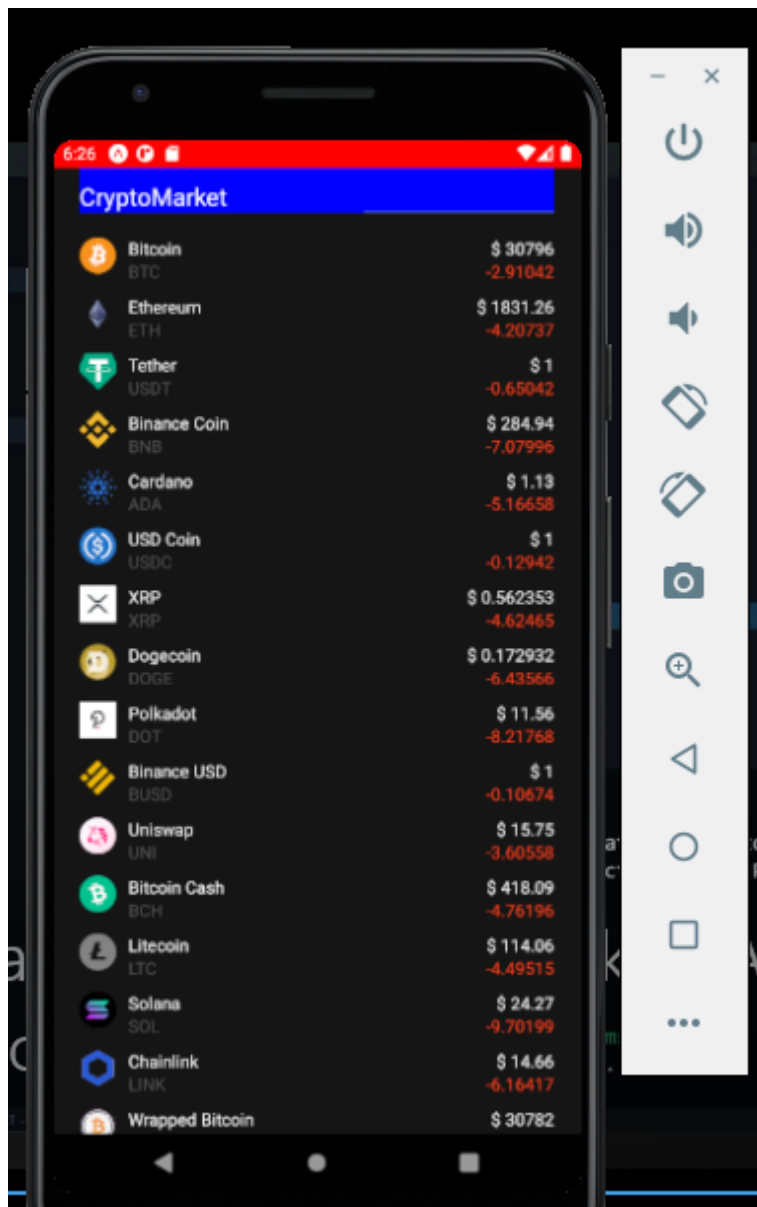
```

    },
    image: {
      width: 30,
      height: 30,
    },
    text: {
      color: "#fff",
    },
    textSymbol: {
      color: "#535353",
      textTransform: "uppercase",
    },
    pricePercentage: {
      textAlign: "right",
    },
    textPrice: {
      textAlign: "right",
      color: "#fff",
    },
    priceUp: {
      color: "#00b589",
    },
    priceDown: {
      color: "#fc4422",
    },
  });

export default CoinItem;

```

Resultado



Filtrado del input

Código para el filtrado del input

```
import React, { useEffect, useState } from "react";
import {
  View as View,
  Text,
  FlatList,
  StyleSheet,
  TextInput,
  StatusBar,
```

```

} from "react-native";
import CoinItem from "../components/CoinItem";
const App = () => {
  const [coins, setCoins] = useState([]);
  const [search, setSearch] = useState("");

  // console.log(coins)
  // console.log(search)

  const loadData = async () => {
    const res = await fetch(
      "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false"
    );
    const data = await res.json();
    setCoins(data);
  };

  useEffect(() => {
    loadData();
  }, []);

  return (
    <View style={styles.container}>
      <StatusBar backgroundColor="red" />
      <View style={styles.header}>
        <Text style={styles.title}>CryptoMarket</Text>
        <TextInput
          style={styles.searchInput}
          placeholder="Search a Coin"
          placeholderTextColor="#858585"
          onChangeText={(text) => setSearch(text)}
        />
      </View>
      <FlatList
        style={styles.list}
        data={coins.filter(
          (coin) =>
            coin.name.toLowerCase().includes(search) ||
            coin.symbol.toLowerCase().includes(search)
        )}
        renderItem={({ item }) => {
          return <CoinItem coin={item} />;
        }}
        showsVerticalScrollIndicator={false}
      />
    </View>
  );
};

```

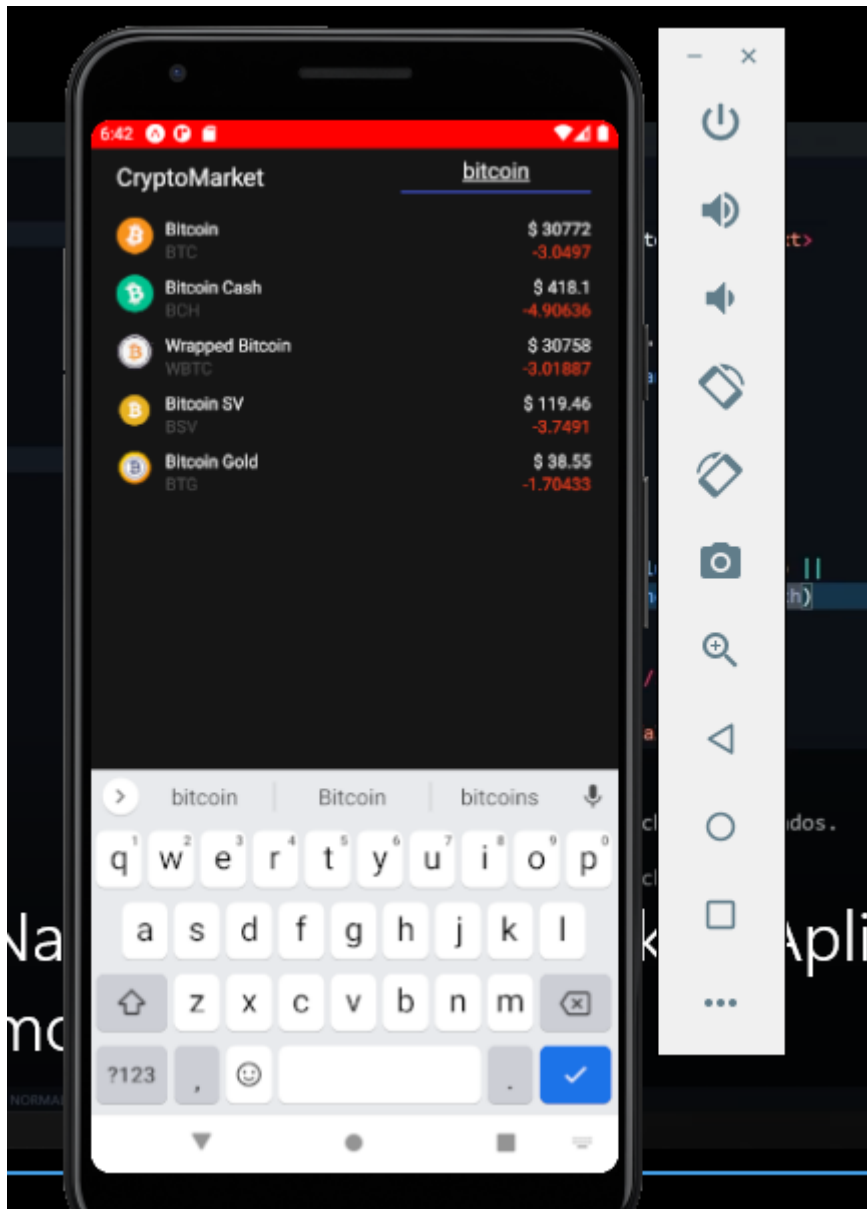
```

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#141414",
    alignItems: "center",
    flex: 1,
  },
  title: {
    color: "#fff",
    marginTop: 10,
    fontSize: 20,
  },
  list: {
    width: "90%",
  },
  header: {
    flexDirection: "row",
    // backgroundColor: "blue",
    justifyContent: "space-between",
    width: "90%",
    marginBottom: 10,
  },
  searchInput: {
    color: "#fff",
    borderBottomWidth: 1,
    borderBottomColor: "#4657ce",
    width: "40%",
    textAlign: "center",
    fontSize: 18,
  },
});

export default App;

```

Resultado filtrado por bitcoin



Código para refrescar app

Refrescado de la pantalla con onRefresh

```
const [coins, setCoins] = useState([]);
const [search, setSearch] = useState("");
const [refreshing, setRefreshing] = useState(false);

// console.log(coins)
// console.log(search)

const loadData = async () => {
  const res = await fetch(
```

```

        "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order
=market_cap_desc&per_page=100&page=1&sparkline=false"
    );
    const data = await res.json();
    setCoins(data);
  };

useEffect(() => {
  loadData();
}, []);

return (
  <View style={styles.container}>
    <StatusBar backgroundColor="red" />
    <View style={styles.header}>
      <Text style={styles.title}>CryptoMarket</Text>
      <TextInput
        style={styles.searchInput}
        placeholder="Search a Coin"
        placeholderTextColor="#858585"
        onChangeText={(text) => setSearch(text)}
      />
    </View>
    <FlatList
      style={styles.list}
      data={coins.filter(
        (coin) =>
          coin.name.toLowerCase().includes(search) ||
          coin.symbol.toLowerCase().includes(search)
      )}
      renderItem={({ item }) => {
        return <CoinItem coin={item} />;
      }}
      showsVerticalScrollIndicator={false}
      refreshing={refreshing}
      onRefresh={async () => {
        // console.log("refresh");
        setRefreshing(true);
        await loadData();
        setRefreshing(false);
      }}
    />
  </View>
);
};

```

Código completo de la app

Código App

```
import React, { useEffect, useState } from "react";
import {
  View as View,
  Text,
  FlatList,
  StyleSheet,
  TextInput,
  StatusBar,
} from "react-native";
import CoinItem from "../components/CoinItem";

const App = () => {
  const [coins, setCoins] = useState([]);
  const [search, setSearch] = useState("");
  const [refreshing, setRefreshing] = useState(false);

  // console.log(coins)
  // console.log(search)

  const loadData = async () => {
    const res = await fetch(
      "https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=market_cap_desc&per_page=100&page=1&sparkline=false"
    );
    const data = await res.json();
    setCoins(data);
  };

  useEffect(() => {
    loadData();
  }, []);

  return (
    <View style={styles.container}>
      <StatusBar backgroundColor="red" />
      <View style={styles.header}>
        <Text style={styles.title}>CryptoMarket</Text>
        <TextInput
          style={styles.searchInput}
          placeholder="Search a Coin"
          placeholderTextColor="#858585"
          onChangeText={(text) => setSearch(text)}
        />
      </View>
    </View>
  );
};
```

```

    <FlatList
      style={styles.list}
      data={coins.filter(
        (coin) =>
          coin.name.toLowerCase().includes(search) ||
          coin.symbol.toLowerCase().includes(search)
      )}
      renderItem={({ item }) => {
        return <CoinItem coin={item} />;
      }}
      showsVerticalScrollIndicator={false}
      refreshing={refreshing}
      onRefresh={async () => {
        // console.log("refresh");
        setRefreshing(true);
        await loadData();
        setRefreshing(false);
      }}
    />
  </View>
);
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#141414",
    alignItems: "center",
    flex: 1,
  },
  title: {
    color: "#fff",
    marginTop: 10,
    fontSize: 20,
  },
  list: {
    width: "90%",
  },
  header: {
    flexDirection: "row",
    // backgroundColor: "blue",
    justifyContent: "space-between",
    width: "90%",
    marginBottom: 10,
  },
  searchInput: {
    color: "#fff",
    borderBottomWidth: 1,
    borderBottomColor: "#4657ce",
    width: "40%",
  },
});

```



```

    textAlign: "center",
    fontSize: 18,
  },
});

export default App;

```

Código de CoinItem

```

import React from "react";
import { View, Text, StyleSheet, Image } from "react-native";

const CoinItem = ({ coin }) => {
  return (
    <View style={styles.container}>
      <View style={styles.coinName}>
        <Image style={styles.image} source={{ uri: coin.image }} />

        <View style={styles.containerNames}>
          <Text style={styles.text}>{coin.name}</Text>
          <Text style={styles.textSymbol}>{coin.symbol}</Text>
        </View>
      </View>

      <View>
        <Text style={styles.textPrice}>$ {coin.current_price}</Text>
        <Text
          style={[
            styles.pricePercentage,
            coin.price_change_percentage_24h > 0
              ? styles.priceUp
              : styles.priceDown,
          ]}
        >
          {coin.price_change_percentage_24h}
        </Text>
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    backgroundColor: "#121212",
    paddingTop: 10,
  },
});

```

```

        flexDirection: "row",
        justifyContent: "space-between",
    },
    containerNames: {
        marginLeft: 10,
    },
    coinName: {
        flexDirection: "row",
    },
    image: {
        width: 30,
        height: 30,
    },
    text: {
        color: "#fff",
    },
    textSymbol: {
        color: "#535353",
        textTransform: "uppercase",
    },
    pricePercentage: {
        textAlign: "right",
    },
    textPrice: {
        textAlign: "right",
        color: "#fff",
    },
    priceUp: {
        color: "#00b589",
    },
    priceDown: {
        color: "#fc4422",
    },
    });

export default CoinItem;

```

Resultado App

