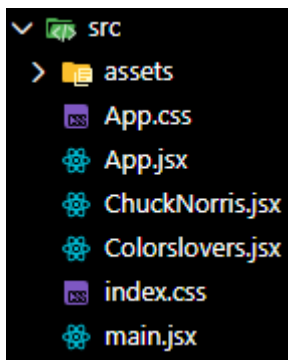


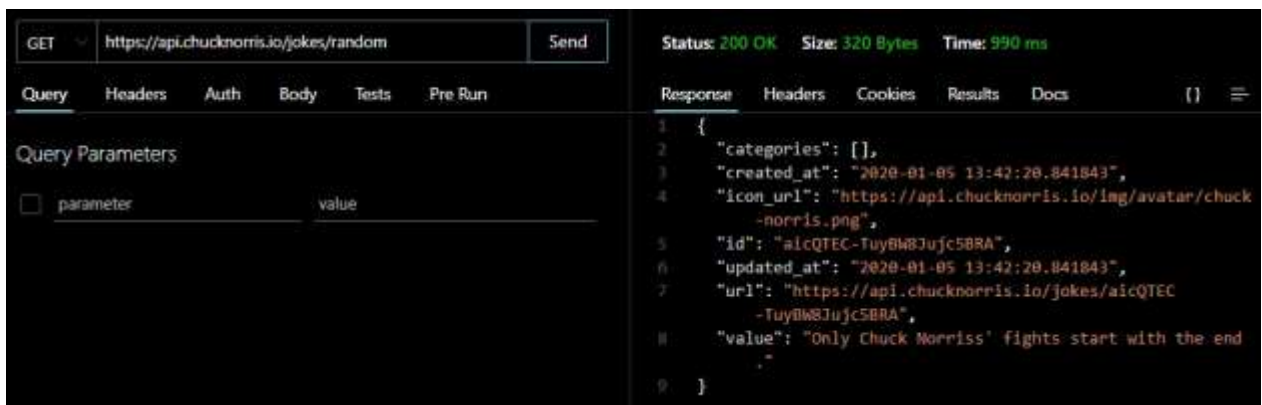
Archivos



App.jsx

```
1  import ChuckNorris from "../ChuckNorris";
2  import Colorslovers from "../Colorslovers";
3
4  function App() {
5    return (
6      <div>
7        <ChuckNorris></ChuckNorris>
8        <Colorslovers></Colorslovers>
9      </div>
10   )
11 }
12 export default App
```

Thunder client (VSC) - (Objeto)



ChuckNorris.jsx

```
1  import { useEffect, useState } from "react";
2
3  function ChuckNorris(){
4    const [id,setId]=useState("");
5    const [icon,setIcon]=useState("");
6    const [value,setValue]=useState("");
7
8    useEffect(()=>{
9      |   cargarDatos()
10     },[]);
11     async function cargarDatos(){
12       const res=await fetch("https://api.chucknorris.io/jokes/random",{method:"GET"});
13       const data=await res.json();
14
15       setId(data.id);
16       setIcon(data.icon_url);
17       |   setValue(data.value);
18     }
19     return(
20       <div>
21         <div>
22           <h2>Ejercicio 1</h2>
23           <p>{id}</p>
24           <img src={icon}></img>
25           <p>{value}</p>
26         </div>
27       </div>
28     )
29   }
30   export default ChuckNorris;
```

Thunder client (VSC) – (Lista)

The screenshot displays the Thunder Client interface within Visual Studio Code. The top bar shows a GET request to `https://www.colourlovers.com/api/colors/new?format=json` with a status of 200 OK, size of 11.32 KB, and time of 1.04 s. The left sidebar contains tabs for Query, Headers, Auth, Body, Tests, and Pre Run. The 'Query' tab is active, showing 'Query Parameters' with 'format' set to 'json' and 'parameter' set to 'value'. The main area displays the 'Response' tab, showing a JSON object with the following structure:

```
[
  {
    "id": 10308812,
    "title": "Distraction",
    "userName": "Bubbletripper",
    "numViews": 0,
    "numVotes": 0,
    "numComments": 0,
    "numHearts": 0,
    "rank": 0,
    "dateCreated": "2025-11-21 08:43:37",
    "hex": "0014C6",
    "rgb": {
      "red": 0,
      "green": 20,
      "blue": 198
    },
    "hsv": {
      "hue": 234,
      "saturation": 100,
      "value": 78
    },
    "description": "",
    "url": "http://www.colourlovers.com/color/0014C6/Distraction",
    "imageUrl": "http://www.colourlovers.com/img/0014C6"
  }
]
```

At the bottom right of the response area, there are buttons for 'Response' and 'Chart'.

ColorsLovers.jsx

```
1 import { useEffect, useState } from "react";
2
3 function ColorsLovers(){
4   const [lovers, setLovers] = useState([]);
5   useEffect(() => {
6     cargarApi();
7   }, []);
8   async function cargarApi(){
9     const datos = await fetch('https://api.allorigins.win/raw?url=https://www.colourlovers.com/api/colors/new?format=json', {method: "GET"});
10    const data = await datos.json();
11    setLovers(data)
12  }
13  return(
14    <div>
15      <h2>Ejercicio 2</h2>
16      <table>
17        <tr>
18          <th>Id</th>
19          <th>Titulo</th>
20          <th>Imagen</th>
21        </tr>
22        <tr>
23          <td>{lovers.map((color) => {
24            <tr>
25              <td>{color.id}</td>
26              <td>{color.title}</td>
27              <td><img src={color.imageUrl}></img></td>
28            </tr>
29          })}</td>
30        </tr>
31      </table>
32    </div>
33  )
34 }
35 export default ColorsLovers;
```