

App Recetas

Alexis Martinez Acosta

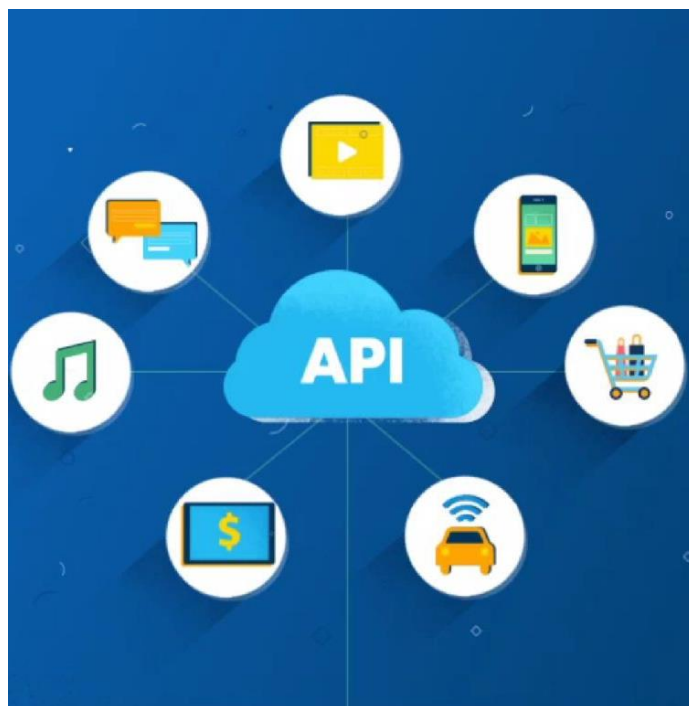
Contenido

- Introducción 2
- Desarrollo..... 3
 - Api..... 5
 - Diseño de la pagina 8
 - El código App.js..... 10
 - App.css..... 12
 - El código Footer.js 13
 - El código footer.css..... 14
 - El código Header.js 16
 - El código header.css..... 17
 - El código Recipes.js 18
 - El código recipes.css 19
- Resultados 19
- Conclusión 23

Introducción

Las Interfaces de Programación de Aplicaciones (API, por sus siglas en inglés) constituyen uno de los servicios de intercambio de información y acceso a datos más comunes en la actualidad. Complementan la disponibilidad de datos en ficheros descargables y aportan una serie de ventajas que hacen que sea un medio de acceso y consumo de datos imprescindible en cualquier Iniciativa de Datos Abiertos. Además, es el mecanismo de acceso idóneo para publicar datos con alta frecuencia de actualización como los datos en tiempo real o dinámicos.

Las APIs son elementos software implementados para cubrir múltiples propósitos. Esta guía pone el foco en la disponibilidad de APIs en el contexto de los Datos Abiertos y esta orientación implica poner el énfasis en las operaciones que permiten el acceso para lectura y descarga de datos y no tanto en la capacidad de creación o modificación de datos por parte de los usuarios de la API.



Desarrollo

1. Generar un directorio donde se alojará la aplicación

C:\POGIT\CRUD

2. Dentro de el directorio anterior, crear 1 carpeta a) Cliente; en la que se alojara el api

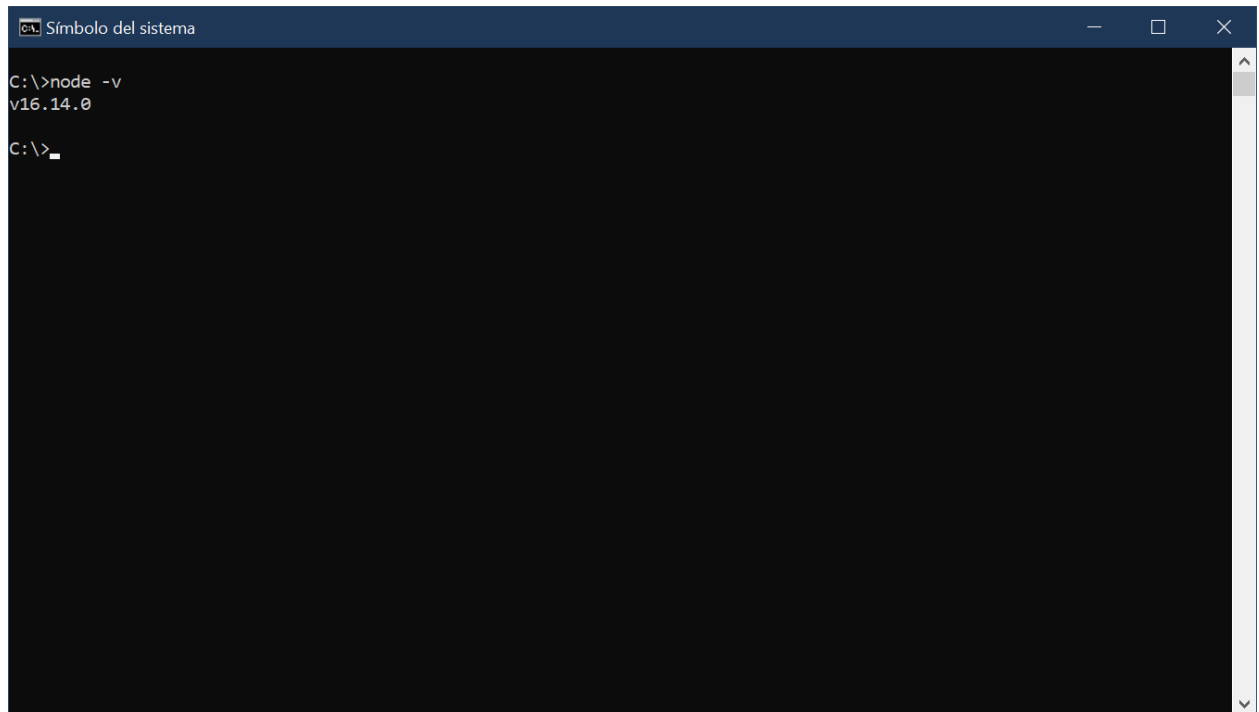
C:\POGIT\CRUD\cliente

3. Una vez realizado lo anterior; se procede a la instalación, preparación y configuración de las diferentes herramientas y la escritura de código para desarrollar la parte

Descargar e instalae Node.js

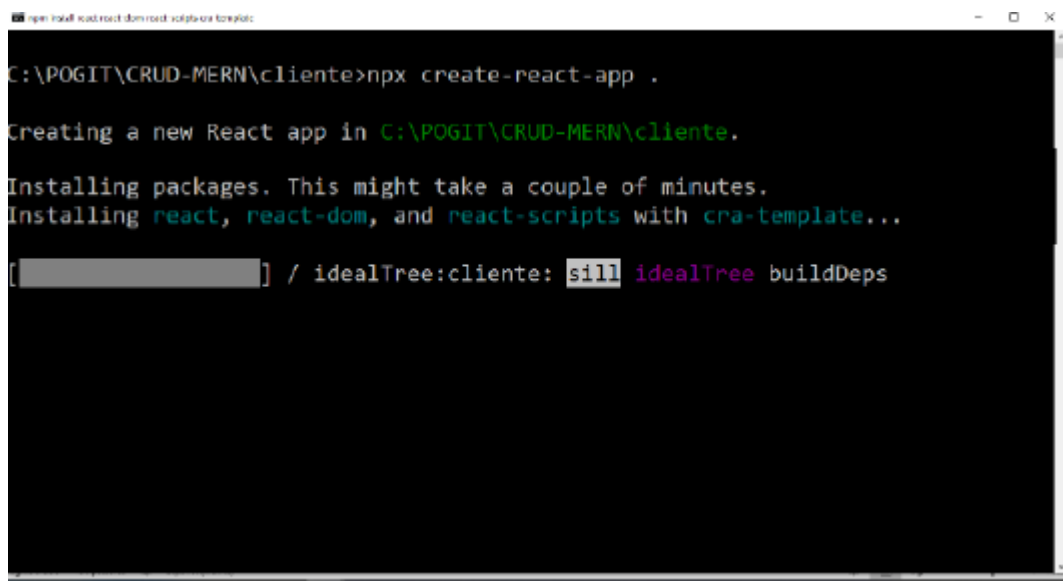


Comprobar si la instalación fue correcta con el comando `node -v` desde cmd



```
Símbolo del sistema
C:\>node -v
v16.14.0
C:\>
```

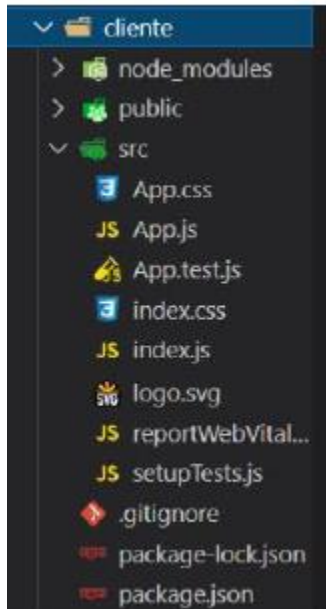
- Después accedemos a el directorio cliente `C:\POGIT\CRUD\cliente` con el fin de preparar el entorno con el comando : `npx-create-react-app .p`



```
npx install react react-dom react-scripts cra-template
C:\POGIT\CRUD-MERN\cliente>npx create-react-app .
Creating a new React app in C:\POGIT\CRUD-MERN\cliente.
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
[REDACTED] / idealTree:cliente: sill idealTree buildDeps
```

- A partir de este punto, se trabajarán 2 archivos del cliente (front-end): **App.js** y **App.css** de la aplicación ReactJS.

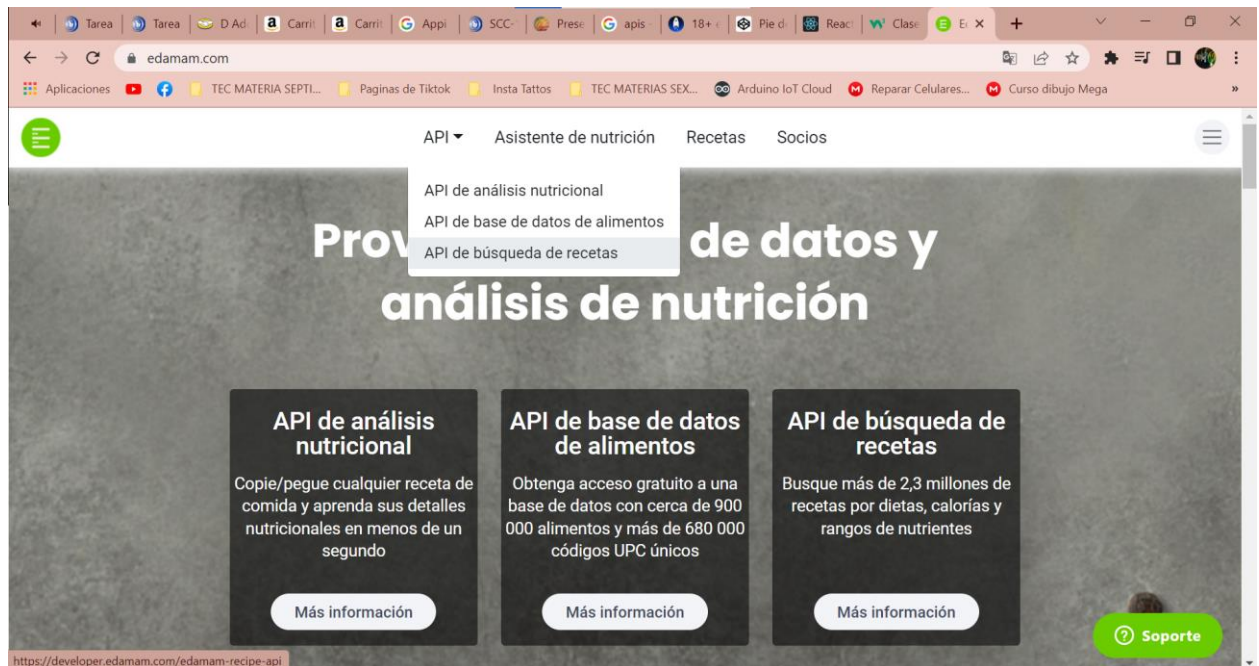
- La estructura de directorios de la aplicación cliente quedaría de la siguiente manera:



5. Para poner en funcionamiento la aplicación cliente, se ejecuta desde consola el siguiente comando: `npm start`

Api

1. Para la api primero se tiene que crear una cuenta en la pagina : <https://www.edamam.com/>
2. Ya en la pagina nos dirigimos a el apartado Api en “Api de búsqueda de receta”

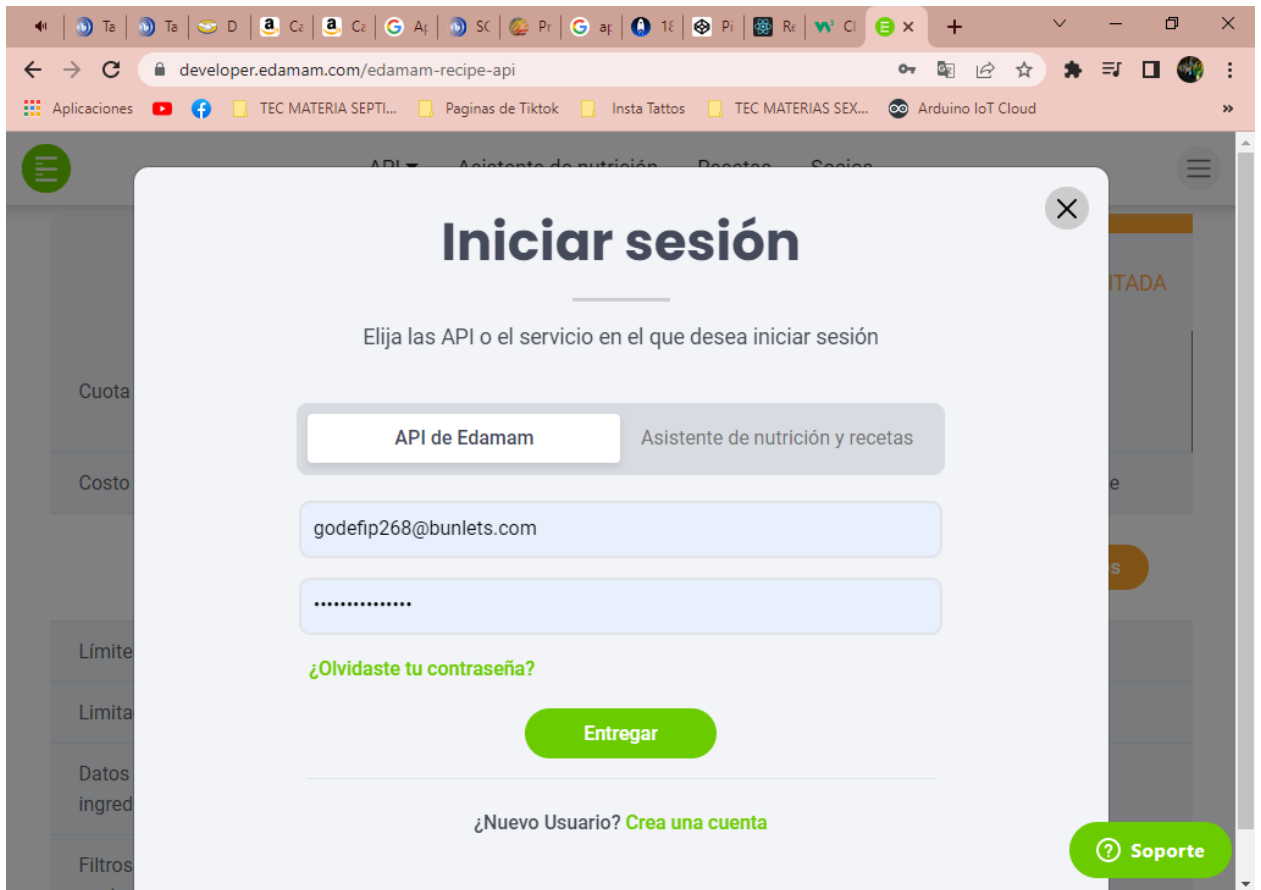


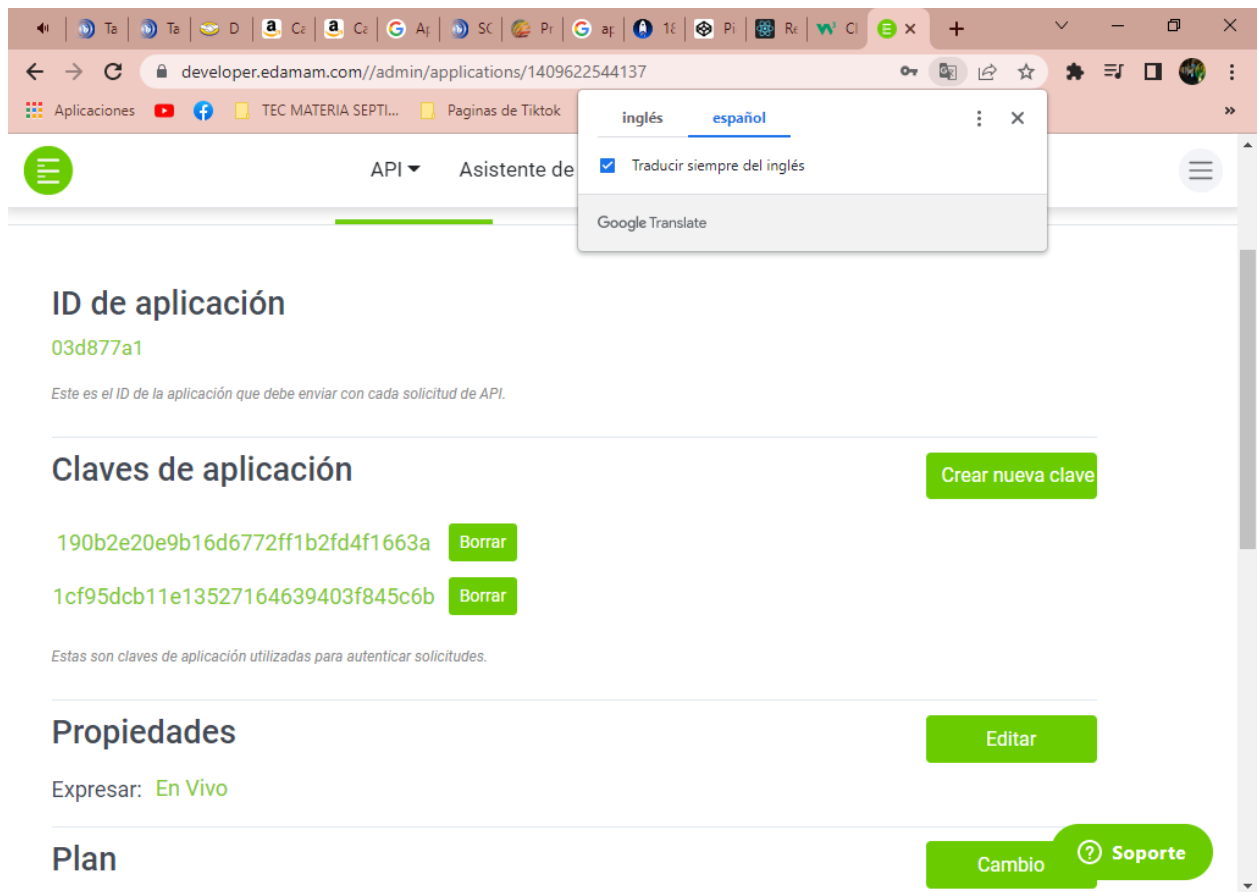
3. Seleccionamos la opción gratis el cual permitirá usarla, pero con ciertas limitaciones

	DESARROLLADOR	NÚCLEO EMPRESARIAL	EMPRESA ILIMITADA
Cuota mensual	GRATIS	\$49 por mes	\$999 por mes
Costo excedente por visita	No permitido	\$0.01/por visita	\$0 por golpe
	Empezar	Suscríbese ahora	Contáctenos
Límites de llamadas a la API	10,000 mes	50,000 mes	ilimitado*
Limitación de llamadas/min	10 minutos	ilimitado*	ilimitado*
Datos de la receta (imagen, ingredientes, título)	✓	✓	✓
Filtros de tipo de comida/plato y cocina	✓	✓	✓

A 'Soporte' button is visible in the bottom right corner.

4. Accedemos con nuestra cuenta creada y ya ahí nos dará dos códigos los cuales no servirá para usar la api





Diseño de la pagina

1. En App.js será en donde se agregará la api así como el Header y el Footer
2. Importamos los archivos que nos servirán para el proyecto, estos los usaremos en el App.js y se modificarán independientemente

```
> src > > App.js > App
import './App.css';
import React, {useEffect, useState} from 'react';
import Recipes from './Recipes';
import Header from './Header';
import Footer from './Footer';
```

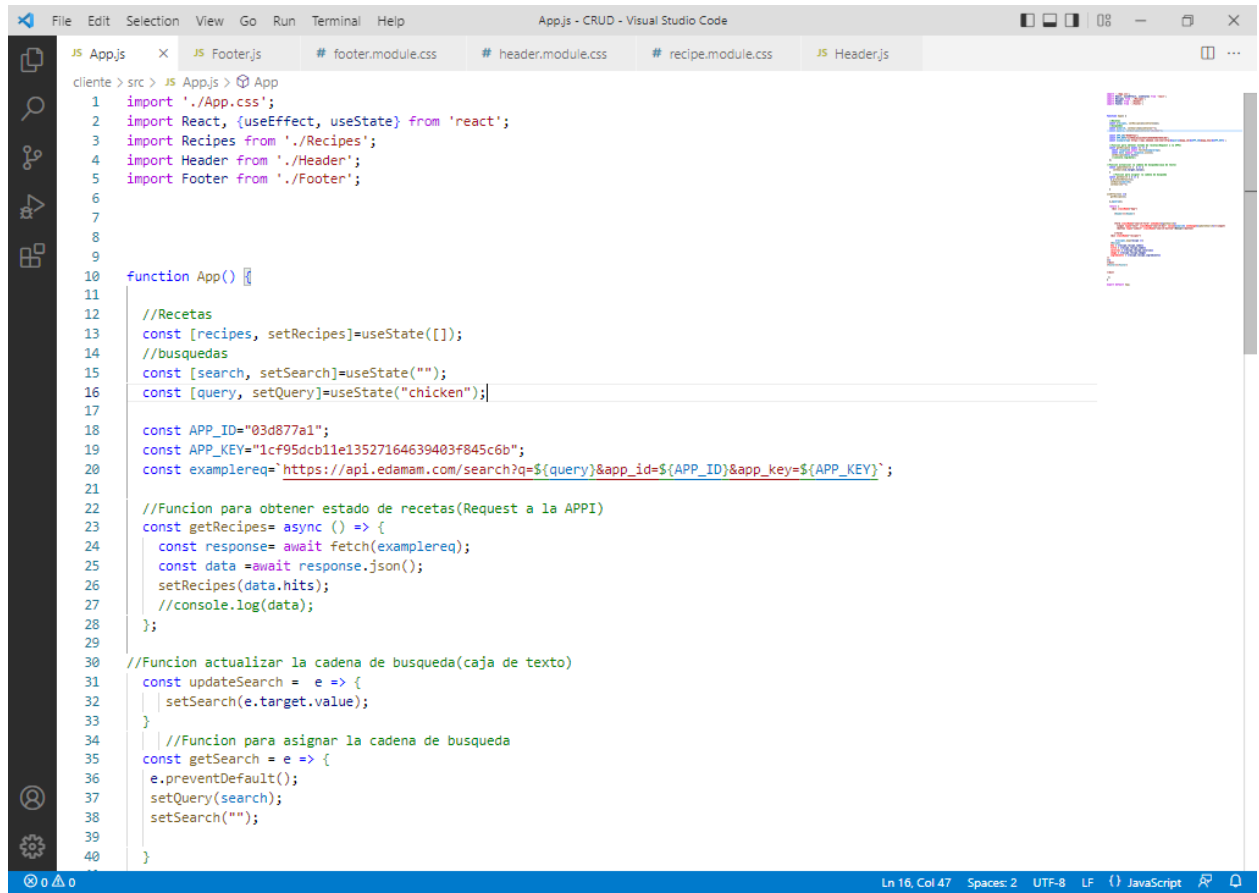
3. Agregaremos los códigos previamente proporcionados con el fin de acceder con nuestra cuenta a él api

```
const APP_ID="03d877a1";  
const APP_KEY="1cf95dcb11e13527164639403f845c6b";  
const exemplereq=`https://api.edamam.com/search?q=${query}&app_id=${APP_ID}&app_key=${APP_KEY}`;
```

Agregaremos esta función para poder consultar las recetas

```
//Funcion para obtener estado de recetas(Request a la APPI)  
const getRecipes= async () => {  
  const response= await fetch(exemplereq);  
  const data =await response.json();  
  setRecipes(data.hits);  
  //console.log(data);  
};
```

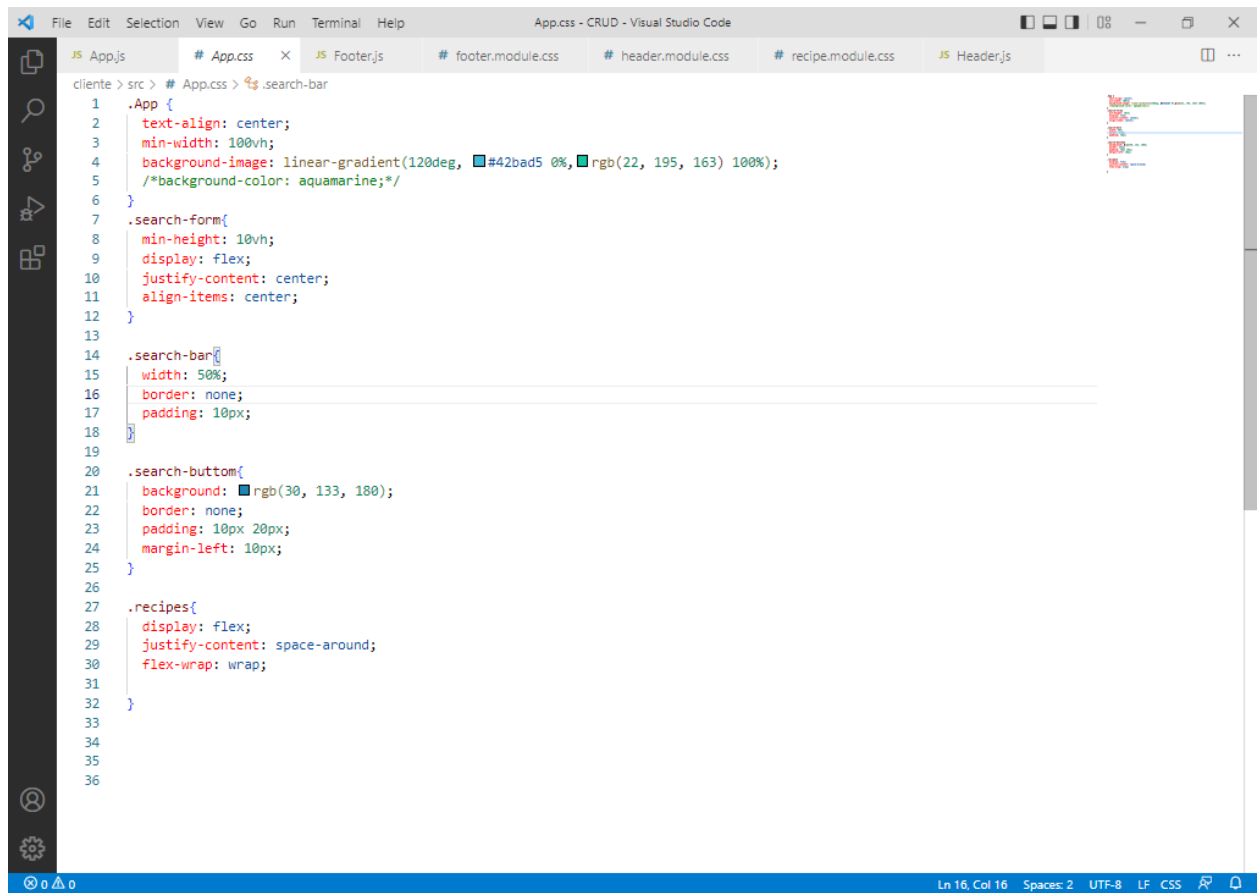
El código App.js



```
cliente > src > JS App.js > App
1 import './App.css';
2 import React, {useEffect, useState} from 'react';
3 import Recipes from './Recipes';
4 import Header from './Header';
5 import Footer from './Footer';
6
7
8
9
10 function App() {
11
12     //Recetas
13     const [recipes, setRecipes]=useState([]);
14     //busquedas
15     const [search, setSearch]=useState("");
16     const [query, setQuery]=useState("chicken");
17
18     const APP_ID="03d877a1";
19     const APP_KEY="1cf95dcb11e13527164639403f845c6b";
20     const exemplereq="https://api.edamam.com/search?q={query}&app_id={APP_ID}&app_key={APP_KEY}";
21
22     //Funcion para obtener estado de recetas(Request a la APPI)
23     const getRecipes= async () => {
24         const response= await fetch(exemplereq);
25         const data =await response.json();
26         setRecipes(data.hits);
27         //console.log(data);
28     };
29
30     //Funcion actualizar la cadena de busqueda(caja de texto)
31     const updateSearch = e => {
32         setSearch(e.target.value);
33     }
34     //Funcion para asignar la cadena de busqueda
35     const getSearch = e => {
36         e.preventDefault();
37         setQuery(search);
38         setSearch("");
39     }
40 }
```

```
File Edit Selection View Go Run Terminal Help App.js - CRUD - Visual Studio Code
JS App.js JS Footer.js # footer.module.css # header.module.css # recipe.module.css JS Header.js
cliente > src > JS App.js > App
41
42 useEffect(() =>{
43   getRecipes();
44 },[query]);
45
46
47 return (
48   <div className="App">
49
50     <Header></Header>
51
52
53
54     <form className="search-form" onSubmit={getSearch}>
55       <input type="text" className="search-bar" value={search} onChange={updateSearch}></input>
56       <button type="submit" className="search-button">Enviar</button>
57     </form>
58
59     <div className="recipes">
60
61       {recipes.map(recipe =>{
62         <Recipes
63           key = {recipe.recipe.label}
64           title = {recipe.recipe.label}
65           calories = {recipe.recipe.calories}
66           image = {recipe.recipe.image}
67           ingredients = {recipe.recipe.ingredients}
68         />
69       )}}
70     </div>
71     <Footer></Footer>
72
73
74   </div>
75
76 );
77 }
78
79 export default App;
80
```

App.css



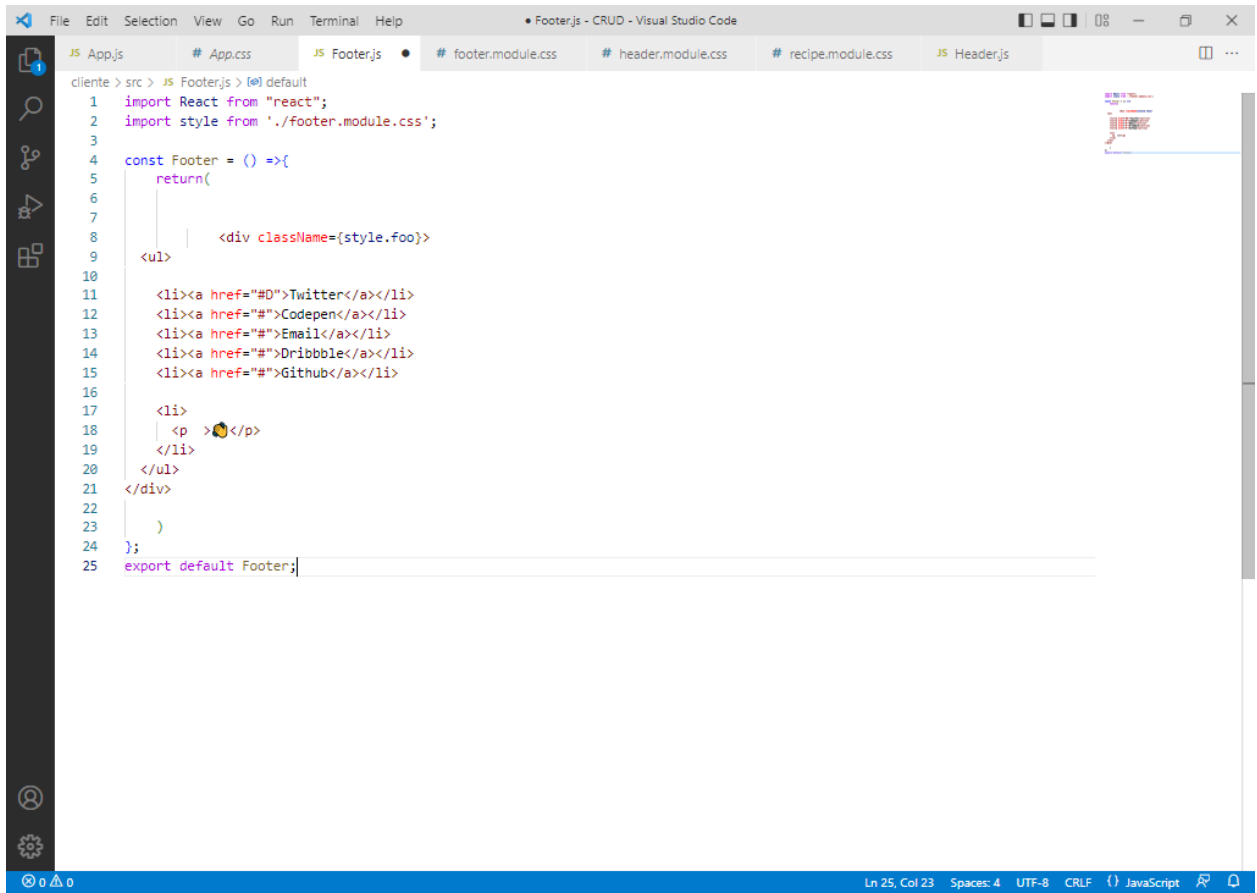
```
File Edit Selection View Go Run Terminal Help
App.css - CRUD - Visual Studio Code

JS App.js # App.css x JS Footer.js # footer.module.css # header.module.css # recipe.module.css JS Header.js

cliente > src > # App.css > $.search-bar
1 .App {
2   text-align: center;
3   min-width: 100vh;
4   background-image: linear-gradient(120deg, #42bad5 0%, rgb(22, 195, 163) 100%);
5   /*background-color: aquamarine;*/
6 }
7 .search-form{
8   min-height: 10vh;
9   display: flex;
10  justify-content: center;
11  align-items: center;
12 }
13
14 .search-bar{
15   width: 50%;
16   border: none;
17   padding: 10px;
18 }
19
20 .search-button{
21   background: rgb(30, 133, 180);
22   border: none;
23   padding: 10px 20px;
24   margin-left: 10px;
25 }
26
27 .recipes{
28   display: flex;
29   justify-content: space-around;
30   flex-wrap: wrap;
31 }
32
33
34
35
36

Ln 16, Col 16 Spaces: 2 UTF-8 LF CSS
```

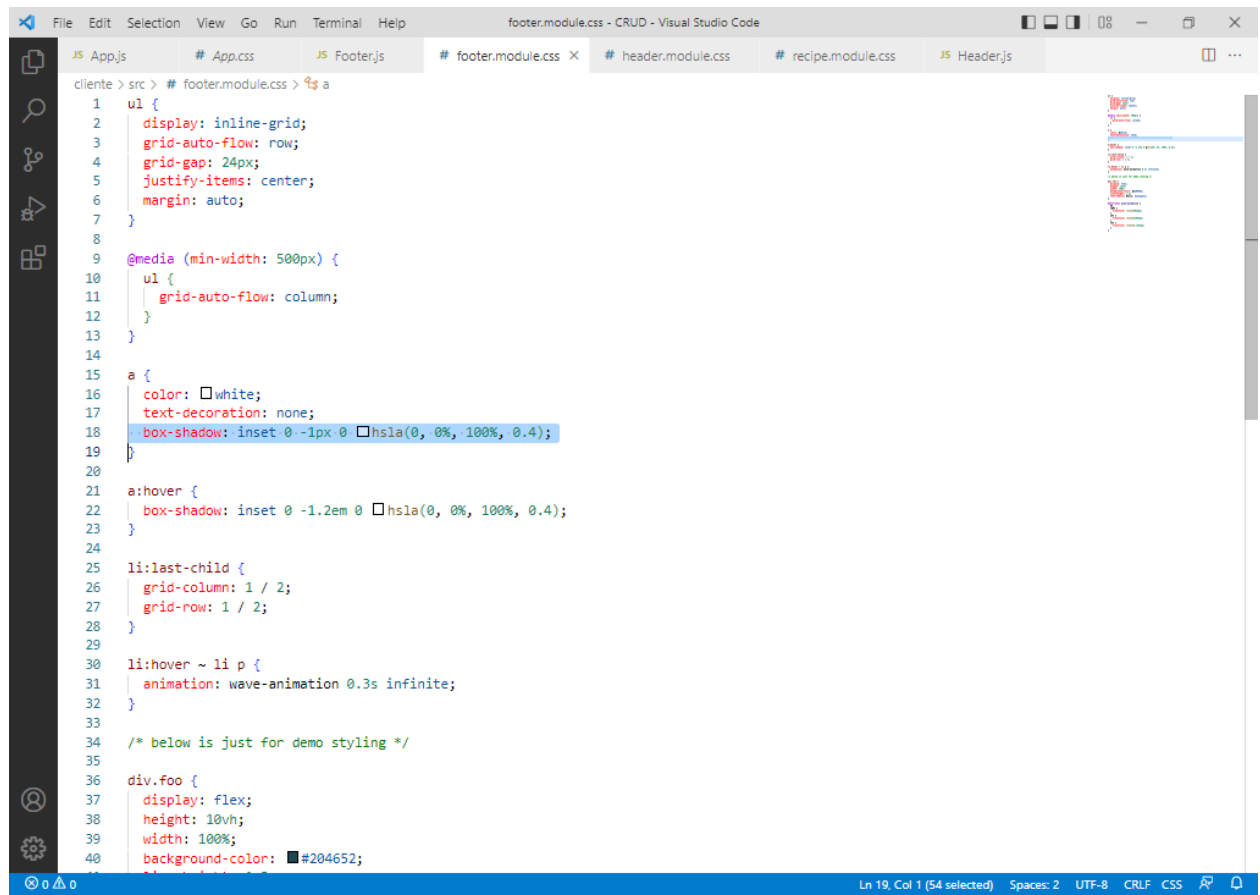
El código Footer.js



```
File Edit Selection View Go Run Terminal Help
Footer.js - CRUD - Visual Studio Code
JS App.js # App.css JS Footer.js # footer.module.css # header.module.css # recipe.module.css JS Header.js
cliente > src > JS Footer.js > default
1 import React from "react";
2 import style from './footer.module.css';
3
4 const Footer = () =>{
5   return(
6
7
8     <div className={style.foo}>
9
10
11     <ul>
12
13       <li><a href="#D">Twitter</a></li>
14       <li><a href="#">Codepen</a></li>
15       <li><a href="#">Email</a></li>
16       <li><a href="#">Dribbble</a></li>
17       <li><a href="#">Github</a></li>
18
19     </ul>
20
21     <li>
22       <p> <img alt="emoji" data-bbox="295 335 315 345"/> </p>
23     </li>
24   </div>
25
26   )
27 };
28 export default Footer;
```

Ln 25, Col 23 Spaces: 4 UTF-8 CRLF JavaScript

El código footer.css



```
1  ul {
2    display: inline-grid;
3    grid-auto-flow: row;
4    grid-gap: 24px;
5    justify-items: center;
6    margin: auto;
7  }
8
9  @media (min-width: 500px) {
10   ul {
11     grid-auto-flow: column;
12   }
13 }
14
15 a {
16   color: white;
17   text-decoration: none;
18   box-shadow: inset 0 -1px 0 hsla(0, 0%, 100%, 0.4);
19 }
20
21 a:hover {
22   box-shadow: inset 0 -1.2em 0 hsla(0, 0%, 100%, 0.4);
23 }
24
25 li:last-child {
26   grid-column: 1 / 2;
27   grid-row: 1 / 2;
28 }
29
30 li:hover ~ li p {
31   animation: wave-animation 0.3s infinite;
32 }
33
34 /* below is just for demo styling */
35
36 div.foo {
37   display: flex;
38   height: 10vh;
39   width: 100%;
40   background-color: #204652;
```

Ln 19, Col 1 (54 selected) Spaces: 2 UTF-8 CRLF CSS

File Edit Selection View Go Run Terminal Help footer.module.css - CRUD - Visual Studio Code

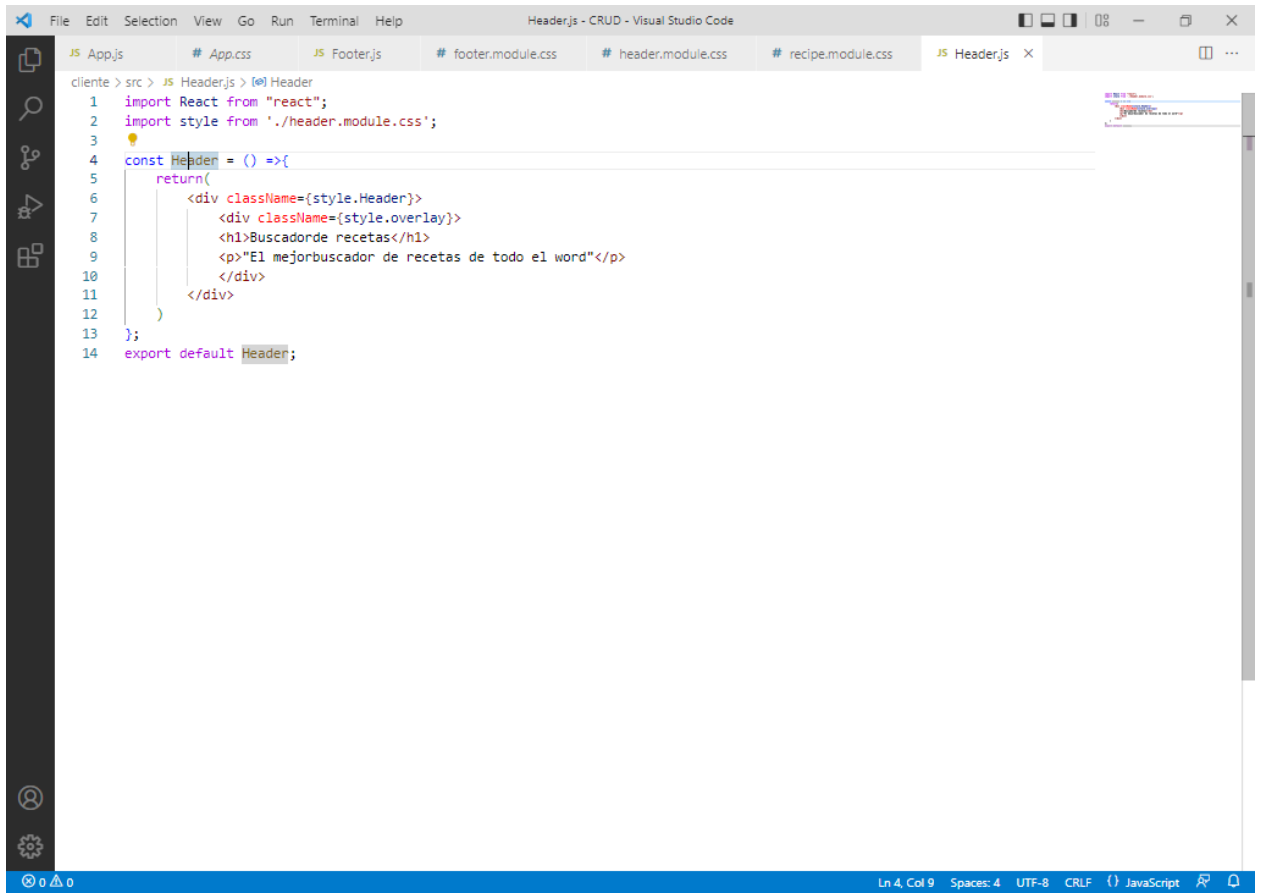
JS App.js # App.css JS Footer.js # footer.module.css X # header.module.css # recipe.module.css JS Header.js

cliente > src > # footer.module.css > a

```
30 li:hover ~ li p {
31   animation: wave-animation 0.3s infinite;
32 }
33
34 /* below is just for demo styling */
35
36 div.foo {
37   display: flex;
38   height: 10vh;
39   width: 100%;
40   background-color: #204652;
41   line-height: 1.3;
42   font-family: Menlo, monospace;
43 }
44
45 @keyframes wave-animation {
46   0%,
47   100% {
48     transform: rotate(0deg);
49   }
50   25% {
51     transform: rotate(20deg);
52   }
53   75% {
54     transform: rotate(-15deg);
55   }
56 }
57
```

Ln 19, Col 1 (54 selected) Spaces: 2 UTF-8 CRLF CSS

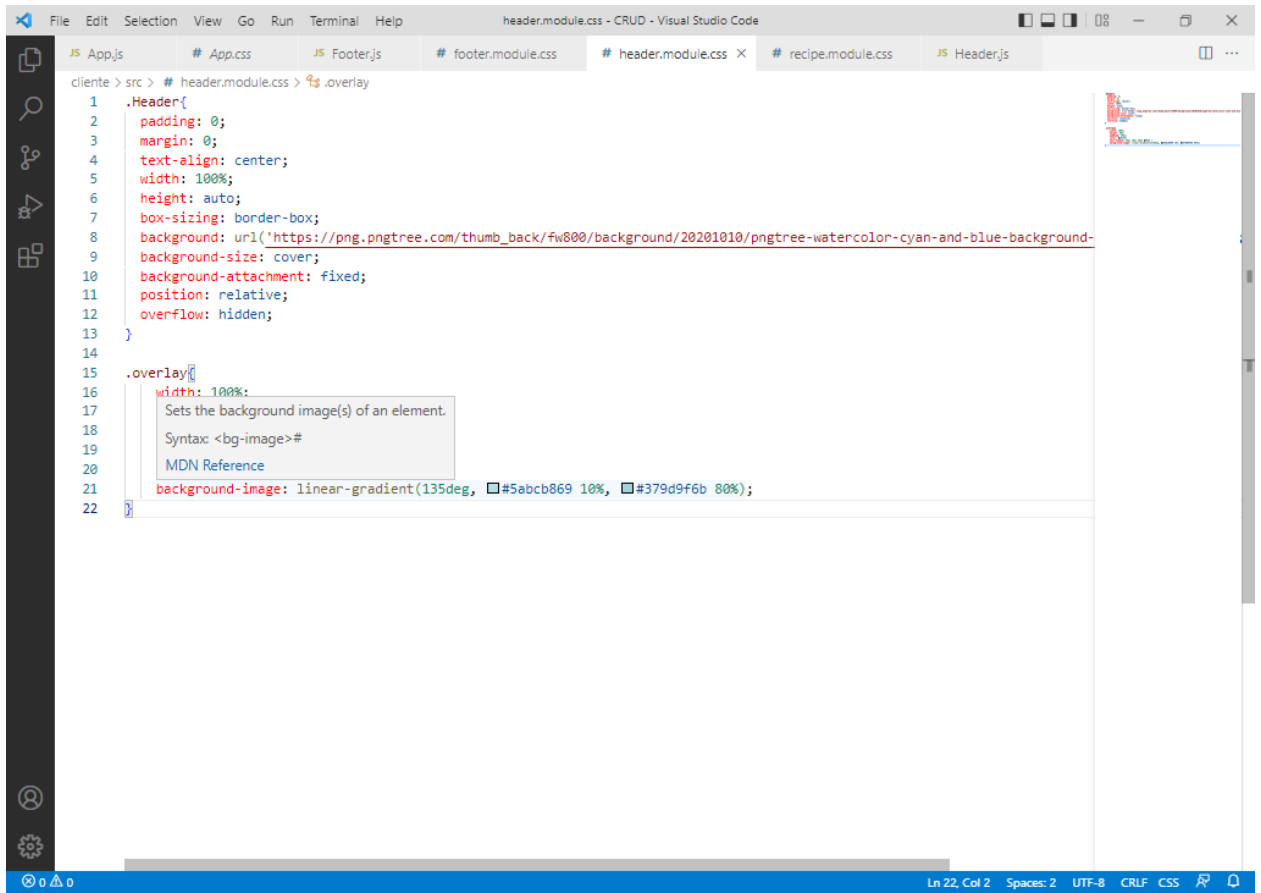
El código Header.js



```
Header.js - CRUD - Visual Studio Code
File Edit Selection View Go Run Terminal Help
JS App.js # App.css JS Footer.js # footer.module.css # header.module.css # recipe.module.css JS Header.js x
cliente > src > JS Header.js > Header
1 import React from "react";
2 import style from './header.module.css';
3
4 const Header = () =>{
5   return(
6     <div className={style.Header}>
7       <div className={style.overlay}>
8         <h1>Buscador de recetas</h1>
9         <p>"El mejor buscador de recetas de todo el word"</p>
10      </div>
11    </div>
12  )
13 };
14 export default Header;
```

Ln 4, Col 9 Spaces: 4 UTF-8 CRLF JavaScript

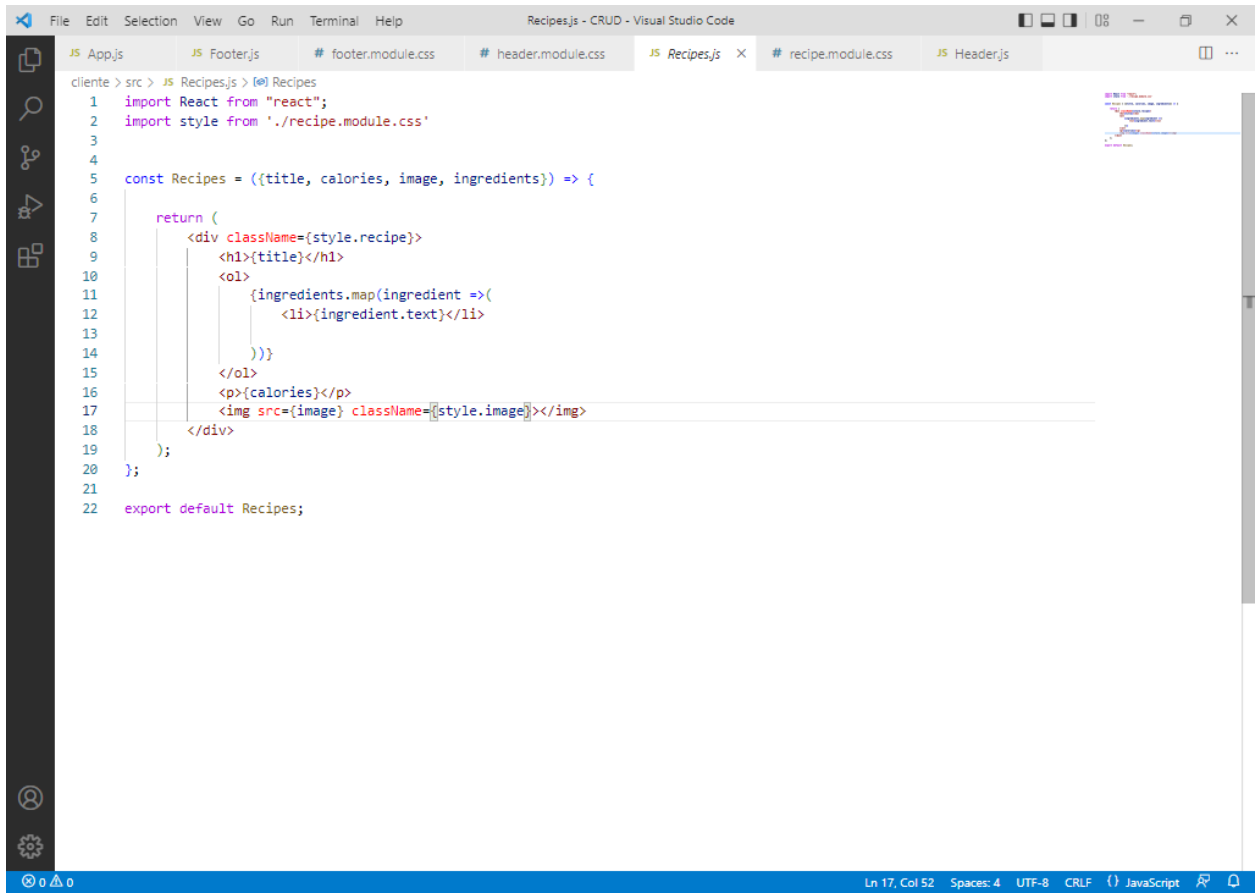
El código header.css



```
1 .Header{
2   padding: 0;
3   margin: 0;
4   text-align: center;
5   width: 100%;
6   height: auto;
7   box-sizing: border-box;
8   background: url('https://png.pngtree.com/thumb_back/fw800/background/20201010/pngtree-watercolor-cyan-and-blue-background-');
9   background-size: cover;
10  background-attachment: fixed;
11  position: relative;
12  overflow: hidden;
13 }
14
15 .overlay{
16   width: 100%;
17   Sets the background image(s) of an element.
18   Syntax: <bg-image>#
19   MDN Reference
20   background-image: linear-gradient(135deg, #5abcb869 10%, #379d9f6b 80%);
21 }
22 }
```

Ln 22, Col 2 Spaces: 2 UTF-8 CRLF CSS

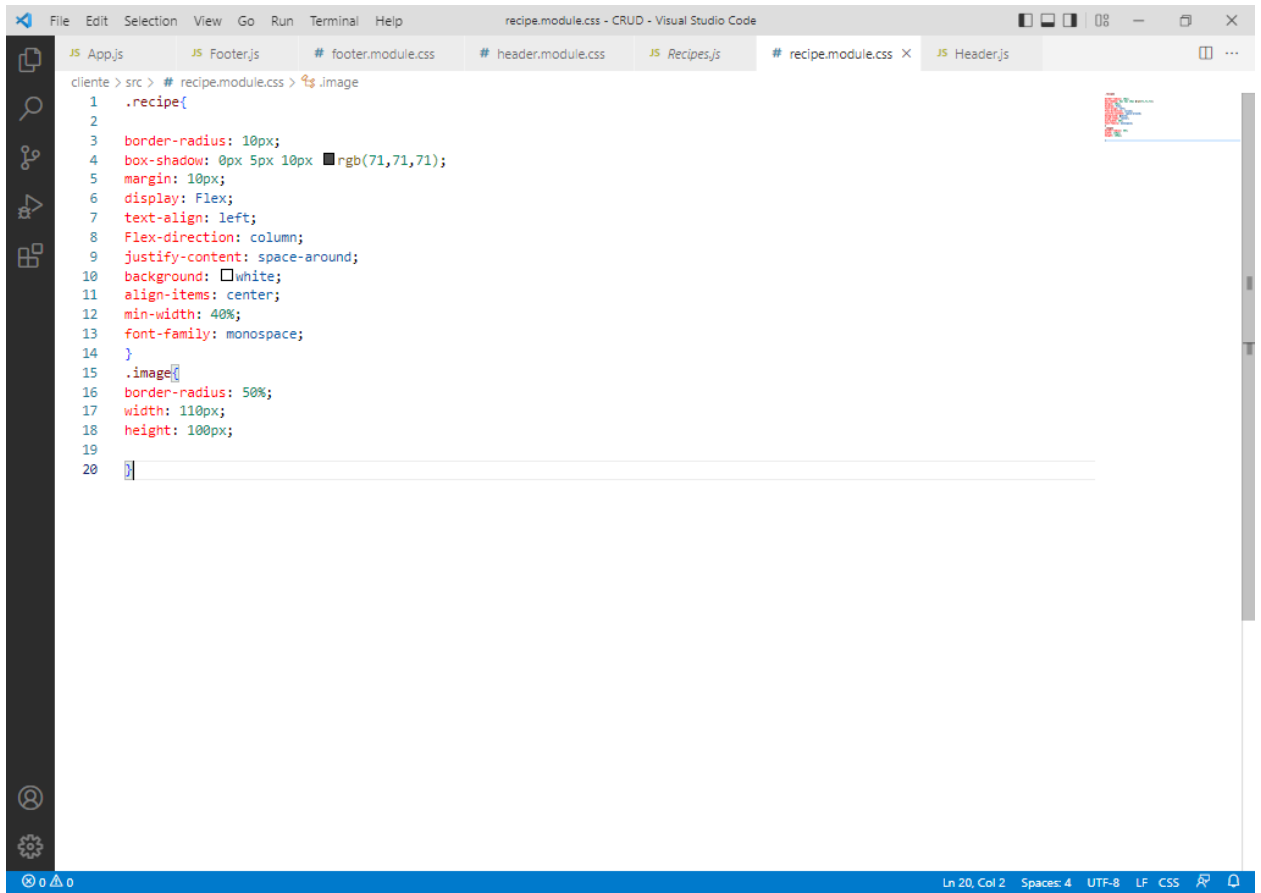
El código Recipes.js



```
File Edit Selection View Go Run Terminal Help
Recipes.js - CRUD - Visual Studio Code
JS App.js JS Footer.js # footer.module.css # header.module.css JS Recipes.js X # recipe.module.css JS Header.js
cliente > src > JS Recipes.js > Recipes
1 import React from "react";
2 import style from './recipe.module.css'
3
4
5 const Recipes = ({title, calories, image, ingredients}) => {
6
7   return (
8     <div className={style.recipe}>
9       <h1>{title}</h1>
10      <ol>
11        {ingredients.map(ingredient => (
12          <li>{ingredient.text}</li>
13        ))}
14      </ol>
15      <p>{calories}</p>
16      <img src={image} className={style.image}></img>
17    </div>
18  );
19
20 };
21
22 export default Recipes;
```

Ln 17, Col 52 Spaces: 4 UTF-8 CRLF {} JavaScript

El código recipes.css



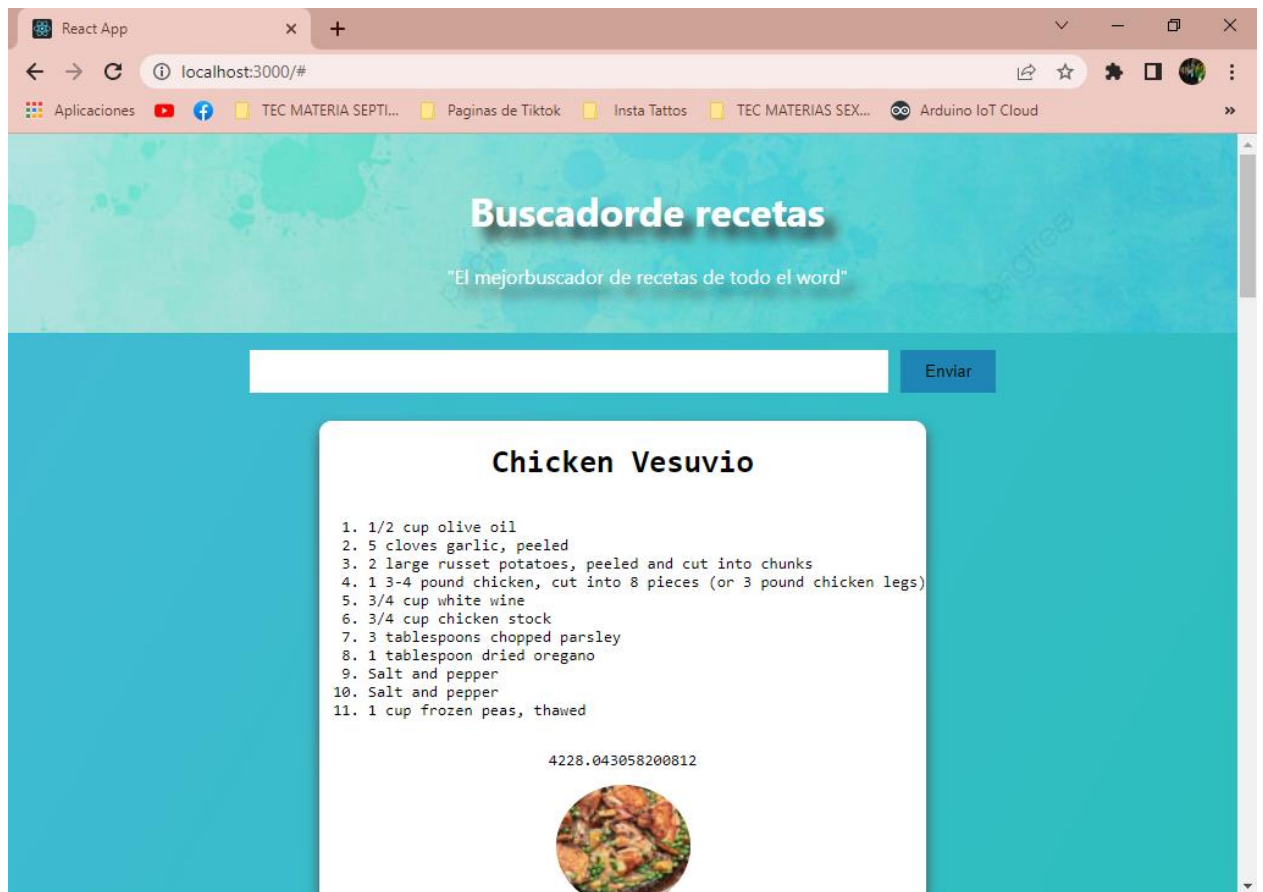
The image shows a screenshot of the Visual Studio Code editor interface. The title bar indicates the file is 'recipe.module.css - CRUD - Visual Studio Code'. The editor has several tabs open: 'JS App.js', 'JS Footer.js', '# footer.module.css', '# header.module.css', 'JS Recipes.js', '# recipe.module.css' (which is the active tab), and 'JS Header.js'. The active tab displays the following CSS code:

```
1  .recipe{
2
3    border-radius: 10px;
4    box-shadow: 0px 5px 10px #rgb(71,71,71);
5    margin: 10px;
6    display: Flex;
7    text-align: left;
8    Flex-direction: column;
9    justify-content: space-around;
10   background: #white;
11   align-items: center;
12   min-width: 40%;
13   font-family: monospace;
14 }
15
16 .image{
17   border-radius: 50%;
18   width: 110px;
19   height: 100px;
20 }
```

The status bar at the bottom shows 'Ln 20, Col 2', 'Spaces: 4', 'UTF-8', 'LF', 'CSS', and icons for search, run, and other editor functions.

Resultados

Estos son los resultados de mi aplicación



React App

localhost:3000/#

Aplicaciones

TEC MATERIA SEPTI...

Paginas de Tiktok

Insta Tattos

TEC MATERIAS SEX...

Arduino IoT Cloud

5. 1 tablespoon butter - cultured unsalted (or olive oil)

6. 240 grams onion sliced thin (1 large onion)

7. 70 grams Anaheim pepper chopped (1 large pepper)

8. 25 grams paprika (about 1/4 cup)


9. 1 cup chicken stock

10. 1/2 teaspoon salt

11. 1/2 cup sour cream

12. 1 tablespoon flour - all-purpose

3033.2012500008163



Baked Chicken


1. 6 bone-in chicken breast halves, or 6 chicken thighs and wings, skin-on

2. 1/2 teaspoon coarse salt

3. 1/2 teaspoon Mrs. Dash seasoning

4. 1/4 teaspoon freshly ground black pepper

901.58575



Catalan Chicken

1. 1 whole 4-pound chicken, quartered

2. 8 slices bacon


3. 30 cloves garlic

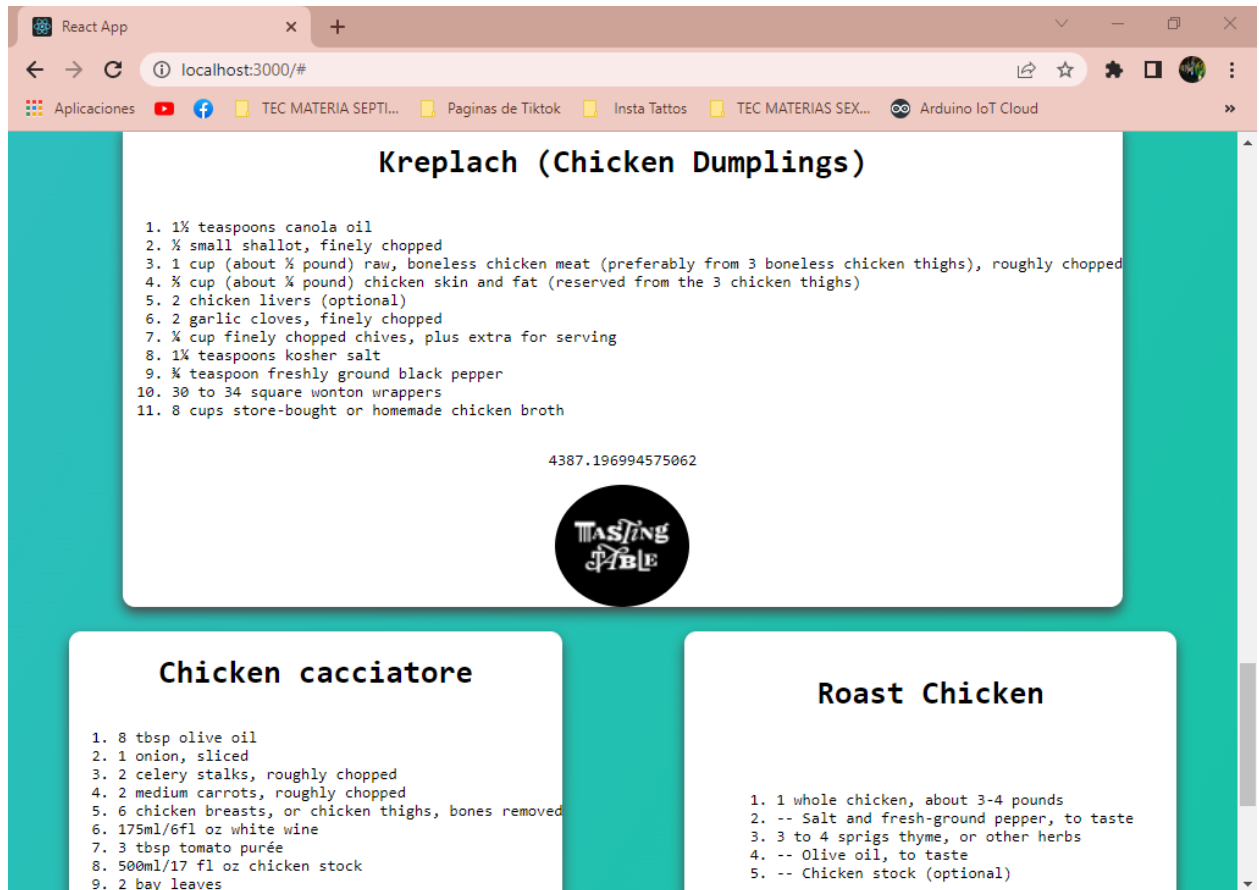
4. 3 lemons, peeled, rinds thinly sliced and reserved

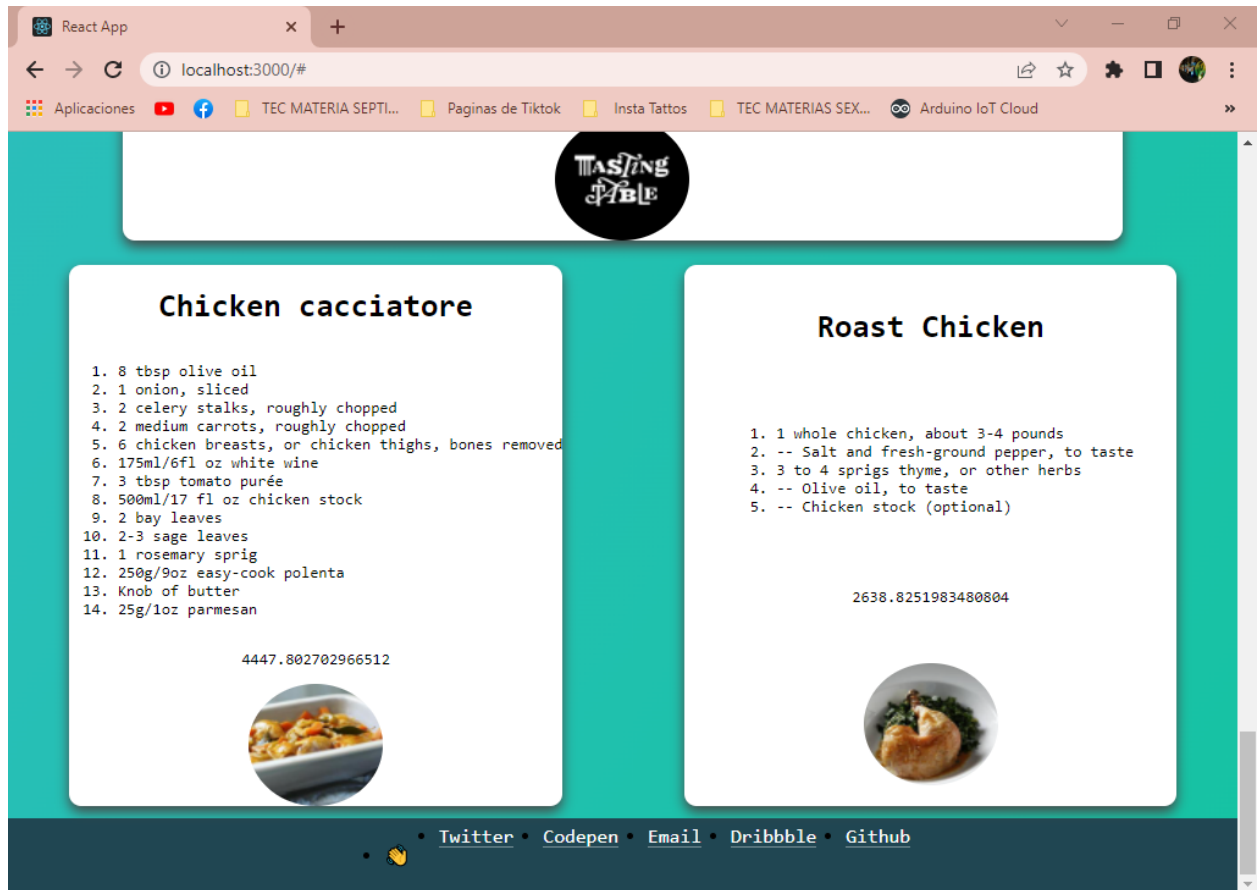
5. 1/2 cup Banyuls or another fortified dessert wine

6. 1 cup veal or chicken stock

3900.8







Conclusión

La aplicación fue desarrollada usando las tecnologías de node.js html y css complementándola con un Api , las apis son ayudan a interactuar con un sistema sin necesidad de necesidad de tener conocimiento de la estructura interna o de las tecnologías usadas en su desarrollo.

Las apis ofrecen una forma fácil de acceder a los datos que permite manipular los datos pedidos y darle el diseño que tú quieras, simplemente usando la información solicitada, como ya vimos en nuestro proyecto las consultas son URI conformado por nuestro códigos de usuario, así como el nombre de el ingrediente que queremos prepara y como resultado obtenemos cientos de recetas las cuales no están alojadas en nuestro sistema solo desarrollamos el cliente y el api se encargaría de el Back-end.