



Industria
Meteorologica

Manual Tecnico

Proyecto creado por
Fernando Arturo Ramirez Corvera RC222309

José Rodrigo Montoya Martínez MM181797

XXX



IDE de Desarrollo de la Pagina web

La pagina web es desarrollada en Visual Studio .El IDE escogido para realizar el sistema distribuido fue Visual Studio mediante React Native

Este programa incluye un jdk y una terminal para hacerla funcionar y si quieren saber porque elegimos este programa pues aqui hay unas caracteristicas



Visual Studio Code

Completando la instalación de Visual Studio Code

El programa completó la instalación de Visual Studio Code en su sistema. Puede ejecutar la aplicación haciendo clic sobre el ícono instalado.

Haga clic en Finalizar para salir del programa de instalación.

Ejecutar Visual Studio Code

Finalizar



Características

Creación de aplicaciones web.

Conectarse a bases de datos

Depurar, probar y mejorar el código

Implementación de aplicaciones.

Creacion de aplicaciones web: Visual Studio puede ayudar a escribir aplicaciones para la Web mediante :Node.js, Python, JavaScript y TypeScript

Conectarse a la base de datos: El Explorador de objetos de SQL Server ofrece una vista de los objetos de base de datos similar a la de SQL Server Management Studio.



JavaScript



JavaScript
Es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas.



CSS

Funciona como complemento a la información que forma parte de un sitio web. Mientras que el código en HTML incluye todos los datos, el código en CSS se encarga de darles formato y presentarlos visualmente a través de un navegador.

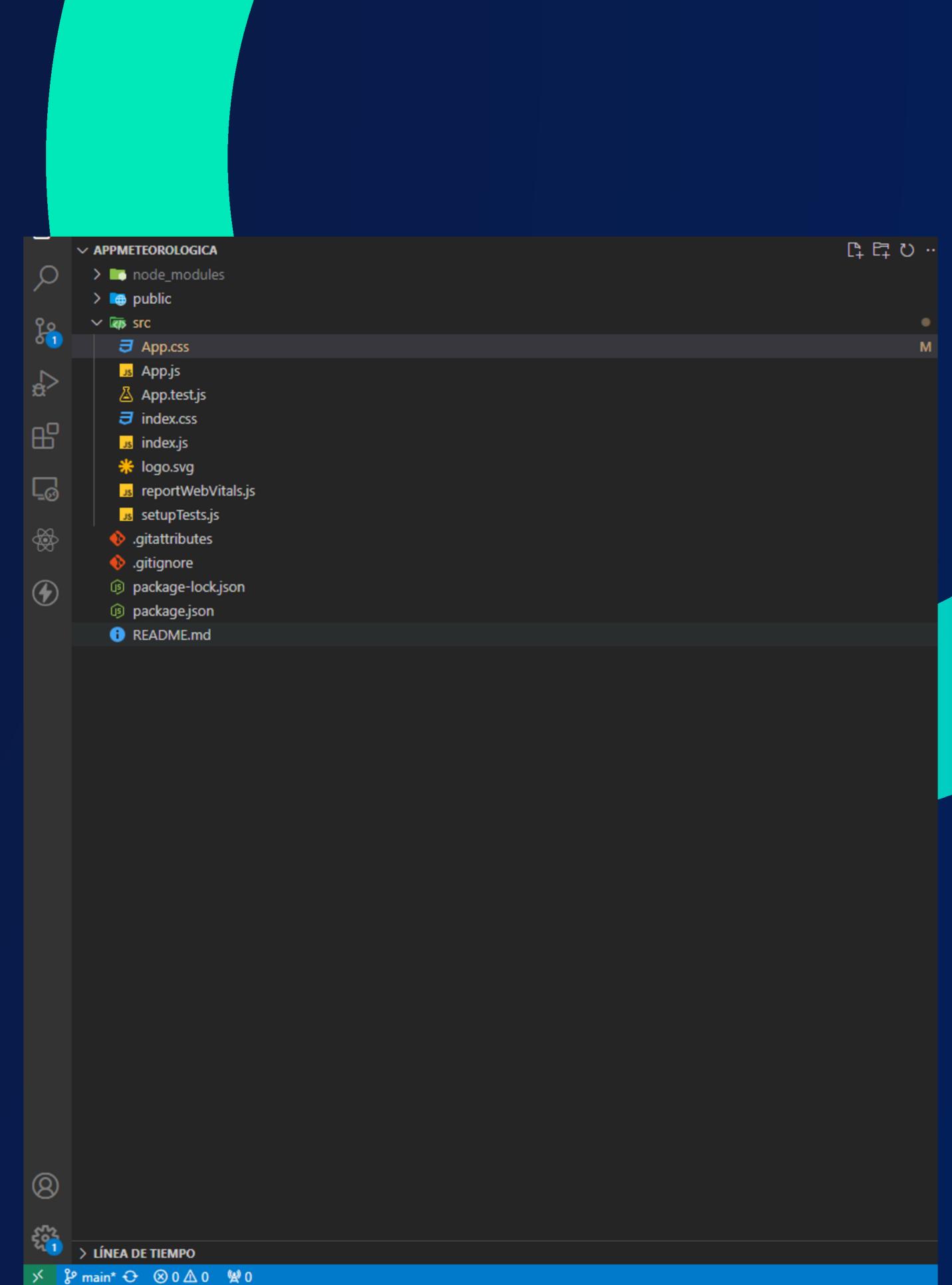


Index.css

Indica el orden de un elemento posicionado y sus descendientes.

Estructura del proyecto

**En el IDE de Visual Studio
en la parte izquierda
desplegamos la carpeta
del proyecto y
observamos su
estructura**



App.js

The screenshot shows a code editor window with the following details:

- File:** App.js
- Project Structure:** APPMETEORLOGICA / src / App.js
- Code Content:** The code is a React component named App. It includes logic for searching weather data from an API and fetching a random city image from Unsplash. The UI consists of a search bar and a button to search for a random city.

```
function App() {
  const searchPressed = () => {
    .then(res) => res.json()
    .then(result) => {
      setWeather(result);
      if (result.name) {
        fetchImage(result.name);
      }
    });
  };

// Función para buscar el clima y la imagen de una ciudad aleatoria
const searchRandomCity = () => {
  const randomCity = cities[Math.floor(Math.random() * cities.length)];
  setSearch(randomCity);
  searchPressed();
};

const fetchImage = (city) => {
  fetch(`${unsplashApi.base}?query=${city}&client_id=${unsplashApi.key}&per_page=1`)
    .then(res) => res.json()
    .then(data) => {
      if (data.results.length > 0) {
        setBackgroundImage(data.results[0].urls.regular); // Configurar la imagen de fondo
      }
    });
};

return (
  <div className="App" style={{ backgroundImage: `url(${backgroundImage})` }}> /* Fondo dinámico */
  <header className="App-header">
    <h1>Weather App</h1>
    <div className="search-box">
      <input type="text" placeholder="Ingresar Ciudad...." value={search} onChange={(e) => setSearch(e.target.value)} className="search-input" />
      <button onClick={searchPressed} className="search-button">Buscar</button>
    </div>
    <div>
      <button onClick={searchRandomCity} className="random-button">Ciudad Aleatoria</button>
    </div>
  </header>
  {typeof weather.main !== "undefined" ? (
    <div className="weather-info">
      <h2>{weather.name}, {weather.sys.country}</h2> /* Mostrar ciudad y país */
      <p className="temp">{weather.main.temp}</p>
      <p className="condition">{weather.weather[0].main}</p>
      <p className="description">{weather.weather[0].description}</p>
    </div>
  ) : (
    <div></div>
  )}
</div>
);
}

export default App;
```

- Bottom Bar:** Includes a timeline icon and the text "LÍNEA DE TIEMPO".

CSS

CSS de la app



The screenshot shows a code editor window with the title bar "AppMeteorologica". The left sidebar displays a file tree for a project named "APPMETEORLOGICA" containing files like node_modules, public, src (with App.css selected), App.js, App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitattributes, .gitignore, package-lock.json, package.json, and README.md. The main editor area shows the content of App.css:

```
/* General app styles */
.App {
  text-align: center;
  font-family: 'Arial', sans-serif;
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  transition: background-image 0.5s ease-in-out;
  height: 100vh;
}

.App-header {
  background-color: rgba(12, 191, 245, 0.5); /* Fondo semitransparente para el contenido */
  color: #0000FF;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}

h1 {
  font-size: 3em;
  margin-bottom: 20px;
}

.search-box {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
}

.search-input {
  padding: 10px;
  border-radius: 30px;
  border: none;
  width: 250px;
  font-size: 1em;
}

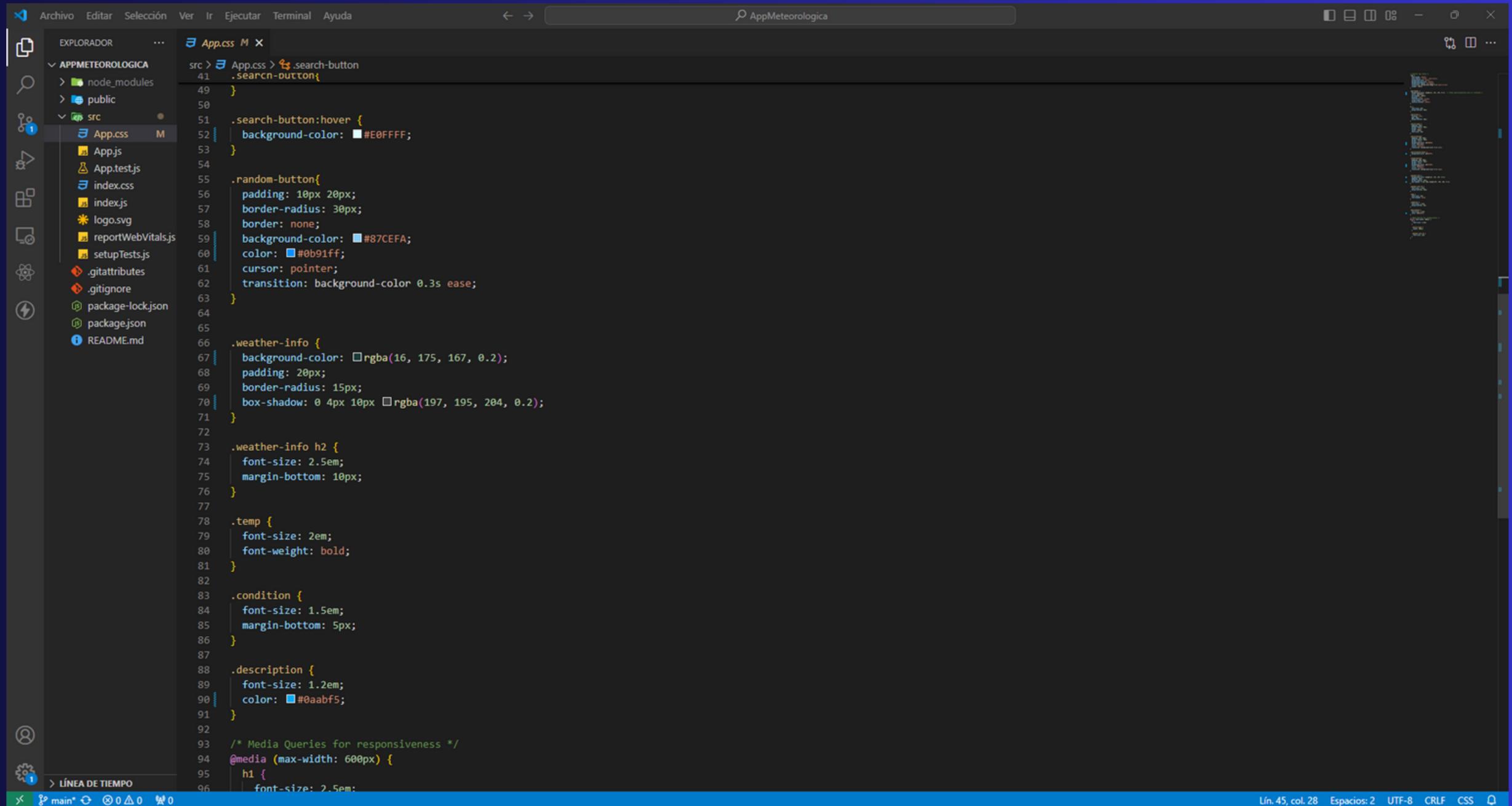
.search-button{
  padding: 10px 20px;
  border-radius: 30px;
  border: none;
  background-color: #ADD8E6;
  color: #1d7085;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
```

At the bottom of the editor, there is a status bar with the text "Lín. 45, col. 28 Espacios: 2 UTF-8 CRLF".



Css de la app

Aqui esta la
segunda parte
del css de la app
meteorologia

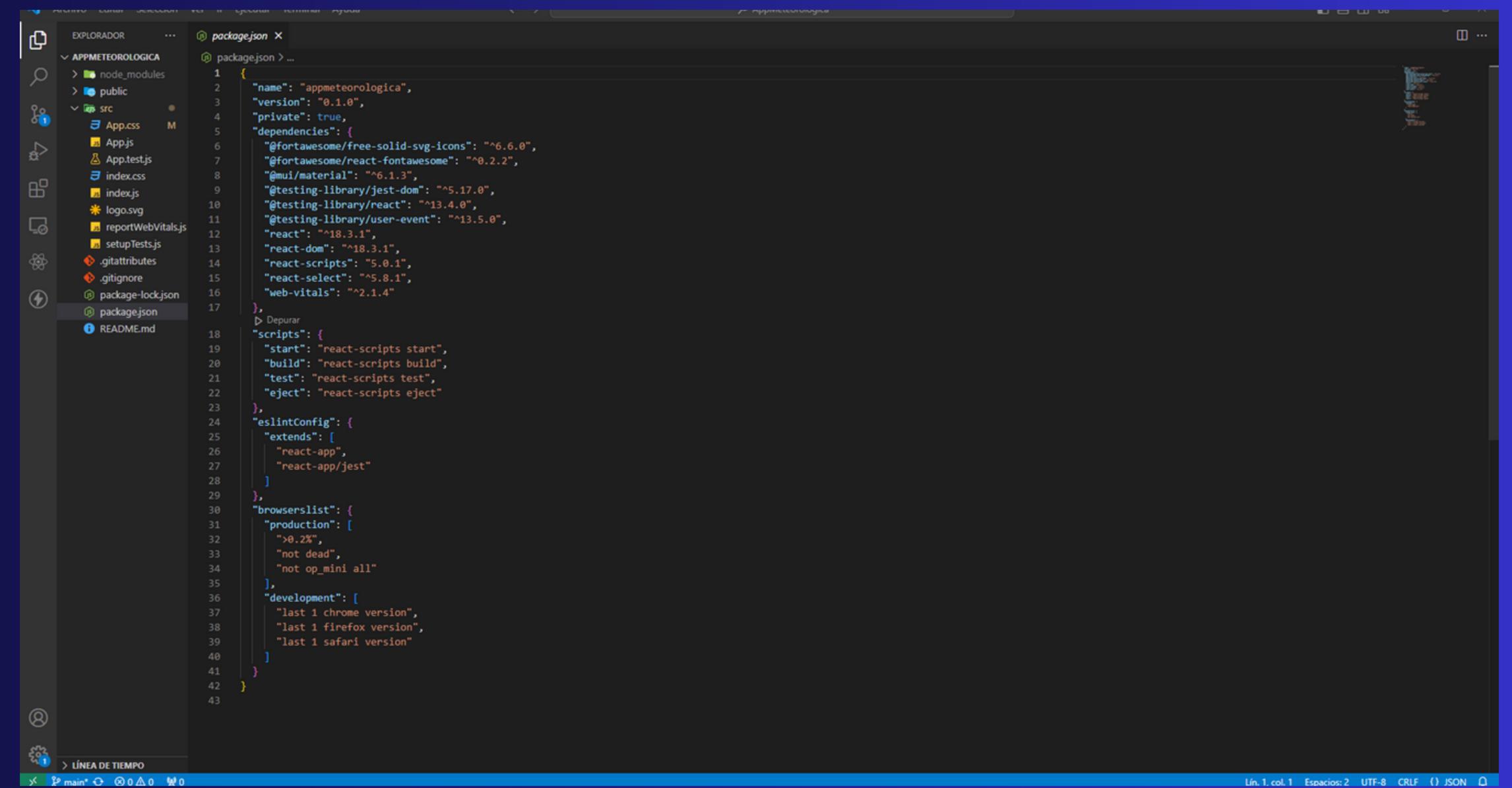


The screenshot shows a code editor window with the title bar "AppMeteorologica". The left sidebar displays a file tree for a project named "APPMETEOROLICA" containing "node_modules", "public", and "src" folders. Inside "src", there are files like "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", "setupTests.js", ".gitattributes", ".gitignore", "package-lock.json", "package.json", and "README.md". The main editor area is titled "App.css" and contains the following CSS code:

```
src > App.css > .search-button
41 .search-button{
42   }
43 .search-button:hover {
44   background-color: #E0FFFF;
45 }
46 .random-button{
47   padding: 10px 20px;
48   border-radius: 30px;
49   border: none;
50   background-color: #87CEFA;
51   color: #0b91ff;
52   cursor: pointer;
53   transition: background-color 0.3s ease;
54 }
55 .weather-info {
56   background-color: rgba(16, 175, 167, 0.2);
57   padding: 20px;
58   border-radius: 15px;
59   box-shadow: 0 4px 10px rgba(197, 195, 204, 0.2);
60 }
61 .weather-info h2 {
62   font-size: 2.5em;
63   margin-bottom: 10px;
64 }
65 .temp {
66   font-size: 2em;
67   font-weight: bold;
68 }
69 .condition {
70   font-size: 1.5em;
71   margin-bottom: 5px;
72 }
73 .description {
74   font-size: 1.2em;
75   color: #0aabf5;
76 }
77 /* Media Queries for responsiveness */
78 @media (max-width: 600px) {
79   h1 {
80     font-size: 2.5em;
81   }
82 }
```

The status bar at the bottom indicates "Lín. 45, col. 28 Espacios: 2 UTF-8 CRLF CSS".

Package.json



The screenshot shows a code editor interface with a dark theme. On the left is a file explorer sidebar titled "EXPLORADOR" showing the project structure:

- APPMETEOROLOGICA
- node_modules
- public
- src
 - App.css
 - App.js
 - App.test.js
 - index.css
 - index.js
 - logo.svg
 - reportWebVitals.js
 - setupTests.js
- .gitattributes
- .gitignore
- package-lock.json
- package.json
- README.md

The main editor area displays the contents of the "package.json" file:

```
1  {
2    "name": "appmeteorologica",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@fortawesome/free-solid-svg-icons": "^6.6.0",
7      "@fortawesome/react-fontawesome": "^0.2.2",
8      "@mui/material": "6.1.3",
9      "@testing-library/jest-dom": "^5.17.0",
10     "@testing-library/react": "13.4.0",
11     "@testing-library/user-event": "13.5.0",
12     "react": "18.3.1",
13     "react-dom": "18.3.1",
14     "react-scripts": "5.0.1",
15     "react-select": "5.8.1",
16     "web-vitals": "2.1.4"
17   },
18   "scripts": {
19     "start": "react-scripts start",
20     "build": "react-scripts build",
21     "test": "react-scripts test",
22     "eject": "react-scripts eject"
23   },
24   "eslintConfig": {
25     "extends": [
26       "react-app",
27       "react-app/jest"
28     ]
29   },
30   "browserslist": {
31     "production": [
32       ">0.2%",
33       "not dead",
34       "not op_mini all"
35     ],
36     "development": [
37       "last 1 chrome version",
38       "last 1 firefox version",
39       "last 1 safari version"
40     ]
41   }
42 }
43 }
```

At the bottom of the editor, there are status bars for "LÍNEA DE TIEMPO", "main", "0 0 0 0", and "JSON".

css

css

APPMETEORLOGICA

src > App.css > search-button

```
55 .random-button{  
56 | background-color: #F87CEFA;  
57 | color: #0091FF;  
58 | cursor: pointer;  
59 | transition: background-color 0.3s ease;  
60 }  
61  
62 .weather-info {  
63 | background-color: #rgba(16, 175, 167, 0.2);  
64 | padding: 20px;  
65 | border-radius: 15px;  
66 | box-shadow: 0 4px 18px #rgba(197, 195, 204, 0.2);  
67 }  
68  
69 .weather-info h2 {  
70 | font-size: 2.5em;  
71 | margin-bottom: 10px;  
72 }  
73  
74 .temp {  
75 | font-size: 2em;  
76 | font-weight: bold;  
77 }  
78  
79 .condition {  
80 | font-size: 1.5em;  
81 | margin-bottom: 5px;  
82 }  
83  
84 .description {  
85 | font-size: 1.2em;  
86 | color: #8aab5f;  
87 }  
88  
89 /* Media Queries for responsiveness */  
90 @media (max-width: 600px) {  
91 | h1 {  
92 | | font-size: 2.5em;  
93 | }  
94  
95 .search-input {  
96 | width: 200px;  
97 }  
98  
99 .weather-info h2 {  
100 | font-size: 2em;  
101 }  
102  
103 .weather-info h2 {  
104 | font-size: 2em;  
105 }  
106 }  
107
```

Gracias

