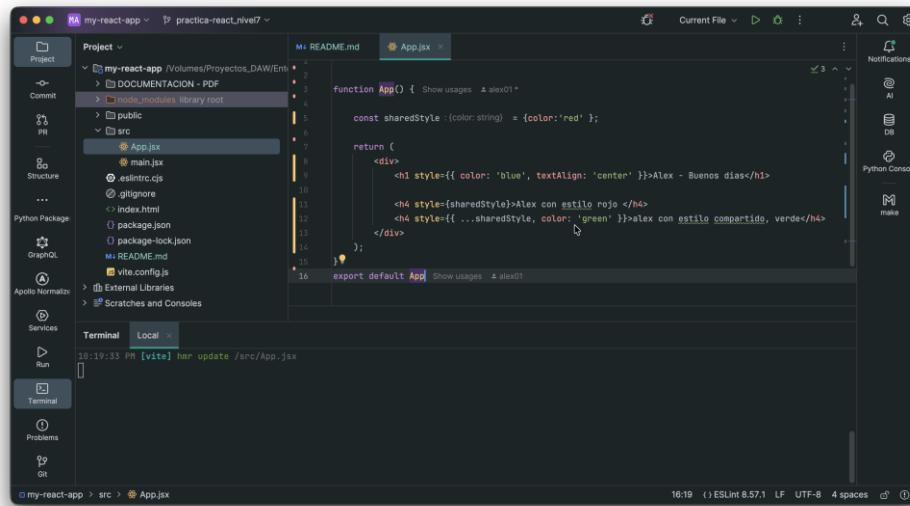


## 1- Parte A: Estilo en línea (inline styles)

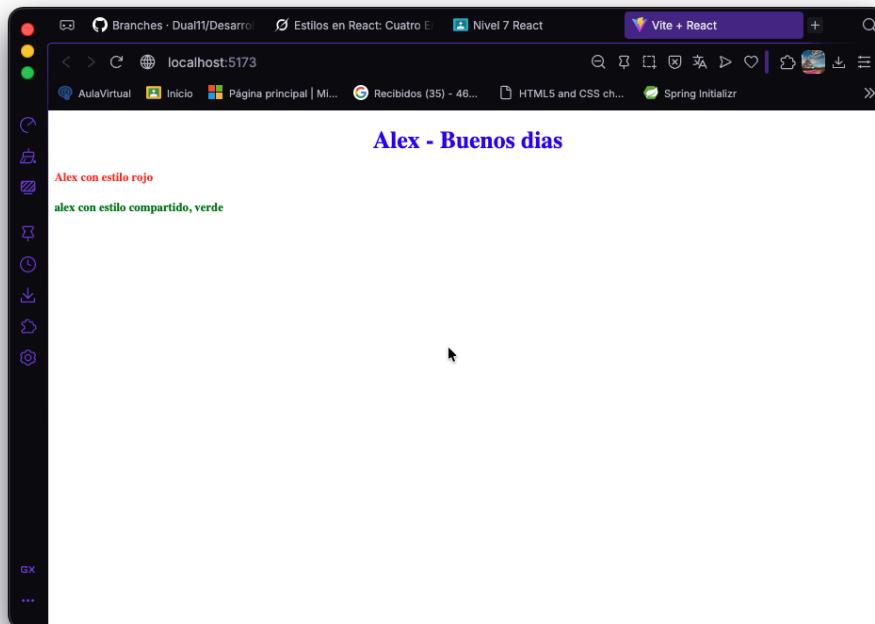


The screenshot shows the VS Code interface with the project 'my-react-app' open. The 'App.jsx' file is the active editor. The code contains inline styles for an H1 and an H4 element:

```
function App() {  const sharedStyle :{color:string} = {color:'red'};  return (    <div>      <h1 style={ color: 'blue', textAlign: 'center' }>Alex - Buenos dias</h1>      <h4 style={[sharedStyle,color]}>alex con estilo rojo</h4>      <h4 style={ ...sharedStyle, color: 'green' }>alex con estilo compartido, verde</h4>    </div>  );}  export default App;
```

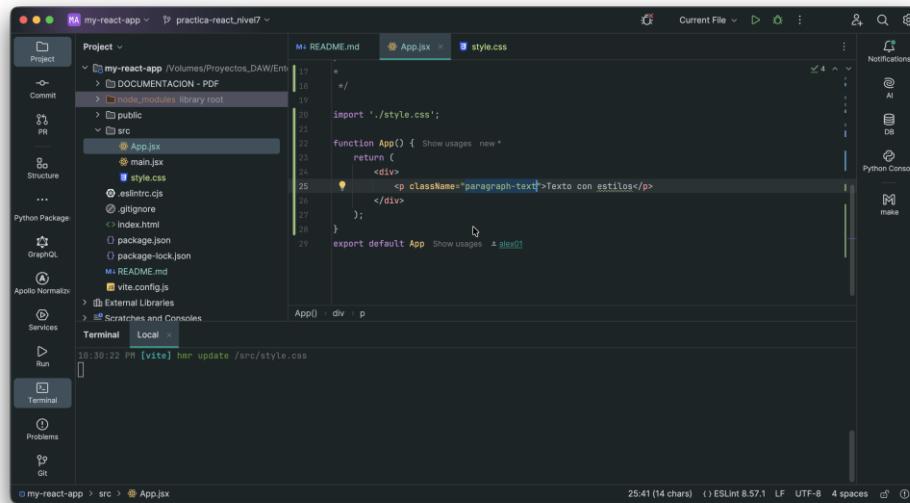
The terminal at the bottom shows the command: `10:19:33 PM [vite] hmr update /src/App.jsx`.

### - Resultado



## 2- Parte B: Archivos CSS (global o por componente)

- En el archivo css, he añadido el color rojo para ver que funciona



```
Project my-react-app /Volumes/Proyectos_DAW/Entrega_1/practica-react_level7
  - Commit
  - PR
  - Structure
  ...
  Python Package
  - GraphQL
  - Apollo Normalizer
  - Services
  - Run
  - Terminal
  - Problems
  - Git

  my-react-app /Volumes/Proyectos_DAW/Entrega_1/practica-react_level7
    - DOCUMENTACION - PDF
    - node_modules library root
    - public
    - src
      - App.jsx
      - main.jsx
      - style.css
      - .eslintrc.cjs
      - .gitignore
      - index.html
      - package.json
      - package-lock.json
      - README.md
      - vite.config.js
      - External Libraries
      - Scratches and Consoles
    - Terminal Local x
      10:30:22 PM [vite] hmr update /src/style.css
    - Local x
      10:30:22 PM [vite] hmr update /src/style.css

  Current File App.jsx style.css
  Notifications AI DB Python Console make

  style.css
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29

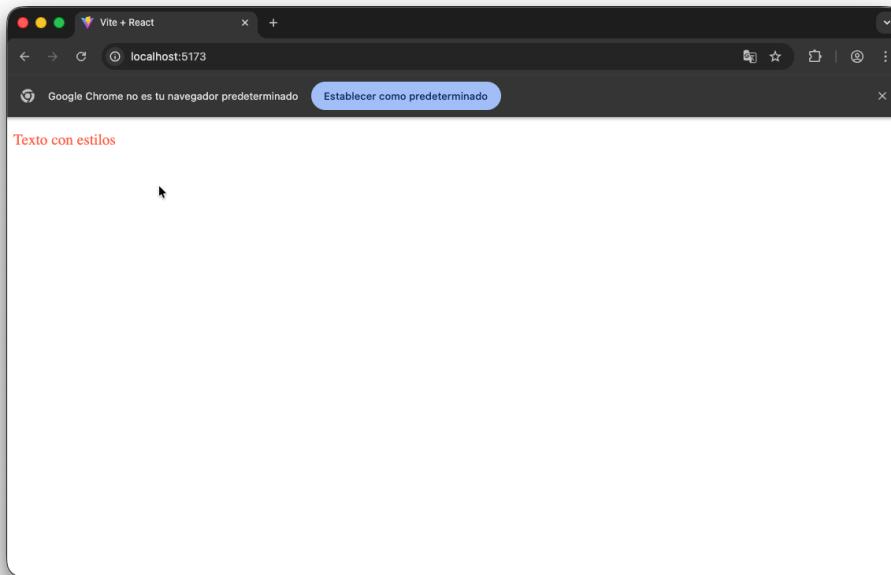
  import './style.css';

  function App() { Show usages new *
    return (
      <div>
        <p className="paragraph-text">Texto con estilos</p>
      </div>
    );
  }

  export default App Show usages & alex07

  App() div p
  25:41 (14 chars) ( ) ESLint 8.57.1 LF UTF-8 4 spaces
```

- **Resultado**



### 3- Parte C: CSS Modules (alcance local por defecto)

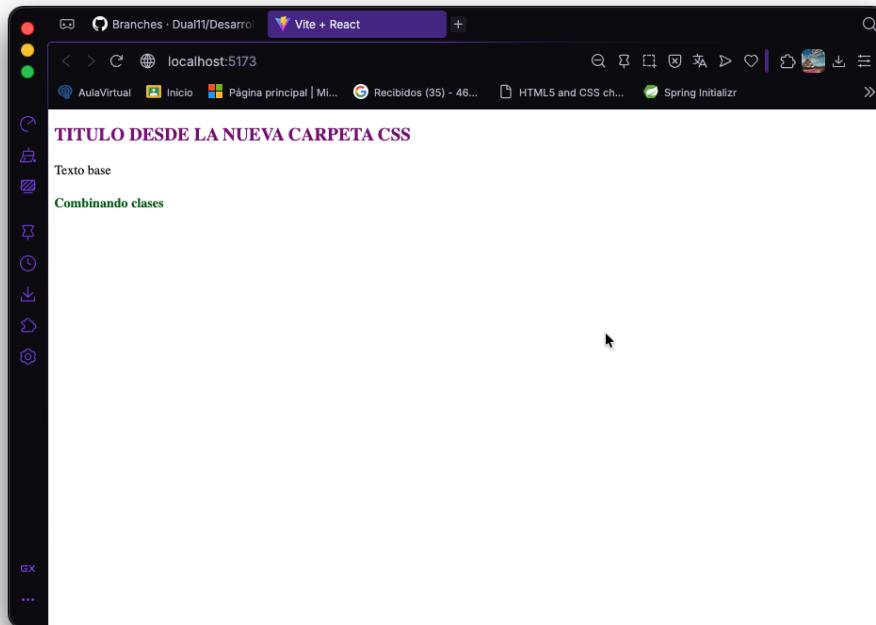
```

import styles from './App.module.css';

function App() {
  return (
    <div>
      <h2 className={styles.title}>Título desde la nueva carpeta CSS</h2>
      <p className={styles.baseText}>Texto base</p>
      <p className={`${styles.baseText} ${styles.special}`}>
        Combinando clases
      </p>
    </div>
  );
}

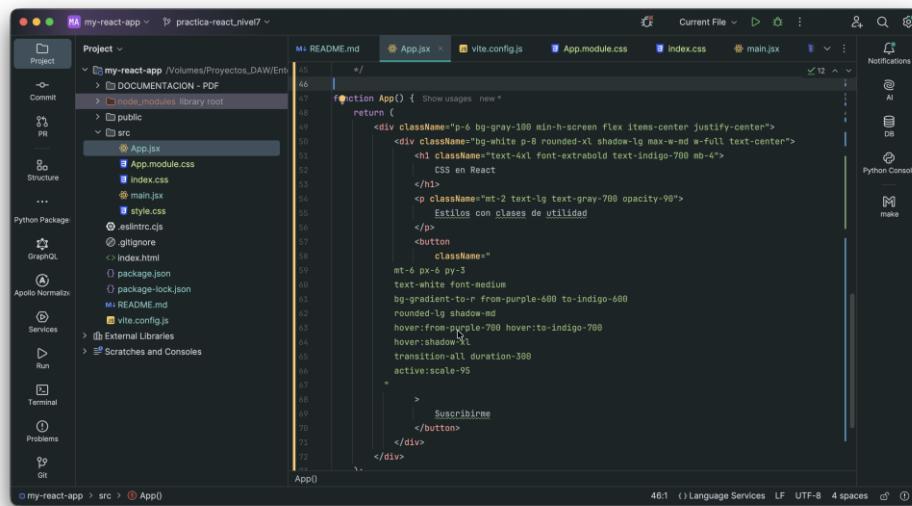
export default App;
  
```

## - Resultado



## 4- Parte D: Tailwind CSS (utilidades)

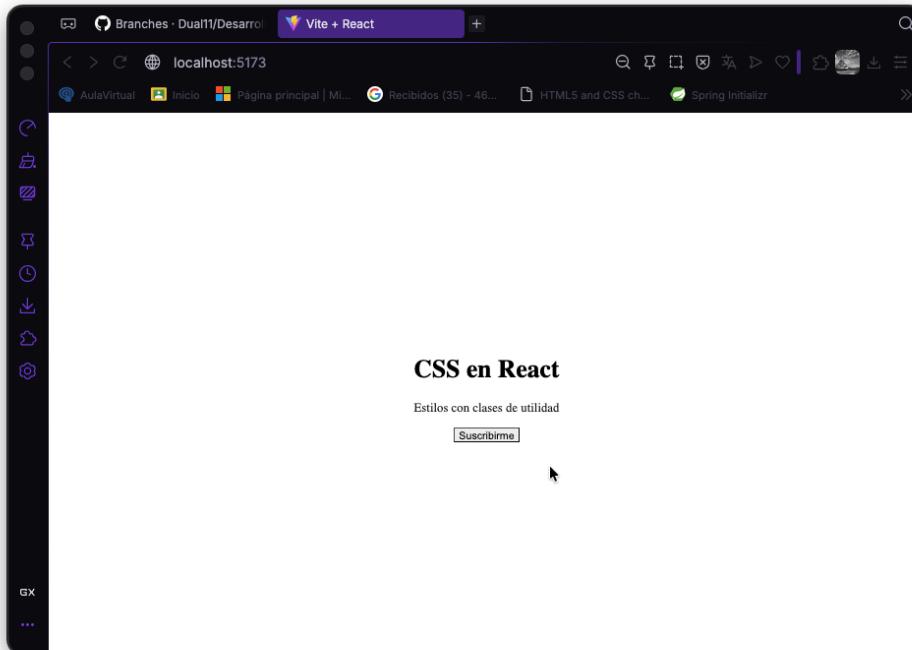
- En el archivo index.css, he añadido los componentes y utilidades de Tailwind.
- Luego en main.jsx, lo he importado para que este globalmente.



```
function App() { Show usages new *
  return (
    <div className="h-6 hq-sqv-100 min-h-screen flex items-center justify-center">
      <h1> CSS en React </h1>
      <p> Estilos con clases de utilidad </p>
      <button>
        <span> Suscríbeme </span>
      </button>
    </div>
  )
}

App()
```

- Resultado: esto me ha recordado a Bootstrap.



## 5- Miniretillo

```
function App() {  
  return (  
    <div className="min-h-screen bg-gray-50 flex items-center justify-center p-4 sm:p-6">  
      <div className="bg-white rounded-2xl shadow-xl overflow-hidden max-w-md w-full transition-all">  
        <h2 className="text-2xl sm:text-3xl font-bold text-gray-800 mb-3">  
          Tarjeta Increíble  
        </h2>  
  
        <p className="text-sm text-gray-600 mb-6 text-base sm:text-lg">  
          Descripción corta. Hover en botón + responsive.  
        </p>  
  
        <p className="text-sm text-gray-500 mb-4">  
          Likes: <strong>{likes}</strong>  
        </p>  
  
        <button onClick={() => setLikes(likes + 1)}>  
          <span>+</span>  
        </button>  
      </div>  
    </div>  
  );  
}  
  
export default App;
```

## - Resultado

