

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN-LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍAS



Carrera: Ingeniería en Sistemas de la Información

Componente: Programación orientada a la web II

Grupo: GP2

Tema: Componentes en React

Elaborado por:

- Wensess Aarón Montalván López

Docente: Juan Leyton

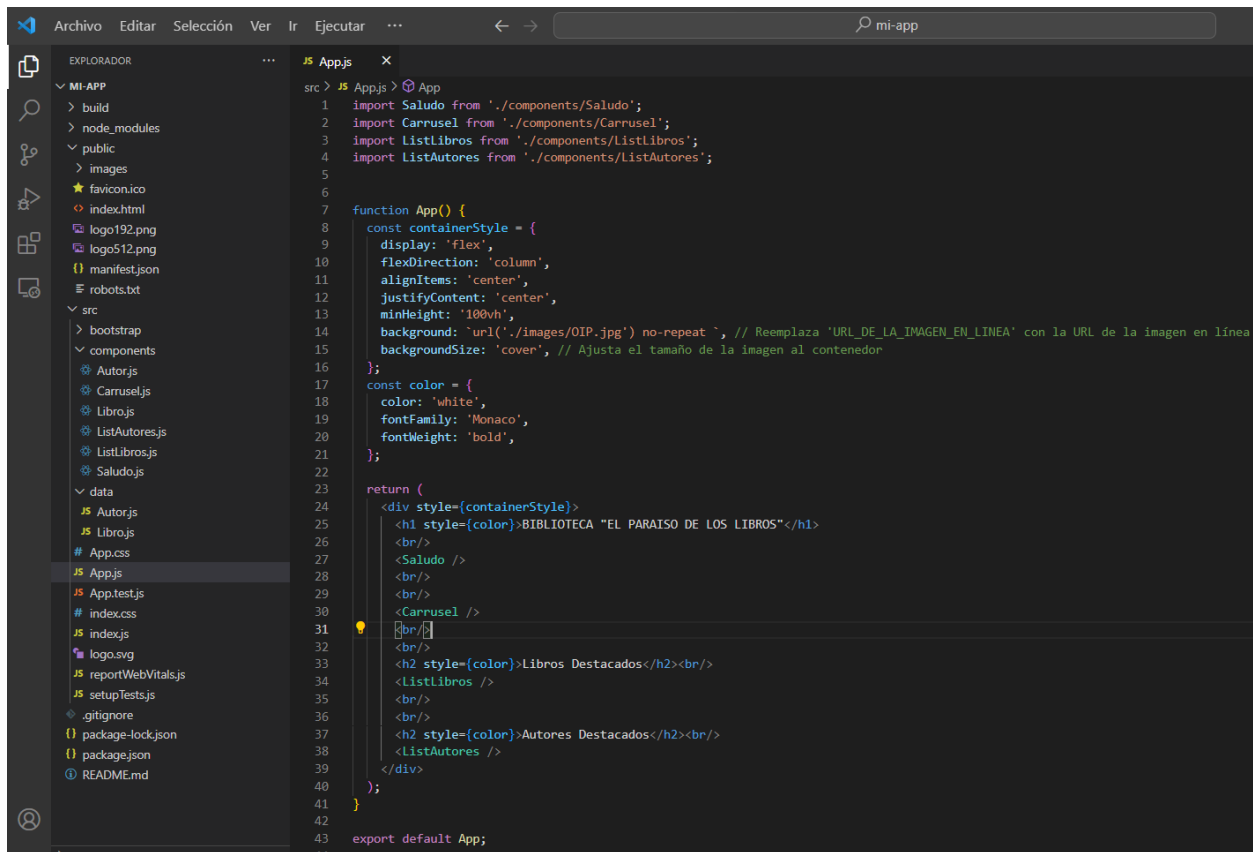
03 de octubre del 2023

“A la libertad por la universidad”

Introducción

El proyecto en React titulado "BIBLIOTECA 'EL PARAISO DE LOS LIBROS'" es una aplicación web diseñada para proporcionar a los usuarios una experiencia interactiva y atractiva relacionada con la lectura y la literatura. La aplicación está diseñada para mostrar una colección de libros destacados, así como autores destacados, en un entorno virtual que busca atraer a los amantes de la lectura y promover la exploración de obras literarias.

En este informe, presentaremos una visión general del proyecto, incluyendo su estructura y funcionalidades clave. Comenzaremos por analizar el archivo `App.js`, que sirve como punto de entrada principal de la aplicación React. Este archivo contiene la estructura principal de la interfaz de usuario, la cual incluye un encabezado con el nombre de la biblioteca y secciones para mostrar saludos, un carrusel de imágenes, libros destacados y autores destacados.



```
1 import Saludo from './components/Saludo';
2 import Carrusel from './components/Carrusel';
3 import ListLibros from './components/ListLibros';
4 import ListAutores from './components/ListAutores';
5
6
7 function App() {
8   const containerStyle = {
9     display: 'flex',
10    flexDirection: 'column',
11    alignItems: 'center',
12    justifyContent: 'center',
13    minHeight: '100vh',
14    background: 'url("./images/OIP.jpg") no-repeat', // Reemplaza 'URL_DE_LA_IMAGEN_EN_LINEA' con la URL de la imagen en línea
15    backgroundSize: 'cover', // Ajusta el tamaño de la imagen al contenedor
16  };
17   const color = {
18     color: 'white',
19     fontFamily: 'Monaco',
20     fontWeight: 'bold',
21   };
22
23   return (
24     <div style={containerStyle}>
25       <h1 style={color}>BIBLIOTECA "EL PARAISO DE LOS LIBROS"</h1>
26       <br/>
27       <Saludo />
28       <br/>
29       <br/>
30       <Carrusel />
31       <br/>
32       <br/>
33       <h2 style={color}>Libros Destacados</h2><br/>
34       <ListLibros />
35       <br/>
36       <br/>
37       <h2 style={color}>Autores Destacados</h2><br/>
38       <ListAutores />
39     </div>
40   );
41 }
42
43 export default App;
```

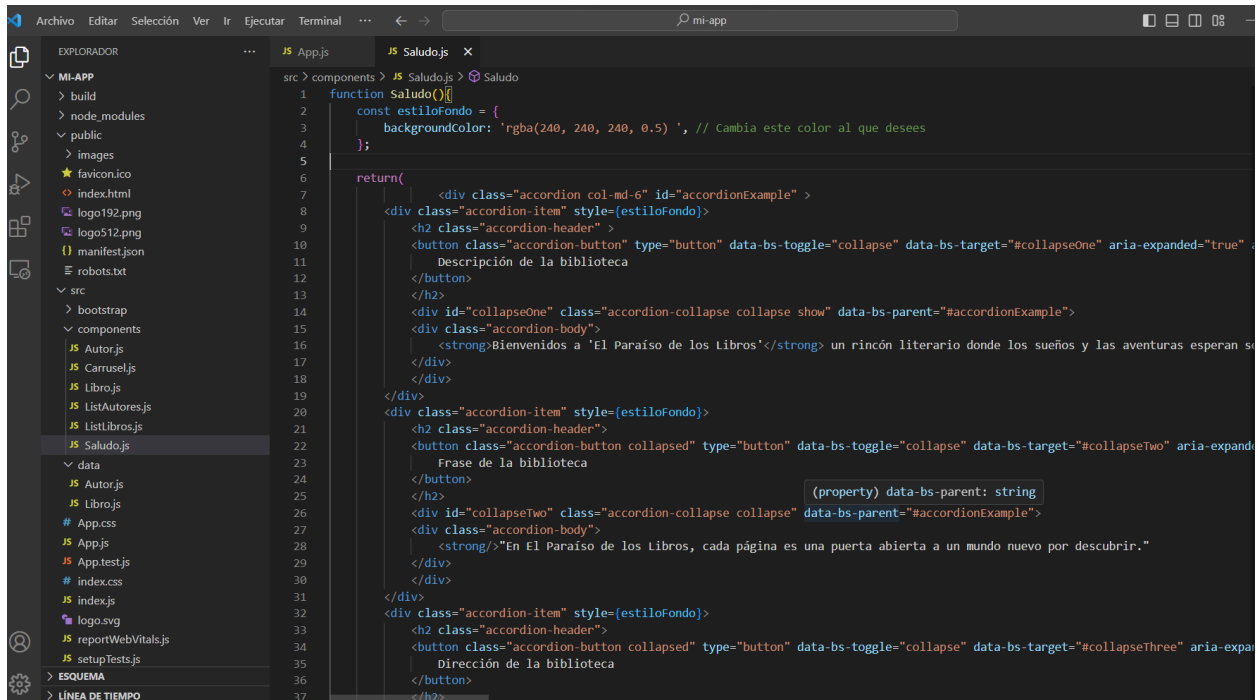
A medida que avancemos en el desarrollo del informe, exploraremos otros componentes y archivos que componen esta aplicación React y explicaremos cómo interactúan entre sí para brindar la experiencia final a los usuarios.

Desarrollo

Componente `Saludo.js`:

El componente `Saludo.js` es uno de los elementos principales de la aplicación. Este componente se encarga de mostrar información relevante sobre la biblioteca "El Paraíso de los Libros". A continuación, se detallan las características y el funcionamiento de este componente:

- **Estilo de Fondo:** Se define un estilo de fondo para el componente `Saludo.js` mediante la variable `estiloFondo`, que establece un fondo semitransparente.
- **Accordion:** El componente utiliza la estructura de un acordeón para mostrar tres secciones de información diferentes: Descripción de la biblioteca, Frase de la biblioteca y Dirección de la biblioteca.
- **Descripción de la biblioteca:** Esta sección proporciona una introducción a la biblioteca, destacando su propósito y su compromiso con los amantes de la lectura. Se utiliza un botón que permite expandir o colapsar el contenido para una mejor visualización.
- **Frase de la biblioteca:** En esta sección se presenta una frase inspiradora relacionada con la biblioteca. También se utiliza un botón para expandir o colapsar el contenido.
- **Dirección de la biblioteca:** Por último, se muestra la dirección física de la biblioteca. Al igual que las secciones anteriores, esta información se encuentra dentro de un elemento colapsable.



```
1 function Saludo() {
2   const estiloFondo = {
3     backgroundColor: 'rgba(240, 240, 240, 0.5)', // Cambia este color al que desees
4   };
5
6   return(
7     <div class="accordion col-md-6" id="accordionExample">
8       <div class="accordion-item" style={estiloFondo}>
9         <h2 class="accordion-header">
10           <button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#collapseOne" aria-expanded="true">
11             Descripción de la biblioteca
12           </button>
13         </h2>
14         <div id="collapseOne" class="accordion-collapse collapse show" data-bs-parent="#accordionExample">
15           <div class="accordion-body">
16             <strong>Bienvenidos a 'El Paraíso de los Libros'</strong> un rincón literario donde los sueños y las aventuras esperan ser descubiertos.
17           </div>
18         </div>
19       </div>
20       <div class="accordion-item" style={estiloFondo}>
21         <h2 class="accordion-header">
22           <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false">
23             Frase de la biblioteca
24           </button>
25         </h2>
26         <div id="collapseTwo" class="accordion-collapse collapse" data-bs-parent="#accordionExample">
27           <div class="accordion-body">
28             <strong>"En El Paraíso de los Libros, cada página es una puerta abierta a un mundo nuevo por descubrir."</strong>
29           </div>
30         </div>
31       </div>
32       <div class="accordion-item" style={estiloFondo}>
33         <h2 class="accordion-header">
34           <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false">
35             Dirección de la biblioteca
36           </button>
37         </h2>
```

```
components 38
JS Autor.js 39
JS Carrusel.js 40
JS Libro.js 41
JS ListAutores.js 42
JS ListLibros.js 43
JS Saludo.js 44
data 45
  JS Autor.js 46
  JS Libro.js 47
# App.css 48
export default Saludo;
```

Componente Carrusel.js:

El componente Carrusel.js es responsable de mostrar un carrusel de imágenes en la aplicación. Su propósito es destacar visualmente la biblioteca y atraer la atención de los usuarios. A continuación, se describen las características y el funcionamiento de este componente:

- **Estilo de Imágenes:** Se define un estilo para las imágenes del carrusel a través de la variable `imageStyle`, que establece la altura deseada de las imágenes.
- **Carrusel de Imágenes:** El componente utiliza la estructura de un carrusel Bootstrap para mostrar tres imágenes diferentes. Cada imagen se encuentra dentro de un elemento de carrusel (`carousel-item`), y la primera imagen se establece como activa para que se muestre por defecto.
- **Botones de Control:** Se incluyen botones de control para avanzar y retroceder en las imágenes del carrusel. Estos botones permiten a los usuarios navegar a través de las imágenes de manera manual.
- **Intervalo Automático:** El carrusel está configurado para cambiar automáticamente de imagen cada 2 segundos (`data-interval="2000"`), lo que añade un elemento de dinamismo a la página de inicio de la biblioteca.

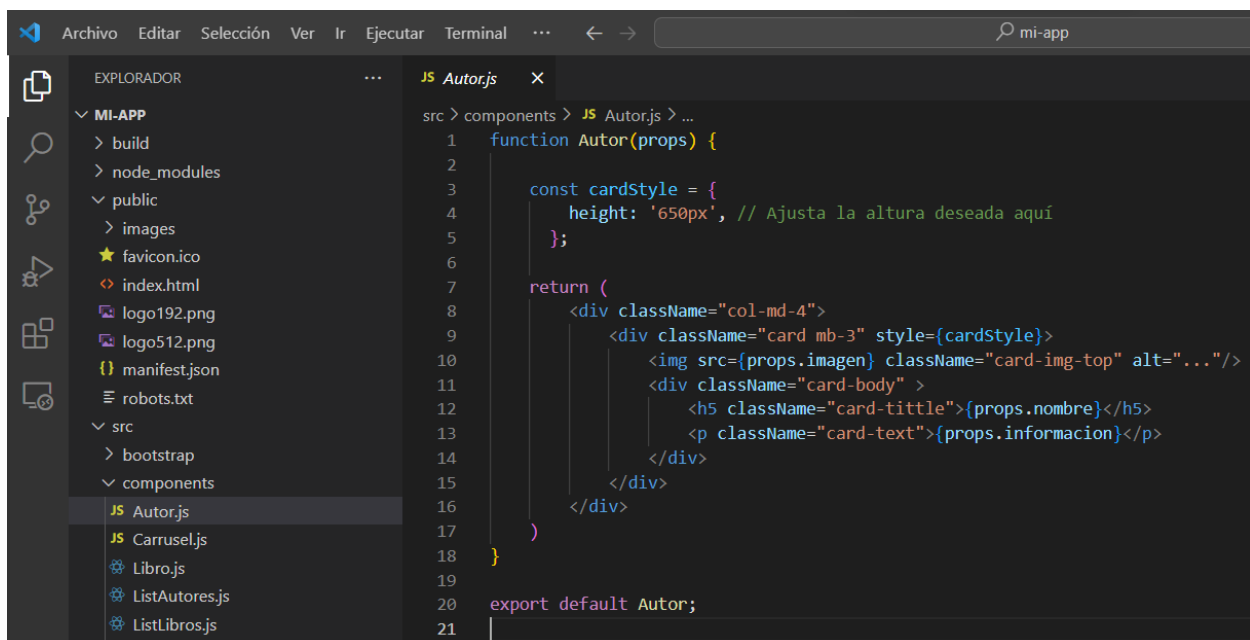
```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  ...  mi-app
EXPLORADOR  JS App.js  JS Carrusel.js X
MI-APP  src > components > JS Carrusel.js > Carrusel
> build
> node_modules
public
  > images
  ★ faviconico
  < index.html
  < logo192.png
  < logo512.png
  < manifest.json
  robots.txt
src
  > bootstrap
  > components
  JS Autor.js
  JS Carrusel.js
  JS Libro.js
  JS ListAutores.js
  JS ListLibros.js
  JS Saludo.js
  > data
  JS Autor.js
  JS Libro.js
# App.css
JS App.js
JS App.test.js
# index.css
# index.js
```

```
1 function Carrusel() {
2   const imageStyle = {
3     height: '500px', // Ajusta la altura deseada aquí
4   };
5
6   return (
7     <div id="carouselExampleControls" className="col-md-6 carousel slide" data-bs-ride="carousel" data-interval="2000">
8       <div className="carousel-inner">
9         <div className="carousel-item active">
10           
11         </div>
12         <div className="carousel-item">
13           
14         </div>
15         <div className="carousel-item">
16           
17         </div>
18       </div>
19       <button className="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="prev">
20         <span className="carousel-control-prev-icon" aria-hidden="true"></span>
21         <span className="visually-hidden">Previous</span>
22       </button>
23       <button className="carousel-control-next" type="button" data-bs-target="#carouselExampleControls" data-bs-slide="next">
24         <span className="carousel-control-next-icon" aria-hidden="true"></span>
25         <span className="visually-hidden">Next</span>
26       </button>
27     </div>
28   );
29 }
30
31 export default Carrusel;
```

Componente `Autor.js`:

El componente `Autor.js` es un componente de React que recibe propiedades (props) y muestra información sobre un autor específico en una tarjeta estilo card. A continuación, se detallan las características y el funcionamiento de este componente:

- **Propiedades (props):** Este componente recibe tres propiedades: `nombre`, `informacion`, e `imagen`. Estas propiedades se utilizan para mostrar el nombre del autor, información sobre el autor y una imagen del autor respectivamente.
- **Estilo de Tarjeta:** Se define un estilo para la tarjeta de autor a través de la variable `cardStyle`. Este estilo puede ajustarse según las preferencias de diseño.
- **Renderización de la Tarjeta:** El componente renderiza la información del autor en una tarjeta utilizando Bootstrap. La imagen del autor se muestra en la parte superior de la tarjeta, seguida del nombre del autor y una breve descripción.

A screenshot of a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) sidebar shows a file tree for a project named 'MI-APP'. The tree includes folders like 'build', 'node_modules', 'public', and 'src', along with various files like 'favicon.ico', 'index.html', and 'manifest.json'. The 'src' folder is expanded, showing subfolders 'bootstrap' and 'components'. The 'components' folder is selected, and the file 'JS Autor.js' is open in the main editor. The code in the editor defines a function 'Autor(props)' that returns a JSX element. It uses a 'cardStyle' object with a 'height' property. The JSX element consists of a 'div' with 'col-md-4' class, containing another 'div' with 'card mb-3' class and the 'cardStyle' prop. Inside this card, there is an 'img' tag for the author's image, followed by a 'div' with 'card-body' class containing an 'h5' tag for the author's name and a 'p' tag for the author's information. The code ends with 'export default Autor;'.

```
1 function Autor(props) {
2
3     const cardStyle = {
4         height: '650px', // Ajusta la altura deseada aquí
5     };
6
7     return (
8         <div className="col-md-4">
9             <div className="card mb-3" style={cardStyle}>
10                 <img src={props.imagen} className="card-img-top" alt="..." />
11                 <div className="card-body">
12                     <h5 className="card-tittle">{props.nombre}</h5>
13                     <p className="card-text">{props.informacion}</p>
14                 </div>
15             </div>
16         </div>
17     )
18 }
19
20 export default Autor;
21
```

Componente `ListAutores.js`:

El componente `ListAutores.js` se encarga de mostrar una lista de autores en la aplicación. Utiliza el archivo de datos `dataAutor.js` para obtener la información de los autores y luego renderiza cada autor utilizando el componente `Autor.js`. A continuación, se describen las características y el funcionamiento de este componente:

- **Importación de Datos:** Se importa el archivo `dataAutor.js`, que contiene la información de los autores, utilizando la declaración `import`.
- **Mapeo de Datos:** Se utiliza la función `map` para recorrer el array de datos `dataAutor` y crear un componente `Autor` para cada autor. Las propiedades

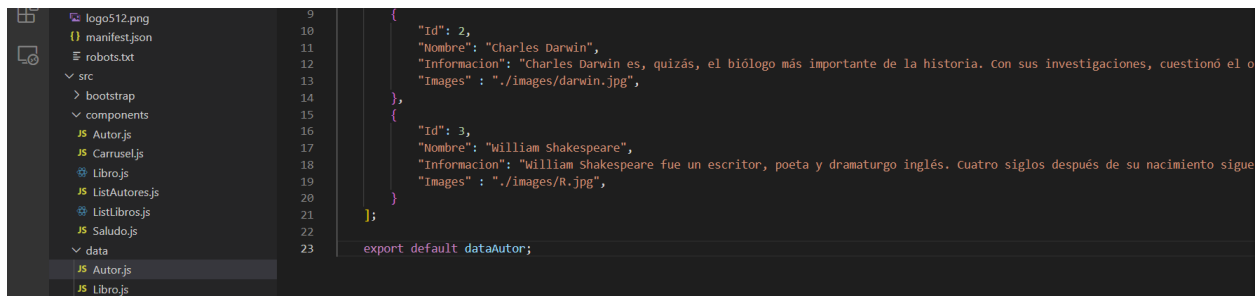
(`nombre`, `informacion`, e `imagen`) se pasan al componente `Autor` durante el mapeo.

- Renderización de Autores: El componente renderiza los autores en una fila (`row`) dentro de un contenedor de columnas (`col-md-6`), asegurando que los autores se muestren de manera ordenada en la interfaz de usuario.

Archivo `dataAutor.js`:

El archivo `dataAutor.js` contiene un array de objetos que representa la información de diferentes autores. Cada objeto contiene propiedades como `Nombre`, `Información` e `Images`, que se utilizan en el componente `ListAutores.js` para mostrar los datos de los autores.

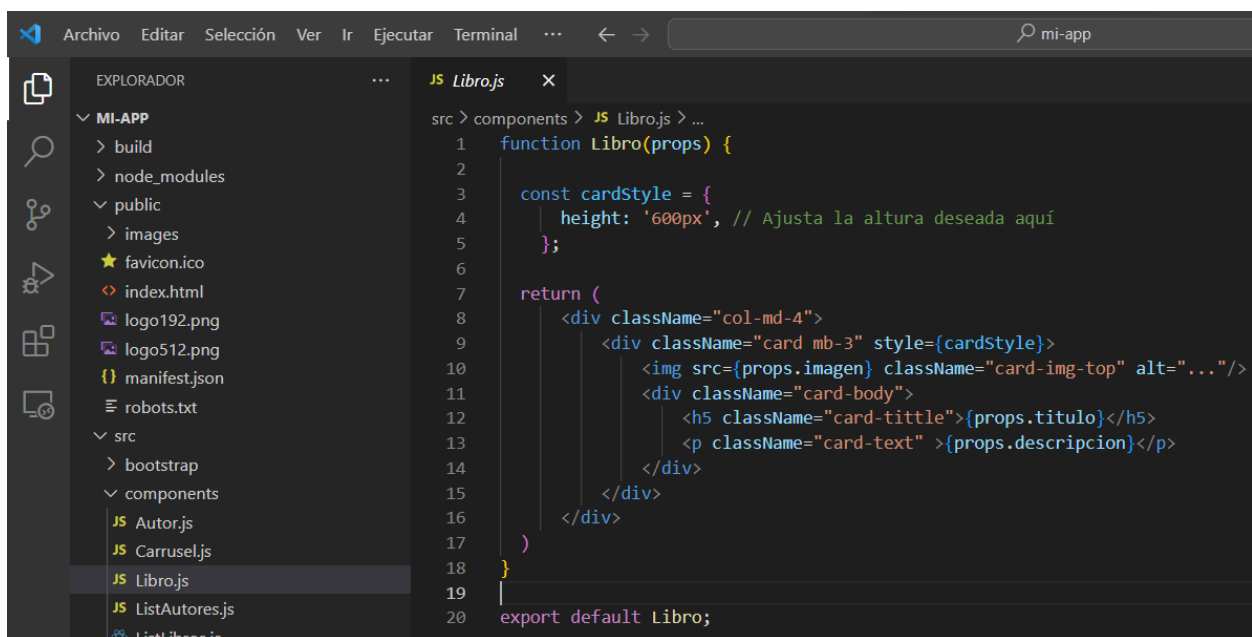
Con estos componentes y datos, la aplicación puede mostrar una lista de autores junto con sus respectivas tarjetas de información en la página de inicio.



Componente `Libro.js`:

El componente `Libro.js` es similar al componente `Autor.js`, pero está diseñado para mostrar información sobre libros en lugar de autores. A continuación, se detallan las características y el funcionamiento de este componente:

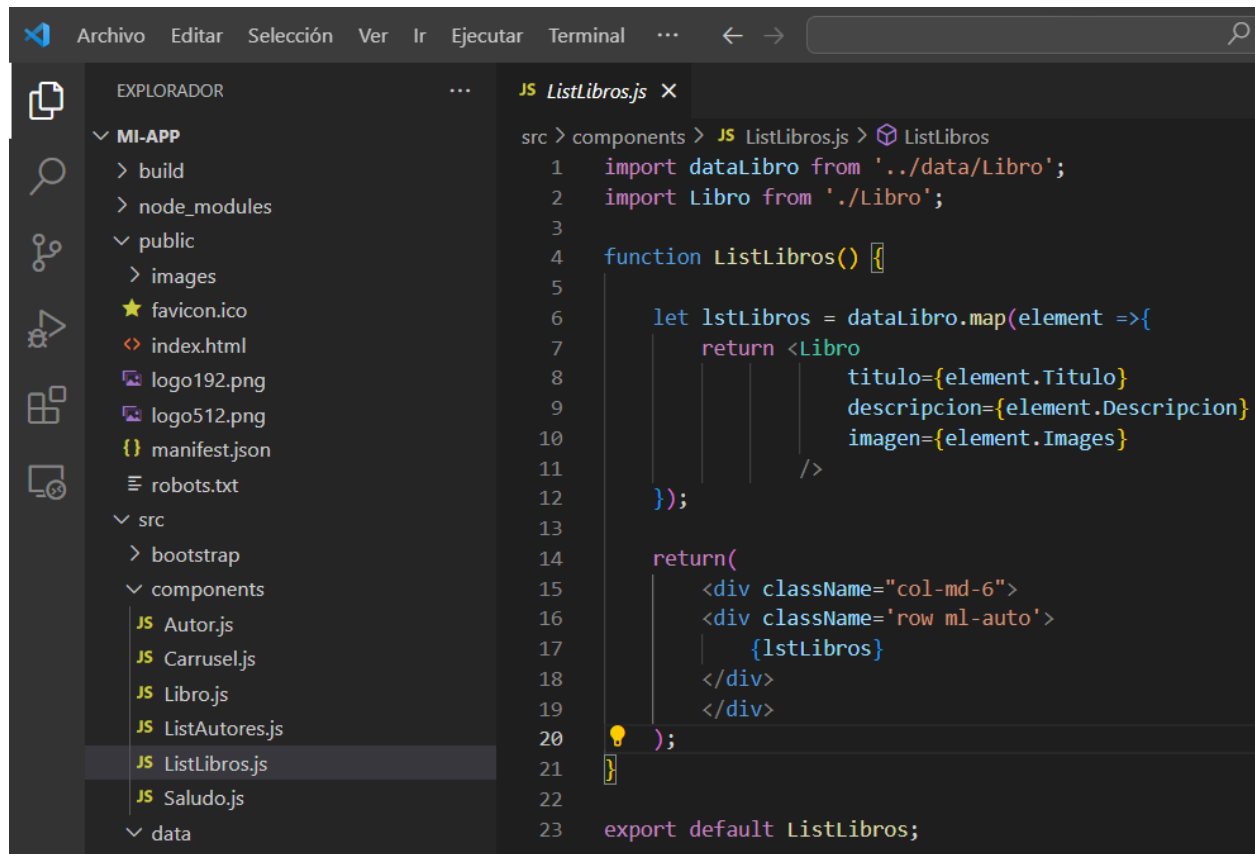
- **Propiedades (props):** Al igual que en `Autor.js`, este componente recibe propiedades (`titulo`, `descripcion` e `imagen`) que se utilizan para mostrar información sobre el libro.
- **Estilo de Tarjeta:** Se define un estilo para la tarjeta de libro a través de la variable `cardStyle`. Este estilo también puede ajustarse según las preferencias de diseño.
- **Renderización de la Tarjeta:** El componente renderiza la información del libro en una tarjeta similar a la utilizada en el componente `Autor.js`. Muestra la imagen del libro, el título del libro y una descripción del mismo.



Componente `ListLibros.js`:

El componente `ListLibros.js` tiene un propósito similar al componente `ListAutores.js`, pero se enfoca en mostrar una lista de libros en lugar de autores. A continuación, se describen las características y el funcionamiento de este componente:

- **Importación de Datos:** Al igual que en `ListAutores.js`, se importa el archivo `dataLibro.js`, que contiene la información de los libros, utilizando la declaración `import`.
- **Mapeo de Datos:** Se utiliza la función `map` para recorrer el array de datos `dataLibro` y crear un componente `Libro` para cada libro. Las propiedades (`titulo`, `descripcion`, e `imagen`) se pasan al componente `Libro` durante el mapeo.
- **Renderización de Libros:** El componente renderiza los libros en una fila (`row`) dentro de un contenedor de columnas (`col-md-6`), asegurando que los libros se muestren de manera ordenada en la interfaz de usuario.

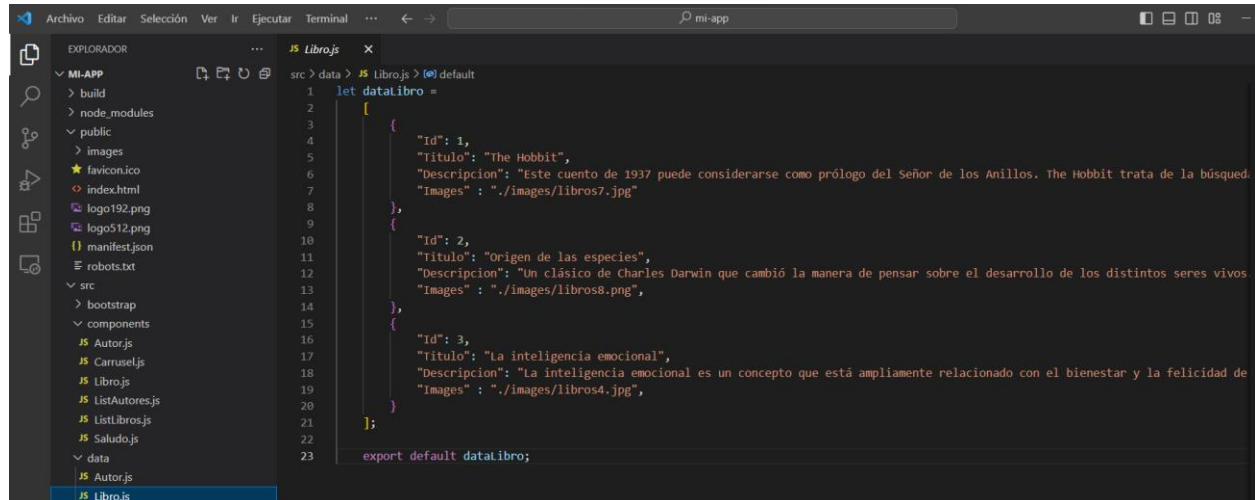


```
src > components > JS ListLibros.js > ListLibros
1  import dataLibro from '../data/Libro';
2  import Libro from './Libro';
3
4  function ListLibros() {
5
6      let lstLibros = dataLibro.map(element =>{
7          return <Libro
8              |
9              |      titulo={element.Titulo}
10             |      descripcion={element.Descripcion}
11             |      imagen={element.Images}
12             |      />
13         });
14
15         return(
16             <div className="col-md-6">
17                 <div className='row ml-auto'>
18                     {lstLibros}
19                 </div>
20             </div>
21         );
22     }
23     export default ListLibros;
```


Archivo `dataLibro.js`:

El archivo `dataLibro.js` contiene un array de objetos que representa la información de diferentes libros. Cada objeto contiene propiedades como `Titulo`, `Descripcion` e `Images`, que se utilizan en el componente `ListLibros.js` para mostrar los datos de los libros.

Con estos componentes y datos, la aplicación puede mostrar una lista de libros junto con sus respectivas tarjetas de información en la página de inicio.



```
1 let dataLibro =
2   [
3     {
4       "Id": 1,
5       "Titulo": "The Hobbit",
6       "Descripcion": "Este cuento de 1937 puede considerarse como prólogo del Señor de los Anillos. The Hobbit trata de la búsqueda del anillo.",
7       "Images" : "../images/libros7.jpg"
8     },
9     {
10      "Id": 2,
11      "Titulo": "Origen de las especies",
12      "Descripcion": "Un clásico de Charles Darwin que cambió la manera de pensar sobre el desarrollo de los distintos seres vivos.",
13      "Images" : "../images/libros8.png",
14    },
15    {
16      "Id": 3,
17      "Titulo": "La inteligencia emocional",
18      "Descripcion": "La inteligencia emocional es un concepto que está ampliamente relacionado con el bienestar y la felicidad de las personas.",
19      "Images" : "../images/libros4.jpg",
20    }
21  ];
22
23 export default dataLibro;
```

Conclusión

En el desarrollo de este proyecto en React titulado "BIBLIOTECA 'EL PARAISO DE LOS LIBROS'", hemos logrado crear una aplicación web interactiva que ofrece una experiencia atractiva y amigable para los amantes de la lectura y la literatura. A lo largo de este informe, hemos explorado varios componentes clave que componen la aplicación, cada uno desempeñando un papel importante en la presentación de información relevante.

El componente `App.js` sirve como el punto de entrada principal de la aplicación, proporcionando una estructura general para la interfaz de usuario. Presenta a los usuarios la biblioteca "El Paraíso de los Libros" a través de secciones interactivas como saludos, un carrusel de imágenes, libros destacados y autores destacados. El componente `Carrusel.js` añade un toque visual atractivo a la página de inicio con un carrusel de imágenes relacionadas con la lectura. Además, hemos explorado dos componentes que utilizan propiedades (props) para mostrar información dinámica. El componente `Autor.js` permite mostrar tarjetas de información sobre autores, mientras que el componente `Libro.js` ofrece una funcionalidad similar pero para libros. Ambos componentes son escalables y flexibles, lo que facilita la presentación de múltiples autores y libros.

La inclusión de archivos de datos como ``dataAutor.js`` y ``dataLibro.js`` proporciona una forma eficiente de gestionar y actualizar la información de autores y libros en la aplicación.

Es importante destacar que, en este proyecto, se ha utilizado Bootstrap para mejorar la estética y la estructura de la interfaz de usuario, lo que ha contribuido significativamente a la presentación visual de la biblioteca en línea. Además, se han aprovechado las imágenes almacenadas en la carpeta de "public" para enriquecer la experiencia de los usuarios, permitiendo que las portadas de los libros y las imágenes de los autores se muestren de manera atractiva en la aplicación.

En resumen, el proyecto en React "BIBLIOTECA 'EL PARAISO DE LOS LIBROS'" ha demostrado la capacidad de crear una aplicación web interactiva y visualmente atractiva para presentar información sobre autores y libros destacados. Este proyecto es un ejemplo de cómo React, Bootstrap y una estructura de componentes bien diseñada pueden combinarse para ofrecer una experiencia en línea atractiva y accesible para los amantes de la literatura.