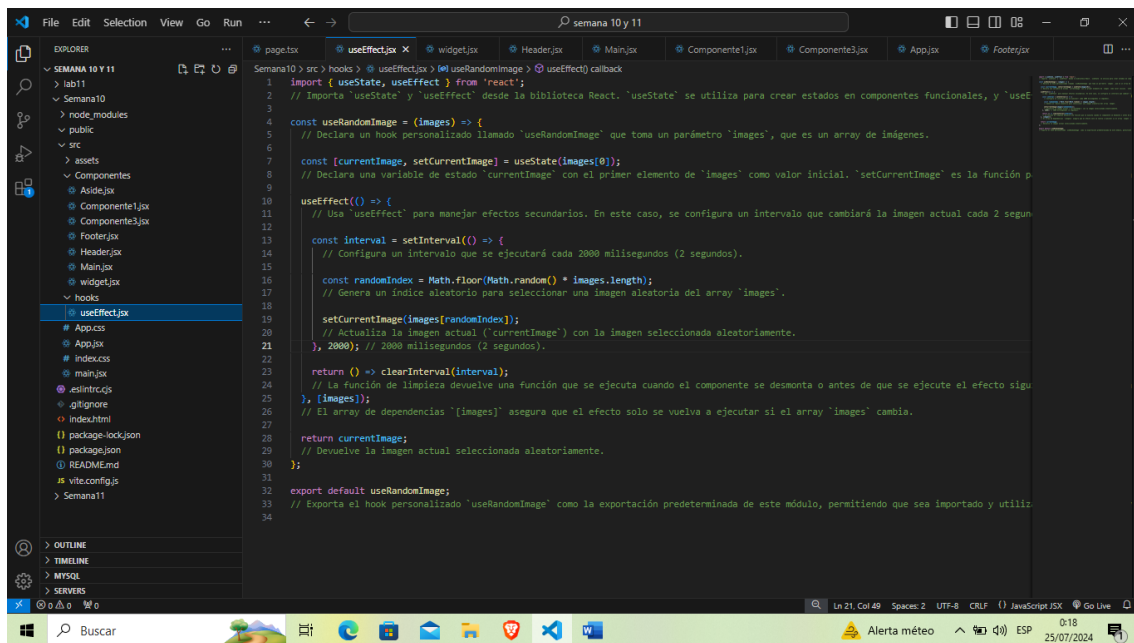


```
1 import React from 'react';
2 // Importa React desde la biblioteca React.
3
4 import useRandomImage from '../hooks/useEffect';
5 // Importa el hook personalizado useRandomImage desde el archivo 'useEffect' dentro de la carpeta 'hooks'.
6
7 import messi1 from '../assets/img/messi1.jpg';
8 import messi2 from '../assets/img/messi2.jpg';
9 import messi3 from '../assets/img/messi3.jpg';
10 import messi4 from '../assets/img/messi4.jpg';
11 // Importa varias imágenes desde la carpeta 'assets/img'.
12
13 const images = [messi1, messi2, messi3, messi4];
14 // Crea un arreglo llamado 'images' que contiene todas las imágenes importadas.
15
16 const ImageWidget = () => {
17   // Declara un componente funcional llamado 'ImageWidget'.
18
19   const currentImage = useRandomImage(images);
20   // Usa el hook useRandomImage con el arreglo 'images' para obtener una imagen aleatoria. La imagen seleccionada se guarda en 'currentImage'.
21
22   return (
23     // Devuelve el JSX que representa la estructura del componente.
24     <div
25       style={{
26         width: '150px',
27         height: '150px',
28         display: 'flex',
29         align-items: 'center',
30         justify-content: 'center',
31         overflow: 'hidden',
32         border: '1px solid #ccc',
33         backgroundColor: 'white', // Fondo blanco para ver la imagen correctamente
34       }}
35     >
36       {/* Un div con estilos en línea que define un contenedor de 150x150 píxeles, centrado y con borde gris claro. */}
37
38       <img
39         src={currentImage}
40         alt="Random"
41         style={{ width: '100%', height: '100%', objectFit: 'cover' }}
42       // Usa una imagen que sea 'currentImage' como fuente, con un texto alternativo 'Random'. La imagen se ajusta completamente al contenedor usando 'objectFit: cover'.
43     />
44   );
45 };
46
47 export default ImageWidget;
48 // Exporta el componente 'ImageWidget' como la exportación predeterminada de este módulo, permitiendo que sea importado y utilizado en otros archivos.
```



```
1 import { useState, useEffect } from 'react';
2 // Importa 'useState' y 'useEffect' desde la biblioteca React. 'useState' se utiliza para crear estados en componentes funcionales, y 'useEffect' para manejar efectos secundarios.
3
4 const useRandomImage = (images) => {
5   // Declara un hook personalizado llamado 'useRandomImage' que toma un parámetro 'images', que es un array de imágenes.
6
7   const [currentImage, setCurrentImage] = useState(images[0]);
8   // Declara una variable de estado 'currentImage' con el primer elemento de 'images' como valor inicial. 'setCurrentImage' es la función para actualizar el estado.
9
10  useEffect(() => {
11    // Usa 'useEffect' para manejar efectos secundarios. En este caso, se configura un intervalo que cambiará la imagen actual cada 2 segundos.
12
13    const interval = setInterval(() => {
14      // Configura un intervalo que se ejecutará cada 2000 milisegundos (2 segundos).
15
16      const randomIndex = Math.floor(Math.random() * images.length);
17      // Genera un índice aleatorio para seleccionar una imagen aleatoria del array 'images'.
18
19      setCurrentImage(images[randomIndex]);
20      // Actualiza la imagen actual ('currentImage') con la imagen seleccionada aleatoriamente.
21    }, 2000); // 2000 milisegundos (2 segundos).
22
23    return () => clearInterval(interval);
24    // La función de limpieza devuelve una función que se ejecuta cuando el componente se desmonta o antes de que se ejecute el efecto siguiente.
25  }, [images]);
26  // El array de dependencias '[images]' asegura que el efecto solo se vuelva a ejecutar si el array 'images' cambia.
27
28  return currentImage;
29  // Devuelve la imagen actual seleccionada aleatoriamente.
30 };
31
32 export default useRandomImage;
33 // Exporta el hook personalizado 'useRandomImage' como la exportación predeterminada de este módulo, permitiendo que sea importado y utilizado en otros archivos.
```

File Edit Selection View Go Run ...

semana 10 y 11

EXPLORER

- SEMANA 10 Y 11
  - lab11
  - Semana10
    - node\_modules
    - public
    - src
      - assets
      - Componentes
        - App.jsx
        - Componente1.jsx
        - Componente3.jsx
        - Footer.jsx
        - Header.jsx
        - Main.jsx
        - widget.jsx
      - hooks
      - useEffect.jsx
      - App.css
      - App.jsx
      - index.css
      - main.jsx
      - eslint.config.js
      - .gitignore
      - index.html
      - package-lock.json
      - package.json
      - README.md
      - vite.config.js
    - Semana11

Componente3.jsx

```
1 import React from 'react';
2 // Importa la biblioteca React, necesaria para definir componentes de React.
3
4 import mesii from '../assets/img/mesii.jpg';
5 // Importa una imagen de Messi desde una ruta relativa.
6
7 const Componente3 = ({ bgColor, onChangeColor }) => {
8   // Define un componente funcional llamado 'Componente3'. Este componente recibe dos props: 'bgColor' y 'onChangeColor'.
9
10  const changeColor = () => {
11    const randomColor = '#' + Math.floor(Math.random() * 16777215).toString(16);
12    // Genera un color hexadecimal aleatorio.
13    onChangeColor(randomColor);
14    // Llama a la función 'onChangeColor' pasada como prop con el nuevo color aleatorio como argumento.
15  };
16
17  return (
18    <div
19      className="component"
20      style={{
21        backgroundColor: bgColor,
22        // Establece el color de fondo del div según el prop 'bgColor'.
23        width: "200px",
24        height: "200px",
25        display: "flex",
26        alignItems: "center",
27        justifyContent: "center",
28        flexDirection: "column"
29      }}
30    >
31      <img src={mesii} alt="Messi" style={{ width: "100px", height: "100px" }} />
32      <div>
33        <div> Muestra la imagen de Messi importada anteriormente, con un tamaño fijo de 100x100px. </div>
34        <button onClick={changeColor}>Cambiar color</button>
35      </div>
36    </div>
37  );
38
39
40 export default Componente3;
```

Ln 14, Col 78 Spaces: 2 UTF-8 CRLF JavaScript JSX Go Live

Alerta méteo 0:19 25/07/2024

File Edit Selection View Go Run ...

semana 10 y 11

EXPLORER

- SEMANA 10 Y 11
  - lab11
  - Semana10
    - node\_modules
    - public
    - src
      - assets
      - Componentes
        - App.jsx
        - Componente1.jsx
        - Componente3.jsx
        - Footer.jsx
        - Header.jsx
        - Main.jsx
        - widget.jsx
      - hooks
      - useEffect.jsx
      - App.css
      - App.jsx
      - index.css
      - main.jsx
      - eslint.config.js
      - .gitignore
      - index.html
      - package-lock.json
      - package.json
      - README.md
      - vite.config.js
    - Semana11

App.jsx

```
1 import React, { useState } from 'react';
2 // Importa React y 'useState' desde la biblioteca React. 'useState' se utiliza para crear y manejar estados en componentes funcionales.
3
4 import Componente1 from './Componentes/Componente1';
5 import Componente3 from './Componentes/Componente3';
6 import ImageWidget from './Componentes/widget';
7 import Header from './Componentes/Header';
8 import Aside from './Componentes/Aside';
9 import Main from './Componentes/Main';
10 import Footer from './Componentes/Footer';
11 // Importa varios componentes desde sus respectivas rutas. Estos componentes se utilizarán dentro del componente 'App'.
12
13 import './App.css';
14 // Importa el archivo de hoja de estilos CSS para aplicar estilos a los componentes.
15
16 const App = () => {
17   // Declara un componente funcional llamado 'App'.
18
19   const [color, setColor] = useState("#3498db");
20   // Declara una variable de estado 'color' con el valor inicial '#3498db'. 'setColor' es la función para actualizar esta variable de estado.
21
22   const changeColors = () => {
23     // Declara una función 'changeColors' que genera y establece un nuevo color aleatorio.
24     const randomColor = '#' + Math.floor(Math.random() * 16777215).toString(16);
25     // Genera un color aleatorio convirtiendo un número aleatorio en una cadena hexadecimal.
26     setColor(randomColor);
27     // Actualiza la variable de estado 'color' con el nuevo color aleatorio.
28   };
29
30   return (
31     // El componente 'App' retorna el siguiente JSX.
32     <div className="App">
33       <div> Fragmento React para agrupar varios elementos sin agregar nodos adicionales al DOM. </div>
34       <div className="App">
35         <div> Div principal con la clase "App". </div>
36         <Componente1 bgColor={color} onChangeColor={setColor} />
37         <div> Renderiza el componente 'Componente1', pasando 'color' como 'bgColor' y 'setColor' como 'onChangeColor'. </div>
38         <Componente3 bgColor={color} onChangeColor={setColor} />
39         <div> Renderiza otro 'Componente1' con las mismas props. </div>
40         <Componente3 bgColor={color} onChangeColor={setColor} />
41       </div>
42     </div>
43   );
44 }
```

Ln 87, Col 1 Spaces: 2 UTF-8 LF JavaScript JSX Go Live

9°C Prac. despejado 0:19 25/07/2024

File Edit Selection View Go Run ...

semana 10 y 11

EXPLORER

- SEMANA 10 Y 11
  - lab11
  - Semana10
    - node\_modules
    - public
    - src
      - assets
      - Componentes
        - Aside.jsx
        - Componente1.jsx
        - Componente2.jsx
        - Footer.jsx
        - Header.jsx
        - Main.jsx
        - widget.jsx
      - hooks
      - useEffect.jsx
      - App.css
      - App.jsx
      - index.css
      - main.jsx
      - esintruc.js
      - .gitignore
      - index.html
    - package-lock.json
    - package.json
    - README.md
    - vite.config.js
    - Semana11

OUTLINE

TIMELINE

INFO

SOURCES

```
16 const App = () => {
17   // El componente 'App' retorna el siguiente JSX.
18   <div className="App">
19     /* Fragmento React para agrupar varios elementos sin agregar nodos adicionales al DOM. */
20     /* Div principal con la clase "App". */
21     <Component1 bgColor={color} onChangeColor={setColor} />
22     /* Renderiza el componente 'Component1', pasando 'color' como 'bgColor' y 'setColor' como 'onChangeColor'. */
23     <Component2 bgColor={color} onChangeColor={setColor} />
24     /* Renderiza otro 'Componente1' con las mismas props. */
25     <Component3 bgColor={color} onChangeColor={setColor} />
26     /* Renderiza el componente 'Component3' con las mismas props. */
27     <ImageWidget />
28     /* Renderiza el componente 'ImageWidget'. */
29     <div>
30       <div className="container">
31         /* Div principal con la clase "container". */
32         <Header />
33         /* Renderiza el componente 'Header'. */
34         <div className="container-center">
35           /* Div con la clase "container-center". */
36           <Aside />
37           /* Renderiza el componente 'Aside'. */
38           <Main />
39           /* Renderiza el componente 'Main'. */
40         </div>
41         <Footer />
42         /* Renderiza el componente 'Footer'. */
43       </div>
44       <button onClick={changeColors}>Cambiar colores</button>
45       /* Botón que, al hacer clic, llama a la función 'changeColors'. */
46     </div>
47   </div>
48 };
49
50 export default App;
51 // Exporta el componente 'App' como la exportación predeterminada de este módulo, permitiendo que sea importado y utilizado en otros archivos.
```

Ln 87, Col 1 Spaces: 2 UTF-8 LF JavaScript JSX Go Live

