

PROYECTO EJERCICIOS CON REACT

Proyecto de estudio de ejercicios para react utilizando librerías básicas de React

Se mostrara un ejercicio y su solución utilizando lo menos posible internet ni uso de aplicaciones de chatbot de inteligencia artificial . Para practicar React, Javascript, Html y CSS.

- Crear proyecto

```
Windows PowerShell
PS C:\A_CURSOS\2024\Proyecto\Frontend> npx create-react-app react-banca
```

- Subir servidor

```
Windows PowerShell
PS C:\A_CURSOS\2024\Proyecto\Frontend\react-banca> npm start

> react-banca@0.1.0 start
> react-scripts start

(node:18864) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(node:18864) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...

One of your dependencies, babel-preset-react-app, is importing the
"@babel/plugin-proposal-private-property-in-object" package without
declaring it in its dependencies. This is currently working because
"@babel/plugin-proposal-private-property-in-object" is already in your
node_modules folder for unrelated reasons, but it may break at any time.

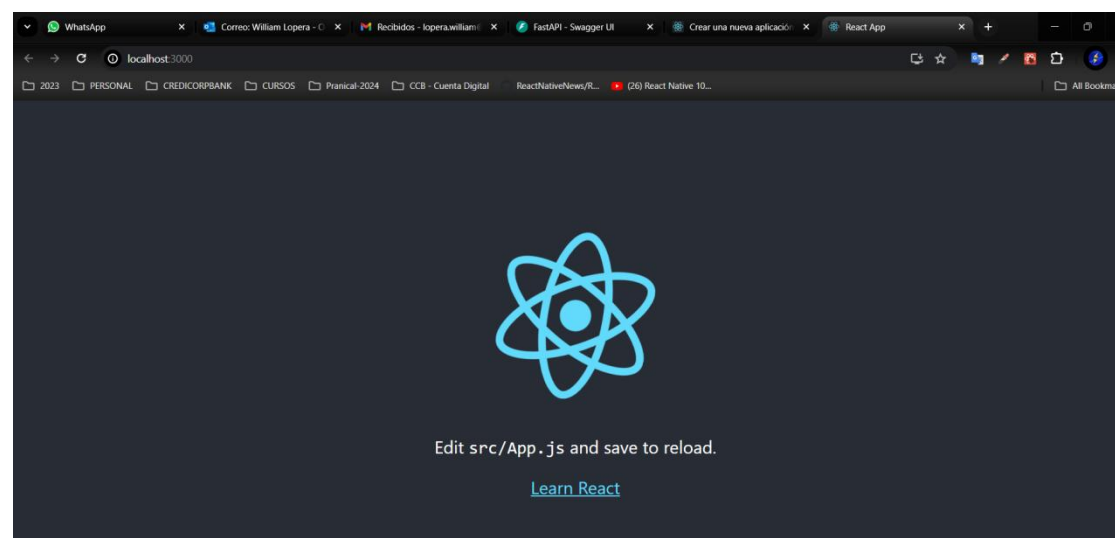
babel-preset-react-app is part of the create-react-app project, which
is not maintained anymore. It is thus unlikely that this bug will
ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to
your devDependencies to work around this error. This will make this message
go away.
Compiled successfully!

You can now view react-banca in the browser.

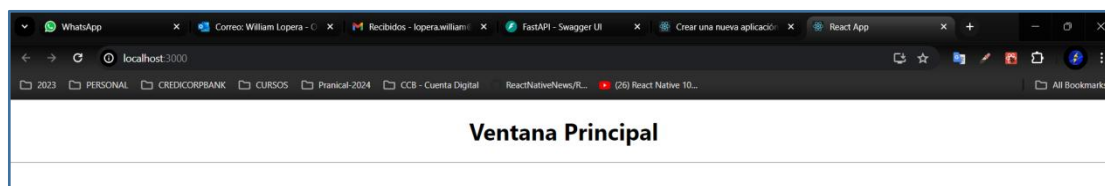
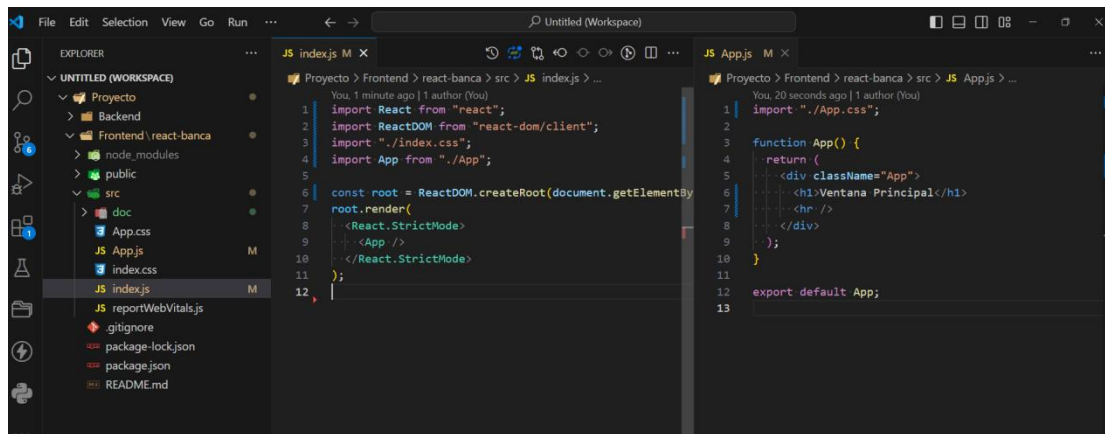
  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.10:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



- *VSCODE: Depurar el código*



Código

Se crearon cuatro componentes importantes:

- *Header: Cabecera de la Pantalla*
- *Footer: Mensajes de información al usuario y del sistema*
- *Menu: Area para agregar valores verticales de Menú (dinámico)*
- *Body: Cuerpo del programa. Este área varia a medida que se seleccionen componentes del menú*

Además se crearon componentes para el manejo de:

- *mensaje: Control de mensajes*
- *HeaderProcess: Control de titulo, descripción y problema a resolver*

Uso de mobx para Tener un almacén común de datos (por ahora mensajes al usuario)

- *Librerías:*

```
"@testing-library/jest-dom": "^5.17.0",
"@testing-library/react": "^13.4.0",
"@testing-library/user-event": "^13.5.0",
"axios": "^1.7.2",
"mobx": "^6.12.4",
"mobx-react": "^9.1.1",
"react": "^18.3.1",
"react-dom": "^18.3.1",
"react-scripts": "5.0.1",
"styled-components": "^6.1.11",
"web-vitals": "^2.1.4"
```

Componentes

Un componente de React es un código pequeño y reutilizable, que es responsable de una parte de la interfaz de usuario de la aplicación. Una aplicación de React es una agregación de componentes. React puede ayudarnos a crear componentes reutilizables. El siguiente diagrama muestra diferentes componentes. Todos los componentes tienen diferentes colores de borde. En React, ensamblamos diferentes componentes para crear una aplicación. Usamos funciones o clases de JavaScript para crear componentes. Si usamos una función, el componente será un componente funcional, pero si usamos una clase, el componente será un componente basado en clases.

Los componentes pueden ser:

Componente funcional / Componente de presentación / Componente sin estado / Componente tonto

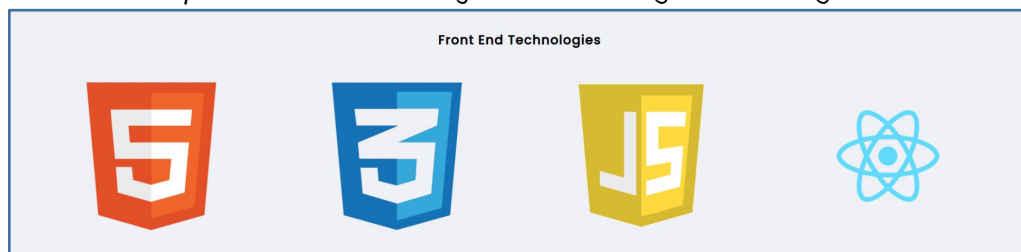
Componente de clase / Componente contenedor / Componente con estado / Componente inteligente

La clasificación de componentes anterior no funciona para la última versión de React, pero es bueno conocer la definición anterior y cómo funcionan las versiones anteriores.

Entonces, cambiemos todo el JSX por componentes. Los componentes en React son funciones o clases de JavaScript que devuelven un JSX. El nombre del componente debe comenzar con mayúscula y, si el nombre consta de dos palabras, debe ser CamelCase (un camello con dos jorobas).

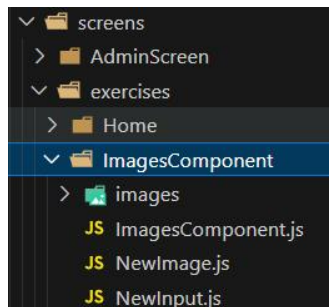
Ejercicio 1

- Cree componentes funcionales y muestre las siguientes imágenes



- *Utilice el componente funcional para crear el siguiente diseño*

- *Suscribir*
- *Regístrese con su dirección de correo electrónico para recibir noticias y actualizaciones.*
- *Tres input (entradas de texto)*
- *Boton de Enviar Subscripción*



```
import React from "react";
import { HeaderProcess } from "../../components/headerProcess/HeaderProcess";
import styled from "styled-components";
import { NewImage } from "./NewImage";
import htmlLogo from "./images/html_logo.png";
import cssLogo from "./images/css_logo.png";
import jsLogo from "./images/js_logo.png";
import reactLogo from "./images/react_logo.png";
import { NewInput } from "./NewInput";

const title = "Componentes";
const exercise =
  "Cree un componente funcional que muestre algunas imágenes y un Formulario con tres entradas de textos y un boton de envío";

export const ImagesComponent = () => {
  const printData = (e) => {
    e.preventDefault();
    const firstname = e.target.firstname.value;
    const lastname = e.target.lastname.value;
    const email = e.target.email.value;
    alert(
      JSON.stringify({
        nombre: firstname,
        apellido: lastname,
        correo: email,
      })
    );
  };
}
```

```

});

return (
  <Container>
    <HeaderProcess title={title} exercise={exercise} />
    <DivContainer>
      <h2 style={{ paddingBottom: "30px" }}>Tecnologías Actuales</h2>
      <DivImage>
        <NewImage url={htmlLogo} />
        <NewImage url={cssLogo} />
        <NewImage url={jsLogo} />
        <NewImage url={reactLogo} />
      </DivImage>
    </DivContainer>
    <Form onSubmit={printData}>
      <P1>SUSCRIBETE</P1>
      <P2>
        Regístrese con su dirección de correo electrónico para recibir
        noticias y actualizaciones.
      </P2>
      <DivInputs>
        <NewInput name="firstname" placeholder="Nombre" />
        <NewInput name="lastname" placeholder="Apellido" />
        <NewInput name="email" placeholder="correo" />
      </DivInputs>
      <Button>Suscribirse</Button>
    </Form>
  </Container>
);
});

const Container = styled.div`
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
`;

const DivContainer = styled.div`
background-color: #a3e4d7;
display: flex;
flex-direction: column;
align-items: center;
width: 98%;
border: 1px solid grey;
border-radius: 10px;
`;

const DivImage = styled.div`
display: flex;
padding-bottom: 20px;
`;

const Form = styled.form`
margin-top: 30px;
background-color: #a3e4d7;
border-radius: 10px;
width: 98%;
border: 1px solid grey;
`;

const P1 = styled.p`

```

```

font-size: 25px;
font-weight: 700;
`;

const P2 = styled.p`
font-size: 15px;
font-weight: 500;
margin-bottom: 50px;
`;

const DivInputs = styled.div`
flex-direction: rows;
`;

const Button = styled.button`
margin-top: 20px;
background-color: #f37474;
width: 18vw;
height: 4vh;
border-radius: 5px;
border: none;
margin-bottom: 30px;
text-align: center;
color: white;
`;

```

```

import React from "react";
import styled from "styled-components";

export const NewImage = ({ url }) => {
  return (
    <div>
      <Img src={url} />
    </div>
  );
};

const Img = styled.img`
width: 15vw;
height: 25vh;
margin: 20px;
`;

```

```

import React from "react";
import styled from "styled-components";

export const NewInput = ({ name, placeholder }) => {
  return <Input name={name} placeholder={placeholder} />;
};

const Input = styled.input`
background-color: white;
color: black;
width: 10vw;
height: 4vh;
border: none;
border-radius: 5px;
margin: 5px;
`;

```

EJERCICIOS PRACTICOS REACT -- WLOPERA @2024

Ejercicios Prácticos

Componente de imágenes

Componentes

Cree un componente funcional que muestre algunas imágenes


Tecnologías Actuales



EJERCICIOS PRACTICOS REACT -- WLOPERA @2024

Cree un componente funcional que muestre algunas imágenes y un Formulario con tres entradas de textos y un boton de envío

Tecnologías Actuales



SUSCRIBETE

Regístrese con su dirección de correo electrónico para recibir noticias y actualizaciones.

Nombre

Apellido

correo

Suscribirse

Matrices

Una matriz es la estructura de datos que se utiliza con más frecuencia para manejar muchos tipos de problemas. En React, usamos `map` para modificar una matriz a una lista de JSX agregando ciertos elementos HTML a cada elemento de una matriz.

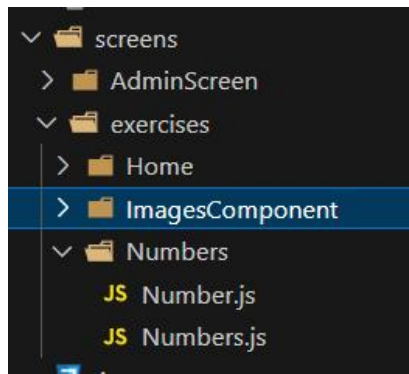
La mayoría de las veces, los datos se presentan en forma de matriz o matriz de objetos. Para representar esta matriz o matriz de objetos, la mayoría de las veces modificamos los datos mediante `map`. En la sección anterior, hemos representado la lista de tecnologías mediante un método `map`.

Las claves ayudan a React a identificar los elementos que han cambiado, añadido o eliminado. Se deben proporcionar claves a los elementos dentro de la matriz para darles una identidad estable. La clave debe ser única. La mayoría de los datos vendrán como un identificador y podemos usar el identificador como clave. Si no pasamos la clave a React durante el mapeo, se genera una advertencia en el navegador. Si los datos no tienen un identificador, tenemos que encontrar una forma de crear un identificador único para cada elemento cuando lo mapeamos.

Ejercicio 2

- En el siguiente diseño, los números pares son verdes, los impares son amarillos y los primos son rojos. Cree los siguientes colores usando el componente `React`
- Crear un área para consultar un número y obtener tipo (Par/Impar/Primo)

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31



Data.js

```
...
const getValues = (max) => {
  const numbers = [];
  for (let i = 0; i < max; i++) {
    numbers.push(i);
  }
  return numbers;
};

export const NUMBERS = getValues(32);
```

```
import React from "react";

export const Number = ({ number, bkColor, width = "6vw" }) => {
  const styles = {
    backgroundColor: bkColor,
    color: "white",
    width: width,
    height: "12vh",
    fontSize: "40px",
    border: "2px solid white",
    alignContent: "center",
  };
  return <div style={styles}>{number}</div>;
};
```

```
import React, { useEffect, useState } from "react";
import { HeaderProcess } from "../../components/headerProcess/HeaderProcess";
import styled from "styled-components";
import { Number } from "../Number";
import { NUMBERS } from "../../data/Data";

const title = "Matrices";
const exercise =
  "Cree una tabla con los números pares son verdes, los impares son amarillos y los primos son rojos.";
```

```
const pair = (number) => number === 0 || (number > 3 && number % 2 === 0);
```

```
const odd = (number) =>
  number === 1 ||
  (number > 7 && (number % 3 === 0 || number % 5 === 0 || number % 7 === 0));
```

```
const getTypeNumber = (number) => {
```

```
const isPair = pair(number);
const isOdd = odd(number);
```

```
if (!isPair && !isOdd) {
  return {
    value: number,
    color: "red",
    type: "primo",
    bkColor: "#FD5E53",
  };
}
```

```
if (isPair) {
  return {
    value: number,
    color: "green",
    type: "par",
    bkColor: "#21BF73",
  };
}
```

```
return {
  value: number,
  color: "yellow",
  type: "impar",
  bkColor: "#FDD83A",
};
};
```

```
const separateIntoGroups = (inputArr) => {
  const groups = [];
  let currentGroup = [];
```

```
  inputArr.forEach((item) => {
    currentGroup.push(item);
```

```
    if (currentGroup.length === 8) {
      groups.push([...currentGroup]);
      currentGroup = [];
    }
  });
});
```

```
  return groups;
};
```

```
export const Numbers = () => {
  const [data, setData] = useState(null);
  const [record, setRecord] = useState(null);
```

```
  useEffect(() => {
    const records = NUMBERS.map((value) => getTypeNumber(value));
    setData(separateIntoGroups(records));
    setRecord(getTypeNumber(0));
  }, []);
```

```
  const handleSetValue = (e) => {
    let data = e.target.value;
```

```
    if (data === "") {
      data = 0;
    }
  }
```

```
const number = parseInt(data);
if (number < 0) {
  return 0;
}
```

```
setRecord(getTypeNumber(number));
e.target.value = number;
};
```

```
return (
  <Container>
    <HeaderProcess title={title} exercise={exercise} />
    {data &&
      data.map((record, index) => (
        <DivRow key={index}>
          {record.map((item) => (
            <Number
              key={item.value}
              number={item.value}
              bkColor={item.bkColor}
            />
          ))}
        </DivRow>
      ))}
    {record && (
      <DivForm>
        <Input value={record.value} onChange={handleSetValue} type="number" />
        <DivRow style={{ marginTop: "20px" }}>
          <Number
            number={record.value}
            bkColor={record.bkColor}
            width={` ${record.value.length / 10}vw` }
          />
          <DivColumn>
            <ul style={{ alignItems: "center" }}>
              <li>value: {record.value}</li>
              <li>color: {record.color}</li>
              <li>tipo: {record.type}</li>
            </ul>
          </DivColumn>
        </DivRow>
      </DivForm>
    )}
  </Container>
);
};
```

```
const Container = styled.div`
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
`;
```

```
const DivRow = styled.div`
  display: flex;
  flex-direction: row;
`;
```

```
const DivColumn = styled.div`
```

```
display: flex;
flex-direction: column;
text-align: left;
`;
```

```
const DivForm = styled.div`
  margin-top: 20px;
`;
```

```
const Input = styled.input`;
```

Ejercicios Prácticos

Componente de Imágenes

Matrices

EJERCICIOS PRACTICOS REACT -- WLOPERA @2024

Matrices

Cree una tabla con los números pares son verdes, los impares son amarillos y los primos son rojos.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

196

196

- value: 196
- color: green
- tipo: par