

UNIVERSIDAD DON BOSCO
ESCUELA DE COMPUTACION
DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA



TITULO:

Guia #02

DOCENTE:

Ing. Emerson Francisco Cartagena Candelario

PRESENTADO POR:

Alvarenga Romero, Josías Isaac #AR200778,

Antiguo Cuscatlán, 16 de agosto de 2023

DESARROLLO

1. Creación de proyecto React, cambiando estilos y modificando App.js (Literales 1,2,3,4,5).

```
.App {
  text-align: center;
}
.App-logo {
  height: 40vmin;
  pointer-events: none;
}
@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}
.App-content {
  background-color: #61dafb;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}
```

```
import './App.css';
import { Form } from './components/Form';

const App = () => {
  return (
    <div className="App">
      <div className="App-content">
        <p>
          Aqui TODO - LIST
        </p>
        <Form/>
      </div>
    </div>
  );
}

export default App;
```

Aqui TODO - LIST

2. Creando primeros componentes y utilizándolos (Literales 6, 7, 8, 9, 10).

Aquí TODO - LIST

Form component

Todo component

```
import React from "react";
import { Todo } from "../Todo";

export const Form = () => {
  return (
    <>
      <h1>Form component</h1>
      <Todo>{}/Todo</>
    </>
  );
}
```

```
import React from "react";

export const Todo = () => {
  return (
    <h1>Todo component</h1>
  );
}
```

3.Utilizacion de Hooks, recibiendo variables en componentes y recorriendo listas con Map (Literales 11, 12, 13).

```
import React from "react";

export const Todo = ({todo}) => {
  return (
    <h1>{todo}</h1>
  );
}
```

Aqui TODO - LIST

todo 1

todo 2

todo 3

```
import React, {useState} from "react";
import { Todo } from "../Todo";

export const Form = () => {
  const [todos, setTodos] = useState([
    {todo: 'todo 1'},
    {todo: 'todo 2'},
    {todo: 'todo 3'},
  ]);

  return (
    <>
      {
        todos.map((value, index) => (
          <Todo todo={value.todo}/>
        ))
      }
    </>
  );
}
```

4. Mejorando el componente Form, manejando Eventos, creando validaciones simples.

```
import React from "react";

export const Todo = ({todo, index, deleteTodo}) => {
  return (
    <>
      <div className="list">
        <h3>{todo}</h3>
        <button className="btn-delete" onClick={() => deleteTodo(index)}>x</button>
      </div>
    </>
  );
}
```

Form.jsx

```
const handleChange = e => setTodo({[e.target.name] : e.target.value});
const handleClick = e => {
  if(Object.keys(todo).length === 0 || todo.todo.trim() === ''){
    alert('el campo no puede estar vacio !!!');
    return;
  }
  setTodos([...todos,todo]);
};

const deleteTodo = (indice) => {
  const newTodos = [...todos];
  newTodos.splice(indice,1);
  setTodos(newTodos);
}

return (
  <>
    <form onSubmit={e => e.preventDefault()}>
      <label>Agregar Tarea</label><br />
      <input type="text" name="todo" onChange={handleChange} />
      <button onClick={handleClick}>Agregar</button>
    </form>
    {
      todos.map((value,index) => (
        <Todo todo={value.todo} key={index} index={index} deleteTodo={deleteTodo}/>
      ))
    }
  </>
)
```

Aqui TODO - LIST

Agregar Tarea

Agregar

todo 1 ☐

todo 2 ☐

todo 3 ☐

10 ☐

12 ☐

13 ☐

Aqui TODO - LIST

Agregar Tarea

Agregar

todo 1 ☐

todo 2 ☐

todo 3 ☐

10 ☐

12 ☐

localhost:3000 dice

el campo no puede estar vacio !!!

Aceptar

5. Actualizando README.md

```
# Pruebas-React: Proyecto de Prueba con React y Webpack
¡Bienvenido a Pruebas-React! Este es un proyecto de muestra creado para demostrar
cómo configurar y ejecutar una aplicación básica de React utilizando Webpack. Sigue
los pasos a continuación para poner en marcha el proyecto en tu máquina local.

## Requisitos Previos
Antes de comenzar, asegúrate de tener instalado Node.js en tu sistema. Puedes
descargarlo e instalarlo desde [nodejs.org](https://nodejs.org/).

## Clonar el Repositorio
1. Abre tu terminal y navega hasta la ubicación donde deseas clonar el repositorio.
2. Ejecuta el siguiente comando para clonar el repositorio:
```
git clone https://github.com/isaac-rom/GuiaLabDPS/
```
3. Ingresa al directorio del proyecto usando el comando:
```
cd guia02\Procedimiento\pruebas-react\
```

## Instalar dependencias
1. En el directorio del proyecto, ejecuta el siguiente comando para instalar las
dependencias necesarias:
```
npm install
```

## Configuración de Desarrollo
1. Ahora que las dependencias están instaladas, puedes iniciar el servidor de
desarrollo. Ejecuta el siguiente comando:
```
npm start
```
2. Abre tu navegador web y visita [http://localhost:3000](http://localhost:3000).
Deberías ver la aplicación de React en funcionamiento.
```