TRABAJO DE RETOS DE LA SEMANA 6

CODIFICACIÓN:

```jsx
import React, { useState } from "react";
import { ToastContainer, toast } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
import axios from "axios";
import Card from "./ui/Card";

const Api = () => {
  const [characters, setCharacters] = useState([]);
  const [isLoading, setIsLoading] = useState(false);

  const getCharacters = async () => {
    setIsLoading(true);
    setTimeout(async () => {
      const {
        data: { results },
      } = await axios.get("https://rickandmortyapi.com/api/character");
      setCharacters([...characters, ...results]);
      setIsLoading(false);
    }, 5000);
  };

  const handleButtonClick = () => {
    toast.info("Cargando imagenes", { autoClose: 4000 });
    getCharacters();
  };
```

```jsx
  return (
    <section>
      <ToastContainer />
      <h3 className="text-gray-300 text-2xl font-title">API RICK AND MORTY</h3>
      <p className="text-gray-400">Consumo de la API RICK AND MORTY</p>
      <button className="Button font-title p-2 mt-3" onClick={handleButtonClick}>
        Consultar
      </button>
      <div className="flex flex-row mt-3 flex-wrap">
        {characters.map((character) => (
          <Card character={character} key={character.id} />
        ))}
      </div>
    </section>
  );
};

export default Api;
```

Componente formulario:

```jsx
import React, { useState } from "react";

const Formulario = () => {
  const [formValues, setFormValues] = useState({});
  const handleInputChange=(e)=>{
    setFormValues({
      ...formValues,
      [e.target.name]:e.target.value
    })
  }
  return (
    <section>
      <h3 className="text-gray-300 text-2xl font-title">Formularios</h3>
      <p className="text-gray-400">UseState,Formularios,Render Condicional</p>
      <div className="flex flex-row mt-3">
        <form className="basis-1/3 border-solid border-2 border-gray-400 p-3 rounded-sm">
          <h5 className="font-title text-gray-200">Registrarme</h5>
          <input type="text" className="w-full mt-1 font-title p-1" name="name" placeholder="Ingrese su Nombre
          <input type="text" className="w-full mt-3 font-title p-1" name="email" placeholder="Ingrese su Corre
          <button type="submit" className="Button font-title p-2 mt-3 w-full">Enviar</button>
        </form>
        <div className="basis-2/3 pl-2">
          <div className="p-3 bg-slate-50 w-1/2 rounded-sm shadow-md shadow-lime-600">
          <h4 className="font-title text-gray-400">Usuario Registrado</h4>
```

```jsx
          <h5 className="font-title text-gray-200">Registrarme</h5>
          <input type="text" className="w-full mt-1 font-title p-1" name="name" placeholder="Ingrese su Nombre
          <input type="text" className="w-full mt-3 font-title p-1" name="email" placeholder="Ingrese su Corre
          <button type="submit" className="Button font-title p-2 mt-3 w-full">Enviar</button>
        </form>
        <div className="basis-2/3 pl-2">
          <div className="p-3 bg-slate-50 w-1/2 rounded-sm shadow-md shadow-lime-600">
          <h4 className="font-title text-gray-400">Usuario Registrado</h4>
          {
            !formValues.name?(<p className="text-black">No existe un usuario registrado.</p>):
            (
              <>
                <p className="text-black">Nombre: {formValues.name}</p>
                <p className="text-black">Correo: {formValues.email}</p>
              </>
            )
          }
          </div>
        </div>
      </div>
    </section>
  );
};

export default Formulario;
```

```jsx
1   import React, { useState } from "react";
2   import "react-toastify/dist/ReactToastify.css";
3   import axios from "axios";
4
5   |
6   const Pokemon = () => {
7       const [pokemonName, setPokemonName] = useState('');
8       const [pokemonData, setPokemonData] = useState(null);
9       const [error, setError] = useState(null);
10
11      const getPokemon = async () => {
12        try {
13          const response = await axios.get(`https://pokeapi.co/api/v2/pokemon/${pokemonName.toLowerCase()}`);
14          setPokemonData(response.data);
15          setError(null);
16        } catch (error) {
17          console.error('Error fetching Pokemon:', error);
18          setPokemonData(null);
19          setError('Pokémon no encontrado');
20        }
21      };
22
23      const handleSubmit = (e) => {
24        e.preventDefault();
25        getPokemon();
26      };
```

```jsx
  return (

    <div >
      <h3 className="text-gray-300 text-2xl font-title">API POKEMON</h3>

      <form onSubmit={handleSubmit} style={{ marginBottom: '20px' }}>
      <input type="text" className="w-full mt-1 font-title p-1" name="name" placeholder="Ingrese Pokemon" va
      <button type="submit" className="Button font-title p-2 mt-3 w-full">Buscar</button>
      </form>
      {error && (
        <p style={{ color: '#E3350D', marginBottom: '20px' }}>{error}</p>
      )}
      {pokemonData && (
        <div >
          <h2 style={{ fontSize: '24px', color: '#E3350D' }}>
            {pokemonData.name}
          </h2>
          <img
            src={pokemonData.sprites.front_default}
            alt={pokemonData.name}
            style={{ maxWidth: '300px', marginBottom: '20px' }}
          />
        </div>
      )}
    </div>
```

RESULTADO:

Alan Rails
Especie: Human
Género: Male
Estado:
Dead

Albert Einstein
Especie: Human
Género: Male
Estado:
Dead

Alexander
Especie: Human
Género: Male
Estado:
Dead

Alien Googah
Especie: Alien
Género: unknown
Estado:
unknown

Alien Morty
Especie: Alien
Género: Male
Estado:
unknown

Alien Rick
Especie: Alien
Género: Male
Estado:
unknown

Amish Cyborg
Especie: Alien
Género: Male
Estado:
Dead

Annie
Especie: Human
Género: Female
Estado:
Alive

Antenna Morty
Especie: Human
Género: Male
Estado:
Alive

API POKEMON

pikachu

Buscar

pikachu