









Hector Josue Orozco Salazar

201314296

Software Avanzado

Tarea Practica


Get Paginado Poke-api

API REST Y SOAP			
GET API REST - POKE API			
#	Name	Image	
1	blastoise		
2	caterpie		
3	metapod		
4	butterfree		
5	bulbasaur		
6	ivysaur		
7	venusaur		
8	charmander		
9	charmeleon		
10	charizard		
11	weedle		
12	squirtle		

Post card Pokémon

API REST Y SOAP

GET API REST - POKE API



celebi

251

Buscar

## Consumo de API's

```
export const Home = () => {  
  
  const [getPoke, setPoke] = useState([]);  
  const [getCard, setCard] = useState({name: "", img:""});  
  const [buscar, setBuscar] = useState("");  
  
  const pokeApi = async () => {  
    const url = `https://pokeapi.co/api/v2/pokemon?limit=50&offset=0`;   
    const res = await fetch(url,{  
      method: "GET",  
      headers: {  
        Accept: "application/json",  
        "Content-Type": "application/json",  
      },  
    })  
    const resJson = await res.json();  
    // setPoke(await resJson.results);  
  
    resJson.results.forEach(async (element) => {  
      // console.log(element);  
      await fetch(element.url,{  
        method: "GET",  
        headers: {  
          Accept: "application/json",  
          "Content-Type": "application/json",  
        },  
      }).then((data)=>{  
        return data.json();  
      }).then ( (data) => {  
        setPoke(prev => [...prev, data]);  
      });  
    });  
  };  
};
```

## Paginado API'S

```
<>  
<h1>API REST Y SOAP</h1>  
  
<Accordion>  
  <Accordion.Item eventKey="0">  
    <Accordion.Header>GET API REST - POKE API</Accordion.Header>  
    <Accordion.Body>  
      <Table striped bordered hover size="sm">  
        <thead>  
          <tr>  
            <th>#</th>  
            <th>Name</th>  
            <th>Image</th>  
          </tr>  
        </thead>  
        <tbody>  
          {  
            getPoke.map((pokemon, i) => {  
              return(  
                <tr >  
                  <th>{i+1}</th>  
                  <th>{pokemon.name}</th>  
                  { /* <th>{console.log(pokemon)}</th> */ }  
                  <th><img src={pokemon.sprites.front_default}</img></th> img elements must ha  
                </tr>  
              )  
            })  
          }  
        </tbody>  
      </Table>  
    </Accordion.Body>  
  </Accordion.Item>  
</Accordion>  
  
<Card style={{ width: '18rem' }}>  
  <img variant="top" src={getCard.sprites.front_default}/> img elements must have an alt prop, either with mean  
  <Card.Body>
```