

# TRABAJO DEL API DE POKEMON

## CODIFICACIÓN:

```
TS App.tsx  X  # index.css  # App.css
pokemon > src > TS App.tsx > ...
1  import React, { useEffect, useState } from 'react';
2
3  interface Pokemon {
4    name: string;
5    sprites: {
6      front_default: string;
7    };
8  };
9
10 const App: React.FC = () => {
11   const [numeroAleatorio, setNumeroAleatorio] = useState<number>(0);
12   const [count, setCount] = useState<number>(0);
13   const [pokemon, setPokemon] = useState<Pokemon | null>(null);
14   const [pokemonName, setPokemonName] = useState<string | null>(null);
15
16
17   const generarNumeroAleatorio = () => {
18     const nuevoNumero: number = Math.floor(Math.random() * 898) + 1;
19     setNumeroAleatorio(nuevoNumero);
20   };
21
22   const obtenerPokemon = async () => {
23     try {
24       const response = await fetch(`https://pokeapi.co/api/v2/pokemon/${numeroAleatorio}`);
25       const data: Pokemon = await response.json();
26       setPokemon(data);
27       setPokemonName(data.name);
28     } catch (error) {
29
30
31
32
33   const increment = () => {
34     setCount(count + 1);
35     setNumeroAleatorio(numeroAleatorio + 1);
36   };
37
38   const decrement = () => {
39     setCount(count - 1);
40     setNumeroAleatorio(numeroAleatorio - 1);
41   };
42
43   useEffect(() => {
44     if (numeroAleatorio === 0) {
45       generarNumeroAleatorio();
46     } else {
47       obtenerPokemon();
48     }
49   }, [numeroAleatorio]);
50
51   return (
52     <div className="flex items-center justify-center">
53       <div className="flex-box p-10 font-mono space-y-4">
54         <div className="grid place-items-center">
55           <h3 className="text-black-300 text-2xl font-title">API POKEMON</h3>
56         </div>
57         <div className="flex space-x-2 mb-4 text-sm font-medium">
58           <div className="flex space-x-4">
59             <button className="px-6 h-12 uppercase font-semibold tracking-wider border-2 border-orange-500 bg-lime-2
```

