

Guía Profesional de TypeScript para React

Gua Profesional de TypeScript para React

Una lista clara y concisa de lo que necesitas dominar para trabajar con React y TypeScript como un profesional, con ejemplos reales.

1. Tipado Bsico

```
let count: number = 0;
let name: string = "Juan";
let isActive: boolean = true;
```

2. Tipos de Objetos

```
type User = {
  name: string;
  email: string;
  age?: number; // opcional
};
```

3. Tipado de Props

```
type ButtonProps = {
  text: string;
  onClick: () => void;
};

const Button: React.FC<ButtonProps> = ({ text, onClick }) => (
  <button onClick={onClick}>{text}</button>
);
```

4. Tipado con useState

```
const [name, setName] = useState<string>("");
const [user, setUser] = useState<User | null>(null);
```

5. Tipado de Eventos

```
const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  console.log(e.target.value);
};
```

Guía Profesional de TypeScript para React

6. Tipado de useEffect

```
useEffect(() => {  
  // lgica  
}, []);
```

7. Tipado en Fetch o Axios

```
type Post = {  
  id: number;  
  title: string;  
  body: string;  
};  
  
const fetchPosts = async (): Promise<Post[]> => {  
  const res = await axios.get<Post[]>("https://jsonplaceholder.typicode.com/posts");  
  return res.data;  
};
```

8. Tipos con useContext

```
type ThemeContextType = {  
  darkMode: boolean;  
  toggleDarkMode: () => void;  
};  
  
const ThemeContext = React.createContext<ThemeContextType | undefined>(undefined);
```

9. Formularios con react-hook-form

```
type FormValues = {  
  name: string;  
  email: string;  
};  
  
const { register, handleSubmit } = useForm<FormValues>();
```

10. Custom Hooks Tipados

```
const useToggle = (initial: boolean): [boolean, () => void] => {  
  const [state, setState] = useState(initial);  
  const toggle = () => setState((s) => !s);  
  return [state, toggle];  
};
```

Guía Profesional de TypeScript para React

Listo, con esto ya te paras duro como profesional en React + TypeScript.