



TypeScript (parte4 Interfaces)

LABORATORIO DE COMPUTACIÓN III

UTNFRA – TÉCNICO UNIVERSITARIO EN PROGRAMACIÓN --

Problemática

```
function enviarMision(xmen:any){  
    console.log("Enviando a: " + xmen.nombre);  
}  
  
let wolverine = {  
    nombre:"James",  
    poder: "Regeneración"  
};  
  
enviarMision(wolverine);
```



```
function enviarMision(xmen:any){  
    console.log("Enviando a: " + xmen.nombre);  
}  
  
let wolverine = {  
    nombreXmen:"James",  
    poder: "Regeneración"  
};  
  
enviarMision(wolverine);
```



TypeScript no avisa del error

```
function enviarMision(xmen:{ nombre:string }){  
    console.log("Enviando a: " + xmen.nombre);  
}  
  
let wolverine = {  
    nombreXmen:"James",  
    poder: "Regeneración"  
};  
  
enviarMision(wolverine);
```

Posible solución: Ahora TypeScript da cuenta del error
Pero si el objeto tiene muchas propiedades o métodos
No queda elegante pasar todos los integrantes del
objeto

Interfaz básica

```
interface Xmen{
  nombre: string,
  poder: string
}

function enviarMision( xmen: Xmen){
  console.log("Enviando a: " + xmen.nombre);
}

function enviarCuartel( xmen: Xmen){
  console.log("Enviando al cuartel: " + xmen.nombre);
}

let wolverine:Xmen = {
  nombre:"James",
  poder: "Regeneración"
};

enviarCuartel(wolverine);
enviarMision(wolverine);
```



```
interface Xmen{
  nombre: string,
  poder: string
}

function enviarMision( xmen: Xmen){
  console.log("Enviando a: " + xmen.nombre);
}

function enviarCuartel( xmen: Xmen){
  console.log("Enviando al cuartel: " + xmen.nombre);
}

let wolverine:Xmen = {
  nombreXmen:"James",
  poder: "Regeneración"
};

enviarCuartel(wolverine);
enviarMision(wolverine);
```



Propiedades Opcionales

```
interface Xmen{
  nombre: string,
  poder: string
}

function enviarMision( xmen: Xmen){
  console.log("Enviando a: " + xmen.nombre);
}

function enviarCuartel( xmen: Xmen){
  console.log("Enviando al cuartel: " + xmen.nombre);
}

let wolverine = {
  nombre:"James",
};

enviarCuartel(wolverine);
enviarMision(wolverine);
```

Wolverine pierde sus poderes, entonces ya no cumple con la interface Xmen. Pero ni en el cuerpo de enviarMision ni en el de enviarCuartel se utilizan. Puedo colocar la propiedad poder como opcional



```
interface Xmen{
  nombre: string,
  poder?: string
}

function enviarMision( xmen: Xmen){
  console.log("Enviando a: " + xmen.nombre);
}

function enviarCuartel( xmen: Xmen){
  console.log("Enviando al cuartel: " + xmen.nombre);
}

let wolverine = {
  nombre:"James",
};

enviarCuartel(wolverine);
enviarMision(wolverine);
```

Métodos de la Interfaz

```
interface Xmen{
  nombre: string,
  poder: string
  regenerar( apellido:string ):void;
}

function enviarMision( xmen: Xmen){
  console.log("Enviando a: " + xmen.nombre);

  xmen.regenerar("Logan");
}

let wolverine = {
  nombre:"James",
  poder: "Regeneración",
  regenerar(apellido:string){
    console.log("Se ha regenerado " + x);
  }
};

enviarMision(wolverine);
```

Interfaces en las clases

```
interface Xmen{
  nombre: string;
  nombreReal?: string;
  regenerar( nombreReal:string ):void;
}

class Mutante implements Xmen{
  nombre:string;
  poder: string;
  regenerar(nombre:string){
    console.log("Hola" + nombre);
  }
}

let Wolverine = new Mutante();

}
```

```
let Wolverine = new Mutante();
```

Wolverine.

- nombre (property) Mutante.nombre: string ⓘ
- poder
- regenerar
- class
- ctor
- dowhile

```
interface Xmen{
  nombre: string;
  nombreReal?: string;
  regenerar( nombreReal:string ):void;
}

class Mutante implements Xmen{
  nombre:string;
  poder: string;
  regenerar(nombre:string){
    console.log("Hola" + nombre);
  }
}

let Wolverine = new Mutante();

(<Xmen>Wolverine).|
```

nombre	(property) Xmen.nombre: string ⓘ
nombreReal	
regenerar	

```
}
```


Interfaces a las funciones

```
interface DosNumerosFunc{  
  (num1:number, num2:number): number  
}  
  
let sumar:DosNumerosFunc;  
  
sumar = function(a:number, b:number):number{  
  return a + b;  
}  
  
let restar:DosNumerosFunc;  
  
restar = function(a:number, b:number):number{  
  return a - b;  
}
```