

POO con TypeScript

by @nicobytes





Nicolas Molina

@nicobytes

Google Developer Expert
Dev and Teacher at **Platzi**

TypeScript

ESNext

JS

Proyecto

Configuración del proyecto con ts-node



```
/* OK */
```

```
declare function fn(x: HTMLDivElement): string;
```

```
declare function fn(x: HTMLElement): number;
```

```
declare function fn(x: unknown): unknown;
```

```
var myElem: HTMLDivElement;
```

```
var x = fn(myElem); // x: string, :)
```



Tu primera Clase

Iniciémos con POO en TypeScript



```
class Punto {  
  constructor ( x , y ){  
    this.x = x;  
    this.y = y;  
  }  
  
  static distancia ( a , b ) {  
    const dx = a.x - b.x;  
    const dy = a.y - b.y;  
  
    return Math.sqrt ( dx * dx + dy * dy );  
  }  
}
```

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing JavaScript.

TypeScript Code



Compilation/Transpiling



Vanilla JS Code

TypeScript file (*.ts)
(Classes, Interface, Modules, Types)

TypeScript Compiler (tsc)
(Target: ES3/ES5/ES6,

JavaScript file (*.js)
(Runs everywhere)

Métodos

Agrega comportamiento a tus objetos



Acceso público

Aprendiendo el acceso por defecto de TS



Acceso privado


Protegiendo nuestras propiedades y métodos



El constructor

El método principal de nuestras clases



A promotional image for the animated series 'Bob the Builder'. In the foreground, Bob, a young boy with a large nose, wearing a yellow hard hat, a red and orange plaid shirt, and blue overalls, is smiling and waving. He has a tool belt with a wrench and a screwdriver. In the background, there are several anthropomorphic construction vehicles: a yellow digger on the left, a red truck in the center, and a blue truck on the right. A small character with a mustache, wearing a plaid shirt and a tool belt, is standing between the red and blue trucks. The scene is set against a blue sky with a white cloud and a pink building with a corrugated metal roof.

BobTM the Builder

EN ESPAÑOL

Getters

Una forma de acceder a los atributos



Setters

Una forma de modificar atributos



Herencia

Abstraer comportamientos



Acceso protegido

Heredando propiedades



Estáticos

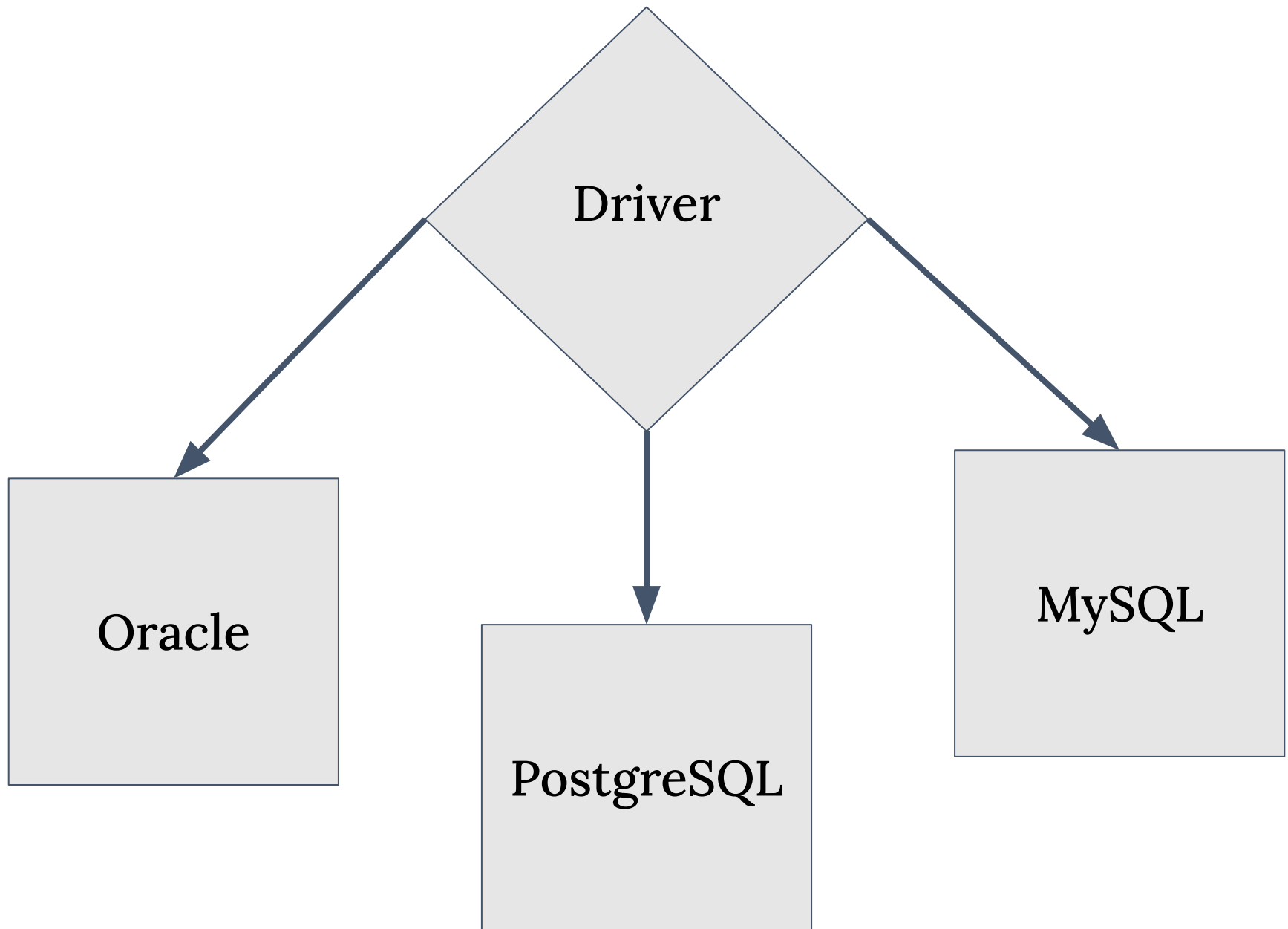
Haciendo nuestra propia Math



Interfaces

Implementando en las clases





Clases abstractas

Restringir la creación de objetos base



Singleton: constructor privado

Aplicando patrones de SOLID



Trabajando con Promesas

Asincronismo en TypeScript



De funciones a clases

Un refactor de nuestro Product Service



Consumiendo ProductMemory

Usando clases para servicios



Creando ProductHttp

Conectando a una Web API



Consumiendo ProductHttp

Usando clases para servicios



Introducción a los Genéricos

Aprendiendo la sintaxis



Genéricos en Clases

Creado un Http Service genérico



Genéricos en Métodos

Creado un Http Service genérico



Decoradores

Agregando funcionalidad extra

