

# Microsoft Azure Devops (MS-TFS)

Por: Carlos Carreño  
[ccarreno@cienciadedatos.es](mailto:ccarreno@cienciadedatos.es)

Octubre, 2020



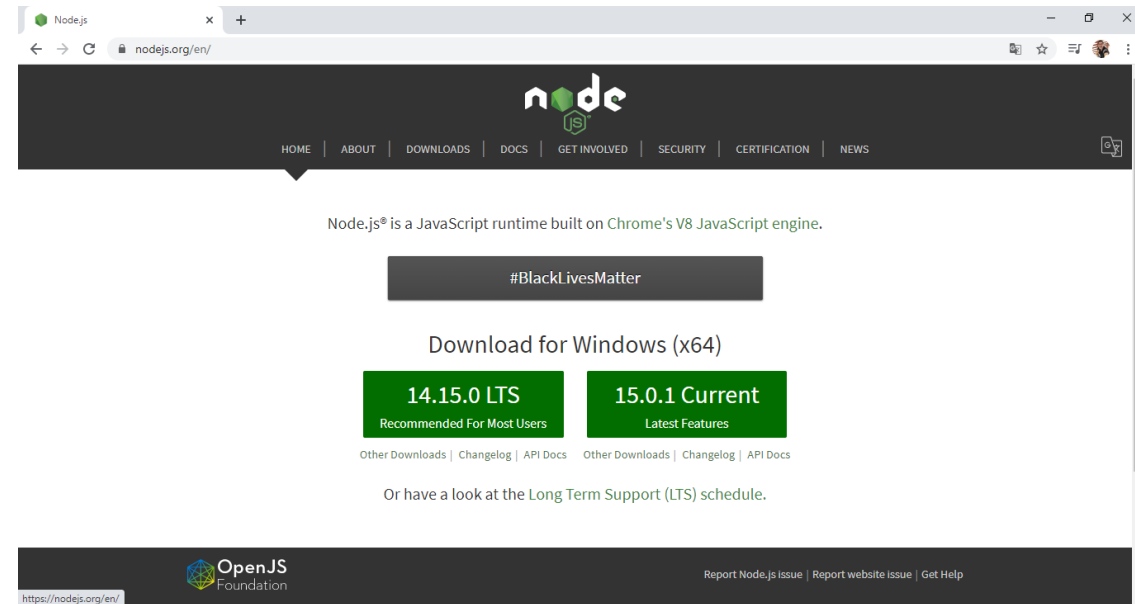
# Unidad 2 Proyecto de Software

- Herramientas de Desarrollo
- Usando Typescript
- Creación del Proyecto
- Desarrollo del Proyecto

# Herramientas de Desarrollo



- Npm
- Typescript
- Visual Studio Code
- WebStorm
- Visual Studio
- Node.js





# Usando Typescript

- Que es Typescript?
  - ☐ TypeScript es un super conjunto tipado de JavaScript que compila a JavaScript plano. Como una analogía, si JavaScript fuera CSS entonces TypeScript sería SCSS.
- Verificando versión de Node
  - ☐ `node --version`
  - ☐ `npm --version`
- Instalación
  - ☐ **`npm install -g typescript`**

# TypeScript: Arquitectura de Componentes



- **language**
  - syntax
    - types annotations
    - es6 syntax
    - syntax closer to object oriented programming languages
- **compiler**
  - converts code to js, combining code, generate source maps
  - written in typescript itself
  - compiles from .ts to .js
- **language service**
  - is a devtool type information, autocomplete, type hitting

# Usando Typescript: Compilando el Programa



- Compilando TypeScript a JavaScript
  - ☐ El código TypeScript esta en un archivo .ts. Los navegadores no podrán ejecutar este código por si mismos. Tendrás que compilar el TypeScript a JavaScript plano que pueda ser entendido por navegadores.
  - ☐ **tsc** first.ts
  - ☐ **tsc** \*.ts
  - ☐ **tsc** first.ts product.ts bot.ts
  - ☐ **tsc** first.ts -w



# Usando Typescript: Tipos de Datos

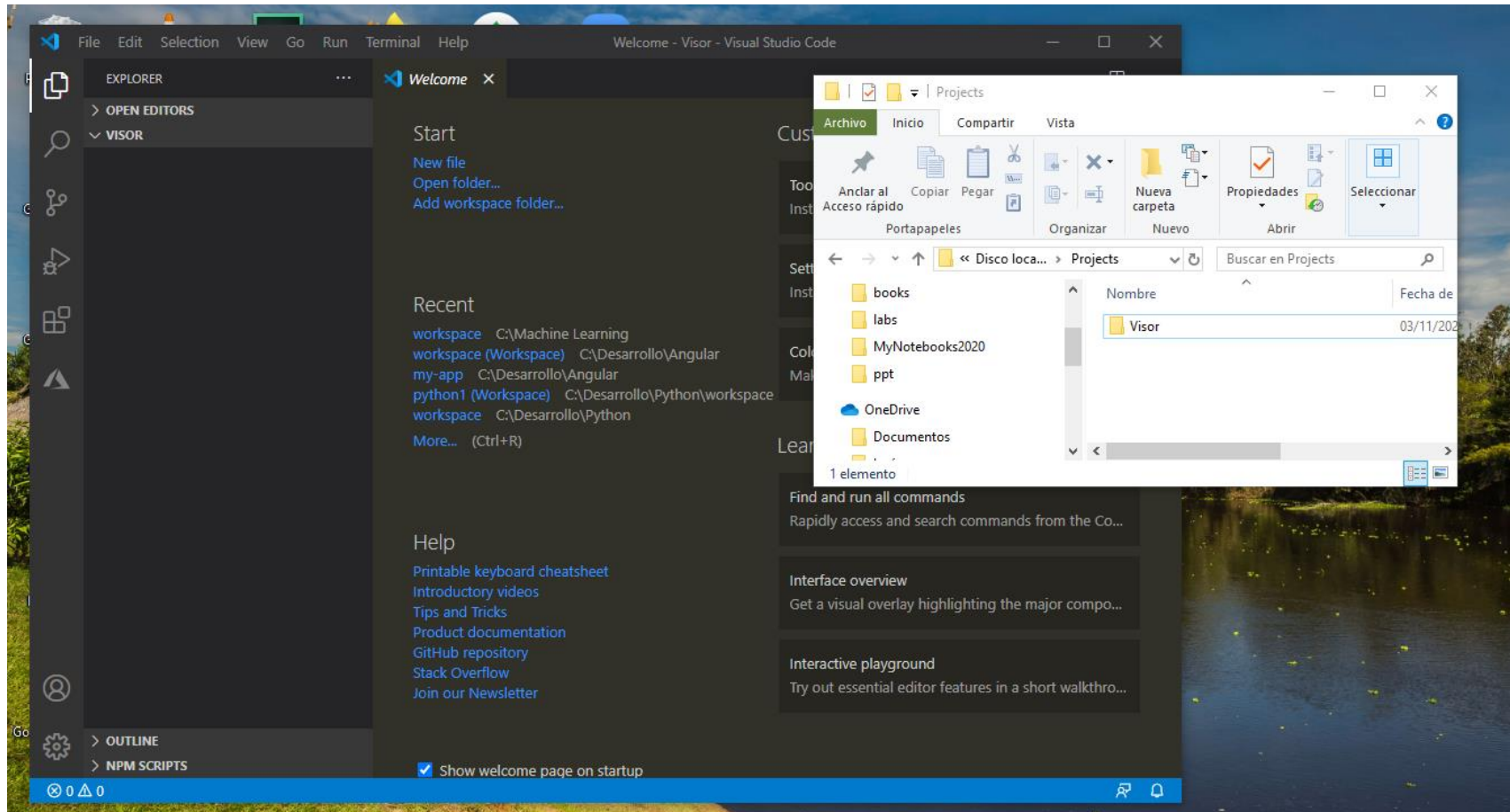
- **Static Type Checking**

- (Optional) check and assign types
- Compilation Errors(reduce bugs, issues)
  - not replace tests, you can use it
- more descriptive code

- **Types**

- String, Number, Boolean, Array, Any, Void, Null, Tuple, Enums, Generics
  - Classes:
    - No protoTypes
    - encapsulation
    - inheritance
    - modifiers

# Creación del Proyecto







# Desarrollo del Proyecto: Typescript

- Tipos de Datos
- Variables
- Estructuras de Control
- Interfaces
- Clases
- Herencia

# Laboratorio



- LAB: Realizar una API con Typescript y Node.js para manejar el CRUD de una tienda en línea de venta de artículos deportivos llamada **“NaikySport”**

```
1  create database naiky_db;
2
3  use naiky_db;
4
5  create table product(
6      id int(11) not null auto_increment primary key,
7      name varchar(180),
8      description varchar(255),
9      image varchar(200),
10     price float not null,
11     stock int not null default 0,
12     created_at timestamp default current_timestamp
13 );
14
15 describe product;
```

# Laboratorio: Arquitectura de la API



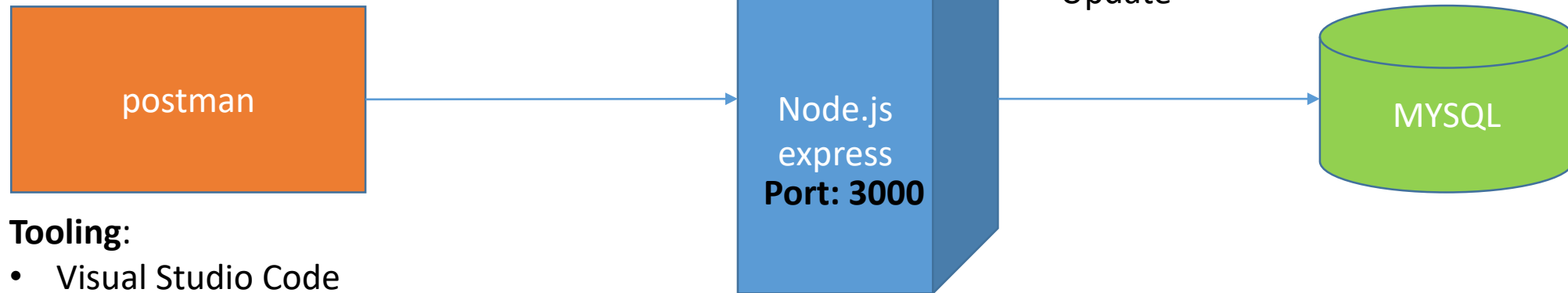
## Procedimiento

- 1) Preparar nuestro entorno
- 2) Preparar acceso a la BD
- 3) Crear las rutas
- 4) Crear los controllers
- 5) Realizar el acceso con postman

/api/product

### CRUD

- Create
- Get
- Delete
- Update



## Tooling:

- Visual Studio Code
- Typescript
- postman

# Laboratorio: Test con Postman



The screenshot displays the Postman application window. The top bar includes the Postman logo, menu items (File, Edit, View, Help), and buttons for 'New', 'Import', 'Runner', and 'My Workspace'. The left sidebar shows a 'Filter' search bar and tabs for 'History', 'Collections', and 'APIs'. The main workspace is titled 'Untitled Request' and shows a POST request to 'localhost:3000/api/product/'. The request body is a JSON object with the following fields: 'name' (TShirt Women), 'description' (TShirt for Women XL), 'image' (a URL to a product image), 'price' (240.78), and 'stock' (100). The response is shown at the bottom, indicating a 'Status: 200 OK' with a 'Time: 479 ms' and 'Size: 309 B'. The response body is a JSON object with a 'message' field: 'Successful - Product Created.'

Postman

File Edit View Help

New Import Runner My Workspace Invite

Filter

History Collections APIs

Save Responses Clear all

Today Yesterday

Launchpad POST localhost:3000/api/product/ No Environment

Untitled Request BUILD

POST localhost:3000/api/product/ Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "name": "TShirt Women",
3   "description": "TShirt for Women XL",
4   "image": "https://cdnmedia.triathlon.com.pe/media/catalog/product/cache/1/image/1200x/040ec09b1e35df139433887a97daa66f/base/832864-011.jpg",
5   "price": 240.78,
6   "stock": 100
7 }
8
```

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 479 ms Size: 309 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Successful - Product Created."
3 }
```

Find and Replace Console Bootcamp Build Browse