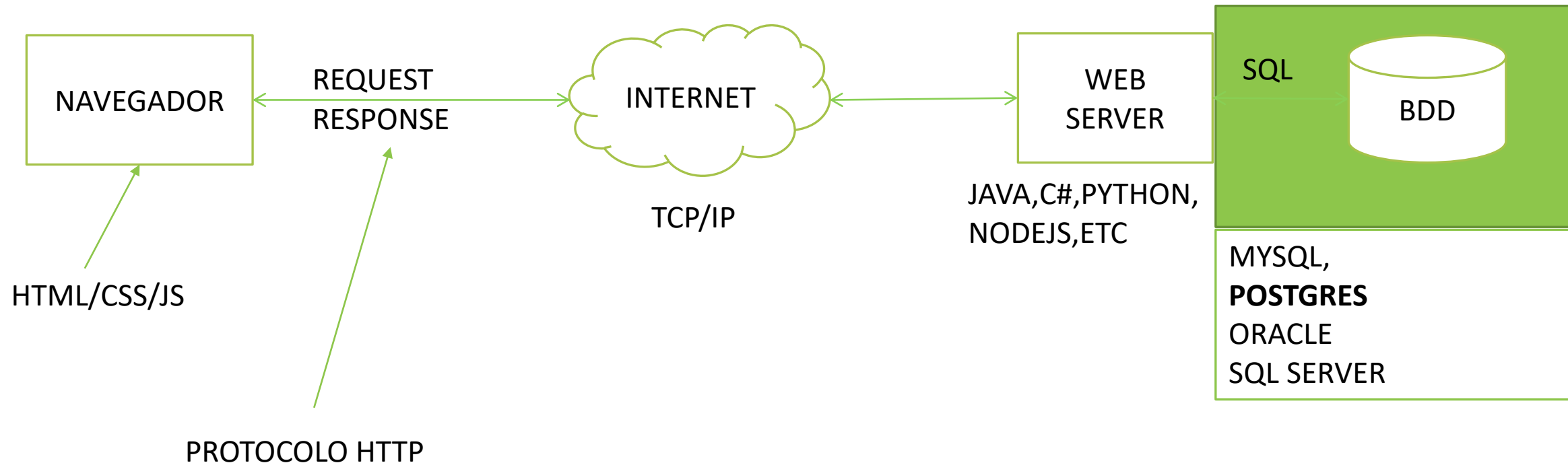


TS

TYPESCRIPT



Diagrama general



Primeros pasos con typescript

Para instalar el typescript a nivel global

```
npm install typescript -g
```

```
D:\curso-codecripto\curso-typescript>node -v  
v16.15.0
```

```
D:\curso-codecripto\curso-typescript>npm -v  
8.5.5
```

```
C:\Users\jviejo.GTT\AppData\Roaming\npm  
└─ typescript@4.7.4
```

1. Creamos el directorio en el escritorio curso-typescript-1
2. Lo abrimos con vscode
3. Hacemos ctrl+n para crear un fichero nuevo
4. Escribimos en el console.log("hello")
5. Hacemos Ctrl+s para grabarlo. Lo llamamos app.ts
6. Abrimos un terminal ctrl+ñ. Nos pasamos al cmd dando cmd en el terminal
7. Hacemos npx tsc --init. Esto crea un fichero llamado tsconfig
8. Hacemos npm init --y. Esto cree al fichero package.json
9. Ejecutamos la order **npx nodemon --exec "npx ts-node" app.ts**

**EJECUTAR LOS PASOS ANTERIORES VARIAS VECES HASTA QUE
TENGAMOS UNA EJECUCIÓN LIMPIA**

Sintaxis de typescript

1. Anotaciones
 1. number, string, null, any
2. Interfaces
3. Arrays
4. Tuplas
5. Diccionarios

```
var fecha:Date = new Date()  
  
var numero:number = 33;  
  
var cadena:string = "HOLA"
```

```
interface IPersona {  
    dni:string  
    nombre:string  
}  
interface IPersonaDir extends IPersona {  
    dir:string  
}
```

```
var juan :IPersona = {  
    dni:"1266I",  
    nombre:"juan",  
}
```

```
var juan :IPersona = {  
    dni:"1266I",  
    nombre:"juan",  
    dir:"cl damas",  
}
```

```
var personas : (IPersona | IPersonaDir)[] = [  
    juan,  
    {  
        dni: "111",  
        nombre: "PEDRO"  
    },  
    {  
        dni: "22",  
        nombre: "aaa",  
        dir: "dire"  
    }  
]
```

```
function add ( a:number, b:number): number
{
    return a + b;
}

console.log(add(10, 20));
```

```
console.log(add("a", 12))
```

Error del editor

SError: × Unable to compile TypeScript:
app.ts:56:17 - error TS2345: **Argument of type 'string' is not assignable to parameter of type 'number'.**

```
56 console.log(add("a", 12))
      ~~~
```

Error del compilador

```
var lista : number[] = [1,2,3,4,5]

var pares = lista.filter((numero:number): boolean => numero % 2 == 0);
```

```
var pares = lista.filter((numero:number): boolean => {
    return numero % 2 == 0
});
```

```
interface ICoordenada {  
    Longitud: number  
    Latitud: number  
}  
  
function printPunto(coordenada: ICoordenada): void {  
    console.log(`longitud ${coordenada.Longitud} latitud  
${coordenada.Latitud}`)  
}
```

```
var coordenadas = [  
  { Longitud: 40, Latitud: 20 },  
  { Longitud: 41, Latitud: 20 },  
  { Longitud: 42, Latitud: 20 },  
  { Longitud: 43, Latitud: 20 },  
]  
  
function pagina(primer: number, ultimo?: number): ICoordenada[] {  
  if (ultimo === undefined)  
    return coordenadas.filter((x: ICoordenada, index: number): boolean =>  
      index >= primero)  
  else  
    return coordenadas.filter((x: ICoordenada, index: number): boolean =>  
      index >= primero && index <= ultimo)  
}
```

```
console.log(pagina(2))  
console.log(pagina(0, 2))
```

```
[ { Longitud: 42, Latitud: 20 }, { Longitud: 43, Latitud: 20 } ]  
[  
  { Longitud: 40, Latitud: 20 },  
  { Longitud: 41, Latitud: 20 },  
  { Longitud: 42, Latitud: 20 }  
]
```