



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



Instituto Tecnológico de Milpa Alta II

Asignatura:

Front-End

Carrera:

Ingeniera en Sistemas Computacionales

Tema:

Obtener datos con angular a php

Alumno:

Blancas Aguilar Daniel

Profesor:

Roldan Aquino Segura

Ciclo escolar:

2023-1

Fecha de Entrega:

30/05/2023

A continuación se muestra la aplicación en donde a través de una API se pueden mandar datos por http y el uso de cabeceras, a continuación el proceso.

```
mysql -u root -p
MariaDB [angular]> show tables;
+-----+
| Tables_in angular |
+-----+
| t_articulos        |
+-----+
1 row in set (0.001 sec)

MariaDB [angular]> desc t_articulos;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | int(11) | NO | PRI | NULL | auto_increment |
| producto | varchar(50) | YES | | NULL | |
| precio | float | YES | | NULL | |
+-----+
3 rows in set (0.001 sec)

MariaDB [angular]>
```

Lo primero es crear la base de datos la cual va a contener nuestros registros dependiendo de que sea el proyecto, en este caso, se creo una tabla para los productos con 2 registros.

```
mysql -u root -p
MariaDB [angular]> show tables;
+-----+
| Tables_in angular |
+-----+
| t_articulos        |
+-----+
1 row in set (0.001 sec)

MariaDB [angular]> desc t_articulos;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| id     | int(11) | NO | PRI | NULL | auto_increment |
| producto | varchar(50) | YES | | NULL | |
| precio | float | YES | | NULL | |
+-----+
3 rows in set (0.001 sec)

MariaDB [angular]> select * from t_articulos;
+-----+
| id | producto | precio |
+-----+
| 1 | Chetos Torciditos | 13 |
| 2 | Chetos de bolitas | 13 |
+-----+
2 rows in set (0.000 sec)

MariaDB [angular]>
```

A continuación se crea el proyecto para poder utilizar php nativo haciendo una conexion a la base de datos y utilizando las cabeceras de protocolos para que cualquiera pueda hacer una conexión.

```
danx398@dan-pc-man:/srv/http/www$ cd /srv/http/www
~/srv/http/www$ ls
apuntes documentation Back Front-end Garuda_Project TicTacToe
~/srv/http/www$ mkdir productos
~/srv/http/www$
```

```
danx39@dan-pc-man:srv/http/www/productos
~/srv/http/www
ls
aportes_documentacion Back Front-end Garuda_Proyect TicTacToe
~/srv/http/www
mkdir productos
~/srv/http/www
cd pro
cd: no existe el fichero o el directorio: pro
~/srv/h
cd productos
~/srv/h/w/productos
touch index.php && touch conexion.php
~/srv/h/w/productos
```

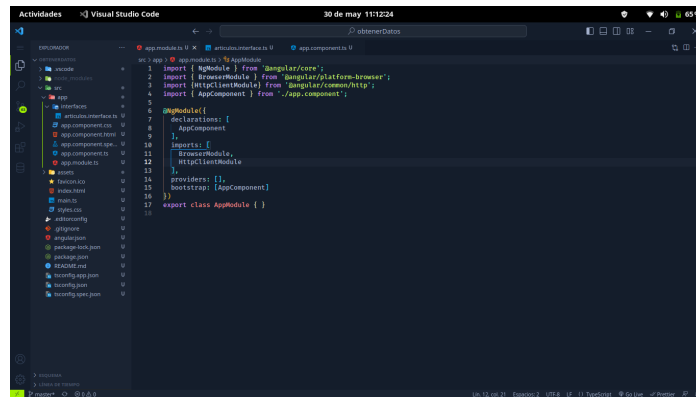
A continuación dentro de cada uno de los archivos se pone lo siguiente:

```
Actividades Visual Studio Code 30 de may 11:03:22
EXPLORADOR
- productos
  - conexion.php
  - index.php
index.php X
1 <?php
2 header('Access-Control-Allow-Origin: *');
3 header('Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept');
4 include "conexion.php";
5 $conexion = new Conexion();
6 $con = $conexion->conectar();
7 $query = "SELECT id, producto, precio FROM t_articulos";
8 $result = mysqli_query($con, $query);
9 $item = [];
10 while ($ver = mysqli_fetch_assoc($result)){
11     $item[] = $ver;
12 }
13 $data = json_encode($item);
14 echo $data;
15
```

```
Actividades Visual Studio Code 30 de may 11:03:29
EXPLORADOR
- productos
  - conexion.php
  - index.php
conexion.php X
1 <?php
2 class Conexion {
3     public function conectar(){
4         return mysqli_connect('localhost','root','','angular');
5     }
6 }
7
```

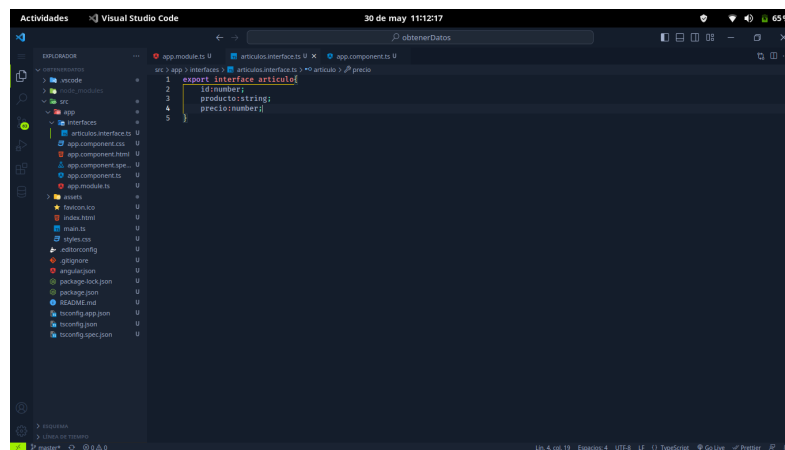
A continuación el siguiente paso es crear el proyecto de angular con el siguiente comando : \$ **ng new obtenerDatos**

En donde lo primero sera importar una librería para poder hacer la llamada de un protocolo HTTP

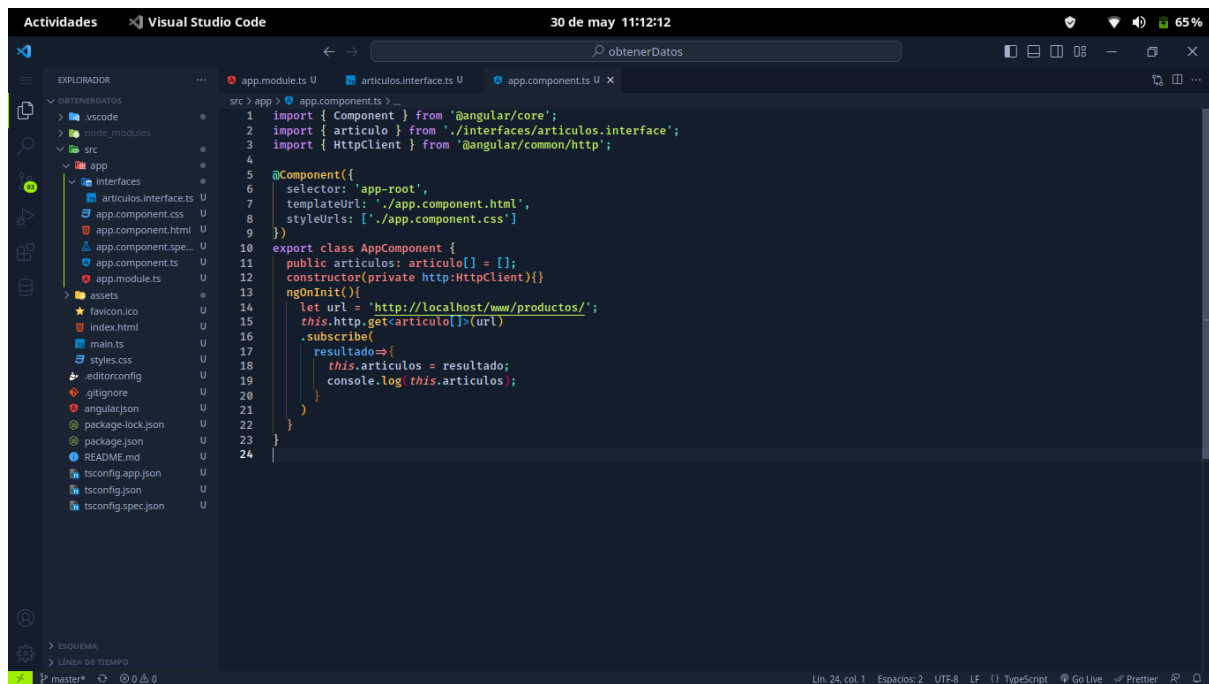


La librería en cuestión se llama HttpClientModule que es la encargada de poder utilizar protocolos http.

Lo que sigue es poder crear una interface la cual va a utilizar para poder definir los tipos de datos con la siguiente nomenclatura

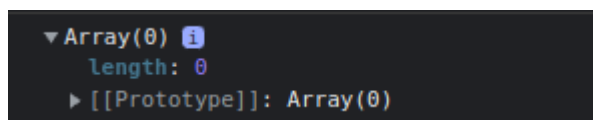


Y por último se configura y se hace la parte de una respuesta.



```
src > app > app.component.ts >
1 import { Component } from '@angular/core';
2 import { articulo } from '../interfaces/articulos.interface';
3 import { HttpClient } from '@angular/common/http';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent {
11   public articulos: articulo[] = [];
12   constructor(private http: HttpClient){}
13   ngOnInit(){
14     let url = 'http://localhost/www/productos/';
15     this.http.get<articulo[]>(url)
16       .subscribe(
17         resultado => {
18           this.articulos = resultado;
19           console.log(this.articulos);
20         }
21       )
22   }
23 }
24
```

En donde a través de un mensaje en consola muestra los registros previamente insertados.



En este caso al estar vacío así se muestra, a continuación se hace la parte de la creación en una tabla con los registros y la validación para que muestre si está vacía o no.

En donde si la tabla tiene datos se ve de la siguiente manera:

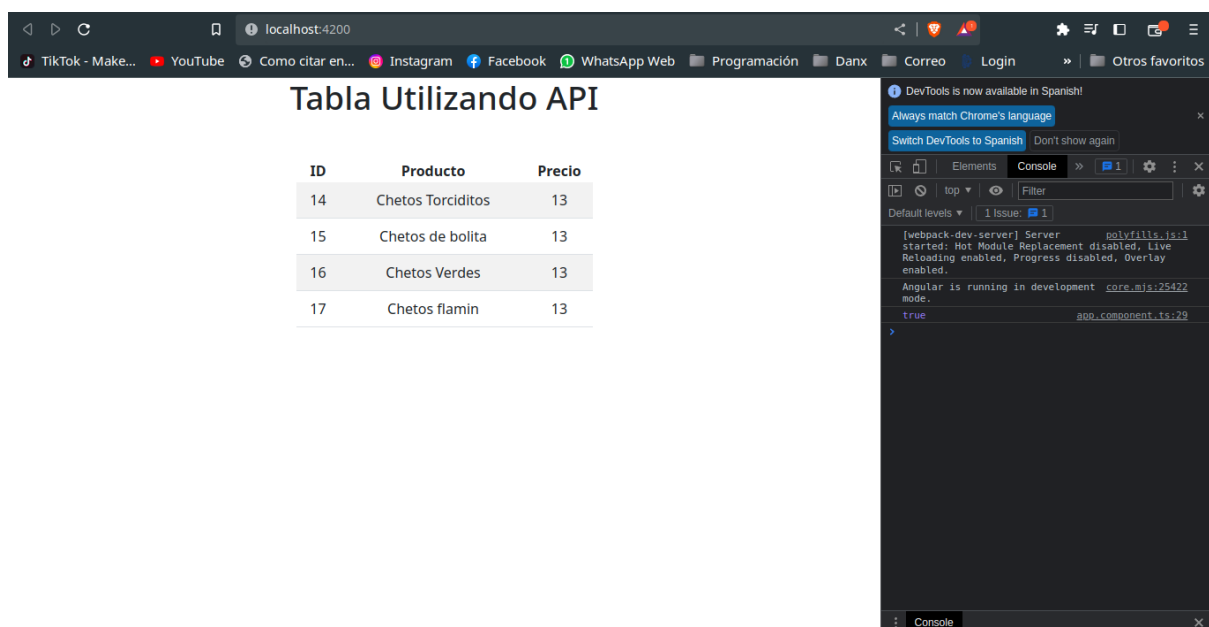


Tabla Utilizando API

ID	Producto	Precio
14	Chetos Torciditos	13
15	Chetos de bolita	13
16	Chetos Verdes	13
17	Chetos flamin	13

DevTools is now available in Spanish! [Always match Chrome's language](#) [Switch DevTools to Spanish](#) Don't show again

Elements Console >> 1

Default levels 1 issue 1

[webpack-dev-server] Server started: Hot Module Replacement disabled, Live Reloading enabled, Progress disabled, Overlay enabled.

Angular is running in development mode, core.mjs:25422

true app.component.ts:29

En caso contrario si esta no tiene datos se puede ver de la siguiente manera y con el siguiente mensaje.

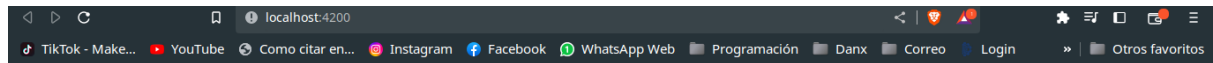


Tabla Utilizando API

No hay datos en la base

Utilizando las directivas `*ngIf= ''` y `*ngFor = ''` para poder hacer lo siguiente.