

Ejecución del proyecto PHP y Angula con MySQL

Este documento redactare el paso a paso para la ejecución del proyecto de angular con base de datos MySQL, utilizando PHP como lenguaje de comunicación entre Angula y MySQL.

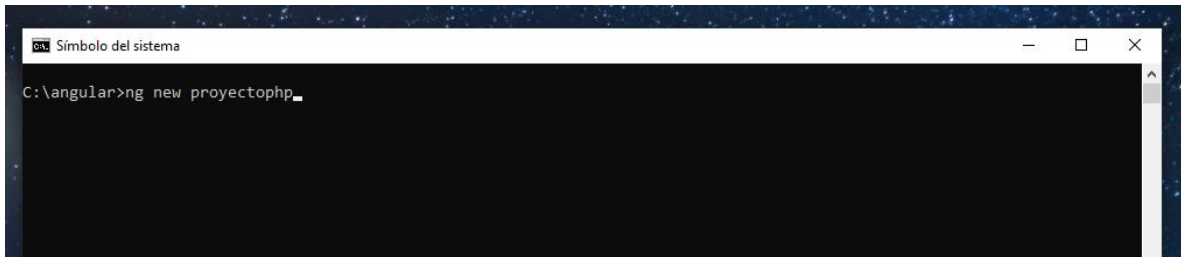
Debemos tener en cuenta que para poder ejecutar este proyecto tendremos que tener instaladas una serie de programas necesarios para la ejecución como lo son:

- Editor de condigo (para este caso usare Visual Studio Code)
- Base de datos MySql
- Angular CLI.

Teniendo esto claro, comenzaremos haciendo una aplicación CRUD que permita gestionar información de artículos de una tienda.

Inicialmente debemos crear un proyecto con angular CLI, creamos una carpeta en el Disco C y desde la línea de comandos CMD de Windows escribimos es siguiente comando.

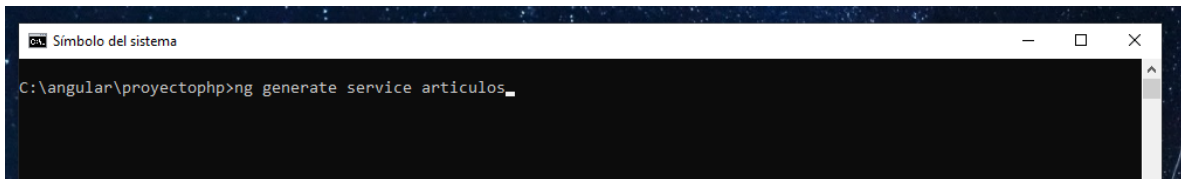
```
ng new proyectophp
```



```
Simbolo del sistema
C:\angular>ng new proyectophp_
```

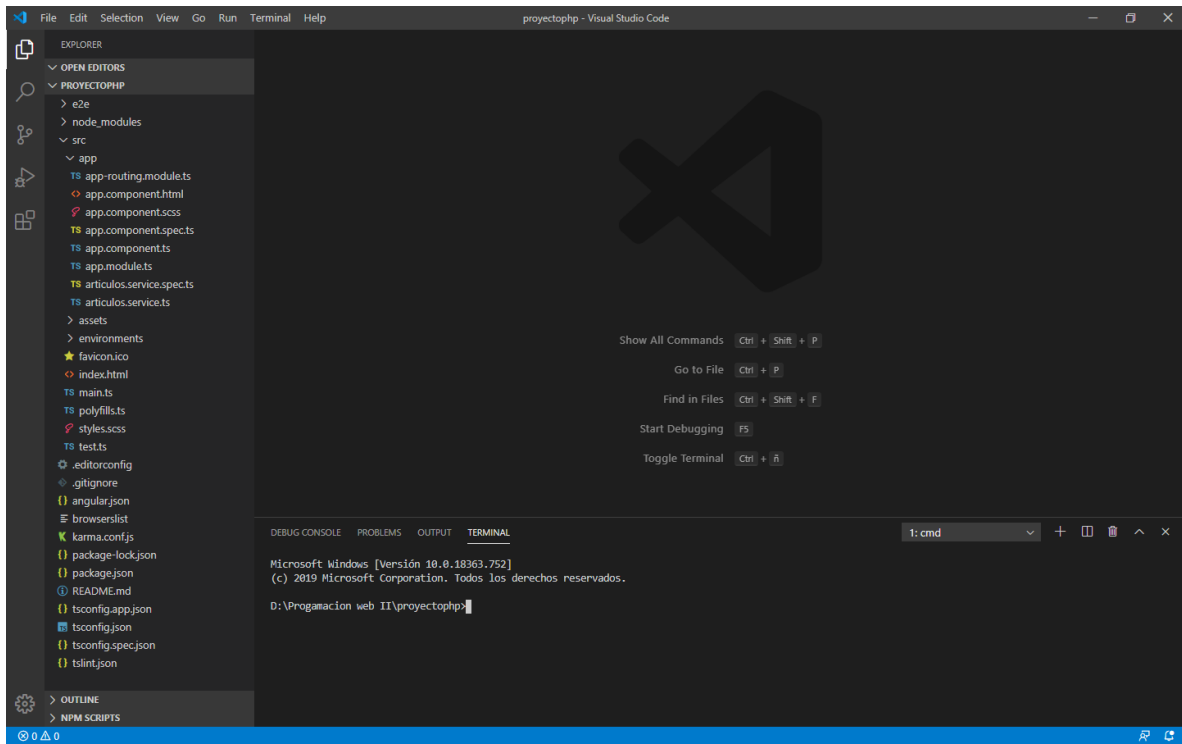
Después de esto creamos el servicio denominado artículos desde la ruta del proyecto

```
ng generate service artículos
```



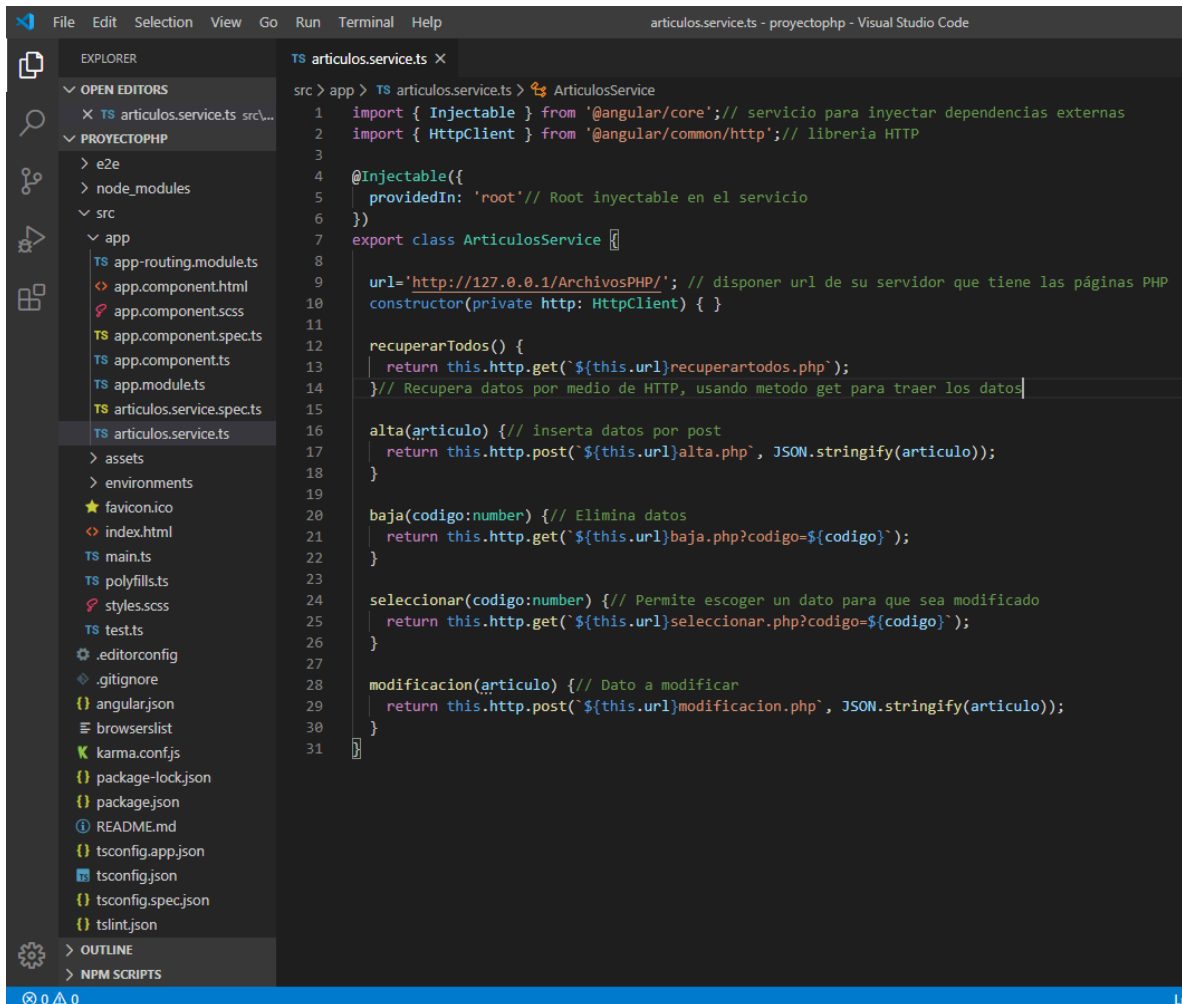
```
Simbolo del sistema
C:\angular\proyectophp>ng generate service artículos_
```

Abrimos el proyecto en el editor de código y quedaría de la siguiente forma:



El servicio artículos es el que se encarga de conectar con la URL del servicio y definir las funciones para utilizar los métodos GET y POST asociados a las peticiones en PHP.

Quedaría de la siguiente forma:

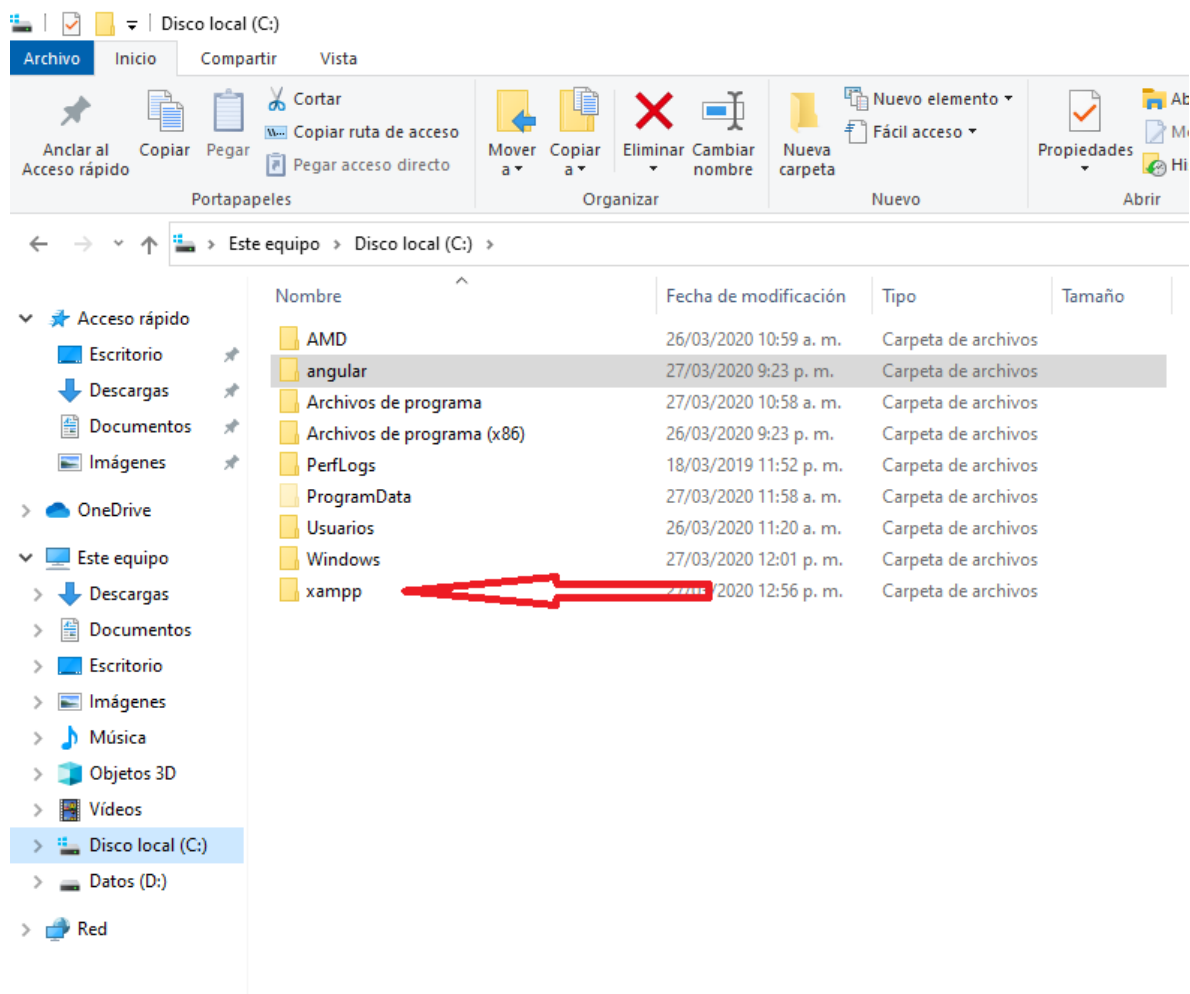


```
File Edit Selection View Go Run Terminal Help
articulos.service.ts - proyectophp - Visual Studio Code

EXPLORER
OPEN EDITORS
  X TS articulos.service.ts src\...
PROYECTOPHP
  > e2e
  > node_modules
  > src
    > app
      TS app-routing.module.ts
      < app.component.html
      < app.component.scss
      TS app.component.spec.ts
      TS app.component.ts
      TS app.module.ts
      TS articulos.service.spec.ts
      TS articulos.service.ts
    > assets
    > environments
    > favicon.ico
    > index.html
    > main.ts
    > polyfills.ts
    > styles.scss
    > test.ts
    > .editorconfig
    > .gitignore
    > angular.json
    > browserslist
    > karma.conf.js
    > package-lock.json
    > package.json
    > README.md
    > tsconfig.app.json
    > tsconfig.json
    > tsconfig.spec.json
    > tslint.json
  > OUTLINE
  > NPM SCRIPTS

TS articulos.service.ts X
src > app > TS articulos.service.ts > ArticulosService
1 import { Injectable } from '@angular/core'; // servicio para inyectar dependencias externas
2 import { HttpClient } from '@angular/common/http'; // librería HTTP
3
4 @Injectable({
5   providedIn: 'root' // Root injectable en el servicio
6 })
7 export class ArticulosService {
8
9   url='http://127.0.0.1/ArchivosPHP/'; // disponer url de su servidor que tiene las páginas PHP
10  constructor(private http: HttpClient) { }
11
12  recuperarTodos() {
13    return this.http.get(`${this.url}recuperartodos.php`);
14  } // Recupera datos por medio de HTTP, usando metodo get para traer los datos
15
16  alta(articulo) { // inserta datos por post
17    return this.http.post(`${this.url}alta.php`, JSON.stringify(articulo));
18  }
19
20  baja(codigo:number) { // Elimina datos
21    return this.http.get(`${this.url}baja.php?codigo=${codigo}`);
22  }
23
24  seleccionar(codigo:number) { // Permite escoger un dato para que sea modificado
25    return this.http.get(`${this.url}seleccionar.php?codigo=${codigo}`);
26  }
27
28  modificacion(articulo) { // Dato a modificar
29    return this.http.post(`${this.url}modificacion.php`, JSON.stringify(articulo));
30  }
31 }
```

En esta parte hay que aclarar que se debe copiar la carpeta con los archivos php (están adjuntos en la aplicación del proyecto) dentro de la ruta del programa xampp que es donde esta ubicada la base de datos.



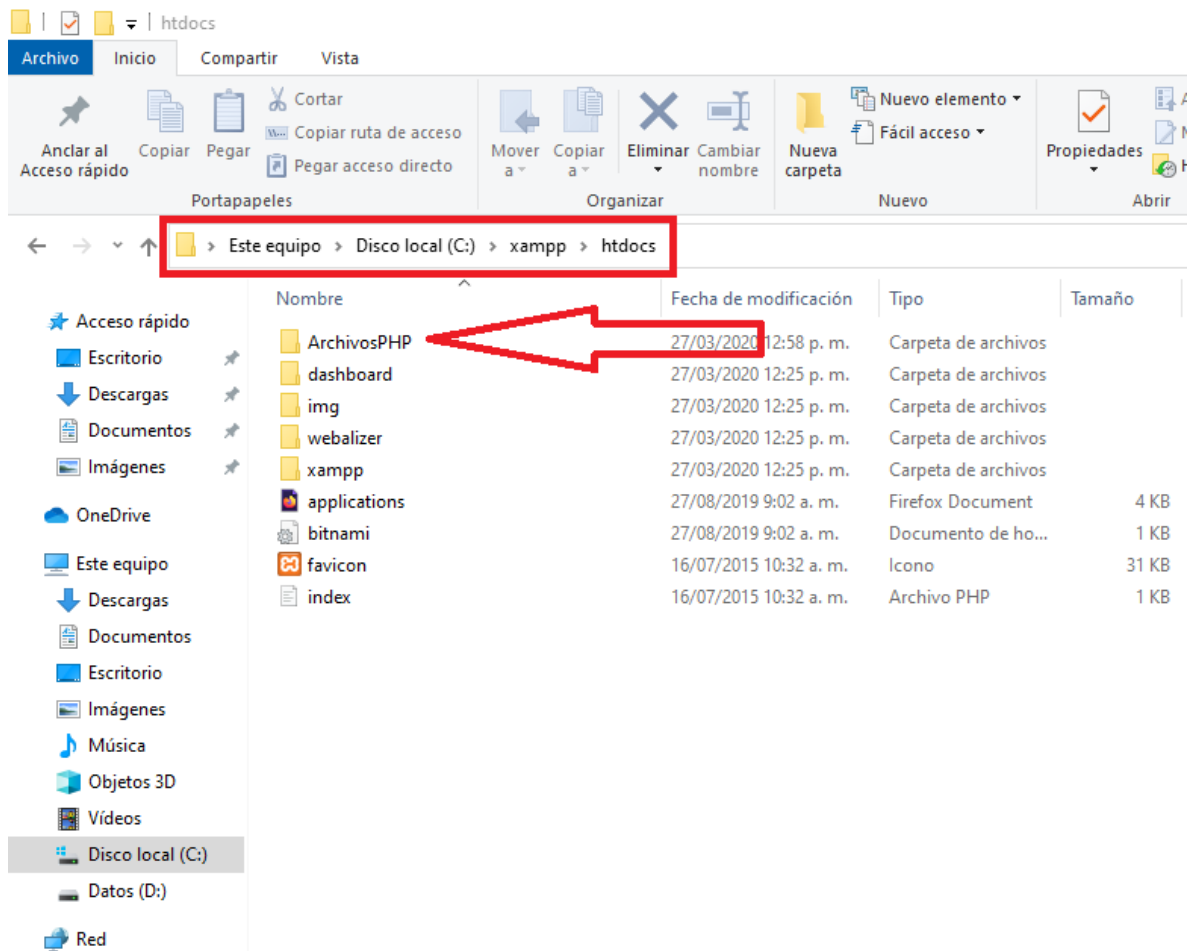
File Explorer window showing the contents of the 'xampp' folder on the local disk (C:). The address bar shows the path: > Este equipo > Disco local (C:) > xampp >.

The left sidebar shows the navigation pane with 'Disco local (C:)' selected.

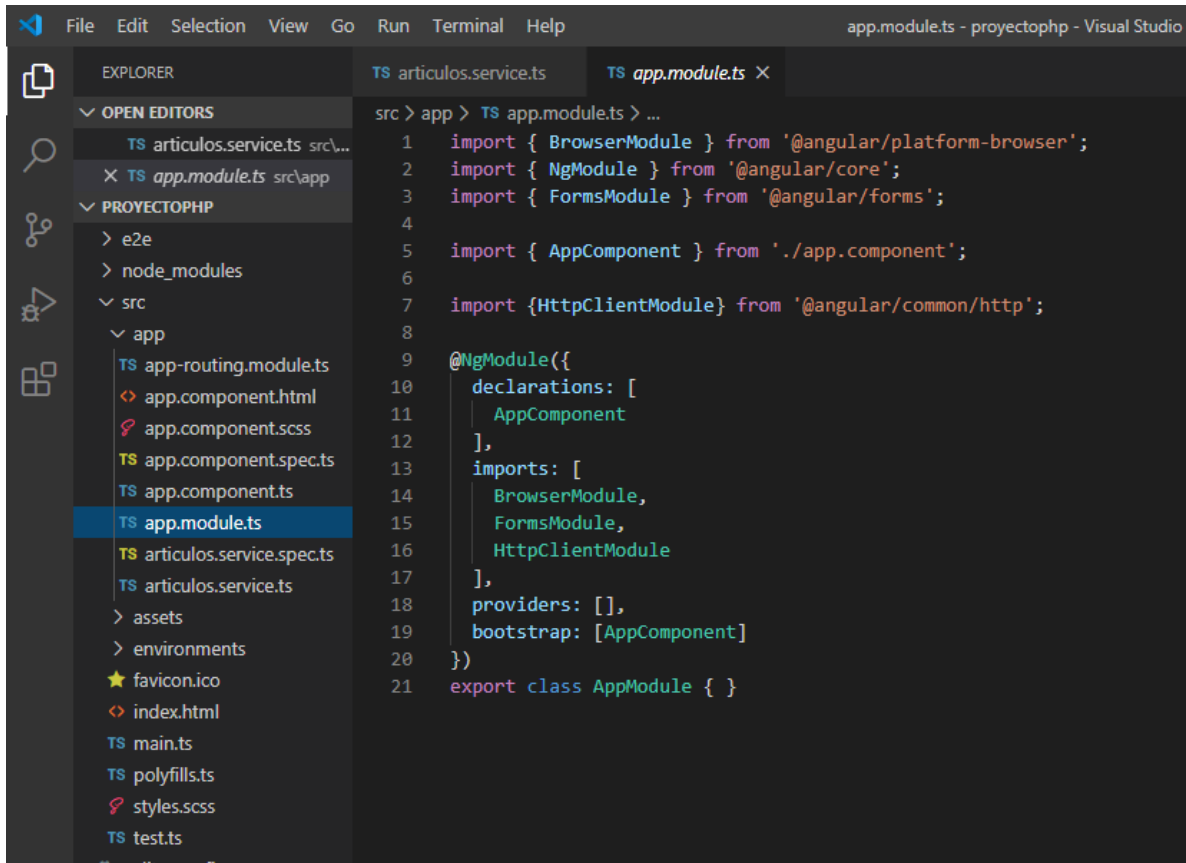
The main pane displays a list of files and folders. A red arrow points to the 'htdocs' folder.

| Nombre | Fecha de modificación | Tipo | Tamaño |
|------------------|------------------------|-----------------------|--------|
| anonymous | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| apache | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| cgi-bin | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| contrib | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| FileZillaFTP | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| htdocs | 27/03/2020 12:10 p. m. | Carpeta de archivos | |
| img | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| install | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| licenses | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| locale | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| mailoutput | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| mailtodisk | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| MercuryMail | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| mysql | 27/03/2020 12:26 p. m. | Carpeta de archivos | |
| perl | 27/03/2020 12:28 p. m. | Carpeta de archivos | |
| php | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| phpMyAdmin | 27/03/2020 1:21 p. m. | Carpeta de archivos | |
| sendmail | 27/03/2020 12:30 p. m. | Carpeta de archivos | |
| src | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| tmp | 27/03/2020 1:21 p. m. | Carpeta de archivos | |
| tomcat | 27/03/2020 12:26 p. m. | Carpeta de archivos | |
| webalizer | 27/03/2020 12:31 p. m. | Carpeta de archivos | |
| webdav | 27/03/2020 12:25 p. m. | Carpeta de archivos | |
| apache_start | 7/06/2013 6:15 a. m. | Archivo por lotes ... | 1 KB |
| apache_stop | 27/03/2020 12:31 p. m. | Archivo por lotes ... | 1 KB |
| catalina_service | 30/03/2013 7:29 a. m. | Archivo por lotes ... | 10 KB |
| catalina_start | 22/10/2019 8:36 a. m. | Archivo por lotes ... | 5 KB |
| catalina_stop | 22/10/2019 8:35 a. m. | Archivo por lotes ... | 5 KB |
| ctlscript | 27/03/2020 12:25 p. m. | Archivo por lotes ... | 3 KB |
| filezilla setup | 30/03/2013 7:29 a. m. | Archivo por lotes ... | 1 KB |

52 elementos



El app.module.ts importa las dependencias de HTTP para poder establecer comunicación con los servicios en PHP.



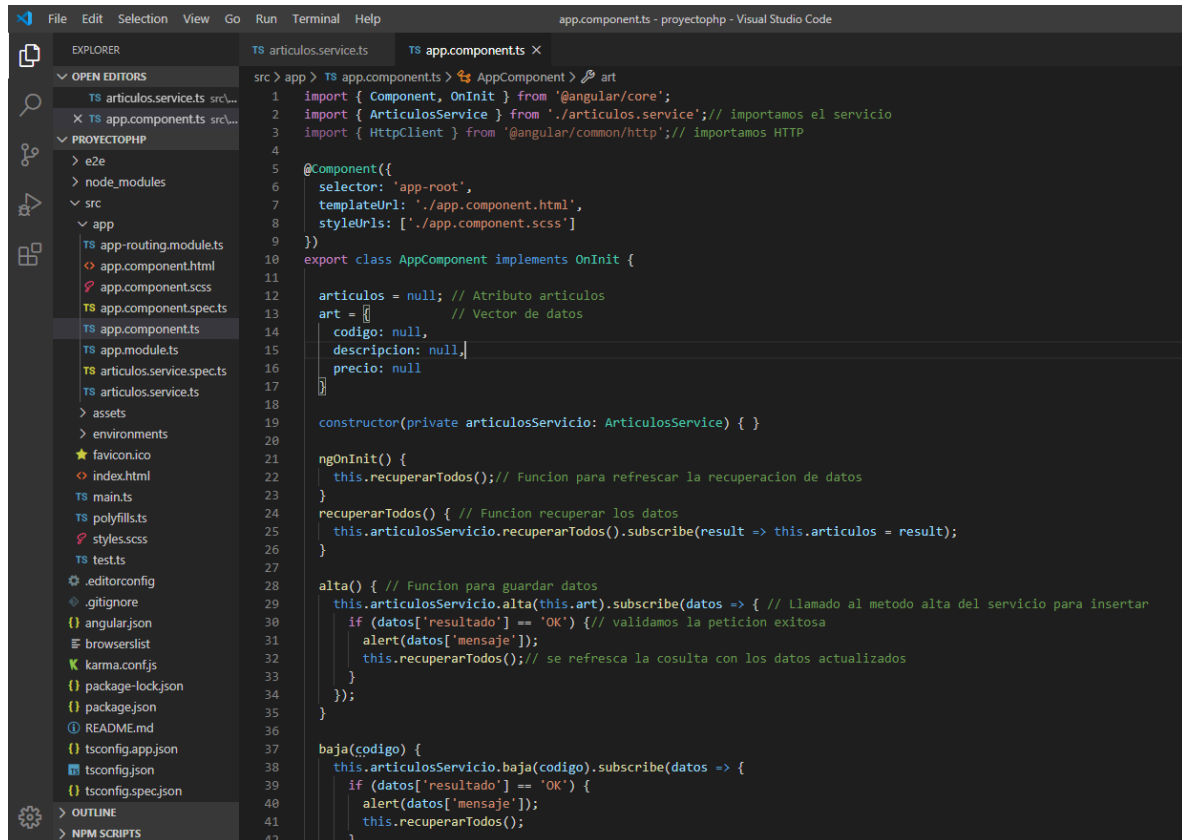
The screenshot shows the Visual Studio interface with the Explorer on the left and the Code Editor on the right. The Explorer shows the project structure with the following files and folders:

- OPEN EDITORS
 - TS articulos.service.ts src\...
 - TS app.module.ts src\app
- PROYECTOPHP
 - e2e
 - node_modules
 - src
 - app
 - TS app-routing.module.ts
 - app.component.html
 - app.component.scss
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.module.ts
 - TS articulos.service.spec.ts
 - TS articulos.service.ts
 - assets
 - environments
 - favicon.ico
 - index.html
 - main.ts
 - polyfills.ts
 - styles.scss
 - test.ts

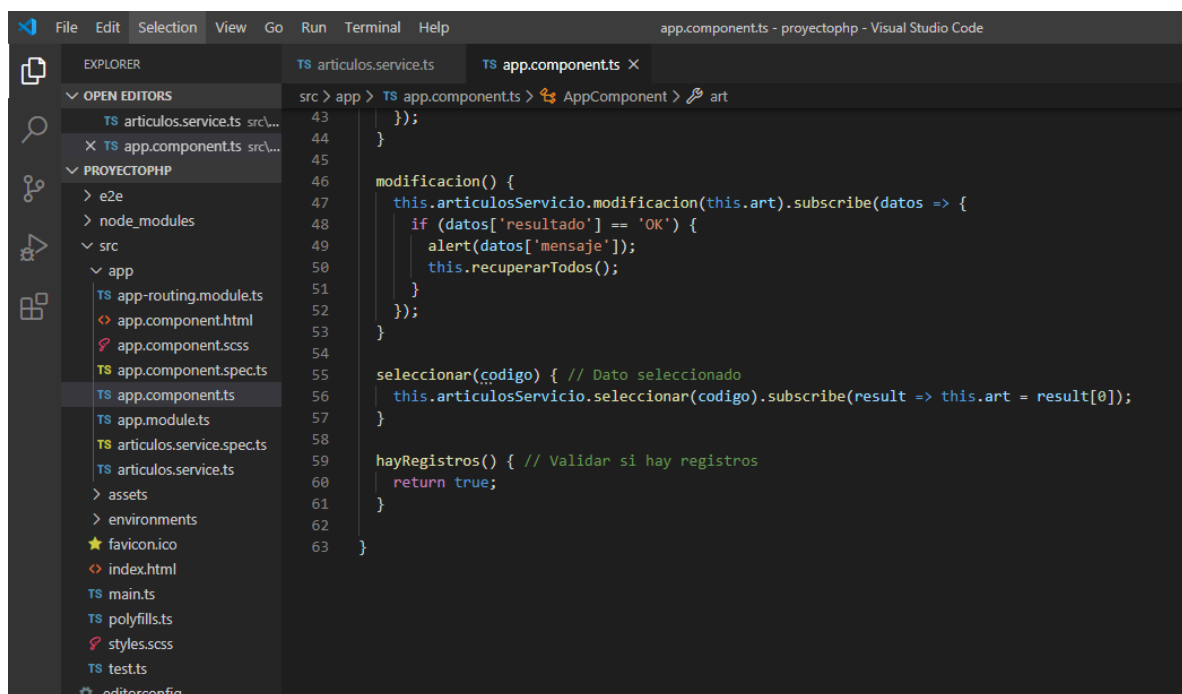
The Code Editor shows the content of app.module.ts:

```
src > app > TS app.module.ts > ...
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3  import { FormsModule } from '@angular/forms';
4
5  import { AppComponent } from './app.component';
6
7  import { HttpClientModule } from '@angular/common/http';
8
9  @NgModule({
10     declarations: [
11         AppComponent
12     ],
13     imports: [
14         BrowserModule,
15         FormsModule,
16         HttpClientModule
17     ],
18     providers: [],
19     bootstrap: [AppComponent]
20 })
21 export class AppModule { }
```

El app.component.ts permite comunicarse con el servicio para realizar las operaciones CRUD definidas.

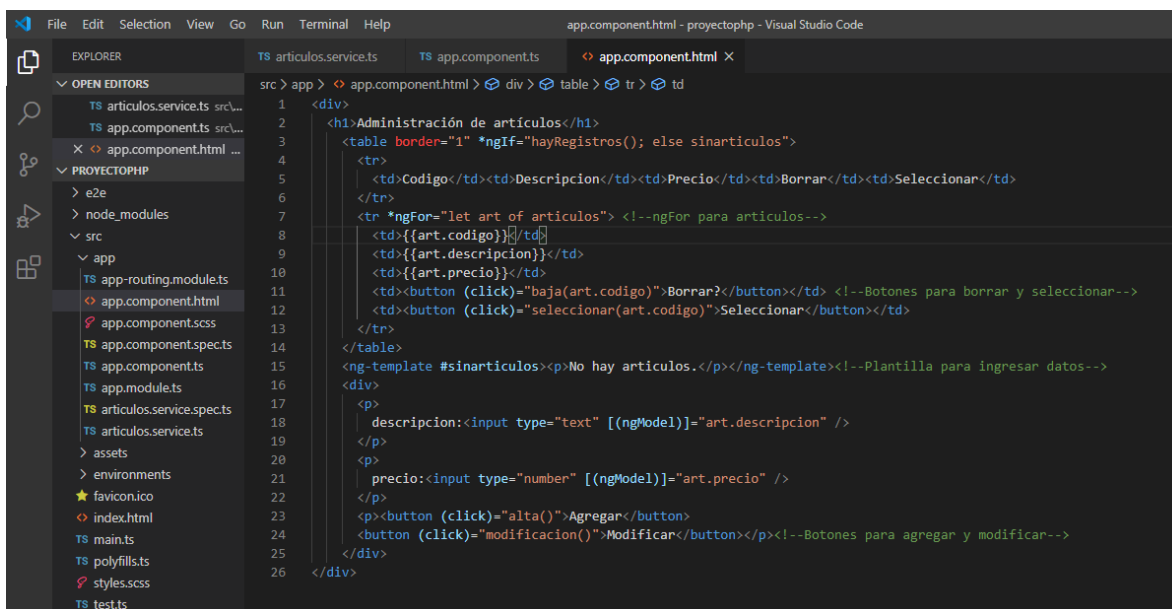


```
1 import { Component, OnInit } from '@angular/core';
2 import { ArticulosService } from './articulos.service'; // importamos el servicio
3 import { HttpClient } from '@angular/common/http'; // importamos HTTP
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.scss']
9 })
10 export class AppComponent implements OnInit {
11
12   articulos = null; // Atributo articulos
13   art = [] // Vector de datos
14   codigo: null,
15   descripcion: null,
16   precio: null
17 }
18
19 constructor(private articulosServicio: ArticulosService) { }
20
21 ngOnInit() {
22   this.recuperarTodos(); // Funcion para refrescar la recuperacion de datos
23 }
24
25 recuperarTodos() { // Funcion recuperar los datos
26   this.articulosServicio.recuperarTodos().subscribe(result => this.articulos = result);
27 }
28
29 alta() { // Funcion para guardar datos
30   this.articulosServicio.alta(this.art).subscribe(datos => { // Llamado al metodo alta del servicio para insertar
31     if (datos['resultado'] == 'OK') { // validamos la peticion exitosa
32       alert(datos['mensaje']);
33       this.recuperarTodos(); // se refresca la consulta con los datos actualizados
34     }
35   });
36 }
37
38 baja(codigo) {
39   this.articulosServicio.baja(codigo).subscribe(datos => {
40     if (datos['resultado'] == 'OK') {
41       alert(datos['mensaje']);
42       this.recuperarTodos();
43     }
44   });
45 }
46
47 modificacion() {
48   this.articulosServicio.modificacion(this.art).subscribe(datos => {
49     if (datos['resultado'] == 'OK') {
50       alert(datos['mensaje']);
51       this.recuperarTodos();
52     }
53   });
54 }
55
56 seleccionar(codigo) { // Dato seleccionado
57   this.articulosServicio.seleccionar(codigo).subscribe(result => this.art = result[0]);
58 }
59
60 hayRegistros() { // Validar si hay registros
61   return true;
62 }
63 }
```



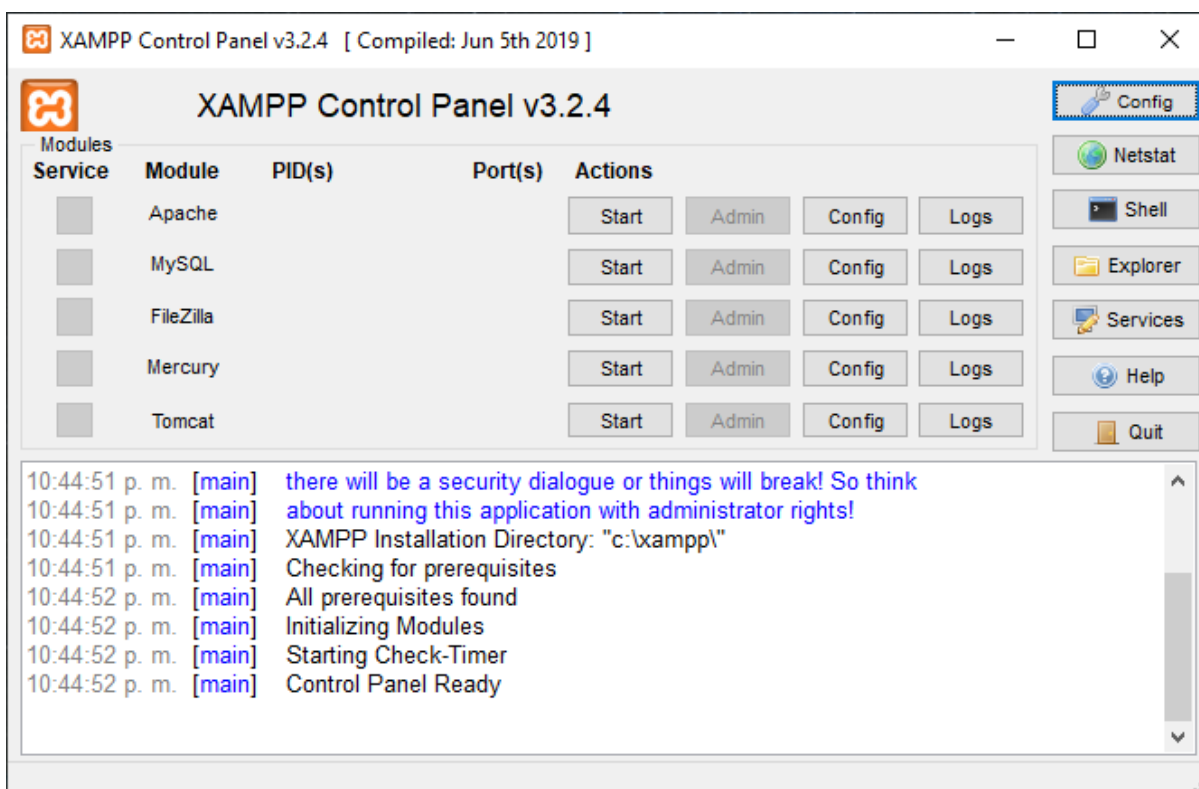
```
1 import { Component, OnInit } from '@angular/core';
2 import { ArticulosService } from './articulos.service'; // importamos el servicio
3 import { HttpClient } from '@angular/common/http'; // importamos HTTP
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.scss']
9 })
10 export class AppComponent implements OnInit {
11
12   articulos = null; // Atributo articulos
13   art = [] // Vector de datos
14   codigo: null,
15   descripcion: null,
16   precio: null
17 }
18
19 constructor(private articulosServicio: ArticulosService) { }
20
21 ngOnInit() {
22   this.recuperarTodos(); // Funcion para refrescar la recuperacion de datos
23 }
24
25 recuperarTodos() { // Funcion recuperar los datos
26   this.articulosServicio.recuperarTodos().subscribe(result => this.articulos = result);
27 }
28
29 alta() { // Funcion para guardar datos
30   this.articulosServicio.alta(this.art).subscribe(datos => { // Llamado al metodo alta del servicio para insertar
31     if (datos['resultado'] == 'OK') { // validamos la peticion exitosa
32       alert(datos['mensaje']);
33       this.recuperarTodos(); // se refresca la consulta con los datos actualizados
34     }
35   });
36 }
37
38 baja(codigo) {
39   this.articulosServicio.baja(codigo).subscribe(datos => {
40     if (datos['resultado'] == 'OK') {
41       alert(datos['mensaje']);
42       this.recuperarTodos();
43     }
44   });
45 }
46
47 modificacion() {
48   this.articulosServicio.modificacion(this.art).subscribe(datos => {
49     if (datos['resultado'] == 'OK') {
50       alert(datos['mensaje']);
51       this.recuperarTodos();
52     }
53   });
54 }
55
56 seleccionar(codigo) { // Dato seleccionado
57   this.articulosServicio.seleccionar(codigo).subscribe(result => this.art = result[0]);
58 }
59
60 hayRegistros() { // Validar si hay registros
61   return true;
62 }
63 }
```


El archivo app.component.html presenta la siguiente estructura.

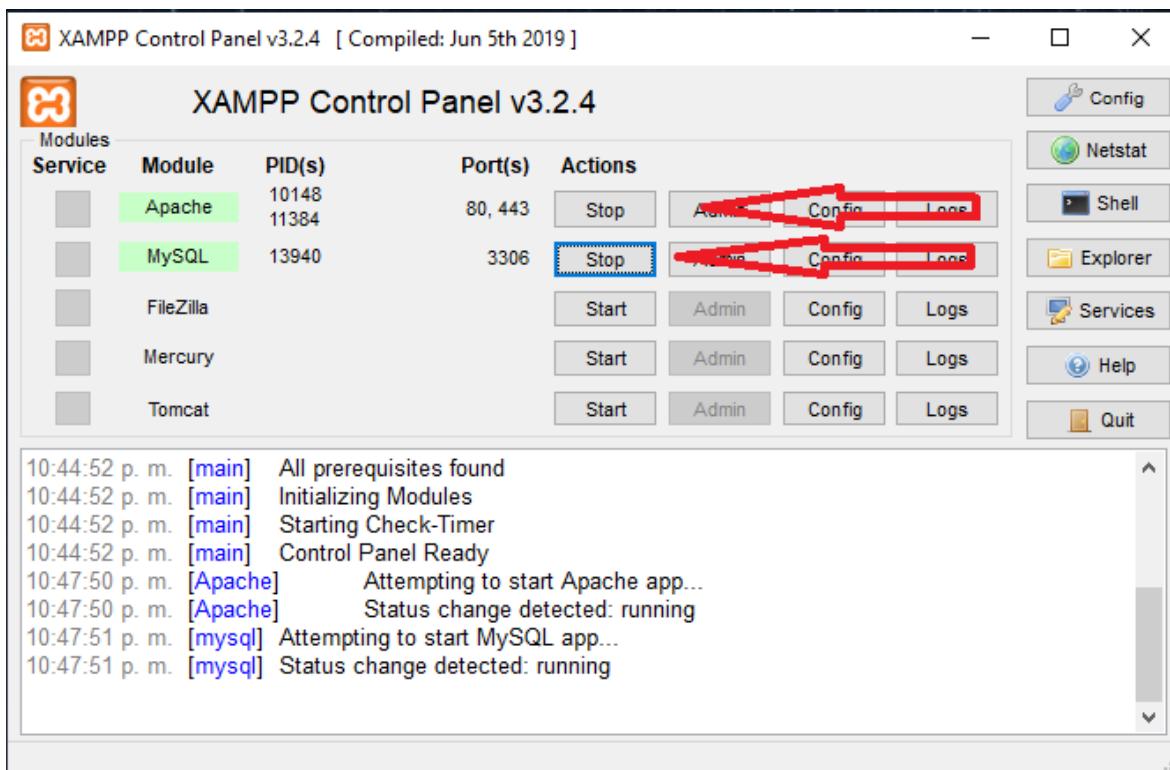


```
1 <div>
2   <h1>Administración de artículos</h1>
3   <table border="1" *ngIf="hayRegistros(); else sinarticulos">
4     <tr>
5       <td>Codigo</td><td>Descripcion</td><td>Precio</td><td>Borrar</td><td>Seleccionar</td>
6     </tr>
7     <tr *ngFor="let art of articulos"> <!--ngFor para articulos-->
8       <td>{{art.codigo}}</td>
9       <td>{{art.descripcion}}</td>
10      <td>{{art.precio}}</td>
11      <td><button (click)="baja(art.codigo)">Borrar</button></td> <!--Botones para borrar y seleccionar-->
12      <td><button (click)="seleccionar(art.codigo)">Seleccionar</button></td>
13    </tr>
14  </table>
15  <ng-template #sinarticulos><p>No hay articulos.</p></ng-template><!--Plantilla para ingresar datos-->
16  <div>
17    <p>
18      descripcion:<input type="text" [(ngModel)]="art.descripcion" />
19    </p>
20    <p>
21      precio:<input type="number" [(ngModel)]="art.precio" />
22    </p>
23    <p><button (click)="alta()">Agregar</button>
24      <button (click)="modificacion()">Modificar</button></p><!--Botones para agregar y modificar-->
25  </div>
26 </div>
```

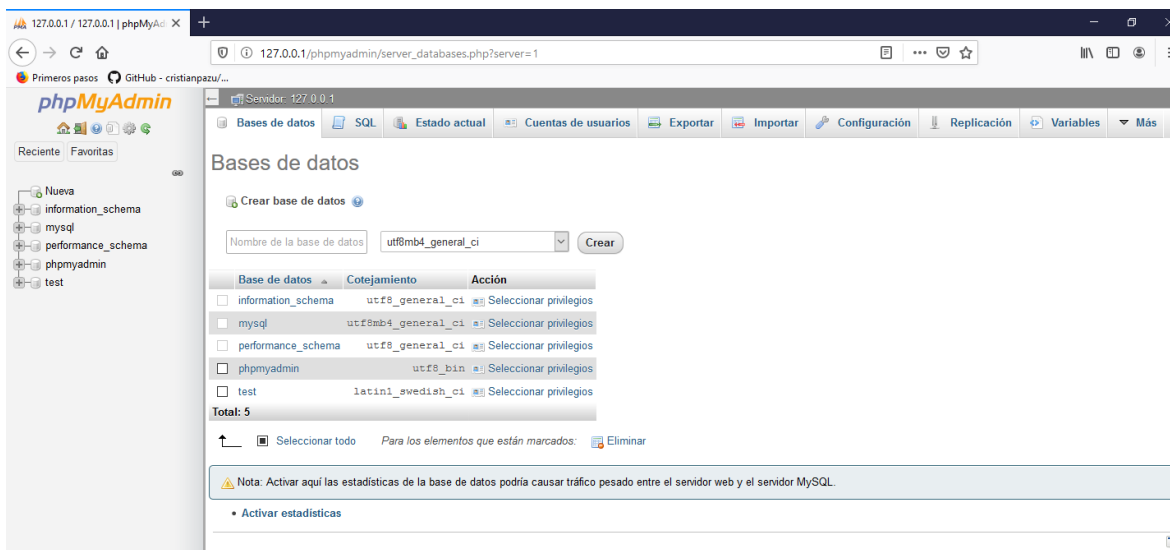
Para ejecutar el gestor de base de datos, debemos abrir la aplicación XAMPP Control Panel ubicada en el menú de inicio.



Se inician los siguientes servicios.



Después de esto abrimos el navegador e iniciamos el gestor de base de datos MySQL, en el navegador ingresamos la siguiente dirección <http://127.0.0.1/phpmyadmin/>



Finalmente creamos la base de datos llamada bd1 con la siguiente estructura.

```
CREATE TABLE articulos (
```

```
    codigo int AUTO_INCREMENT,
```

```
    descripcion varchar(50),
```

```
    precio float,
```

```
    PRIMARY KEY (codigo)
```

```
);
```

127.0.0.1 / 127.0.0.1 | phpMyAdmin

127.0.0.1/phpmyadmin/server_databases.php?server=1

Primeros pasos GitHub - cristianpazu/...

phpMyAdmin

Reciente Favoritas

Nueva

- information_schema
- mysql
- performance_schema
- phpmyadmin
- test

Servidor: 127.0.0.1

Bases de datos SQL Estado actual Cuentas de usuarios Exportar

Bases de datos

Crear base de datos

bd1 utf8mb4_general_ci **Crear**

| Base de datos | Cotejamiento | Acción |
|---|--------------------|---|
| <input type="checkbox"/> information_schema | utf8_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> mysql | utf8mb4_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> performance_schema | utf8_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> phpmyadmin | utf8_bin | Seleccionar privilegios |
| <input type="checkbox"/> test | latin1_swedish_ci | Seleccionar privilegios |

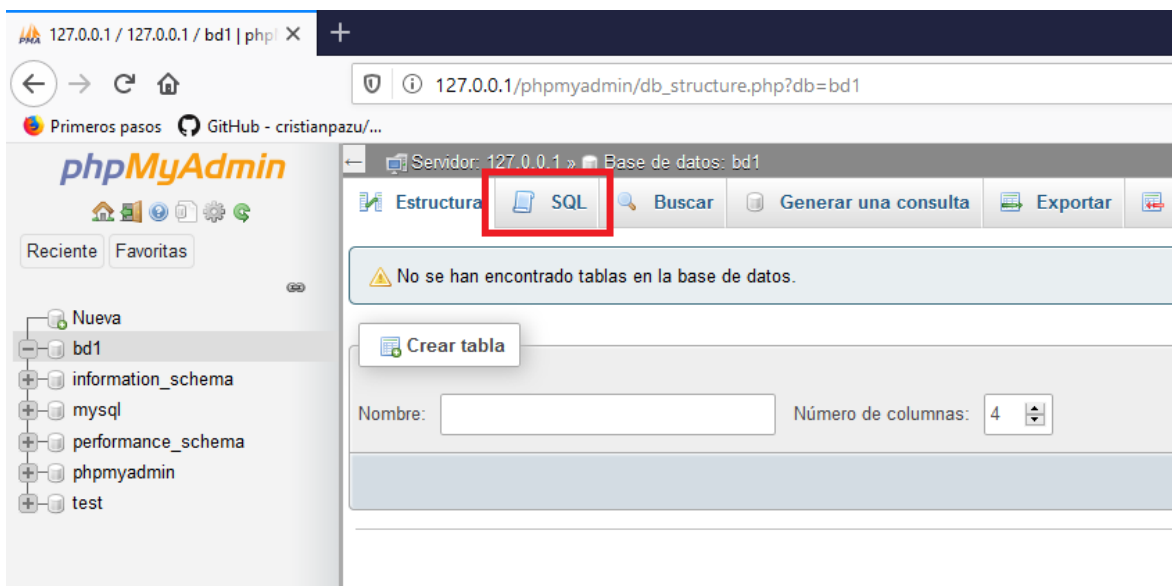
Total: 5

☐ Seleccionar todo Para los elementos que están marcados: [Eliminar](#)

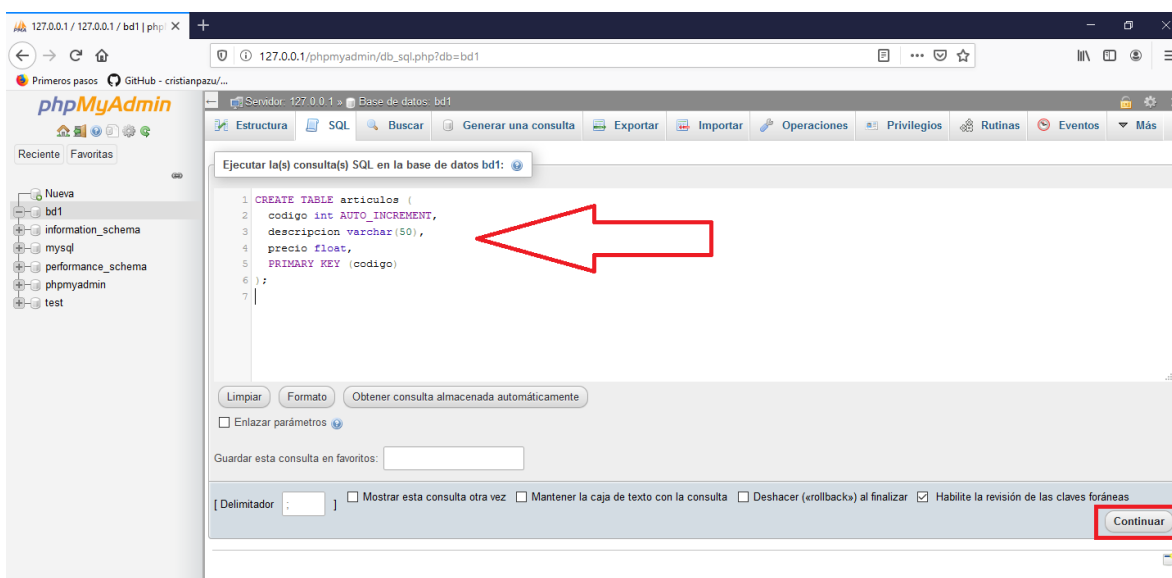
Nota: Activar aquí las estadísticas de la base de datos podría causar tráfico pesado entre el servidor web y el servidor de base de datos.

- [Activar estadísticas](#)

Nos aparecerá la siguiente pantalla.



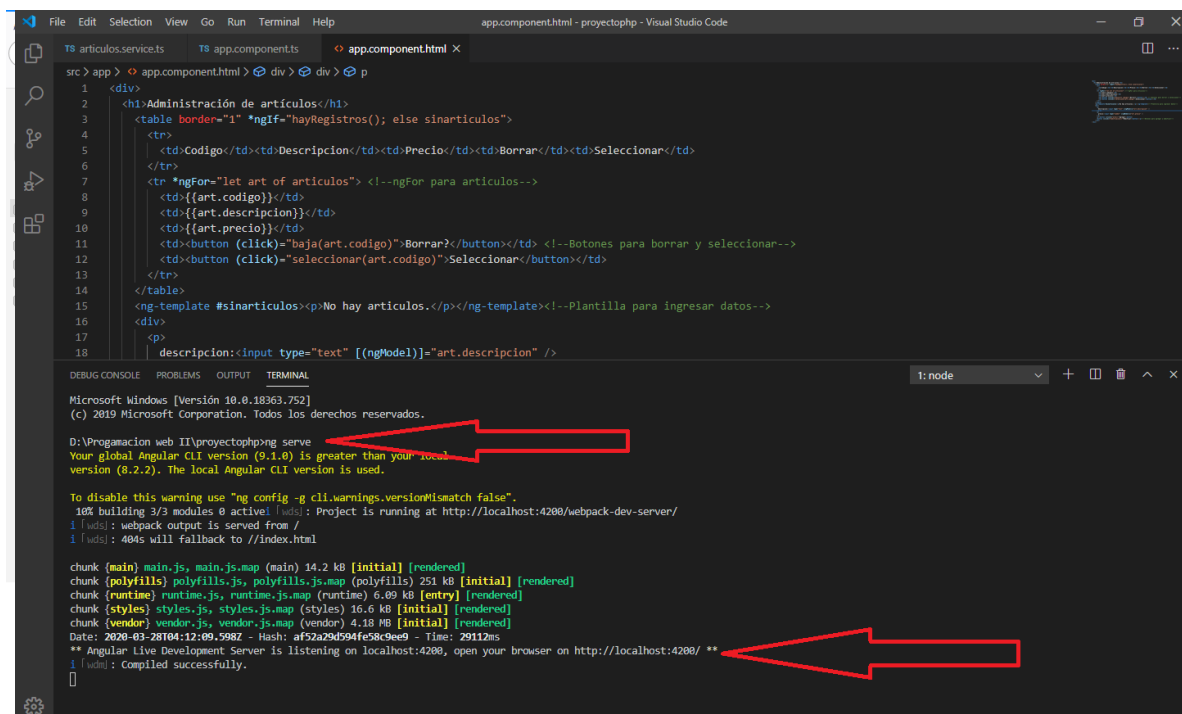
En esta parte ubicamos el SQL, pegamos la estructura de la tabla y finalizamos con continuar.



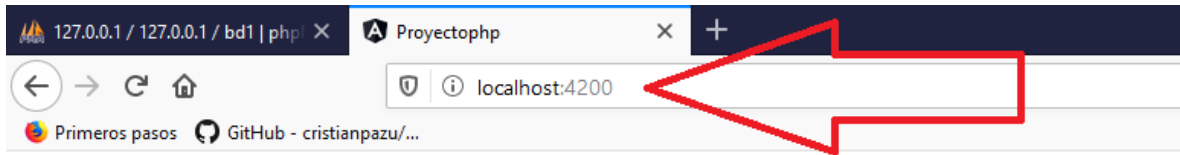
Una vez terminada nos aparecerá la siguiente pantalla donde nos muestra que se creó correctamente.



Terminada la base de datos, ejecutamos el proyecto abriendo una nueva terminal y ejecutando la siguiente línea de comandos ng serve.



Nos indicara que debemos abrir el navegador e ingresar la siguiente dirección <http://localhost:4200/>



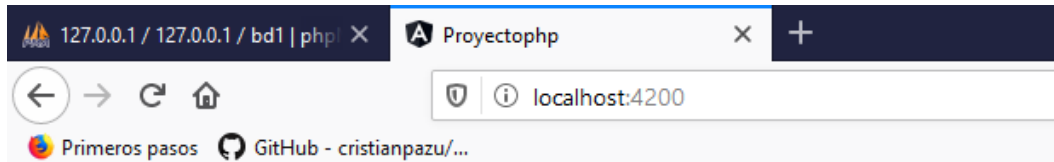
Administración de artículos

| Codigo | Descripcion | Precio | Borrar | Seleccionar |
|--------|-------------|--------|--------|-------------|
|--------|-------------|--------|--------|-------------|

descripcion:

precio:

Vemos como en la aplicación se puede realizar el CRUD completo.



Administración de artículos

| Codigo | Descripcion | Precio | Borrar | Seleccionar |
|--------|-------------|--------|--|--|
| 1 | Papas | 2000 | <input data-bbox="625 1402 722 1438" type="button" value="Borrar?"/> | <input data-bbox="755 1402 868 1438" type="button" value="Seleccionar"/> |
| 2 | Naranjas | 1200 | <input data-bbox="625 1449 722 1484" type="button" value="Borrar?"/> | <input data-bbox="755 1449 868 1484" type="button" value="Seleccionar"/> |
| 3 | Manzanas | 2500 | <input data-bbox="625 1495 722 1530" type="button" value="Borrar?"/> | <input data-bbox="755 1495 868 1530" type="button" value="Seleccionar"/> |
| 4 | Lechugas | 1800 | <input data-bbox="625 1541 722 1577" type="button" value="Borrar?"/> | <input data-bbox="755 1541 868 1577" type="button" value="Seleccionar"/> |

descripcion:

precio: