



INSTITUTO POLITECNICO NACIONAL “IPN”



ESCUELA SUPERIOR DE COMPUTO “ESCOM”

LANDEROS CORTES PEDRO JONAS

SISTEMAS DISTRIBUIDOS

TAREA 8

2023630402

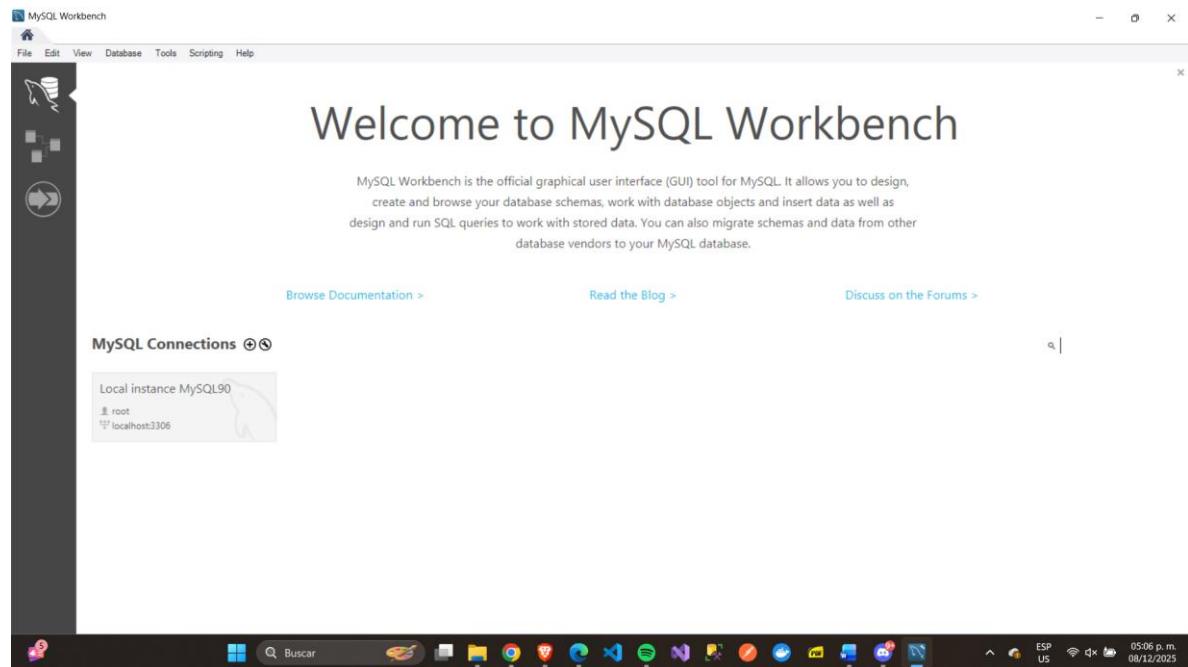
Indice

MYSQL Local	3
Creación de base de datos y tablas	4
Crear un usuario local para la base de datos.....	4
Backend	5
Creación de la carpeta con la nomenclatura.....	5
Instalacion de extensiones	5
Creacion del proyecto	6
Instalacion de paquetes	7
Modificación de localsettings	8
Compilación de proyecto	8
Creacion de las tablas stock, fotos_articulos y carrito_compra	9
Creacion de funciones en el backed.	11
Alta_articulo.....	11
consulta_articulos	12
compra_articulo	12
elimina_articulo_carrito_compra	13
elimina_carrito_compra	13
modifica_carrito_compra	14
Pruebas backend local	14
Creacion de la function app	18
Deploy de la app	19
Cuenta de almacenamiento.....	21
Recurso compartido de archivos.....	22
Creacion.....	22
Subida de archivos	22
Montaje de almacenamiento en la aplicación de funciones	23
Configuración de variables de entorno en aplicación de funciones	24
Root.....	24

Server	24
Creacion de base de datos PaaS	25
Pruebas de la app desde el celular	30
Pruebas de la function app desde Postman.....	46
Conclusiones	50

MYSQL Local

Ya se tenía una instalación previamente, por lo que ya no fue necesario volverlo a instalar



Creación de base de datos y tablas

The screenshot shows the MySQL Workbench interface. In the SQL Editor, the following SQL code is run:

```
1 • Create database servicio_web;
2 • use servicio_web;
3 • create table usuarios (
4     id_usuario integer auto_increment primary key,
5     password varchar(64) not null,
6     token varchar(20),
7     email varchar(100) not null,
8     nombre varchar(100) not null,
9     apellido_paterno varchar(100) not null,
10    apellido_materno varchar(100),
11    fecha_nacimiento datetime not null,
12    telefono bigint,
13    genero char(1) );
14
15 • create table fotos_usuarios (
16     id_foto integer auto_increment primary key,
```

The Output pane shows the results of the actions:

#	Time	Action	Message	Duration / Fetch
1	17:08:24	create database servicio_web	Error Code: 1007. Can't create database 'servicio_web'; database exists	0.015 sec
2	17:08:21	DROP DATABASE servicio_web	5 row(s) affected	0.360 sec
3	17:08:29	create database servicio_web	0 row(s) affected	0.031 sec
4	17:08:29	use servicio_web	0 row(s) affected	0.000 sec
5	17:08:29	create table usuarios (id_usuario integer auto_increment primary key, password varchar(64) not null, token varchar(20), email varchar(100) not null, nombre varchar(100) not null, apellido_paterno varchar(100) not null, apellido_materno varchar(100), fecha_nacimiento datetime not null, telefono bigint, genero char(1));	0 row(s) affected	0.015 sec
6	17:08:29	create table fotos_usuarios (id_foto integer auto_increment primary key, foto longblob, id_usuario integer)	0 row(s) affected	0.016 sec
7	17:08:29	alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.140 sec
8	17:08:29	create unique index usuarios_email on usuarios(email)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

Crear un usuario local para la base de datos

The screenshot shows the MySQL Workbench interface. In the SQL Editor, the following SQL code is run:

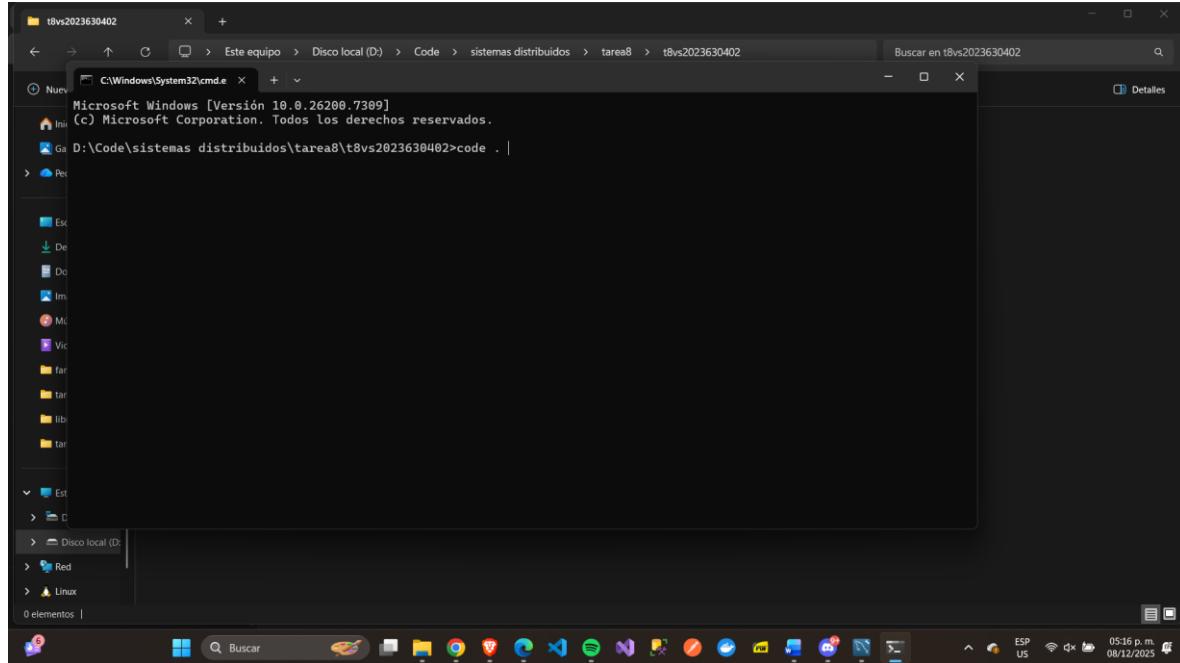
```
1 • CREATE USER 'hugo'@'localhost' IDENTIFIED BY '2023630402';
2 • GRANT ALL PRIVILEGES ON servicio_web.* TO 'hugo'@'localhost';
3 • FLUSH PRIVILEGES;
```

The Output pane shows the results of the actions:

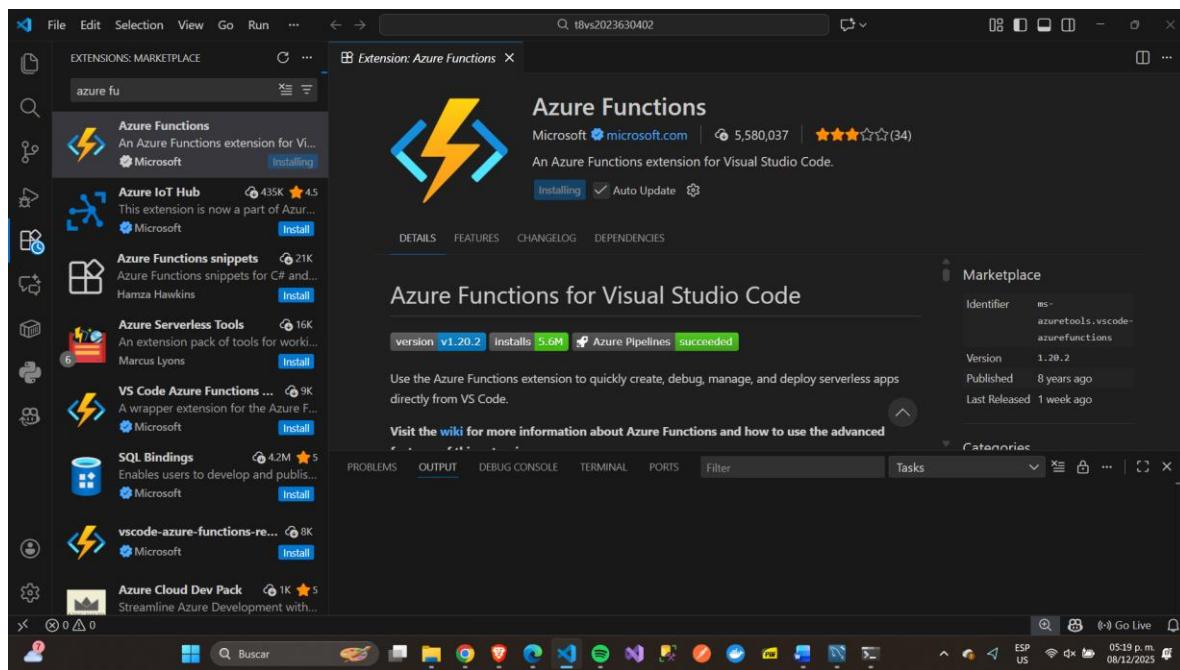
#	Time	Action	Message	Duration / Fetch
1	17:08:29	create table usuarios (id_usuario integer auto_increment primary key, password varchar(64) not null, token varchar(20), email varchar(100) not null, nombre varchar(100) not null, apellido_paterno varchar(100) not null, apellido_materno varchar(100), fecha_nacimiento datetime not null, telefono bigint, genero char(1));	0 row(s) affected	0.015 sec
2	17:08:29	create table fotos_usuarios (id_foto integer auto_increment primary key, foto longblob, id_usuario integer)	0 row(s) affected	0.016 sec
3	17:08:29	alter table fotos_usuarios add foreign key (id_usuario) references usuarios(id_usuario)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.140 sec
4	17:09:10	create unique index usuarios_email on usuarios(email)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec
5	17:10:16	CREATE USER hugo@localhost identified by 'aqui-va-la-contraseña-del-usuario-hugo'	Error Code: 1396. Operation CREATE USER failed for hugo@localhost'	0.015 sec
6	17:10:41	create user hugo@localhost identified by '2023630402'	Error Code: 1396. Operation CREATE USER failed for hugo@localhost'	0.000 sec
7	17:10:46	DROP USER hugo@localhost	0 row(s) affected	0.000 sec
8	17:10:38	CREATE USER hugo@localhost IDENTIFIED BY '2023630402'	0 row(s) affected	0.000 sec
9	17:10:38	GRANT ALL PRIVILEGES ON servicio_web.* TO hugo@localhost	0 row(s) affected	0.000 sec
10	17:10:38	FLUSH PRIVILEGES	0 row(s) affected	0.016 sec

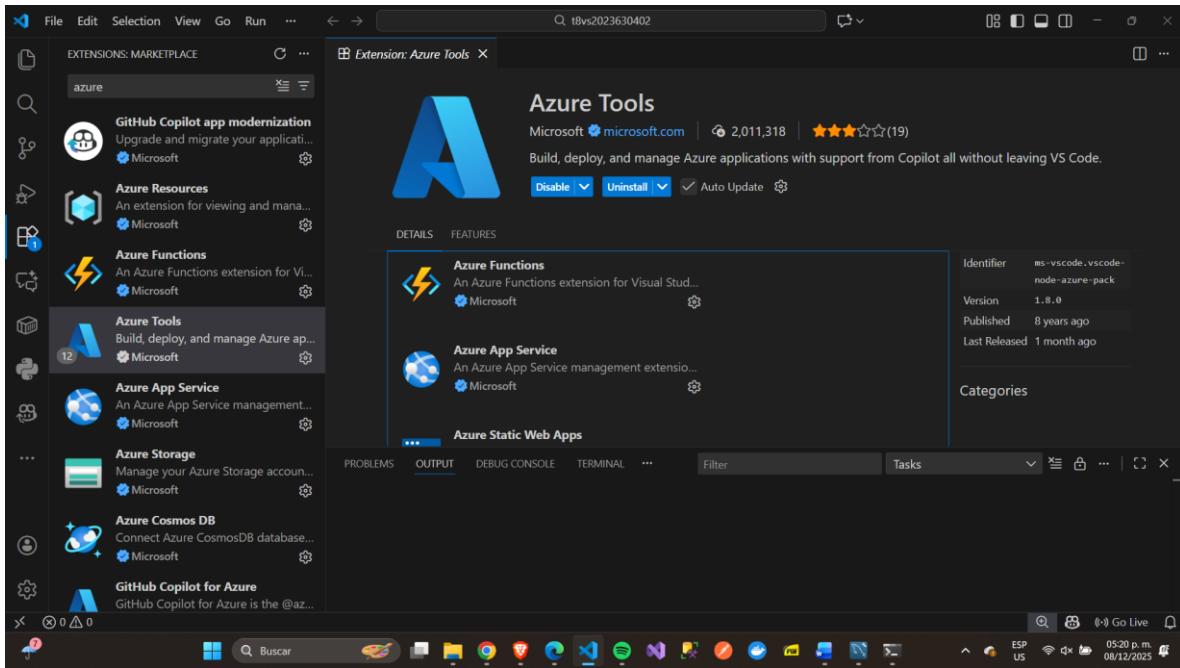
Backend

Creación de la carpeta con la nomenclatura

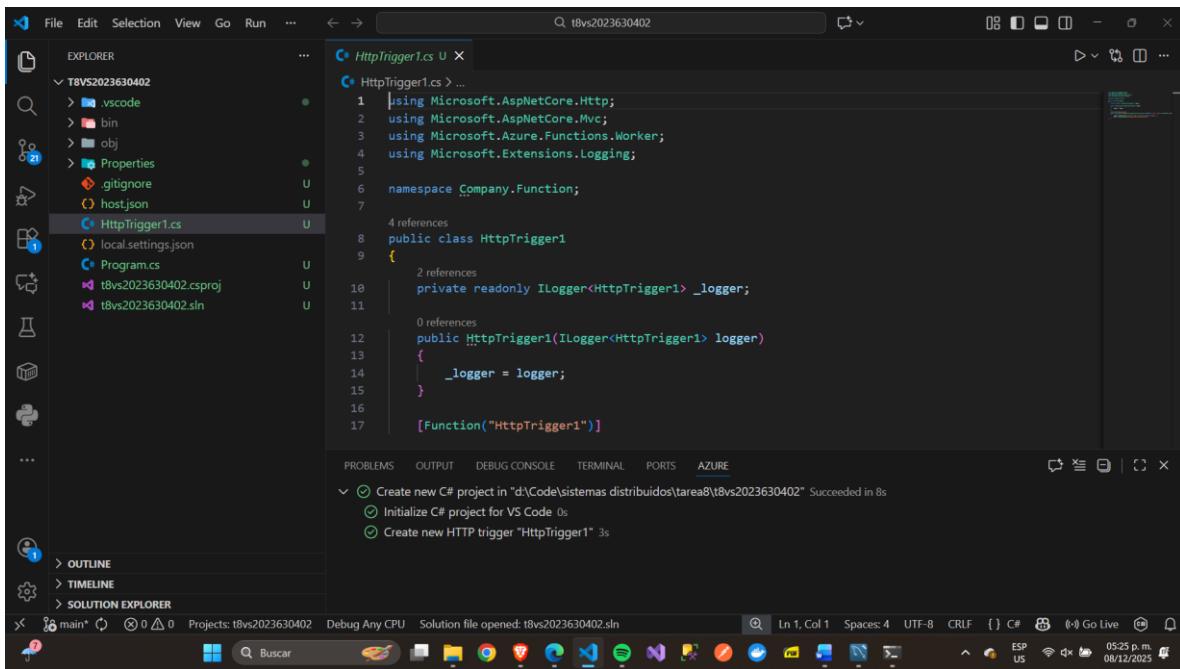


Instalacion de extensiones





Creacion del proyecto



Instalacion de paquetes

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "T8Vs2023630402".
- Code Editor:** Displays the "login.cs" file content.
- Terminal:** Shows the command "dotnet add package Newtonsoft.Json" being run in the terminal.
- Output:** Displays the results of the package addition, including validation of certificates and download information from nuget.org.

```
// (c) Carlos Pineda Guerrero. 2025
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Extensions.Logging;
using Newtonsoft.Json;
using MySql.Data.MySqlClient;
using System.Security.Cryptography;
using System.Text;

namespace servicio;

public class login
{
    public string mensaje;
}
```

PS D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402> dotnet add package Newtonsoft.Json
.NET para la firma de código.
info : La validación de la cadena de certificados X.509 utilizará el almacén de confianza predeterminado seleccionado por .NET para la marca de tiempo.
info : Agregando PackageReference para el paquete "Newtonsoft.Json" al proyecto "D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402\t8vs2023630402.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/newtonsoft.json/index.json

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "T8Vs2023630402".
- Code Editor:** Displays the "login.cs" file content.
- Terminal:** Shows the command "dotnet add package MySQL.Data" being run in the terminal.
- Output:** Displays the results of the package addition, including the addition of the MySQL package reference to the csproj file.

```
// (c) Carlos Pineda Guerrero. 2025
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Extensions.Logging;
using Newtonsoft.Json;
using MySql.Data.MySqlClient;
using System.Security.Cryptography;
using System.Text;

namespace servicio;

public class login
{
    public string mensaje;
}
```

PS D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402> dotnet add package MySQL.Data
distribuidos\tarea8\t8vs2023630402\t8vs2023630402.csproj".
info : Se agregó PackageReference para la versión "9.5.0" del paquete "MySQL.Data" al archivo "D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402\t8vs2023630402.csproj".
info : Escribiendo el archivo de recursos en el disco. Ruta de acceso: D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402\obj\project.assets.json
log : Se ha restaurado D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402\t8vs2023630402.csproj (en 3.82 s).
PS D:\Code\sistemas_distribuidos\tarea8\t8vs2023630402>

Modificación de localsettings

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "T8vs2023630402".
- Editor:** Displays the content of the "localsettings.json" file. The code is as follows:

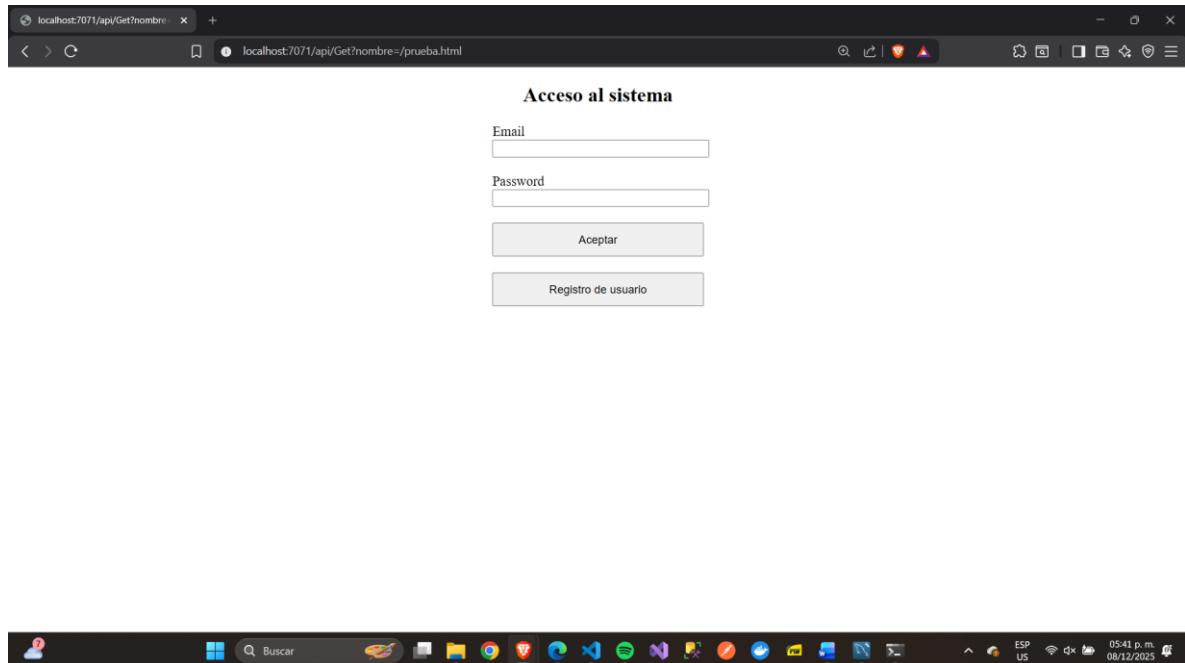
```
1  {
2      "IsEncrypted": false,
3      "Values": [
4          "AzureWebJobsStorage": "UseDevelopmentStorage=true",
5          "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
6          "Server": "localhost",
7          "UserID": "hugo",
8          "Password": "2023630402",
9          "Database": "servicio_web",
10         "ROOT": "D:\\Code\\sistemas distribuidos\\tarea8\\frontend-t8-2023630402"
11     ]
12 }
13 }
```

- Terminal:** Shows the command "PS D:\\Code\\sistemas distribuidos\\tarea8\\t8vs2023630402" in the terminal window.
- Bottom Status Bar:** Includes icons for PowerShell, file navigation, and system status like battery level and date/time.

Compilación de proyecto

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "T8vs2023630402".
- Run and Debug View:** The "VARIABLES" tab is selected, showing the content of the "localsettings.json" file.
- Editor:** Displays the same "localsettings.json" code as in the previous screenshot.
- Terminal:** Shows the output of a command, likely "func host start", with logs related to Azure Functions execution.
- Bottom Status Bar:** Includes icons for PowerShell, file navigation, and system status like battery level and date/time.



Creacion de las tablas stock, fotos_articulos y carrito_compra

Script utilizado:

```
USE servicio_web;
```

```
-- 1) Tabla de artículos en stock
```

```
CREATE TABLE stock (
    id_articulo INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    existencia INT NOT NULL,
    id_usuario INT NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
```

);

-- 2) Tabla de fotos de artículos

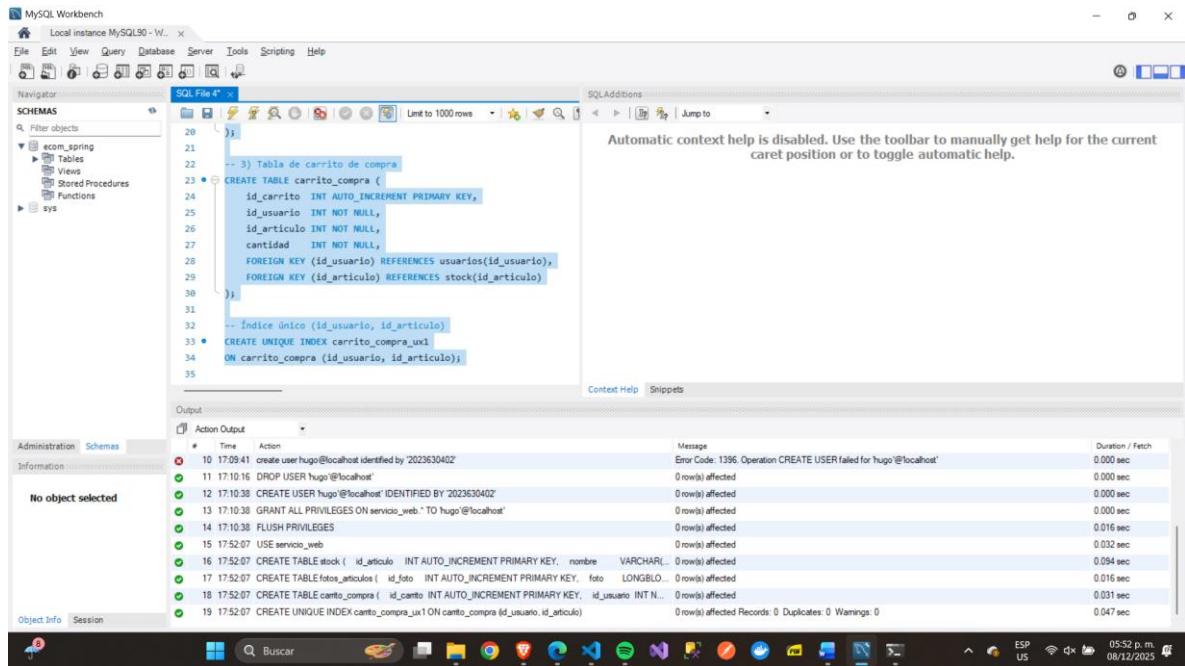
```
CREATE TABLE fotos_articulos (
    id_foto INT AUTO_INCREMENT PRIMARY KEY,
    foto LONGBLOB NOT NULL,
    id_articulo INT NOT NULL,
    FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
);
```

-- 3) Tabla de carrito de compra

```
CREATE TABLE carrito_compra (
    id_carrito INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    id_articulo INT NOT NULL,
    cantidad INT NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
    FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
);
```

-- Índice único (id_usuario, id_articulo)

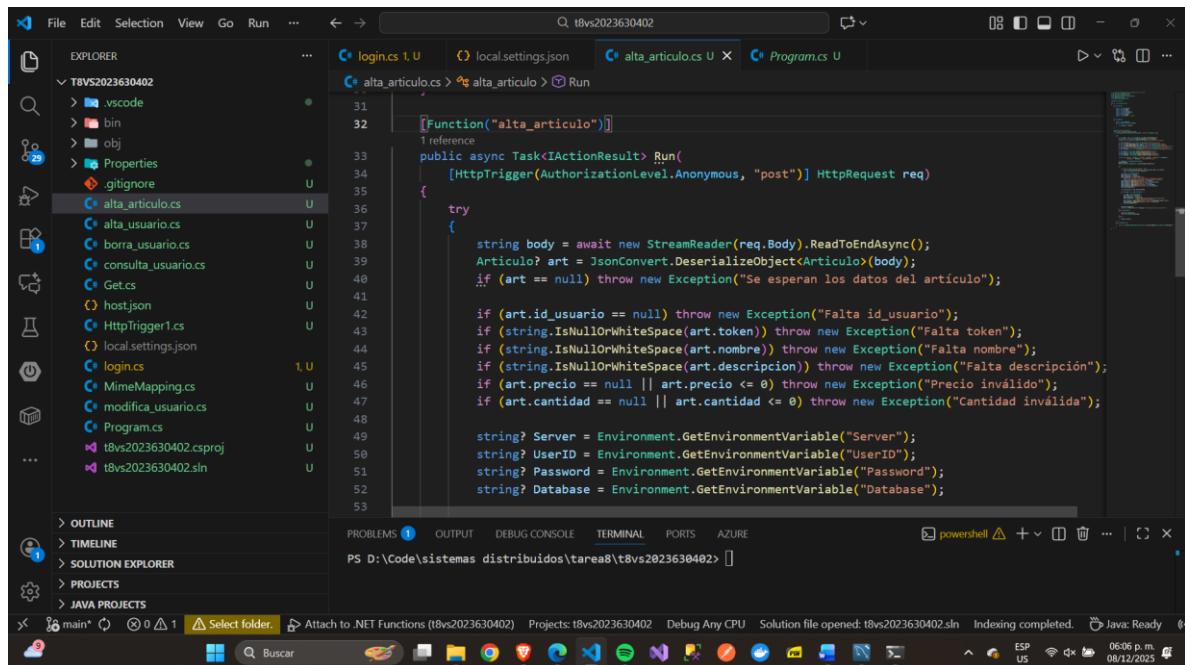
```
CREATE UNIQUE INDEX carrito_compra_ux1
ON carrito_compra (id_usuario, id_articulo);
```



Creacion de funciones en el backed.

NOTA: Las funciones completas están adjuntas en el repositorio y en la entrega de la tarea.

Alta_articulo



consulta_articulos

The screenshot shows the Visual Studio Code interface with the 'consulta_articulos.cs' file open in the editor. The code is a .NET function for querying articles. It includes validation logic for parameters like 'palabra', 'id_usuario', and 'token'. It also checks environment variables for 'Server', 'UserID', and 'Password'. The code uses the 'HttpRequest' class to read the body and the 'JsonConvert' class to deserialize objects.

```
[Function("consulta_articulos")]
public async Task<IActionResult> Run(
    [HttpTrigger(AuthorizationLevel.Anonymous, "get")] HttpRequest req)
{
    try
    {
        string? palabra = req.Query["palabra"];
        string? id_usuario_str = req.Query["id_usuario"];
        string? token = req.Query["token"];

        if (string.IsNullOrEmpty(palabra))
            throw new Exception("Debe indicar una palabra clave");
        if (string.IsNullOrEmpty(id_usuario_str))
            throw new Exception("Falta id_usuario");
        if (string.IsNullOrEmpty(token))
            throw new Exception("Falta token");

        int id_usuario = int.Parse(id_usuario_str);

        string? Server = Environment.GetEnvironmentVariable("Server");
        string? UserID = Environment.GetEnvironmentVariable("UserID");
        string? Password = Environment.GetEnvironmentVariable("Password");
    }
}
```

compra_articulo

The screenshot shows the Visual Studio Code interface with the 'compra_articulo.cs' file open in the editor. This is another .NET function, specifically for purchasing an item. It reads data from the request body, deserializes it into a 'Peticion' object, and then performs validation on the 'id_usuario', 'token', and 'id_articulo' fields. It also handles the connection string 'cs' by concatenating environment variables for 'Server', 'UserID', 'Password', and 'Database'. The code uses 'StreamReader' to read the request body and 'JsonConvert' for deserialization.

```
[Function("compra_articulo")]
public async Task<IActionResult> Run(
    [HttpTrigger(AuthorizationLevel.Anonymous, "post")] HttpRequest req)
{
    try
    {
        string body = await new StreamReader(req.Body).ReadToEndAsync();
        Peticion? p = JsonConvert.DeserializeObject<Peticion>(body);
        if (p == null) throw new Exception("Se esperan los datos de la compra");

        if (p.id_usuario == null) throw new Exception("Falta id_usuario");
        if (string.IsNullOrEmpty(p.token)) throw new Exception("Falta token");
        if (p.id_articulo == null) throw new Exception("Falta id_articulo");
        if (p.cantidad == null || p.cantidad <= 0) throw new Exception("Cantidad inválida");

        string? Server = Environment.GetEnvironmentVariable("Server");
        string? UserID = Environment.GetEnvironmentVariable("UserID");
        string? Password = Environment.GetEnvironmentVariable("Password");
        string? Database = Environment.GetEnvironmentVariable("Database");

        string cs = "Server=" + Server + "&UserID=" + UserID + "&Password=" + Password +
                   "&Database=" + Database + "&SslMode=Preferred";
    }
}
```

elimina_articulo_carrito_compra

The screenshot shows the Visual Studio Code interface with the file 'elimina_articulo_carrito_compra.cs' open in the editor. The code implements a function to delete an item from a shopping cart. It checks for required parameters like 'id_usuario' and 'token', converts them to integers, and then performs the deletion. Environment variables are used to get database connection details.

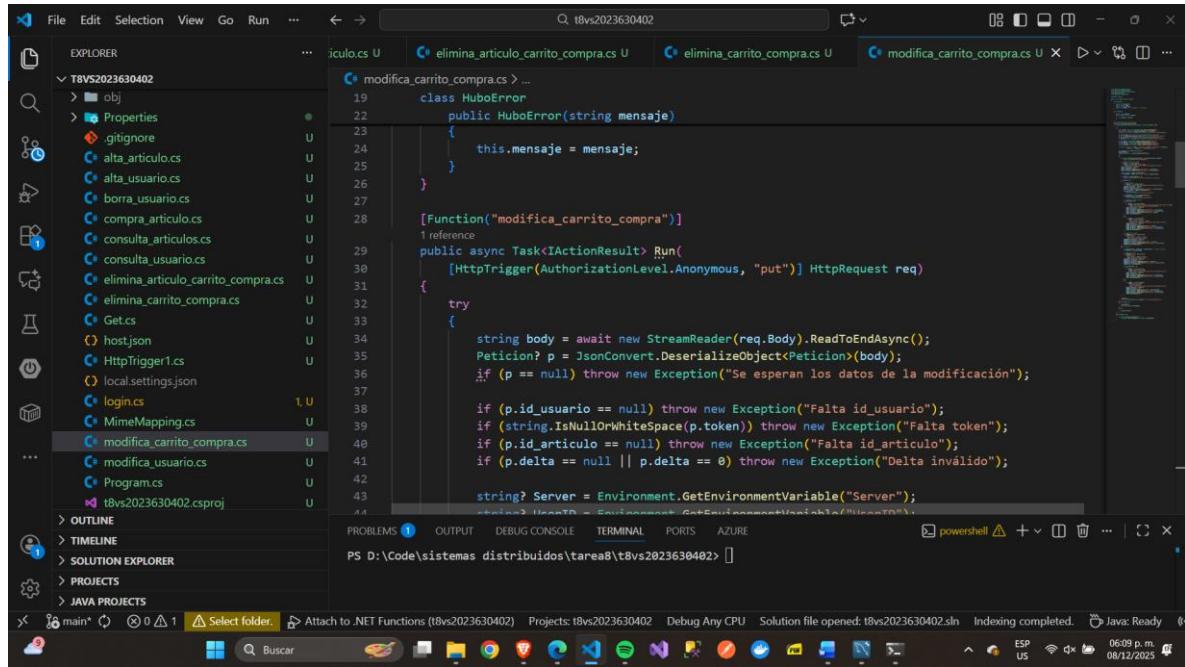
```
11     class HuboError
12         public HuboError(string message)
13     }
14
15     [Function("elimina_articulo_carrito_compra")]
16     1 reference
17     public async Task<IActionResult> Run(
18         [HttpTrigger(AuthorizationLevel.Anonymous, "delete")] HttpRequest req
19     )
20     {
21         try
22         {
23             string? id_usuario_str = req.Query["id_usuario"];
24             string? token = req.Query["token"];
25             string? id_articulo_str = req.Query["id_articulo"];
26
27             if (string.IsNullOrWhiteSpace(id_usuario_str)) throw new Exception("Falta id_usuario");
28             if (string.IsNullOrWhiteSpace(token)) throw new Exception("Falta token");
29             if (string.IsNullOrWhiteSpace(id_articulo_str)) throw new Exception("Falta id_articulo");
30
31             int id_usuario = int.Parse(id_usuario_str);
32             int id_articulo = int.Parse(id_articulo_str);
33
34             string? Server = Environment.GetEnvironmentVariable("Server");
35             string? UserID = Environment.GetEnvironmentVariable("UserID");
36
37             string cs = "Server=" + Server + "&UserID=" + UserID + "&Password=" + Password +
38             "&Database=" + Database + "&SslMode=Preferred";
39
40         }
41     }
42 }
```

elimina_carrito_compra

The screenshot shows the Visual Studio Code interface with the file 'elimina_carrito_compra.cs' open in the editor. This version of the function also checks for 'id_usuario' and 'token' parameters. It then uses environment variables to build a connection string ('cs') for a database, combining 'Server', 'UserID', 'Password', 'Database', and 'SslMode'. The code is identical to the one above it.

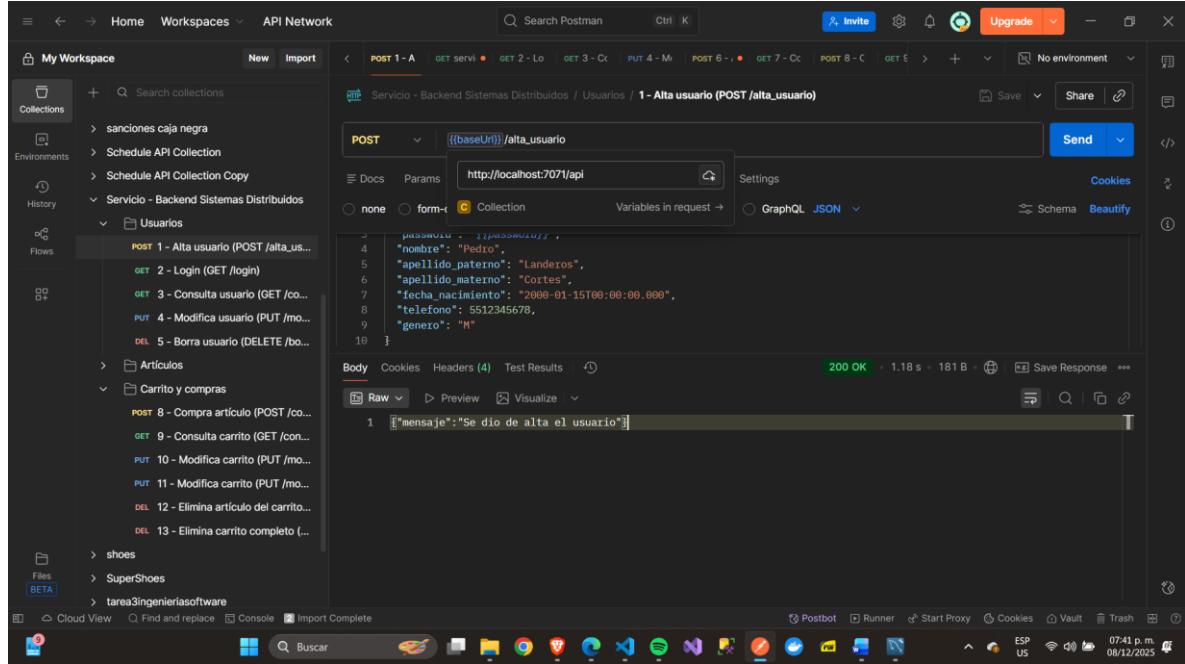
```
18     }
19
20     [Function("elimina_carrito_compra")]
21     1 reference
22     public async Task<IActionResult> Run(
23         [HttpTrigger(AuthorizationLevel.Anonymous, "delete")] HttpRequest req
24     )
25     {
26         try
27         {
28             string? id_usuario_str = req.Query["id_usuario"];
29             string? token = req.Query["token"];
30
31             if (string.IsNullOrWhiteSpace(id_usuario_str)) throw new Exception("Falta id_usuario");
32             if (string.IsNullOrWhiteSpace(token)) throw new Exception("Falta token");
33
34             int id_usuario = int.Parse(id_usuario_str);
35
36             string? Server = Environment.GetEnvironmentVariable("Server");
37             string? UserID = Environment.GetEnvironmentVariable("UserID");
38             string? Password = Environment.GetEnvironmentVariable("Password");
39             string? Database = Environment.GetEnvironmentVariable("Database");
40
41             string cs = "Server=" + Server + "&UserID=" + UserID + "&Password=" + Password +
42             "&Database=" + Database + "&SslMode=Preferred";
43
44         }
45     }
46 }
```

modifica_carrito_compra



```
19     class HuboError
20     {
21         public HuboError(string mensaje)
22         {
23             this.mensaje = mensaje;
24         }
25     }
26
27     [Function("modifica_carrito_compra")]
28     1 reference
29     public async Task<ActionResult> Run(
30         [HttpTrigger(AuthorizationLevel.Anonymous, "put")] HttpRequest req
31     )
32     {
33         try
34         {
35             string body = await new StreamReader(req.Body).ReadToEndAsync();
36             Peticion? p = JsonConvert.DeserializeObject<Peticion>(body);
37             if (p == null) throw new Exception("Se esperan los datos de la modificación");
38
39             if (p.id_usuario == null) throw new Exception("Falta id_usuario");
40             if (string.IsNullOrWhiteSpace(p.token)) throw new Exception("Falta token");
41             if (p.id_articulo == null) throw new Exception("Falta id_articulo");
42             if (p.delta == null || p.delta == 0) throw new Exception("Delta inválido");
43
44             string? Server = Environment.GetEnvironmentVariable("Server");
45             string? UserToken = Environment.GetEnvironmentVariable("UserToken");
46             string? token = Environment.GetEnvironmentVariable("token");
47             string? id_usuario = Environment.GetEnvironmentVariable("id_usuario");
48             string? id_articulo = Environment.GetEnvironmentVariable("id_articulo");
49             string? delta = Environment.GetEnvironmentVariable("delta");
50             string? mensaje = Environment.GetEnvironmentVariable("mensaje");
51
52             string resultado = await ModificarCarrito(modificarCarrito, id_usuario, id_articulo, delta, token, mensaje);
53             return new StringContent(resultado);
54         }
55         catch (Exception ex)
56         {
57             return new StringContent(ex.Message);
58         }
59     }
60 }
```

Pruebas backend local



POST 1 - Alta usuario (POST /alta_usuario)

```
1
2
3
4
5
6
7
8
9
10
password : "1234567890",
"nombre": "Pedro",
"apellido_paterno": "Landeros",
"apellido_materno": "Cortes",
"fecha_nacimiento": "1990-01-15T00:00:00.000",
"telefono": 5512345678,
"genero": "M"
```

Body Cookies Headers (4) Test Results

Raw Preview Visualize

1 `["mensaje":"Se dio de alta el usuario"]`

The screenshot shows the Postman interface with the 'My Workspace' collection selected. The left sidebar lists various environments, flows, and collections. The main workspace displays a GET request for '2 - Login (GET /login)'. The URL is `((baseUrl)) /login?email= {{email}} &password= {{password}}`. The 'Params' tab is active, showing query parameters: 'email' with value `((email))` and 'password' with value `((password))`. The 'Body' tab shows a raw JSON response: `{"token": "J6OyGMYjNv0xEYmaU352"}`. The status bar at the bottom indicates a 200 OK response with 108 ms duration and 174 B size.

This screenshot shows the same Postman session after a token has been generated. The '3 - Consulta usuario (GET /consulta_usuario)' request is now active. The URL is `((baseUrl)) /consulta_usuario?email= {{email}} &token= {{token}}`. The 'Params' tab shows 'email' with value `((email))` and 'token' with value `((token))`, which is expanded to show the value `J6OyGMYjNv0xEYmaU352`. The 'Body' tab shows a raw JSON response: `{"id_usuario": 2, "email": "a@gmail.com", "nombre": "Pedro", "apellido_paterno": "Landeros", "fecha_nacimiento": "2000-01-15T00:00:00", "telefono": "5512345678", "genero": "M", "fo...`. A tooltip message 'Screenshot copied to clipboard' is visible in the bottom right corner.

My Workspace

POST 6 - Alta artículo (POST /alta_articulo)

```
1 {  
2   "id_usuario": "{{id_usuario}}",  
3   "token": "{{token}}"  
4   "nombre": "Laptop  
5   "descripcion": "  
6   "precio": 15000,  
7   "cantidad": 10  
8 }  
9 
```

Body Cookies Headers (4) Test Results

200 OK + 90 ms · 183 B Save Response

Raw Preview Visualize

1 {"mensaje": "Se dio de alta el artículo"}

POST 1 - Alta usuario (POST /alta_usuario)

GET 2 - Login (GET /login)

GET 3 - Consulta usuario (GET /co...)

PUT 4 - Modifica usuario (PUT /mo...)

DEL 5 - Borra usuario (DELETE /bo...)

POST 6 - Alta artículo (POST /alta_a...)

GET 7 - Consulta artículos (GET /c...)

POST 8 - Compra artículo (POST /co...)

GET 9 - Consulta carrito (GET /con...)

PUT 10 - Modifica carrito (PUT /mo...)

PUT 11 - Modifica carrito (PUT /mo...)

DEL 12 - Elimina artículo del carri...)

DEL 13 - Elimina carrito completo ...)

shoes

Cloud View Find and replace Console Import Complete

Postbot Runner Start Proxy Cookies Vault Trash

ESP US 07:43 p. m. 08/12/2025

My Workspace

GET 7 - Consulta artículos (GET /consulta_articulos)

Key	Value	Description
palabra	laptop	
id_usuario	{{id_usuario}}	
token	{{token}}	

Body Cookies Headers (4) Test Results

200 OK + 96 ms · 274 B Save Response

Raw Preview Visualize

1 [{"id_articulo":1,"nombre": "Laptop Gamer 2","descripcion": "Laptop potente para juegos","precio":15000.50,"cantidad":10, "foto":null}]

POST 1 - Alta usuario (POST /alta_usuario)

GET 2 - Login (GET /login)

GET 3 - Consulta usuario (GET /co...)

PUT 4 - Modifica usuario (PUT /mo...)

DEL 5 - Borra usuario (DELETE /bo...)

POST 6 - Alta artículo (POST /alta_a...)

GET 7 - Consulta artículos (GET /c...)

POST 8 - Compra artículo (POST /co...)

GET 9 - Consulta carrito (GET /con...)

PUT 10 - Modifica carrito (PUT /mo...)

PUT 11 - Modifica carrito (PUT /mo...)

DEL 12 - Elimina artículo del carri...)

DEL 13 - Elimina carrito completo ...)

shoes

Cloud View Find and replace Console Import Complete

Postbot Runner Start Proxy Cookies Vault Trash

ESP US 07:43 p. m. 08/12/2025

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** {{baseUrl}} /compra_articulo
- Method:** POST
- Headers:** Authorization, Content-Type (x-www-form-urlencoded)
- Body:** Raw JSON

```
1 {
2   "id_usuario": "{{id_usuario}}",
3   "token": "{{token}}",
4   "id_articulo": "{{id_articulo}}",
5   "cantidad": 2
6 }
```

- Response:** 200 OK, 285 ms, 185 B. Response body: {"mensaje": "Articulo agregado al carrito"}

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** {{baseUrl}} /consulta_carrito_compra
- Method:** GET
- Headers:** Authorization, Content-Type (application/json)
- Params:** id_usuario, token
- Response:** 200 OK, 56 ms, 273 B. Response body: [{"id_articulo": 1, "nombre": "Laptop Gamer 2", "descripcion": "Laptop potente para juegos", "precio": 15000.5, "cantidad": 2, "foto": null}]

The screenshot shows the Postman interface with a collection named 'Servicio - Backend Sistemas Distribuidos'. A specific DELETE request is selected, targeting the endpoint `{baseUri}/elmina_articulo_carrito_compra?id_usuario={{id_usuario}}&id_articulo={{id_articulo}}&token={{token}}`. The request includes three query parameters: `id_usuario`, `id_articulo`, and `token`. The response is successful (200 OK) with the message `{"mensaje": "Articulo eliminado del carrito"}`.

Creacion de la function app

Se hizo la creación desde vsc, con .net 8, el nombre indicado, región us east 2.

The screenshot shows the Visual Studio Code interface with the Azure Functions extension installed. The sidebar shows resources like Azure Functions, Logic App, Managed Identities, etc. The main area shows four function files: `elimina_articulo_carrito_compra.cs`, `elimina_carrito_compra.cs`, `modifica_carrito_compra.cs`, and `consulta_carrito_compra.cs`. The terminal window shows deployment logs for a function app named `FLEX-t8ap2023630402-4f13`, indicating it was successfully deployed.

Aplicación de funciones

Nombre	Estado	Ubicación	Plan de tarifa	Plan de App Service	Suscripción	Tipo de aplicación
t8ap2023630402	En ejecución	East US 2	Consumo flexible	FLEX-t8ap2023630402-0419	Azure for Students	Aplicación de funciones

Deploy de la app

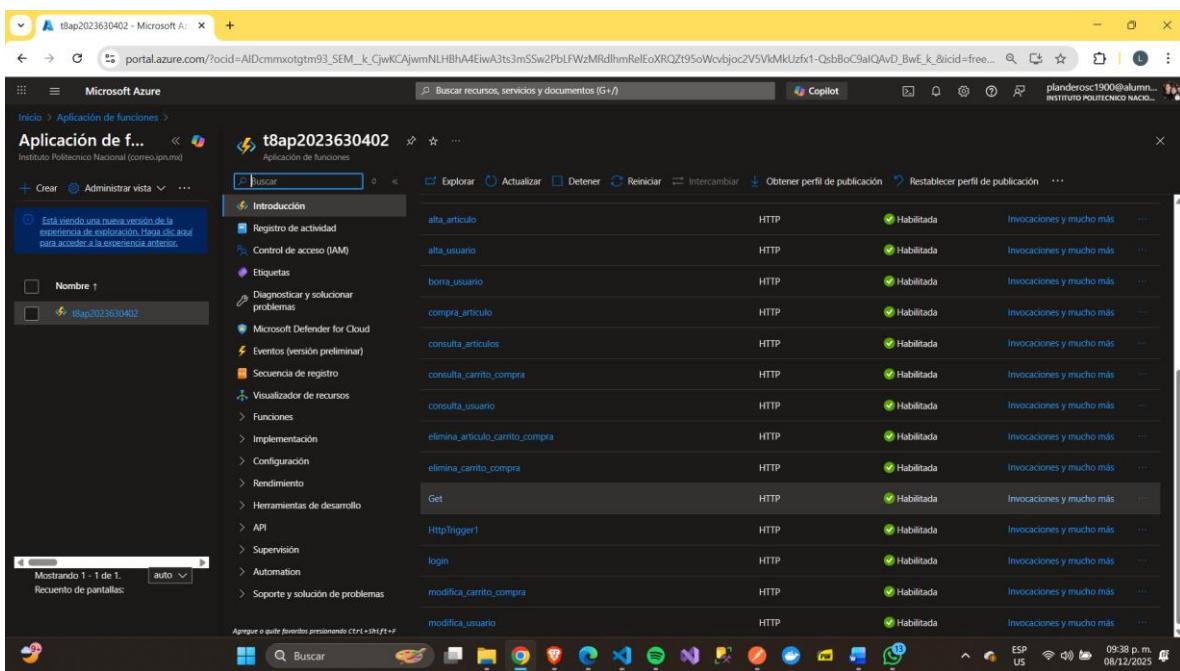
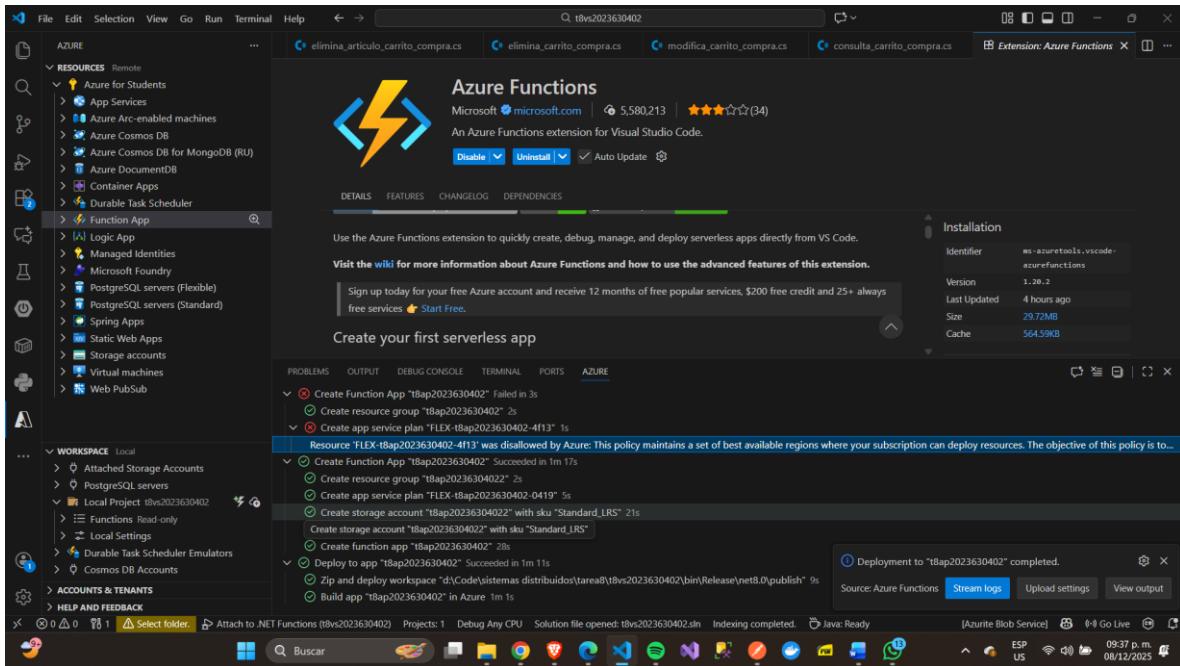
Azure Functions

Are you sure you want to deploy to "t8ap2023630402"? This will overwrite any previous deployment and cannot be undone.

Deployment

Installation

Identifier	ms-azurertools.vscode-azurefunctions
Version	1.26.2
Last Updated	4 hours ago
Size	29.72MB
Cache	564.59KB



Cuenta de almacenamiento

The screenshot shows the 'Crear una cuenta de almacenamiento' (Create storage account) wizard in the Microsoft Azure portal. The 'Básicos' (Basic) tab is selected. The account details are as follows:

Configuración	Valor
Suscripción	Azure for Students
Grupo de recursos	t8ap2023630402
Ubicación	East US 2
Nombre de la cuenta de almacenamiento	t8af2023630402
Tipo de almacenamiento preferido	Estándar
Rendimiento	Estándar
Replicación	Almacenamiento con redundancia local (LRS)

Below the basic settings, there are sections for 'Avanzado' (Advanced) and 'Seguridad' (Security). The security section includes options for SFTP, hierarchical namespace, and replication. At the bottom, there are 'Anterior' (Previous), 'Siguiente' (Next), and 'Crear' (Create) buttons.

The screenshot shows the 't8af2023630402_1765252138054 | Información general' (General information) page. The 'Implementación' (Implementation) section indicates that the implementation was successful. Key details include:

- Nombre de implementación: t8af2023630402_1765252138054
- Suscripción: Azure for Students
- Grupo de recursos: t8ap2023630402
- Hora de inicio: 8/12/2025, 21:50:04
- Id. de correlación: 662ff03d-f6ea-4292-95bc-ace2de54a70f

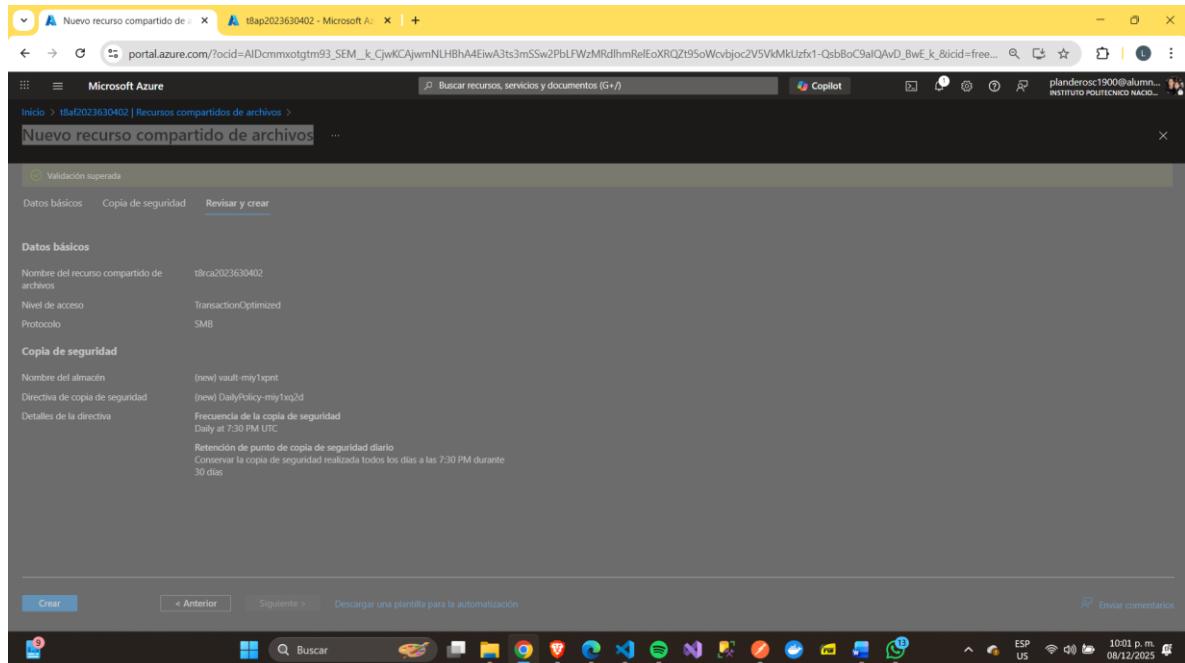
On the right side, there are several promotional cards:

- Implementación correcta**: La implementación 't8af2023630402_1765252138054' se realizó correctamente en el grupo de recursos 't8ap2023630402'.
- Cost Management**: Obtenga una notificación para permanecer dentro del presupuesto y evitar cargos inesperados en su factura.
- Microsoft Defender for Cloud**: Proteja sus aplicaciones e infraestructura.
- Tutoriales gratuitos de Microsoft**: Comience a aprender hoy >
- Trabajar con un experto**: Los expertos de Azure son asociados proveedores de servicios que pueden ayudar a administrar sus recursos en Azure y ser la primera línea de soporte técnico.

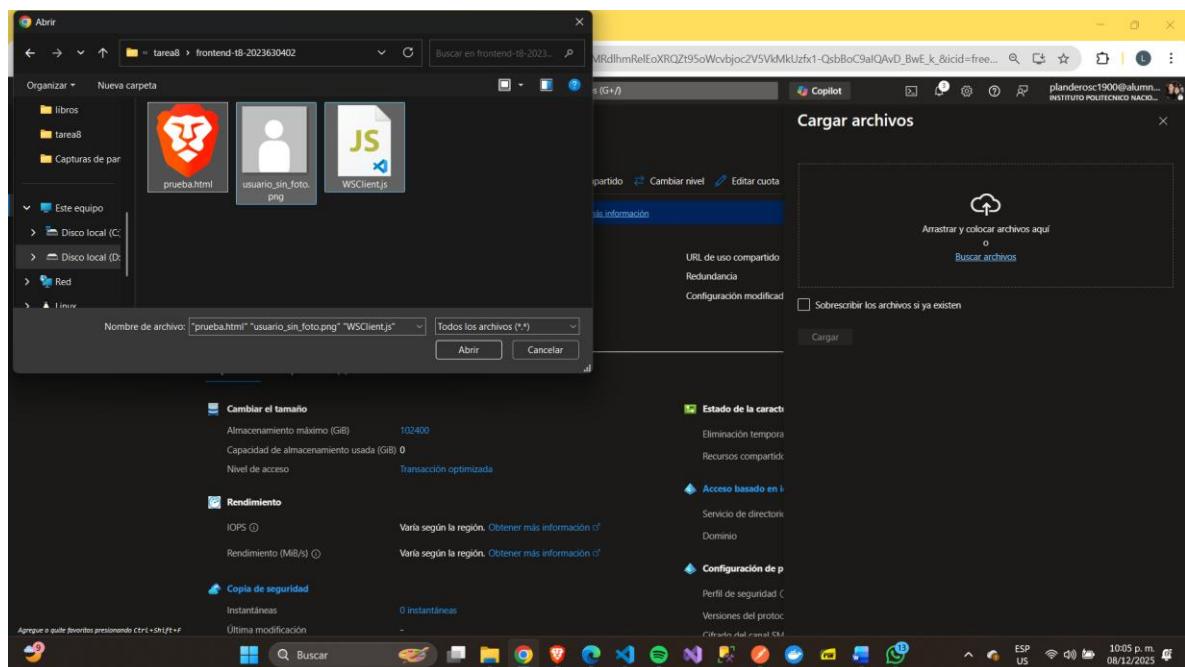
At the bottom, there are navigation links for 'Ir al recurso' (Go to resource) and 'Añadir al panel' (Add to dashboard).

Recurso compartido de archivos

Creacion



Subida de archivos



The screenshot shows the Microsoft Azure Storage Shared Access Shared File System (SAS) creation interface. On the left, there's a sidebar with navigation links like 'Información general', 'Diagnosticar y solucionar problemas', 'Control de acceso (BAM)', 'Examinar', and 'Operaciones'. The main area displays storage details: Cuenta de almacenamiento: t8ap2023630402, Grupo de recursos: t8ap2023630402, Ubicación: East US 2, Suscripción: Azure for Students, and Id. de suscripción: 81d3d13e-e29d-4bd2-be90-9103abf7700. A central panel titled 'Habilitar Backup para el recurso compartido "t8rca2023630402" para proteger los datos.' has a 'Configuración modificada' button. To the right, a 'Cargar archivos' section shows three files uploaded: prueba.html (23.19 KB / 23.19 KB), usuario_sin_foto.png (1.62 KB / 1.62 KB), and WSClient.js (3.42 KB / 3.42 KB). Below these are sections for 'Cargas actuales' and 'Descargar'. At the bottom, there's a toolbar with icons for 'Cargar', 'Estado de la carpeta', 'Eliminación temporal', 'Recursos compartidos', 'Dominio', and 'Configuración de p'. The taskbar at the bottom shows various application icons.

Montaje de almacenamiento en la aplicación de funciones

The screenshot shows the Microsoft Azure Function App configuration interface. The left sidebar includes 'Eventos (versión preliminar)', 'Secuencia de registro', 'Visualizador de recursos', 'Funciones' (selected), 'Claves de la aplicación', 'Servidores proxy', 'Implementación' (selected), 'Centro de implementación', 'Configuración' (selected), 'Variables de entorno', 'Configuración (versión preliminar)' (selected), 'Durable Functions', 'Configuración de implementación', 'Autenticación', 'Azure OpenAI', 'Base de datos de vectores', and 'Identidad'. The main area shows 'Asignaciones de ruta de acceso' (selected tab) for 't8ap2023630402'. It displays a table with one entry: Nombre: t8pm2023630402, Ruta de acceso de montaje: /t8pm2023630402, Tipo: AzureFiles, Nombre de cuenta: t8ap2023630402, and Nombre del recurso: t8ap2023630402. To the right, the 'Agregar montaje de almacenamiento' dialog is open, showing fields for Nombre: t8pm2023630402, Opciones de configuración: Básico (selected), Nombre de cuenta: t8ap2023630402, and Tipo de almacenamiento: Azure Files (selected). Other options include Azure Blob (solo lectura), Protocolo: SMB (selected), and Contenedor de almacenamiento: t8rca2023630402. The status bar at the bottom indicates the date and time: 10:45 p.m. 08/12/2025.

Configuración de variables de entorno en aplicación de funciones

Root

The screenshot shows the Azure portal interface for managing environment variables of a function app. The left sidebar navigation includes 'Eventos (versión preliminar)', 'Seguridad de registro', 'Visualizador de recursos', 'Funciones' (selected), 'Claves de la aplicación', 'Servidores proxy', 'Implementación' (selected), 'Centro de implementación', 'Configuración' (selected), and 'Variables de entorno' (selected). The main content area displays the 'Configuración de aplicación' tab under 'Variables de entorno'. A table lists environment variables:

Nombre	Valor	Configuración de ranura de...	Origen	Eliminar
APPLICATIONINSIGHTS_CONNECTION_STRING	Mostrar valor		App Service	
AzureWebJobsStorage	Mostrar valor		App Service	
DEPLOYMENT_STORAGE_CONNECTION_STRING	Mostrar valor		App Service	
ROOT	/t8pm2023630402		App Service	

Server

The screenshot shows the Azure portal interface for managing environment variables of a function app named t8ap2023630402. The left sidebar navigation includes 'Introducción', 'Registro de actividad', 'Control de acceso (IAM)', 'Etiquetas', 'Diagnosticar y solucionar problemas', 'Microsoft Defender for Cloud', 'Eventos (versión preliminar)', 'Seguridad de registro', 'Visualizador de recursos', 'Funciones' (selected), 'Claves de la aplicación', 'Servidores proxy', 'Implementación' (selected), 'Centro de implementación', 'Configuración' (selected), and 'Variables de entorno' (selected). The main content area displays the 'Configuración de aplicación' tab under 'Variables de entorno'. A table lists environment variables, including a new 'Server' entry:

Nombre	Valor	Configuración de ranura de...	Origen	Eliminar
APPLICATIONINSIGHTS_CONNECTION_STRING	InstrumentationKey=4e8a2ad1-e626-4874-bd4-2d5cf060638c;IngestionEndpoint=https://		App Service	
AzureWebJobsStorage	DefaultEndpointProtocol=https;AccountName=t8ap20236304022;AccountKey=gEZw7+M		App Service	
Database	servicio_web		App Service	
DEPLOYMENT_STORAGE_CONNECTION_STRING	DefaultEndpointsProtocol=https;AccountName=t8ap20236304022;AccountKey=gEZw7+M		App Service	
Password	Lancort20+12		App Service	
ROOT	/t8pm2023630402		App Service	
Server	t8mysql2023630402p.mysql.database.azure.com		App Service	
Userid	hugo		App Service	

Creacion de base de datos PaaS

Se intento hacer una primera creación y no se pudo

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with 'Información general' (General information) and a list of resources: Entradas, Salidas, and Plantilla. The main content area displays the 'MySqlFlexibleServer_e277092cbfd3474cae5b2c1fe31f3bf2 | Información general' page. It shows the implementation status is 'en curso' (in progress). The 'Detalles de implementación' (Implementation details) section lists two resources: 'firewallRules-d3bf' (Implementation, Created) and 't8mysql2023630402' (Servidor flexible de Azure Database, Created). Below this, a 'Pasos siguientes' (Next steps) section provides links to tutorials for MySQL, Azure Database Migration Service, and Azure Database for MySQL. To the right, a 'Notificaciones' (Notifications) panel is open, showing two error messages:

- Error en la validación de la implementación**: Error en la validación de la implementación. Detalles adicionales de la API subyacente que pueden ser útiles: El recurso "t8mysql2023630402" ya existe en la ubicación "eastus2" del grupo de recursos "t8mysql2023630402". No se puede crear un recurso con el mismo nombre en la ubicación "microsoftcentral". Seleccione un nuevo nombre para el recurso.
- Error en la validación de la implementación**: Error en la validación de la implementación. Detalles adicionales de la API subyacente que pueden ser útiles: El recurso "t8mysql2023630402" ya existe en la ubicación "eastus2" del grupo de recursos "t8mysql2023630402". No se puede crear un recurso con el mismo nombre en la ubicación "microsoftcentral". Seleccione un nuevo nombre para el recurso.

Y cuando se intento volver a usar el mismo nombre marco un error incluso cuando la base de datos ya se había eliminado, por lo que se agrego una p al final del nuevo nombre.

The screenshot shows the Microsoft Azure portal interface. The main title is "MySQLFlexibleServer_e277092cbfd3474cae5b2c1fe31f3bf2 | Información general". A green checkmark indicates "Se completó la implementación". Key details include:

- Nombre de implementación : MySQLFlexibleServer_e277092cbfd3474cae5b2c1fe31f3bf2
- Suscripción : Azure for Students
- Grupo de recursos : t8ap2023630402
- Hora de inicio : 8/12/2025, 23:10:58
- Id. de correlación : 1actbad6-213a-41ca-b428-c300b6f97868

Below this, there's a section titled "Detalles de implementación" and a "Pasos siguientes" section with several recommended steps. At the bottom are "Ir al recurso" and "Migrar" buttons.

The screenshot shows the Microsoft Azure portal interface for an Azure Database for MySQL flexible server named "t8mysql2023630402p". The main title is "t8mysql2023630402p | Información general". Key details include:

- Suscripción (mover) : Azure for Students
- Id. de suscripción : 81d3d13e-e29d-4bd2-be90-9103abfb7700
- Grupo de recursos (mover) : t8ap2023630402
- Punto de conexión : t8mysql2023630402p.mysql.database.azure.com
- Estado : Ready
- Ubicación : México Central
- Punto de conexión : t8mysql2023630402p.mysql.database.azure.com
- Configuración : Con capacidad de 8GB, 81ms, 1 núcleo virtual, 2 GB de RAM, 20 de almacenamiento
- Versión de MySQL : 8.0 Actualizar
- Zona de disponibilidad : 1
- Creado el : 2025-12-09 05:11:17.8681249 UTC

Below this, there's a "Prepárese para la migración" section with a "Iniciar" button.

Script para la creación de la base de datos y las tablas en PaaS

-- 1) Crear base de datos

CREATE DATABASE servicio_web;

-- Usar la base de datos

```
USE servicio_web;
```

```
-- 2) Tabla usuarios
```

```
CREATE TABLE usuarios (
    id_usuario      INT AUTO_INCREMENT PRIMARY KEY,
    password        VARCHAR(64) NOT NULL,
    token           VARCHAR(20),
    email           VARCHAR(100) NOT NULL,
    nombre          VARCHAR(100) NOT NULL,
    apellido_paterno VARCHAR(100) NOT NULL,
    apellido_materno VARCHAR(100),
    fecha_nacimiento DATETIME NOT NULL,
    telefono        BIGINT,
    genero          CHAR(1)
) ENGINE=InnoDB;
```

```
-- 3) Tabla fotos_usuarios
```

```
CREATE TABLE fotos_usuarios (
    id_foto   INT AUTO_INCREMENT PRIMARY KEY,
    foto      LONGBLOB,
    id_usuario INT NOT NULL
) ENGINE=InnoDB;
```

```
-- FK de fotos_usuarios → usuarios
```

```
ALTER TABLE fotos_usuarios
ADD CONSTRAINT fk_fotos_usuarios_usuario
```

```
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario);
```

-- Índice único en email

```
CREATE UNIQUE INDEX usuarios_1 ON usuarios(email);
```

-- 4) Tabla stock (OJO: aquí ya va 'cantidad', no 'existencia')

```
CREATE TABLE stock (
    id_articulo INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    precio DECIMAL(10,2) NOT NULL,
    cantidad INT NOT NULL,      -- ← ya corregido
    id_usuario INT NOT NULL,
    CONSTRAINT fk_stock_usuario
        FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
) ENGINE=InnoDB;
```

-- 5) Tabla fotos_articulos

```
CREATE TABLE fotos_articulos (
    id_foto INT AUTO_INCREMENT PRIMARY KEY,
    foto LONGBLOB NOT NULL,
    id_articulo INT NOT NULL,
    CONSTRAINT fk_foto_articulo
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
) ENGINE=InnoDB;
```

-- 6) Tabla carrito_compra

```
CREATE TABLE carrito_compra (
    id_carrito INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT NOT NULL,
    id_articulo INT NOT NULL,
    cantidad INT NOT NULL,
    CONSTRAINT fk_carrito_usuario
        FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),
    CONSTRAINT fk_carrito_articulo
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
) ENGINE=InnoDB;
```

-- Índice único (id_usuario, id_articulo)

```
CREATE UNIQUE INDEX carrito_compra_ux1
ON carrito_compra (id_usuario, id_articulo);
```

The screenshot shows a Microsoft Azure MySQL service instance named 't8mysql[2023630402]'. The database 'servicio_web' is selected. The 'Tables' section displays the 'carrito_compra' table. The table has the following structure:

Field	Type	Null	Key	Default	Extra
id_articulo	int	NO	PRI	NULL	auto_increment
descripcion	varchar(250)	NO	MUL	NULL	
precio	decimal(10,2)	NO	MUL	NULL	
id_usuario	int	NO	MUL	NULL	

Below the table definition, the command to create the unique index is shown:

```
MySQL [servicio_web]> CREATE UNIQUE INDEX carrito_compra_ux1
ON carrito_compra (id_usuario, id_articulo);
Query OK, 0 rows affected (0.00 sec)
```

At the bottom of the interface, there are tabs for 'Tables', 'Views', 'Functions', 'Procedures', and 'Triggers'.

Pruebas de la app desde el celular

11:43



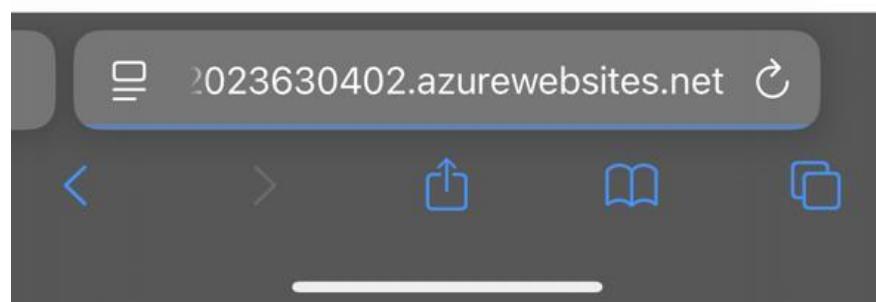
Acceso al sistema

Email

Password

Aceptar

Registro de usuario



11:43



Registro de usuario

Email *

test2@gmail.com

Contraseña *

Nombre *

Pedro

Apellido paterno *

Landeros

Apellido materno

Cortes

Se registró el usuario

Close

Género

Masculino



Choose Files

no files selected

Registrar usuario

Cancelar



≡

2023630402.azurewebsites.net



11:44



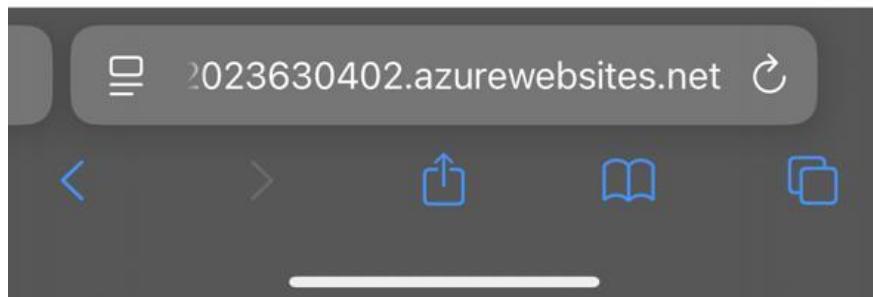
Perfil del usuario

Captura de artículo

Compra de artículos

Eliminar perfil

Salir



11:44



Perfil del usuario

Email *

test2@gmail.com

Contraseña *

Nombre *

Pedro

Apellido paterno *

Landeros

Apellido materno

Cortes

Fecha de nacimiento *

1 Dec 2025 at 11:43 p.m.

Teléfono

11223344

Género

Masculino



Choose Files

no files selected

Quitar foto

Guardar cambios

Regresar



2023630402.azurewebsites.net



11:44



Captura de artículo

Nombre *

Artículo test af

Descripción *

Quwuqu

Precio *

12

Existencia *

123

Se dio de alta el artículo

[Close](#)



[Choose Files](#)



2AFAF4C2-08...AF049.jpeg

[Guardar artículo](#)

[Regresar](#)



2023630402.azurewebsites.net



11:44



Compra de artículos

Palabra clave

Test

Buscar

Carrito de compra



Artículo test af

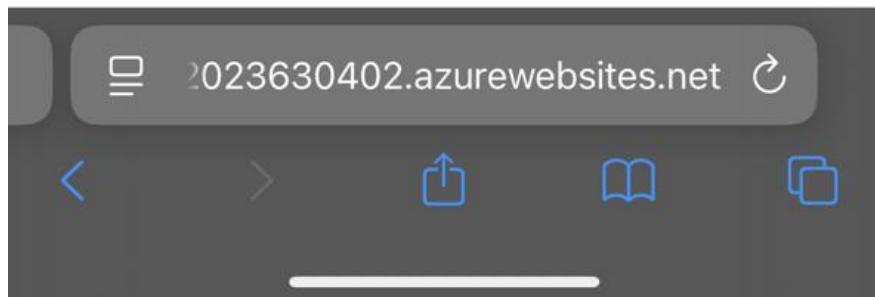
Quwuqu

Precio: \$12

Cantidad:

Compra

[Regresar al menú](#)



11:45



Compra de artículos

Palabra clave

Test

Buscar

Carrito de compra



Artículo test af

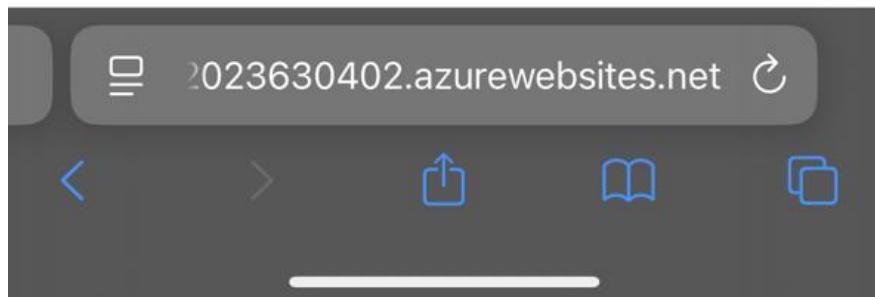
Quwuqu

Precio: \$12

Cantidad:

Compra

[Regresar al menú](#)



11:45



Compra de artículos

Palabra clave

Test

Buscar

Carrito de compra

Se agregó el artículo al carrito

Close

Artículo test ar

Quwuqu

Precio: \$12

Cantidad:

Compra

Regresar al menú



2023630402.azurewebsites.net



11:45



Artículos en el carrito



Artículo test af

Precio: \$12

Cantidad: 4

Subtotal: \$48.00

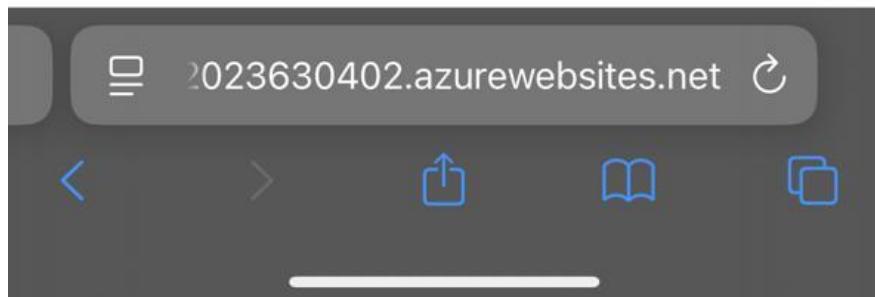
[Eliminar artículo](#)

Total: \$48.00

[Eliminar carrito](#)

[Seguir comprando](#)

[Regresar al menú](#)



11:45



Artículos en el carrito



Artículo test af

Precio: \$12

Cantidad: 4

Subtotal: \$48.00

¿Deseas eliminar este artículo del carrito?

Cancel **OK**

[Seguir comprando](#)

[Regresar al menú](#)



2023630402.azurewebsites.net



11:45



Artículos en el carrito



Artículo test af

Precio: \$12

Cantidad: 4

Subtotal: \$48.00

Se eliminó el artículo del carrito

[Close](#)

[Seguir comprando](#)

[Regresar al menú](#)



2023630402.azurewebsites.net



11:45



Compra de artículos

Palabra clave

Test

Buscar

Carrito de compra



Artículo test af

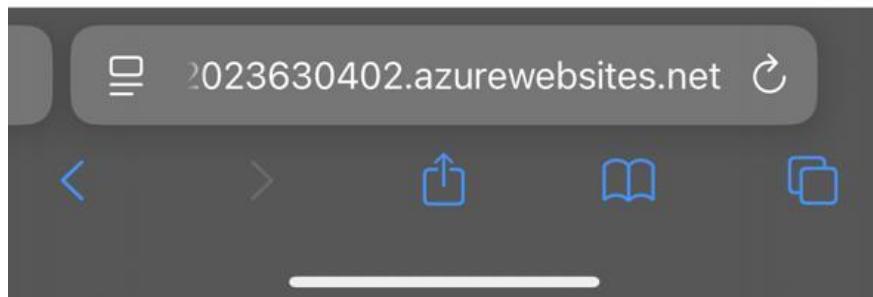
Quwuqu

Precio: \$12

Cantidad:

Compra

[Regresar al menú](#)



11:45



Artículos en el carrito



Artículo test af

Precio: \$12

Cantidad: 3

Subtotal: \$36.00

¿Deseas eliminar todos los artículos del carrito?

Cancel **OK**

[Seguir comprando](#)

[Regresar al menú](#)



2023630402.azurewebsites.net



11:45



Artículos en el carrito



Artículo test af

Precio: \$12

Cantidad: 3

Subtotal: \$36.00

Se eliminó el carrito de compra

[Close](#)

[Seguir comprando](#)

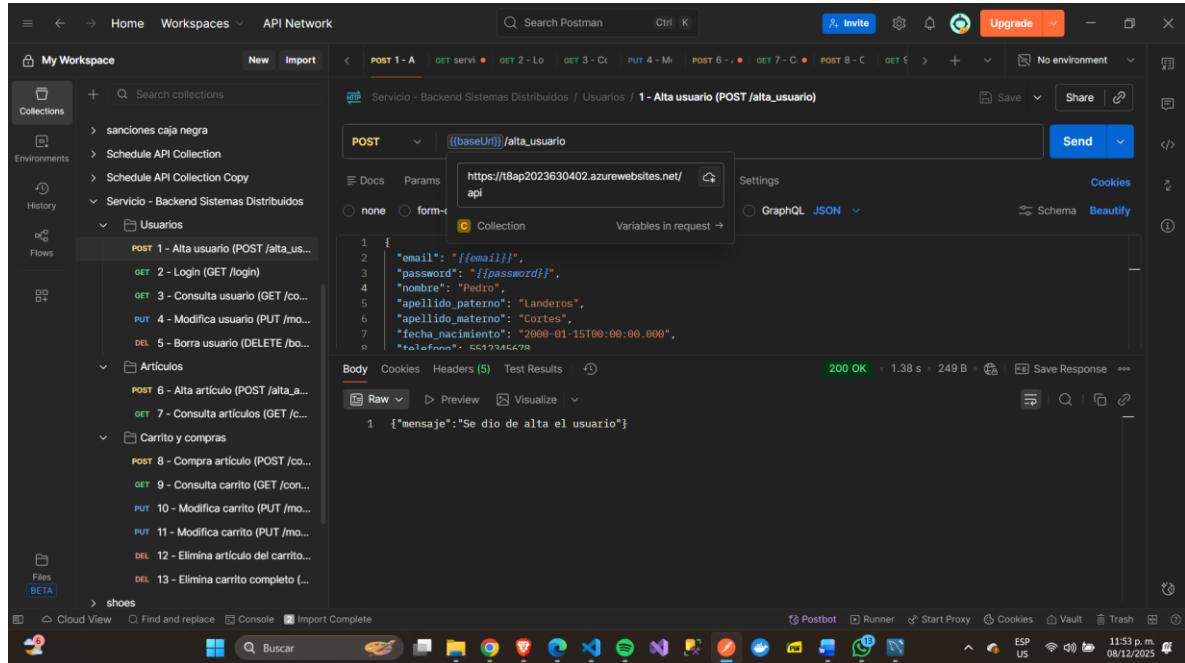
[Regresar al menú](#)



2023630402.azurewebsites.net



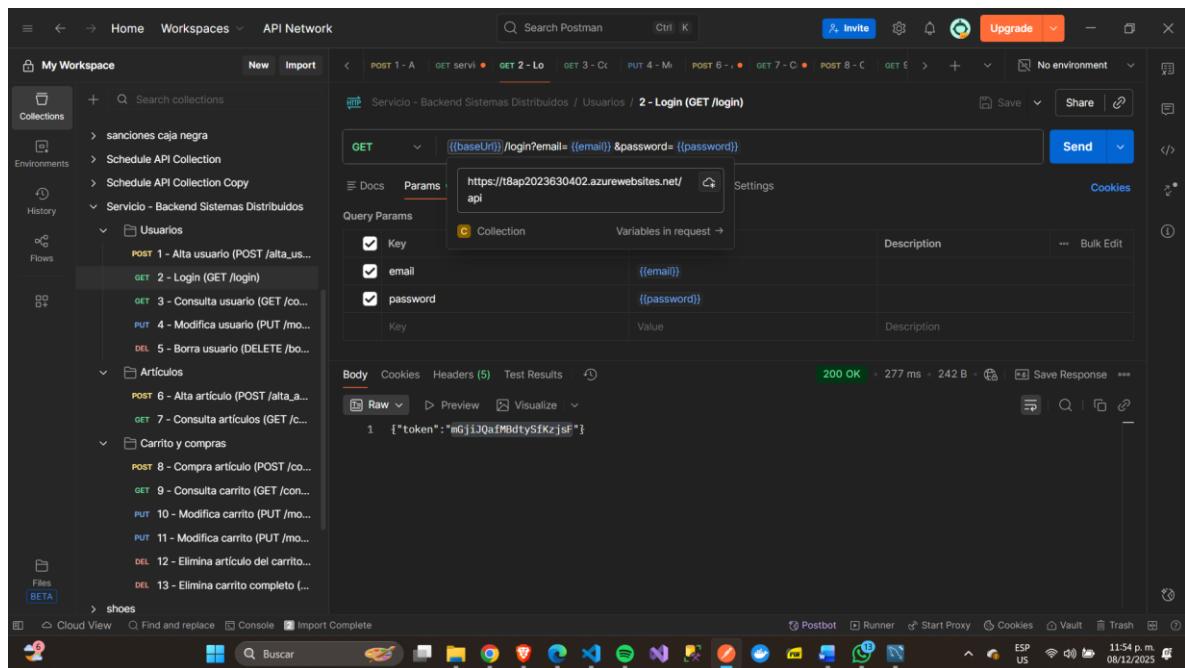
Pruebas de la function app desde Postman



The screenshot shows the Postman interface with the following details:

- Collection:** Servicio - Backend Sistemas Distribuidos / Usuarios
- Request:** POST 1 - Alta usuario (POST /alta_usuario)
- URL:** https://t8ap2023630402.azurewebsites.net/api
- Body (JSON):**

```
1 {
2   "email": "{{email}}",
3   "password": "{{password}}",
4   "nombre": "Pedro",
5   "apellido_paterno": "Landeros",
6   "apellido_materno": "Cortes",
7   "fecha_nacimiento": "2000-01-15T00:00:00.000",
8   "telefono": "+527746479"
```
- Response:** 200 OK (1.38 s, 249 B)



The screenshot shows the Postman interface with the following details:

- Collection:** Servicio - Backend Sistemas Distribuidos / Usuarios
- Request:** GET 2 - Login (GET /login)
- URL:** https://t8ap2023630402.azurewebsites.net/api
- Query Params:**

Key	Description
email	{{{email}}}
password	{{{password}}}
- Body (JSON):**

```
1 { "token": "eGjIjQafMBdtySfKzjs!" }
```
- Response:** 200 OK (277 ms, 242 B)

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** {{baseUrl}} /consulta_usuario?email= {{email}} &token= {{token}}
- Method:** GET
- Params:** email: {{email}}, token: {{token}}
- Body:** Raw JSON response:

```
[{"id_usuario": 3, "email": "a@gmail.com", "nombre": "Pedro", "apellido_paterno": "Landeros", "apellido_materno": "Cortes", "fecha_nacimiento": "2000-01-15T00:00:00", "telefono": 5512345678, "genero": "M", "foto": null}]
```
- Headers:** (5)
- Test Results:** (1)
- Bottom Status Bar:** 411 B, Save Response, 11:54 p.m., 08/12/2025.

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** My Workspace, Collections, Environments, History, Flows, Files (BETA).
- Request URL:** {{baseUrl}} /alta_articulo
- Method:** POST
- Headers:** (10)
- Body:** Raw JSON body:

```
{ "id_usuario": {{id_usuario}}, "token": {{token}}, "nombre": "Lapto", "descripcion": " ", "precio": 15000, "cantidad": 10 }
```
- Headers:** (5)
- Test Results:** (1)
- Bottom Status Bar:** 200 OK, 417 ms, 251 B, Save Response, 11:54 p.m., 08/12/2025.

My Workspace

POST 1 - Alta usuario (POST /alta_usu...
GET 2 - Login (GET /login)
GET 3 - Consulta usuario (GET /co...
PUT 4 - Modifica usuario (PUT /mo...
DEL 5 - Borra usuario (DELETE /bo...
POST 6 - Alta artículo (POST /alta_a...
GET 7 - Consulta artículos (GET /c...
POST 8 - Compra artículo (POST /co...
GET 9 - Consulta carrito (GET /co...
PUT 10 - Modifica carrito (PUT /mo...
PUT 11 - Modifica carrito (PUT /mo...
DEL 12 - Elimina artículo del carri...
DEL 13 - Elimina carrito completo ...
shoes

7 - Consulta artículos (GET /consulta_articulos)

Params

Key	Value	Description
palabra	laptop	
id_usuario	1	
token	123456789	

Body

```
[{"id_articulo":2,"nombre":"Laptop Gamer 2","descripcion":"laptop potente para juegos","precio":15600.5,"cantidad":10,"foto":null}]
```

200 OK

11:55 p. m.
08/12/2025

My Workspace

POST 1 - Alta usuario (POST /alta_usu...
GET 2 - Login (GET /login)
GET 3 - Consulta usuario (GET /co...
PUT 4 - Modifica usuario (PUT /mo...
DEL 5 - Borra usuario (DELETE /bo...
POST 6 - Alta artículo (POST /alta_a...
GET 7 - Consulta artículos (GET /c...
POST 8 - Compra artículo (POST /co...
GET 9 - Consulta carrito (GET /co...
PUT 10 - Modifica carrito (PUT /mo...
PUT 11 - Modifica carrito (PUT /mo...
DEL 12 - Elimina artículo del carri...
DEL 13 - Elimina carrito completo ...
shoes

7 - Consulta artículos (GET /consulta_articulos)

Params

Key	Value	Description
palabra	test	
id_usuario	1	
token	123456789	

Body

```
[{"id_articulo":1,"nombre":"Articulo test af","descripcion":"Ouwuqu","precio":12.00,"cantidad":123,"foto":"/9j/40cicXhpzATAU8AkgAAAACMEPAIAAAAAGAAAAkQEAAIAAAKAAAAMESAMAAAABAYAAAeAUAAAABAAAoggEbAUAAAABAAAoggFoAMAAAA..."}]
```

200 OK

11:55 p. m.
08/12/2025

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, environments, flows, and files. The main workspace displays a POST request to `((baseUrl)) /compra_articulo`. The request body is defined as follows:

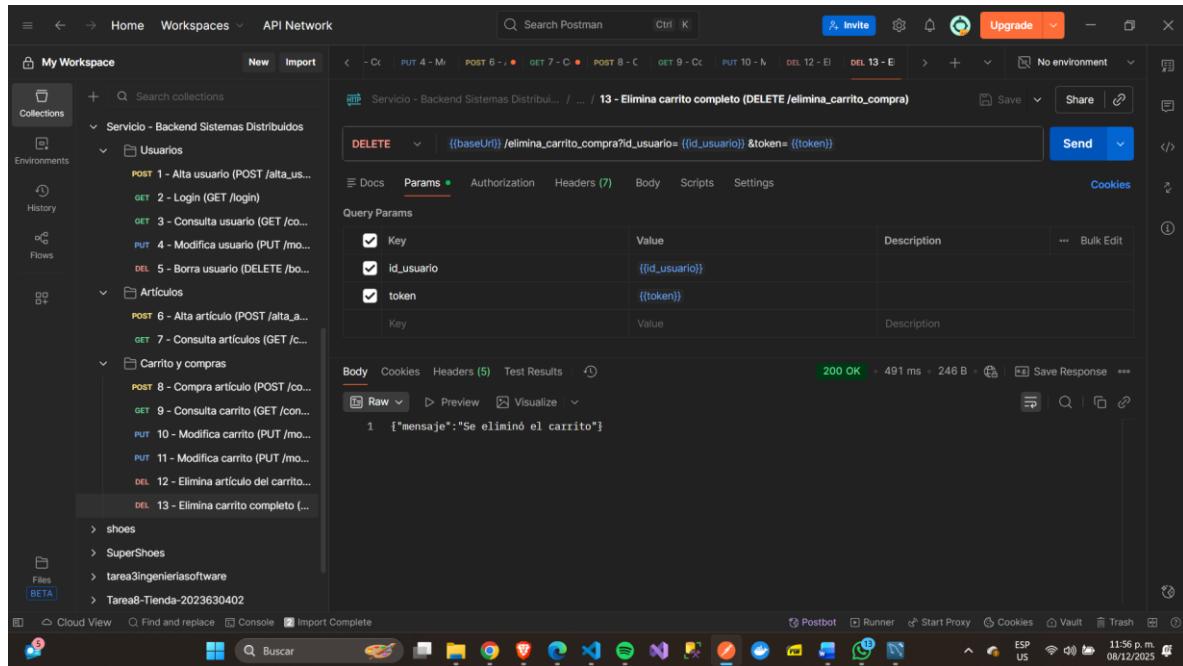
```
1 {
  "id_usuario": "{{id_usuario}}",
  "token": "{{token}}",
  "id_articulo": "{{id_articulo}}",
  "cantidad": 2
}
```

The response status is 200 OK with a duration of 516 ms and a size of 253 B. The response body contains the message: `{"mensaje": "Articulo agregado al carrito"}`.

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists collections, environments, flows, and files. The main workspace displays a GET request to `((baseUrl)) /consulta_carrito_compra?id_usuario={{id_usuario}}&token={{token}}`. The request includes the following query parameters:

Key	Value	Description
<code>id_usuario</code>	<code>{{id_usuario}}</code>	
<code>token</code>	<code>{{token}}</code>	

The response status is 200 OK with a duration of 671 ms and a size of 1.14 MB. The response body is a large JSON object containing detailed information about the cart items.



Conclusiones

El desarrollo de este prototipo de sistema de comercio electrónico basado en una arquitectura serverless me permitió comprender y aplicar de manera práctica diferentes conceptos fundamentales relacionados con los servicios en la nube. A lo largo del proyecto pude implementar un backend funcional utilizando Azure Functions en .NET 8, integrado con una base de datos MySQL desplegada como servidor flexible PaaS. También configuré el acceso seguro mediante TLS, reglas de firewall y variables de entorno, lo cual me ayudó a entender mejor cómo se establece la comunicación entre servicios distribuidos.

Además, implementé cada uno de los requerimientos funcionales solicitados: el registro y autenticación de usuarios, la captura y consulta de artículos, la compra mediante actualización transaccional de stock y la administración completa del carrito de compra. Para lograrlo, trabajé con transacciones, validaciones y control de concurrencia, lo que me permitió reforzar mi habilidad para manejar la integridad de datos en un sistema distribuido. De igual manera, adapté el front-end para interactuar correctamente con las funciones, y pude desplegarlo en Azure Files, verificando la correcta entrega de contenido en la nube.

El proceso de despliegue me llevó también a crear y configurar recursos en Azure, como la Function App, el almacenamiento, el File Share y el servidor MySQL, así

como su conexión mediante credenciales y cadenas de conexión. Este flujo completo me ayudó a comprender cómo diferentes servicios cloud se integran para formar una solución funcional, escalable y sin necesidad de administrar servidores directamente.

Finalmente, las pruebas realizadas tanto desde herramientas locales como desde un dispositivo móvil confirmaron que el sistema cumple con los requerimientos del back-end y del front-end. En general, este proyecto representó una experiencia completa que me permitió unir desarrollo, configuración de infraestructura y despliegue en la nube, fortaleciendo mi entendimiento sobre cómo se construyen aplicaciones modernas basadas en arquitectura serverless.