



**Instituto Politécnico Nacional**  
**Escuela Superior de Computo**



## **Sistemas Distribuidos**

### **Tarea 9**

**Prototipo de sistema de comercio electrónico utilizando  
microservicios sobre kubernetes**

Nombre del alumno:

García Quiroz Gustavo Ivan

Grupo: 7CV4

Nombre del profesor: Guerrero Carlos Pineda

Fecha de entrega: 23/12/2025

## ÍNDICE

1	Introducción .....	1
2	Objetivos .....	3
2.1	Objetivo general.....	3
2.2	Objetivos específicos .....	3
3	Alcance y nomenclatura oficial.....	5
4	Arquitectura del sistema.....	7
5	Desarrollo.....	9
6	Recursos y configuración en Azure.....	9
6.1	Instalación y configuración inicial.....	9
6.2	Grupo de Recursos .....	11
6.3	Azure Container Registry (ACR).....	11
6.4	Azure Kubernetes Service (AKS).....	15
6.5	MySQL Flexible Server (PaaS) .....	25
6.6	Storage Account y Azure Files (front-end) .....	33
6.7	Estructurar los microservicios en VS Code Crea cuatro proyectos Azure Functions .NET 8 aislado.....	41
6.8	Creación de los esquemas SQL de cada microservicio .....	47
7	Back-end: Diseño y funciones por microservicio .....	51
7.1	Microservicio Gestión de usuarios — t9_2022630278_gu .....	52
7.2	Microservicio Gestión de artículos — t9_2022630278_ga .....	53
7.3	Microservicio Gestión de compras — t9_2022630278_gc .....	54
7.4	Microservicio Servidor web — t9_2022630278_sw .....	56
7.5	Dockerfiles de cada microservicio.....	58
8	Conexión al cluster AKS desde VS Code o Cloud Shell .....	63

8.1	Crear PV y PVC para Azure Files (front-end).....	75
9	API Gateway en AKS .....	80
9.1	Implementación de SimpleAPIServer.java.....	80
9.2	Construcción de imagen Docker y publicación en ACR .....	81
10	Evidencias requeridas .....	90
10.1	Evidencias del front end en dispositivo móvil.....	90
10.2	Evidencias de pruebas unitarias del back end (curl).....	101
10.2.1	Alta usuario (GU): .....	101
10.2.2	Login (GU): .....	102
10.2.3	Consulta usuario (GU): .....	103
10.2.4	Verifica acceso (GU):.....	104
10.2.5	Alta artículo (GA) — inserta artículo y llama a GC.alta_articulo: .....	105
10.2.6	Consulta artículos (GA):.....	106
10.2.7	Compra artículo (GC):.....	107
10.2.8	Consulta carrito (GC):.....	108
10.2.9	Modificar carrito (+1) (GC):.....	109
10.2.10	Modificar carrito (-1) (GC): .....	110
10.2.11	Eliminar artículo del carrito (GC):.....	111
10.2.12	Eliminar carrito (GC):.....	112
10.2.13	Finaliza compra (GC, opcional):.....	113
11	Conclusiones .....	115
	Enlace del chat de la IA generativa .....	116
12	Referencias (Formato IEEE).....	117

# 1 Introducción

Este reporte documenta el desarrollo de la “Tarea 9: Prototipo de sistema de comercio electrónico utilizando microservicios sobre Kubernetes”. El objetivo fue construir un sistema distribuido basado en microservicios que corre en Azure Kubernetes Service (AKS), con infraestructura en la región Canada Central, siguiendo la nomenclatura y lineamientos indicados por el profesor. Para lograr una ejecución ágil y consistente, se reutilizó el código de la Tarea 8 (back-end en .NET 8 y front-end web) y se separó lógicamente el dominio en cuatro microservicios: Gestión de usuarios, Gestión de artículos, Gestión de compras y Servidor web estático (Get.cs).

Durante el desarrollo se utilizó la inteligencia artificial de GitHub Copilot como apoyo para acelerar tareas de codificación y operación: generación de esqueletos de funciones HTTP en C#, plantillas de Dockerfiles, manifests YAML para Kubernetes, rutas del API Gateway (SimpleAPIGateway.java), y ejemplos de pruebas con curl. Copilot facilitó la estandarización del código, la alineación con la nomenclatura requerida y la verificación de escenarios funcionales, complementando el trabajo manual y la comprensión arquitectónica realizada por el alumno.

La arquitectura resultante orquesta los componentes de Azure necesarios para un despliegue realista de un e commerce minimalista: AKS para ejecutar contenedores, ACR como registro de imágenes, MySQL en PaaS para persistencia separada por microservicio, y Azure Files (PV/PVC) para publicar el front-end a través del microservicio de servidor web. La exposición pública se hace mediante el API Gateway, que enruta las funciones /api/\* hacia servicios internos de ClusterIP. Con esta aproximación, se logra aislamiento de datos, escalabilidad básica (réplicas de pods), y claridad en los límites de cada servicio.

Componentes principales:

- Microservicios en .NET 8: Gestión de usuarios (GU), Gestión de artículos (GA), Gestión de compras (GC) y Servidor web (SW).

- API Gateway (SimpleAPIServer.java) desplegado como LoadBalancer y enrutando a servicios ClusterIP.
- MySQL Flexible Server con tres bases de datos: t9\_2022630278\_bdgu, t9\_2022630278\_bdga y t9\_2022630278\_bdgc.
- Azure Files montado en el microservicio SW mediante PV/PVC para servir el front-end (prueba.html, WSClient.js).
- Registro de contenedores ACR y despliegue en AKS con nomenclatura oficial.

## **2 Objetivos**

### **2.1 Objetivo general**

Diseñar, contenedorizar y desplegar un prototipo de e-commerce basado en microservicios sobre AKS (Canada Central), con un API Gateway frontal, bases de datos independientes por servicio en MySQL PaaS, publicación del front-end vía Servidor web (Get.cs), y validación completa de funcionalidades mediante pruebas en navegador/dispositivo móvil y curl.

### **2.2 Objetivos específicos**

- Construir cuatro proyectos contenedorizados:
  - t9\_2022630278\_gu (Gestión de usuarios), t9\_2022630278\_ga (Gestión de artículos), t9\_2022630278\_gc (Gestión de compras) y t9\_2022630278\_sw (Servidor web).
- Proveer y configurar recursos de Azure en Canada Central: ACR (t92022630278acr), AKS, MySQL Flexible Server (t9-2022630278-bd) y Storage Account con Azure Files para el front-end.
- Crear las tres bases de datos con esquemas y restricciones exigidas (índices únicos en “carrito\_compra” y “stock” de GC; tablas “usuarios” y “fotos\_usuarios” en GU; “stock” y “fotos\_articulos” en GA).
- Implementar endpoints conforme a requisitos:
  - GU (alta, consulta, modifica, borra, login, verifica\_acceso), GA (alta\_articulo con llamada a GC y consulta\_articulos), GC (alta\_articulo de cantidad, compra\_articulo con transacciones, consulta/elimina/modifica carrito, y opcional finaliza\_compra).
- Asegurar autenticación por token con verificación centralizada en GU y consumo por GA/GC vía HTTP (sin acceso cruzado a bases de datos).
- Desplegar el API Gateway con tabla de enrutamiento hacia servicios ClusterIP y exponer públicamente con Service tipo LoadBalancer.

- Montar PV/PVC en el microservicio SW para servir el front-end y validar la navegación completa: captura de artículo, compra, carrito, total y acciones (+, -, eliminar, eliminar carrito, seguir comprando).
- Publicar imágenes en ACR, aplicar manifests en AKS y verificar el estado de pods/servicios; escalar réplicas para resiliencia básica.
- Ejecutar pruebas unitarias con curl para cada endpoint y recolectar evidencias en dispositivo móvil siguiendo los lineamientos (capturas completas, legibles y con fecha/hora).

### **3 Alcance y nomenclatura oficial**

Se realizó el prototipo de sistema de comercio electrónico bajo una arquitectura de microservicios ejecutando en Azure Kubernetes Service (AKS), con separación de responsabilidades y de bases de datos por servicio. Se instaló y configuró la infraestructura en la región Canada Central, y se accedió a cada componente utilizando la nomenclatura oficial definida por el profesor. El alcance cubre: construcción de imágenes Docker, publicación en Azure Container Registry (ACR), despliegue en AKS con Services de tipo ClusterIP por microservicio, API Gateway con LoadBalancer, y montaje de un recurso compartido de archivos (Azure Files) vía PV/PVC para el front-end.

La práctica se enfocó en cumplir los requerimientos funcionales (alta, consulta, modificación y borrado de usuarios; administración de artículos; compras y carrito de compra; servidor web de archivos estáticos) y no funcionales (aislamiento de datos, uso de MySQL PaaS, nomenclatura obligatoria), además de recolectar evidencias para el reporte (capturas completas, legibles y con fecha/hora).

- Nomenclatura oficial aplicada:
  - Proyectos (Visual Studio Code, contenedrizados):
    - t9\_2022630278\_gu — Gestión de usuarios
    - t9\_2022630278\_ga — Gestión de artículos
    - t9\_2022630278\_gc — Gestión de compras
    - t9\_2022630278\_sw — Servidor web (Get.cs)
  - Bases de datos (MySQL PaaS):
    - t9\_2022630278\_bdgu — Gestión de usuarios (usuarios, fotos\_usuarios)
    - t9\_2022630278\_bdga — Gestión de artículos (stock, fotos\_articulos)
    - t9\_2022630278\_bdgc — Gestión de compras (stock con índice único e id\_articulo, carrito\_compra con índice único sobre id\_usuario + id\_articulo)

- Instancia MySQL Flexible Server:
  - t9-2022630278-bd
- Registro de contenedores (ACR):
  - t92022630278acr

## 4 Arquitectura del sistema

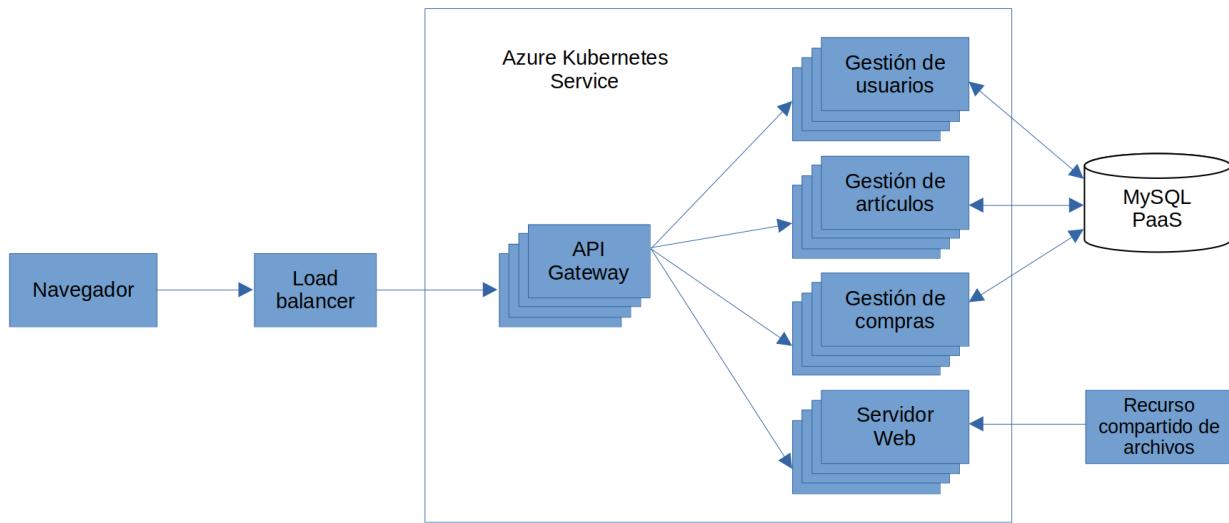
Se realizó la arquitectura de referencia indicada en clase, donde el navegador accede al API Gateway expuesto por un Service de tipo LoadBalancer. El API Gateway enruta cada URL “/api/\*” hacia el Service ClusterIP del microservicio correspondiente. Cada microservicio usa su propia base de datos en MySQL PaaS, y no se accedió a la base de datos de otro microservicio. El microservicio de Servidor Web montó un PV/PVC respaldado por Azure Files para servir el front-end (contenido estático).

La arquitectura se compone de los siguientes elementos principales, ejecutando en AKS con réplicas configurables:

- API Gateway (SimpleAPIGateway.java): se instaló en contenedor, se expuso con LoadBalancer, y se accedió vía HTTPS/HTTP hacia “/api/\*”.
- Gestión de usuarios (t9\_2022630278\_gu): funciones de alta, consulta, modifica, borra, login y verifica\_acceso; su DB es “t9\_2022630278\_bdgu”.
- Gestión de artículos (t9\_2022630278\_ga): funciones de alta\_articulo (insertando en su propia “stock” y enviando cantidad a GC) y consulta\_articulos; su DB es “t9\_2022630278\_bdga”.
- Gestión de compras (t9\_2022630278\_gc): funciones de alta\_articulo (cantidad), compra\_articulo, consulta\_carrito, elimina\_articulo\_carrito\_compra, elimina\_carrito\_compra y modifica\_carrito\_compra; su DB es “t9\_2022630278\_bdgc” con transacciones en operaciones críticas.
- Servidor web (t9\_2022630278\_sw): función Get.cs que sirvió archivos estáticos desde el mount del PV/PVC (Azure Files).

Se accedió al diagrama arquitectónico para verificar el flujo: navegador → LoadBalancer → API Gateway → Services ClusterIP (GU, GA, GC, SW) → MySQL PaaS y Azure Files. Después de validar el routing y el aislamiento de datos, se insertó la figura correspondiente.

Antes de la figura, debería verse el diagrama completo que vincula el navegador con el LoadBalancer y, desde ahí, al API Gateway y a cada microservicio; se recomienda incluir la versión final del diagrama empleado en clase y anotar que podrán existir varias réplicas por servicio.



**Figura 5. Diagrama de arquitectura del sistema en AKS con API Gateway y microservicios**

## 5 Desarrollo

## 6 Recursos y configuración en Azure

### 6.1 Instalación y configuración inicial

Se realizó la instalación y/o actualización de las herramientas necesarias para el desarrollo del sistema:

Logo	Herramienta
	Azure CLI
	Docker Desktop
	Visual Studio Code con las extensiones de C# (ms-dotnettools.csharp) y Azure Functions
	Azure Functions Core Tools (para .NET 8 aislado)
	El cliente de Kubernetes (kubectl) mediante el comando az aks install-cli.

**Tabla 1 Herramientas utilizadas en la tarea.**

Posteriormente, se inició sesión en Azure con az login, verificando la suscripción activa de Azure for Students mediante az account show.

```

Command Prompt x + 
C:\Users\ivan->az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No   Subscription name   Subscription ID           Tenant
--- 
[1] *  Azure for Students  fd58a3da-fcef-47d1-ac0e-5b891faa4251  Instituto Politecnico Nacional

The default is marked with an *; the default tenant is 'Instituto Politecnico Nacional' and subscription is 'Azure for Students' (fd58a3da-fcef-47d1-ac0e-5b891faa4251).

Select a subscription and tenant (Type a number or Enter for no changes): 1

Tenant: Instituto Politecnico Nacional
Subscription: Azure for Students (fd58a3da-fcef-47d1-ac0e-5b891faa4251)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

C:\Users\ivan->az account show
21°C Mostly sunny  Search  05:36 p.m. 21/12/2025
ENG ES Wi-Fi ID  Command Prompt x + 
[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

C:\Users\ivan->az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": "f94bf4d9-8097-4794-adf6-a5466ca28563",
  "id": "fd58a3da-fcef-47d1-ac0e-5b891faa4251",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure for Students",
  "state": "Enabled",
  "tenantDefaultDomain": "correo.ipn.mx",
  "tenantDisplayName": "Instituto Politecnico Nacional",
  "tenantId": "f94bf4d9-8097-4794-adf6-a5466ca28563",
  "user": {
    "name": "ggarciaq1800@alumno.ipn.mx",
    "type": "user"
  }
}

C:\Users\ivan->
21°C Mostly sunny  Search  05:36 p.m. 21/12/2025

```

**Figura 6.1 Inicio de sesión**

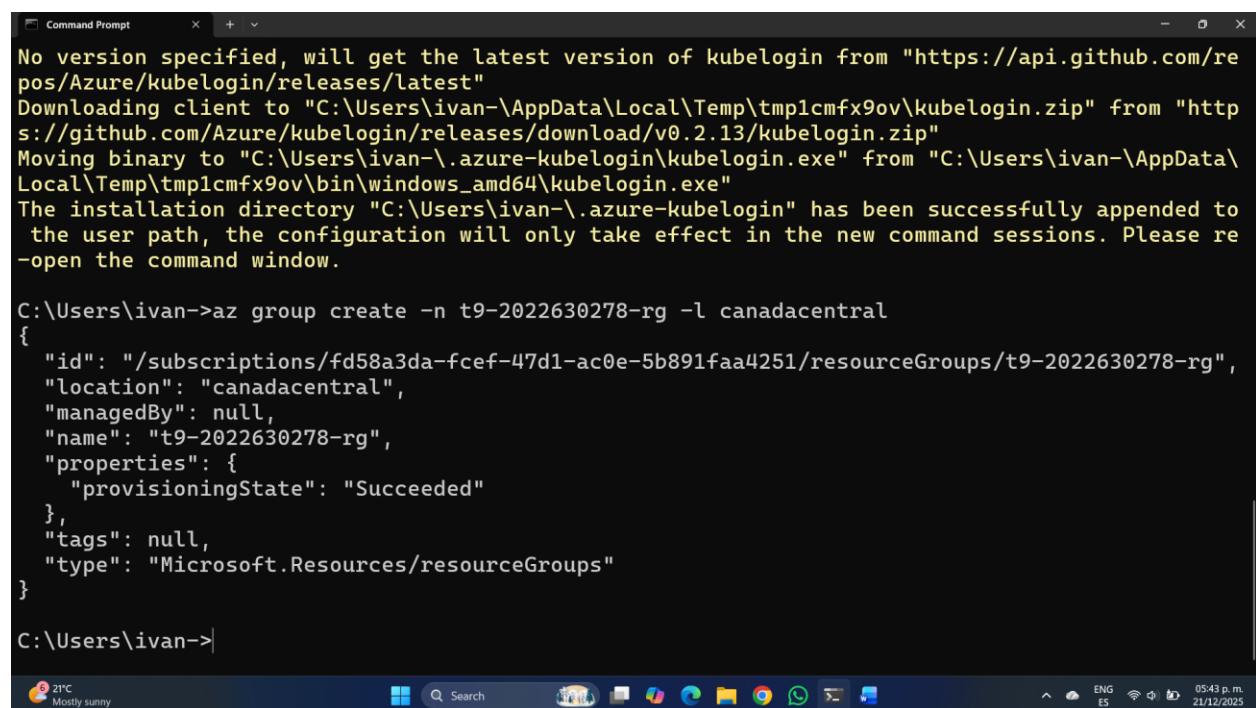
Se realizó la provisión y configuración de los recursos en Azure en la región Canada Central (canadacentral), respetando la nomenclatura oficial. Se instaló el registro de contenedores, el clúster de Kubernetes, la base de datos MySQL en PaaS y el almacenamiento de archivos para el front-end. Adicionalmente, se accedió a los paneles

de configuración para crear y validar variables de entorno, secretos y mapas de configuración utilizados por los microservicios.

## 6.2 Grupo de Recursos

Se creó el Grupo de recursos que se va a usar en esta práctica usando el siguiente comando en la terminal de Windows:

```
az group create -n t9-2022630278-rg -l canadacentral
```



```
No version specified, will get the latest version of kubelogin from "https://api.github.com/repos/Azure/kubelogin/releases/latest"
Downloading client to "C:\Users\ivan-\AppData\Local\Temp\tmplcmfx9ov\kubelogin.zip" from "https://github.com/Azure/kubelogin/releases/download/v0.2.13/kubelogin.zip"
Moving binary to "C:\Users\ivan-\.azure-kubelogin\kubelogin.exe" from "C:\Users\ivan-\AppData\Local\Temp\tmplcmfx9ov\bin\windows_amd64\kubelogin.exe"
The installation directory "C:\Users\ivan-\.azure-kubelogin" has been successfully appended to the user path, the configuration will only take effect in the new command sessions. Please re-open the command window.

C:\Users\ivan->az group create -n t9-2022630278-rg -l canadacentral
{
  "id": "/subscriptions/fd58a3da-fcef-47d1-ac0e-5b891faa4251/resourceGroups/t9-2022630278-rg",
  "location": "canadacentral",
  "managedBy": null,
  "name": "t9-2022630278-rg",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}

C:\Users\ivan->
```

**Figura 6.2 Creación del grupo de recursos en Azure “t9-2022630278-rg” creado en Canada Central**

## 6.3 Azure Container Registry (ACR)

Se realizó la creación del registro de contenedores con la nomenclatura requerida: t92022630278acr. Este registro almacenó las imágenes Docker de los cuatro microservicios (GU, GA, GC, SW) y del API Gateway. Se configuró el acceso desde AKS para permitir pull de imágenes al momento del despliegue.

- Se accedió al portal de Azure y se verificó el nombre, la región y la suscripción de estudiantes.
- Se instaló el vínculo de acceso entre AKS y ACR (attach) para autorizar la extracción de imágenes desde el clúster.

Antes de la figura, debería verse el recurso ACR en “Overview”, mostrando nombre t92022630278acr, ubicación Canada Central y el estado del servicio.

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with 'Azure services' (Create a resource, Cost Management), 'Resources' (Recent, Favorite, Name: Azure for Students, Marketplace), and 'Navigate' (Subscriptions, Resource groups, All resources, Dashboard). The main area has a search bar ('Container Registries') and a 'Services' tab showing 12 items. It also includes sections for Firewalls, Virtual networks, and More services. A 'Last Viewed' section shows 'Log Analytics workspace' (2 weeks ago) and 'DefaultWorkspace-fd58a3da-fcef-47d1-ac0e-5b091faa4251-CCAN' (2 hours ago). The bottom navigation bar includes Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management.

The screenshot shows the 'Create container registry' wizard in the Microsoft Azure portal. The current step is 'Basics'. The 'Project details' section includes a 'Subscription' dropdown set to 'Azure for Students' and a 'Resource group' dropdown set to 't9-2022630278-rg'. The 'Instance details' section includes a 'Registry name' input field containing 't92022630278acr', a 'Location' dropdown set to 'Canada Central', and a 'Domain name label scope' dropdown set to 'Unsecure'. The 'Registry domain name' field shows 't92022630278acr.azurecr.io'. There are buttons for 'Review + create' and 'Next: Networking >'. The bottom navigation bar includes Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management.

**Create container registry**

Subscription: Azure for Students  
Resource group: t9-2022630278-rg

Instance details

Registry name: t92022630278acr.azurecr.io

Location: Canada Central

Domain name label scope: Unsecure

Registry domain name: t92022630278acr.azurecr.io

Use availability zones:  Availability zones are activated on premium registries and in regions that support availability zones. [Learn more](#)

Pricing plan: Standard

Role assignment permissions mode: RBAC Registry + ABAC Repository Permissions (radio button selected)

Buttons: Review + create, < Previous, Next: Networking >

**Create container registry**

Validation passed

Basics Networking Encryption Tags Review + create

Registry details

Basics

Registry name	t92022630278acr
Subscription	Azure for Students
Resource Group	t9-2022630278-rg
Location	Canada Central
Domain name label scope	Unsecure
Availability zones	Disabled
Pricing plan	Standard
Role assignment permissions mode	RBAC Registry Permissions

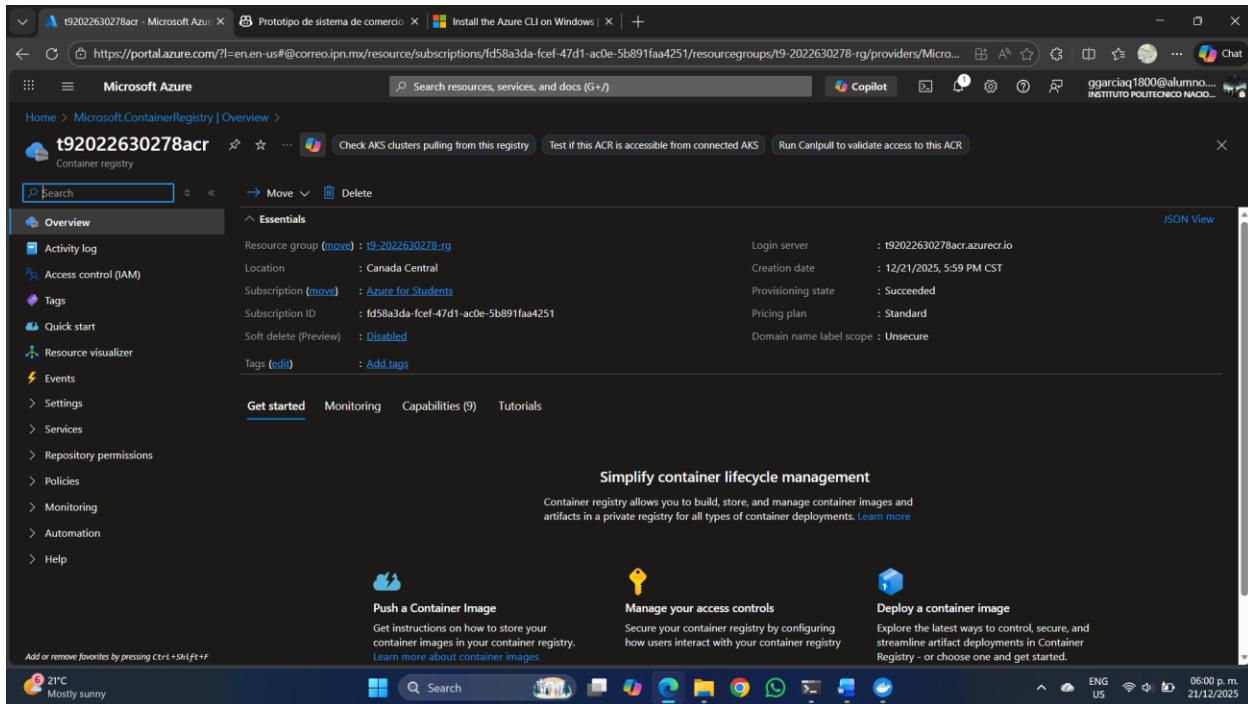
Networking

Public network access: Yes

Encryption

Customer-Managed Key: Disabled

Buttons: Create, < Previous, Next >, Download a template for automation



**Figura 6.3 Azure Container Registry “t92022630278acr” creado en Canada Central**

## 6.4 Azure Kubernetes Service (AKS)

Se realizó la creación del clúster AKS con dos nodos, habilitando las credenciales de acceso para kubectl. Se accedió a la vista de “Workloads” para confirmar el correcto despliegue posterior de los pods y servicios (ClusterIP y LoadBalancer). Se instaló la integración con ACR para facilitar el uso de las imágenes publicadas.

- Se accedió al panel de AKS para revisar nodos, namespaces y recursos de trabajo.
- Se instalaron las herramientas de línea de comando necesarias (Azure CLI y kubectl) y se aplicaron los manifests de Kubernetes en pasos posteriores.

Antes de la figura, debería verse el clúster AKS en estado “Running” con su nombre asignado, región Canada Central y detalles de nodos; se recomienda capturar “Properties” y “Workloads”.

Kubernetes service

All Services (92) Marketplace (31)

Services

Kubernetes service - Automatic

Kubernetes services

Virtual machines Firewalls More services

Microsoft Entra ID

Azure Kubernetes Service AAD Server Service Principal

Azure Kubernetes Service - Fleet RP Service Principal

Last Viewed

a few seconds ago

a few seconds ago

2 hours ago

2 weeks ago

See all

Log Analytics workspace

Subscriptions Resource groups All resources Dashboard

Tools

21°C Mostly sunny

Search

ENG US 06:00 p.m. 21/12/2025

Kubernetes center (preview) | Clusters

No Kubernetes services to display

+ Create

Create Kubernetes cluster

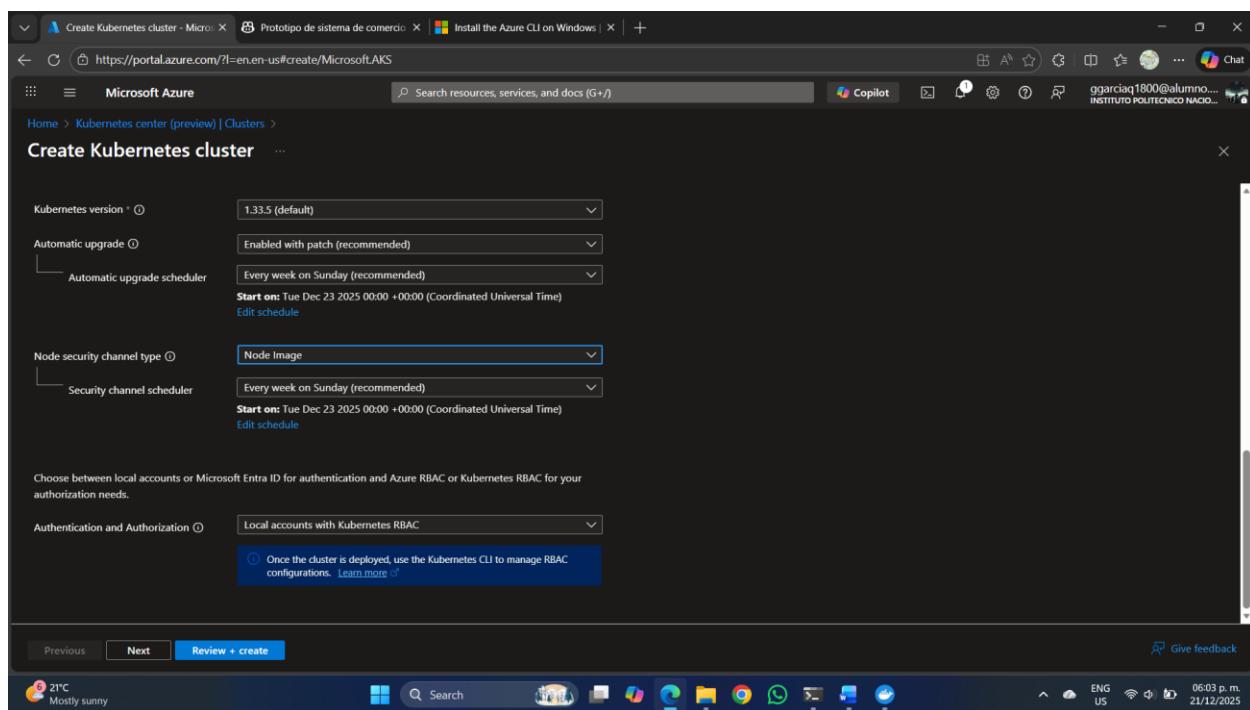
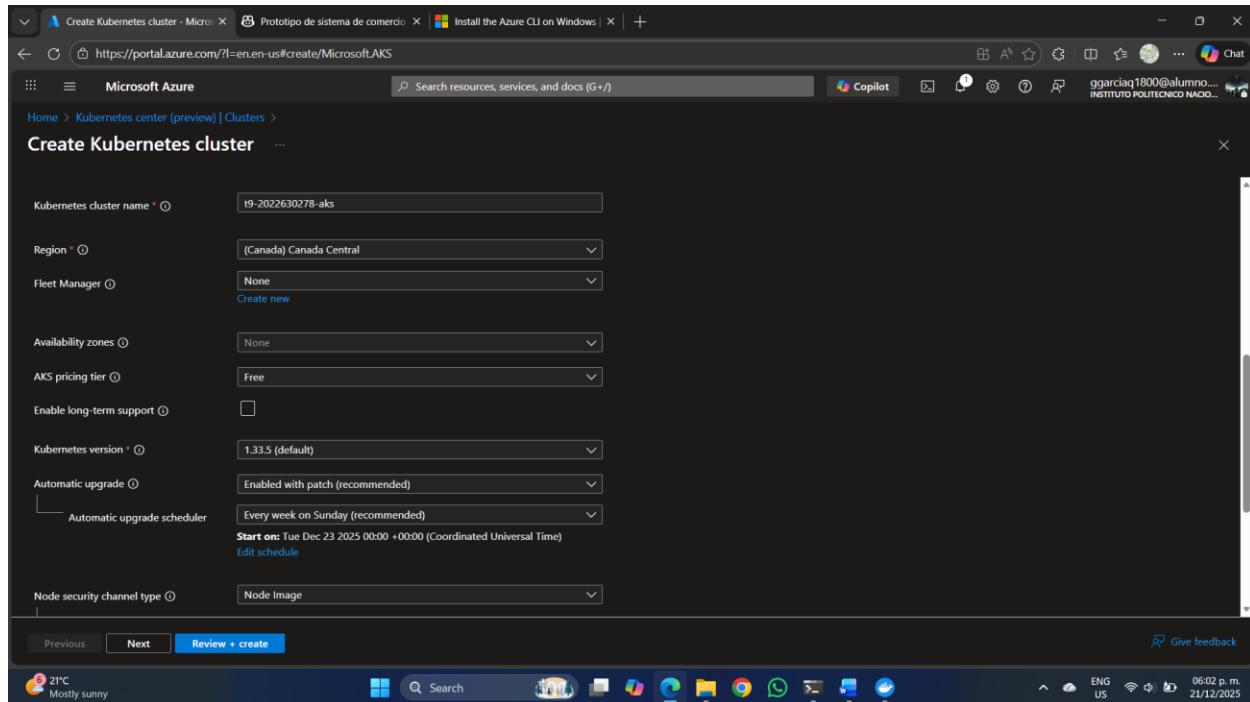
Basics

Subscription: Azure for Students

Resource group: t9-2022630278-rg

Cluster preset configuration: Dev/Test

Review + create



The screenshot shows the 'Update node pool' page in the Azure portal. The 'Node pool name' is set to 'agentpool'. The 'Mode' is selected as 'System'. The 'OS SKU' is set to 'Ubuntu Linux'. The 'Availability zones' dropdown is set to 'Zones 1, 2, 3'. The 'Enable Azure Spot instances' checkbox is unchecked. At the bottom, there are 'Next: Optional settings >' and 'Update' buttons.

The screenshot shows the 'Create Kubernetes cluster' page in the Azure portal. The 'Node pools' tab is selected. A table lists a single node pool named 'agentpool' with the following details:

Name	Mode	Node size	OS SKU	Node count	Ava
agentpool	System	Standard_D2s_v3 (change)	Ubuntu	2 - 5	1,2,

At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Microsoft Azure portal. The current step is 'Node pool configuration'. A table lists a single node pool named 'agentpool' with a 'Standard\_D2s\_v3' mode and 'Ubuntu' OS SKU. The 'Node count' is set to '2 - 5' and 'Availability zone' is '1,2'. Below the table, there's a section for 'Enable virtual nodes' with a checkbox. Under 'Node pool OS disk encryption', it says 'By default, all disks in AKS are encrypted at rest with Microsoft-managed keys.' An 'Encryption type' dropdown is set to '(Default) Encryption at-rest with a platform-managed key'. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Microsoft Azure portal. The current step is 'Networking'. It includes tabs for 'Recommend VM sizes for Kubernetes Service', 'Recommend VM sizes for my AI workloads', 'Basics', 'Node pools', 'Networking' (which is selected), 'Integrations', 'Monitoring', 'Security', 'Advanced', 'Tags', and 'Review + create'. In the 'Networking' section, it says 'Azure provides various networking controls to help manage and secure access to your Kubernetes cluster.' Under 'Private access', there's a checkbox for 'Enable private cluster'. Under 'Public access', there's a checkbox for 'Set authorized IP ranges'. Under 'Container networking', 'Network configuration' is set to 'Azure CNI Overlay' (selected with a blue radio button). A note says 'Assigns pod IP addresses from a private IP space. Best for scalability.' Another option, 'Azure CNI Node Subnet', is also listed. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Network configuration'. It includes the following settings:

- Azure CNI Node Subnet**: Previously named Azure CNI. Assigns pod IP addresses from your host VNet. Best for workloads where pods must be reachable by other VNet resources.
- Bring your own Azure virtual network**: An unchecked checkbox.
- DNS name prefix**: t9-2022630278-aks-dns
- Enable Cilium dataplane and network policy engine**: An unchecked checkbox.
- Network policy engine**: A radio button group with "None" selected (blue), which allows all ingress and egress traffic to the pods. Other options include "Calico" (open-source networking solution) and "Azure" (native networking solution).
- Load balancer**: Standard.

At the bottom, there are "Previous", "Next", and "Review + create" buttons. The status bar at the bottom right shows the date and time: 06:18 p.m. 21/12/2025.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Integration services'.

Key sections visible include:

- Recommend VM sizes for Kubernetes Service** and **Recommend VM sizes for Kubernetes Service for my AI workloads**.
- Basics**, **Node pools**, **Networking**, **Integrations** (selected), **Monitoring**, **Security**, **Advanced**, **Tags**, and **Review + create** buttons.
- Azure Container Registry**: A section for connecting the cluster to an Azure Container Registry.
- Service mesh - Istio**: A section for enabling Istio for traffic management and observability.
- Azure Policy**: A section for applying policy enforcements and safeguards for AKS clusters.

At the bottom, there are "Previous", "Next", and "Review + create" buttons. The status bar at the bottom right shows the date and time: 06:19 p.m. 21/12/2025.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Connect your AKS cluster with additional services.' It includes sections for 'Azure Container Registry' (selected container registry 't92022630278acr'), 'Service mesh - Istio' (checkbox checked), 'Azure Policy' (radio button set to 'Enabled'), and a note about Azure policy being recommended for dev/test configuration. Navigation buttons at the bottom are 'Previous', 'Next', and 'Review + create'.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Recommend VM sizes for Kubernetes Service'. It displays tabs for 'Basics', 'Node pools', 'Networking', 'Integrations', 'Monitoring', 'Security', 'Advanced', and 'Tags'. Under 'Basics', settings include: Subscription (Azure for Students), Resource group (t9-2022630278-rg), Region (Canada Central), Kubernetes cluster name (t9-2022630278-aks), Kubernetes version (1.33.5), Automatic upgrade (patch), Automatic upgrade scheduler (Every week on Sunday (recommended)), Node security channel type (NodeImage), and Security channel scheduler (Every week on Sunday (recommended)). A 'Node pools' section shows 1 pool. Navigation buttons at the bottom are 'Previous', 'Next', and 'Create'.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Node pools'. The configuration includes:

- Node pools**: 1
- Enable virtual nodes**: Disabled
- Node Auto-provisioning**: Disabled

**Access** settings:

- Resource identity**: System-assigned managed identity
- Local accounts**: Enabled
- Authentication and Authorization**: Local accounts with Kubernetes RBAC
- Encryption type**: (Default) Encryption at-rest with a platform-managed key

**Networking** settings:

- Private cluster**: Disabled
- Authorized IP ranges**: Disabled
- Network configuration**: Azure CNI Overlay
- DNS name prefix**: t9-2022630278-aks-dns
- Network policy engine**: None
- Load balancer**: Standard

At the bottom, there are 'Previous', 'Next', and 'Create' buttons. The 'Create' button is highlighted in blue.

The screenshot shows the 'Create Kubernetes cluster' wizard on the Azure portal. The current step is 'Integrations'. The configuration includes:

- Container registry resource group**: t9-2022630278-rg
- Container registry**: t9022630278acr
- Service mesh**: Disabled
- Azure Policy**: Disabled

**Monitoring** settings:

- Enable Container Logs**: Disabled
- Enable Prometheus metrics**: Disabled
- Enable Grafana**: Disabled
- Alerts**: Disabled

**Advanced** settings:

- Infrastructure resource group**: MC\_t9-2022630278-rg\_t9-2022630278-aks\_canadacentral

**Security** settings:

- Microsoft Defender for Cloud**: Free
- OpenID Connect (OIDC)**: Enabled

At the bottom, there are 'Previous', 'Next', and 'Create' buttons. The 'Create' button is highlighted in blue.

The image consists of two screenshots of the Microsoft Azure portal interface.

**Screenshot 1: Deployment Overview**

- Deployment Name:** microsoft.aks-1766362783723
- Subscription:** Azure for Students
- Resource Group:** t9-2022630278-rg
- Start Time:** 12/21/2025, 6:20:37 PM
- Correlation ID:** 09a2e8b6-0d1d-4dde-b98c-a31e34d543a

**Cost Management:** Get notified to stay within your budget and prevent unexpected charges on your bill. [Set up cost alerts >](#)

**Microsoft Defender for Cloud:** Secure your apps and infrastructure. [Go to Microsoft Defender for Cloud >](#)

**Free Microsoft tutorials:** Start learning today >

**Work with an expert:** Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. [Find an Azure expert >](#)

**Screenshot 2: AKS Cluster Overview**

- Cluster Name:** t9-2022630278-aks
- Resource Group:** t9-2022630278-rg
- Power State:** Running
- Cluster Operation Status:** Succeeded
- Subscription:** Azure for Students
- Location:** Canada Central
- Subscription ID:** fd58a3da-fcef-47d1-ac0e-5b891faa4251
- Fleet Manager:** Click here to assign
- Kubernetes Version:** 1.33.5
- API Server Address:** t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io
- SKU:** Base
- Pricing Tier:** Free
- Network Configuration:** Azure CNI Overlay
- Node Pools:** 1 node pool
- Container Registries:** t9-2022630278acr
- Created Time:** December 21, 2025 at 06:20 PM

**Figura 7. Clúster AKS operativo con vista de nodos y workloads del proyecto T9**

## 6.5 MySQL Flexible Server (PaaS)

Se realizó la provisión de la instancia MySQL Flexible Server con el nombre oficial t9-2022630278-bd. Se accedió al panel de bases de datos para crear y validar las tres bases de datos:

- t9\_2022630278\_bdgu para Gestión de usuarios (tablas usuarios y fotos\_usuarios).
- t9\_2022630278\_bdga para Gestión de artículos (tablas stock y fotos\_articulos).
- t9\_2022630278\_bdgc para Gestión de compras (tablas stock con índice único y carrito\_compra con índice único).

Se instaló el esquema SQL en cada base y se verificó la conectividad utilizando las credenciales definidas para los microservicios. Se accedió a la sección de “Connection strings” para registrar los parámetros de servidor, usuario y contraseña en los secretos de Kubernetes.

Antes de la figura, debería verse la instancia t9-2022630278-bd con las tres bases creadas y los parámetros de conexión visibles; se recomienda capturar la lista de DB y el panel de cadenas de conexión.

The screenshot shows the Microsoft Azure portal's home page. At the top, there is a search bar with the query "Azure Database for MySQL Flexible Server". Below the search bar, there are sections for "Azure services" (Create a resource, Kubernetes services) and "Resources" (Recent, Favorite). The "Recent" tab is selected, displaying a list of resources: t9-2022630278-aks (Kubernetes service), t9-2022630278-rg (Resource group), t92022630278acr (Container registry), Azure for Students (Subscription), and DefaultWorkspace-fd58a3da-fcef-47d1-ac0e-5b891faa4251-ccAN (Log Analytics workspace). On the right side, there are links for "Copilot", "Route tables", "Virtual machines", and "More services". The bottom navigation bar includes links for Subscriptions, Resource groups, All resources, and Dashboard.

The screenshot shows the Microsoft Azure portal's "Azure Database for MySQL flexible servers" browse experience. The top navigation bar includes the search bar with the query "Azure Database for MySQL flexible servers", the "Home" link, and the user's email (ggarciaq1800@alumno.ipn.mx). The main content area displays a message: "No Azure Database for MySQL flexible servers to display". It explains that Azure Database for MySQL Flexible Server offers flexible deployment options with advanced customization features for optimized performance and cost efficiency. There is a "Create" button with a plus sign and a "Give feedback" link. The bottom navigation bar includes links for Subscriptions, Resource groups, All resources, and Dashboard, along with the system tray showing the date and time (06:27 p.m., 21/12/2025).

**Flexible server**

Did you know that new users in Azure can use MySQL - Flexible Server free for up to 750 hours using Azure free account? [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription: Azure for Students

Resource group: t9-2022630278-rg

**Server details**

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

**Estimated costs**

	USD 13.50/month
Compute	Free upto 750 hours Standard_B1ms (1 vCore)
Storage	Free upto 32 GB 20 GiB (USD 0.13 per GiB)
Auto scale IOPS	Auto scale IOPS is billed on usage in per million request increments. <a href="#">Learn more</a>
Backup Retention	Backup retention is billed based on additional storage used for retaining backups. <a href="#">Learn more</a>

**Flexible server**

Did you know that new users in Azure can use MySQL - Flexible Server free for up to 750 hours using Azure free account? [Learn more](#)

**Server details**

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name: t9-2022630278-bd

Region: Canada Central

MySQL version: 8.0

Workload type: Dev/Test

Compute + storage: Burstable, B1ms  
1 vCores, 2 GiB RAM, 20 GiB storage, Auto scale IOPS  
Geo-redundancy: Disabled  
[Configure server](#)

Availability zone: No preference

**Estimated costs**

	USD 13.50/month
Compute	Free upto 750 hours Standard_B1ms (1 vCore)
Storage	Free upto 32 GB 20 GiB (USD 0.13 per GiB)
Auto scale IOPS	Auto scale IOPS is billed on usage in per million request increments. <a href="#">Learn more</a>
Backup Retention	Backup retention is billed based on additional storage used for retaining backups. <a href="#">Learn more</a>

**High availability**

High availability provides additional server resilience in the event of a failure. You can also specify high availability options in 'Compute + storage'.

High availability (radio buttons):

- Zone-redundant (99.99% SLA) - a standby server is always available within another zone in the same region as the primary server
- Local-redundant (99.95% SLA) - a standby server is always available within the same zone as the primary server
- Disabled (99.9% SLA)**

**Authentication**

Select the authentication method you would like to support for accessing this MySQL server. Enabling MySQL password authentication allows you to authenticate with user names and passwords that are stored inside MySQL.

Enabling Microsoft Entra authentication allows you to create user names in MySQL, which are mapped to accounts stored in Microsoft Entra ID. Users or applications authenticated against Microsoft Entra ID, can retrieve tokens that are presented to MySQL as their corresponding time-limited password. [Learn more](#)

Authentication method (radio buttons):

- MySQL authentication only**
- Microsoft Entra authentication only
- MySQL and Microsoft Entra authentication

**Compute** **USD 13.50/month**

Free upto 750 hours  
Standard\_B1ms (1 vCore) 13.50

**Storage** **USD 2.54/month**

Free upto 32 GB  
20 GiB (USD 0.13 per GiB) 20 x 0.13

**Auto scale IOPS**

Auto scale IOPS is billed on usage in per million request increments. [Learn more](#)

**Backup Retention**

Backup retention is billed based on additional storage used for retaining backups. [Learn more](#)

**Bandwidth**

Outbound data transfer across services in different regions will incur additional charges. Any inbound data transfer is free. [Learn more](#)

**Review + create** **Next : Networking >**

**Administrator login \***

**Password \***

**Confirm password \***

**Estimated total** **USD 16.04/month**

Charges will apply if you use above the free tier. [Learn more](#)

**>Password and confirm password matched.** [Clear](#)

**Review + create** **Next : Networking >**

**Flexible server**

Estimated Server Creation Time (in minutes): 5

**Product details**

Azure Database for MySQL by Microsoft

[Terms of use](#) | [Privacy policy](#)

**Basics (Change)**

Subscription	Azure for Students
Resource group	t9-2022630278-rg
Server name	t9-2022630278-bd
Administrator login	x
Location	Canada Central
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0

**Estimated costs**

Category	Description	Cost
Compute	USD 13.50/month	Free upto 750 hours Standard_B1ms (1 vCore) 13.50
Storage	USD 2.54/month	Free upto 32 GB 20 GiB (USD 0.13 per GiB) 20 x 0.13
Auto scale IOPS		Auto scale IOPS is billed on usage in per million request increments. <a href="#">Learn more</a>
Backup Retention		Backup retention is billed based on additional storage used for retaining backups. <a href="#">Learn more</a>
Bandwidth		

[Create](#) [+ Previous](#) [Download a template for automation](#)

**Networking (Change)**

Connectivity method	Public access (allowed IP addresses) and Private endpoint
Allow public access to this resource through the internet using a public IP address	Yes

**Basics (Change)**

Subscription	Azure for Students
Resource group	t9-2022630278-rg
Server name	t9-2022630278-bd
Administrator login	x
Location	Canada Central
Availability zone	No preference
High availability	Not enabled
MySQL version	8.0
Compute + storage	Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 storage, Auto scale IOPS
Backup retention period (in days)	7 day(s)
Storage autogrow	Enabled
Geo-redundancy	Not enabled
Zonal Resiliency	No

**Estimated costs**

Category	Description	Cost
Compute	USD 13.50/month	Free upto 750 hours Standard_B1ms (1 vCore) 13.50
Storage	USD 2.54/month	Free upto 32 GB 20 GiB (USD 0.13 per GiB) 20 x 0.13
Auto scale IOPS		Auto scale IOPS is billed on usage in per million request increments. <a href="#">Learn more</a>
Backup Retention		Backup retention is billed based on additional storage used for retaining backups. <a href="#">Learn more</a>
Bandwidth		

[Create](#) [+ Previous](#) [Download a template for automation](#)

**Flexible server**

**Storage (Change)**

- Storage account: **chaveus**
- Geo-redundancy: **Not enabled**
- Zonal Resiliency: **No**

**Networking (Change)**

- Connectivity method: **Public access (allowed IP addresses) and Private endpoint**
- Allow public access to this resource through the internet using a public IP address: **Yes**
- Allow public access from any Azure service within Azure to this server: **No**
- Firewall rules: **0**
- SSL/TLS: **SSL is enforced and TLS version is 1.2. This can be changed after server is created. Learn more**

**Additional configuration (Change)**

- Data encryption: **Service-managed key**
- Server Parameter - **lower\_case\_table\_names**: **1**

**Tags (Change)**

**Estimated total** **USD 16.04/month**

**Charges will apply if you use above the free monthly limits. Please check your usage of free services. Final charges will appear in your local currency.**

**Create** **< Previous** **Download a template for automation**

**t9-2022630278-bd - Microsoft Azure**

**Overview**

**Azure Database for MySQL - Live Webinar series:** Learn about the latest updates (with demos) and interact directly with product group on the 2nd Wednesday of every month! [Subscribe to our YouTube channel](#)

**Essentials**

Subscription (move)	: Azure for Students	Endpoint	: t9-2022630278-bd.mysql.database.azure.com
Subscription ID	: ffd58a3da-fcef-47d1-ac0e-5b891faa4251	Administrator login	: x
Resource group (move)	: t9-2022630278-rg	Configuration	: Burstable_B1ms_1 vCores, 2 GiB RAM, 20 storage, 360 IOPS
Status	: Ready	MySQL version	: 8.0 <a href="#">Upgrade</a>
Location	: Canada Central	Availability zone	: 3
Tags (edit)	: Add tags	Created on	: 2025-12-22 00:34:38.4538981 UTC

**Getting started** **Properties** **Recommendations** **Monitoring** **Tutorials**

We've prepared a checklist to get you started

**Get ready for migration**

**Figura 8. MySQL Flexible Server “t9-2022630278-bd” con bases  
t9\_2022630278\_bdgu, t9\_2022630278\_bdga y t9\_2022630278\_bdgc**

## Creación de bases de datos

The screenshot shows the Microsoft Azure portal interface. On the left, there is a sidebar with various service icons and a 'Databases' section selected. The main content area displays a table of existing databases (mysql, information\_schema, performance\_schema, sys) and a 'Create database' dialog box. The dialog box has fields for 'Name' (t9\_2022630278\_bdgu), 'Character set' (utf8), and 'Collation' (utf8\_general\_ci). A 'Save' button is visible at the top right of the dialog.

This screenshot is nearly identical to the one above, showing the same Azure portal interface and 'Create database' dialog. The only difference is the 'Name' field in the dialog, which now contains 't9\_2022630278\_bdgi'. The rest of the interface, including the sidebar and the list of existing databases, remains the same.

**Create database**

Name:

Character set:

Collation:

Name	Character set	Collation	Schema type
mysql	utf8mb4	utf8mb4_0900...	System
information_schema	utf8mb3	utf8mb3_gener...	System
performance_schema	utf8mb4	utf8mb4_0900...	System
sys	utf8mb4	utf8mb4_0900...	System
t9_2022630278_bdgu	utf8mb3	utf8mb3_gener...	User
t9_2022630278_bdga	utf8mb3	utf8mb3_gener...	User

**Successfully created MySQL database.**

Successfully created 't9\_2022630278\_bdgc' MySQL database.

Name	Character set	Collation	Schema type
information_schema	utf8mb3	utf8mb3_gener...	System
mysql	utf8mb4	utf8mb4_0900...	System
performance_schema	utf8mb4	utf8mb4_0900...	System
sys	utf8mb4	utf8mb4_0900...	System
t9_2022630278_bdga	utf8mb3	utf8mb3_gener...	User
t9_2022630278_bdgc	utf8mb3	utf8mb3_gener...	User
t9_2022630278_bdgu	utf8mb3	utf8mb3_gener...	User

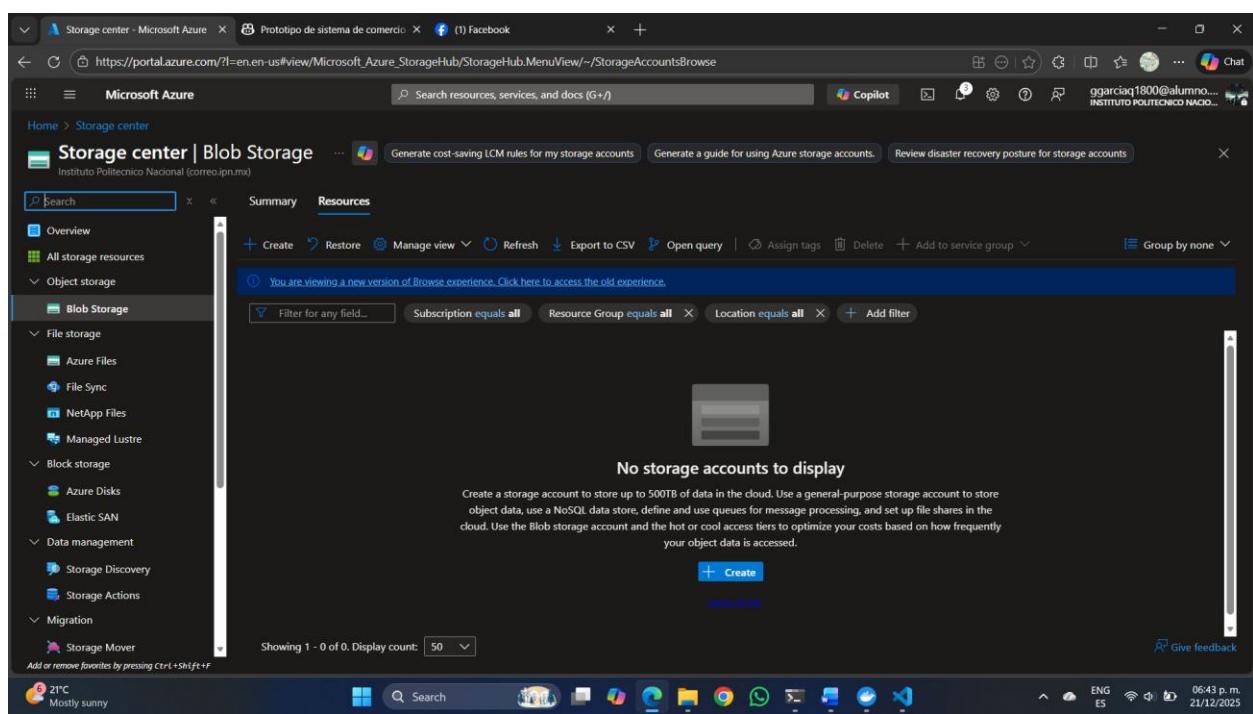
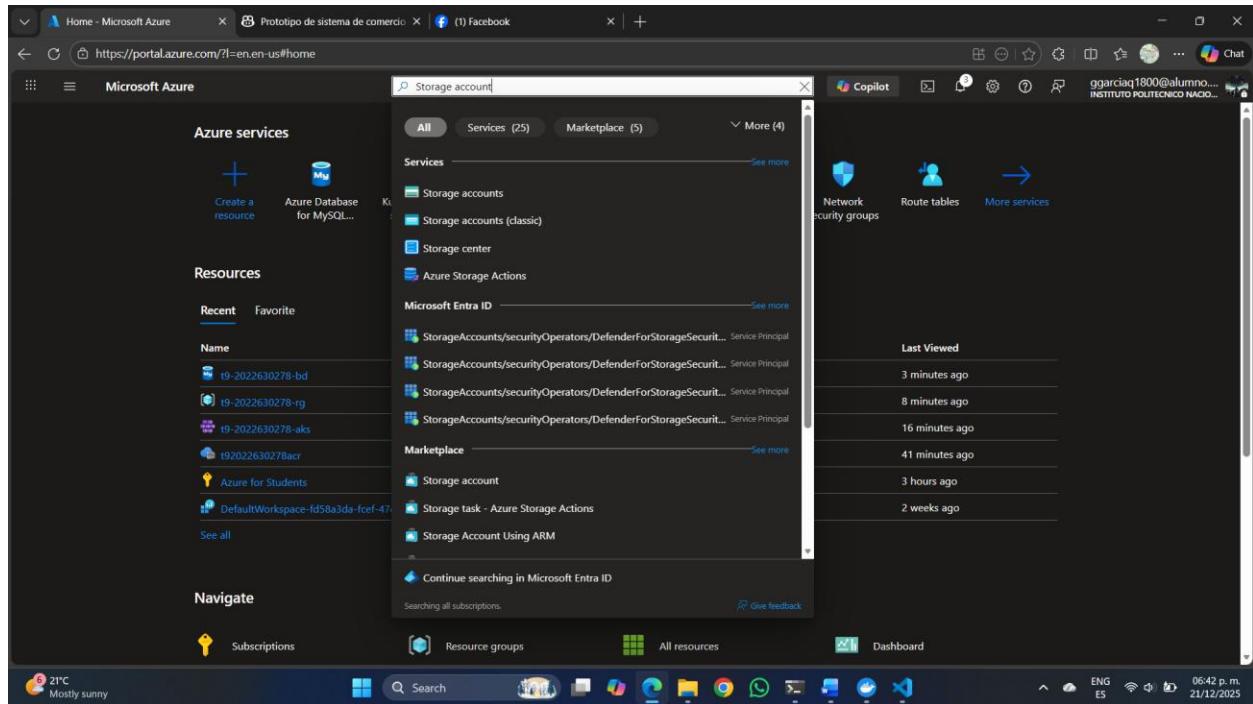
**Figura 8. MySQL Flexible Server “t9-2022630278-bd” con bases  
t9\_2022630278\_bdgu, t9\_2022630278\_bdga y t9\_2022630278\_bdgc**

## 6.6 Storage Account y Azure Files (front-end)

Se realizó la creación del Storage Account y del file share destinado a servir el front-end a través del microservicio de Servidor Web (Get.cs). Se accedió al file share para subir los archivos: prueba.html, WSClient.js y la imagen base del usuario. Se instaló el PV/PVC en Kubernetes y se configuró el mount en el pod del servicio SW, utilizando la variable de entorno ROOT para apuntar a la ruta montada.

- Se accedió al recurso de almacenamiento y se verificó el nombre, región y file share t9webfiles.
- Se instalaron los archivos del front-end en el file share, listándolos para corroborar disponibilidad.

Antes de la figura, debería verse el Storage Account con el file share seleccionado y los archivos del front-end visibles; la captura debe incluir nombre del recurso y la región.



The screenshot shows the 'Create a storage account' wizard on the 'Basics' tab. The page title is 'Create a storage account ...'. Below it, a sub-header reads: 'Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.' A link 'Learn more about Azure storage accounts' is provided.

**Project details**

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

**Subscription:** Azure for Students

**Resource group:** t9-2022630278-rg  
Create new

**Instance details**

**Storage account name:** t92022630278sa

**Region:** (Canada) Canada Central  
Deploy to an Azure Extended Zone

**Preferred storage type:** Azure Files

A small note at the bottom states: 'This helps us provide relevant guidance. It doesn't restrict your storage to this resource type.' Below the form are 'Previous', 'Next', and 'Review + create' buttons. The status bar at the bottom right shows the date and time: 06:48 p.m. 21/12/2025.

The screenshot shows the 'Create a storage account' wizard on the 'Instance details' step. The page title is 'Create a storage account ...'. The 'Storage account name' field contains 't92022630278sa'. The 'Region' dropdown is set to '(Canada) Canada Central' with the option 'Deploy to an Azure Extended Zone' below it. The 'Preferred storage type' dropdown is set to 'Azure Files'.

**Performance**

**Standard:** Recommended for general purpose file shares and cost sensitive applications, such as HDD file shares

**Premium:** Recommended for application requiring low-latency or high IOPS/throughput, such as SSD file shares

**File share billing**

**Pay-as-you-go file shares:** Billed based on usage

**Provisioned v2:** Provision capacity, throughput, and IOPS individually (new)

**Redundancy**

Locally-redundant storage (LRS)

Below the form are 'Previous', 'Next', and 'Review + create' buttons. The status bar at the bottom right shows the date and time: 06:48 p.m. 21/12/2025.

The screenshot shows the 'Create a storage account' review step in the Azure portal. The 'Review + create' tab is selected. The configuration includes:

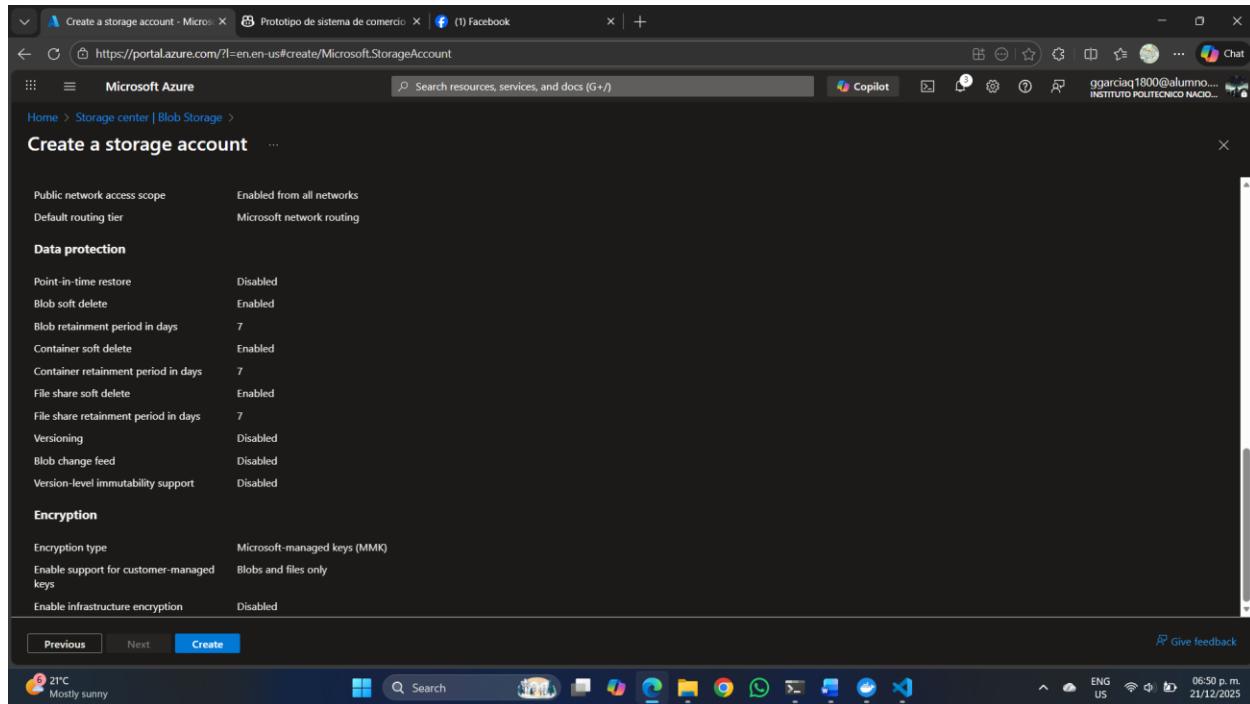
Setting	Value
Subscription	Azure for Students
Resource group	t9-2022630278-rg
Location	Canada Central
Storage account name	t92022630278sa
Preferred storage type	Azure Files
Performance	Standard
File share billing	Pay-as-you-go file shares
Replication	Locally-redundant storage (LRS)
Access tier	Hot

Below the table, there are 'Previous', 'Next', and 'Create' buttons. The status bar at the bottom shows the date and time as 06:49 p.m. 21/12/2025.

The screenshot shows the 'Create a storage account' review step in the Azure portal. The 'Review + create' tab is selected. The configuration includes:

Setting	Value
Access tier	Hot
Enable large file shares	Enabled
Secure transfer	Enabled
Blob anonymous access	Enabled
Allow storage account key access	Enabled
Default to Microsoft Entra authorization in the Azure portal	Disabled
Minimum TLS version	Version 1.2
Permitted scope for copy operations (preview)	From any storage account
Public network access	Enabled
Public network access scope	Enabled from all networks
Default routing tier	Microsoft network routing
Point-in-time restore	Disabled
blob soft delete	Enabled

Below the table, there are 'Previous', 'Next', and 'Create' buttons. The status bar at the bottom shows the date and time as 06:50 p.m. 21/12/2025.



**Essentials**

Resource group (move)	: t9-2022630278-rg	Performance	: Standard
Location	: canadacentral	Replication	: Locally-redundant storage (LRS)
Subscription (move)	: Azure for Students	Account kind	: StorageV2 (general purpose v2)
Subscription ID	: fd58a3da-fcef-47d1-ac0e-5b891faa4251	Provisioning state	: Succeeded
Disk state	: Available	Created	: 12/21/2025, 6:51:09 PM

**Properties**

Monitoring	Capabilities (7)	Recommendations (0)	Tutorials	Tools + SDKs
------------	------------------	---------------------	-----------	--------------

**Blob service**

Hierarchical namespace	Disabled	Require secure transfer for REST API operations	Enabled
Default access tier	Hot	Storage account key access	Enabled
Blob anonymous access	Enabled	Minimum TLS version	Version 1.2
Blob soft delete	Enabled (7 days)	Infrastructure encryption	Disabled
Container soft delete	Enabled (7 days)		
Versioning	Disabled		
Change feed	Disabled		
NFS v3	Disabled		
Allow cross-tenant replication	Disabled		

**Security**

Public network access	Enabled
Public network access scope	Enable from all networks

**Networking**

Public network access	Enabled
Public network access scope	Enable from all networks

Deployment status: Deployment succeeded. Deployment 't92022630278sa\_1766364592213' to resource group 't9-2022630278-rg' was successful.

**Figura 9. Storage Account**

This screenshot shows the 'File shares' settings page for a storage account named 't92022630278sa'. The left sidebar lists various storage management options like Overview, Activity log, Tags, and Data migration. Under 'Data storage', 'File shares' is selected. The main pane displays 'File share settings' with the following details:

- Identity-based access: Not configured
- Default share-level permissions: Disabled
- Soft delete: 7 days
- Maximum capacity: 100 TiB
- Security: Maximum compatibility

A search bar at the top allows searching by prefix (case-sensitive). A toggle switch 'Show deleted shares' is also present. Below the settings, a table lists file shares with columns for Name, Modified, Access tier, and Quota. A message indicates there are no file shares yet.

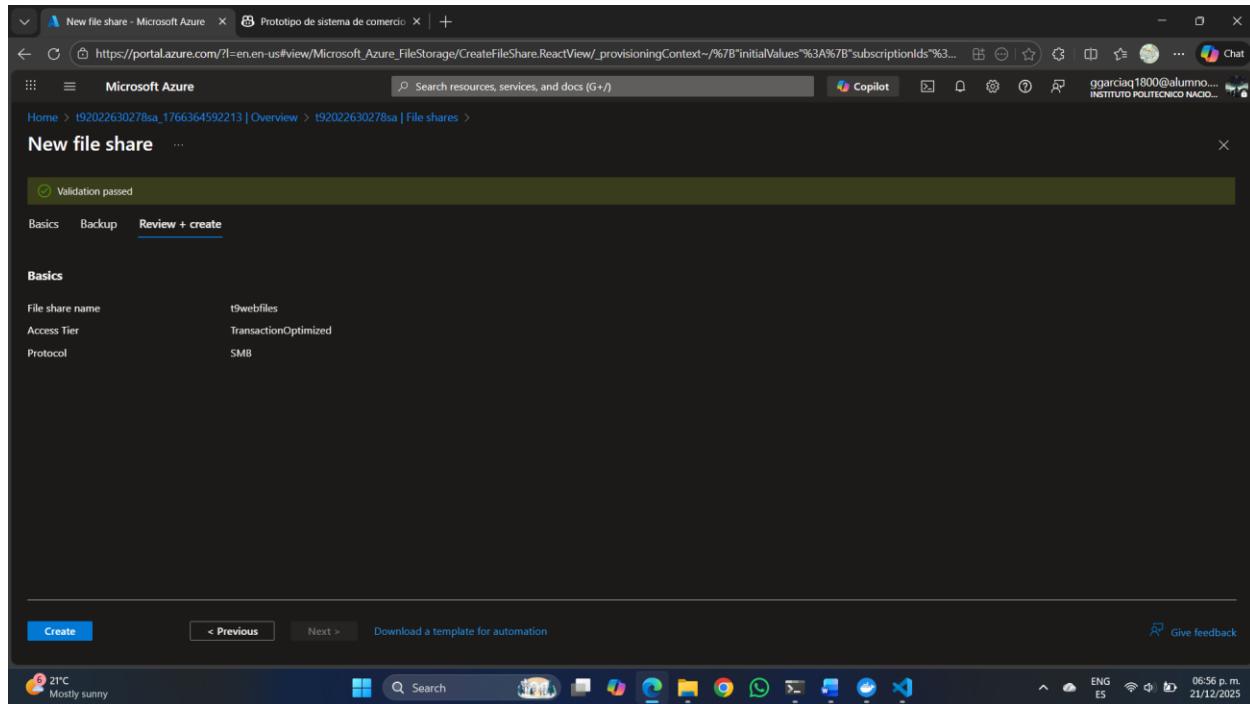
This screenshot shows the 'New file share' creation wizard. The 'Basics' step is active, showing the following configuration:

- Name: t9webfiles
- Access tier: Transaction optimized

The 'Performance' section includes:

- Maximum IO/s: 20000
- Maximum capacity: 100 TiB

A note at the bottom states: "To use the SMB protocol with this share, check if you can communicate over port 445. These scripts for Windows clients and Linux clients can help. Learn how to circumvent port 445 issues." Navigation buttons at the bottom include 'Review + create', '< Previous', 'Next : Backup >', and 'Give feedback'.



t9webfiles - Microsoft Azure

Protótipo de sistema de comercio

Microsoft Azure

File share name: t9webfiles

Storage account: t92022630278sa

Resource group (move): t9-2022630278-rg

Location: Canada Central

Subscription (move): Azure for Students

Subscription ID: fd58a3da-fcef-47d1-ac0e-5b891faa4251

Share URL: https://t92022630278sa.file.core.windows.net/t9webfiles

Redundancy: Locally-redundant storage (LRS)

Configuration modified: 12/21/2025, 6:56:51 PM

Properties    Capabilities (2)    Tutorials

Size

Maximum storage (GiB): 102400  
Used storage capacity (GiB): 0  
Access tier: Transaction optimized

Performance

IOPS: Varies by region. Learn more  
Throughput (MiB/sec): Varies by region. Learn more

Feature status

Soft delete: 7 days  
Large file shares: Enabled

Identity-based access

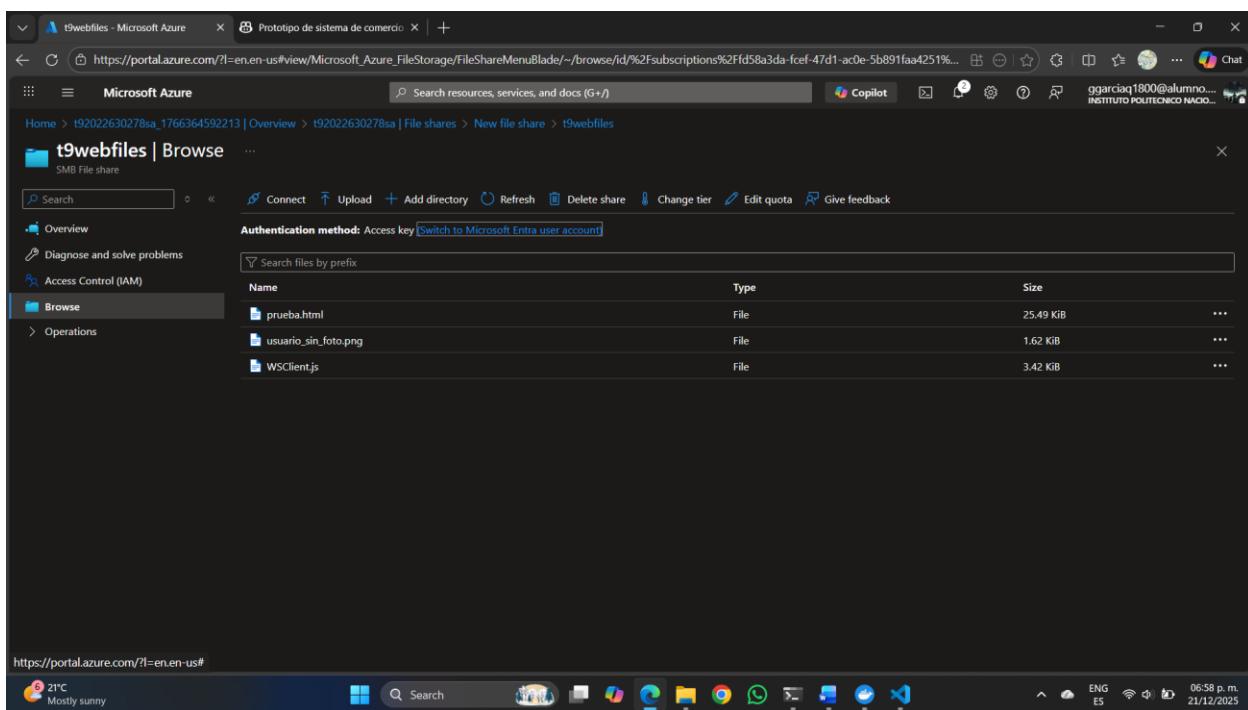
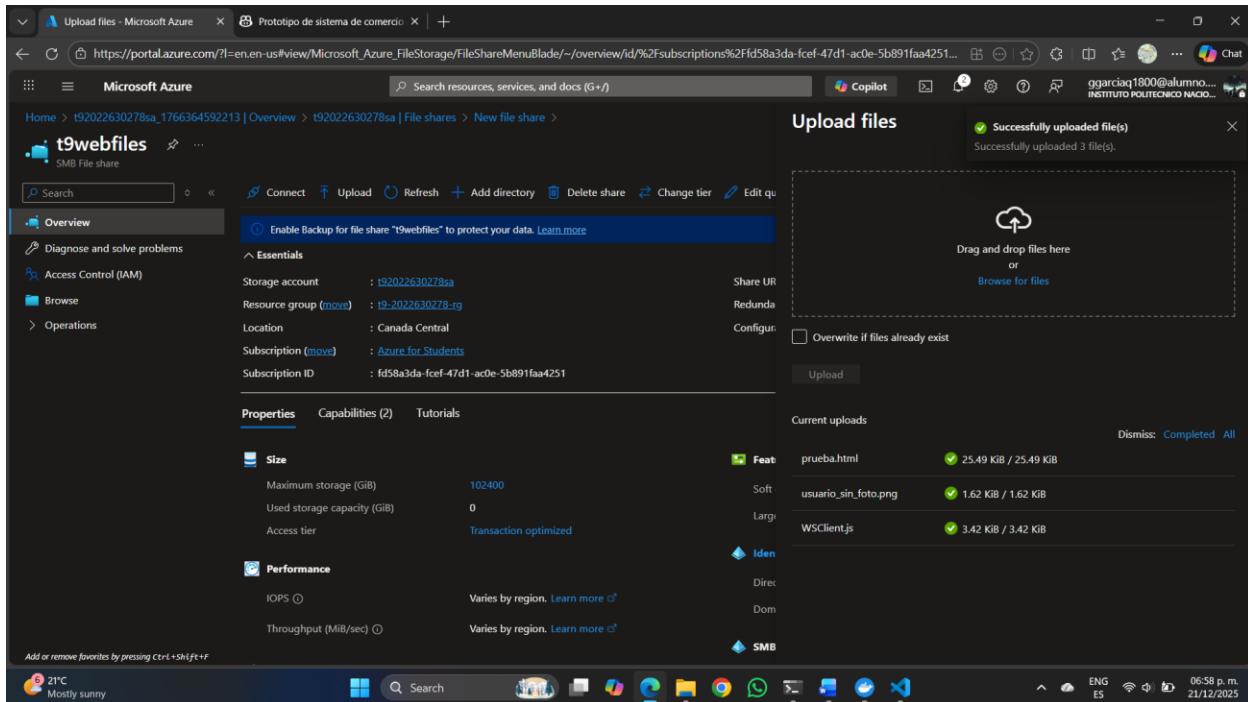
Directory service: Not configured  
Domain: -

SMB protocol settings

Add or remove favorites by pressing Ctrl+Shift+F.

21°C Mostly sunny

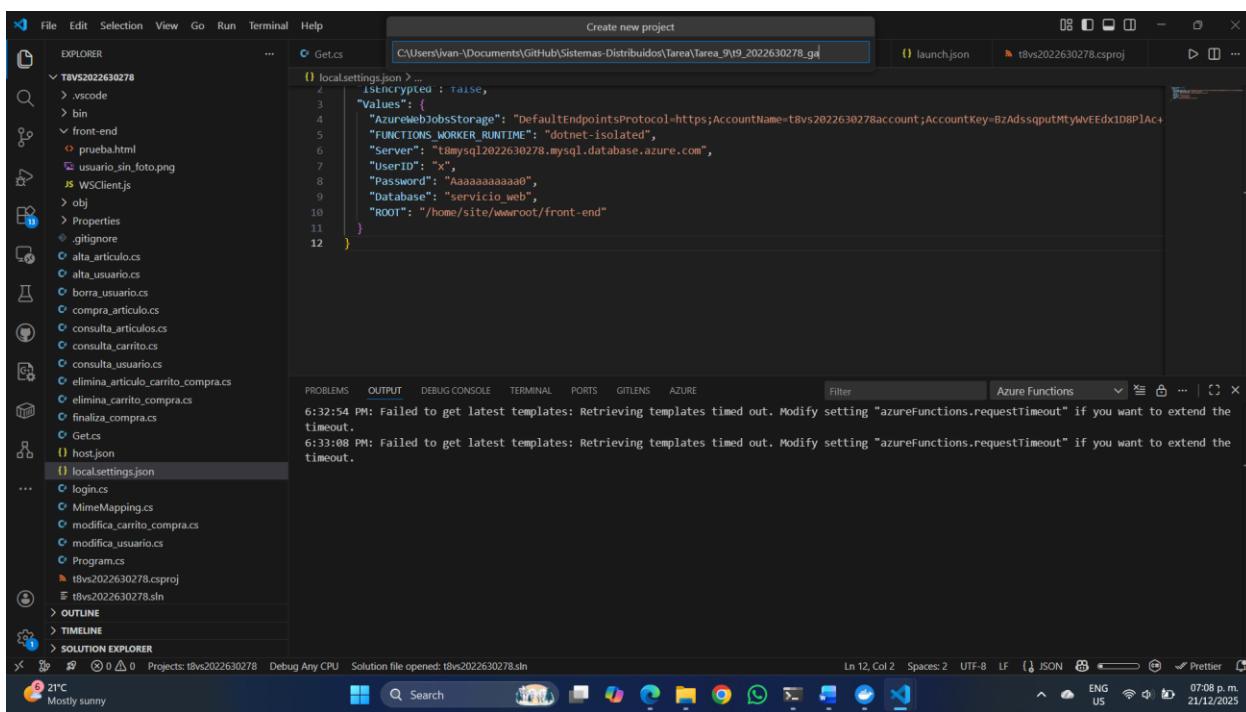
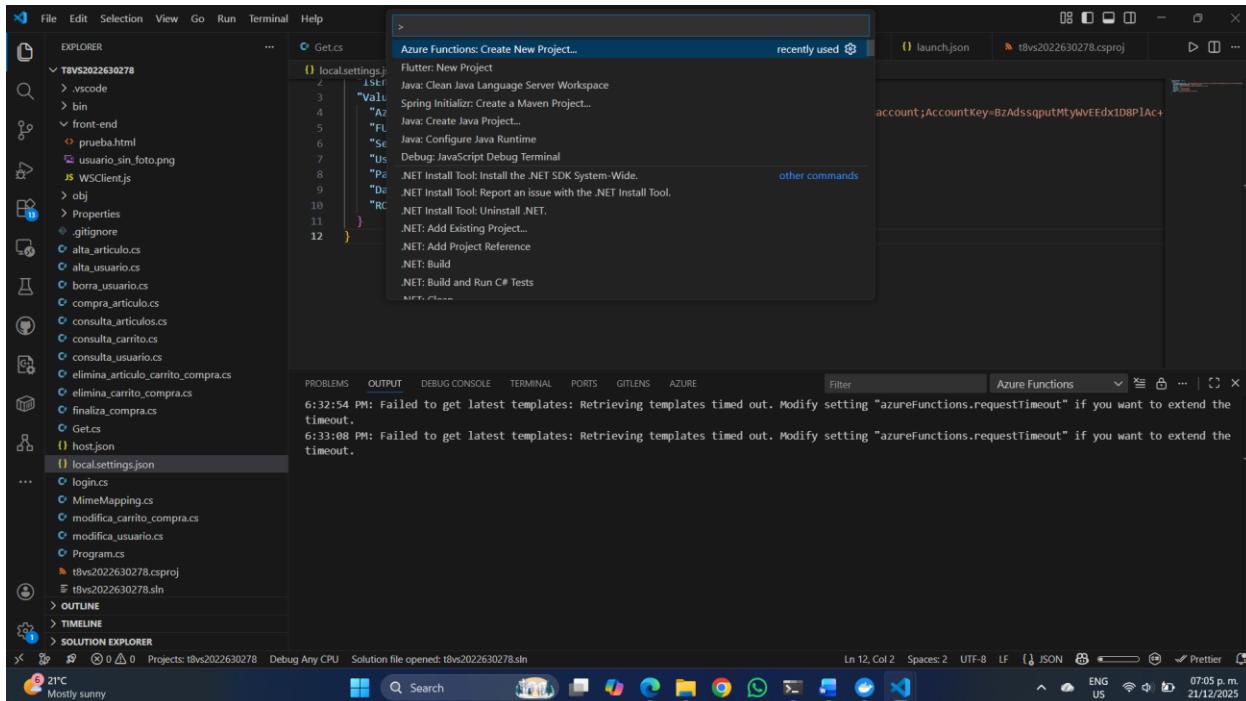
06:56 p.m. 21/12/2025

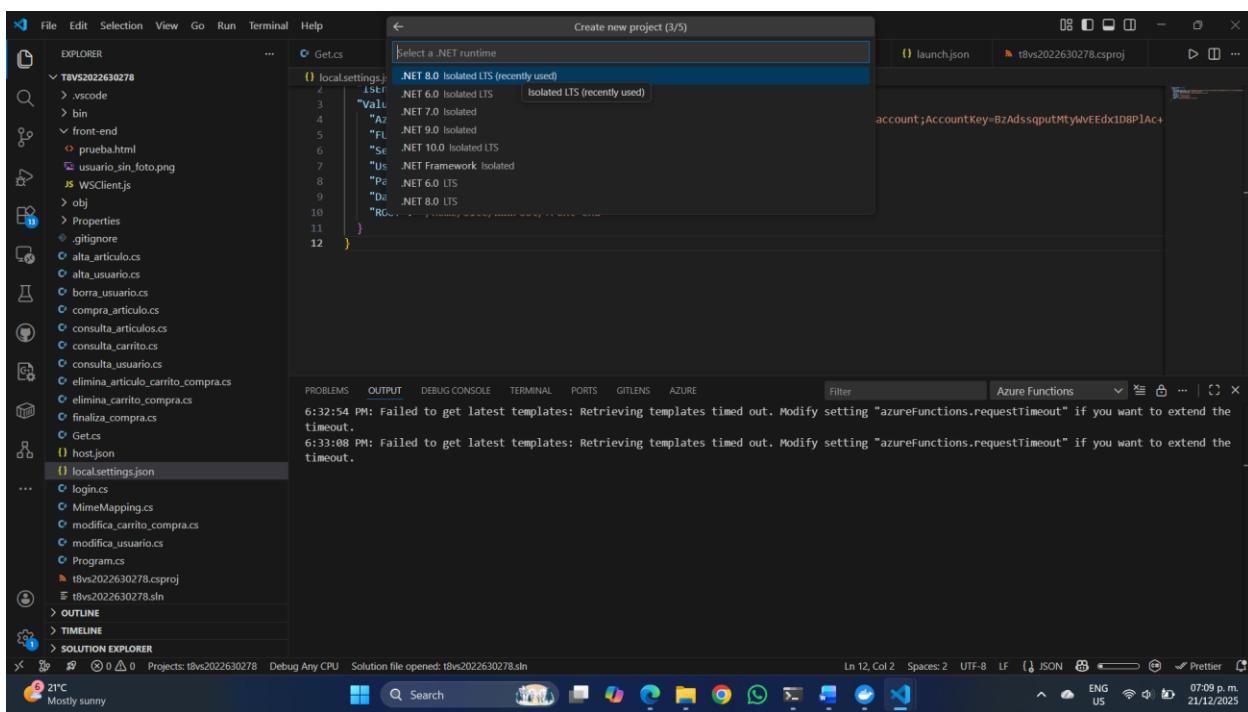
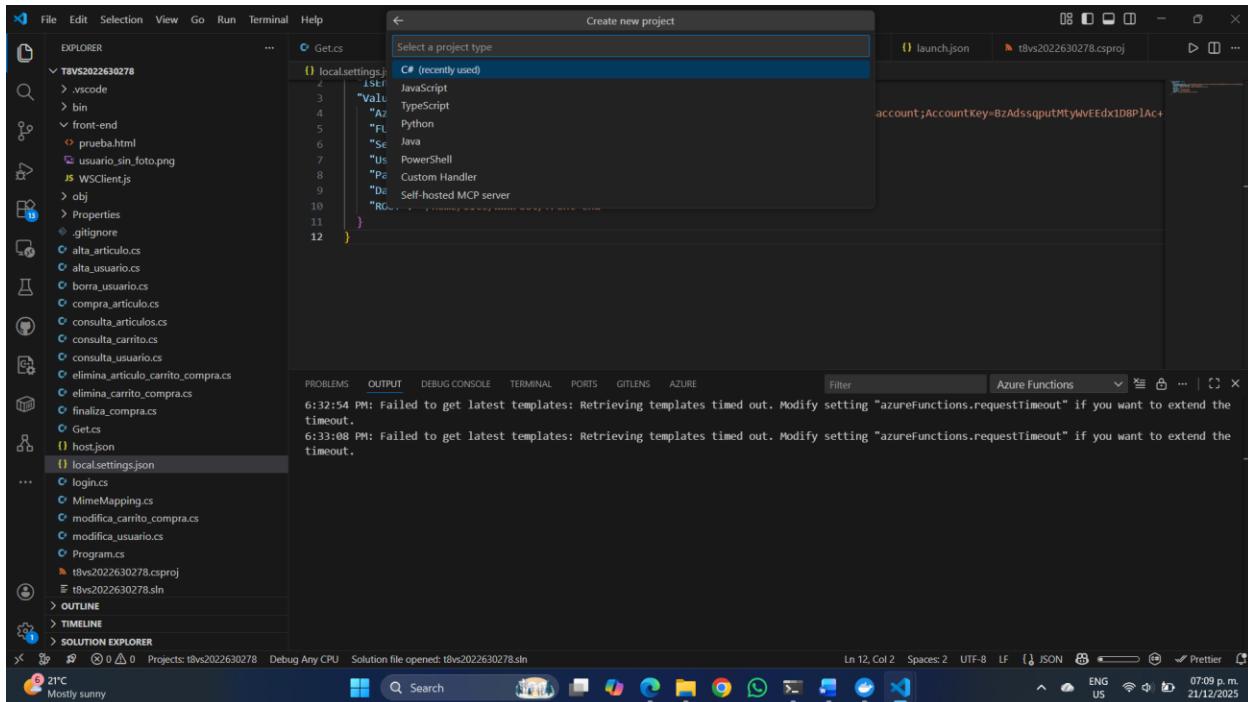


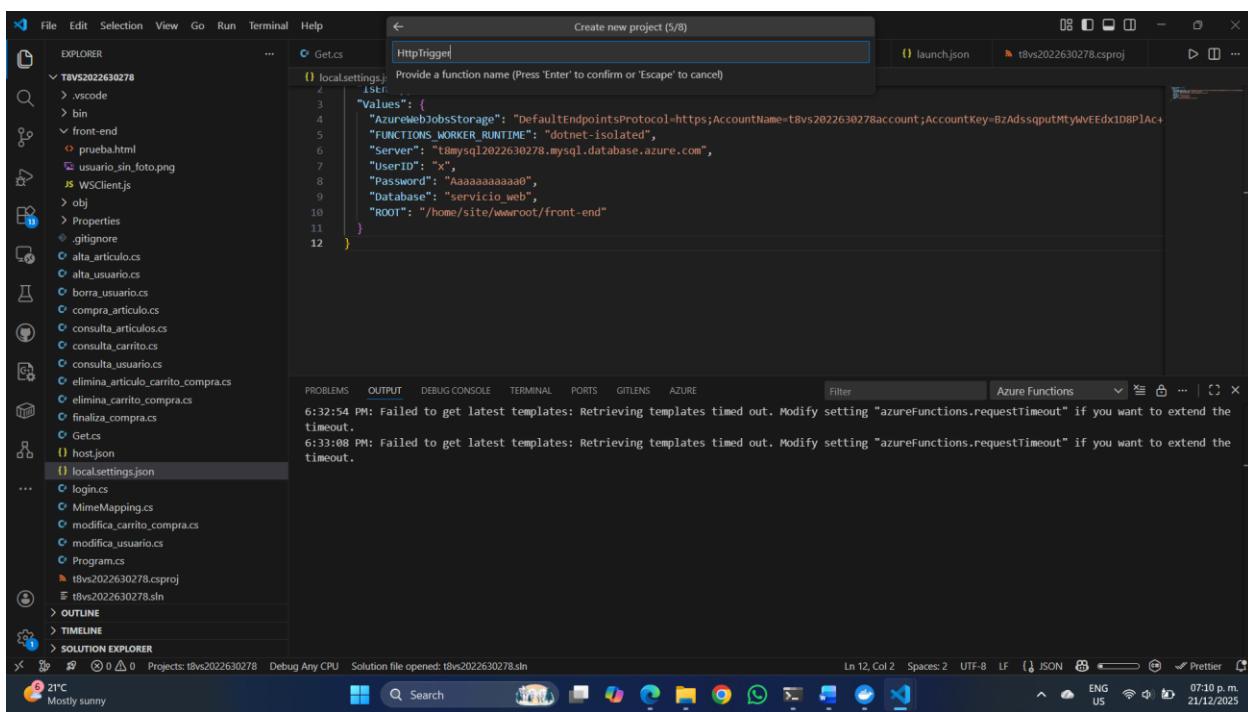
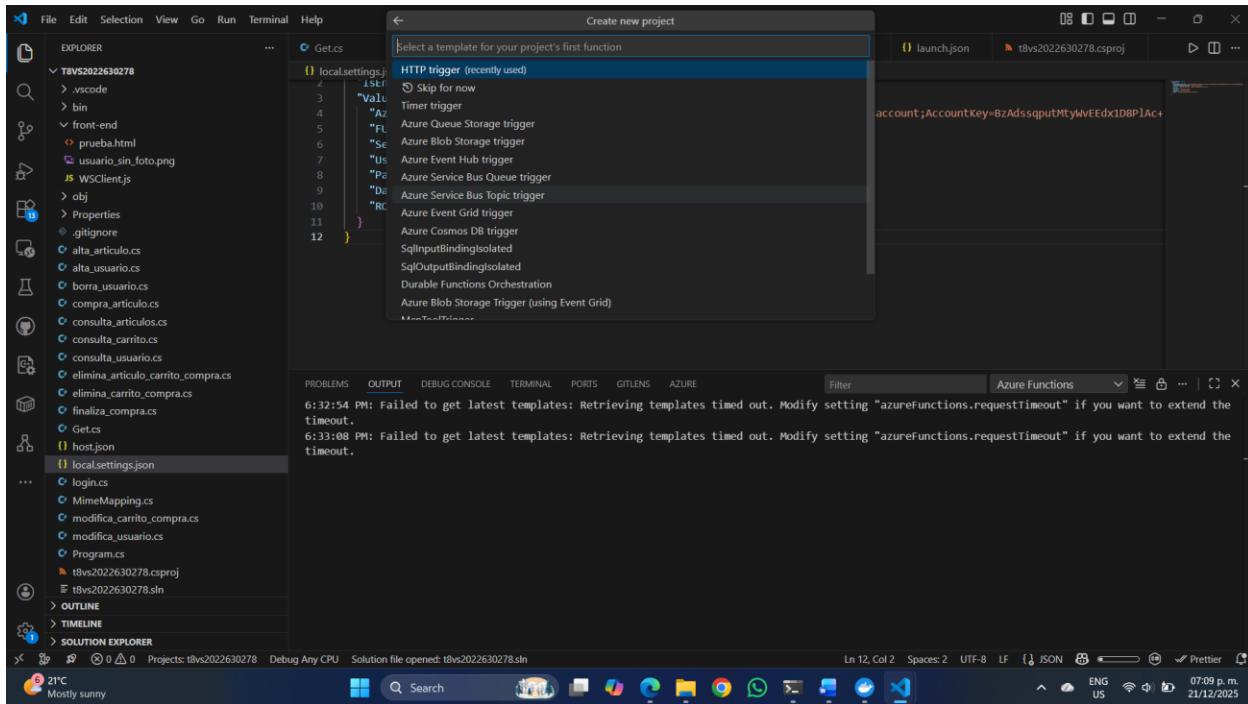
**Figura 9. Azure Files “t9webfiles” contenido el front-end (prueba.html, WSClient.js)**

## 6.7 Estructurar los microservicios en VS Code Crea cuatro proyectos Azure Functions .NET 8 aislado

Se realizó la creación de cuatro proyectos Azure Functions







```
local.settings.json
1  {
2      "IsEncrypted": false,
3      "Values": {
4          "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=t8vs2022630278account;AccountKey=BzAdssqputMtyWVEEdx1D8PlAc+",
5          "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
6          "Server": "t8mysql2022630278.mysql.database.azure.com",
7          "UserID": "x",
8          "Password": "aaaaaaaaaa0",
9          "Database": "servicio_web",
10         "ROOT": "/home/site/wwwroot/front-end"
11     }
12 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS AZURE Filter Azure Functions

6:32:54 PM: Failed to get latest templates: Retrieving templates timed out. Modify setting "azureFunctions.requestTimeout" if you want to extend the timeout.  
6:33:08 PM: Failed to get latest templates: Retrieving templates timed out. Modify setting "azureFunctions.requestTimeout" if you want to extend the timeout.

SOLUTION EXPLORER

LN 12, COL 2 SPACES: 2 UTF-8 LF JSON PRETTIER

21°C Mostly sunny

```
local.settings.json
1  {
2      "IsEncrypted": false,
3      "Values": {
4          "AccountName": "t8vs2022630278",
5          "AccountKey": "BzAdssqputMtyWVEEdx1D8PlAc+",
6          "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
7          "Server": "t8mysql2022630278.mysql.database.azure.com",
8          "UserID": "x",
9          "Password": "aaaaaaaaaa0",
10         "Database": "servicio_web",
11         "ROOT": "/home/site/wwwroot/front-end"
12     }
13 }
```

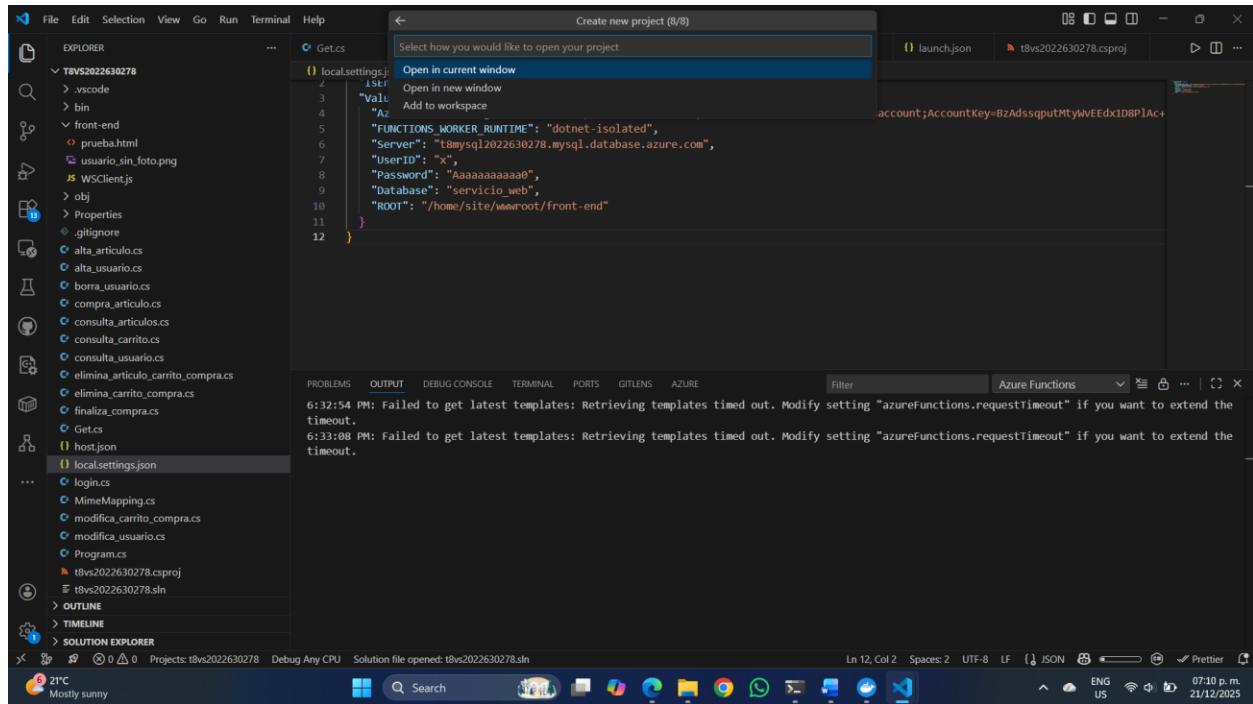
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS AZURE Filter Azure Functions

6:32:54 PM: Failed to get latest templates: Retrieving templates timed out. Modify setting "azureFunctions.requestTimeout" if you want to extend the timeout.  
6:33:08 PM: Failed to get latest templates: Retrieving templates timed out. Modify setting "azureFunctions.requestTimeout" if you want to extend the timeout.

SOLUTION EXPLORER

LN 12, COL 2 SPACES: 2 UTF-8 LF JSON PRETTIER

21°C Mostly sunny

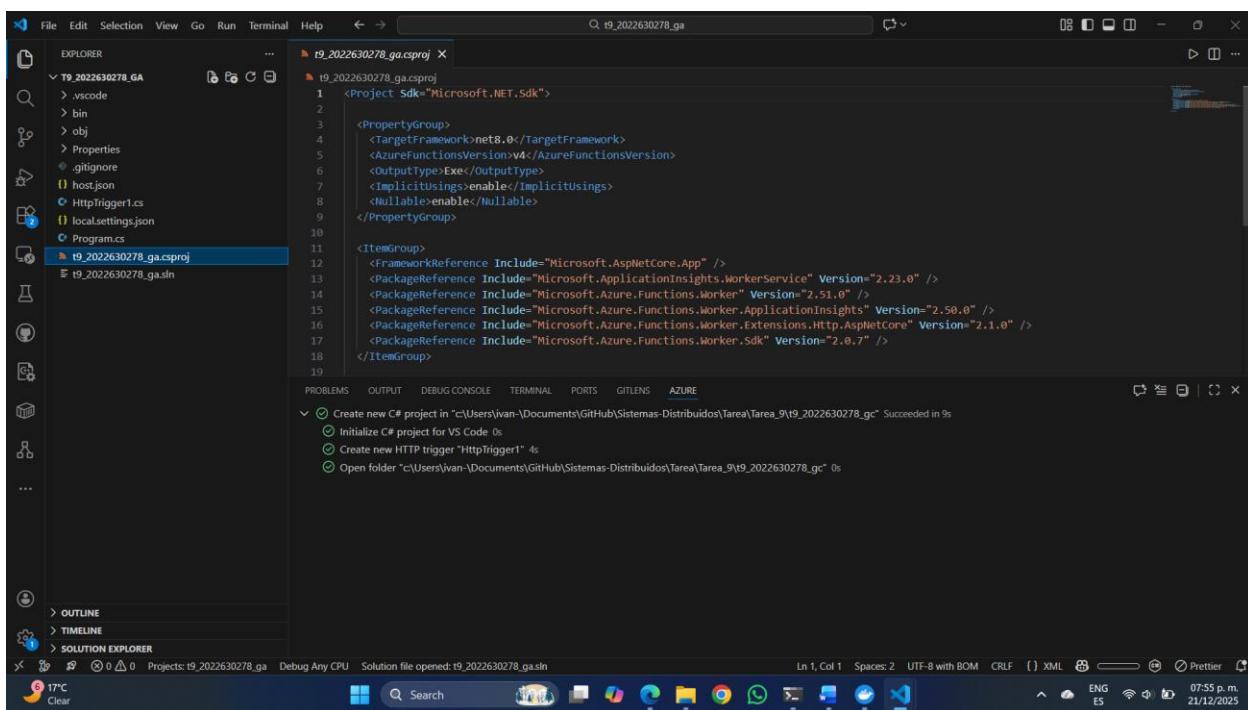
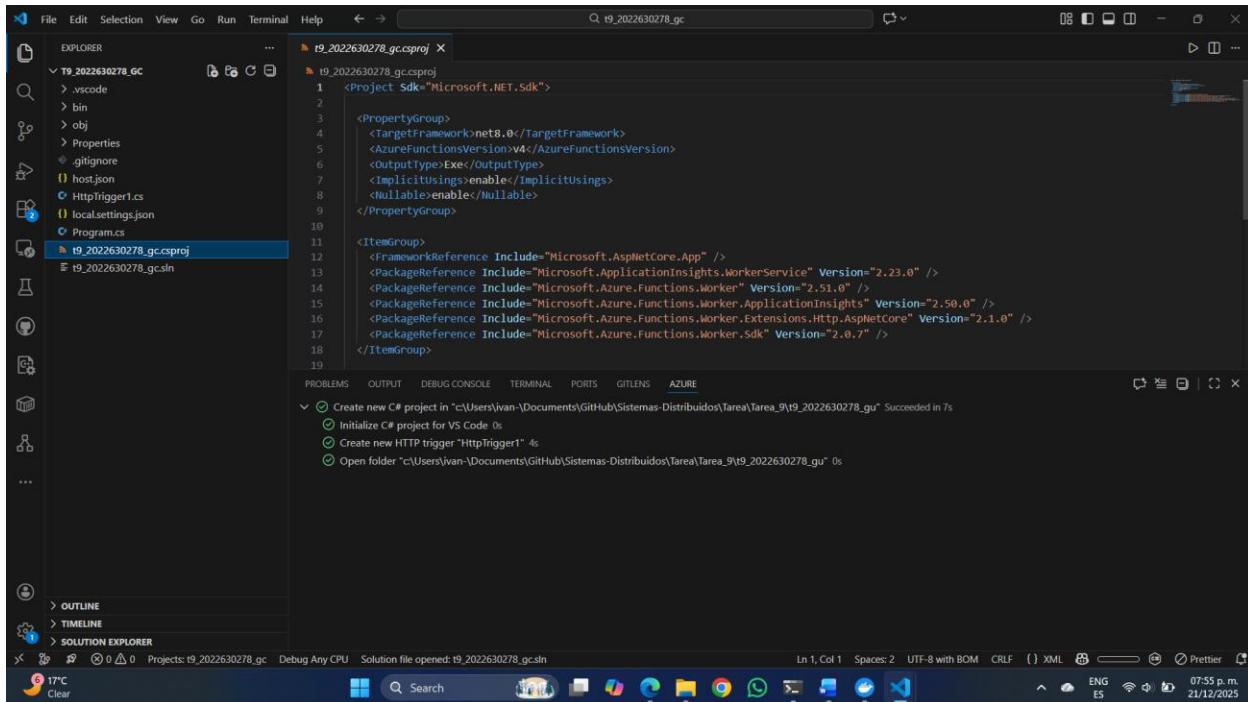


**Figura 9.1 Creacion del primer proyecto**

Se repiten los pasos para la creacion de los cuatro proyectos de visual studio code

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <AzureFunctionsVersion>v4</AzureFunctionsVersion>
    <OutputType>Exe</OutputType>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
  <ItemGroup>
    <FrameworkReference Include="Microsoft.AspNetCore.App" />
    <PackageReference Include="Microsoft.ApplicationInsights.WorkerService" Version="2.23.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker" Version="2.51.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.ApplicationInsights" Version="2.50.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.Extensions.Http.AspNetCore" Version="2.1.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.Sdk" Version="2.0.7" />
  </ItemGroup>
</Project>
```

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <AzureFunctionsVersion>v4</AzureFunctionsVersion>
    <OutputType>Exe</OutputType>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
  <ItemGroup>
    <FrameworkReference Include="Microsoft.AspNetCore.App" />
    <PackageReference Include="Microsoft.ApplicationInsights.WorkerService" Version="2.23.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker" Version="2.51.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.ApplicationInsights" Version="2.50.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.Extensions.Http.AspNetCore" Version="2.1.0" />
    <PackageReference Include="Microsoft.Azure.Functions.Worker.Sdk" Version="2.0.7" />
  </ItemGroup>
</Project>
```



**Figura 9.2 Proyecto creado**

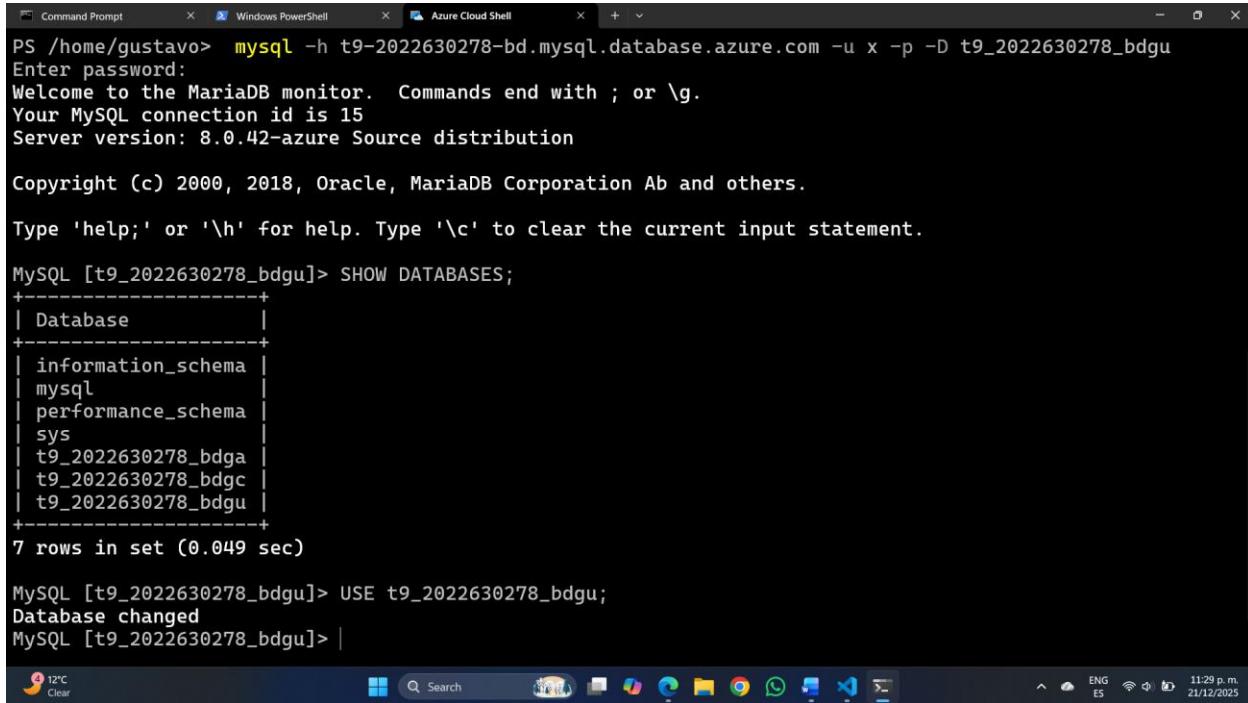
## 6.8 Creación de los esquemas SQL de cada microservicio

Conéctate con el comando:

```
mysql -h t9-2022630278-bd.mysql.database.azure.com -P 3306 -u x -p -D t9_2022630278_bdgu
```

Seleccionamos la base:

```
USE t9_2022630278_bdgu;
```



```
PS /home/gustavo> mysql -h t9-2022630278-bd.mysql.database.azure.com -u x -p -D t9_2022630278_bdgu
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.42-azure Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [t9_2022630278_bdgu]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| t9_2022630278_bdga |
| t9_2022630278_bdgc |
| t9_2022630278_bdgu |
+-----+
7 rows in set (0.049 sec)

MySQL [t9_2022630278_bdgu]> USE t9_2022630278_bdgu;
Database changed
MySQL [t9_2022630278_bdgu]> |
```

**Figura 9.3 Base de datos “t9\_2022630278\_bdgu” seleccionado**

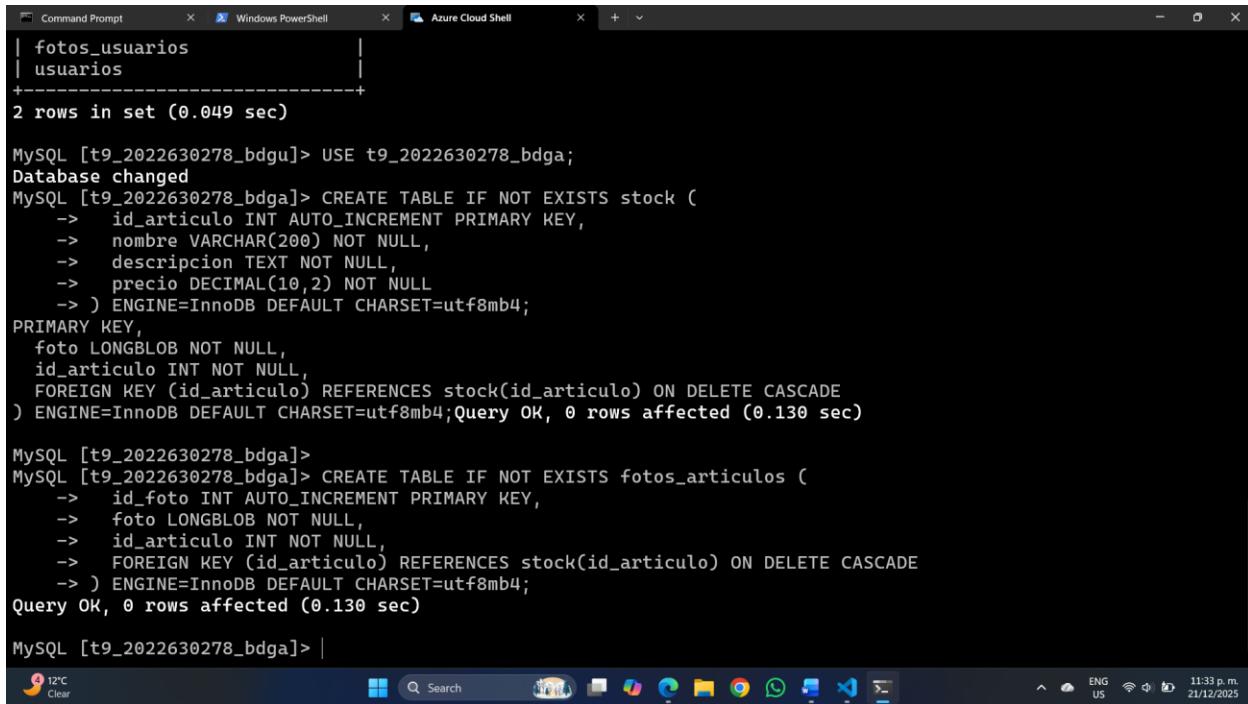
Ejecutamos las sentencias de creación (repite el patrón para cada base).

- Para GU:

```
+-----+  
7 rows in set (0.049 sec)  
  
MySQL [t9_2022630278_bdgu]> USE t9_2022630278_bdgu;  
Database changed  
MySQL [t9_2022630278_bdgu]> CREATE TABLE IF NOT EXISTS usuarios (  
->     id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
->     email VARCHAR(255) UNIQUE NOT NULL,  
->     password VARCHAR(128) 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),  
->     token VARCHAR(64)  
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
Query OK, 0 rows affected (0.154 sec)  
  
MySQL [t9_2022630278_bdgu]>  
MySQL [t9_2022630278_bdgu]> CREATE TABLE IF NOT EXISTS fotos_usuarios (  
->     id_foto INT AUTO_INCREMENT PRIMARY KEY,  
->     foto LONGBLOB NOT NULL,  
->     id_usuario INT NOT NULL,  
->     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario) ON DELETE CASCADE  
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
Query OK, 0 rows affected (0.123 sec)  
  
MySQL [t9_2022630278_bdgu]> |  
  12°C Clear  Search  11:32 p.m.  
ENG US 21/12/2025
```

**Figura 9.3 Base de datos “t9\_2022630278\_bdgu”**

- Para GA:



```

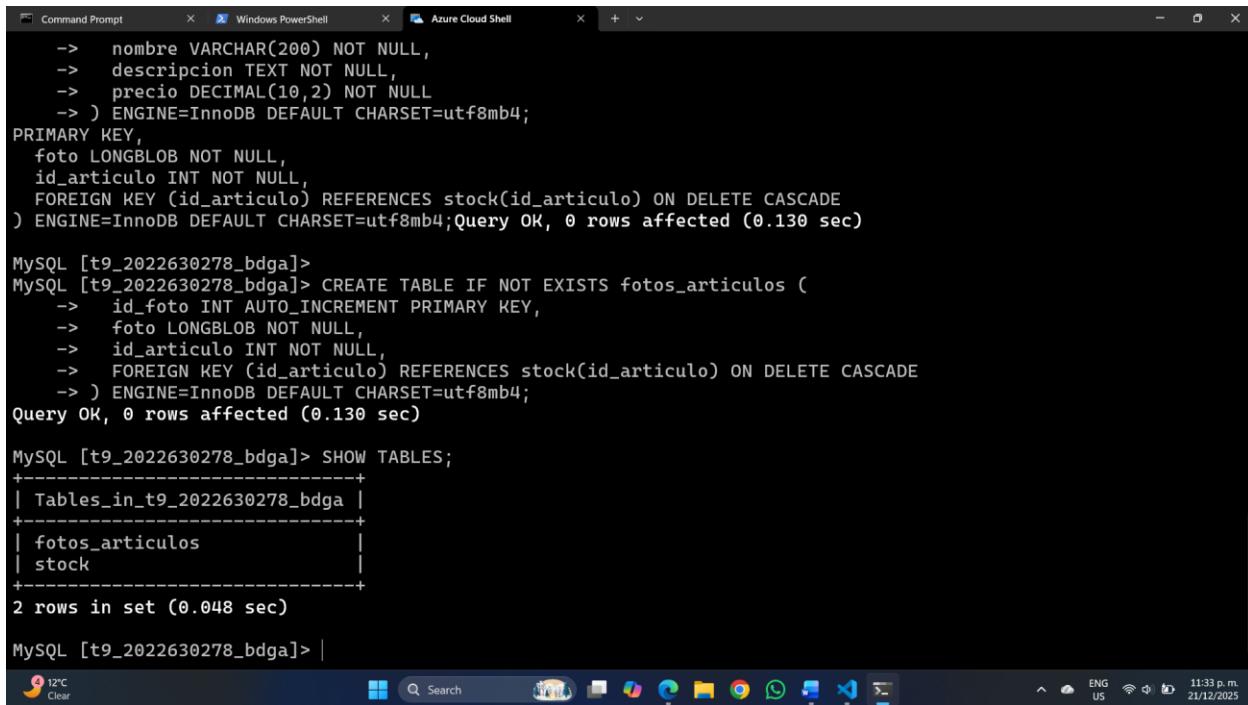
| fotos_usuarios           |
| usuarios                  |
+-----+
2 rows in set (0.049 sec)

MySQL [t9_2022630278_bdga]> USE t9_2022630278_bdga;
Database changed
MySQL [t9_2022630278_bdga]> CREATE TABLE IF NOT EXISTS stock (
->   id_articulo INT AUTO_INCREMENT PRIMARY KEY,
->   nombre VARCHAR(200) NOT NULL,
->   descripcion TEXT NOT NULL,
->   precio DECIMAL(10,2) NOT NULL
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
PRIMARY KEY,
foto LONGBLOB NOT NULL,
id_articulo INT NOT NULL,
FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;Query OK, 0 rows affected (0.130 sec)

MySQL [t9_2022630278_bdga]>
MySQL [t9_2022630278_bdga]> CREATE TABLE IF NOT EXISTS 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) ON DELETE CASCADE
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.130 sec)

MySQL [t9_2022630278_bdga]> |

```



```

->   nombre VARCHAR(200) NOT NULL,
->   descripcion TEXT NOT NULL,
->   precio DECIMAL(10,2) NOT NULL
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
PRIMARY KEY,
foto LONGBLOB NOT NULL,
id_articulo INT NOT NULL,
FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;Query OK, 0 rows affected (0.130 sec)

MySQL [t9_2022630278_bdga]>
MySQL [t9_2022630278_bdga]> CREATE TABLE IF NOT EXISTS 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) ON DELETE CASCADE
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.130 sec)

MySQL [t9_2022630278_bdga]> SHOW TABLES;
+-----+
| Tables_in_t9_2022630278_bdga |
+-----+
| fotos_articulos               |
| stock                          |
+-----+
2 rows in set (0.048 sec)

MySQL [t9_2022630278_bdga]> |

```

**Figura 9.4 Base de datos “t9\_2022630278\_bdga”**

- Para GC:

```
MySQL [t9_2022630278_bdga]> SHOW TABLES;
+-----+
| Tables_in_t9_2022630278_bdga |
+-----+
| fotos_articulos           |
| stock                       |
+-----+
2 rows in set (0.048 sec)

MySQL [t9_2022630278_bdga]> USE t9_2022630278_bdgc;
Database changed
MySQL [t9_2022630278_bdgc]> CREATE TABLE IF NOT EXISTS stock (
    -> id_articulo INT PRIMARY KEY,
    -> cantidad INT NOT NULL,
    -> UNIQUE KEY uq_stock_id_articulo (id_articulo)
    ->) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.133 sec)

MySQL [t9_2022630278_bdgc]> CREATE TABLE IF NOT EXISTS carrito_compra (
    -> id_usuario INT NOT NULL,
    -> id_articulo INT NOT NULL,
    -> cantidad INT NOT NULL,
    -> UNIQUE KEY uq_carrito (id_usuario, id_articulo)
    ->) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.144 sec)

MySQL [t9_2022630278_bdgc]> SELECT DATABASE();
12°C Clear
```

```
MySQL [t9_2022630278_bdga]> USE t9_2022630278_bdgc;
Database changed
MySQL [t9_2022630278_bdgc]> CREATE TABLE IF NOT EXISTS stock (
    -> id_articulo INT PRIMARY KEY,
    -> cantidad INT NOT NULL,
    -> UNIQUE KEY uq_stock_id_articulo (id_articulo)
    ->) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.133 sec)

MySQL [t9_2022630278_bdgc]> CREATE TABLE IF NOT EXISTS carrito_compra (
    -> id_usuario INT NOT NULL,
    -> id_articulo INT NOT NULL,
    -> cantidad INT NOT NULL,
    -> UNIQUE KEY uq_carrito (id_usuario, id_articulo)
    ->) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.144 sec)

MySQL [t9_2022630278_bdgc]> SHOW TABLES;
+-----+
| Tables_in_t9_2022630278_bdgc |
+-----+
| carrito_compra                |
| stock                          |
+-----+
2 rows in set (0.049 sec)

MySQL [t9_2022630278_bdgc]> |
```

Figura 9.3 Base de datos “t9\_2022630278\_bdgc”

## 7 Back-end: Diseño y funciones por microservicio

Se realizó el diseño del back-end separando la lógica en cuatro microservicios desplegados en AKS: Gestión de usuarios (GU), Gestión de artículos (GA), Gestión de compras (GC) y Servidor web (SW). Cada microservicio se instaló como una aplicación contenedrizada .NET 8 (Azure Functions aislado), con su propia base de datos MySQL en PaaS, y se accedió a través del API Gateway mediante rutas /api/\*. Se mantuvo aislamiento de datos: no se accedió a la base de datos de otro microservicio; las validaciones de acceso se realizaron vía HTTP al endpoint verifica\_acceso de GU.

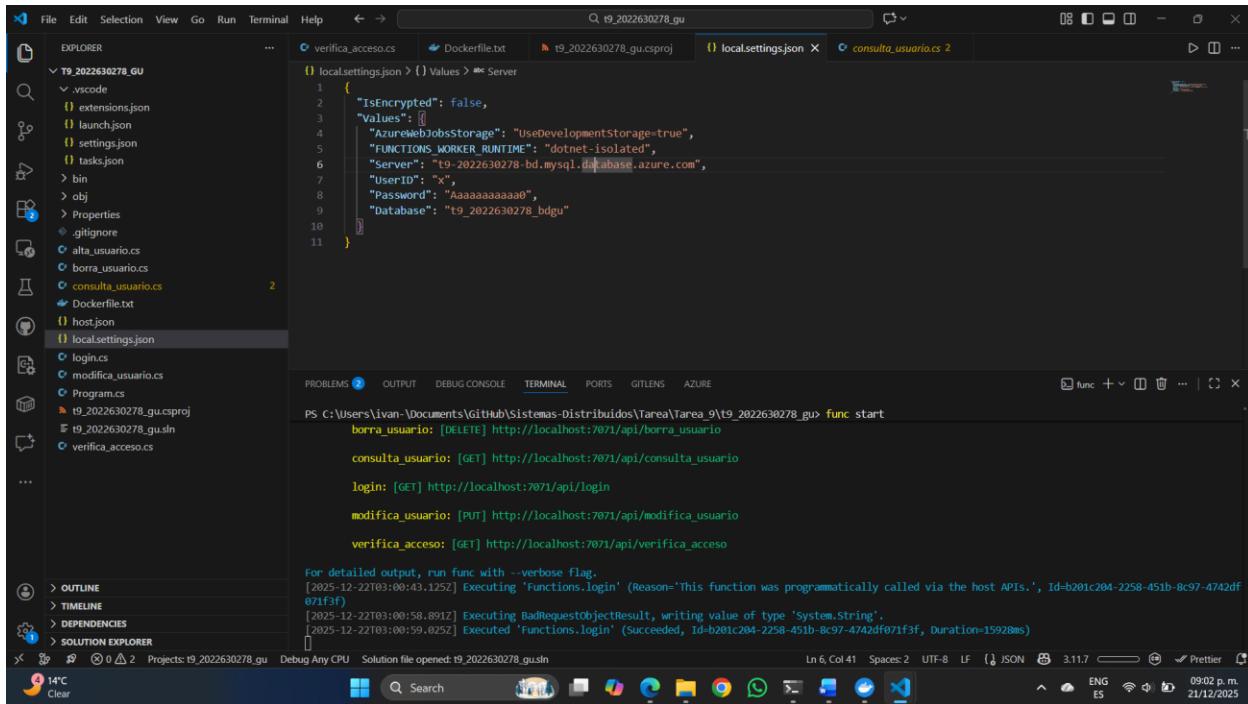
A continuación, se describe el diseño, las funciones y las evidencias esperadas por microservicio. Después de cada paso se coloca la figura indicada.

## 7.1 Microservicio Gestión de usuarios — t9\_2022630278\_gu

Se realizó la implementación de las funciones necesarias para administrar usuarios y tokens. Se instaló la base t9\_2022630278\_bdgu con tablas usuarios y fotos\_usuarios, y se accedió a los endpoints desde el API Gateway. Se agregó el endpoint verifica\_acceso para que GA y GC validaran el token sin acceder directamente a la DB de GU.

- Endpoints implementados:
  - alta\_usuario (POST): alta con datos personales y foto opcional.
  - consulta\_usuario (GET): consulta por email (requiere token).
  - modifica\_usuario (PUT): actualización de perfil y contraseña (requiere token).
  - borra\_usuario (DELETE): borrado de usuario y foto (requiere token).
  - login (GET): emisión de token tras validar email + password (hash).
  - verifica\_acceso (GET): válida email + token o id\_usuario + token y responde 200/400.

Se accedió a la DB para validar integridad de restricciones y se realizaron pruebas unitarias locales con curl verificando códigos 200/400.



**Figura 12. Proyecto t9\_2022630278\_gu**

## 7.2 Microservicio Gestión de artículos — t9\_2022630278\_ga

Se realizó la implementación de las funciones para administrar los metadatos de artículos y sus fotos. Se instaló la base t9\_2022630278\_bdga con tablas stock (id\_articulo auto-incremental, nombre, descripción, precio) y fotos\_articulos (foto y referencia al artículo). Se accedió a los endpoints desde el API Gateway y se integró con el microservicio GC para manejar la cantidad.

- Endpoints implementados:
  - alta\_articulo (POST): inserta en su stock y fotos\_articulos; después de insertar, se accedió a GC/alta\_articulo con { id\_articulo, cantidad, id\_usuario, token } para registrar existencias en la base de compras.
  - consulta\_articulos (GET): búsqueda por palabra clave (LIKE en nombre y descripcion) y retorno de id\_articulo, foto (base64), nombre, descripcion, precio.
- Comportamiento clave:

- Se realizó la verificación de acceso llamando a GU/verifica\_acceso.
- Se instaló el uso de HttpClient para invocar GC/alta\_articulo tras el alta en GA; se accedió al DNS interno del servicio GC vía ClusterIP.

```

localsettings.json > {} Values
1  {
2    "IsEncrypted": false,
3    "Values": {
4      "AzureWebJobsStorage": "UseDevelopmentStorage=true",
5      "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
6      "Server": "t9_2022630278-bd.mysql.database.azure.com",
7      "UserID": "x",
8      "Password": "aaaaaaaaaa0",
9      "Database": "t9_2022630278_bdga",
10     "USERS_URL": "http://localhost:7071",
11     "COMPRA_URL": "http://localhost:7073"
12   }
13 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS AZURE
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga> func start --port 7072
[2025-12-22T04:22:22,330Z] Worker process started and initialized.

Functions:
  alta_articulo: [POST] http://localhost:7072/api/alta_articulo
  consulta_articulos: [GET] http://localhost:7072/api/consulta_articulos
  consulta_articulo_por_id: [GET] http://localhost:7072/api/consulta_articulo_por_id

For detailed output, run func with --verbose flag.
[2025-12-22T04:22:32,785Z] Executing 'Functions.consulta_articulos' (Reason='This function was programmatically called via the host APIs.', Id=533d4f1e-91aa-429c-907e-2f1a2f384acf)
[2025-12-22T04:22:33,103Z] Executing BadRequestObjectResult, writing value of type 'System.String'.
[2025-12-22T04:22:33,165Z] Executed 'Functions.consulta_articulos' (Succeeded, Id=533d4f1e-91aa-429c-907e-2f1a2f384acf, Duration=398ms)

Ln 10 Col 42 Spaces: 2 UTF-8 LF JSON 3.11.7 Prettier 10:23 p.m. ENG ES 21/12/2025

```

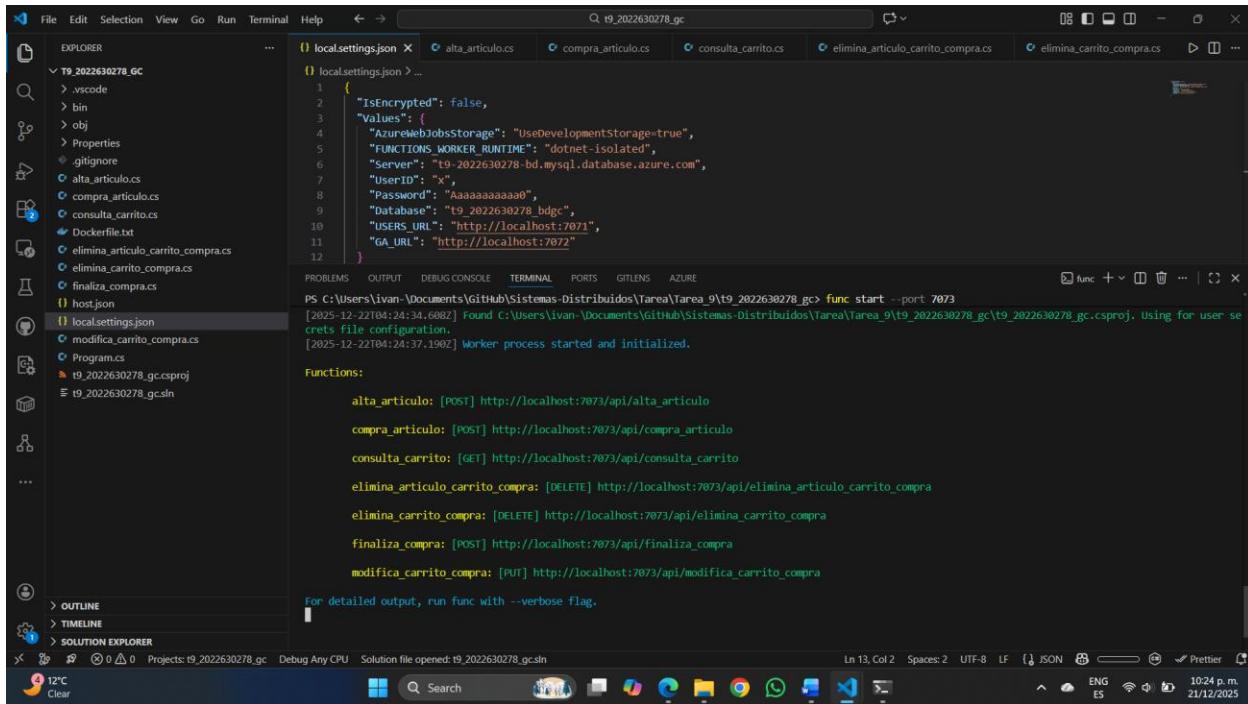
**Figura 14. Proyecto t9\_2022630278\_ga**

### 7.3 Microservicio Gestión de compras — t9\_2022630278\_gc

Se realizó la implementación de las funciones relacionadas con existencias (stock por artículo), el carrito de compra y operaciones transaccionales. Se instaló la base t9\_2022630278\_bdgc con stock (id\_articulo, cantidad, índice único) y carrito\_compra (id\_usuario, id\_articulo, cantidad, índice único). Se accedió a estas funciones vía API Gateway, respetando el aislamiento respecto a GA y GU.

- Endpoints implementados:
  - alta\_articulo (POST): da de alta id\_articulo y cantidad en tabla stock; requiere verifica\_acceso.

- compra\_articulo (POST): si cantidad <= existencia, inserta/actualiza carrito\_compra y descuenta en stock dentro de una transacción; en caso contrario responde 400 con “No hay suficientes artículos en stock”.
  - consulta\_carrito (GET): retorna arreglo con id\_articulo, cantidad, nombre, precio y foto (vía JOIN con datos necesarios).
  - elimina\_articulo\_carrito\_compra (DELETE): regresa cantidad a stock y elimina el registro de carrito\_compra dentro de transacción.
  - elimina\_carrito\_compra (DELETE): recorre artículos del carrito del usuario, regresa cantidades a stock y elimina en transacción.
  - modifica\_carrito\_compra (PUT): incremento (+1/-1) ajusta carrito\_compra y stock en transacción; responde 400 si no hay stock disponible para +1, o si ya no hay más artículos para -1.
  - finaliza\_compra (POST, opcional): calcula total, registra orden y detalle (si se implementó), vacía carrito\_compra sin regresar stock.
- Aseguramiento:
  - Se realizó la verificación de acceso llamando a GU/verifica\_acceso.
  - Se instaló el manejo de transacciones (BEGIN/COMMIT/ROLLBACK) en operaciones críticas de carrito y existencias.



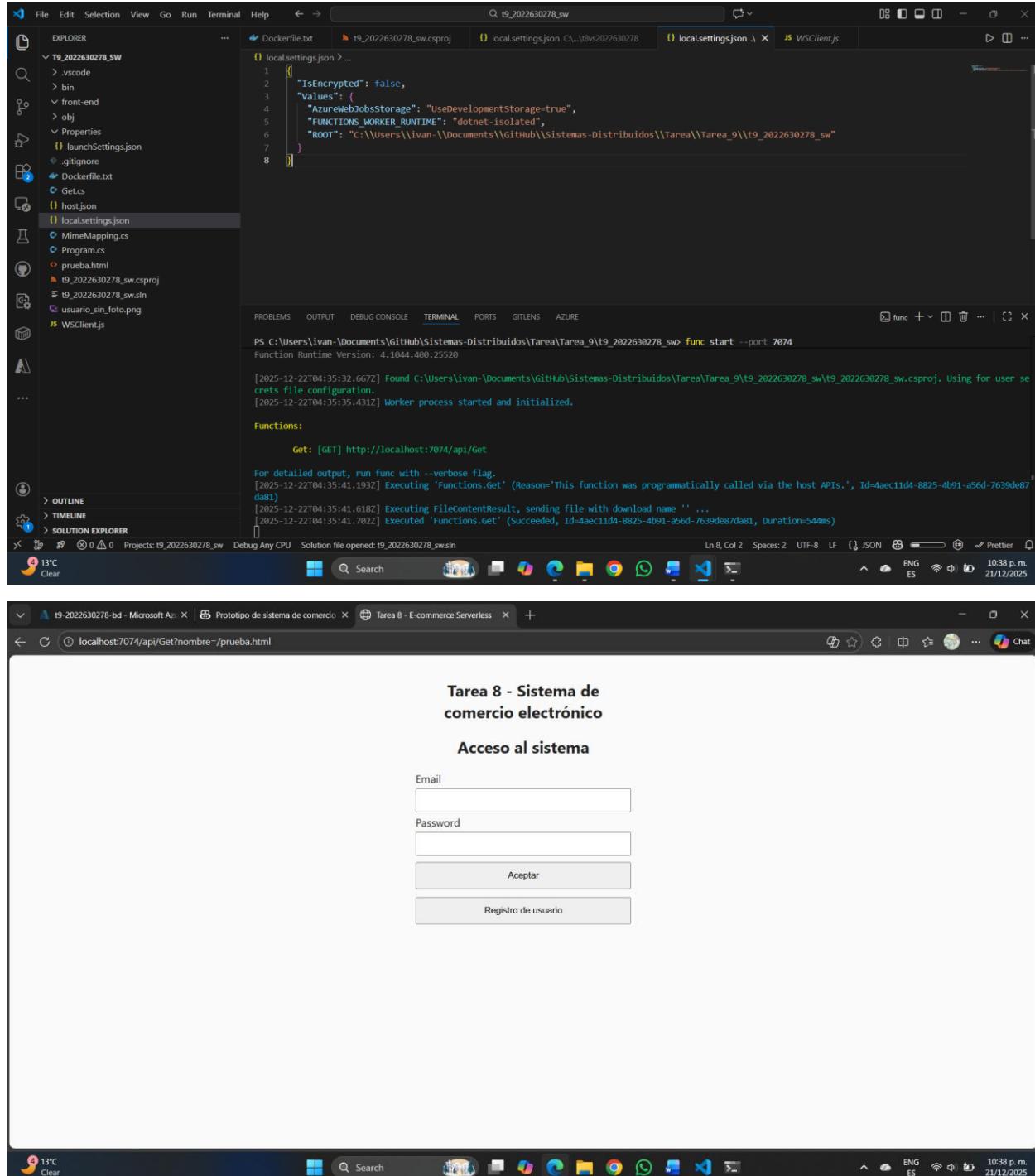
**Figura 16. Proyecto t9\_2022630278\_gc**

## 7.4 Microservicio Servidor web — t9\_2022630278\_sw

Se realizó la publicación del front-end mediante la función Get.cs. Se instaló el PV/PVC con Azure Files y se accedió al mount desde el contenedor, configurando la variable de entorno ROOT para servir archivos estáticos (prueba.html, WSClient.js). Este servicio se expuso internamente como ClusterIP y el API Gateway redirigió /api/Get?nombre=/prueba.html hacia SW.

- Función implementada:
  - Get (GET): sirve contenido estático desde ROOT + path, con manejo de If-Modified-Since, descarga o cache según encabezados.
- Configuración:
  - Se instaló el volumen webfiles (PVC) y se accedió como /mnt/webfiles dentro del contenedor SW.

- Se accedió a /api/Get?nombre=/prueba.html desde el navegador, validando que la aplicación front-end se cargó correctamente desde Azure Files.



**Figura 18. Montaje PV/PVC en SW y carga del front-end mediante /api/Get**

## 7.5 Dockerfiles de cada microservicio

Se agrego a la carpeta de cada proyecto el archivo Dockerfile y se uso el mismo para todos los proyectos. La creación del archivo Dockerfile y la construcción de la imagen tuvo los siguientes pasos:

Pasos:

- Crea Dockerfile en cada proyecto y se repite para GA, GC y SW con el mismo Dockerfile. Ejemplo:

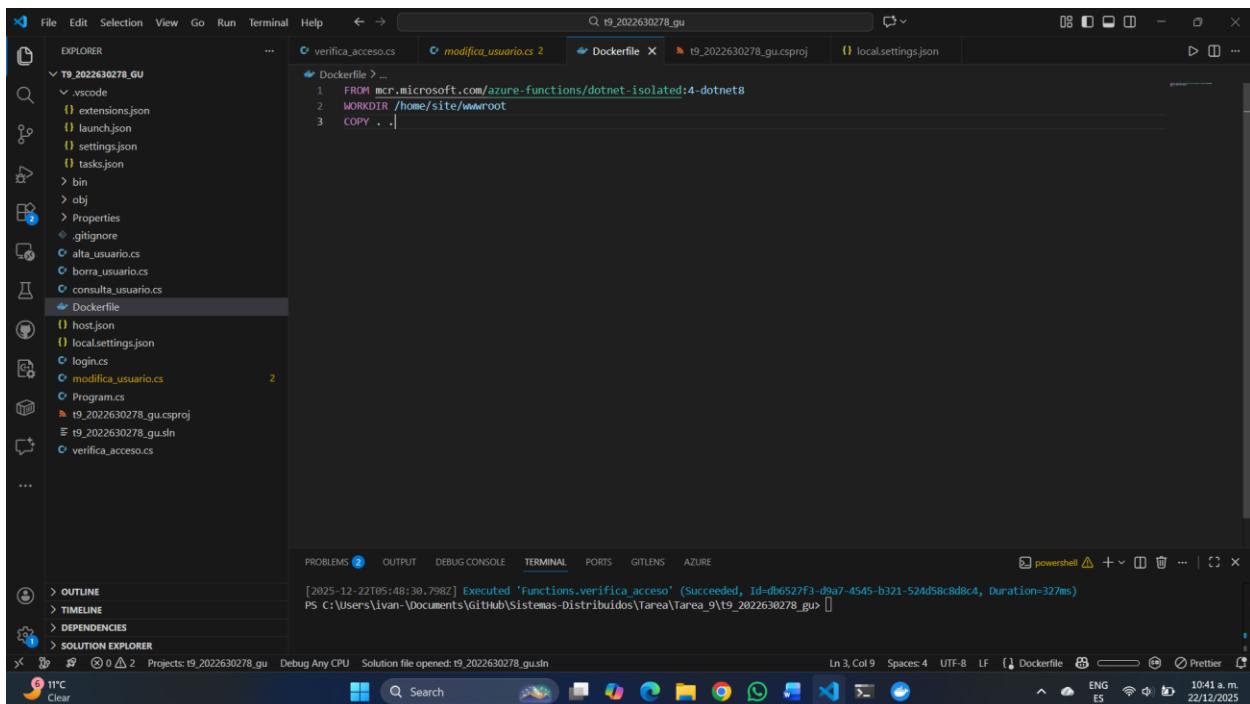


Figura 10.1 Archivo Dockerfile

- Elegir la imagen base recomendada: mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet8
- Publicar y empujar cada imagen a ACR.

Se abrió la terminal en cada proyecto y se ejecutó Build y push (Ejemplo para GU)

- az acr login --name t92022630278acr
- docker build -t t92022630278acr.azurecr.io/t9\_2022630278\_gu:latest .
- docker push t92022630278acr.azurecr.io/t9\_2022630278\_gu:latest

- Se realizo lo mismo para:
  - t9\_2022630278\_ga:latest
  - t9\_2022630278\_gc:latest
  - t9\_2022630278\_sw:latest

Para GU:

The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- File Explorer:** Shows the project structure for `t9_2022630278_GU`, including files like `.vscode`, `extensions.json`, `launch.json`, `settings.json`, `tasks.json`, `bin`, `obj`, `Properties`, `.gitignore`, `alta_usuario.cs`, `borra_usuario.cs`, `consulta_usuario.cs`, and `Dockerfile`.
- Editor:** Multiple tabs are open:
  - `verifica_acceso.cs`
  - `modifica_usuario.cs`
  - `Dockerfile`: Contains the following Dockerfile code:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
RUN cd /src/dotnet-function-app && \
    mkdir -p /home/site/wwwroot && \
    dotnet publish *.csproj --output /home/site/wwwroot
# To enable ssh & remote debugging on app service change the base image to the one below
```
  - `t9_2022630278_gu.csproj`
  - `localsettings.json`
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, GITLENS, and AZURE.
- Bottom Right:** Icons for powershell, GitHub, and other tools.

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

- File Explorer:** Shows a project structure for "T9\_2022630278\_GU".
- Dockerfile:** The file contains the following Dockerfile code:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
RUN cd /src/dotnet-function-app && \
    mkdir -p /home/site/wwwroot && \
    dotnet publish *.csproj --output /home/site/wwwroot

# To enable ssh & remote debugging on app service change the base image to the one below
# FROM mcr.microsoft.com/azure-functions/dotnet:8.0 AS base
# WORKDIR /home/site/wwwroot
# COPY ./publish .
# EXPOSE 80
# ENV DOTNET_RUNNING_IN_CONTAINER=true
```
- Terminal:** The terminal shows the command to build the Docker image:

```
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_gu> docker build -t t9_2022630278acr.azurecr.io/t9_2022630278_gu:latest .>> >> naming to t9_2022630278acr.azurecr.io/t9_2022630278_gu:latest>> >> unpacking to t9_2022630278acr.azurecr.io/t9_2022630278_gu:latest
```
- Output:** The output pane shows the build logs, including the creation of various Docker layers and the final digest.

## Figura 10.2 Despliegue de Dockerfiles

Para GA:

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

- Explorer:** Shows the project structure for "T9\_2022630278\_GA" with files like ".vscode", "bin", "obj", "Properties", ".gitignore", "alta\_articulo.cs", "consulta\_articulo\_por\_ids.cs", "consulta\_articulos.cs", "Dockerfile", "host.json", "local.settings.json", "Program.cs", and "t9\_2022630278\_ga.csproj".
- Dockerfile:** Content:

```

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
RUN cd /src/dotnet-function-app && \
    mkdir -p /home/site/wwwroot && \
    dotnet publish *.csproj --output /home/site/wwwroot

# To enable ssh & remote debugging on app service change the base image to the one below
# FROM mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0-appservice
FROM mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0
ENV AzureWebJobsScriptRoot=/home/site/wwwroot \
    AzureFunctionsJobHost__Logging__Console_IsEnabled=true

```
- Terminal:** Output from the command `az acr login --name t92022630278acr` followed by `docker build -t t9\_2022630278acr.azurecr.io/t9\_2022630278\_ga:latest`:

```

PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga> az acr login --name t92022630278acr
Login Succeeded
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga> docker build -t t9_2022630278acr.azurecr.io/t9_2022630278_ga:latest .
[+] Building 934.2s (11/11) FINISHED
--> [internal] load build definition from Dockerfile
--> => transferring dockerfile: 683B
--> [internal] load metadata for mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0
--> [internal] load metadata for mcr.microsoft.com/dotnet:8.0
--> [internal] load .dockerignore
--> => transferring context: 2B
--> [stage-1 1/2] FROM mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0@sha256:b592a0d6f978bc85cd6d4593ada13aa80446a79e94f40433fd40776ace9f
--> => resolve mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0@sha256:b592a0d6f978bc85cd6d4593ada13aa80446a79e94f40433fd40776ace9f
--> => 40776ace9f@sha256:d877d426a83c7fa409a795c2a45d5aa5fc4d4fb95f280bfb64ca6a81f7eb1474a9217fcfe57efbaecac9 98B / 22.60MB
--> => sha256:ae2e5253467ff7478b5f280bfb64ca6a81f7eb1474a9217fcfe57efbaecac9 98B / 22.60MB
--> => sha256:ae1219a1a4948a2c9ec7b55f280bfb64ca6a81f7eb1474a9217fcfe57efbaecac9 98B / 1458B
--> => sha256:73f2cf1fa2b2a70f3a0810476fe85e2a77962a38be96804649a23af24a628B / 6280B
--> => sha256:fa0d94df2a997a0jeif1295<17fbba01f9a5939222cf36a71bbcaeaaf877 126.59kB / 126.59kB
--> => sha256:acd50b79d215e1f4cf0a107edbcc391ae069b0b2f73c91ac2fb0a13348bf668B / 6680B
--> => sha256:94dcf7d5bb2d46267940ff7fa159b9b2144513795c01a7f12c519fied 2.72kB / 2.72kB
--> => sha256:69f1b47cb3918f1bf4001c6cd7afeze6b1a423572ec5bc295198cfb92f3e1 110.27MB / 110.27MB
--> => sha256:93b045b25b07001bc8971a436fb79591e5d10f6d8a5b6f0575e890ab3c99a9808 84.22MB / 84.22MB
--> => sha256:7f4659358cedee556a47795108ecda6b5323ac987071e1c5f79c85b6c11a64f5 135.51MB / 135.51MB

```

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

- Explorer:** Shows the project structure for "T9\_2022630278\_GA" with files like ".vscode", "bin", "obj", "Properties", ".gitignore", "alta\_articulo.cs", "consulta\_articulo\_por\_ids.cs", "consulta\_articulos.cs", "Dockerfile", "host.json", "local.settings.json", "Program.cs", and "t9\_2022630278\_ga.csproj".
- Dockerfile:** Content:

```

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
RUN cd /src/dotnet-function-app && \
    mkdir -p /home/site/wwwroot && \
    dotnet publish *.csproj --output /home/site/wwwroot

# To enable ssh & remote debugging on app service change the base image to the one below
# FROM mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0-appservice
FROM mcr.microsoft.com/azure-functions/dotnet-isolated:4-dotnet-isolated8.0
ENV AzureWebJobsScriptRoot=/home/site/wwwroot \
    AzureFunctionsJobHost__Logging__Console_IsEnabled=true

```
- Terminal:** Output from the command `ps aux | grep gpkqa` followed by `docker build -t t9\_2022630278acr.azurecr.io/t9\_2022630278\_ga:latest`:

```

PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga> ps aux | grep gpkqa
gpkqa 3768 0.0 0.0 0 0 ? [idle] 0:00 gpkqa3768:4@80:sysvnxk60 [ctrl + click]
view build details: docker-desktop://dashboard/build/desktop-linux/desktop-
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga> docker push t9_2022630278acr.azurecr.io/t9_2022630278_ga:latest
The push refers to repository [t9_2022630278acr.azurecr.io/t9_2022630278_ga]
39f35af708c0: waiting
d877d426a83c: waiting
7f14fdf6e40e: waiting
7f14fdf6e40e: mounted from t9_2022630278_ga
7f2659358ce6: mounted from t9_2022630278_ga
aecdce840e1c: mounted from t9_2022630278_ga
e4f696eda1f7b: mounted from t9_2022630278_ga
e9b7e85daff: pushed
840419c9e935: pushed
f6b1028c9327: mounted from t9_2022630278_ga
0e11259e1a49: mounted from t9_2022630278_ga
4b5ab07f27f7: mounted from t9_2022630278_ga
73fc2ff96280: mounted from t9_2022630278_ga
93b045b25b67: mounted from t9_2022630278_ga
04cd7d9b2d2: mounted from t9_2022630278_ga
69f1b47cb3: mounted from t9_2022630278_ga
9fad94df2a1: mounted from t9_2022630278_ga
ac05b67d5e: mounted from t9_2022630278_ga
5e2b3ded5d6: mounted from t9_2022630278_ga
latest: digest: sha256:16e93b88b2972d665497ee6b88a87f8941a3efbe54a6b38695934fb316d945 size: 856
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\t9_2022630278_ga>

```

Figura 10.3 Despliegue de Dockerfiles

Para GC:

The screenshot shows a Visual Studio Code interface with several tabs open. The left sidebar displays a file tree for a project named 't9\_2022630278\_gc'. The 'SOLUTION EXPLORER' tab is selected, showing files like 'EXPLORER', 'Dockerfile', 'alta\_articulo.cs', 'compra\_articulo.cs', and 'consulta\_carrito.cs'. The 'TERMINAL' tab is active, showing the following Dockerfile content:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
```

Below the terminal, the status bar indicates 'Ln 14, Col 71'. The bottom right corner shows icons for ENG ES, battery level (11%), and the date/time (22/12/2023 11:46 a.m.).

The screenshot shows a Visual Studio Code interface with several tabs open. The main editor tab contains a Dockerfile:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env  
COPY . /src/dotnet-function-app
```

Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, GITLENS, and AZURE. The DEBUG CONSOLE tab shows command-line output:

```
PS C:\Users\ivan\Documents\GitHub\sistemas-distribuidos\Tarea_9\t9_2022630278_gc> docker build -t t9_2022630278acr.azurecr.io/t9_2022630278_gc:latest  
View build details: docker-desktop://dashboard/desktop/linux/desktop/dwbc8cjar1z80c7kusxsysa2g  
PS C:\Users\ivan\Documents\GitHub\sistemas-distribuidos\Tarea_9\t9_2022630278_gc> docker push t9_2022630278acr.azurecr.io/t9_2022630278_gc:latest  
The push was successful.  
Pushed to repository [t9_2022630278acr.azurecr.io/t9_2022630278_gc]  
768269d42e: Pushed  
ae4cc09d40e1: Mounted from t9_2022630278_gu  
04cd17db9b2d: Mounted from t9_2022630278_gu  
a0e5253467ff: Mounted from t9_2022630278_gu  
7f2659358ce6: Mounted from t9_2022630278_gu  
5a020a0cb390: Pushed  
7f1f4ff6e40e: Mounted from t9_2022630278_gu  
d8774426a83c: Mounted from t9_2022630278_gu  
5e3b4e0d50b4: Mounted from t9_2022630278_gu  
acd50879d25e: Mounted from t9_2022630278_gu  
0e11259e1a90: Mounted from t9_2022630278_gu  
93ba45b25b07: Mounted from t9_2022630278_gu  
fb1b10289327: Mounted from t9_2022630278_gu  
39f3a5d78c0: Mounted from t9_2022630278_gu  
69ff1847c2b3: Mounted from t9_2022630278_gu  
9fad654d8f2a: Mounted from t9_2022630278_gu  
e4f6904af7fb: Mounted from t9_2022630278_gu  
4b5a007f27f7: Mounted from t9_2022630278_gu  
73fc0ff9b280: Mounted from t9_2022630278_gu  
latest: digest: sha256:aab0570f46b7ef71fc991f9d4879f6f21c7bfd25f6e7143c2bb944ae208eb548 size: 856
```

The bottom status bar shows the file is open at Line 14, Column 71, with tabs for Dockerfile, Prettier, and others.

## Figura 10.4 Despliegue de Dockerfiles

Para SW:

A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with a 'Dockerfile' selected. The main editor area shows a Dockerfile with several lines of code, including `FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env` and `COPY ./src/Isolate\_Functions .`. The status bar at the bottom indicates 'Projects:t9\_2022630278\_sw'. The bottom right corner shows the system tray with icons for battery, signal, and network.

The screenshot shows a Visual Studio Code interface with several tabs open. The left sidebar includes icons for Explorer, File, Edits, Selection, View, Go, Run, Terminal, Help, and a search bar. The Explorer tab shows a project structure for 'T9\_2022630278\_SW' containing files like '.vscode', 'bin', 'front-end', 'obj', 'Properties', 'launchSettings.json', '.gitignore', 'Dockerfile', 'Get.cs', 'host.json', 'local.settings.json', 'MimeMapping.cs', 'Programs.cs', 'prueba.html', 't9\_2022630278\_sw.csproj', 't9\_2022630278\_sw.sln', 'usuario\_sin foto.png', and 'WSClient.js'. The 'SOLUTION EXPLORER' panel at the bottom shows 'OUTLINE', 'TIMELINE', and 'SOLUTION EXPLORER' sections. The main area displays a Dockerfile and its build logs in a terminal window.

```
File Edit Selection View Go Run Terminal Help < > t9_2022630278_sw localsettings.json C:\_t9_2022630278_sw localsettings.json \ prueba.html

Dockerfile x t9_2022630278_sw.csproj localsettings.json C:\_t9_2022630278_sw localsettings.json \ prueba.html

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS installer-env
COPY . /src/dotnet-function-app
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea_9\t9_2022630278_sw> docker build -t t9_2022630278acr.azurecr.io/t9_2022630278_sw:latest .
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea_9\t9_2022630278_sw> docker push t9_2022630278acr.azurecr.io/t9_2022630278_sw:latest
The push refers to repository [t9_2022630278acr.azurecr.io/t9_2022630278_sw]
28c2e6a95e39: Waiting
28c2e6a95e39: Waiting
9fad65d0f2a: Waiting
28c2e6a95e39: Pushed
ffb10289327: Mounted from t9_2022630278_gu
4b5ab07f2ff: Mounted from t9_2022630278_gu
d62bf9f9a281: Pushed
a4e525467ff: Mounted from t9_2022630278_gu
acd56b79d25e: Mounted from t9_2022630278_gu
ae4ce00d0e1c: Mounted from t9_2022630278_gu
0e11259f1a49: Mounted from t9_2022630278_gu
e4f690d4f7fb: Mounted from t9_2022630278_gu
69ff1847c203: Mounted from t9_2022630278_gu
d8776426a83c: Mounted from t9_2022630278_gu
04cd7d9bb2d: Mounted from t9_2022630278_gu
7f1afdf6ea0e: Mounted from t9_2022630278_gu
39f3aa708c0: Mounted from t9_2022630278_gu
73fc2ff9b280: Mounted from t9_2022630278_gu
93b045b25bb7: Mounted from t9_2022630278_gu
latest: digest: sha256:1b8bf75588fe2a2a1fd1b861610d9823ad1e22aef741d5b6ffe12005d970225 size: 856
PS C:\Users\ivan\Documents\GitHub\Sistemas-Distribuidos\Tarea_9\t9_2022630278_sw>
```

## **Figura 10.4 Despliegue de Dockerfiles**

## 8 Conexión al cluster AKS desde VS Code o Cloud Shell

- Portal > AKS > t9-2022630278-aks > “Connect” > Copia el comando para descargar kubeconfig en Cloud Shell o usa local:

```
az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
```

Verifica:

```
kubectl get nodes
```

**t9-2022630278-aks**

**Essentials**

- Resource group: t9-2022630278-rg
- Power state: Running
- Cluster operation status: Succeeded
- Subscription: Azure for Students
- Location: Canada Central
- Subscription ID: fd58a3da-fcef-47d1-ac0e-5b891faa4251
- Fleet Manager: Click here to assign
- Kubernetes version: 1.33.5
- API server address: t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io
- SKU: Base
- Pricing tier: Free
- Network configuration: Azure CNI Overlay
- Node pools: 1 node pool
- Container registries: t9-2022630278acr
- Created time: December 21, 2025 at 06:20 PM

**Properties**

**Kubernetes services**

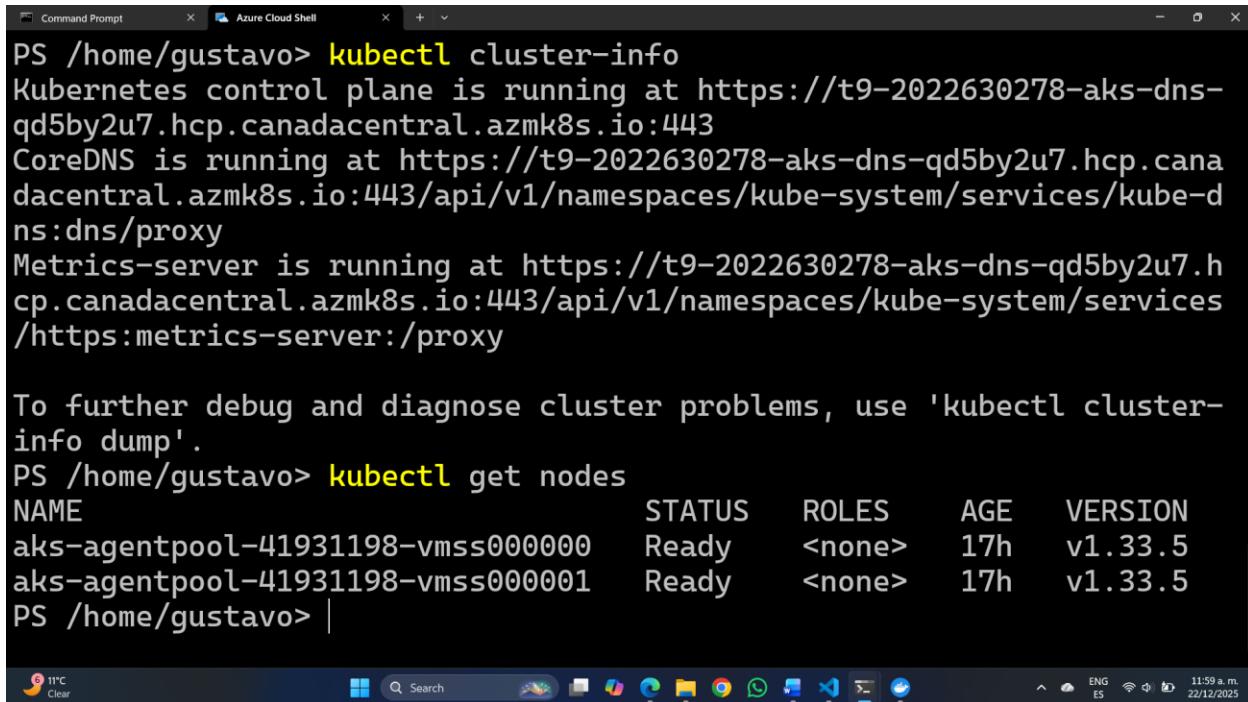
- Encryption type: Encryption at-rest with a platform-managed key
- Virtual node pools: Not enabled

**Networking**

- API server address: t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io
- Network configuration: Azure CNI Overlay
- Pod CIDR: 10.244.0.0/16
- Service CIDR: 10.0.0.0/16

```
f-a6b8-5f174b11ec1az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
Merged "t9-2022630278-aks" as current context in /home/gustavo/.kube/config
PS /home/gustavo>
PS /home/gustavo> add-AzADAppPermission -ObjectId 9cc74d5e-1162-4b90-8696-65f3d6a3f7d0 -ApiId 00000003-0000-0000-c000-000000000000 -PermissionId 5f8c59db-677d-491f-a6b8-5f174b11ec1az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
Merged "t9-2022630278-aks" as current context in /home/gustavo/.kube/config
PS /home/gustavo> kubectl cluster-info
Kubernetes control plane is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443
CoreDNS is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS /home/gustavo> |
```



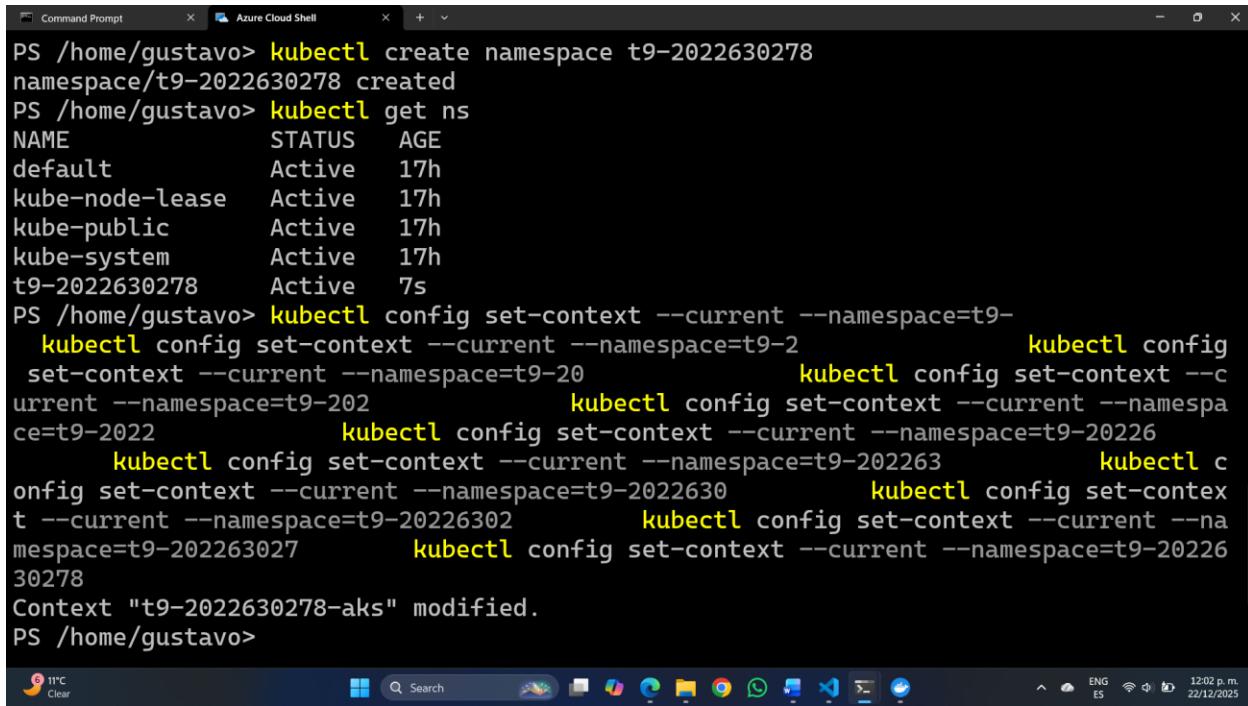
```
PS /home/gustavo> kubectl cluster-info
Kubernetes control plane is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443
CoreDNS is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
Metrics-server is running at https://t9-2022630278-aks-dns-qd5by2u7.hcp.canadacentral.azmk8s.io:443/api/v1/namespaces/kube-system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS /home/gustavo> kubectl get nodes
NAME                      STATUS   ROLES      AGE   VERSION
aks-agentpool-41931198-vmss000000   Ready    <none>    17h   v1.33.5
aks-agentpool-41931198-vmss000001   Ready    <none>    17h   v1.33.5
PS /home/gustavo> |
```

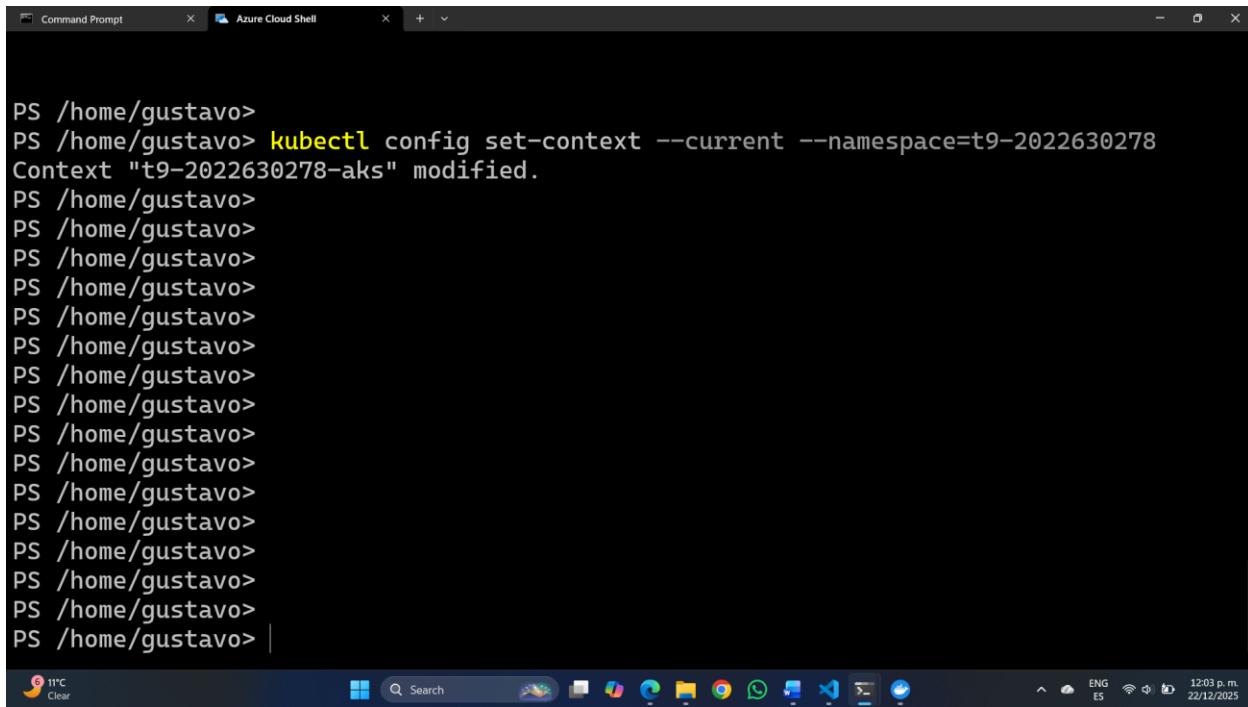
## Variables de entorno, Secrets y ConfigMaps

Se realizó la configuración de variables de entorno tanto en local como en Kubernetes. En Kubernetes, se instalaron Secrets para credenciales sensibles (Server, UserID, Password, Database por microservicio) y ConfigMaps para parámetros no sensibles (URLs internas de servicios y ROOT del servidor web). Se accedió a los manifests y se aplicaron con kubectl, verificando su creación correcta en el namespace del proyecto.

- Se crearon el namespace y se fijaron en el contexto actual usando estos comandos en una terminal de azure cloud Shell:
  - kubectl create namespace t9-2022630278
  - kubectl get ns
  - kubectl config set-context --current --namespace=t9-2022630278



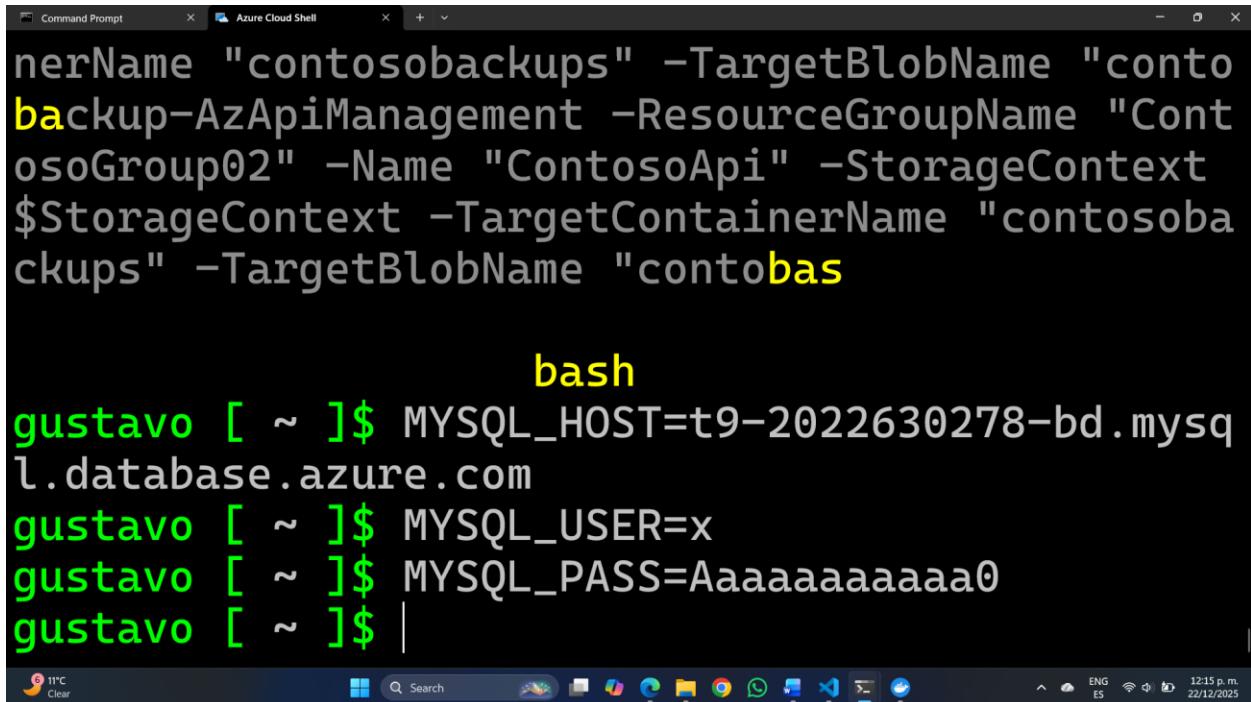
```
PS /home/gustavo> kubectl create namespace t9-2022630278
namespace/t9-2022630278 created
PS /home/gustavo> kubectl get ns
NAME      STATUS   AGE
default   Active   17h
kube-node-lease Active  17h
kube-public Active  17h
kube-system Active  17h
t9-2022630278 Active  7s
PS /home/gustavo> kubectl config set-context --current --namespace=t9-
    kubectl config set-context --current --namespace=t9-2022630278          kubectl config
    set-context --current --namespace=t9-2022630278          kubectl config set-context --c
urrent --namespace=t9-2022630278          kubectl config set-context --current --namespa
ce=t9-2022630278          kubectl config set-context --current --namespace=t9-2022630278
    kubectl config set-context --current --namespace=t9-2022630278          kubectl c
onfig set-context --current --namespace=t9-2022630278          kubectl config set-contex
t --current --namespace=t9-2022630278          kubectl config set-context --current --na
mespace=t9-2022630278          kubectl config set-context --current --namespace=t9-20226
30278
Context "t9-2022630278-aks" modified.
PS /home/gustavo>
```



```
PS /home/gustavo>
PS /home/gustavo> kubectl config set-context --current --namespace=t9-2022630278
Context "t9-2022630278-aks" modified.
PS /home/gustavo>
PS /home/gustavo> |
```

**Figura 10.2 Creacion del namespace**

- Se instaló el Secret mysql-credentials-gu, mysql-credentials-ga, mysql-credentials-gc con los parámetros de conexión de MySQL por servicio.

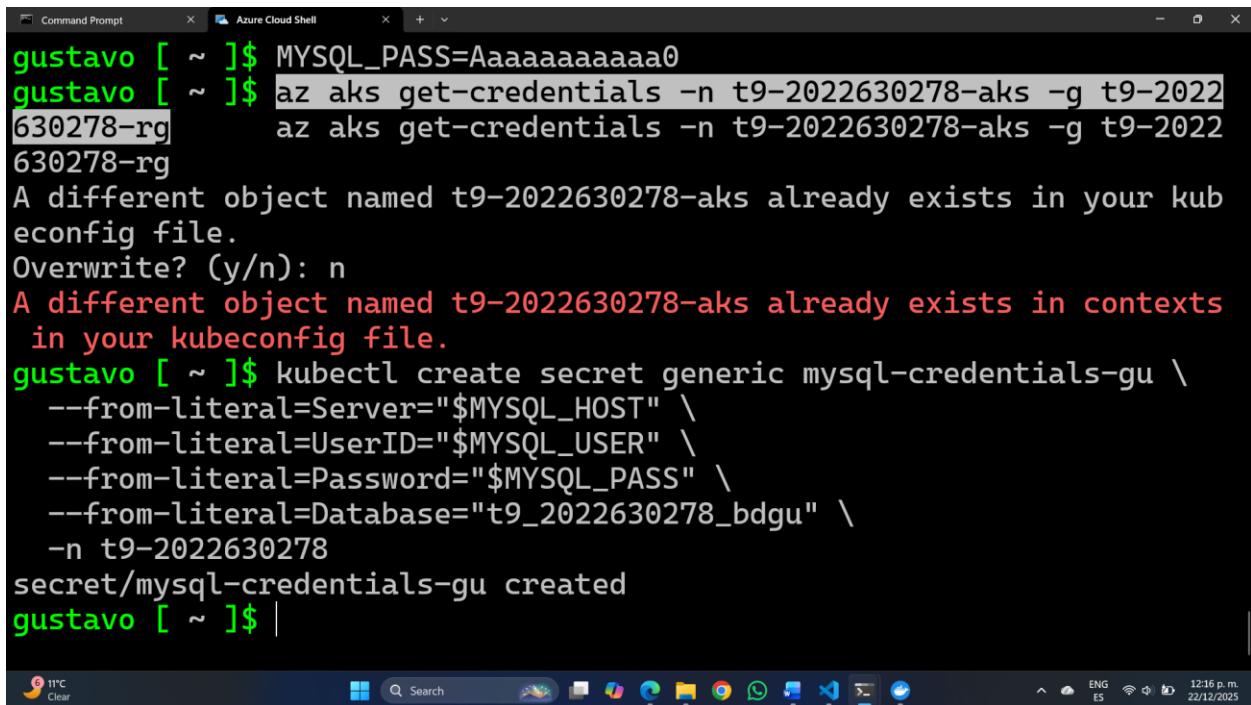


```
nerName "contosobackups" -TargetBlobName "conto
backup-AzApiManagement -ResourceGroupName "Cont
osoGroup02" -Name "ContosoApi" -StorageContext
$StorageContext -TargetContainerName "contosoba
ckups" -TargetBlobName "contobas

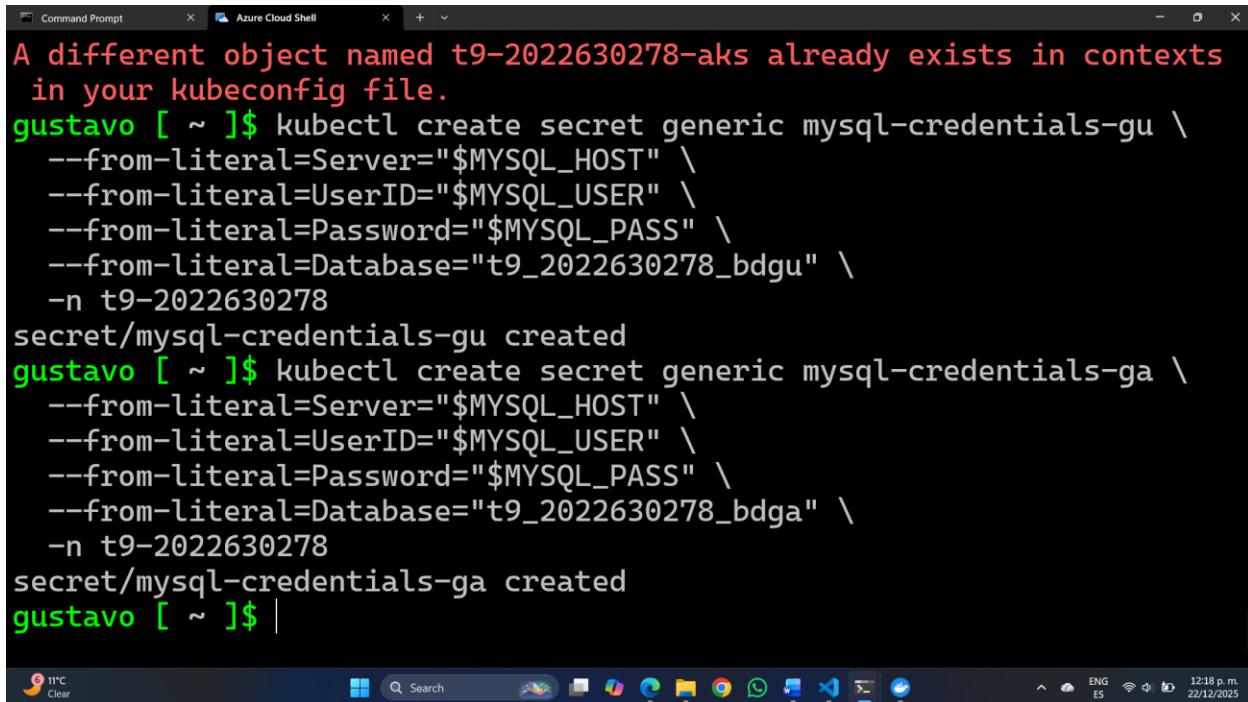
bash
gustavo [ ~ ]$ MYSQL_HOST=t9-2022630278-bd.mysql.database.azure.com
gustavo [ ~ ]$ MYSQL_USER=x
gustavo [ ~ ]$ MYSQL_PASS=Aaaaaaaaaaaa0
gustavo [ ~ ]$ |
```

Figura 10.6 Creación de variables de entorno para ingresar a MySQL

Para GU:



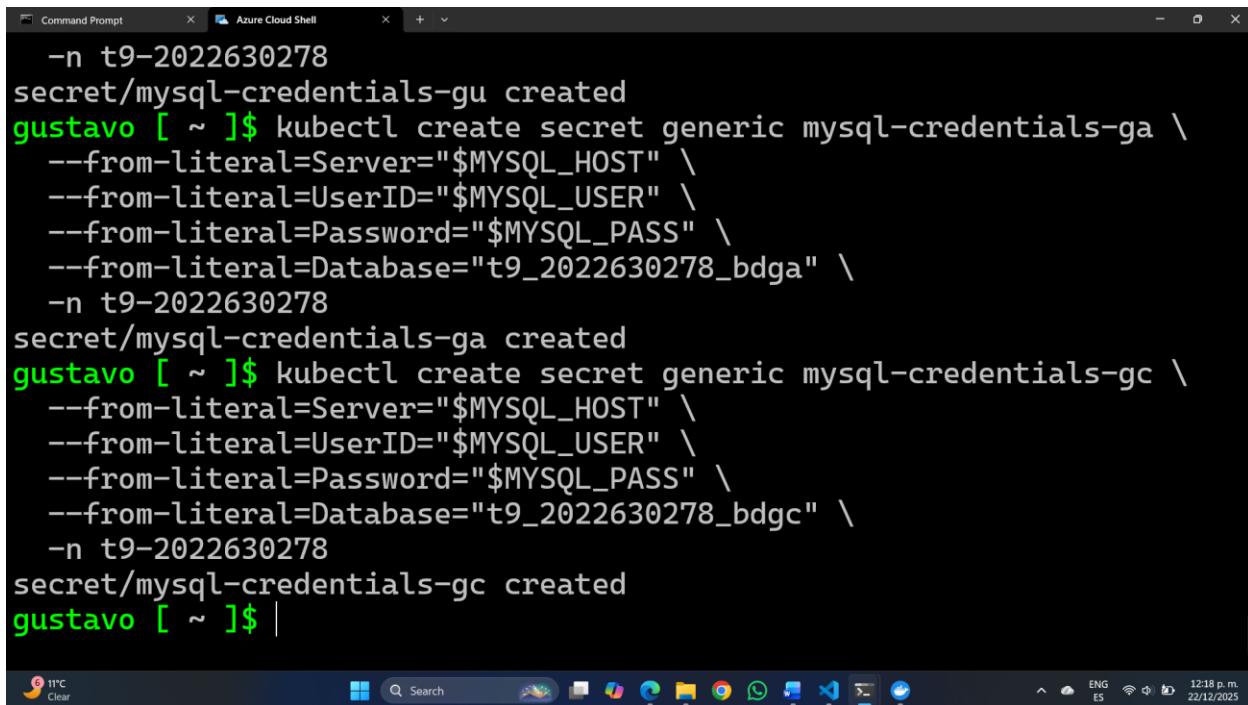
```
gustavo [ ~ ]$ MYSQL_PASS=Aaaaaaaaaaaa0
gustavo [ ~ ]$ az aks get-credentials -n t9-2022630278-aks -g t9-2022
630278-rg      az aks get-credentials -n t9-2022630278-aks -g t9-2022
630278-rg
A different object named t9-2022630278-aks already exists in your kub
econfig file.
Overwrite? (y/n): n
A different object named t9-2022630278-aks already exists in contexts
in your kubeconfig file.
gustavo [ ~ ]$ kubectl create secret generic mysql-credentials-gu \
--from-literal=Server="$MYSQL_HOST" \
--from-literal=UserID="$MYSQL_USER" \
--from-literal=Password="$MYSQL_PASS" \
--from-literal=Database="t9_2022630278_bdgu" \
-n t9-2022630278
secret/mysql-credentials-gu created
gustavo [ ~ ]$ |
```



```
A different object named t9-2022630278-aks already exists in contexts
in your kubeconfig file.
gustavo [ ~ ]$ kubectl create secret generic mysql-credentials-gu \
--from-literal=Server="$MYSQL_HOST" \
--from-literal=UserID="$MYSQL_USER" \
--from-literal=Password="$MYSQL_PASS" \
--from-literal=Database="t9_2022630278_bdgu" \
-n t9-2022630278
secret/mysql-credentials-gu created
gustavo [ ~ ]$ kubectl create secret generic mysql-credentials-ga \
--from-literal=Server="$MYSQL_HOST" \
--from-literal=UserID="$MYSQL_USER" \
--from-literal=Password="$MYSQL_PASS" \
--from-literal=Database="t9_2022630278_bdga" \
-n t9-2022630278
secret/mysql-credentials-ga created
gustavo [ ~ ]$ |
```

Figura 10.8 Creación de secretos para gu

Para GC:



```
-n t9-2022630278
secret/mysql-credentials-gu created
gustavo [ ~ ]$ kubectl create secret generic mysql-credentials-ga \
--from-literal=Server="$MYSQL_HOST" \
--from-literal=UserID="$MYSQL_USER" \
--from-literal=Password="$MYSQL_PASS" \
--from-literal=Database="t9_2022630278_bdga" \
-n t9-2022630278
secret/mysql-credentials-ga created
gustavo [ ~ ]$ kubectl create secret generic mysql-credentials-gc \
--from-literal=Server="$MYSQL_HOST" \
--from-literal=UserID="$MYSQL_USER" \
--from-literal=Password="$MYSQL_PASS" \
--from-literal=Database="t9_2022630278_bdgc" \
-n t9-2022630278
secret/mysql-credentials-gc created
gustavo [ ~ ]$ |
```

Figura 10.9 Creación de secretos para gc

Verifica que existen:

- `kubectl get secrets`
- `kubectl describe secret mysql-credentials-gu`
- `kubectl describe secret mysql-credentials-ga`
- `kubectl describe secret mysql-credentials-gc`

```
Command Prompt Azure Cloud Shell + - X
secret/mysql-credentials-gc created
gustavo [ ~ ]$ kubectl get secrets
NAME          TYPE    DATA  AGE
mysql-credentials-ga  Opaque  4    77s
mysql-credentials-gc  Opaque  4    52s
mysql-credentials-gu  Opaque  4    2m54s
gustavo [ ~ ]$ kubectl describe secret mysql-credentials-gu
Name:         mysql-credentials-gu
Namespace:    t9-2022630278
Labels:       <none>
Annotations: <none>

Type:  Opaque

Data
====

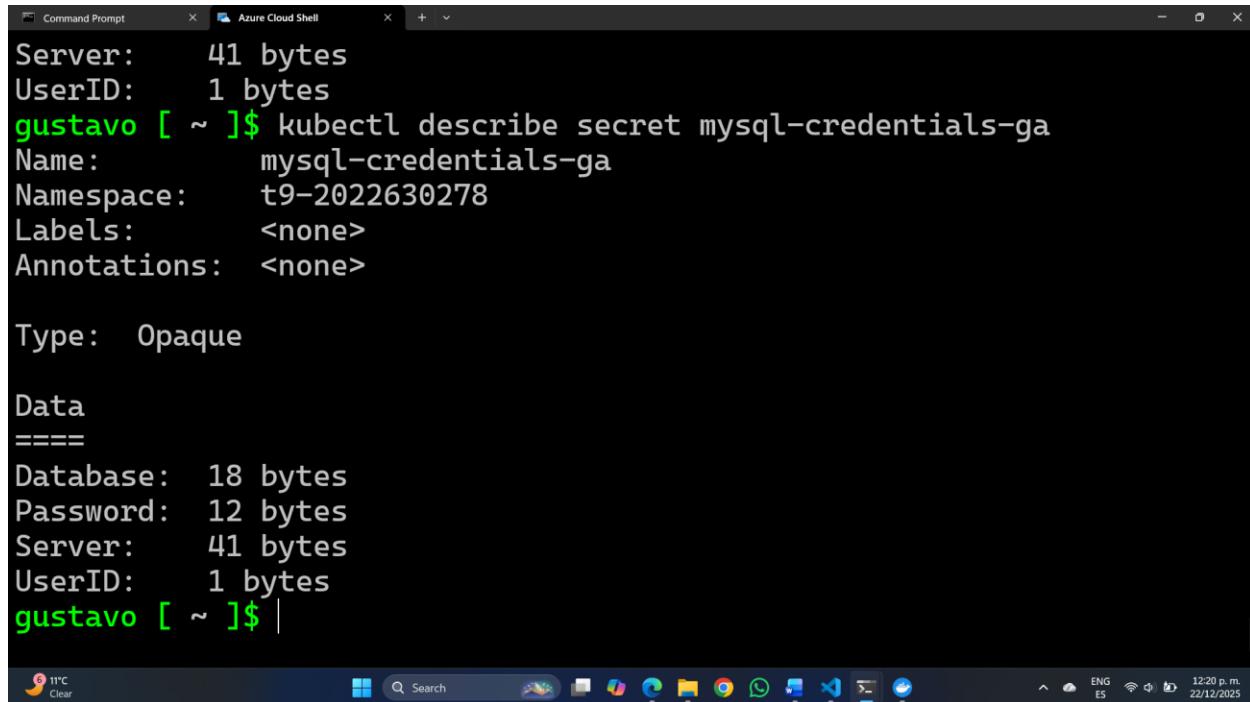
Database:  18 bytes
```

```
11°C Clear  Search  12:19 p.m. 22/12/2025
Command Prompt Azure Cloud Shell + - X
mysql-credentials-gc  Opaque  4    52s
mysql-credentials-gu  Opaque  4    2m54s
gustavo [ ~ ]$ kubectl describe secret mysql-credentials-gu
Name:         mysql-credentials-gu
Namespace:    t9-2022630278
Labels:       <none>
Annotations: <none>

Type:  Opaque

Data
====

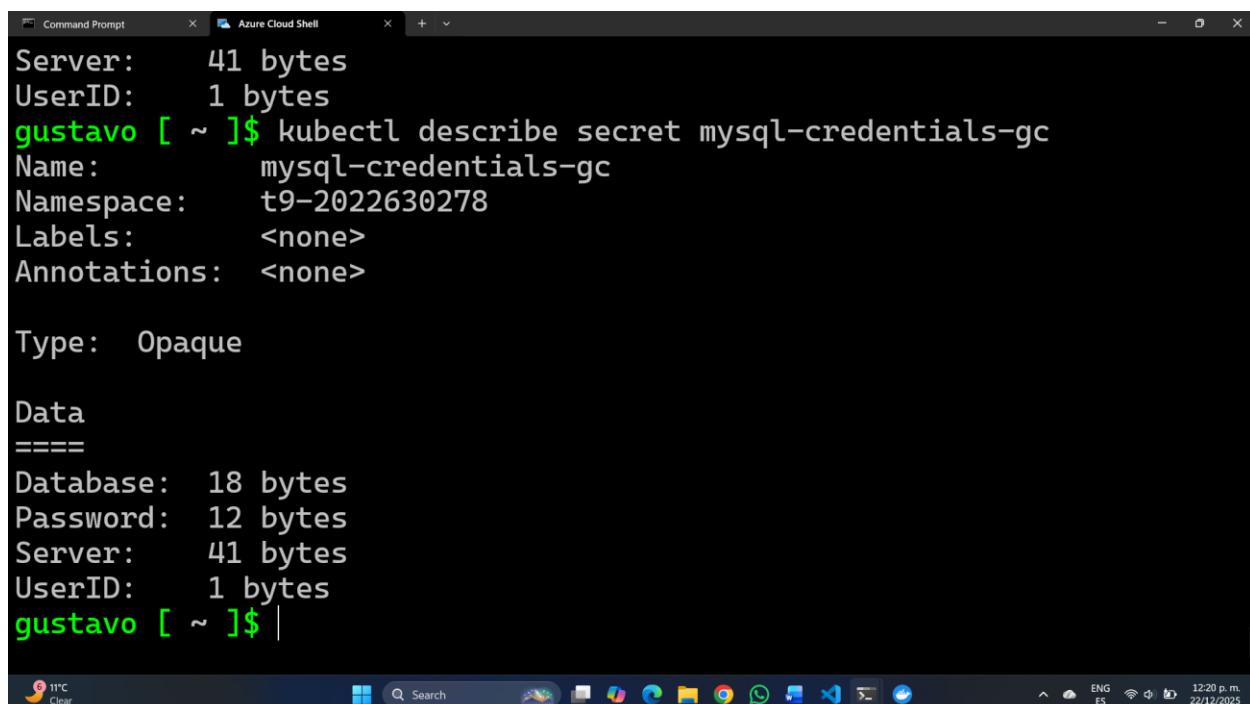
Database:  18 bytes
Password:  12 bytes
Server:    41 bytes
UserID:    1 bytes
gustavo [ ~ ]$ |
```



```
Server: 41 bytes
UserID: 1 bytes
gustavo [ ~ ]$ kubectl describe secret mysql-credentials-ga
Name:      mysql-credentials-ga
Namespace: t9-2022630278
Labels:    <none>
Annotations: <none>

Type:  Opaque

Data
=====
Database: 18 bytes
Password: 12 bytes
Server: 41 bytes
UserID: 1 bytes
gustavo [ ~ ]$ |
```



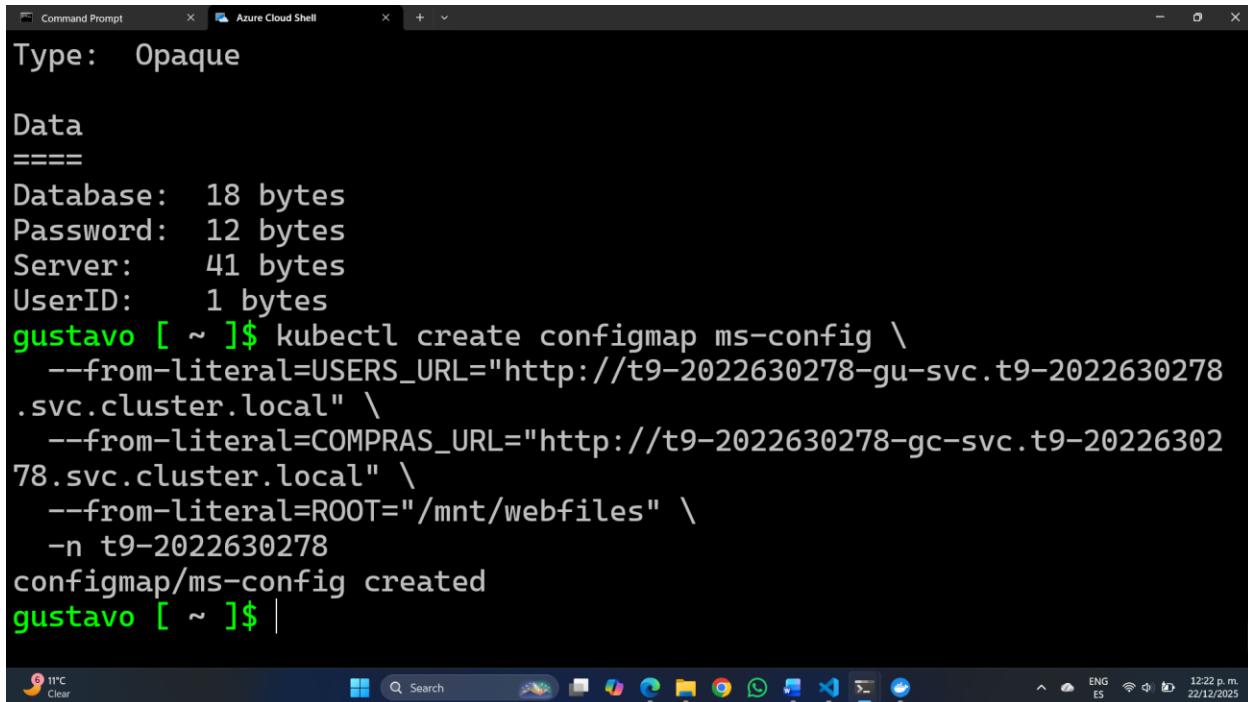
```
Server: 41 bytes
UserID: 1 bytes
gustavo [ ~ ]$ kubectl describe secret mysql-credentials-gc
Name:      mysql-credentials-gc
Namespace: t9-2022630278
Labels:    <none>
Annotations: <none>

Type:  Opaque

Data
=====
Database: 18 bytes
Password: 12 bytes
Server: 41 bytes
UserID: 1 bytes
gustavo [ ~ ]$ |
```

**Figura 10.9 Verificar secretos**

- Se instaló el ConfigMap ms-config con USERS\_URL, COMPRAS\_URL y ROOT.



```
Type: Opaque

Data
=====
Database: 18 bytes
Password: 12 bytes
Server: 41 bytes
UserID: 1 bytes
gustavo [ ~ ]$ kubectl create configmap ms-config \
--from-literal=USERS_URL="http://t9-2022630278-gu-svc.t9-2022630278.svc.cluster.local" \
--from-literal=COMPRAS_URL="http://t9-2022630278-gc-svc.t9-2022630278.svc.cluster.local" \
--from-literal=ROOT="/mnt/webfiles" \
-n t9-2022630278
configmap/ms-config created
gustavo [ ~ ]$ |
```

**Figura 10.10 Creación de secretos para urls**

- Se accedió a la sección de configuración de cada Deployment para validar envFrom y env apuntando a Secrets y ConfigMaps.

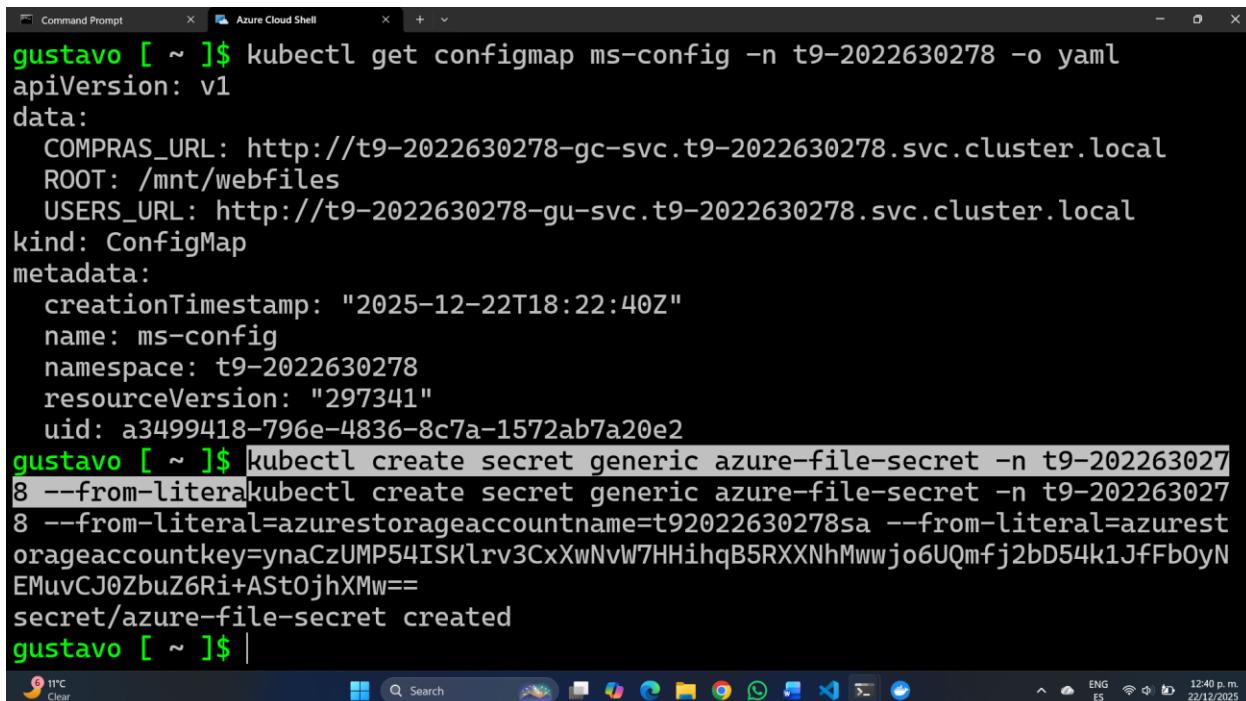
```
gustavo [ ~ ]$ kubectl get secrets -n t9-2022630278
NAME          TYPE    DATA   AGE
mysql-credentials-ga  Opaque  4      4m51s
mysql-credentials-gc  Opaque  4      4m26s
mysql-credentials-gu  Opaque  4      6m28s
gustavo [ ~ ]$ kubectl describe secret mysql-credentials-gu -n t9-2022630278
        kubectl describe secret mysql-credentials-gu -n t9-2022630278
Name:         mysql-credentials-gu
Namespace:    t9-2022630278
Labels:       <none>
Annotations: <none>
Type:        Opaque
Data
=====
Database:  18 bytes
Password:  12 bytes
Server:    41 bytes
UserID:    1 bytes
gustavo [ ~ ]$
```

```
Data
=====
Database:  18 bytes
Password:  12 bytes
Server:    41 bytes
UserID:    1 bytes
gustavo [ ~ ]$ kubectl get configmap ms-config -n t9-2022630278 -o yaml
apiVersion: v1
data:
  COMPRAS_URL: http://t9-2022630278-gc-svc.t9-2022630278.svc.cluster.local
  ROOT: /mnt/webfiles
  USERS_URL: http://t9-2022630278-gu-svc.t9-2022630278.svc.cluster.local
kind: ConfigMap
metadata:
  creationTimestamp: "2025-12-22T18:22:40Z"
  name: ms-config
  namespace: t9-2022630278
  resourceVersion: "297341"
  uid: a3499418-796e-4836-8c7a-1572ab7a20e2
gustavo [ ~ ]$ |
```

**Figura 10.10 ConfigMaps y Secrets del namespace del proyecto con variables de entorno aplicadas**

## 8.1 Crear PV y PVC para Azure Files (front-end)

- Se realizaron los Portal > Storage account > t92022630278sa > Access keys > copia la key1
- Cloud Shell:
  - kubectl create secret generic azure-file-secret -n t9-2022630278 --from-literal=azurestorageaccountname=t92022630278sa --from-literal=azurestorageaccountkey=<KEY1>



The screenshot shows a Windows desktop environment with the Azure Cloud Shell open in a Command Prompt window. The terminal output is as follows:

```
gustavo [ ~ ]$ kubectl get configmap ms-config -n t9-2022630278 -o yaml
apiVersion: v1
data:
  COMPRAS_URL: http://t9-2022630278-gc-svc.t9-2022630278.svc.cluster.local
  ROOT: /mnt/webfiles
  USERS_URL: http://t9-2022630278-gu-svc.t9-2022630278.svc.cluster.local
kind: ConfigMap
metadata:
  creationTimestamp: "2025-12-22T18:22:40Z"
  name: ms-config
  namespace: t9-2022630278
  resourceVersion: "297341"
  uid: a3499418-796e-4836-8c7a-1572ab7a20e2
gustavo [ ~ ]$ kubectl create secret generic azure-file-secret -n t9-2022630278 --from-literal=azurestorageaccountname=t92022630278sa --from-literal=azurestorageaccountkey=ynaCzUMP54ISKlrV3CxXwNvW7HHihqB5RXXNhMwwjo6UQmfj2bD54k1JfFbOyNEMuvCJ0ZbuZ6Ri+AST0jhXMw==
secret/azure-file-secret created
gustavo [ ~ ]$ |
```

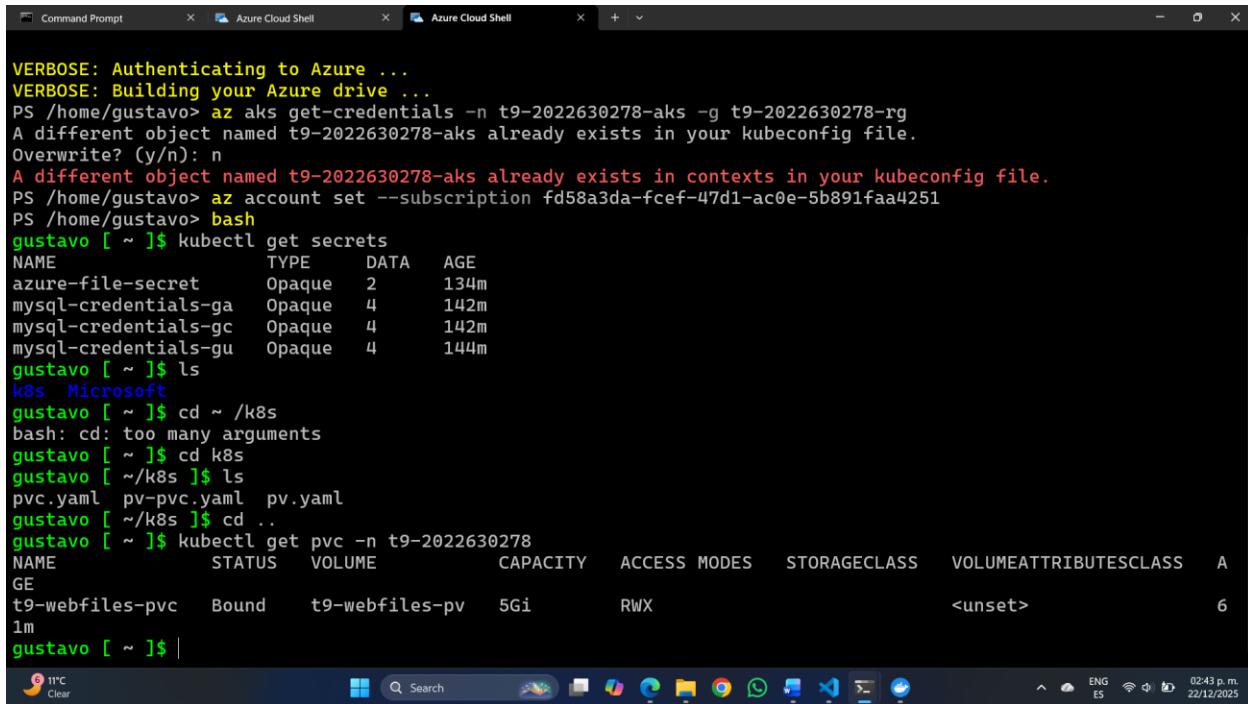
The status bar at the bottom of the terminal window shows the date and time as 12:40 p.m. on 22/12/2025.

Figura 10.11 Creación del secreto para la llave

2. Aplica PV/PVC:

```
gustavo [ ~ ]$ cat > k8s/pv-pvc.yaml <<EOF
apiVersion: v1
kind: PersistentVolume
metadata:
  name: t9-webfiles-pv
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  csi:
    driver: file.csi.azure.com
    volumeHandle: t9-webfiles-pv
  volumeAttributes:
    shareName: ${SHARE_NAME}
  nodeStageSecretRef:
    name: azure-file-secret
    namespace: ${NS}
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: t9-webfiles-pv
EOF
```

```
gustavo [ ~ ]$ kubectl apply -f k8s/pv-pvc.yaml
persistentvolume/t9-webfiles-pv created
persistentvolumeclaim/t9-webfiles-pvc created
gustavo [ ~ ]$ kubectl get pv
NAME      CAPACITY   ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM     STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
t9-webfiles-pv  5Gi        RWX           Retain       Available  <unset>
gustavo [ ~ ]$ kubectl get pvc
NAME      STATUS    VOLUME      CAPACITY   ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
t9-webfiles-pvc  Pending   t9-webfiles-pv  0          default      <unset>          47s
gustavo [ ~ ]$ kubectl get pv -n t9-2022630278
NAME      CAPACITY   ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM     STORAGECLASS  VOLUMEATTRIBUTESCLASS  REASON  AGE
t9-webfiles-pv  5Gi        RWX           Retain       Available  <unset>
gustavo [ ~ ]$ kubectl get pvc -n t9-2022630278
NAME      STATUS    VOLUME      CAPACITY   ACCESS MODES  STORAGECLASS  VOLUMEATTRIBUTESCLASS  AGE
t9-webfiles-pvc  Pending   t9-webfiles-pv  0          default      <unset>          97s
gustavo [ ~ ]$ |
```



```
VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/gustavo> az aks get-credentials -n t9-2022630278-aks -g t9-2022630278-rg
A different object named t9-2022630278-aks already exists in your kubeconfig file.
Overwrite? (y/n): n
A different object named t9-2022630278-aks already exists in contexts in your kubeconfig file.
PS /home/gustavo> az account set --subscription fd58a3da-fcef-47d1-ac0e-5b891faa4251
PS /home/gustavo> bash
gustavo [ ~ ]$ kubectl get secrets
NAME          TYPE      DATA   AGE
azure-file-secret  Opaque    2    134m
mysql-credentials-ga  Opaque    4    142m
mysql-credentials-gc  Opaque    4    142m
mysql-credentials-gu  Opaque    4    144m
gustavo [ ~ ]$ ls
k8s Microsoft
gustavo [ ~ ]$ cd ~ /k8s
bash: cd: too many arguments
gustavo [ ~ ]$ cd k8s
gustavo [ ~/k8s ]$ ls
pvc.yaml  pv-pvc.yaml  pv.yaml
gustavo [ ~/k8s ]$ cd ..
gustavo [ ~ ]$ kubectl get pvc -n t9-2022630278
NAME        STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
t9-webfiles-pvc  Bound   t9-webfiles-pv   5Gi        RWX           <unset>          6m
gustavo [ ~ ]$ |
```

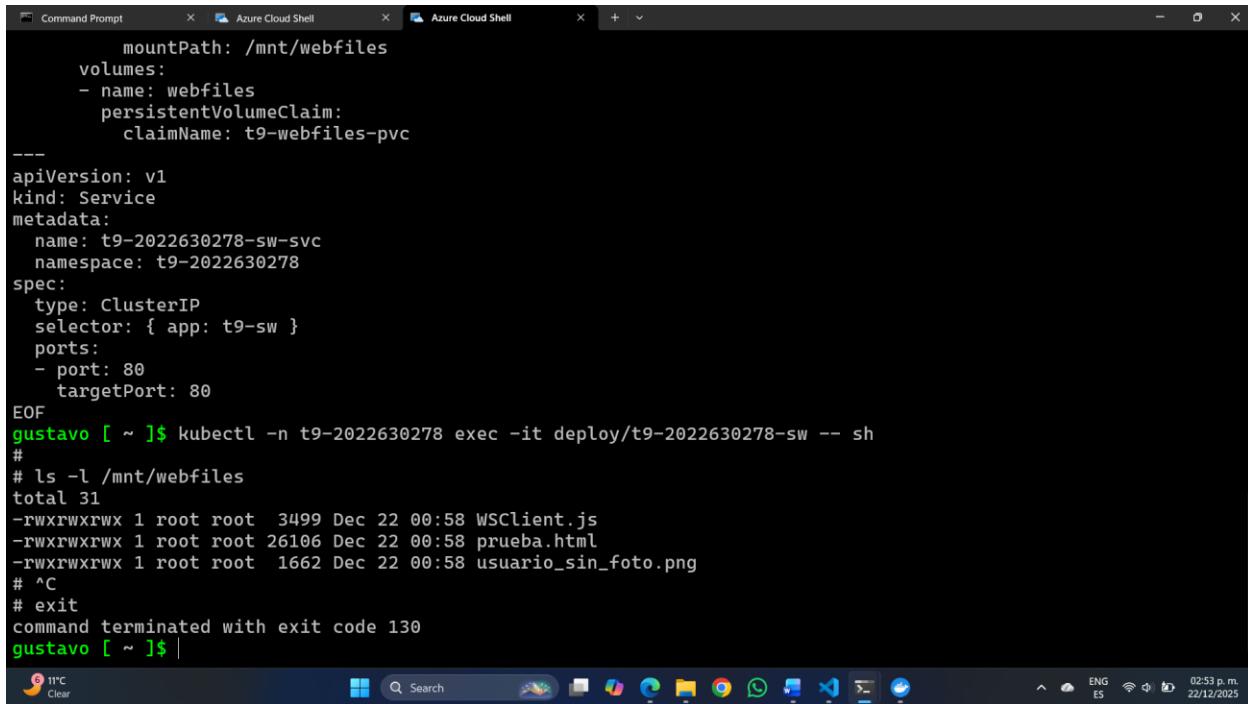
Figura 10.12 verificación de estatus de los yaml

#### 4. Montar el PVC en el microservicio “Servidor web”

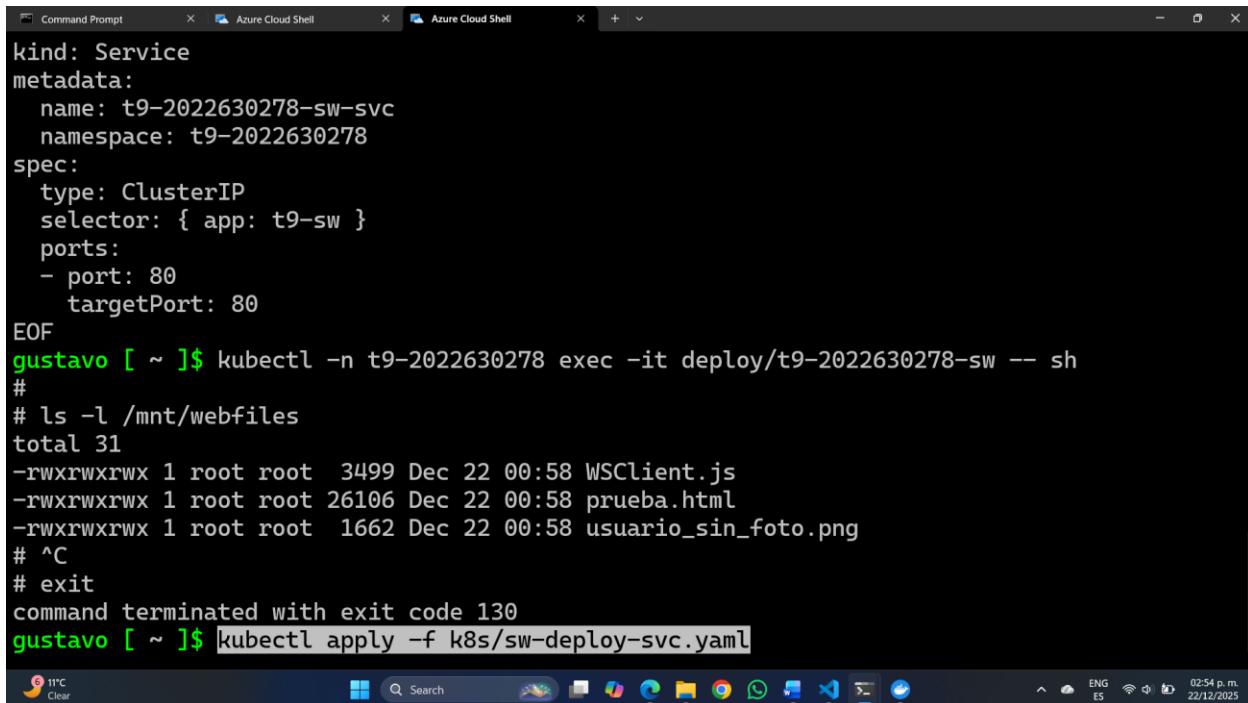
- En el Deployment del servicio web (t9-2022630278-sw), monta el PVC y usa ROOT=/mnt/webfiles. Si ya tienes tu manifiesto, revisa que tenga esto:

```
gustavo [ ~ ]$ cat > k8s/sw-deploy-svc.yaml <<'EOF'
apiVersion: apps/v1
kind: Deployment
metadata:
  name: t9-2022630278-sw
  namespace: t9-2022630278
spec:
  replicas: 2
  selector:
    matchLabels: { app: t9-sw }
  template:
    metadata:
      labels: { app: t9-sw }
    spec:
      containers:
        - name: sw
          image: t92022630278acr.azurecr.io/t9_2022630278_sw:latest
          ports:
            - containerPort: 80
          env:
            - name: ROOTB-SW-SVC
              value: /mnt/webfiles
          volumeMounts:
            - name: webfiles
              mountPath: /mnt/webfiles
      volumes:
        - name: webfiles
          persistentVolumeClaim:
            claimName: t9-webfiles-pvc
EOF
```

```
gustavo [ ~ ]$ cat > k8s/sw-deploy-svc.yaml <<'EOF'
containers:
  - name: sw
    image: t92022630278acr.azurecr.io/t9_2022630278_sw:latest
    ports:
      - containerPort: 80
    env:
      - name: ROOTB-SW-SVC
        value: /mnt/webfiles
    volumeMounts:
      - name: webfiles
        mountPath: /mnt/webfiles
volumes:
  - name: webfiles
    persistentVolumeClaim:
      claimName: t9-webfiles-pvc
---
apiVersion: v1
kind: Service
metadata:
  name: t9-2022630278-sw-svc
  namespace: t9-2022630278
spec:
  type: ClusterIP
  selector: { app: t9-sw }
  ports:
    - port: 80
      targetPort: 80
EOF
```



```
mountPath: /mnt/webfiles
volumes:
- name: webfiles
  persistentVolumeClaim:
    claimName: t9-webfiles-pvc
---
apiVersion: v1
kind: Service
metadata:
  name: t9-2022630278-sw-svc
  namespace: t9-2022630278
spec:
  type: ClusterIP
  selector: { app: t9-sw }
  ports:
  - port: 80
    targetPort: 80
EOF
gustavo [ ~ ]$ kubectl -n t9-2022630278 exec -it deploy/t9-2022630278-sw -- sh
#
# ls -l /mnt/Webfiles
total 31
-rwxrwxrwx 1 root root  3499 Dec 22 00:58 WSClient.js
-rwxrwxrwx 1 root root 26106 Dec 22 00:58 prueba.html
-rwxrwxrwx 1 root root 1662 Dec 22 00:58 usuario_sin_foto.png
# ^C
# exit
command terminated with exit code 130
gustavo [ ~ ]$
```



```
kind: Service
metadata:
  name: t9-2022630278-sw-svc
  namespace: t9-2022630278
spec:
  type: ClusterIP
  selector: { app: t9-sw }
  ports:
  - port: 80
    targetPort: 80
EOF
gustavo [ ~ ]$ kubectl -n t9-2022630278 exec -it deploy/t9-2022630278-sw -- sh
#
# ls -l /mnt/webfiles
total 31
-rwxrwxrwx 1 root root  3499 Dec 22 00:58 WSClient.js
-rwxrwxrwx 1 root root 26106 Dec 22 00:58 prueba.html
-rwxrwxrwx 1 root root 1662 Dec 22 00:58 usuario_sin_foto.png
# ^C
# exit
command terminated with exit code 130
gustavo [ ~ ]$ kubectl apply -f k8s/sw-deploy-svc.yaml
```

Figura 10.14 Montar el PVC en el microservicio “Servidor web”

## 9 API Gateway en AKS

Se realizó la configuración del API Gateway basado en el programa SimpleAPIGateway.java para actuar como punto de entrada único del sistema. Se instaló el gateway como un contenedor en AKS y se expuso mediante un Service de tipo LoadBalancer, de forma que todas las rutas /api/\* se accedieron desde el navegador o mediante curl, y fueron redirigidas al Service ClusterIP del microservicio correspondiente (GU, GA, GC, SW).

La implantación incluyó: ajuste de la tabla de enrutamiento, construcción y publicación de la imagen Docker en ACR, despliegue de Deployment y Service en AKS, y pruebas end-to-end para validar el correcto reenvío hacia los microservicios internos.

### 9.1 Implementación de SimpleAPIGateway.java

Se realizó la modificación de la tabla de enrutamiento para que el gateway redirigiera rutas /api/\* a los Services internos de Kubernetes. Se instaló la configuración de variables de entorno para el keystore y la password, y se mantuvo el comportamiento original de lectura de encabezados y cuerpo, reenviando la petición al host:port indicado por la tabla.

- Se ajustó tabla\_enrutamiento con entradas:
  - /api/login, /api/alta\_usuario, /api/consulta\_usuario, /api/modifica\_usuario, /api/borra\_usuario, /api/verifica\_acceso → t9-2022630278-gu-svc:80
  - /api/alta\_articulo, /api/consulta\_articulos → t9-2022630278-ga-svc:80
  - /api/compra\_articulo, /api/consulta\_carrito, /api/elimina\_articulo\_carrito\_compra, /api/elimina\_carrito\_compra, /api/modifica\_carrito\_compra, /api/finaliza\_compra → t9-2022630278-gc-svc:80
  - /api/Get → t9-2022630278-sw-svc:80
- Se accedió a la clase principal para confirmar que el socket seguro escuchara en el puerto configurado (por defecto 443) y que los hilos Worker\_1 y Worker\_2 conservaran su lógica de proxy.

```

1 import java.io.InputStream;
2 import java.io.OutputStream;
3 import java.io.IOException;
4 import java.net.Socket;
5 import java.net.ServerSocket;
6 import javax.net.ssl.SSLSocketFactory;
7 import java.io.ByteArrayOutputStream;
8
9 class SimpleAPIGateway
10 {
11     // Rutas -> Servicios (ClusterIP) y puerto 80 (mismo namespace t9-2022630278)
12     static String[][] tabla_enrutamiento =
13     {
14         {"api/Login", "t9-2022630278-gu-svc", "80"}, ...
15         {"api/alta_usuario", "t9-2022630278-gu-svc", "80"}, ...
16         {"api/consulta_usuario", "t9-2022630278-gu-svc", "80"}, ...
17         {"api/modifica_usuario", "t9-2022630278-gu-svc", "80"}, ...
18         {"api/borra_usuario", "t9-2022630278-gu-svc", "80"}, ...
19         {"api/verifica_acceso", "t9-2022630278-gu-svc", "80"}, ...
20         {"api/alta_articulo", "t9-2022630278-ga-svc", "80"}, ...
21         {"api/consulta_articulos", "t9-2022630278-ga-svc", "80"}, ...
22         {"api/compra_articulo", "t9-2022630278-gc-svc", "80"}, ...
23         {"api/consulta_carrito", "t9-2022630278-gc-svc", "80"}, ...
24         {"api/elimina_articulo_carrito_compra", "t9-2022630278-gc-svc", "80"}, ...
25         {"api/elimina_carrito_compra", "t9-2022630278-gc-svc", "80"}, ...
26         {"api/modifica_carrito_compra", "t9-2022630278-gc-svc", "80"}, ...
27         {"api/finaliza_compra", "t9-2022630278-gc-svc", "80"}, ...
28         {"api/Get", "t9-2022630278-sw-svc", "80"}, ...
29     };
30
31     static int TIMEOUT_READ = 1000; // milisegundos
32     static Object obj = new Object();
33
34     static class Worker_1 extends Thread
35     {
36         Socket cliente_1,cliente_2;
37     }
38
39 }

```

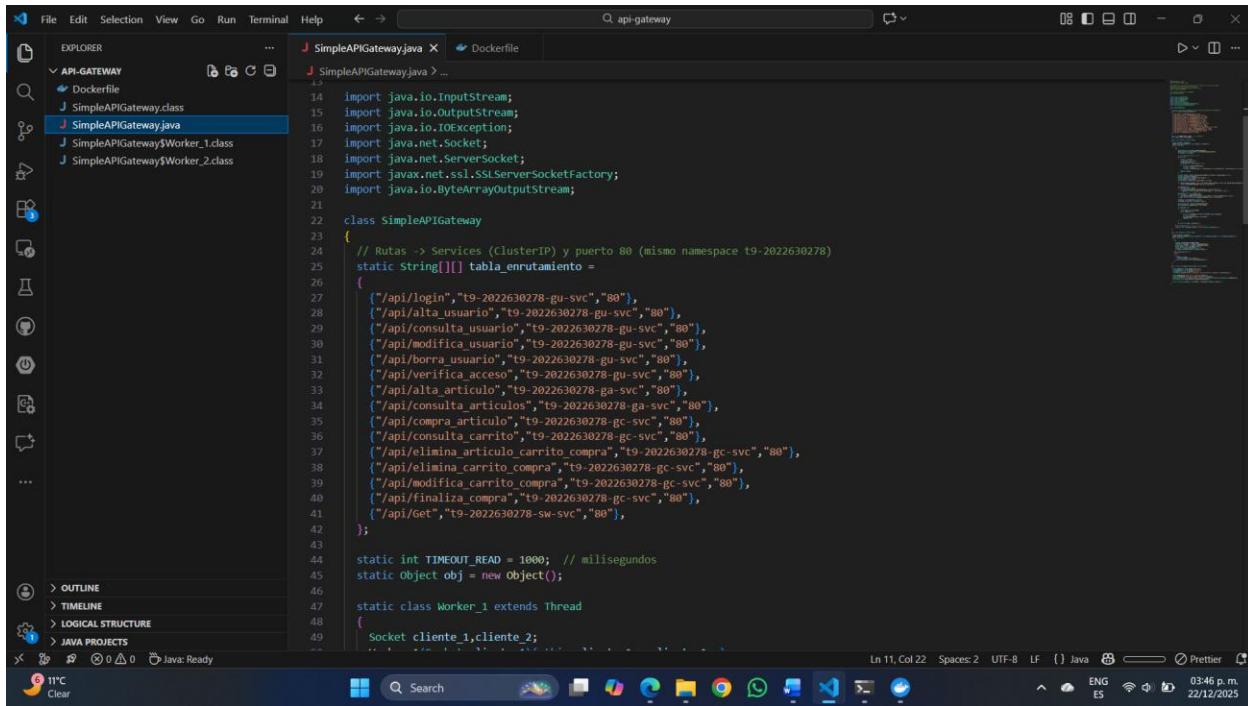
**Figura 19. Tabla de enrutamiento en SimpleAPIGateway.java apuntando a los Services ClusterIP**

## 9.2 Construcción de imagen Docker y publicación en ACR

Se realizó la construcción de la imagen Docker del API Gateway y se instaló en el registro ACR (t92022630278acr). Se accedió al repositorio del registro para confirmar el tag más reciente (latest) y se verificó que AKS tuviera permisos de pull.

Paso 1. Crear el código del gateway (local, en VS Code)

- Crea una carpeta local llamada api-gateway.
- Dentro, crea el archivo SimpleAPIGateway.java con la tabla de enrutamiento que apuntará a los Services de K8s.



```
SimpleAPIGateway.java
import java.io.InputStream;
import java.io.OutputStream;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;
import javax.net.ssl.SSLSocketFactory;
import java.io.ByteArrayOutputStream;
class SimpleAPIGateway
{
    // Rutas -> Servicios (ClusterIP) y puerto 80 (mismo namespace t9-2022630278)
    static String[][] tabla_enrutamiento =
    {
        {" /api/login","t9-2022630278-gu-svc","80"}, 
        {" /api/alta_usuario","t9-2022630278-gu-svc","80"}, 
        {" /api/consulta_usuario","t9-2022630278-gu-svc","80"}, 
        {" /api/modifica_usuario","t9-2022630278-gu-svc","80"}, 
        {" /api/borra_usuario","t9-2022630278-gu-svc","80"}, 
        {" /api/verifica_acceso","t9-2022630278-gu-svc","80"}, 
        {" /api/alta_articulo","t9-2022630278-ga-svc","80"}, 
        {" /api/consulta_articulos","t9-2022630278-ga-svc","80"}, 
        {" /api/compra_articulo","t9-2022630278-gc-svc","80"}, 
        {" /api/consulta_carrito","t9-2022630278-gc-svc","80"}, 
        {" /api/elimina_articulo_carrito_compra","t9-2022630278-gc-svc","80"}, 
        {" /api/elimina_carrito_compra","t9-2022630278-gc-svc","80"}, 
        {" /api/modifica_articulo_compra","t9-2022630278-gc-svc","80"}, 
        {" /api/finaliza_compra","t9-2022630278-gc-svc","80"}, 
        {" /api/Get","t9-2022630278-sw-svc","80"}, 
    };
    static int TIMEOUT_READ = 1000; // milisegundos
    static Object obj = new Object();
    static class Worker_1 extends Thread
    {
        Socket cliente_1,cliente_2;
    }
}
```

**Figura 19.1 Tabla de enrutamiento en SimpleAPIGateway.java apuntando a los Services ClusterIP**

Paso 2. Crear el Dockerfile del gateway (local, en VS Code)

- En la misma carpeta api-gateway, crea el Dockerfile:

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

- File Explorer:** Shows a project named "API-GATEWAY" containing a "Dockerfile" and several Java class files: "SimpleAPIGateway.class", "SimpleAPIGateway.java", "SimpleAPIGateway\$Worker\_1.class", and "SimpleAPIGateway\$Worker\_2.class".
- Editor:** Displays the "Dockerfile" content:

```
FROM eclipse-temurin:17-jdk
WORKDIR /app
COPY SimpleAPIGateway.java .
RUN javac SimpleAPIGateway.java
# No pongas secretos en ENV; se inyectan en Kubernetes
ENV keystore=/keystore/keystore_servidor.jks
ENV password=1234567
CMD ["java","SimpleAPIGateway"]
```

The status bar at the bottom indicates "Java: Ready".

**Figura 19.2 Creación del Dockerfile**

Paso 3. Build y push de la imagen a ACR (local, terminal en VS Code)

- Abre una terminal en la carpeta api-gateway y ejecuta:
  - az acr login -n t92022630278acr
  - docker build -t t92022630278acr.azurecr.io/simple-api-gateway:latest .
  - docker push t92022630278acr.azurecr.io/simple-api-gateway:latest

```

FROM eclipse-temurin:17-jdk
WORKDIR /app
COPY SimpleAPIGateway.java .
RUN java SimpleAPIGateway.java
ENV keystore=/keystore/keystore_
ENV password=123456
CMD ["java","SimpleAPIGateway"]

```

This removes the need to install a JDK manually with apt-get.

2. How to build the image (no RUN in PowerShell)

From your api-gateway folder, run in PowerShell:

```
docker build -t api-gateway .
```

Don't type RUN ... in the terminal; just edit the Dockerfile and let docker build process the RUN instructions during the build.

If you want, I can also help you move the password out of ENV into a build arg or runtime env to avoid the warning about sensitive data.

GPT-5.1 x

SimpleAPIGateway.java

Explore and understand your code

Java: Ready

```

FROM eclipse-temurin:17-jdk
WORKDIR /app
COPY SimpleAPIGateway.java .
RUN java SimpleAPIGateway.java
# NO pongas secretos en ENV; se inyectan en Kubernetes
ENV keystore=/keystore/keystore_servidor.jks
ENV password=123456
CMD ["java","SimpleAPIGateway"]

```

Dockerfile > -

View build details: docker desktop://dashboard/build/desktop\_linux/desktop\_linux/ejx5vuuxt580kddn7qdh

PS C:\Users\ivan\Documents\github\Sistemas-Distribuidos\Tarea\Tarea\_9\api-gateway> docker push t92022630278acr.azurecr.io/simple-api-gateway:latest

The push refers to repository [t92022630278acr.azurecr.io/simple-api-gateway]

9958d88ac0b4: Pushed  
20043066d5d5: Pushed  
94ab5add45d5: Pushed  
ff877379834c: Pushed  
67cf99ea3a75: Pushed  
b88b6ae1b108: Pushed  
077c4ca28173: Pushed  
75ab242b0442: Pushed  
02080376fd82: Pushed

latest: digest: sha256:38156a1bb2e0c5361558116d7f4c7a8444851e8fb6695af0/b36beb0045558 size: 856

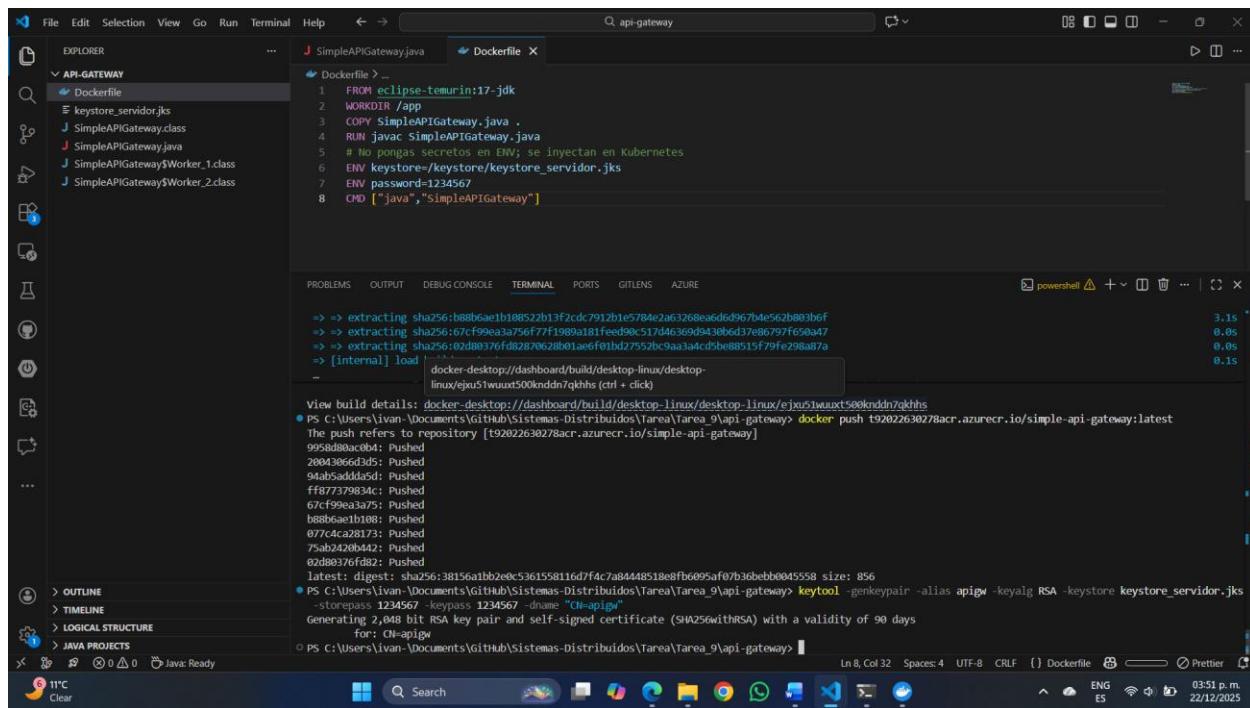
PS C:\Users\ivan\Documents\github\Sistemas-Distribuidos\Tarea\Tarea\_9\api-gateway>

**Figura 19.2 Construcción del Dockerfile**

#### Paso 4. Generar el keystore JKS para TLS (local, terminal)

- Ejecuta en tu PC (necesitas Java instalado) o en un entorno con keytool:

- keytool -genkeypair -alias apigw -keyalg RSA -keystore keystore\_servidor.jks -storepass 1234567 -keypass 1234567 -dname "CN=apigw"
  - Se generará el archivo keystore\_servidor.jks en tu carpeta actual.



### **Figura 19.3 Generar el keystore JKS para TLS**

Paso 5. Crear el Secret en Kubernetes con el keystore (kubectl: Cloud Shell o local)

- Asegúrate del namespace:
    - `kubectl create namespace t9-2022630278` (si aún no existe)
  - Crea el secreto (ejecuta en la carpeta donde está el archivo .jks):
    - `kubectl create secret generic apigw-keystore -n t9-2022630278 --from-file=keystore_servidor.jks=./keystore_servidor.jks`
  - Verifica:
    - `kubectl get secret -n t9-2022630278 apigw-keystore`

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows files in the API-GATEWAY project, including Dockerfile, keystore\_servidor.jks, SimpleAPIGateway.class, SimpleAPIGateway\$Worker\_1.class, and SimpleAPIGateway\$Worker\_2.class.
- Dockerfile**: Contains the following code:

```

FROM eclipse-temurin:17-jdk
WORKDIR /app
COPY SimpleAPIGateway.java .
RUN javac SimpleAPIGateway.java
# No pongas secretos en ENV: se inyectan en Kubernetes
ENV keystore=/keystore/keystore_servidor.jks
ENV password=1234567
CMD ["java","SimpleAPIGateway"]

```
- TERMINAL**: Shows the command `kubectl` being used to create a namespace and a secret from the keystore:

```

PS C:\Users\ivan...\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\api-gateway> kubectl create namespace t9-2022630278
Error from server (AlreadyExists): namespaces "t9-2022630278" already exists
PS C:\Users\ivan...\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\api-gateway> kubectl create secret generic apigw-keystore -n t9-2022630278 --from-file=keysto...
secret/apigw-keystore created
PS C:\Users\ivan...\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_9\api-gateway> kubectl get secret -n t9-2022630278 apigw-keystore
NAME          TYPE        DATA   AGE
apigw-keystore  Opaque      1     8s

```
- STATUS BAR**: Shows the terminal has 8 lines, 3.1s execution time, and was run at 03:52 p.m. on 22/12/2025.

**Figura 19.4 Crear el Secret en Kubernetes con el keystore**

Paso 6. Despliegue del gateway y Service LoadBalancer (kubectl: VS Code o Cloud Shell)

- Crea el manifiesto YAML y aplícalo.

Aplica el manifiesto y `kubectl apply -f k8s/apigw-deploy-lb.yaml`:

```
gustavo [ ~ ]$ cat > k8s/apigw-deploy-lb.yaml <<'EOF'  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: t9-2022630278-apigw  
  namespace: t9-2022630278  
spec:  
  replicas: 2  
  selector:  
    matchLabels: { app: t9-apigw }  
  template:  
    metadata:  
      labels: { app: t9-apigw }  
    spec:  
      containers:  
      - name: apigw  
        image: t92022630278acr.azurecr.io/simple-api-gateway:latest  
        ports:  
        - containerPort: 443  
        env:  
        - name: keystore  
          value: /keystore/keystore_servidor.jks  
        - name: password  
          value: "1234567"  
        volumeMounts:  
        - name: keystore  
          mountPath: /keystore  
      readinessProbe:  
        tcpSocket: { port: 443 }  
        initialDelaySeconds: 5  
        periodSeconds: 10  
      livenessProbe:  
        tcpSocket: { port: 443 }  
        initialDelaySeconds: 15  
        periodSeconds: 20  
      volumes:  
      - name: keystore  
        secret:  
          secretName: apigw-keystore  
---  
apiVersion: v1
```

```
volumeMounts:  
- name: keystore  
  mountPath: /keystore  
readinessProbe:  
  tcpSocket: { port: 443 }  
  initialDelaySeconds: 5  
  periodSeconds: 10  
livenessProbe:  
  tcpSocket: { port: 443 }  
  initialDelaySeconds: 15  
  periodSeconds: 20  
volumes:  
- name: keystore  
  secret:  
    secretName: apigw-keystore  
---  
apiVersion: v1  
EOF targetPort: 443t9-apigw }vc  
gustavo [ ~ ]$ kubectl apply -f k8s/apigw-deploy-lb.yaml  
deployment.apps/t9-2022630278-apigw created  
service/t9-2022630278-apigw-svc created  
gustavo [ ~ ]$ kubectl get pods -n t9-2022630278 -l app=t9-apigw  
NAME t9-2022630278-apigw-5fbfb84c465-6r5sf t9-2022630278-apigw-5fbfb84c465-rq9f5 READY 0/1 0/1 STATUS Running Running RESTARTS 0 0 AGE 12s 12s
```

Figura 19.6 Despliegue del gateway y Service LoadBalancer

Paso 7. Verificar despliegue y obtener la IP pública (kubectl)

- Estado de pods:

- kubectl get pods -n t9-2022630278 -l app=t9-apigw
- IP pública del LoadBalancer:
  - kubectl get svc -n t9-2022630278 t9-2022630278-apigw-svc
  - Copia el valor de EXTERNAL-IP (puede tardar 1–3 min)

```

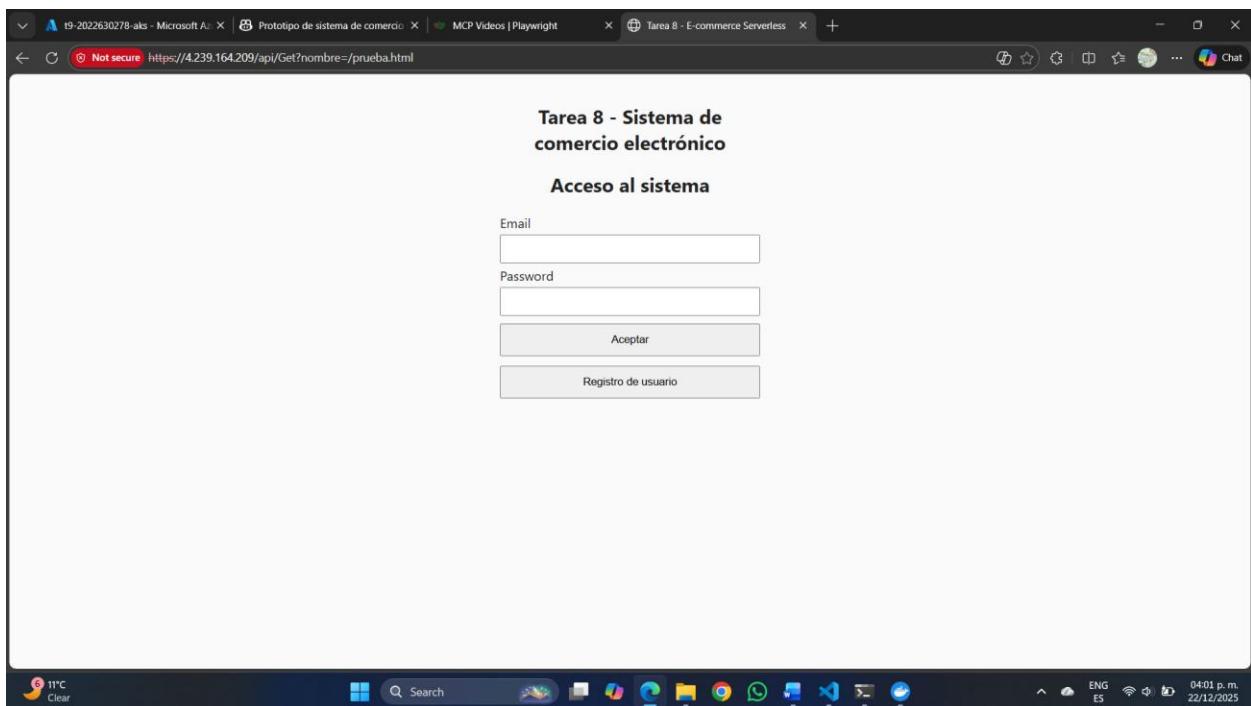
livenessProbe:
  tcpSocket: { port: 443 }
  initialDelaySeconds: 15
  periodSeconds: 20
volumes:
- name: keystore
  secret:
    secretName: apigw-keystore
---
apiVersion: v1
EOF targetPort: 443t9-apigw }vc
gustavo [ ~ ]$ kubectl apply -f k8s/apigw-deploy-lb.yaml
deployment.apps/t9-2022630278-apigw created
service/t9-2022630278-apigw-svc created
gustavo [ ~ ]$ kubectl get pods -n t9-2022630278 -l app=t9-apigw
NAME                    READY   STATUS    RESTARTS   AGE
t9-2022630278-apigw-5bfb84c465-6r5sf  0/1     Running   0          12s
t9-2022630278-apigw-5bfb84c465-rq9f5  0/1     Running   0          12s
gustavo [ ~ ]$ kubectl get svc -n t9-2022630278 t9-2022630278-apigw-svc
NAME              TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
t9-2022630278-apigw-svc   LoadBalancer   10.0.90.57   4.239.164.209   443:32074/TCP   2m9
s
gustavo [ ~ ]$ |

```

**Figura 19.8 Despliegue del gateway y Service LoadBalancer**

Paso 8. Probar fin a fin (navegador y curl)

- Navegador (PC o móvil):
  - URL: <https://<EXTERNAL-IP>/api/Get?nombre=/prueba.html>
  - Nota: el certificado es self-signed; acepta la advertencia para pruebas.



**Figura 20. Imagen del API Gateway publicada en ACR y lista para ser consumida por AKS**

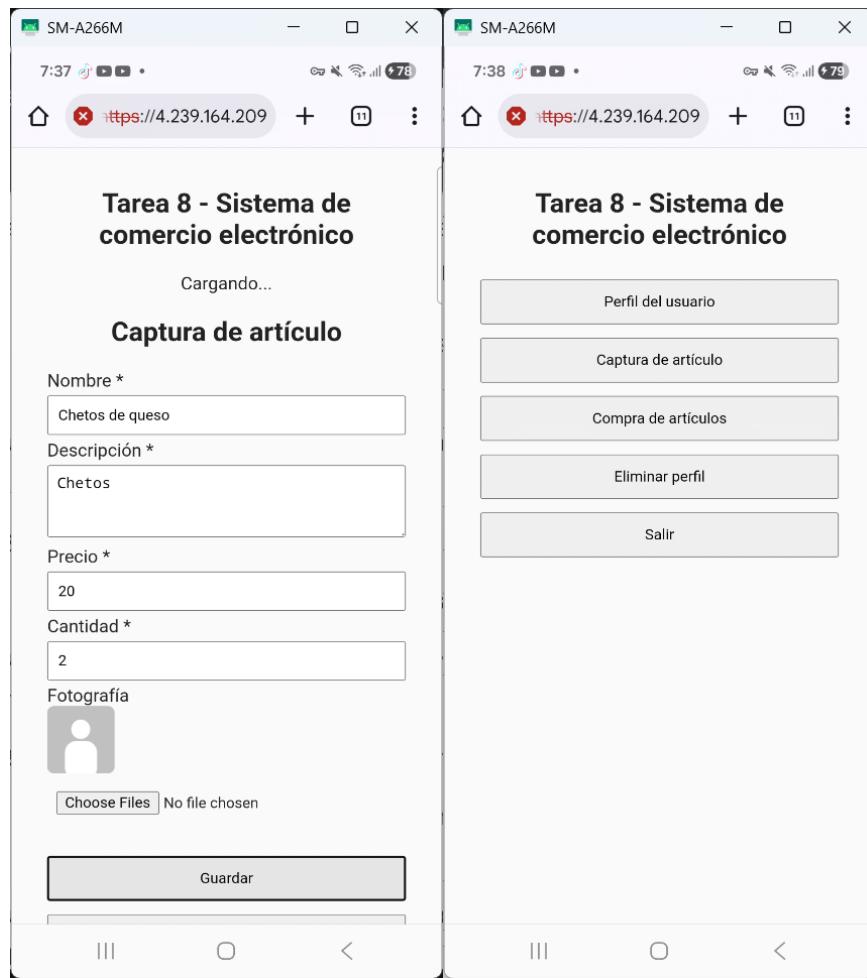
## **10 Evidencias requeridas**

Se realizó la recolección sistemática de evidencias para el reporte, cumpliendo estrictamente los lineamientos del curso: capturas completas (sin recortes), legibles, con la barra de estado/fecha/hora visibles y sin necesidad de zoom. Se instaló un orden lógico que inicia con variables/configuración en Azure/Kubernetes, continúa con pruebas del front-end en dispositivo móvil y finaliza con pruebas unitarias del back-end mediante curl. Se accedió a cada panel y a cada endpoint del sistema por medio del API Gateway, dejando constancia fotográfica inmediata después de cada paso.

### **10.1 Evidencias del front end en dispositivo móvil**

Para cada requerimiento funcional del front-end, incluya descripción breve y su captura:

1. Captura de artículo
  - Descripción: Pantalla “Captura de artículo” con nombre, descripción, precio, cantidad y fotografía.



**Figura 53. Pantalla “Captura de artículo” con formulario completado y acción Guardar exitosa**

2. Compra de artículos (búsqueda)
  - Descripción: Búsqueda por palabra clave en nombre/descripción (LIKE).



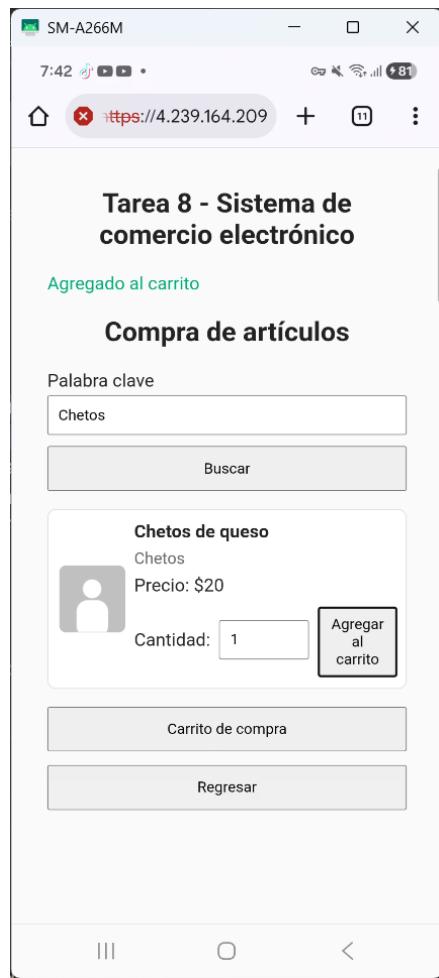
**Figura 54. Pantalla “Compra de artículos” con resultados de búsqueda por palabra clave**

3. Controles por artículo (+, -, cantidad y “Agregar al carrito”)
  - Descripción: Para cada resultado, campo “Cantidad” (default=1), botón “Agregar al carrito”.



**Figura 55. Tarjetas de artículo con foto, nombre, descripción, precio y controles de cantidad**

4. Compra (agregado al carrito)
  - Descripción: Al presionar “Compra”/“Agregar al carrito” se registra en backend GC.



**Figura 56. Confirmación “Agregado al carrito” tras la acción de compra**

##### 5. Carrito de compra

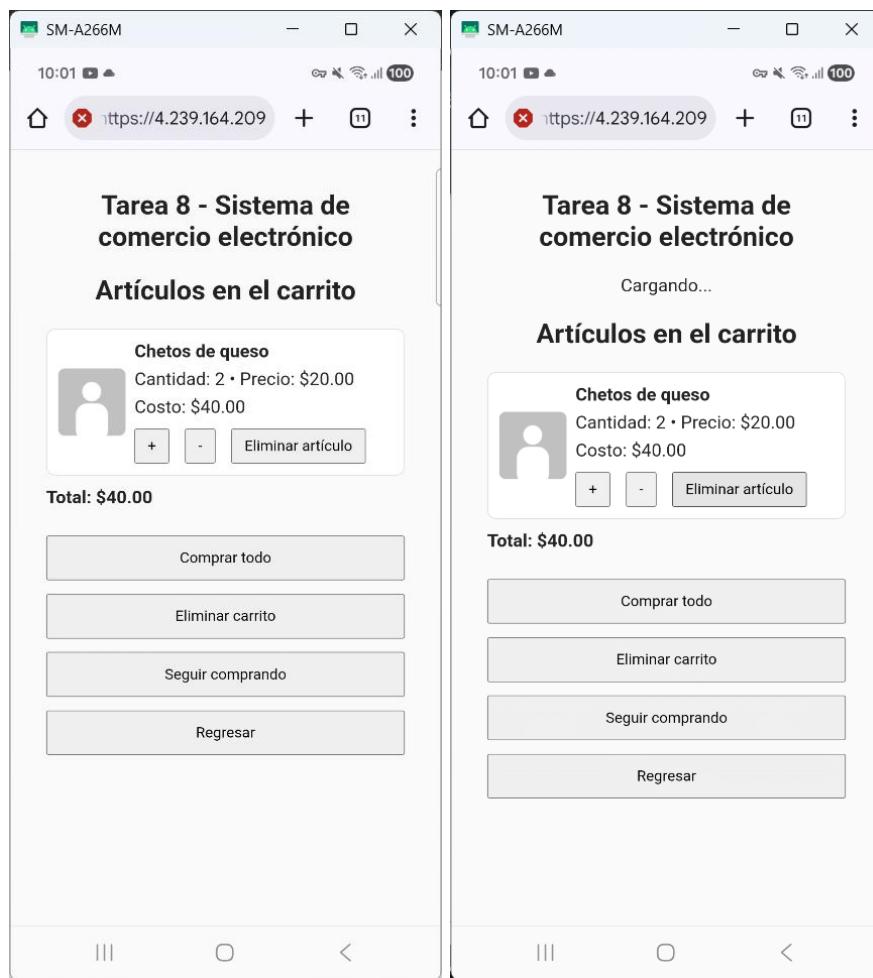
- Descripción: Pantalla “Artículos en el carrito” con imagen, nombre, cantidad, precio, costo y total.

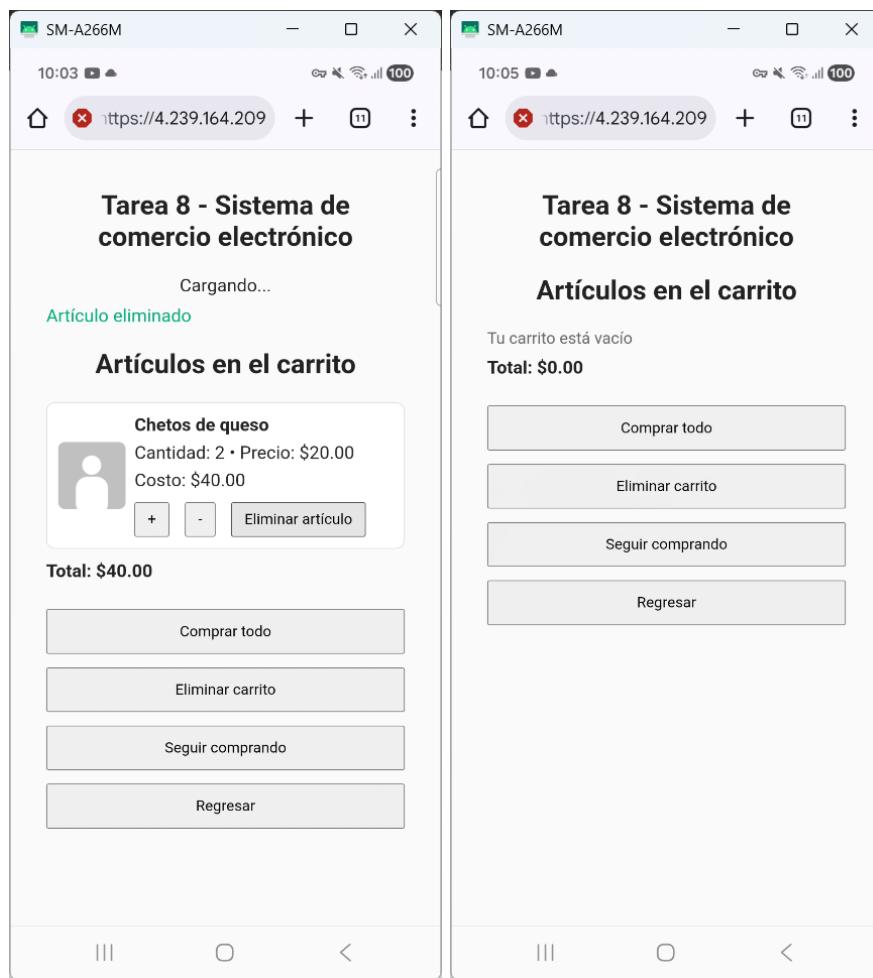


**Figura 57. Listado del carrito con imagen, cantidad, precio, costo por artículo y total**

6. Eliminar artículo del carrito

- Descripción: Botón “Eliminar artículo” que borra del carrito y regresa la cantidad al stock.

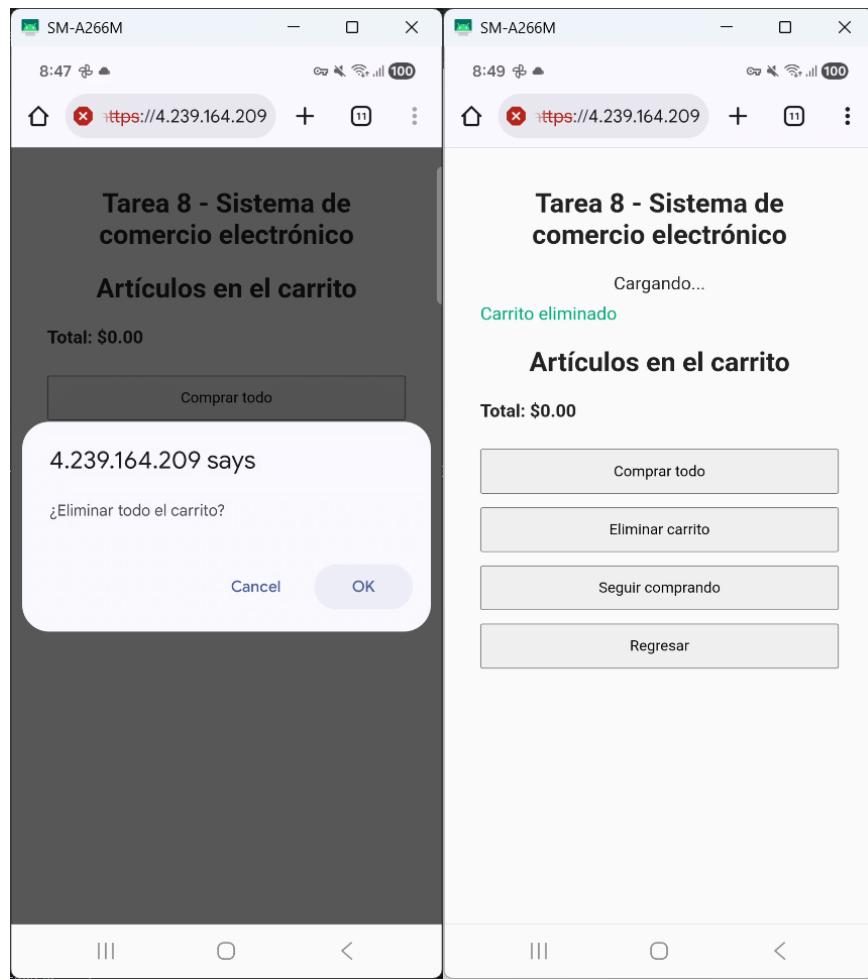




**Figura 58. Eliminación de artículo del carrito y actualización del total**

7. Eliminar carrito

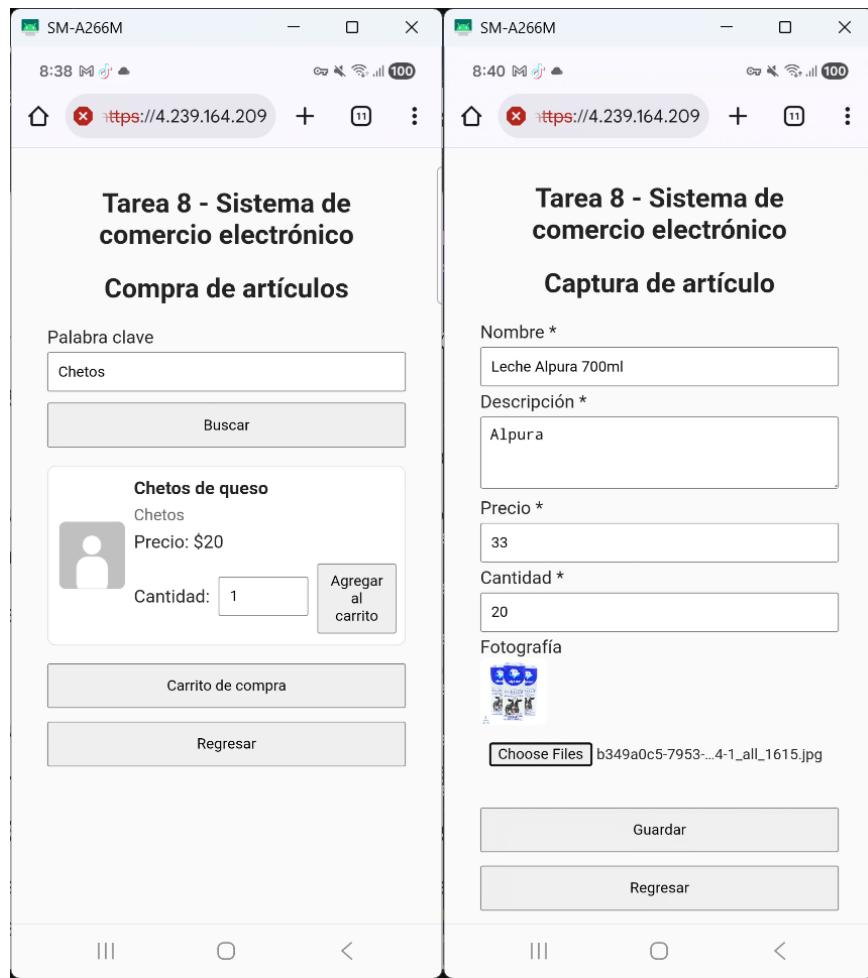
- Descripción: Botón “Eliminar carrito” que borra todo y regresa cantidades al stock.



**Figura 59. Eliminación del carrito completo y confirmación de carrito vacío**

8. Seguir comprando

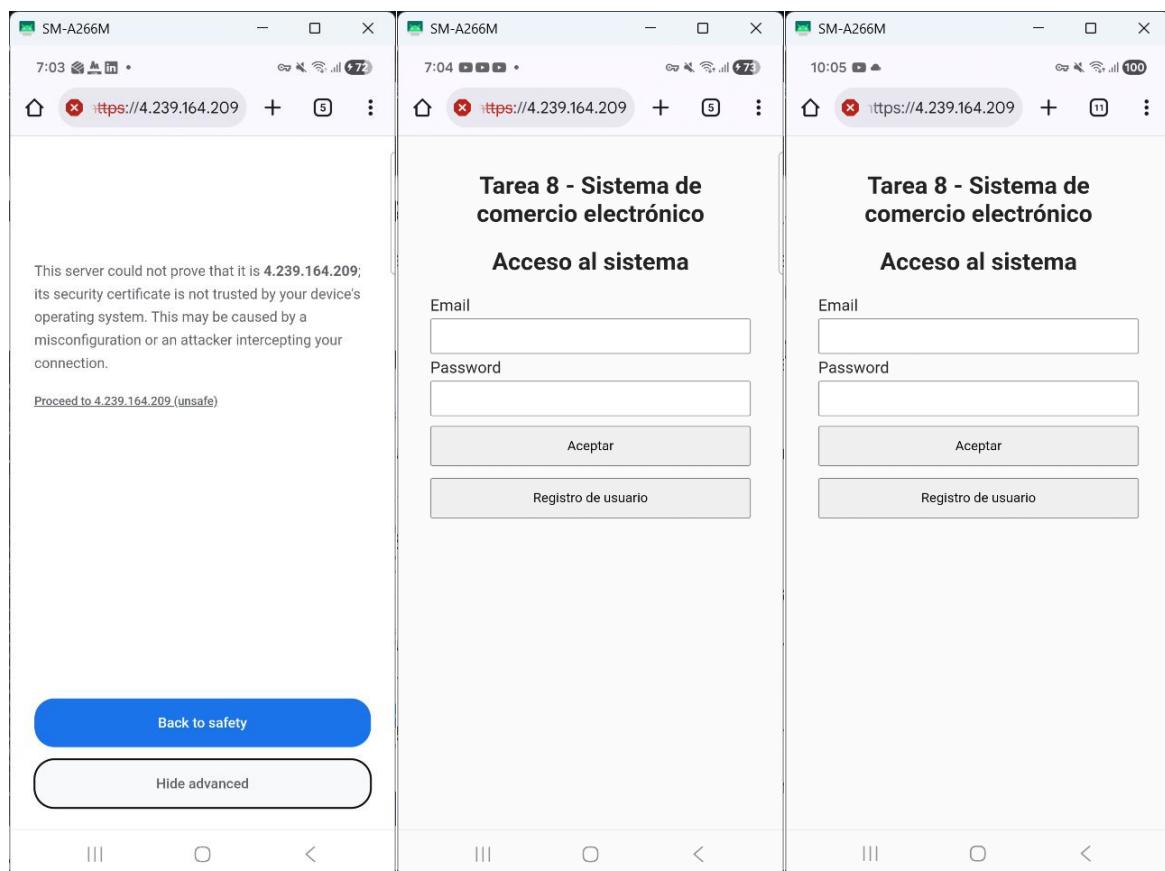
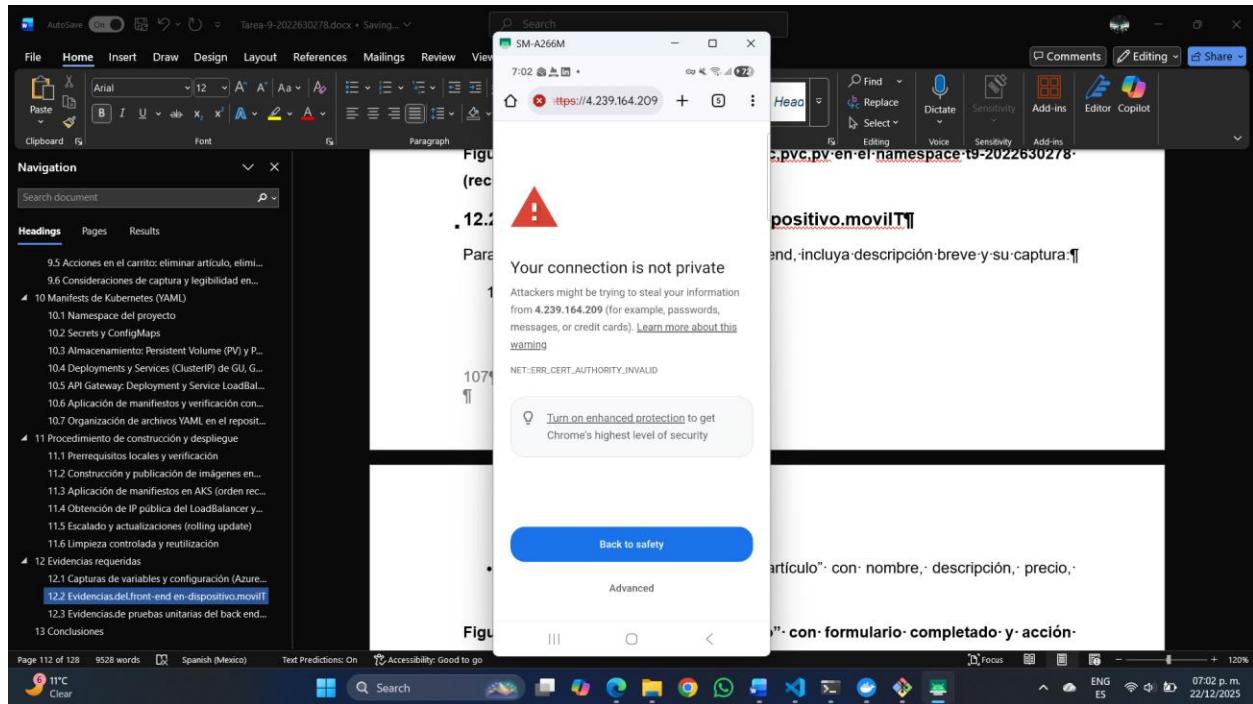
- Descripción: Botón “Seguir comprando” que regresa a “Compra de artículos”.

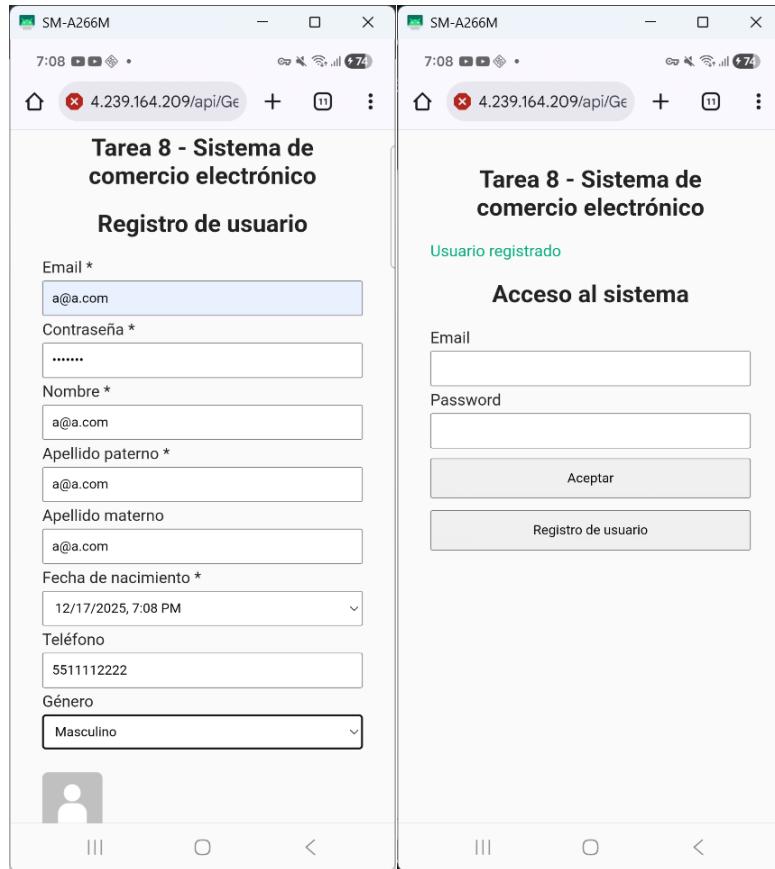


**Figura 60. Navegación “Seguir comprando” de la pantalla de carrito hacia compra de artículos**

9. Acceso mediante API Gateway:

- URL del front-end: <https://4.239.164.209/api/Get?nombre=/prueba.html>



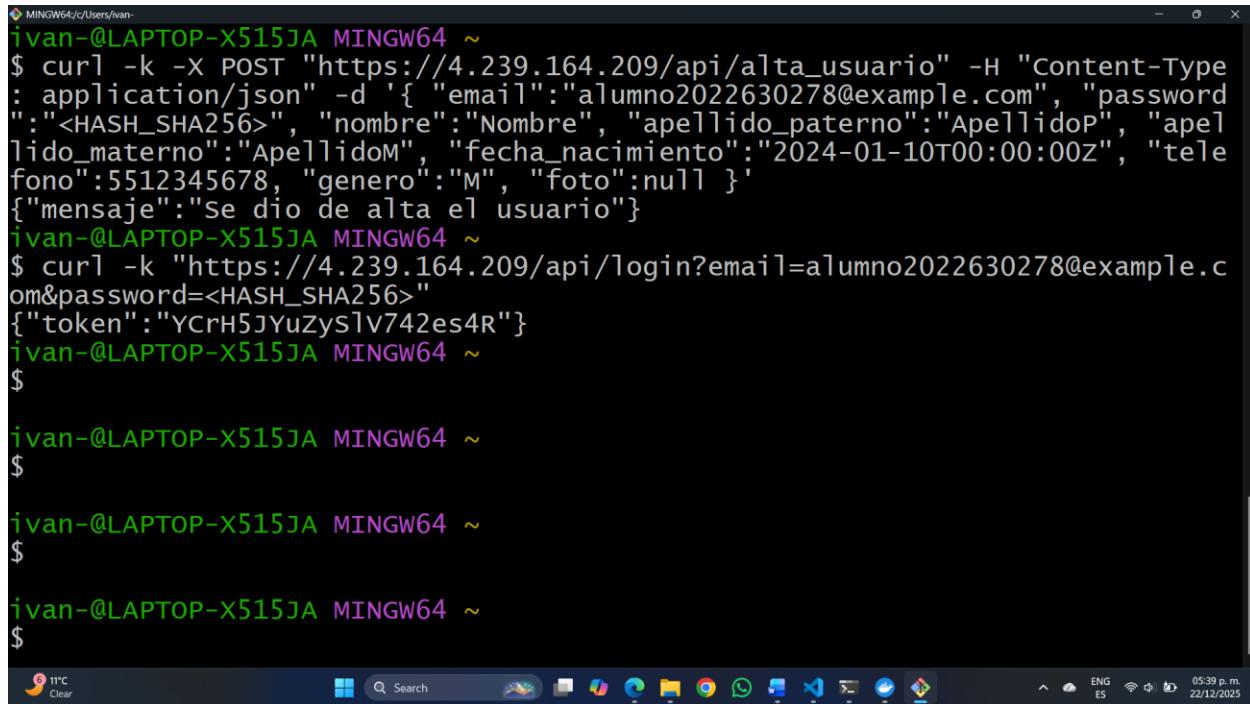


**Figura 61. Acceso al front-end vía API Gateway mostrando la URL <https://4.239.164.209/api/Get?nombre=/prueba.html>**

## 10.2 Evidencias de pruebas unitarias del back end (curl)

### 10.2.1 Alta usuario (GU):

```
curl -k -X POST "https://4.239.164.209/api/alta_usuario" -H "Content-Type: application/json" -d '{"email":"alumno2022630278@example.com","password":"5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8","nombre":"Nombre","apellido_paterno":"ApellidoP","apellido_materno":"ApellidoM","fecha_nacimiento":"2024-01-10T00:00:00Z","telefono":5512345678,"genero":"M","foto":null}'
```



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/alta_usuario" -H "Content-Type: application/json" -d '{ "email":"alumno2022630278@example.com", "password": "<HASH_SHA256>", "nombre":"Nombre", "apellido_paterno":"ApellidoP", "apellido_materno":"ApellidoM", "fecha_nacimiento":"2024-01-10T00:00:00Z", "telefono": 5512345678, "genero": "M", "foto": null }'
{"mensaje": "Se dio de alta el usuario"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/login?email=alumno2022630278@example.com&password=<HASH_SHA256>" 
{"token": "YCrH5JYuzy8lV742es4R"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

The screenshot shows a Windows terminal window titled 'MINGW64/c/Users/ivan-' with a black background. It displays two consecutive curl commands. The first command performs a POST request to 'https://4.239.164.209/api/alta\_usuario' with a JSON payload containing user information like email, password, names, and phone number. The response is a success message: '{"mensaje": "Se dio de alta el usuario"}'. The second command performs a GET request to 'https://4.239.164.209/api/login' with parameters 'email' and 'password' set to the values from the first command. The response is a token: '{"token": "YCrH5JYuzy8lV742es4R"}'. Below the terminal window, the Windows taskbar is visible, showing the Start button, a search bar, and various pinned icons for apps like File Explorer, Edge, and File History. The system tray shows the date and time as '22/12/2025 05:39 p.m.' and language settings as 'ENG ES'.

Figura 62. curl de alta\_usuario (GU) en una sola línea mostrando respuesta 200

### 10.2.2 Login (GU):

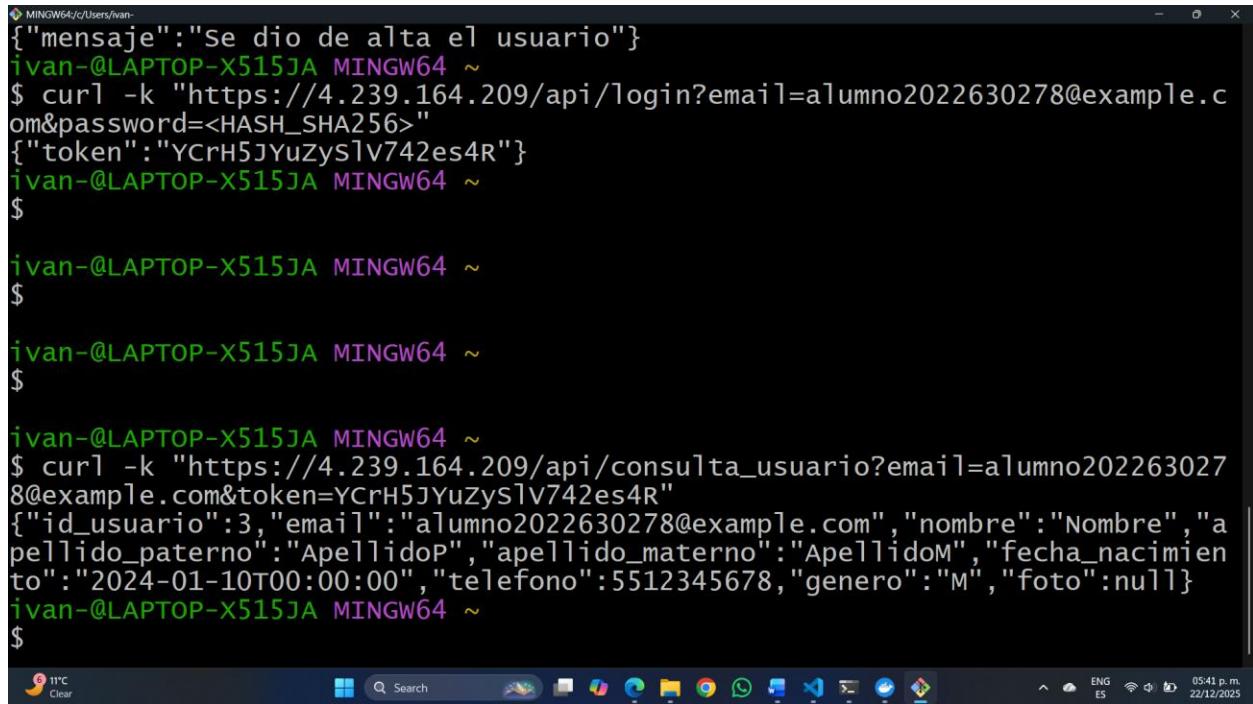
```
curl https://4.239.164.209/api/login?email=alumno2022630278@example.com&password=<HASH_SHA256> -k
```

```
MINGW64/c/Users/ivan-  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X POST "https://4.239.164.209/api/alta_usuario" -H "Content-Type": "application/json" -d '{ "email":"alumno2022630278@example.com", "password": "<HASH_SHA256>", "nombre":"Nombre", "apellido_paterno":"ApellidoP", "apellido_materno":"ApellidoM", "fecha_nacimiento":"2024-01-10T00:00:00Z", "telefono": 5512345678, "genero": "M", "foto": null }'  
{"mensaje": "Se dio de alta el usuario"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/login?email=alumno2022630278@example.com&password=<HASH_SHA256>"  
{"token": "YCrH5JYuZyslV742es4R"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ |
```

Figura 63. curl de login (GU) devolviendo token

### 10.2.3 Consulta usuario (GU):

```
curl -k  
"https://4.239.164.209/api/consulta_usuario?email=alumno2022630278@example.com  
&token=YCrH5JYuZyslV742es4R"
```



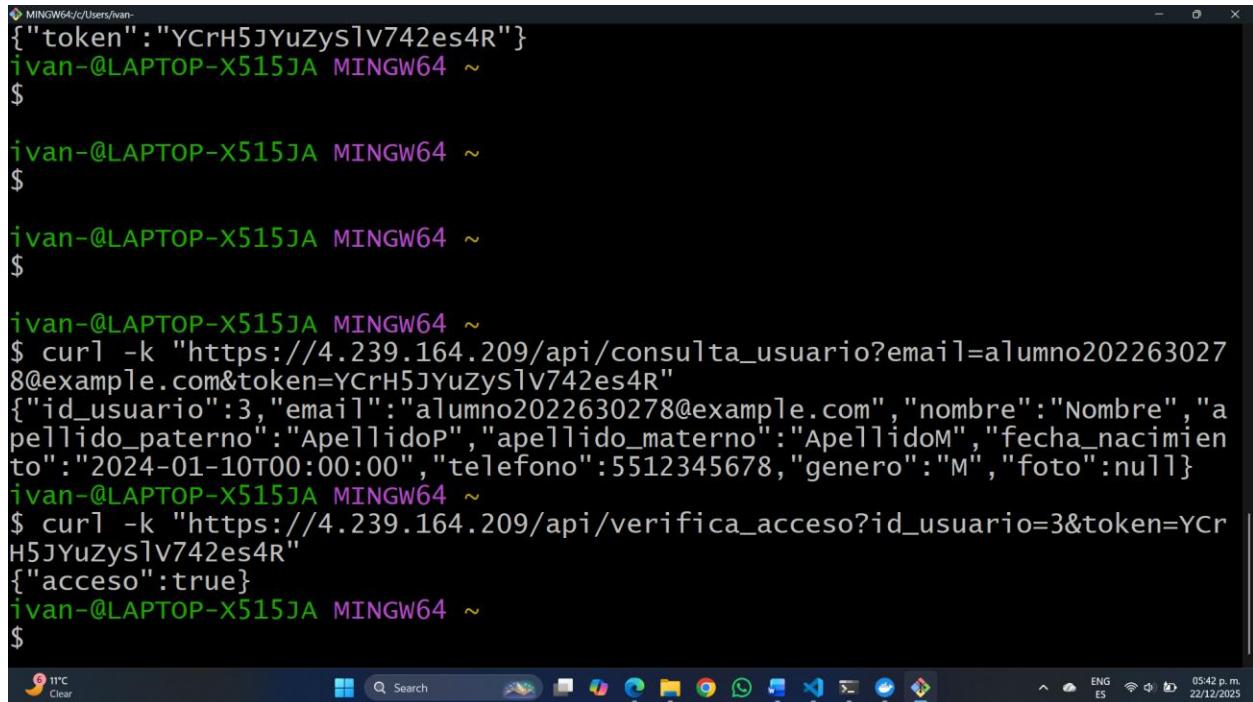
A screenshot of a Windows desktop environment showing a terminal window titled "MINGW64/c/Users/ivan-". The terminal displays a series of curl commands and their outputs. The first command logs in a user with email "alumno2022630278@example.com" and password "<HASH\_SHA256>". It returns a message "se dio de alta el usuario" and a token "YCrH5JYuZys1v742es4R". Subsequent commands show the user navigating through the system, eventually reaching a command to "curl -k https://4.239.164.209/api/consulta\_usuario?email=alumno2022630278@example.com&token=YCrH5JYuZys1v742es4R". This command returns a JSON object containing user profile information: id\_usuario: 3, email: "alumno2022630278@example.com", nombre: "Nombre", apellido\_paterno: "ApellidoP", apellido\_materno: "ApellidoM", fecha\_nacimiento: "2024-01-10T00:00:00", telefono: 5512345678, genero: "M", foto: null. The terminal ends with a prompt "\$". The desktop taskbar at the bottom shows various icons for applications like File Explorer, Edge, and Google Chrome.

```
{"mensaje":"se dio de alta el usuario"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/login?email=alumno2022630278@example.com&password=<HASH_SHA256>"  
{"token":"YCrH5JYuZys1v742es4R"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_usuario?email=alumno2022630278@example.com&token=YCrH5JYuZys1v742es4R"  
{"id_usuario":3,"email":"alumno2022630278@example.com","nombre":"Nombre","apellido_paterno":"ApellidoP","apellido_materno":"ApellidoM","fecha_nacimiento":"2024-01-10T00:00:00","telefono":5512345678,"genero":"M","foto":null}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$
```

Figura 64. curl de consulta\_usuario (GU) mostrando id\_usuario y perfil

#### 10.2.4 Verifica acceso (GU):

```
curl https://4.239.164.209/api/verifica_acceso?id_usuario=3&token=YCrH5JYuZySIV742es4R" -k
```

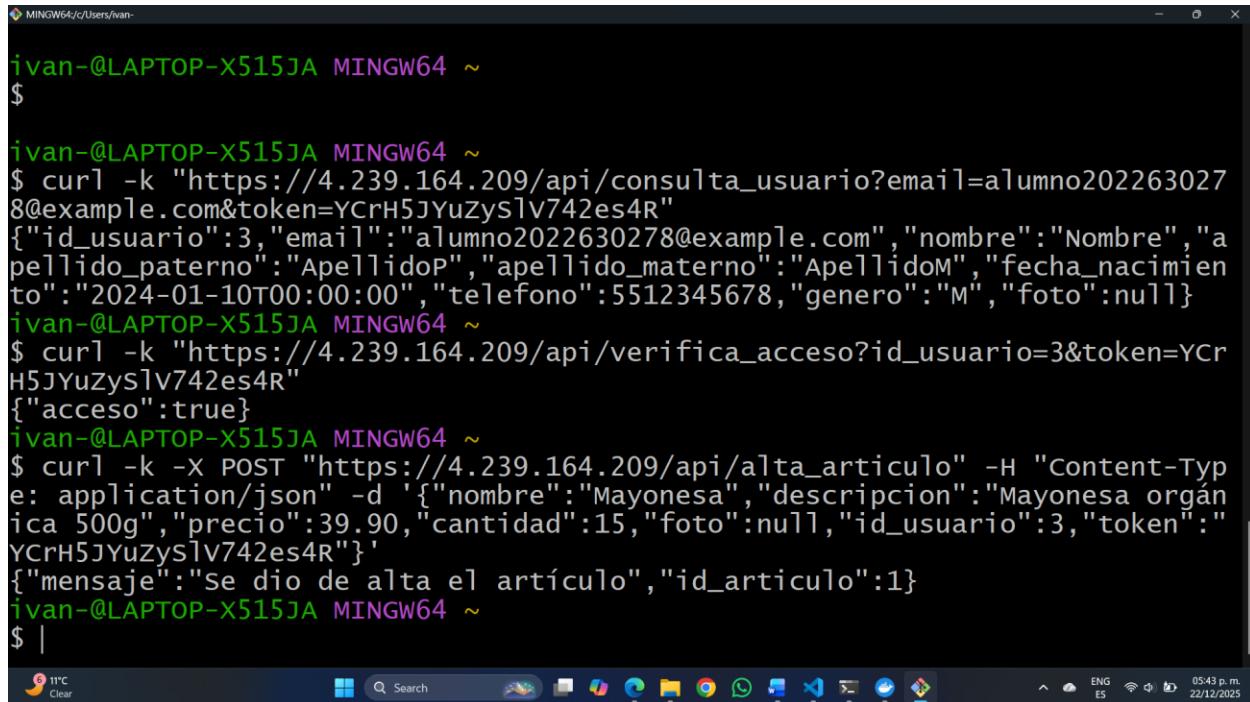


```
MINGW64/c/Users/ivan-{"token":"YCrH5JYuZys1v742es4R"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_usuario?email=alumno202263027  
8@example.com&token=YCrH5JYuZys1v742es4R"  
{"id_usuario":3,"email":"alumno2022630278@example.com","nombre":"Nombre","a  
pellido_paterno":"ApellidoP","apellido_materno":"ApellidoM","fecha_nacimiento":  
"2024-01-10T00:00:00","telefono":5512345678,"genero":"M","foto":null}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/verifica_acceso?id_usuario=3&token=YCr  
H5JYuZys1v742es4R"  
{"acceso":true}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$
```

Figura 65. curl de verifica\_acceso (GU) con validación 200/400

#### 10.2.5 Alta artículo (GA) — inserta artículo y llama a GC.alta\_articulo:

```
curl -k -X POST "https://4.239.164.209/api/alta_articulo" -H "Content-Type:  
application/json" -d '{"nombre":"Mayonesa","descripcion":"Mayonesa orgánica  
500g","precio":39.90,"cantidad":15,"foto":null,"id_usuario":3,"token":"YCrH5JYuZySIV74  
2es4R"}'
```

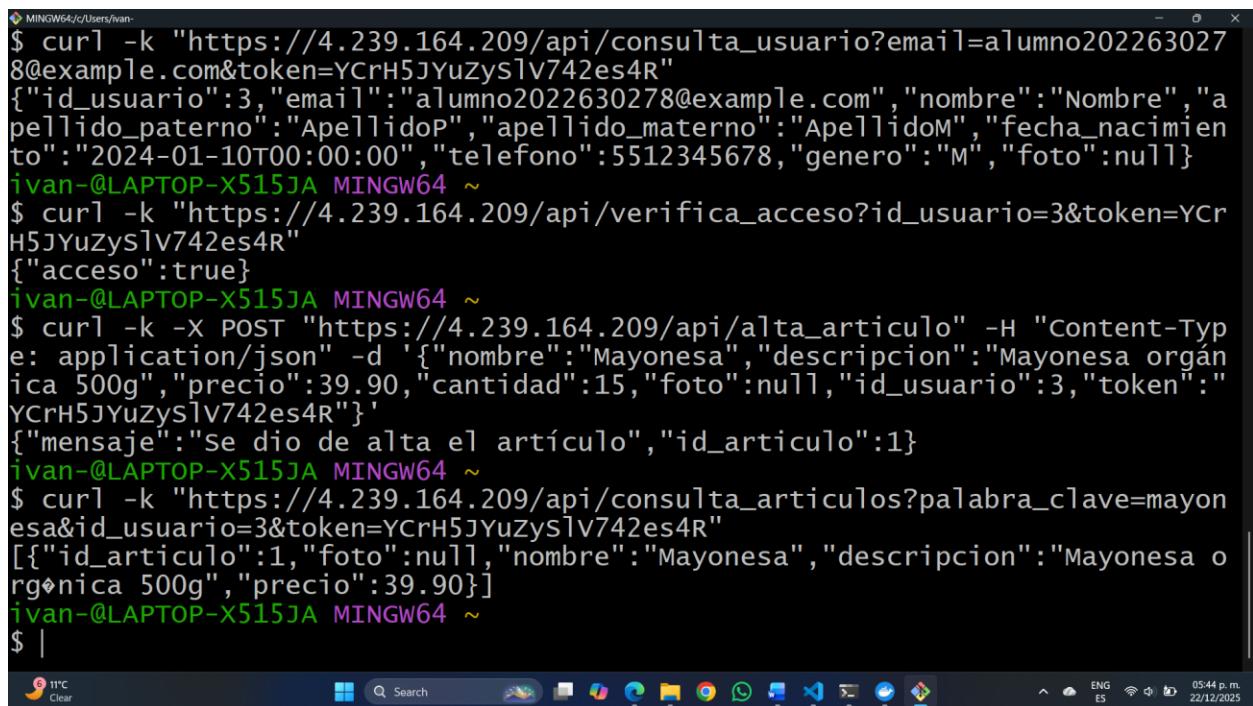


```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_usuario?email=alumno202263027
8@example.com&token=YCrH5JYuZys1v742es4R"
{"id_usuario":3,"email":"alumno2022630278@example.com","nombre":"Nombre","a
pellido_paterno":"ApellidoP","apellido_materno":"ApellidoM","fecha_nacimiento":
"2024-01-10T00:00:00","telefono":5512345678,"genero":"M","foto":null}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/verifica_acceso?id_usuario=3&token=YCr
H5JYuZys1v742es4R"
{"acceso":true}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/alta_articulo" -H "Content-Typ
e: application/json" -d '{"nombre":"Mayonesa","descripcion":"Mayonesa orgán
ica 500g","precio":39.90,"cantidad":15,"foto":null,"id_usuario":3,"token":"
YCrH5JYuZys1v742es4R"}'
>{"mensaje":"Se dio de alta el artículo","id_articulo":1}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Figura 66. curl de alta\_articulo (GA) mostrando id\_articulo creado

### 10.2.6 Consulta artículos (GA):

```
curl -k
"https://4.239.164.209/api/consulta_articulos?palabra_clave=mayonesa&id_usuario=3&
token=YCrH5JYuZys1v742es4R"
```



```
$ curl -k "https://4.239.164.209/api/consulta_usuario?email=alumno2022630278@example.com&token=YCrH5JYuZyslV742es4R"
{"id_usuario":3,"email":"alumno2022630278@example.com","nombre":"Nombre","apellido_paterno":"ApellidoP","apellido_materno":"ApellidoM","fecha_nacimiento":"2024-01-10T00:00:00","telefono":5512345678,"genero":"M","foto":null}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/verifica_acceso?id_usuario=3&token=YCrH5JYuZyslV742es4R"
{"acceso":true}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/alta_articulo" -H "Content-Type: application/json" -d '{"nombre":"Mayonesa","descripcion":"Mayonesa orgánica 500g","precio":39.90,"cantidad":15,"foto":null,"id_usuario":3,"token":"YCrH5JYuZyslV742es4R"}'
{"mensaje":"Se dio de alta el artículo","id_articulo":1}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_articulos?palabra_clave=mayonesa&id_usuario=3&token=YCrH5JYuZyslV742es4R"
[{"id_articulo":1,"foto":null,"nombre":"Mayonesa","descripcion":"Mayonesa orgánica 500g","precio":39.90}]
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Figura 67. curl de consulta\_articulos (GA) regresando arreglo de resultados

### 10.2.7 Compra artículo (GC):

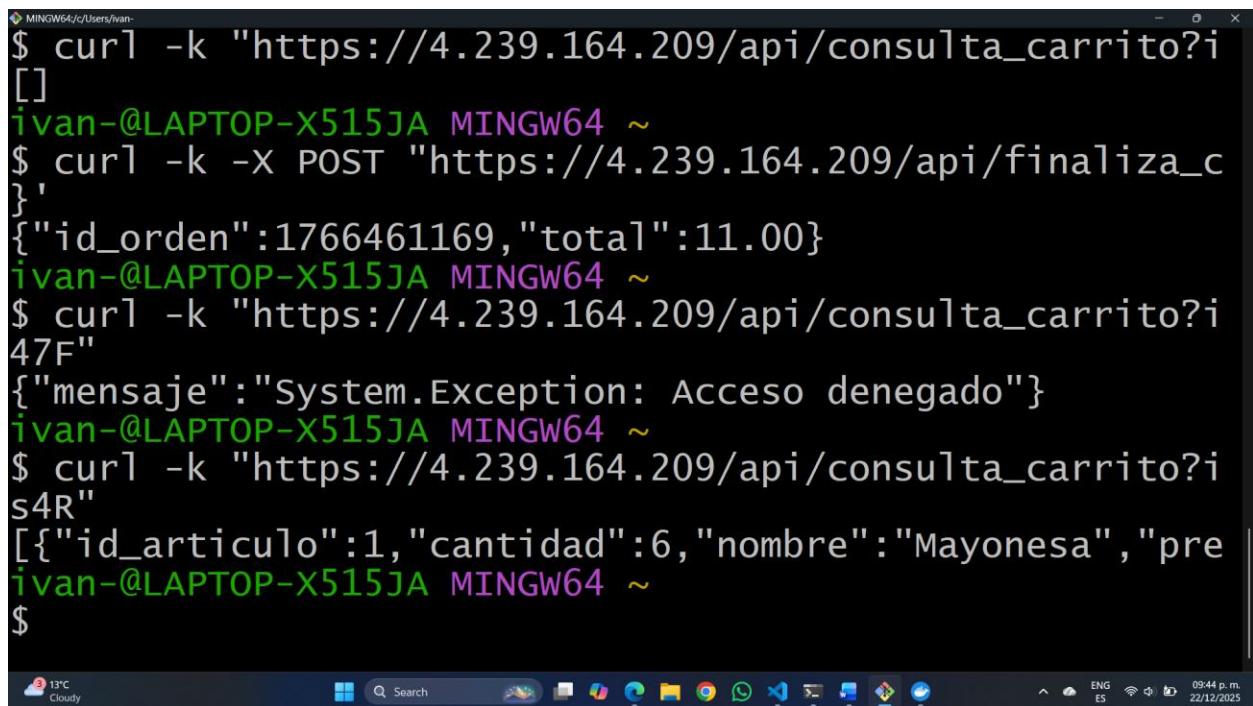
```
curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":1,"cantidad":2,"id_usuario":3,"token":"YCrH5JYuZyslV742es4R"}'
```

```
$ curl -k -X POST "https://4.239.164.209/api/alta_articulo" -H "Content-Type: application/json" -d '{"nombre":"Mayonesa", "descripcion":"Mayonesa orgánica 500g", "precio":39.90, "cantidad":15, "foto":null, "id_usuario":3, "token":"YCrH5JYuZyslV742es4R"}'  
{"mensaje":"Se dio de alta el artículo", "id_articulo":1}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_articulos?palabra_clave=mayonesa&id_usuario=3&token=YCrH5JYuZyslV742es4R"  
[{"id_articulo":1, "foto":null, "nombre": "Mayonesa", "descripcion": "Mayonesa orgánica 500g", "precio":39.90}]  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":501, "cantidad":2, "id_usuario":3, "token": "YCrH5JYuZyslV742es4R"}'  
{"mensaje": "El artículo no existe"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":1, "cantidad":2, "id_usuario":3, "token": "YCrH5JYuZyslV742es4R"}'  
{"mensaje": "Compra registrada"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$
```

Figura 68. curl de compra\_articulo (GC) con respuesta 200 o 400 por stock

### 10.2.8 Consulta carrito (GC):

```
curl -k  
"https://4.239.164.209/api/consulta_carrito?id_usuario=3&token=YCrH5JYuZySlV742es4R"
```

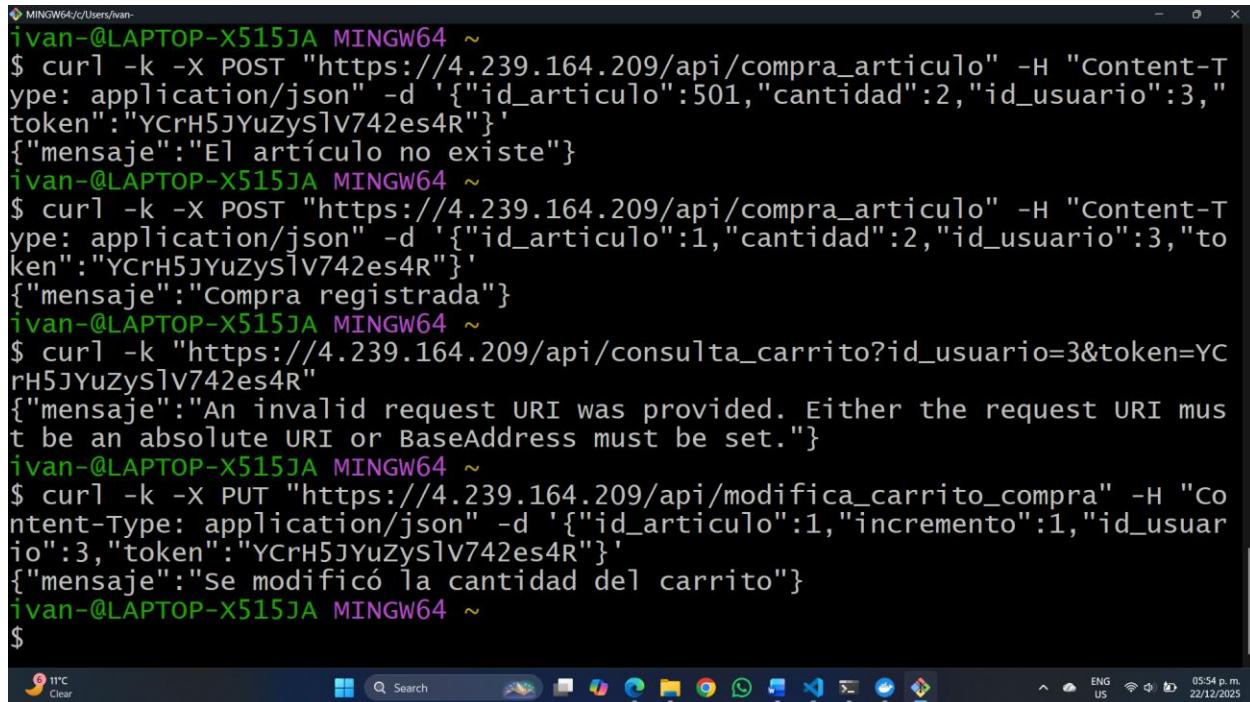


```
MINGW64/c/Users/ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_carrito?i
[]
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/finaliza_c
}
{"id_orden":1766461169,"total":11.00}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_carrito?i
47F"
>{"mensaje":"System.Exception: Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_carrito?i
s4R"
[{"id_articulo":1,"cantidad":6,"nombre":"Mayonesa","pre
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Figura 69. curl de consulta\_carrito (GC) mostrando artículos y costos

#### 10.2.9 Modificar carrito (+1) (GC):

```
curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":1,"id_usuario":3,"token":"YCrH5JYuZySIV742es4R"}'
```

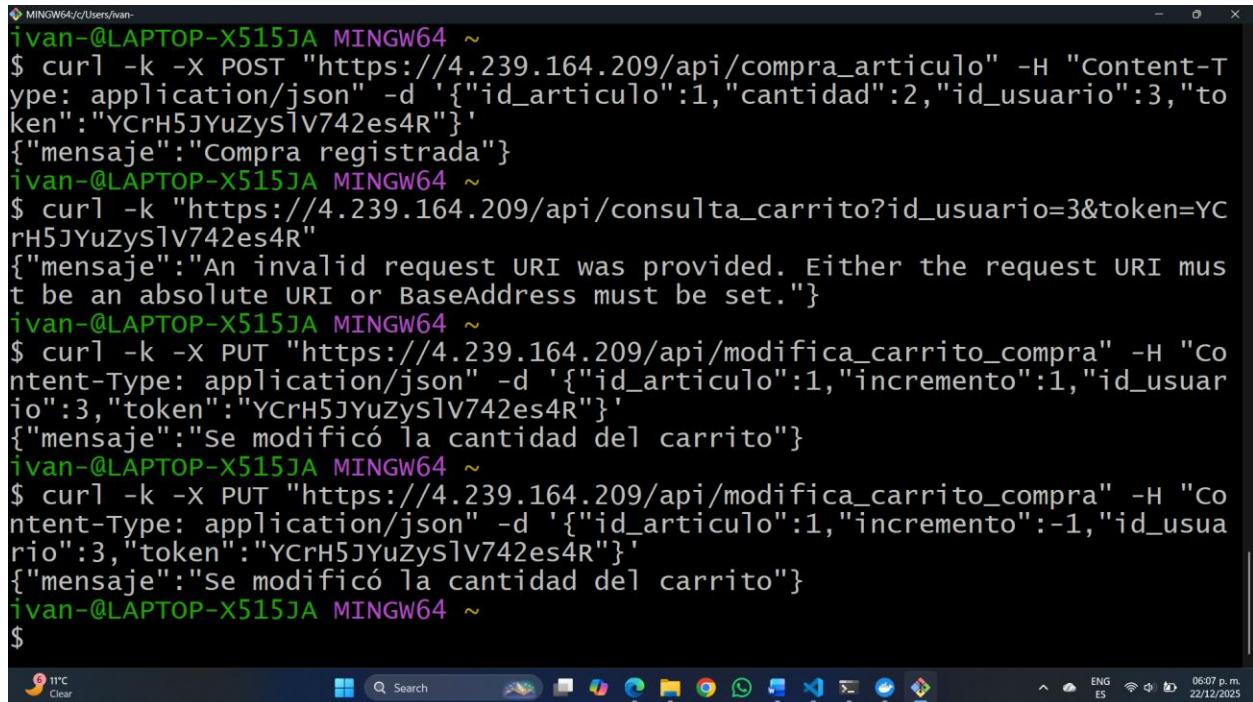


```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":501,"cantidad":2,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
>{"mensaje":"El artículo no existe"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":1,"cantidad":2,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
>{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_carrito?id_usuario=3&token=YCrH5JYuZys1v742es4R"
>{"mensaje":"An invalid request URI was provided. Either the request URI must be an absolute URI or BaseAddress must be set."}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":1,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
>{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Figura 70. curl de modifica\_carrito\_compra (+1) ajustando carrito y stock

#### 10.2.10 Modificar carrito (-1) (GC):

```
curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":1,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
```



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":1,"cantidad":2,"id_usuario":3,"token":"YCrH5JYuZyslV742es4R"}'
{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k "https://4.239.164.209/api/consulta_carrito?id_usuario=3&token=YCrH5JYuZyslV742es4R"
{"mensaje":"An invalid request URI was provided. Either the request URI must be an absolute URI or BaseAddress must be set."}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":1,"id_usuario":3,"token":"YCrH5JYuZyslV742es4R"}'
{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":-1,"id_usuario":3,"token":"YCrH5JYuZyslV742es4R"}'
{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Figura 71. curl de `modifica_carrito_compra` (-1) con validación “No hay más artículos”

### 10.2.11 Eliminar artículo del carrito (GC):

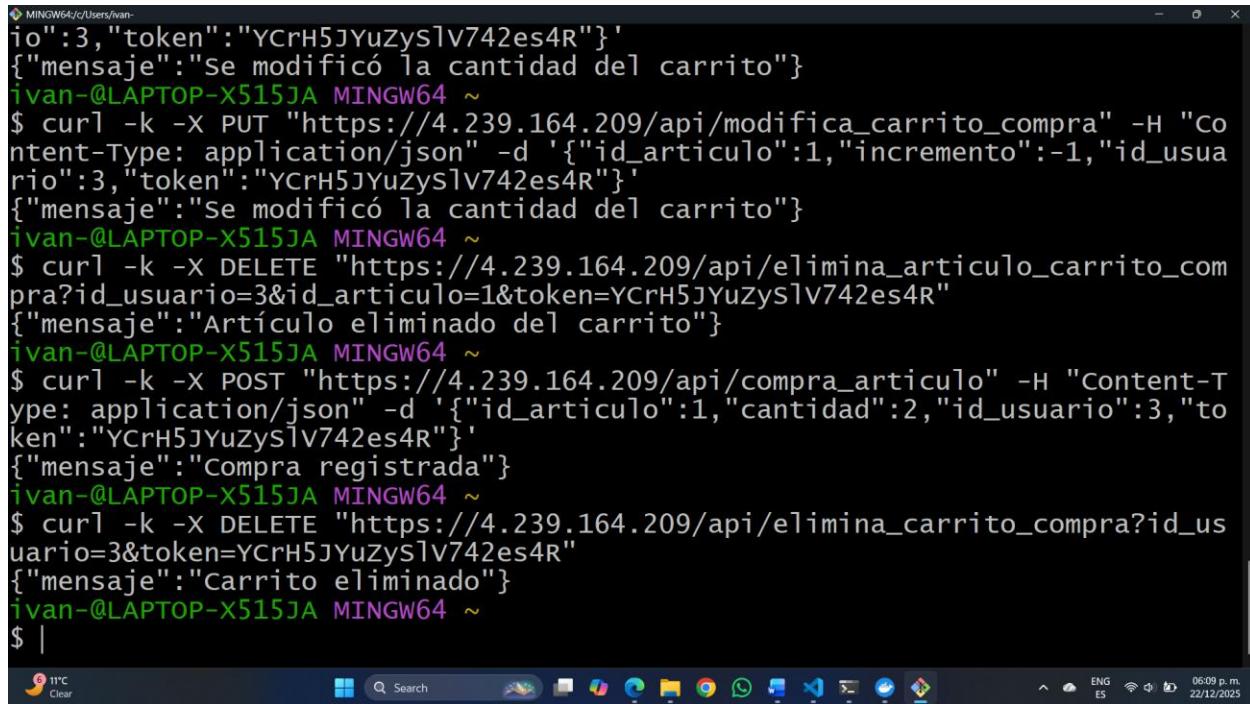
```
curl           -k           -X           DELETE
"https://4.239.164.209/api/elimina_articulo_carrito_compra?id_usuario=3&id_articulo=1
&token=YCrH5JYuZyslV742es4R"
```

```
MINGW64/c/Users/ivan-  
{"mensaje":"Compra registrada"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_carrito?id_usuario=3&token=YCrH5JYuZys1v742es4R"  
>{"mensaje":"An invalid request URI was provided. Either the request URI must be an absolute URI or BaseAddress must be set."}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":1,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'  
>{"mensaje":"Se modificó la cantidad del carrito"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":-1,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'  
>{"mensaje":"Se modificó la cantidad del carrito"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X DELETE "https://4.239.164.209/api/elimina_articulo_carrito_compra?id_usuario=3&id_articulo=1&token=YCrH5JYuZys1v742es4R"  
>{"mensaje":"Artículo eliminado del carrito"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ |
```

Figura 72. curl de `elimina_articulo_carrito_compra` (GC) regresando cantidad al stock

### 10.2.12 Eliminar carrito (GC):

```
curl           -k           -X           DELETE  
"https://4.239.164.209/api/elimina_carrito_compra?id_usuario=3&token=YCrH5JYuZySI  
V742es4R"
```



```
MINGW64/c/Users/ivan-
{
    "mensaje": "Se modificó la cantidad del carrito"
}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X PUT "https://4.239.164.209/api/modifica_carrito_compra" -H "Content-Type: application/json" -d '{"id_articulo":1,"incremento":-1,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
{
    "mensaje": "Se modificó la cantidad del carrito"
}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X DELETE "https://4.239.164.209/api/elimina_articulo_carrito_compra?id_usuario=3&id_articulo=1&token=YCrH5JYuZys1v742es4R"
{
    "mensaje": "Artículo eliminado del carrito"
}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X POST "https://4.239.164.209/api/compra_articulo" -H "Content-Type: application/json" -d '{"id_articulo":1,"cantidad":2,"id_usuario":3,"token":"YCrH5JYuZys1v742es4R"}'
{
    "mensaje": "Compra registrada"
}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -k -X DELETE "https://4.239.164.209/api/elimina_carrito_compra?id_usuario=3&token=YCrH5JYuZys1v742es4R"
{
    "mensaje": "Carrito eliminado"
}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

**Figura 73. curl de elimina\_carrito\_compra (GC) vaciando el carrito y restaurando stock**

#### 10.2.13 Finaliza compra (GC, opcional):

```
curl           -k           -X           POST
"https://4.239.164.209/api/finaliza_compra?id_usuario=3&token=YCrH5JYuZys1v742es4R" -H "Content-Type: application/json" -d '{}'
```

```
MINGW64/c/Users/ivan-  
}'  
{"mensaje":"This instance has already started one or mo  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X POST "https://4.239.164.209/api/finaliza_c  
}'  
{"mensaje":"This instance has already started one or mo  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_carrito?i  
{"mensaje":"This instance has already started one or mo  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k "https://4.239.164.209/api/consulta_carrito?i  
[]  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -k -X POST "https://4.239.164.209/api/finaliza_c  
}'  
{"id_orden":1766461169,"total":11.00}
```

Figura 74. curl de finaliza\_compra (GC) creando orden y total de compra

## 11 Conclusiones

Se realizó exitosamente el prototipo de e-commerce basado en microservicios sobre AKS, cumpliendo los requerimientos funcionales y no funcionales, la separación de datos por servicio y la nomenclatura establecida. Se instaló un flujo de construcción y despliegue reproducible (build → push → apply → probar), y se accedió al sistema a través de un API Gateway que centraliza el enrutamiento y facilita el control de entrada.

La práctica demostró que la arquitectura propuesta es viable y escalable: cada microservicio presenta responsabilidades bien definidas, sus bases de datos están aisladas y las rutas internas se gestionan con Services ClusterIP, mientras el acceso público se realiza mediante LoadBalancer. El uso de Azure Files para el front-end simplificó la publicación de estáticos sin acoplarlos al contenedor del servidor web.

Se utilizó la IA de GitHub Copilot como apoyo en la generación de esqueletos de código, Dockerfiles y manifests de Kubernetes, acelerando el desarrollo y manteniendo la coherencia con la nomenclatura. Este apoyo complementó la comprensión técnica y las decisiones de diseño tomadas por el alumno.

Trabajo futuro recomendado:

- Se instalarán probes de liveness/readiness más específicos, límites/solicitudes de CPU/memoria y HPA para auto-escalado.
- Se realizará la integración de CI/CD (GitHub Actions) para automatizar builds y despliegues a ACR/AKS.
- Se accederán soluciones gestionadas de certificados y/o Ingress para TLS administrado.
- Se instalará observabilidad con Application Insights/Log Analytics y trazas entre servicios.

## **Enlace del chat de la IA generativa**

Enlace: <https://github.com/copilot/share/ca754016-4b40-8ca3-a851-a20e20b240d0>

## 12 Referencias (Formato IEEE)

- [1] Microsoft, “Azure Kubernetes Service (AKS) documentation,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/aks/>. Accedido: 21-Dic-2025.
- [2] Microsoft, “Azure Container Registry documentation,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/container-registry/>. Accedido: 21-Dic-2025.
- [3] Microsoft, “Azure Database for MySQL – Flexible Server,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/mysql/flexible-server/>. Accedido: 21-Dic-2025.
- [4] Microsoft, “Azure Files and CSI driver for Kubernetes,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/aks/azure-files-csi>. Accedido: 21-Dic-2025.
- [5] The Kubernetes Authors, “Kubernetes Concepts: Pods, Deployments, Services, Volumes,” Kubernetes Docs, 2025. [Online]. Available: <https://kubernetes.io/docs/>. Accedido: 21-Dic-2025.
- [6] Microsoft, “Azure Functions – .NET isolated worker model,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/azure-functions/dotnet-isolated-process-guide>. Accedido: 21-Dic-2025.
- [7] GitHub, “GitHub Copilot documentation,” GitHub Docs, 2025. [Online]. Available: <https://docs.github.com/copilot>. Accedido: 21-Dic-2025.
- [8] Oracle, “Class SSLServerSocket,” Java SE Documentation, 2025. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/javax/net/ssl/SSLServerSocket.html>. Accedido: 21-Dic-2025.
- [9] Oracle, “MySQL Connector/.NET and ADO.NET,” MySQL Docs, 2025. [Online]. Available: <https://dev.mysql.com/doc/connector-net/en/>. Accedido: 21-Dic-2025.
- [10] Microsoft, “Application Insights for Azure Functions,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/azure-functions/functions-monitoring>. Accedido: 21-Dic-2025.