



Instituto Politécnico Nacional
Escuela Superior de Computo



Sistemas Distribuidos

Tarea 7. Prototipo de un sistema de comercio electrónico utilizando un servicio web REST para Tomcat

Nombre del alumno:

García Quiroz Gustavo Ivan

Grupo: 7CV4

Nombre del profesor: Guerrero Carlos Pineda

Fecha de entrega: 23/11/2025

ÍNDICE

1	Resumen.....	1
2	Introducción	2
3	Objetivos	4
3.1	Objetivo general.....	4
3.2	Objetivos específicos	4
4	Requerimientos	5
4.1	Requerimientos funcionales del back-end	5
4.2	Requerimientos funcionales del front-end.....	6
4.3	Requerimientos no funcionales.....	7
5	Arquitectura del Sistema	9
5.1	Visión global por capas	9
5.2	Componentes principales	10
5.3	Flujo de autenticación y autorización	11
5.4	Flujo de compra y consistencia transaccional.....	11
5.5	Modelo lógico y relaciones (resumen conceptual)	13
6	Desarrollo.....	14
7	Infraestructura en Azure	14
7.1	Creación de la máquina virtual.....	14
7.2	Características y configuración técnica.....	17
8	Configuración del Entorno.....	20
8.1	Verificación / Instalación de Java	20
8.2	Verificación / Instalación de Tomcat	20
8.3	Base de datos (MySQL).....	21
8.3.1	Alterar tabla usuarios	21

8.4	Incorporación de librerías (Jersey/Jackson).....	25
8.5	Limpieza y despliegue base.....	26
8.5.1	Limpieza del despliegue antiguo del servicio	26
8.5.2	Preparar directorio de trabajo para compilar.....	27
8.5.3	Configurar context.xml (conexión a MySQL)	29
8.5.4	Configurar compila.sh	31
8.5.5	Compilar el servicio	33
8.5.6	Copiar archivos front-end (prueba.html, WSClient.js, usuario_sin_foto.png)	
	34	
8.5.7	8. (Opcional) Reiniciar Tomcat si hubo cambios críticos	35
8.5.8	Probar acceso desde navegador	36
8.6	Modificación/Actualización de Servicio.java.....	37
8.7	Scripts de compilación (compila.sh y compila.bat).....	43
9	Implementación del Back-End.....	47
9.1	Estructura del proyecto	47
9.2	Modificación del método login.....	48
9.3	Endpoints añadidos y mapeo a requerimientos funcionales	49
9.4	Correspondencia con requerimientos funcionales	49
10	Implementación del Front-end (prueba.html y JS) para la Tarea 7	51
10.1	Enfoque y organización	51
10.2	Funciones JavaScript clave	52
11	Endpoints REST	55
11.1	Convenciones generales	55
11.2	Endpoint: login	56
11.3	Endpoint: alta_usuario	56

11.4	Endpoint: consulta_usuario.....	56
11.5	Endpoint: modifica_usuario.....	57
11.6	Endpoint: borra_usuario.....	57
11.7	Endpoint: alta_articulo	57
11.8	Endpoint: consulta_articulos	58
11.9	Endpoint: compra_articulo	58
11.10	Endpoint: elimina_articulo_carrito_compra.....	59
11.11	Endpoint: elimina_carrito_compra	59
11.12	Endpoint: modifica_carrito_compra	59
11.13	Endpoint: consulta_carrito_compra	60
12	Pruebas Unitarias del Back-End (curl).....	61
12.1	Metodología y Preparación	61
12.2	Evidencias por Endpoint	62
12.2.1	login	63
12.2.2	alta_usuario	64
12.2.3	consulta_usuario.....	66
12.2.4	modifica_usuario.....	67
12.2.5	borra_usuario (opcional, si se prueba al final)	68
12.2.6	alta_articulo	70
12.2.7	consulta_articulos	72
12.2.8	compra_articulo	73
12.2.9	consulta_carrito_compra.....	76
12.2.10	elimina_articulo_carrito_compra	77
12.2.11	elimina_carrito_compra	79
12.3	Tabla Resumen de Pruebas.....	81

13	Pruebas Funcionales del Front-End (Móvil)	86
13.1	Dispositivo y Entorno	86
13.2	Casos de Prueba por Requerimiento Funcional	87
13.3	Validación de Mensajes y Errores.....	104
14	Enlace del chat de la IA generativa	105
15	Conclusiones	106
16	Referencias (Formato IEEE).....	108

1 Resumen

Este documento presenta el desarrollo de un prototipo de sistema de comercio electrónico sencillo, compuesto por un servicio web REST (back-end) desplegado en Apache Tomcat y una aplicación web HTML/JavaScript (front-end) ejecutada en un navegador móvil. El prototipo extiende la funcionalidad previa de gestión de usuarios (implementada en la Tarea 2) para incorporar operaciones de administración de artículos (alta y consulta), manipulación de un carrito de compra y control de transacciones sobre una base de datos MySQL alojada en una máquina virtual de Azure (VM nombrada conforme a la boleta del alumno).

Se integran mecanismos básicos de autenticación mediante tokens y manejo de contraseñas con hash SHA-256 (calculado en el cliente). La arquitectura sigue un enfoque multicapa mínimo y agrupa las operaciones críticas (compra, modificación y eliminación) dentro de transacciones para garantizar consistencia entre el stock y el carrito. El resultado es una base sólida para evolucionar hacia soluciones más completas (pasarela de pagos, catálogo avanzado o refactor a frameworks modernos).

La herramienta de IA GitHub Copilot se utilizó como apoyo puntual en la generación de fragmentos de código y en la estructuración acelerada del back-end y partes del front-end, manteniendo supervisión humana para validación funcional y semántica.

NOTA: La plataforma de Moodle no permitió subir el reporte en buena calidad debido al límite de 2 MB por archivo. Sugiero revisar el siguiente documento de Google drive con el reporte en buena calidad:
https://drive.google.com/file/d/12RmsPrHCOPWLXeO1s6OF_BQuNRVdosLM/view?usp=sharing

2 Introducción

La Tarea 7 se orienta a la creación de un prototipo de comercio electrónico elemental que permita la administración de artículos y la simulación del proceso de compra mediante un carrito asociado a cada usuario registrado. Este trabajo parte del avance logrado en la Tarea 2, donde ya se disponía de un servicio REST para operaciones CRUD de usuarios y manejo de fotografías. Sobre esa base, se añadieron nuevas tablas (stock, fotos_articulos y carrito_compra) y se ampliaron los endpoints del servicio para cubrir las acciones de alta de artículos, búsqueda, compra, modificación y eliminación de elementos del carrito.

El contexto tecnológico se centra en:

1. Apache Tomcat como contenedor de servlets y punto de despliegue del servicio REST con Jersey + Jackson.
2. MySQL como gestor de persistencia, reforzado mediante el uso de transacciones para mantener la coherencia entre las cantidades del stock y las operaciones del carrito.
3. HTML + JavaScript (sin framework obligatorio) para el front-end responsive orientado a pruebas en dispositivo móvil.
4. Infraestructura en Azure for Students donde se aprovisiona la VM (nombre: T7-2022630278) en región de Canadá, siguiendo los lineamientos del curso.

Durante la implementación se empleó la inteligencia artificial de GitHub Copilot como herramienta de asistencia en la generación de código repetitivo, puntos de refactor y sugerencias de estructura, manteniéndose la revisión manual para garantizar:

- Correcta adecuación de cada endpoint a las reglas de negocio.
- Validez de las transacciones y manejo apropiado de errores HTTP (códigos 200 y 400).
- Coherencia entre las capturas del front-end y las pruebas unitarias mediante curl.

La arquitectura favorece una separación clara entre responsabilidades del cliente y del servidor: el front-end gestiona captura, interacción y presentación, mientras el back-end expone operaciones REST atómicas o transaccionales. El token de sesión (generado en el login) actúa como llave para la autorización posterior. La protección de contraseñas se realiza calculando el hash SHA-256 en el cliente antes del envío, reduciendo el riesgo de exposición de texto plano, aunque se reconocen mejoras futuras (salting, almacenamiento más robusto).

3 Objetivos

3.1 Objetivo general

Implementar un prototipo funcional de comercio electrónico con gestión de usuarios, artículos y carrito de compra, empleando un servicio REST sobre Tomcat y una interfaz HTML/JavaScript accesible desde móvil.

3.2 Objetivos específicos

- Integrar nuevas tablas (stock, fotos_articulos, carrito_compra) con llaves y restricciones adecuadas.
- Desarrollar endpoints REST que cubran alta, consulta y operaciones sobre el carrito, respetando reglas de negocio y validaciones.
- Asegurar integridad mediante transacciones en operaciones críticas (compra, modificación, eliminación).
- Adaptar el front-end para incluir pantallas de captura de artículo, compra y carrito.
- Probar cada método con curl (back-end) y cada requerimiento en dispositivo móvil (front-end).
- Incorporar autenticación basada en token y hash de contraseñas previo al envío.
- Documentar de forma clara los requerimientos funcionales y no funcionales.

4 Requerimientos

Esta sección detalla los requerimientos de la solución, organizados en tres grupos: funcionales del back-end, funcionales del front-end y no funcionales. Se parte del enunciado oficial, adaptando el lenguaje para claridad en el reporte.

4.1 Requerimientos funcionales del back-end

El servicio REST expone métodos que operan sobre usuarios (heredados de la Tarea 2) y sobre las nuevas entidades del comercio electrónico. Las operaciones nuevas necesitan validar el acceso mediante el token emitido en el login. Además, aplican reglas de negocio sobre inventarios y cantidades del carrito.

Principales requerimientos (nuevos para la Tarea 7):

- **alta_articulo:** Inserta un artículo (nombre, descripción, precio, cantidad y fotografía) en la tabla stock y opcionalmente la foto en fotos_articulos.
- **consulta_articulos:** Recupera artículos cuyo nombre o descripción contengan una palabra clave (búsqueda con LIKE). Regresa id_articulo, foto, nombre, descripción y precio.
- **compra_articulo:** Agrega un artículo al carrito del usuario si hay existencias suficientes; descuenta la cantidad del stock en una transacción (INSERT/UPDATE + UPDATE).
 - Si la cantidad solicitada supera existencias se retorna error (400) con mensaje “No hay suficientes artículos en stock”.
- **elimina_articulo_carrito_compra:** Remueve un artículo del carrito y regresa las unidades al stock en una transacción.
- **elimina_carrito_compra:** Vacía por completo el carrito del usuario, regresando cada cantidad al stock dentro de una transacción.
- **modifica_carrito_compra:** Ajusta (+1 / -1) la cantidad comprada de un artículo en el carrito, sincronizando stock y validando límites (no decrementar por debajo de 1; no incrementar sin existencias).

- `consulta_carrito_compra`: Devuelve los artículos en el carrito con foto, nombre, cantidad, precio y costo, además del total acumulado.

Cada método que modifica tablas relacionadas (`stock` y `carrito_compra`) utiliza transacciones para evitar inconsistencias. Los errores de acceso (token inválido o inexistente) retornan código 400 con objeto de error.

Lista complementaria (heredadas de la Tarea 2 y aún relevantes):

- `login`
- `alta_usuario`
- `consulta_usuario`
- `modifica_usuario`
- `borra_usuario`

4.2 Requerimientos funcionales del front-end

La interfaz HTML/JavaScript debe permitir un flujo claro y accesible desde un dispositivo móvil (layout simple, controles táctiles). Se agregan dos opciones al menú principal: “Captura de artículo” y “Compra de artículos”, además de la pantalla asociada al carrito.

Elementos clave:

- Pantalla “Captura de artículo”: Formulario para nombre, descripción, precio, cantidad y foto. Invoca `alta_articulo`.
- Pantalla “Compra de artículos”: Campo para palabra clave; muestra resultados con miniatura, nombre, descripción, precio y controles (cantidad, botones +/-, botón “Compra”).
- Botón “Carrito de compra”: Muestra pantalla “Artículos en el carrito” con cada artículo, su cantidad, precio individual, costo (cantidad × precio) y total general.
- Controles para modificar cantidad (+/-) sobre artículos ya en el carrito (`modifica_carrito_compra`).

- Botón “Eliminar artículo” para retirar un artículo del carrito (elimina_articulo_carrito_compra).
- Botón “Eliminar carrito” para vaciarlo por completo (elimina_carrito_compra).
- Botón “Seguir comprando” para volver a la pantalla de búsqueda.
- Manejo de autenticación: login previo y token persistente en sesión.
- Hash SHA-256 de contraseñas antes de enviar (ya existente de la Tarea 2).

4.3 Requerimientos no funcionales

Estos aseguran condiciones estructurales y de calidad:

Parámetros de infraestructura y despliegue:

- Máquina virtual Azure creada desde la imagen de la Tarea 2.
- Nombre estricto de la VM: T7-2022630278 (relacionado con la boleta).
- Región Canadá (Azure for Students).
- Exposición del puerto 8080 para acceso al front-end y pruebas con curl.

Base de datos y consistencia:

- Uso de transacciones para operaciones que combinan cambios en stock y carrito (compra, modificación de cantidad, eliminación de artículos, vaciado completo).
- Integridad referencial mediante llaves foráneas: fotos_articulos → stock, carrito_compra → usuarios y stock.
- Índice único (PRIMARY KEY compuesta) en carrito_compra (id_usuario, id_articulo) que evita duplicados del mismo artículo dentro del mismo carrito.

Calidad y mantenibilidad:

- Código Java organizado en POJOs (Usuario, Articulo, CarritoItem, HuboError).
- Serialización JSON consistente vía Jackson.
- Separación clara de responsabilidades (front-end ligero, back-end transaccional).

- Manejo de errores con códigos HTTP diferenciados (200 éxito / 400 error de lógica o acceso).

Seguridad básica:

- Token de sesión emitido en login.
- Hash de contraseñas en el cliente antes de envío (SHA-256).
- Validaciones de campos obligatorios y rangos (precio > 0, cantidad >= 0).

Pruebas y evidencia:

- Capturas de pantalla en dispositivo móvil para cada requerimiento funcional front-end.
- Salidas de curl por cada endpoint del back-end.

Aspectos potenciales de mejora (identificados, no obligatorios en esta fase):

- Paginación de resultados de consulta_articulos.
- Salt y almacenamiento de contraseñas con algoritmo robusto en servidor.
- Rol de administrador para separación de alta de artículos vs. compra.
- Cache o CDN para imágenes de productos.

5 Arquitectura del Sistema

La arquitectura del prototipo se compone de cuatro capas principales que interactúan de forma secuencial y transaccional: (1) Cliente Web móvil (HTML/JavaScript), (2) Servicio REST en Tomcat (Jersey + Jackson), (3) Base de Datos MySQL y (4) Infraestructura de despliegue en Azure (VM nombrada T7-2022630278). El flujo general parte de la autenticación (emisión de token), continúa con operaciones de artículo y carrito, y finaliza en la persistencia consistente de los datos.

La solución mantiene un enfoque minimalista de micro-arquitectura: cada endpoint encapsula una operación concreta y, cuando es crítico, agrupa múltiples sentencias SQL dentro de una transacción explícita (compra, modificación del carrito, eliminación de artículos, vaciado completo). El front-end es deliberadamente ligero para evidenciar la interacción y facilitar pruebas móviles.

5.1 Visión global por capas

Capa	Tecnología	Responsabilidad principal	Interacción destacada
Presentación	HTML + JavaScript (WSClient.js)	Captura de datos (usuarios, artículos, búsquedas, carrito), renderización y eventos táctiles	Emite peticiones HTTP (GET/POST/PUT/DELETE) al servicio REST
Lógica de negocio / API	Java (Servlet Container: Tomcat + Jersey)	Validación de parámetros, verificación de acceso (token), reglas de stock y carrito, control transaccional	Mapea JSON ↔ POJOs y coordina transacciones hacia la BD
Persistencia	MySQL	Almacenamiento de usuarios, artículos, fotos y carrito; integridad referencial	Recibe consultas y actualizaciones dentro o fuera de transacciones
Infraestructura	VM Azure (Linux/Windows según imagen base)	Hosting de Tomcat y MySQL, exposición de puerto 8080, seguridad básica (NSG)	Provee disponibilidad y acceso remoto para front-end y pruebas curl

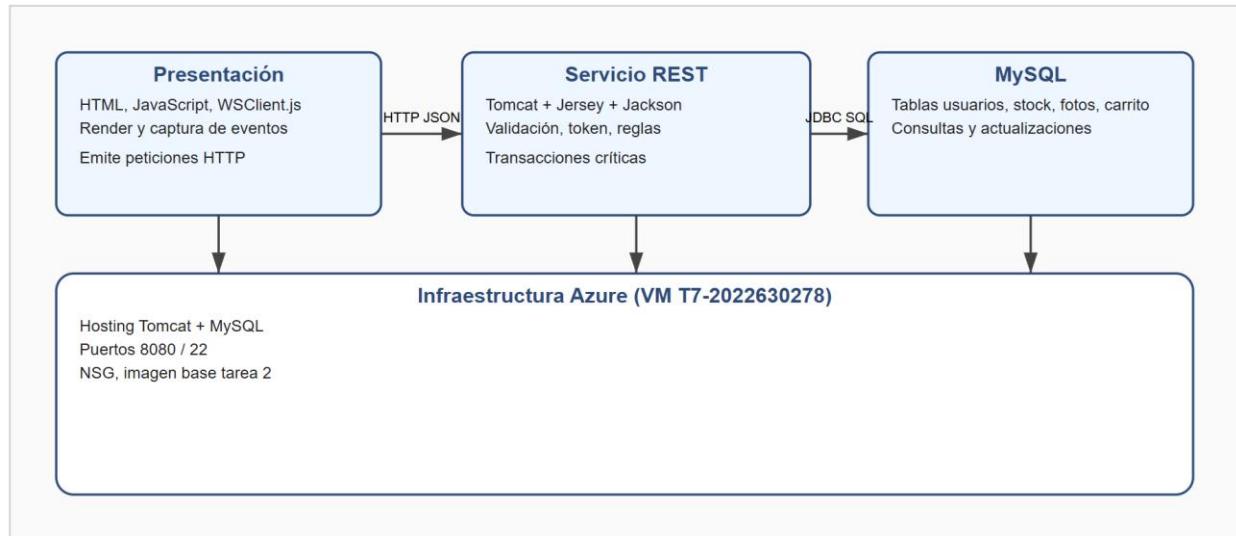


Imagen 6: Diagrama de arquitectura por capas (Móvil / Tomcat REST / MySQL / VM).

5.2 Componentes principales

- Front-end (prueba.html): Gestiona pantallas de Login, Perfil, Captura de artículo, Compra de artículos y Carrito. Invoca WSClient.js para emitir peticiones uniformes.
- WSClient.js: Wrapper simplificado sobre fetch que codifica parámetros y maneja JSON para métodos GET, POST, PUT y DELETE.
- Servicio.java: Clase central con endpoints REST; inicializa DataSource vía JNDI (context.xml). Orquesta verificación de token, queries y transacciones.
- Clases POJO (Usuario, Articulo, CarritoItem, HuboError): Definen estructura de datos, favoreciendo serialización/deserialización con Jackson.
- context.xml: Declara recurso de datasource para conexión a MySQL.
- web.xml: Configura servlet de Jersey, paquete “servicio” y soporte para JacksonFeature.

- Base de Datos: Tablas usuarios, fotos_usuarios, stock, fotos_articulos, carrito_compra (relaciones y llaves primarias/foráneas).

5.3 Flujo de autenticación y autorización

1. El usuario ingresa email y contraseña (plaintext sólo en el navegador).
2. Se calcula hash SHA-256 en el cliente antes del envío.
3. El endpoint login valida la combinación email + password en la BD.
4. Se genera un token aleatorio (20 caracteres) y se almacena en usuarios.token.
5. El front-end conserva token e id_usuario en memoria local (variables JS).
6. Cada operación protegida (artículos / carrito) envía id_usuario y token para verificación.

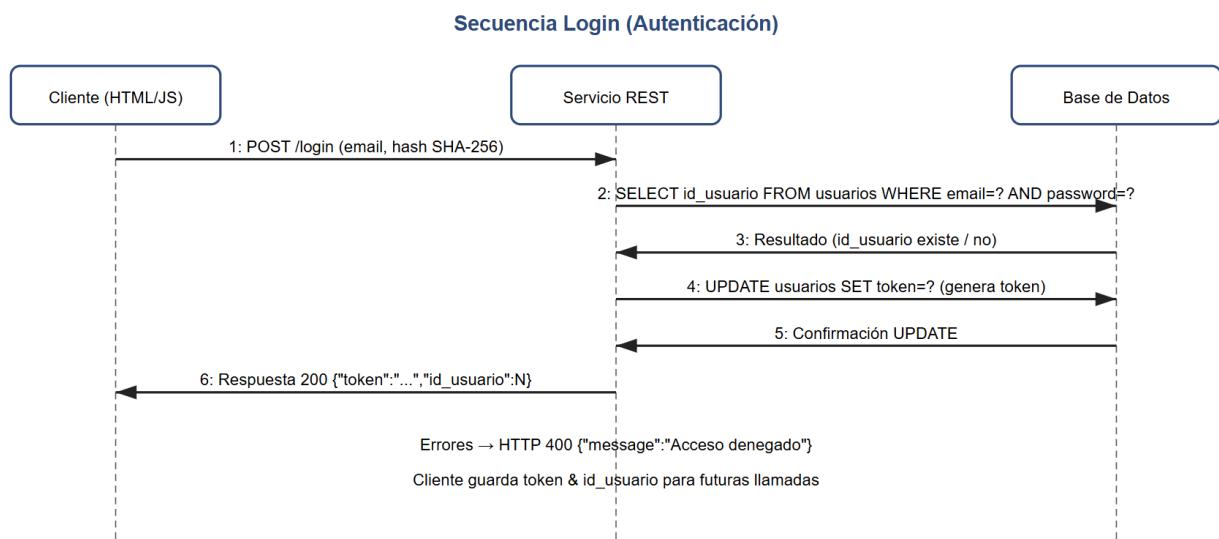


Imagen 7: Diagrama de secuencia del login (Cliente → Servicio → BD → Servicio → Cliente).

5.4 Flujo de compra y consistencia transaccional

La operación de compra (compra_articulo) ejemplifica el patrón transaccional:

1. Verificación de token e id_usuario.
2. Bloqueo (SELECT ... FOR UPDATE) o lectura directa del stock para confirmar existencias suficientes.
3. Inserción o actualización del carrito (carrito_compra).
4. Descuento de la cantidad en stock.
5. Commit; si ocurre error, rollback y respuesta HTTP 400 con mensaje detallado.

Igual mecanismo aplica en:

- modifica_carrito_compra: Ajuste (+/-) cantidad con control de límites y actualización inversa del stock.
- elimina_articulo_carrito_compra: Recupera unidades al stock y borra el registro del carrito.
- elimina_carrito_compra: Itera sobre todos los registros y devuelve las cantidades correspondientes al stock.

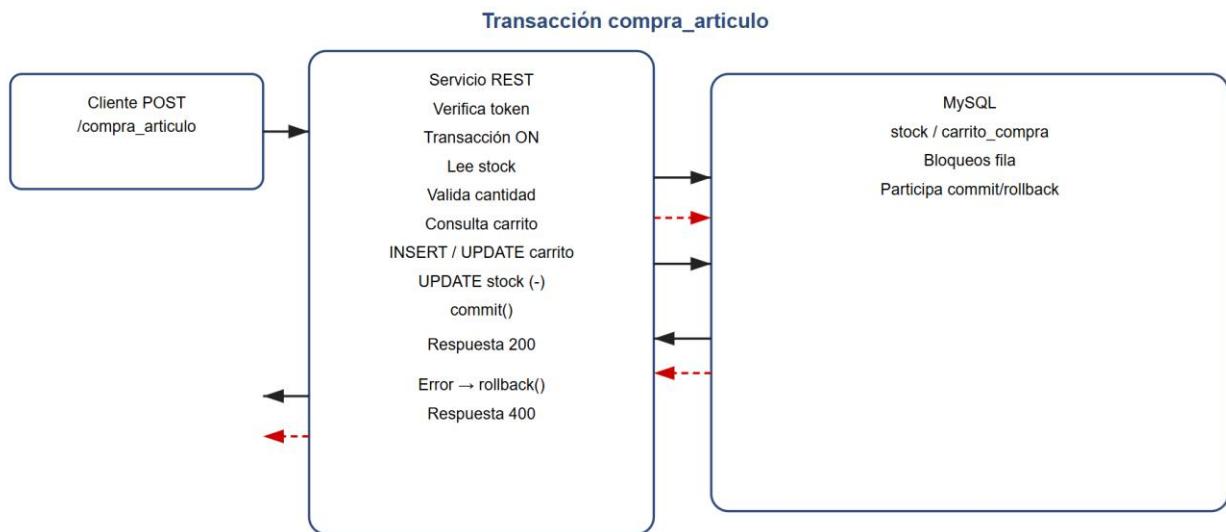
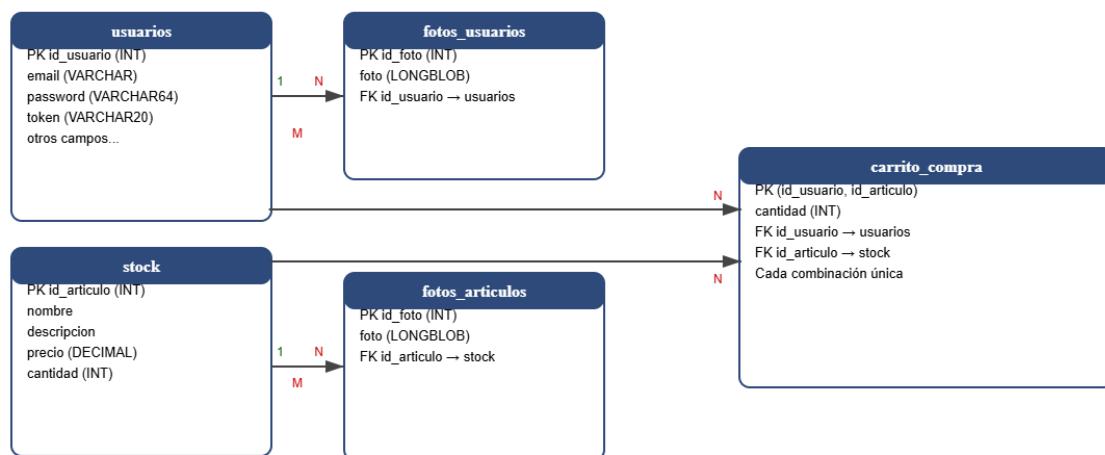


Imagen 8: Diagrama de transacción de compra (Stock ↔ Carrito ↔ Commit/Rollback).

5.5 Modelo lógico y relaciones (resumen conceptual)

Tabla	Llave primaria	Relaciones	Descripción
usuarios	id_usuario	fotos_usuarios (1:N), carrito_compra (1:N)	Información y credenciales (hash + token)
fotos_usuarios	id_foto	FK → usuarios.id_usuario	Foto opcional de perfil
stock	id_articulo	fotos_articulos (1:N), carrito_compra (N:M vía usuarios)	Inventario y datos de cada artículo
fotos_articulos	id_foto	FK → stock.id_articulo	Fotografía del artículo
carrito_compra	(id_usuario, id_articulo)	FK → usuarios, FK → stock	Asociación de artículos a usuario con cantidad

Entidad - Relación Simplificado



Relaciones: usuarios-fotos (1:N), stock-fotos_articulos (1:N), usuarios-stock (M:N) vía carrito_compra.

Imagen 9: Diagrama entidad-relación simplificado (Usuarios, Stock, Carrito, Fotos).

6 Desarrollo

7 Infraestructura en Azure

Esta sección documenta la preparación y validación de la infraestructura necesaria para ejecutar el prototipo: la máquina virtual (VM) creada a partir de la imagen de la Tarea 2, la configuración de puertos, los servicios (Tomcat y MySQL) y las evidencias solicitadas por el profesor. Todas las capturas deben ser pantallas completas (sin recortes) mostrando fecha y hora visibles, conforme a los lineamientos del curso.

La VM se nombra estrictamente según la boleta: T7-2022630278. Se selecciona una región en Canadá dentro del programa Azure for Students (Canada Central o Canada East) para cumplir el requerimiento no funcional. La reutilización de la imagen previa acelera la disponibilidad de Tomcat y de la base de datos, reduciendo pasos de instalación.

7.1 Creación de la máquina virtual

El proceso parte de la imagen generada en la Tarea 2 (que ya incluye entorno Java, Tomcat y MySQL configurado). En el portal de Azure se elige “Create a resource” → “Virtual machine” y se llenan los parámetros mínimos: nombre, región, tamaño, autenticación y red. Se verifica que el nombre coincide exactamente con el formato requerido para evitar rechazo de la tarea.

Pasos esenciales (los más relevantes para evidencia):

1. Seleccionar la suscripción (Students) y el grupo de recursos (reutilizar o crear uno específico p.e. rg-t7-ecommerce).
2. Definir nombre de la VM: T7-2022630278.
3. Elegir región: Canada Central (o Canada East si la primera no disponible).
4. Seleccionar tamaño (ej. B1s o B2s según costo / desempeño).
5. Método de autenticación: clave SSH (recomendado) o contraseña segura.
6. Disco: estándar SSD (suficiente para el prototipo).

7. Activar puerto HTTP (8080) mediante reglas de NSG personalizadas (no hay regla predeterminada para 8080).
8. Revisar pestañas “Networking” y “Management” antes de crear.
9. Confirmar en la pantalla final de validación y presionar “Create”.

Basics

Subscription	Azure for Students
Resource group	(new) T7-2022630278-rg
Virtual machine name	T7-2022630278
Region	Canada Central
Availability options	Availability zone
Zone options	Self-selected zone
Availability zone	1
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	No
Image	1/T2-2022630278-imagen/latest - Gen2

< Previous Next > Create Download a template for automation Give feedback

High UV Now

23°C Sunny

ENG ES 01:07 p.m. 22/11/2025

Create a virtual machine

Validation passed

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload | Help me create a low cost VM

Disk

OS disk size	Image default
OS disk type	Standard SSD LRS
Use managed disks	Yes
Delete OS disk with VM	Disabled
Ephemeral OS disk	No

Networking

Virtual network	vnet-canadacentral
Subnet	snet-canadacentral-1
Public IP	(new) T7-2022630278-ip
NIC network security group	(new) T7-2022630278-nsg
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled

< Previous | Next > | Create | Download a template for automation | Give feedback

23°C Sunny | Search | Copilot | INSTITUTO POLITÉCNICO NACIONAL | 01:07 p.m. 22/11/2025

Create a virtual machine

Validation passed

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload | Help me create a low cost VM

Management

Microsoft Defender for Cloud	None
System assigned managed identity	Off
Login with Microsoft Entra ID	Off
Auto-shutdown	Off
Enable periodic assessment	Off
Enable hotpatch	Off
Patch orchestration options	Image Default

Monitoring

Alerts	Off
Boot diagnostics	On
Enable OS guest diagnostics	Off
Enable application health monitoring	Off

Advanced

Extensions	None
------------	------

< Previous | Next > | Create | Download a template for automation | Give feedback

23°C Sunny | Search | Copilot | INSTITUTO POLITÉCNICO NACIONAL | 01:08 p.m. 22/11/2025

Imagen 11: Pantalla de resumen/Review + Create antes de lanzar la VM con validación “Passed”.

7.2 Características y configuración técnica

Una vez creada, se documentan las especificaciones principales para transparentar el entorno de ejecución del servicio:

Parámetro	Valor Propuesto / Observado	Justificación
Nombre	T7-2022630278	Cumple formato boleta requerido
Región	Canada Central	Requerimiento no funcional
Tamaño	B1s (1 vCPU, 1 GiB) o B2s	Suficiente para Tomcat + MySQL en pruebas
SO Base	Linux (Ubuntu Server) o Windows Server (según imagen T2)	Compatibilidad con stack Java/Tomcat
Disco OS	Standard SSD 30 GiB	Lectura/escritura adecuada para pruebas
Puertos abiertos	22 (SSH), 8080 (HTTP)	Acceso administrativo y aplicación web
Seguridad	NSG con reglas específicas, bloqueo de puertos no usados	Minimiza superficie de ataque
Servicios clave	Tomcat en :8080, MySQL local, Java JDK	Requisitos del prototipo
FQDN/Pública	IP pública asignada por Azure	Necesaria para acceso móvil y pruebas curl

T7-2022630278 - Microsoft Azure

Home > CreateVm-T2-2022630278-imagen-20251122130614 | Overview >

T7-2022630278 Virtual machine

Help me copy this VM in any region | Manage this VM with Azure CLI

Search

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Resource visualizer
- Connect
- Connect
- Bastion
- Networking
- Network settings
- Load balancing
- Application security groups
- Network manager
- Settings
- Availability + scale

Tags (edit) : Add tags

Properties Monitoring Capabilities (7) Recommendations Tutorials

Virtual machine

Computer name	T2-2022630278
Operating system	Linux (ubuntu 24.04)
VM generation	V2
VM architecture	x64

Networking

Public IP address	4.248.144.161 (Network interface t7-2022630278110_z1)
1 associated public IPs	-
Public IP address (IPv6)	-
Private IP address	172.16.0.4

Operating system : Linux (ubuntu 24.04)
Size : Standard B1s (1 vcpu, 1 GiB memory)
Primary NIC public IP : 4.248.144.161
1 associated public IPs
Virtual network/subnet : vnet-canadacentral/snet-canadacentral-1
DNS name : Not configured
Health state : -
Time created : 11/22/2025, 7:08 PM UTC

JSON View

High UV Now

Search

01:40 p.m. 22/11/2025

Imagen 12: Detalle de la VM en el portal (Overview con Status: Running, IP pública y región).

T7-2022630278-nsg | Inbound security rules

Network security group

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Resource visualizer

Settings

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Monitoring

Automation

Help

Add Hide default rules Refresh Delete Give feedback

Filter by name

Priority ↑.	Name ↑	Port ↑	Protocol ↑	Source ↑	Destination ↑	Action ↑
1000	default-allow-ssh	22	TCP	Any	Any	Allow
1010	Allow-Tomcat-8080	8080	TCP	Any	Any	Allow
1020	Allow-MySQL-3306-Hom...	3306	TCP	4.248.144.161	Any	Allow
65000	AllowVNetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerI...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

25°C Sunny

Search

ENG ES

03:48 p.m. 22/11/2025

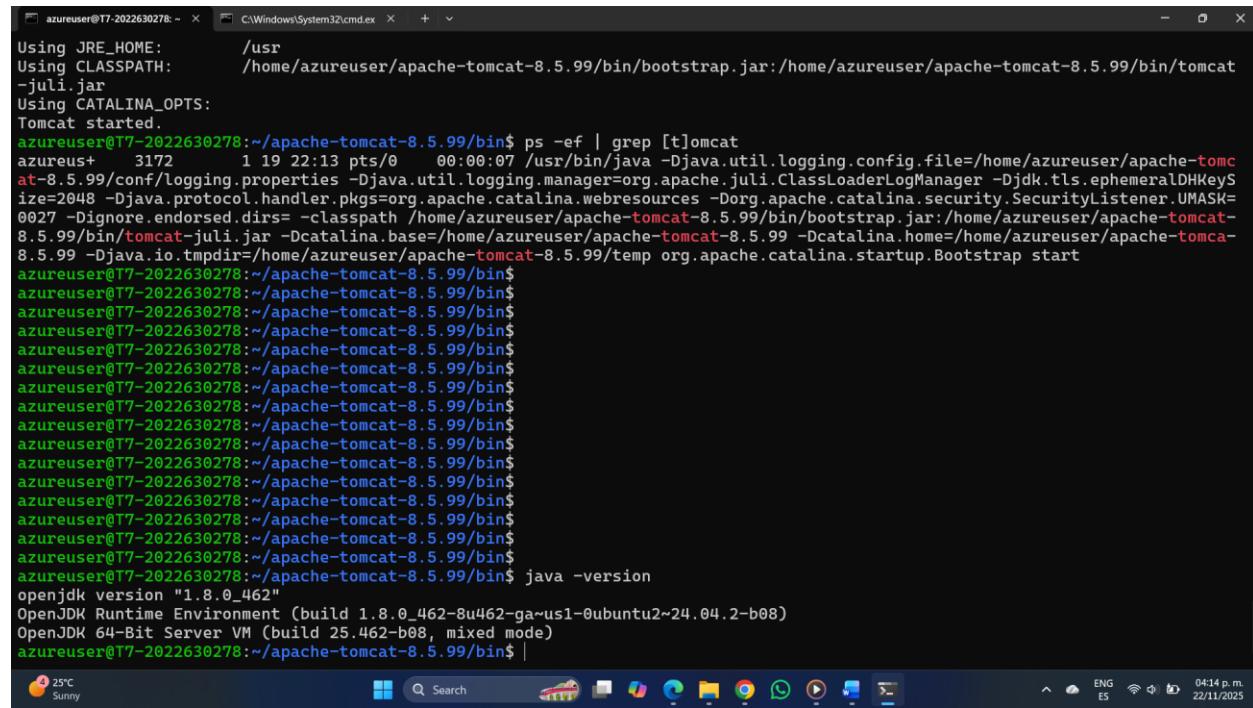
Imagen 13: Configuración de NSG mostrando reglas para puertos 22 y 8080 (toda la pantalla).

8 Configuración del Entorno

Esta sección documenta de forma detallada cada paso seguido para preparar el entorno de ejecución del prototipo: instalación/verificación de Java y Tomcat, configuración del datasource (MySQL), incorporación de librerías de Jersey/Jackson, compilación y despliegue del servicio REST, copia de archivos del front-end y validaciones finales. Todos los pasos incluyen comandos y archivos clave para garantizar reproducibilidad. Las capturas asociadas deben ser pantallas completas con fecha y hora (lineamientos del curso).

8.1 Verificación / Instalación de Java

Antes de compilar el servicio se comprueba la versión de Java disponible en la VM (imagen heredada de Tarea 2).



A screenshot of a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the output of the command 'java -version'. The output shows the Java version as '1.8.0_462' and the OpenJDK Runtime Environment as 'build 1.8.0_462-8u462-ga~us1-0ubuntu2~24.04.2-b08'. The desktop icons at the bottom include File Explorer, Edge, File Manager, Task View, and others.

```
Using JRE_HOME:      /usr
Using CLASSPATH:    /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
azureuser@T7-2022630278:~/apache-tomcat-8.5.99/bin$ ps -ef | grep [t]omcat
azureus+  3172      1 19 22:13 pts/0    00:00:07 /usr/bin/java -Djava.util.logging.config.file=/home/azureuser/apache-tomcat-8.5.99/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar -Dcatalina.base=/home/azureuser/apache-tomcat-8.5.99 -Dcatalina.home=/home/azureuser/apache-tomcat-8.5.99 -Djava.io.tmpdir=/home/azureuser/apache-tomcat-8.5.99/temp org.apache.catalina.startup.Bootstrap start
azureuser@T7-2022630278:~/apache-tomcat-8.5.99/bin$
azureuser@T7-2022630278:~/apache-tomcat-8.5.99/bin$ java -version
openjdk version "1.8.0_462"
OpenJDK Runtime Environment (build 1.8.0_462-8u462-ga~us1-0ubuntu2~24.04.2-b08)
OpenJDK 64-Bit Server VM (build 25.462-b08, mixed mode)
azureuser@T7-2022630278:~/apache-tomcat-8.5.99/bin$ |
```

Imagen 16: Consola mostrando java -version con hora visible.

8.2 Verificación / Instalación de Tomcat

La imagen de la Tarea 2 ya incluye Tomcat. Se valida que el servicio esté activo.

Imagen 17: comando ps -ef | grep tomcat de Tomcat activo.

8.3 Base de datos (MySQL)

A continuación se explica con detalle cómo realizar la sección de la Base de Datos

8.3.1 Alterar tabla usuarios

Se agregaron los campos password y token:

```
azureuser@T7-2022630278:~ $ sudo mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| servicio_web |
| sys |
+-----+
5 rows in set (0.05 sec)

mysql> USE servicio_web;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> DESCRIBE usuarios;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id_usuario | int | NO | PRI | NULL | auto_increment |
+-----+-----+-----+-----+-----+-----+
```

```
azureuser@T7-2022630278:~ $ sudo mysql -u root
[sudo] password for azureuser:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.43-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> ALTER TABLE usuarios
    -> ADD COLUMN password VARCHAR(64) NOT NULL AFTER email,
    -> ADD COLUMN token VARCHAR(20) NULL AFTER password;
Query OK, 0 rows affected (0.43 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE usuarios
    -> ADD COLUMN password VARCHAR(64) NULL AFTER email,
    -> ADD COLUMN token VARCHAR(20) NULL AFTER password;
ERROR 1060 (42S21): Duplicate column name 'password'
mysql> exit
Bye
azureuser@T7-2022630278:$
```

```
azureuser@T7-2022630278: ~ x Simbolo del sistema x + v
| information_schema |
| mysql |
| performance_schema |
| servicio_web |
| sys |
+-----+
5 rows in set (0.06 sec)

mysql> USE servicio_web;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> DESCRIBE usuarios;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_usuario | int | NO | PRI | NULL | auto_increment |
| email | varchar(100) | NO | UNI | NULL |
| password | varchar(64) | NO | | NULL |
| token | varchar(20) | YES | | NULL |
| nombre | varchar(100) | NO | | NULL |
| apellido_paterno | varchar(100) | NO | | NULL |
| apellido_materno | varchar(100) | YES | | NULL |
| fecha_nacimiento | datetime | NO | | NULL |
| telefono | bigint | YES | | NULL |
| genero | char(1) | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> |
22°C Clear Search 06:32 p. m.
ENG ES 22/11/2025
```

2.2 Crear tablas del e-commerce

```
azureuser@T7-2022630278: ~ └─ Simbolo del sistema └─ + └ x
mysql> --
mysql> -- Tabla de articulos (stock)
mysql> CREATE TABLE IF NOT EXISTS stock (
    ->     id_articulo INT AUTO_INCREMENT PRIMARY KEY,
    ->     nombre VARCHAR(100) NOT NULL,
    ->     descripcion TEXT NOT NULL,
    ->     precio DECIMAL(10,2) NOT NULL,
    ->     cantidad INT NOT NULL
    -> ) ENGINE=InnoDB;
fotos_articulos (
    id_foto INT AUTO_INCREMENT PRIMARY KEY,
    foto LONGBLOB NOT NULL,
    id_articulo INT NOT NULL,
    CONSTRAINT fk_fotos_articulos_stock
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
        ON DELETE CASCADE
) ENGINE=InnoDB;

-- Tabla del carrito de compra
CREATE TABLE IF NOT EXISTS carrito_compra (
    id_usuario INT NOT NULL,
    id_articulo INT NOT NULL,
    cantidad INT NOT NULL,
    PRIMARY KEY (id_usuario, id_articulo),
    CONSTRAINT fk_carrito_usuario
        FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
        ON DELETE CASCADE,
    CONSTRAINT fk_carrito_articulo
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
        ON DELETE CASCADE
) ENGINE=InnoDB;Query OK, 0 rows affected (0.21 sec)

22°C Clear └─ Search └─ + └ x └─ ENG ES └─ 06:33 p.m. └─ 22/11/2025
```

```
azureuser@T7-2022630278: ~ └─ Simbolo del sistema └─ + └ x
-- Tabla del carrito de compra
CREATE TABLE IF NOT EXISTS carrito_compra (
    id_usuario INT NOT NULL,
    id_articulo INT NOT NULL,
    cantidad INT NOT NULL,
    PRIMARY KEY (id_usuario, id_articulo),
    CONSTRAINT fk_carrito_usuario
        FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
        ON DELETE CASCADE,
    CONSTRAINT fk_carrito_articulo
        FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
        ON DELETE CASCADE
) ENGINE=InnoDB;Query OK, 0 rows affected (0.21 sec)

mysql>
mysql> -- Tabla de fotos de articulos
mysql> CREATE TABLE IF NOT EXISTS fotos_articulos (
    ->     id_foto INT AUTO_INCREMENT PRIMARY KEY,
    ->     foto LONGBLOB NOT NULL,
    ->     id_articulo INT NOT NULL,
    ->     CONSTRAINT fk_fotos_articulos_stock
    ->         FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
    ->         ON DELETE CASCADE
    -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.18 sec)

mysql>
mysql> -- Tabla del carrito de compra
mysql> CREATE TABLE IF NOT EXISTS carrito_compra (
    ->     id_usuario INT NOT NULL,
```



```
22°C Clear └─ Search └─ + └ x └─ ENG ES └─ 06:33 p.m. └─ 22/11/2025
```

```
| azureuser@T7-2022630278: ~ x Símbolo del sistema x + v
) ENGINE=InnoDB;Query OK, 0 rows affected (0.21 sec)

mysql>
mysql> -- Tabla de fotos de artículos
mysql> CREATE TABLE IF NOT EXISTS fotos_articulos (
->   id_foto INT AUTO_INCREMENT PRIMARY KEY,
->   foto LONGBLOB NOT NULL,
->   id_articulo INT NOT NULL,
->   CONSTRAINT fk_fotos_articulos_stock
->     FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
->     ON DELETE CASCADE
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.18 sec)

mysql>
mysql> -- Tabla del carrito de compra
mysql> CREATE TABLE IF NOT EXISTS carrito_compra (
->   id_usuario INT NOT NULL,
->   id_articulo INT NOT NULL,
->   cantidad INT NOT NULL,
->   PRIMARY KEY (id_usuario, id_articulo),
->   CONSTRAINT fk_carrito_usuario
->     FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)
->     ON DELETE CASCADE,
->   CONSTRAINT fk_carrito_articulo
->     FOREIGN KEY (id_articulo) REFERENCES stock(id_articulo)
->     ON DELETE CASCADE
-> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.07 sec)

mysql> |
```



8.4 Incorporación de librerías (Jersey/Jackson)

Se descargan los JAR indicados (versiones exactas):

- jersey-media-json-jackson-2.24.jar
- jersey-entity-filtering-2.24.jar
- jackson-core-2.6.3.jar
- jackson-databind-2.6.3.jar
- jackson-annotations-2.6.3.jar
- jackson-jaxrs-json-provider-2.6.3.jar
- jackson-jaxrs-base-2.6.3.jar
- jackson-module-jaxb-annotations-2.6.3.jar

A continuación, se presentan los pasos del procedimiento completo para realizar la descarga directa de las librerías Jackson/Jersey dentro de la VM, usando la ruta de Tomcat:

CATALINA_HOME = /home/azureuser/apache-tomcat-8.5.99

1. Ubicarse en el directorio de librerías de Tomcat
2. Descargar directamente las 8 librerías solicitadas
3. Verificar que llegaron

Debe listar los 8 nombres.

```
azur&lt;user@T7-2022630278: ~ x C:\Windows\System32\cmd.exe x + v
Servicio WSClient.js iniciar.sh prueba_url.html tarea2
Servicio.zip apache-tomcat-8.5.99 prueba_json.html setup_tomcat8.sh usuario_sin_foto.png
azur&lt;user@T7-2022630278: ~$ echo $CATALINA_HOME
/home/azureuser/apache-tomcat-8.5.99
azur&lt;user@T7-2022630278: ~$ echo $CATALINA_HOME
/home/azureuser/apache-tomcat-8.5.99
azur&lt;user@T7-2022630278: ~$ cd /home/azureuser/apache-tomcat-8.5.99/lib
azur&lt;user@T7-2022630278: ~/apache-tomcat-8.5.99/lib$ cd /home/azureuser/apache-tomcat-8.5.99/lib

wget -q https://repo1.maven.org/maven2/org/glassfish/jersey/media/jersey-media-json-jackson/2.24/jersey-media-json-jackson-2.24.jar
wget -q https://repo1.maven.org/maven2/org/glassfish/jersey/ext/jersey-entity-filtering/2.24/jersey-entity-filtering-2.24.jar

wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-core/2.6.3/jackson-core-2.6.3.jar
wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-databind/2.6.3/jackson-databind-2.6.3.jar
wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/core/jackson-annotations/2.6.3/jackson-annotations-2.6.3.jar
wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-json-provider/2.6.3/jackson-jaxrs-json-provider-2.6.3.jar
wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/jaxrs/jackson-jaxrs-base/2.6.3/jackson-jaxrs-base-2.6.3.jar
wget -q https://repo1.maven.org/maven2/com/fasterxml/jackson/module/jackson-module-jaxb-annotations/2.6.3/jackson-module-jaxb-annotations-2.6.3.jar
azur&lt;user@T7-2022630278: ~/apache-tomcat-8.5.99/lib$ ls -1 | grep -E 'jersey-media-json-jackson|jersey-entity-filtering|jackson-core-2.6.3|jackson-databind-2.6.3|jackson-annotations-2.6.3|jackson-jaxrs-json-provider|jackson-jaxrs-base|jackson-module-jaxb-annotations'
jackson-annotations-2.6.3.jar
jackson-core-2.6.3.jar
jackson-databind-2.6.3.jar
jackson-jaxrs-base-2.6.3.jar
jackson-jaxrs-json-provider-2.6.3.jar
jackson-module-jaxb-annotations-2.6.3.jar
jersey-entity-filtering-2.24.jar
jersey-media-json-jackson-2.24.jar
azur&lt;user@T7-2022630278: ~/apache-tomcat-8.5.99/lib$ |
```

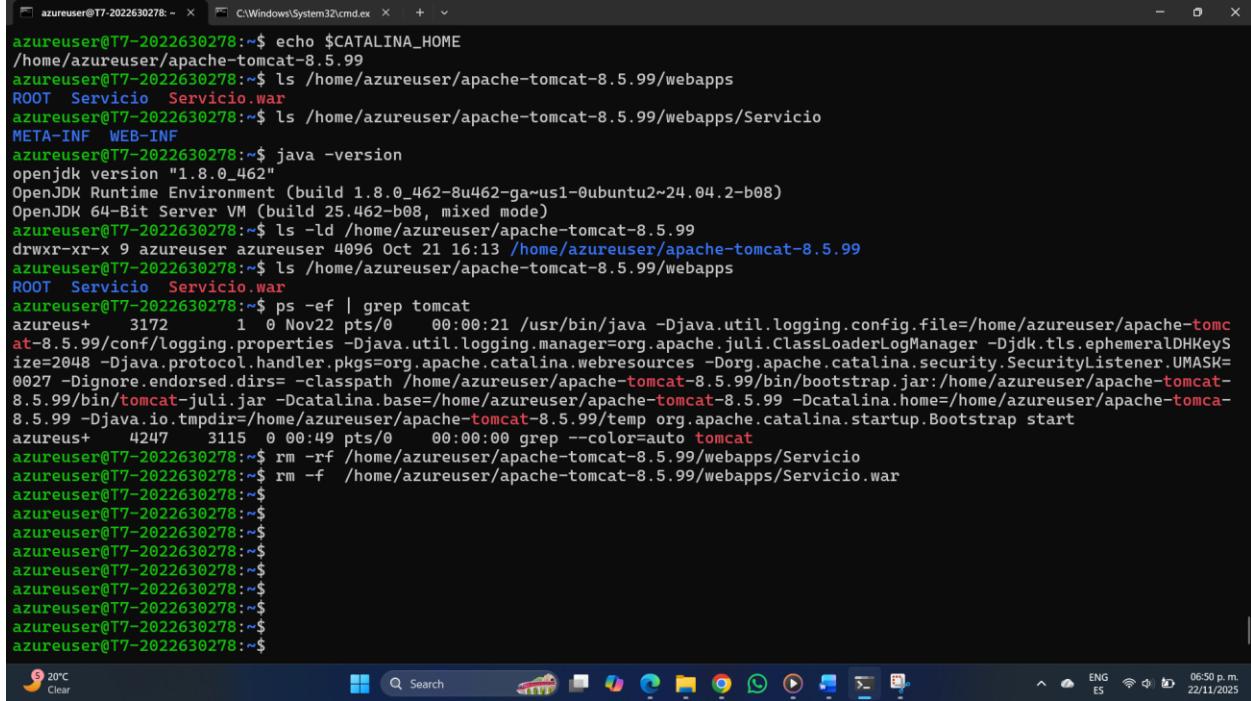
Imagen 21: Listado de archivos JAR dentro de \$CATALINA_HOME/lib.

8.5 Limpieza y despliegue base

8.5.1 Limpieza del despliegue antiguo del servicio

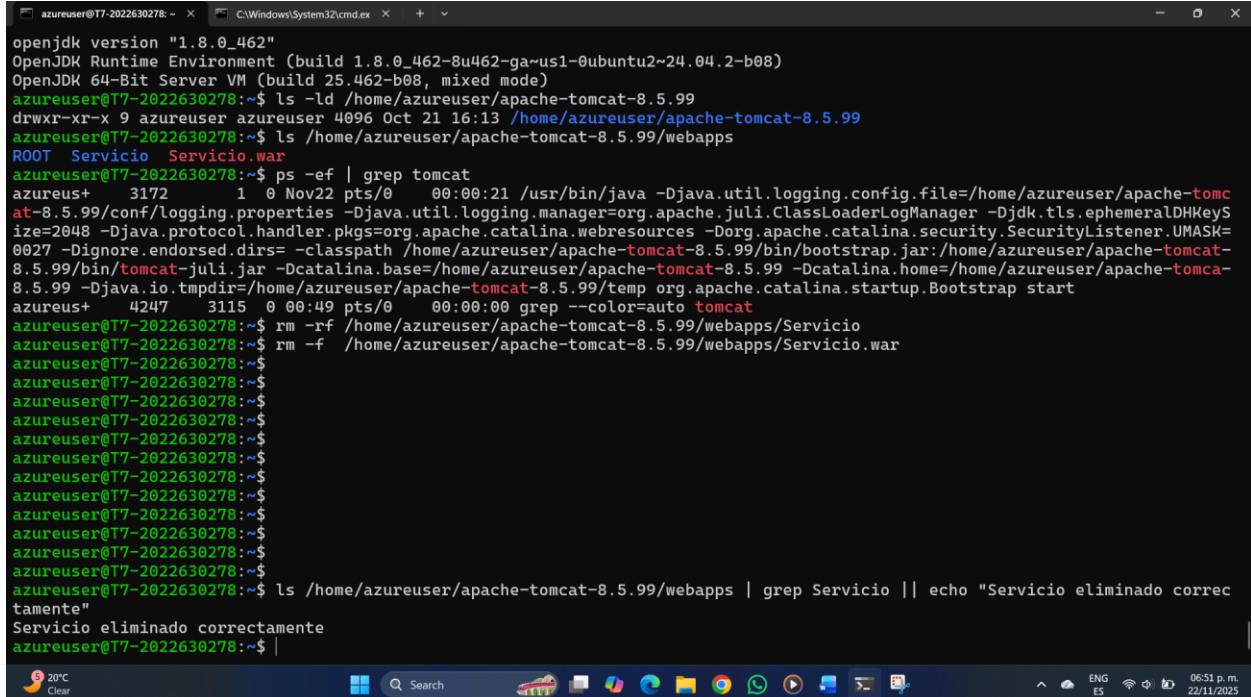
El servicio se despliega como aplicación web estándar: carpeta Servicio/ y/o archivo Servicio.war dentro de webapps.

Se eliminan los restos previos para evitar clases obsoletas:



```
azureuser@T7-2022630278:~$ echo $CATALINA_HOME
/home/azureuser/apache-tomcat-8.5.99
azureuser@T7-2022630278:~$ ls /home/azureuser/apache-tomcat-8.5.99/webapps
ROOT Servicio Servicio.war
azureuser@T7-2022630278:~$ ls /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio
META-INF WEB-INF
azureuser@T7-2022630278:~$ java -version
openjdk version "1.8.0_462"
OpenJDK Runtime Environment (build 1.8.0_462-8u462-ga~us1-0ubuntu2~24.04.2-b08)
OpenJDK 64-Bit Server VM (build 25.462-b08, mixed mode)
azureuser@T7-2022630278:~$ ls -ld /home/azureuser/apache-tomcat-8.5.99
drwxr-xr-x 9 azureuser azureuser 4096 Oct 21 16:13 /home/azureuser/apache-tomcat-8.5.99
azureuser@T7-2022630278:~$ ls /home/azureuser/apache-tomcat-8.5.99/webapps
ROOT Servicio Servicio.war
azureuser@T7-2022630278:~$ ps -ef | grep tomcat
azureus+ 3172 1 0 Nov22 pts/0 00:00:21 /usr/bin/java -Djava.util.logging.config.file=/home/azureuser/apache-tomcat-8.5.99/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar -Dcatalina.base=/home/azureuser/apache-tomcat-8.5.99 -Dcatalina.home=/home/azureuser/apache-tomcat-8.5.99 -Djava.io.tmpdir=/home/azureuser/apache-tomcat-8.5.99/temp org.apache.catalina.startup.Bootstrap start
azureus+ 4247 3115 0 00:49 pts/0 00:00:00 grep --color=auto tomcat
azureuser@T7-2022630278:~$ rm -rf /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio
azureuser@T7-2022630278:~$ rm -f /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio.war
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
```

Se revisa que ya no exista:



```
azureuser@T7-2022630278:~$ echo $CATALINA_HOME
/home/azureuser/apache-tomcat-8.5.99
OpenJDK Runtime Environment (build 1.8.0_462-8u462-ga~us1-0ubuntu2~24.04.2-b08)
OpenJDK 64-Bit Server VM (build 25.462-b08, mixed mode)
azureuser@T7-2022630278:~$ ls -ld /home/azureuser/apache-tomcat-8.5.99
drwxr-xr-x 9 azureuser azureuser 4096 Oct 21 16:13 /home/azureuser/apache-tomcat-8.5.99
azureuser@T7-2022630278:~$ ls /home/azureuser/apache-tomcat-8.5.99/webapps
ROOT Servicio Servicio.war
azureuser@T7-2022630278:~$ ps -ef | grep tomcat
azureus+ 3172 1 0 Nov22 pts/0 00:00:21 /usr/bin/java -Djava.util.logging.config.file=/home/azureuser/apache-tomcat-8.5.99/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endorsed.dirs= -classpath /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar -Dcatalina.base=/home/azureuser/apache-tomcat-8.5.99 -Dcatalina.home=/home/azureuser/apache-tomcat-8.5.99 -Djava.io.tmpdir=/home/azureuser/apache-tomcat-8.5.99/temp org.apache.catalina.startup.Bootstrap start
azureus+ 4247 3115 0 00:49 pts/0 00:00:00 grep --color=auto tomcat
azureuser@T7-2022630278:~$ rm -rf /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio
azureuser@T7-2022630278:~$ rm -f /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio.war
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$
azureuser@T7-2022630278:~$ ls /home/azureuser/apache-tomcat-8.5.99/webapps | grep Servicio || echo "Servicio eliminado correctamente"
Servicio eliminado correctamente
azureuser@T7-2022630278:~$ |
```

8.5.2 Preparar directorio de trabajo para compilar

Crea un directorio de trabajo fuera de webapps (para no mezclar fuentes con despliegue). Por ejemplo:

```
mkdir -p ~/tarea7_backend
```

```
cd ~/tarea7_backend
```

Se copia o sube por SFTP el archivo Servicio.zip aquí. No se pone dentro de webapps, porque se compila primero y sólo el WAR final se copia a webapps.

```

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>sftp azureuser@4.248.144.161
azureuser@4.248.144.161's password:
Connected to 4.248.144.161.
sftp> cd tarea7_backend
sftp> put Servicio.zip
Uploading Servicio.zip to /home/azureuser/tarea7_backend/Servicio.zip
Servicio.zip
Servicio.zip
% 6809      15.7KB/s  00:00
sftp> put prueba.html
Uploading prueba.html to /home/azureuser/tarea7_backend/prueba.html
prueba.html
sftp> put WSClient.js
Uploading WSClient.js to /home/azureuser/tarea7_backend/WSClient.js
WSClient.js
sftp> put usuario_sin_foto.png
Uploading usuario_sin_foto.png to /home/azureuser/tarea7_backend/usuario_sin_foto.png
usuario_sin_foto.png
sftp> ls -l
-rw-*****    ? azureuser azureuser      6809 Nov 22 19:01 Servicio.zip
-rw-*****    ? azureuser azureuser      3692 Nov 22 19:01 WSClient.js
-rw-*****    ? azureuser azureuser      11520 Nov 22 19:01 prueba.html
drwx*****    ? azureuser azureuser      4096 Nov 22 18:56 uploads
-rw-*****    ? azureuser azureuser      1662 Nov 22 19:01 usuario_sin_foto.png
sftp> bye

C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>
C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>
C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>
C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>
C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>
C:\Users\ivan-\Documents\GitHub\Sistemas-Distribuidos\Tarea\Tarea_7>

```

Se descomprime Servicio.zip y se verifica :

```

azureuser@T7-2022630278:~$ mkdir -p ~/tarea7_backend
azureuser@T7-2022630278:~$ ls
Servicio      iniciar.sh      prueba_url.html  tarea2      usuario_sin_foto.png
Servicio.zip  apache-tomcat-8.5.99  prueba_json.html setup_tomcat8.sh  tarea7_backend
azureuser@T7-2022630278:~$ cd ~/tarea7_backend
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio.zip  WSClient.js  prueba.html  uploads  usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ unzip Servicio.zip
Archive:  Servicio.zip
  creating: Servicio/
  inflating: Servicio/compila.bat
  inflating: Servicio/compila.sh
  creating: Servicio/META-INF/
  inflating: Servicio/META-INF/context.xml
  creating: Servicio/servicio/
  inflating: Servicio/servicio/Servicio.java
  inflating: Servicio/servicio/Usuario.java
  inflating: Servicio/servicio/HuboError.java
  creating: Servicio/WEB-INF/
  creating: Servicio/WEB-INF/classes/
  creating: Servicio/WEB-INF/classes/servicio/
  inflating: Servicio/WEB-INF/web.xml
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio  Servicio.zip  WSClient.js  prueba.html  uploads  usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ ls Servicio
META-INF  WEB-INF  compila.bat  compila.sh  servicio
azureuser@T7-2022630278:~/tarea7_backend$
azureuser@T7-2022630278:~/tarea7_backend$ |

```

8.5.3 Configurar context.xml (conexión a MySQL)

Se edita el archivo con: nano Servicio/META-INF/context.xml

```
azureuser@T7-2022630278: ~ C:\Windows\System32\cmd.exe + - x
Memory usage: 69% IPv4 address for eth0: 172.16.0.4
Swap usage: 0%
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Expanded Security Maintenance for Applications is not enabled.
50 updates can be applied immediately.
29 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

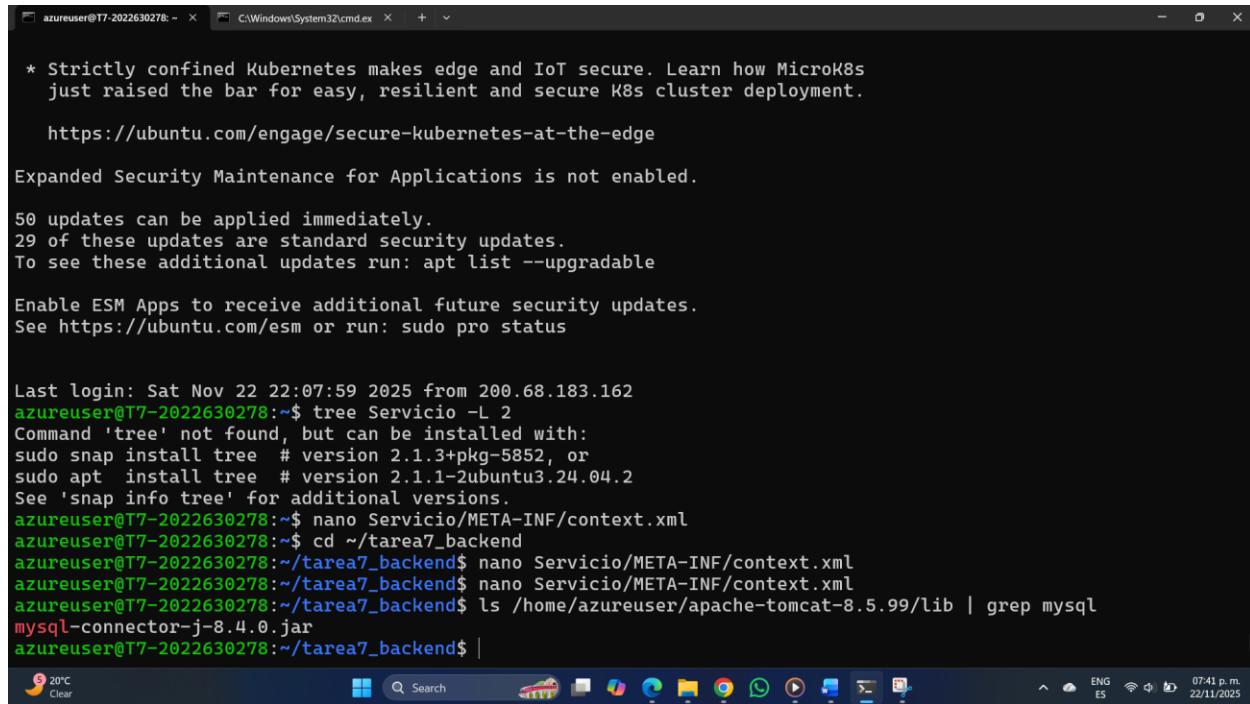
Last login: Sat Nov 22 22:07:59 2025 from 200.68.183.162
azureuser@T7-2022630278:~$ tree Servicio -L 2
Command 'tree' not found, but can be installed with:
sudo snap install tree # version 2.1.3+pkg-5852, or
sudo apt install tree # version 2.1.1-2ubuntu3.24.04.2
See 'snap info tree' for additional versions.
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~/tarea7_backend$ |
```

Nos Asegúramos de colocar el usuario y password de MySQL:

```
az user@T7-2022630278: ~ C:\Windows\System32\cmd.exe + - x
GNU nano 7.2 Servicio/META-INF/context.xml
<Context>
  <Resource name="jdbc/datasource_Servicio" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="x" password=""
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost/servicio_web?serverTimezone=UTC"/>
</Context>
```

Se verifica que el driver mysql-connector esté dentro de CATALINA_HOME/lib:

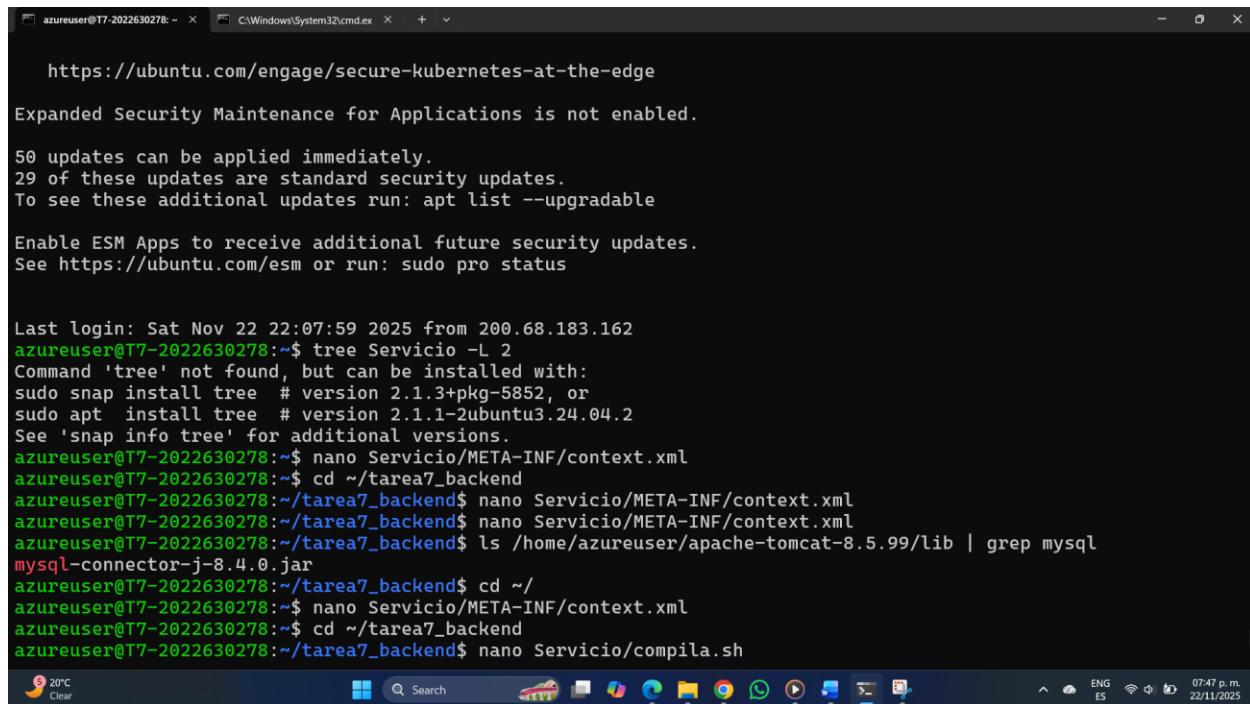
```
ls /home/azureuser/apache-tomcat-8.5.99/lib | grep mysql
```



```
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
Expanded Security Maintenance for Applications is not enabled.  
  
50 updates can be applied immediately.  
29 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sat Nov 22 22:07:59 2025 from 200.68.183.162  
azureuser@T7-2022630278:~$ tree Servicio -L 2  
Command 'tree' not found, but can be installed with:  
sudo snap install tree # version 2.1.3+pkg-5852, or  
sudo apt install tree # version 2.1.1-2ubuntu3.24.04.2  
See 'snap info tree' for additional versions.  
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml  
azureuser@T7-2022630278:~$ cd ~/tarea7_backend  
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml  
azureuser@T7-2022630278:~/tarea7_backend$ ls /home/azureuser/apache-tomcat-8.5.99/lib | grep mysql  
mysql-connector-j-8.4.0.jar  
azureuser@T7-2022630278:~/tarea7_backend$ |
```

8.5.4 Configurar compila.sh

Se edito el script compila.sh: nano Servicio/compila.sh



```
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
Expanded Security Maintenance for Applications is not enabled.  
  
50 updates can be applied immediately.  
29 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Sat Nov 22 22:07:59 2025 from 200.68.183.162  
azureuser@T7-2022630278:~$ tree Servicio -L 2  
Command 'tree' not found, but can be installed with:  
sudo snap install tree # version 2.1.3+pkg-5852, or  
sudo apt install tree # version 2.1.1-2ubuntu3.24.04.2  
See 'snap info tree' for additional versions.  
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml  
azureuser@T7-2022630278:~$ cd ~/tarea7_backend  
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml  
azureuser@T7-2022630278:~/tarea7_backend$ ls /home/azureuser/apache-tomcat-8.5.99/lib | grep mysql  
mysql-connector-j-8.4.0.jar  
azureuser@T7-2022630278:~/tarea7_backend$ cd ~  
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml  
azureuser@T7-2022630278:~$ cd ~/tarea7_backend  
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/compila.sh
```

Reemplaza la línea CATALINA_HOME:

The screenshot shows a terminal window titled "Servicio/compila.sh" running on a Windows system. The terminal contains a shell script named "compila.sh" which performs the following steps:

```
GNU nano 7.2                               Servicio/compila.sh
#Definir y descomentar a siguiente variable de entorno:
export CATALINA_HOME=/home/azureuser/apache-tomcat-8.5.99
javac -cp "$CATALINA_HOME/lib/*:" servicio/Servicio.java
rm WEB-INF/classes/servicio/*
cp servicio/*.class WEB-INF/classes/servicio/.
jar cvf Servicio.war WEB-INF META-INF
rm -rf $CATALINA_HOME/webapps/Servicio.war $CATALINA_HOME/webapps/Servicio
cp Servicio.war $CATALINA_HOME/webapps/
```

The terminal interface includes a menu bar with options like Help, Write Out, Where Is, Cut, Paste, Execute, Location, and Undo. Below the menu is a toolbar with icons for Clear, Search, and various system functions. The status bar at the bottom shows the date and time: 07:47 p.m. 22/11/2025.

Se regresa al directorio del servicio y se da permisos de ejecución al script:

```
cd ~/tarea7_backend/Servicio
```

```
chmod +x compila.sh
```

The screenshot shows a terminal window with a history of commands. The user has navigated to the directory containing the compiled Java application and executed the script. The output shows the application's context XML files and the MySQL connector jar file, followed by the execution of the "compila.sh" script which sets environment variables and runs the Java application.

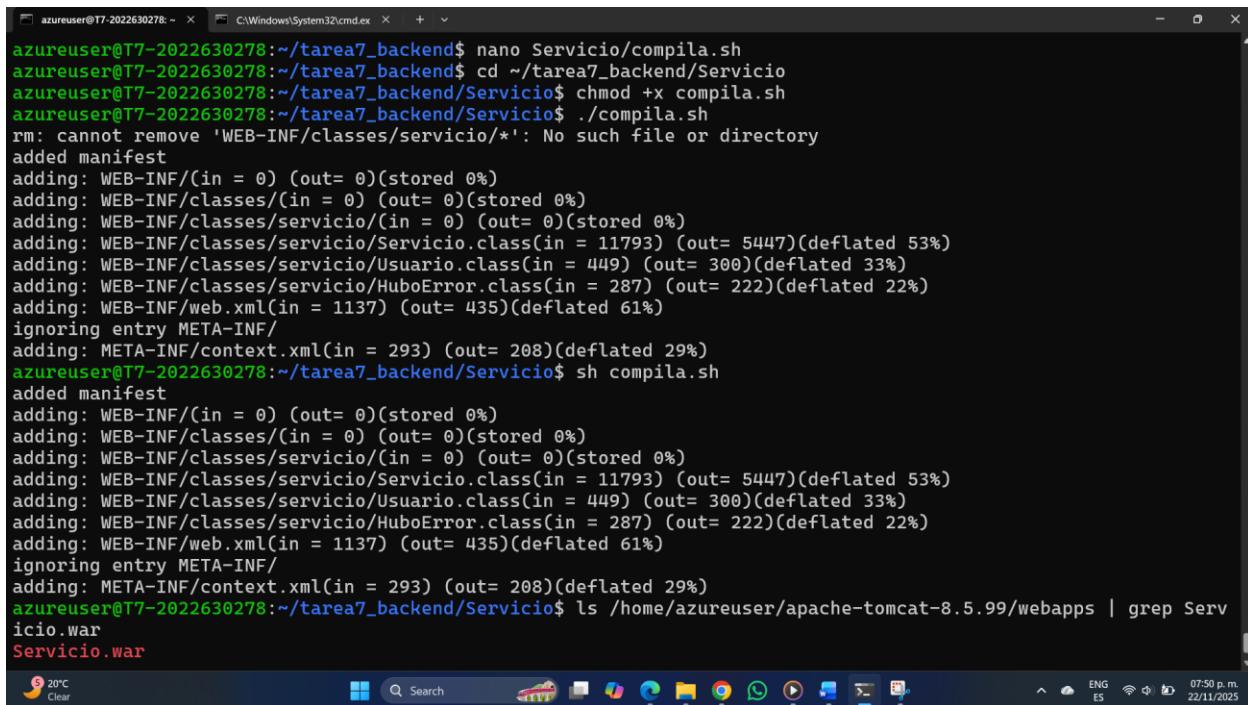
```
Last login: Sat Nov 22 22:07:59 2025 from 200.68.183.162
azureuser@T7-2022630278:~$ tree Servicio -L 2
Command 'tree' not found, but can be installed with:
sudo snap install tree # version 2.1.3+pkg-5852, or
sudo apt install tree # version 2.1.1-2ubuntu3.24.04.2
See 'snap info tree' for additional versions.
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~$ cd ~/tarea7_backend
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~/tarea7_backend$ ls /home/azureuser/apache-tomcat-8.5.99/lib | grep mysql
mysql-connector-j-8.4.0.jar
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/
azureuser@T7-2022630278:~$ nano Servicio/META-INF/context.xml
azureuser@T7-2022630278:~$ cd ~/tarea7_backend
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/compila.sh
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/
azureuser@T7-2022630278:~$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF  WEB-INF  compila.bat  compila.sh  servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ echo CATALINA_HOME
CATALINA_HOME
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/compila.sh
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ |
```

8.5.5 Compilar el servicio

Se ejecuto:

```
./compila.sh
```

```
sh compila.sh
```



```
azureuser@T7-2022630278:~/tarea7_backend$ nano Servicio/compila.sh
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ./compila.sh
rm: cannot remove 'WEB-INF/classes/servicio/*': No such file or directory
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 11793) (out= 5447)(deflated 53%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sh compila.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 11793) (out= 5447)(deflated 53%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls /home/azureuser/apache-tomcat-8.5.99/webapps | grep Servicio.war
Servicio.war
```

Se comprobó que el WAR fue copiado y Tomcat debería auto-desplegarlo en segundos (crear carpeta Servicio). Se verifica:

```
ls /home/azureuser/apache-tomcat-8.5.99/webapps | grep Servicio.war
```

```

azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls -l /home/azureuser/apache-tomcat-8.5.99/webapps
total 12
drwxrwxr-x 2 azureuser azureuser 4096 Oct 21 16:40 ROOT
-rw-rw-r-- 1 azureuser azureuser 8038 Nov 23 01:49 Servicio.war
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/Servicio
bash: cd: /home/azureuser/tarea7_backend/Servicio/Servicio: No such file or directory
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls
HuboError.class HuboError.java Servicio.class Servicio.java Usuario.class Usuario.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ./compila.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 11793) (out= 5447)(deflated 53%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls -l /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio
.war
-rw-rw-r-- 1 azureuser azureuser 8038 Nov 23 01:54 /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio.war
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ |

```

The screenshot shows a Windows terminal window with a black background and white text. It displays a command-line session where a Java application is compiled into a WAR file named 'Servicio.war'. The session starts with listing files in the '/home/azureuser/apache-tomcat-8.5.99/webapps' directory, then navigating to the 'Servicio' directory. It then lists files in 'WEB-INF' and 'servicio' sub-directories. The 'compila.sh' script is run, which adds manifest information and compresses Java classes and XML files into the WAR file. Finally, the 'Servicio.war' file is listed again in the webapps directory.

`ls /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio`

```

azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sleep 5
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ /home/azureuser/apache-tomcat-8.5.99/bin/startup.sh
Using CATALINA_BASE:   /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls /home/azureuser/apache-tomcat-8.5.99/webapps/Servicio
META-INF WEB-INF
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ |

```

The screenshot shows a Windows terminal window with a black background and white text. It displays a command-line session where the 'startup.sh' script is run to start a Tomcat server. After a 5-second delay, the 'ls' command is run in the '/home/azureuser/apache-tomcat-8.5.99/webapps/Servicio' directory, showing the presence of 'META-INF' and 'WEB-INF' directories. The session ends with a final '|'. This indicates that the previous command (likely 'compila.sh') was followed by a pipe character, indicating it was part of a pipeline.

8.5.6 Copiar archivos front-end (prueba.html, WSClient.js, usuario_sin_foto.png)

Los archivos están pensados para estar en ROOT (la app que responde en /):

Destino:

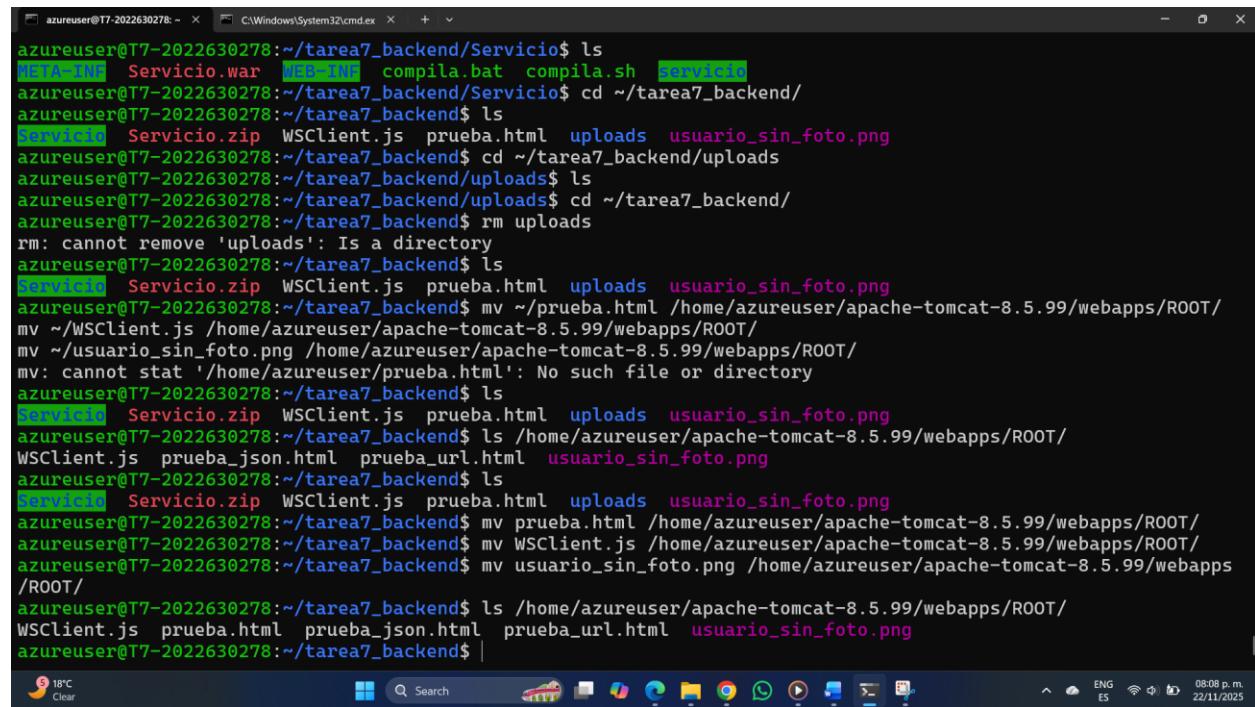
/home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/

Se subieron los archivos vía SFTP a otro directorio, entonces dentro de la VM se ejecuto:

```
mv prueba.html /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
```

```
mv WSClient.js /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
```

```
mv usuario_sin_foto.png /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
```



```
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip WSClient.js prueba.html uploads usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/tarea7_backend/uploads
azureuser@T7-2022630278:~/tarea7_backend/uploads$ ls
azureuser@T7-2022630278:~/tarea7_backend/uploads$ cd ~/tarea7_backend/
azureuser@T7-2022630278:~/tarea7_backend$ rm uploads
rm: cannot remove 'uploads': Is a directory
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip WSClient.js prueba.html uploads usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ mv ~/prueba.html /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
mv ~/WSClient.js /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
mv ~/usuario_sin_foto.png /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
mv: cannot stat '/home/azureuser/prueba.html': No such file or directory
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip WSClient.js prueba.html uploads usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ ls /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
WSClient.js prueba_json.html prueba_url.html usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip WSClient.js prueba.html uploads usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ mv prueba.html /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
azureuser@T7-2022630278:~/tarea7_backend$ mv WSClient.js /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
azureuser@T7-2022630278:~/tarea7_backend$ mv usuario_sin_foto.png /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
azureuser@T7-2022630278:~/tarea7_backend$ ls /home/azureuser/apache-tomcat-8.5.99/webapps/ROOT/
WSClient.js prueba.html prueba_json.html prueba_url.html usuario_sin_foto.png
azureuser@T7-2022630278:~/tarea7_backend$ |
```

Imagen 24: Explorador/terminal mostrando archivos en webapps/ROOT.

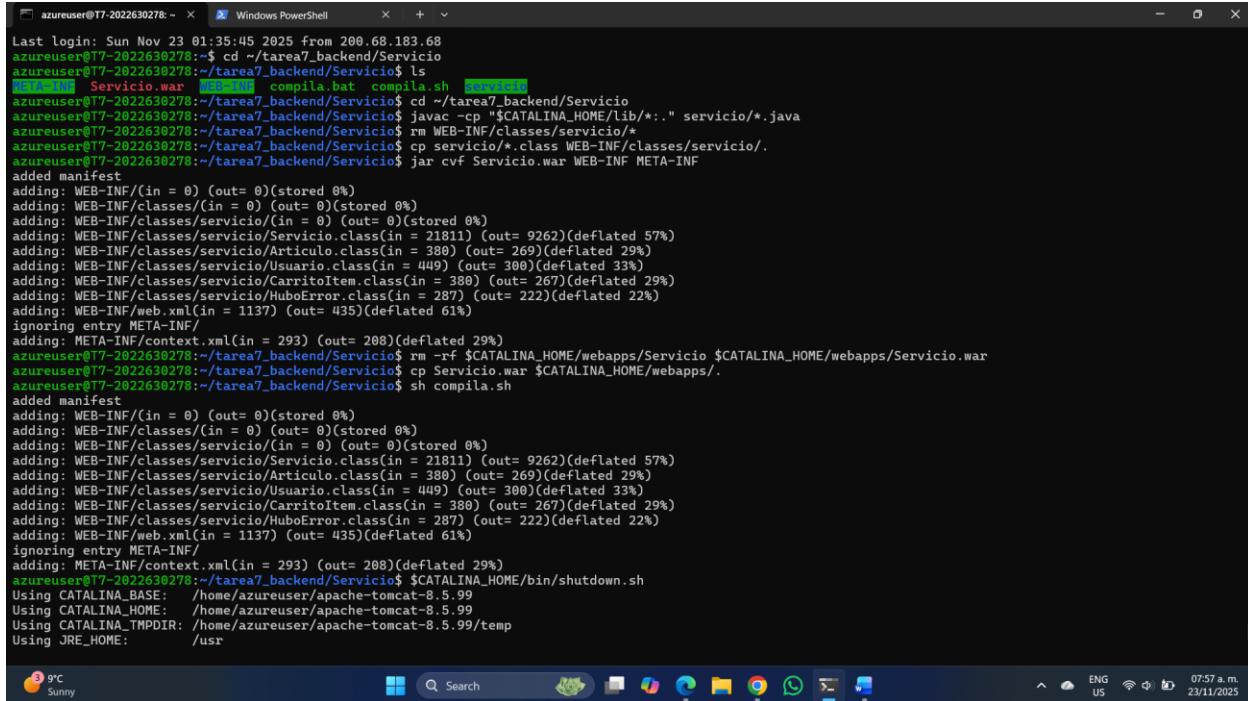
8.5.7 8. (Opcional) Reiniciar Tomcat si hubo cambios críticos

Si el WAR se desplegó correctamente normalmente no hace falta, pero si cambiaste libs en lib/, sí conviene reiniciar.

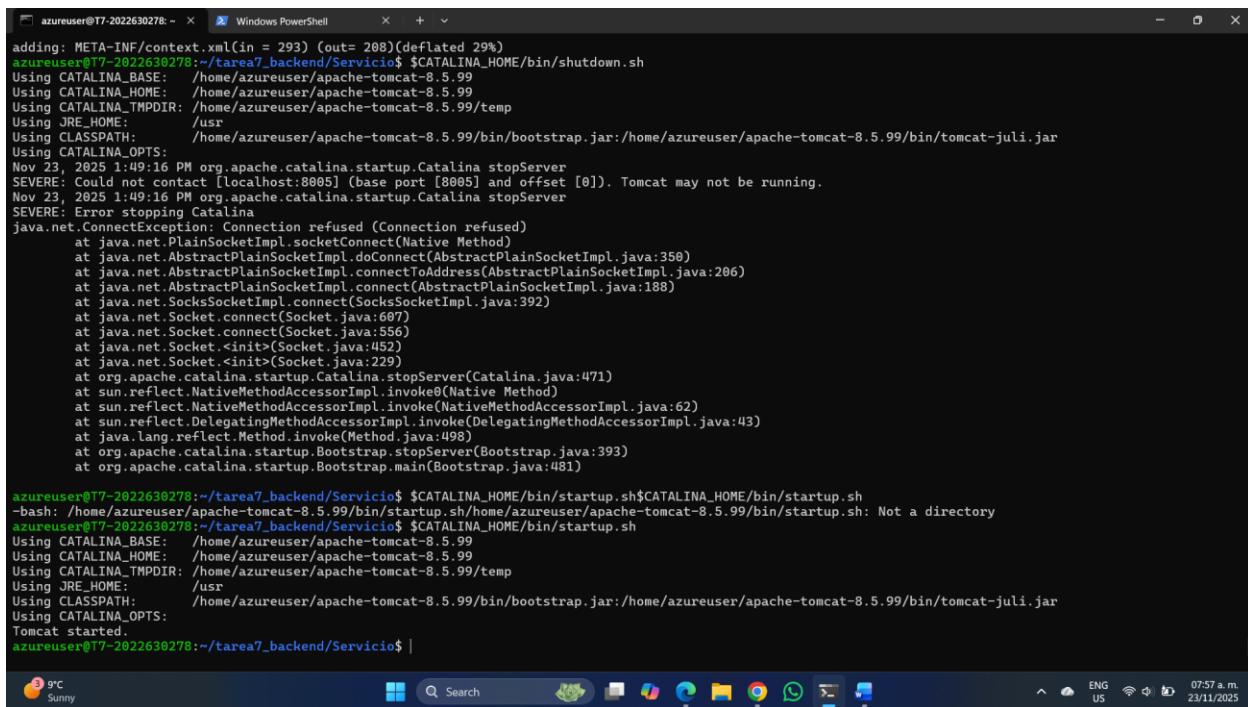
```
/home/azureuser/apache-tomcat-8.5.99/bin/shutdown.sh
```

```
sleep 5
```

/home/azureuser/apache-tomcat-8.5.99/bin/startup.sh



```
azureuser@T7-2022630278:~ $ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
Servicio.war  compil.bat  compil.sh  service
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ javac -cp "$CATALINA_HOME/lib/*.*" servicio/*.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ rm WEB-INF/classes/servicio/*
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cp servicio/*.class WEB-INF/classes/servicio/.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ jar cvf Servicio.war WEB-INF META-INF
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
adding: WEB-INF/classes/servicio/Articulo.class(in = 380) (out= 269)(deflated 29%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/CarritoItem.class(in = 380) (out= 267)(deflated 29%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ rm -rf $CATALINA_HOME/webapps/Servicio $CATALINA_HOME/webapps/Servicio.war
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cp Servicio.war $CATALINA_HOME/webapps/.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sh compil.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
adding: WEB-INF/classes/servicio/Articulo.class(in = 380) (out= 269)(deflated 29%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/CarritoItem.class(in = 380) (out= 267)(deflated 29%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/shutdown.sh
Using CATALINA_BASE:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME:        /usr
```



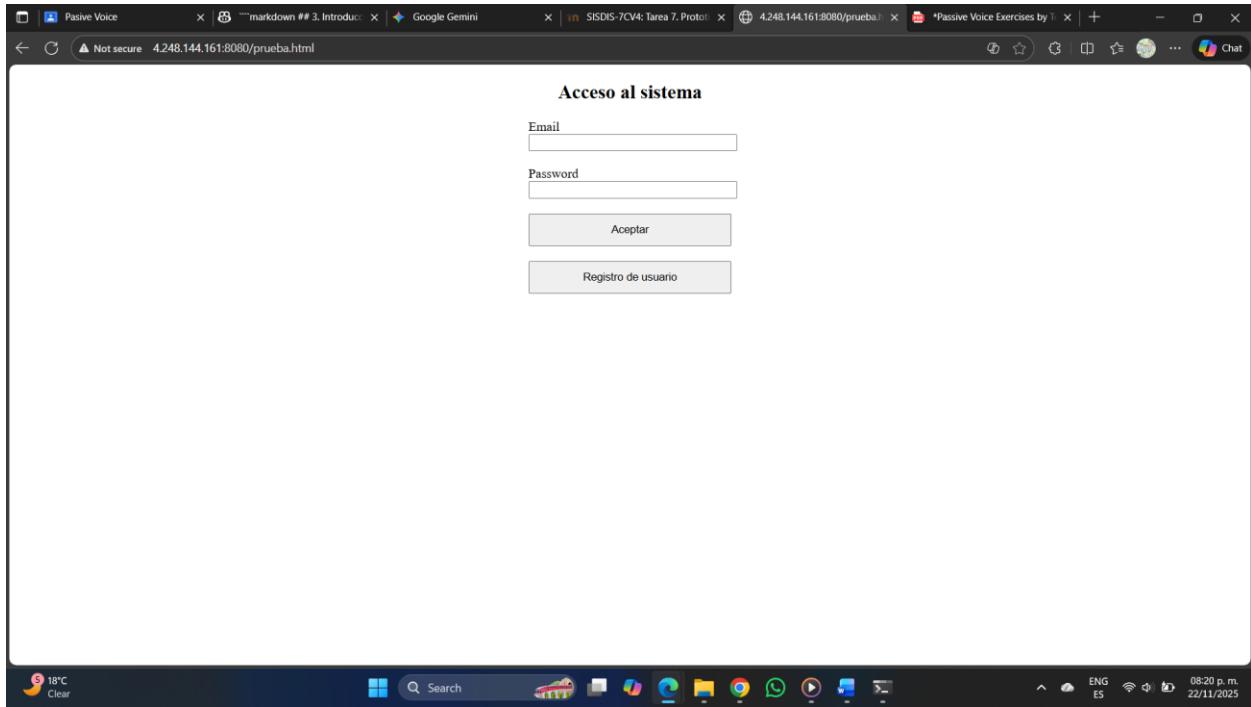
```
azureuser@T7-2022630278:~ $ cd ~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/shutdown.sh
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/shutdown.sh
Using CATALINA_BASE:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME:        /usr
Using CLASSPATH:      /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Nov 23, 2025 1:49:16 PM org.apache.catalina.startup.Catalina stopServer
SEVERE: Could not contact [localhost:8005] (base port [8005] and offset [0]). Tomcat may not be running.
Nov 23, 2025 1:49:16 PM org.apache.catalina.startup.Catalina stopServer
SEVERE: Error stopping Catalina
java.net.ConnectException: Connection refused (Connection refused)
    at java.net.PlainSocketImpl.socketConnect(Native Method)
    at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
    at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
    at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
    at java.net.Socket.connect(Socket.java:697)
    at java.net.Socket.connect(Socket.java:556)
    at java.net.Socket.<init>(Socket.java:452)
    at java.net.Socket.<init>(Socket.java:229)
    at org.apache.catalina.startup.Catalina.stopServer(Catalina.java:471)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:393)
    at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:481)

azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/startup.sh$CATALINA_HOME/bin/startup.sh
-bash: /home/azureuser/apache-tomcat-8.5.99/bin/startup.sh: home/azureuser/apache-tomcat-8.5.99/bin/startup.sh: Not a directory
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/startup.sh
Using CATALINA_BASE:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME:  /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME:        /usr
Using CLASSPATH:      /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ |
```

8.5.8 Probar acceso desde navegador

URL base (suponiendo que la IP pública de tu VM es, por ejemplo, 20.30.40.50):

<http://20.30.40.50:8080/prueba.html>

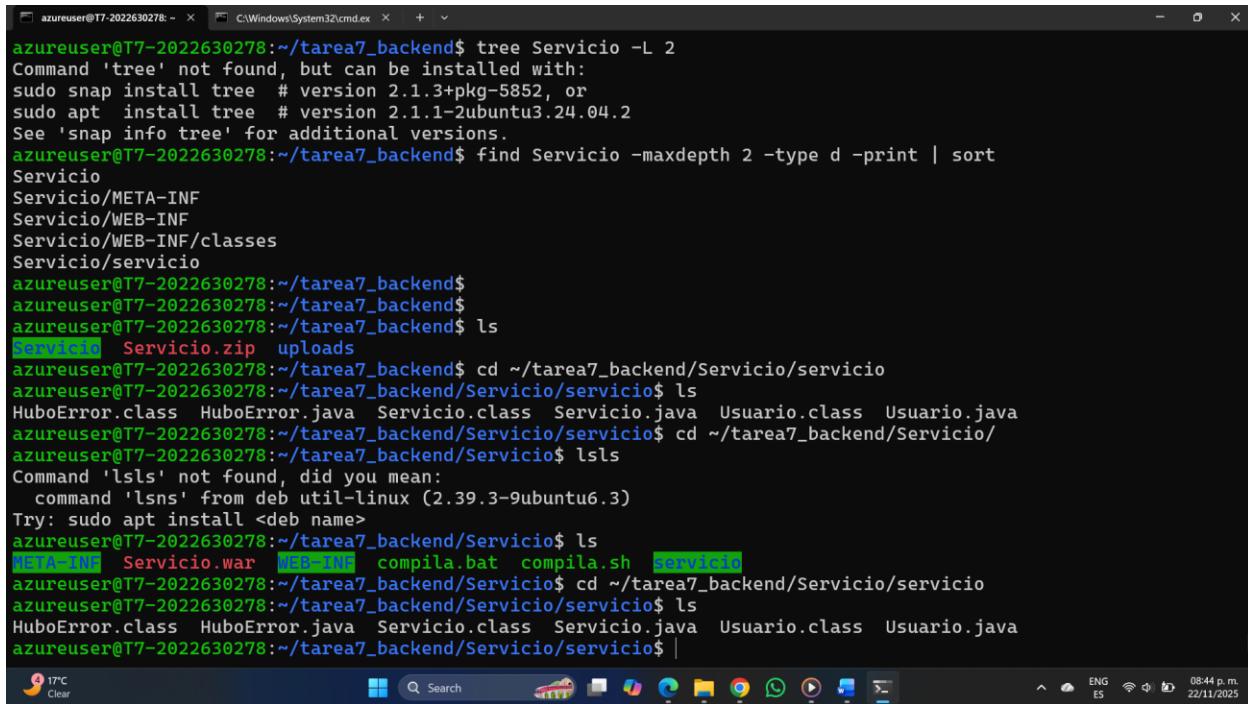


8.6 Modificación/Actualización de Servicio.java

Se integran los nuevos endpoints y POJOs (Tarea 7). Acciones:

1. Nos movimos a la carpeta Servicio usando:

```
cd ~/tarea7_backend/Servicio/servicio
```



```
azureuser@T7-2022630278:~/tarea7_backend$ tree Servicio -L 2
Command 'tree' not found, but can be installed with:
sudo snap install tree  # version 2.1.3+pkg-5852, or
sudo apt install tree  # version 2.1.1-2ubuntu3.24.04.2
See 'snap info tree' for additional versions.
azureuser@T7-2022630278:~/tarea7_backend$ find Servicio -maxdepth 2 -type d -print | sort
Servicio
Servicio/META-INF
Servicio/WEB-INF
Servicio/WEB-INF/classes
Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend$ 
azureuser@T7-2022630278:~/tarea7_backend$ 
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip uploads
azureuser@T7-2022630278:~/tarea7_backend$ cd ~/tarea7_backend/Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls
HuboError.class HuboError.java Servicio.class Servicio.java Usuario.class Usuario.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ cd ~/tarea7_backend/Servicio/
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
Command 'lsls' not found, did you mean:
  command 'lsns' from deb util-linux (2.39.3-9ubuntu6.3)
Try: sudo apt install <deb name>
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls
HuboError.class HuboError.java Servicio.class Servicio.java Usuario.class Usuario.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ |
```

Trabajaremos DENTRO de Servicio/, agregando nuevos .java en Servicio/servicio/.

2. Archivos nuevos que debes crear (POJOs) Necesitamos dos clases nuevas para el dominio del e-commerce:
 - Articulo.java: representa artículos del stock (con foto como byte[]])

The screenshot shows a Windows desktop environment. In the foreground, a terminal window titled "Articulo.java *" is open, displaying Java code for a class named "Articulo". The code defines a class with attributes: id_articulo (Integer), nombre (String), descripcion (String), precio (Double), cantidad (Integer), and foto (byte[]). The terminal window also shows command-line navigation and nano editor status. Below the terminal is a standard Windows taskbar with icons for Start, Search, File Explorer, Edge, Google Chrome, and other applications. The system tray shows the date and time (08:48 p.m. 22/11/2025) and battery status.

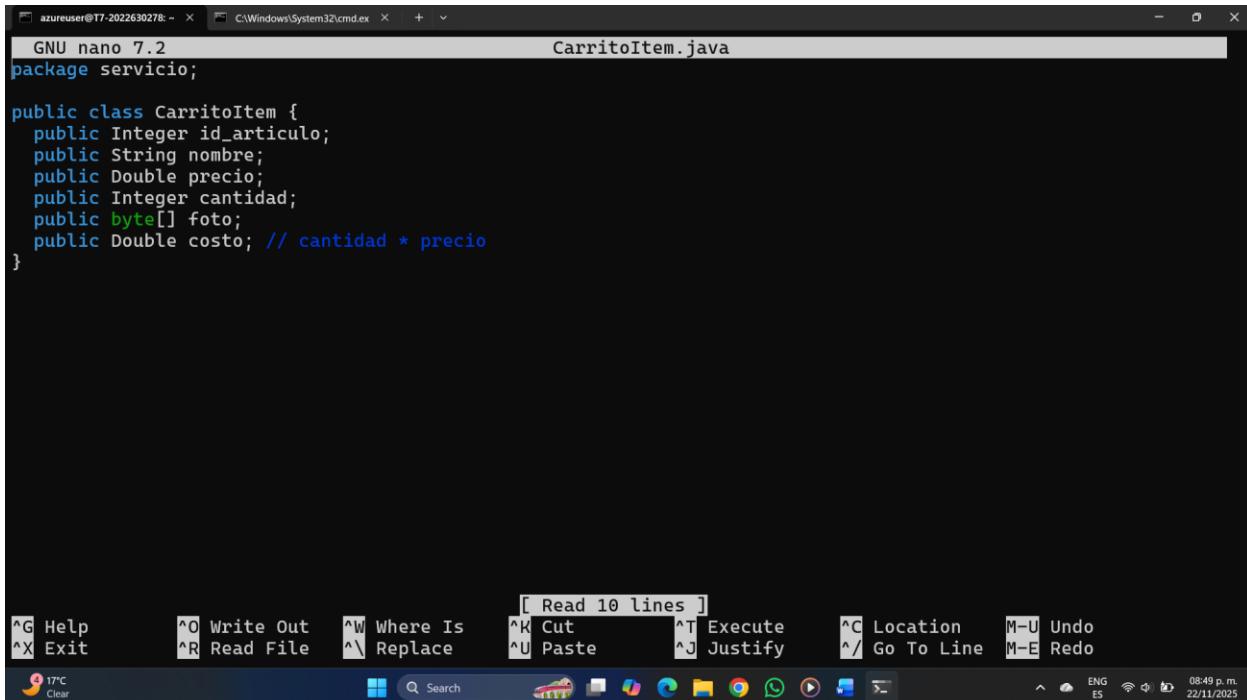
```
GNU nano 7.2                               Articulo.java *
package servicio;

public class Articulo {
    public Integer id_articulo;
    public String nombre;
    public String descripcion;
    public Double precio;
    public Integer cantidad;
    public byte[] foto;
}

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
Clear 17°C
```

Imagen 35: Editor mostrando el código de Articulo.java.

- CarritoItem.java: representa un renglón del carrito con costo (cantidad x precio)



```

azureuser@T7-2022630278: ~ x C:\Windows\System32\cmd.ex x + -
GNU nano 7.2                                     CarritoItem.java
package servicio;

public class CarritoItem {
    public Integer id_articulo;
    public String nombre;
    public Double precio;
    public Integer cantidad;
    public byte[] foto;
    public Double costo; // cantidad * precio
}

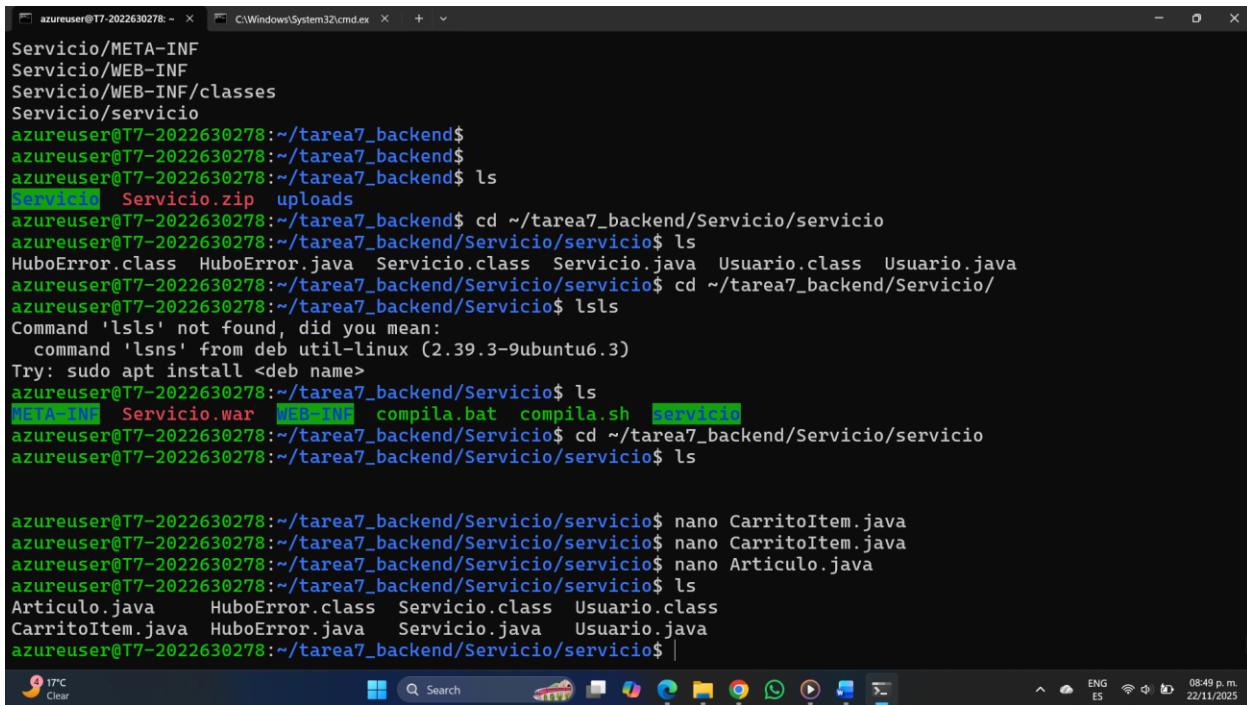
```

[Read 10 lines]

Keyboard Shortcuts:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^X Exit
- ^R Read File
- ^A Replace
- ^K Cut
- ^U Paste
- ^T Execute
- ^J Justify
- ^C Location
- ^Y Go To Line
- M-U Undo
- M-E Redo

17°C Clear



```

Servicio/META-INF
Servicio/WEB-INF
Servicio/WEB-INF/classes
Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend$ 
azureuser@T7-2022630278:~/tarea7_backend$ 
azureuser@T7-2022630278:~/tarea7_backend$ ls
Servicio Servicio.zip uploads
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls
HuboError.class HuboError.java Servicio.class Servicio.java Usuario.class Usuario.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ cd ~/tarea7_backend/Servicio/
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
Command 'lsls' not found, did you mean:
  command 'lsns' from deb util-linux (2.39.3-9ubuntu6.3)
Try: sudo apt install <deb name>
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls

azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano CarritoItem.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano CarritoItem.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano Articulo.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls
Articulo.java HuboError.class Servicio.class Usuario.class
CarritoItem.java HuboError.java Servicio.java Usuario.java
azureuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ |

```

17°C Clear

Imagen 36: Editor mostrando el código de CarritoItem.java.

3. Cambiamos en Servicio.java:

- Ajustar login para devolver id_usuario además de token (el front-end lo usará).
- Añadir endpoints de back-end requeridos:
 - alta_articulo (POST, JSON)
 - consulta_articulos (GET)
 - compra_articulo (POST, JSON, con transacción)
 - elimina_articulo_carrito_compra (DELETE, con transacción)
 - elimina_carrito_compra (DELETE, con transacción)
 - modifica_carrito_compra (PUT, con transacción)
 - consulta_carrito_compra (GET) para que el front-end liste los artículos y total

```
azuser@T7-2022630278:~ C:\Windows\System32\cmd.exe + - Servicio/WEB-INF  
Servicio/WEB-INF/classes  
Servicio/servicio  
azuser@T7-2022630278:~/tarea7_backend$ azuser@T7-2022630278:~/tarea7_backend$ azuser@T7-2022630278:~/tarea7_backend$ ls  
Servicio.zip uploads  
azuser@T7-2022630278:~/tarea7_backend$ cd ~/tarea7_backend/Servicio/servicio  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls  
HuboError.class HuboError.java Servicio.class Servicio.java Usuario.class Usuario.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ cd ~/tarea7_backend/Servicio/  
azuser@T7-2022630278:~/tarea7_backend/Servicio$ ls  
Command 'lsls' not found, did you mean:  
  command 'lsns' from deb util-linux (2.39.3-9ubuntu6.3)  
Try: sudo apt install <deb name>  
azuser@T7-2022630278:~/tarea7_backend/Servicio$ ls  
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio  
azuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/servicio  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls  
  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano CarritoItem.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano CarritoItem.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano Articulo.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ ls  
Articulo.java HuboError.class Servicio.class Usuario.class  
CarritoItem.java HuboError.java Servicio.java Usuario.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ nano Servicio.java  
azuser@T7-2022630278:~/tarea7_backend/Servicio/servicio$ |
```

```
azuser@17-2022630278:~ C:\Windows\System32\cmd.exe + - Service.java *
```

```
GNU nano 7.2
/*
 * Servicio.java
 * Servicio web REST
 * Carlos Pineda Guerrero. 2025
 */

package servicio;

import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.DELETE;
import javax.ws.rs.Path;
import javax.ws.rs.Consumes;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.QueryParam;
import javax.ws.rs.core.Response;
import javax.ws.rs.core.Application;

import java.sql.*;
import javax.sql.DataSource;
import javax.naming.Context;
import javax.naming.InitialContext,
```

Imagen 22: Editor mostrando parte final de Servicio.java con nuevos métodos.

8.7 Scripts de compilación (compila.sh y compila.bat)

4. Ajustes de compilación: compila.sh y dependencias

- Asegúramos tener en \$CATALINA_HOME/lib:
 - mysql-connector-j (com.mysql.cj.jdbc.Driver)
 - jersey-container-servlet (o jersey-servlet), jersey-core/servidor según tu Tarea 2
 - Los jars que te pidieron para Jackson/Jersey JSON:
 - jersey-media-json-jackson-2.24.jar
 - jersey-entity-filtering-2.24.jar
 - jackson-core-2.6.3.jar
 - jackson-databind-2.6.3.jar
 - jackson-annotations-2.6.3.jar
 - jackson-jaxrs-json-provider-2.6.3.jar
 - jackson-jaxrs-base-2.6.3.jar
 - jackson-module-jaxb-annotations-2.6.3.jar
- Editamos Servicio/compila.sh para compilar todos los .java del paquete servicio

Definimos la ruta de Tomcat (ajustar según instalación) y compilamos (incluye todas las clases del paquete servicio)

```
echo $CATALINA_HOME
```

```
javac -cp "$CATALINA_HOME/lib/*:." servicio/*.java
```

Limpiamos y copiamos las clases compiladas:

```
rm WEB-INF/classes/servicio/*
```

```
cp servicio/*.class WEB-INF/classes/servicio/.
```

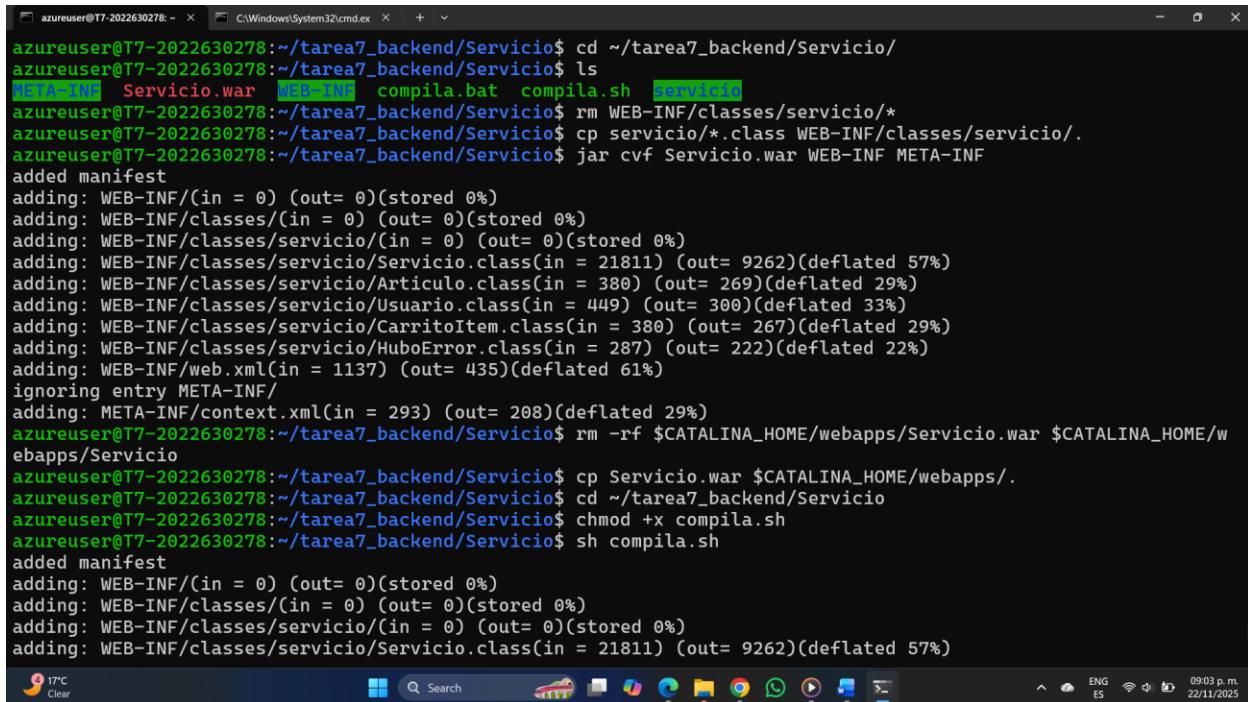
Empaquetamos WAR:

```
jar cvf Servicio.war WEB-INF META-INF
```

Desplegamos:

```
rm -rf $CATALINA_HOME/webapps/Servicio.war $CATALINA_HOME/webapps/Servicio
```

```
cp Servicio.war $CATALINA_HOME/webapps/.
```



```
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio/
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ ls
META-INF Servicio.war WEB-INF compila.bat compila.sh servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ rm WEB-INF/classes/servicio/*
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cp servicio/*.class WEB-INF/classes/servicio/.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ jar cvf Servicio.war WEB-INF META-INF
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
adding: WEB-INF/classes/servicio/Articulo.class(in = 380) (out= 269)(deflated 29%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/CarritoItem.class(in = 380) (out= 267)(deflated 29%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ rm -rf $CATALINA_HOME/webapps/Servicio.war $CATALINA_HOME/webapps/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cp Servicio.war $CATALINA_HOME/webapps/.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sh compila.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
```

Imagen 23: Consola mostrando ejecución exitosa de compila.sh (o compila.bat).

- Damos permisos de ejecución y compila:
 - cd ~/tarea7_backend/Servicio
 - chmod +x compila.sh
 - sh compila.sh
- Tomcat redeploya automáticamente el WAR copiado; si no, reinicia:
 - \$CATALINA_HOME/bin/shutdown.sh
 - \$CATALINA_HOME/bin/startup.sh

```
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sh compila.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
adding: WEB-INF/classes/servicio/Articulo.class(in = 380) (out= 269)(deflated 29%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/CarritoItem.class(in = 380) (out= 267)(deflated 29%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/shutdown.sh
Using CATALINA_BASE: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/startup.sh
Using CATALINA_BASE: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
```

```
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ cd ~/tarea7_backend/Servicio
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ chmod +x compila.sh
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ sh compila.sh
added manifest
adding: WEB-INF/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/(in = 0) (out= 0)(stored 0%)
adding: WEB-INF/classes/servicio/Servicio.class(in = 21811) (out= 9262)(deflated 57%)
adding: WEB-INF/classes/servicio/Articulo.class(in = 380) (out= 269)(deflated 29%)
adding: WEB-INF/classes/servicio/Usuario.class(in = 449) (out= 300)(deflated 33%)
adding: WEB-INF/classes/servicio/CarritoItem.class(in = 380) (out= 267)(deflated 29%)
adding: WEB-INF/classes/servicio/HuboError.class(in = 287) (out= 222)(deflated 22%)
adding: WEB-INF/web.xml(in = 1137) (out= 435)(deflated 61%)
ignoring entry META-INF/
adding: META-INF/context.xml(in = 293) (out= 208)(deflated 29%)
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/shutdown.sh
Using CATALINA_BASE: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ $CATALINA_HOME/bin/startup.sh
Using CATALINA_BASE: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_HOME: /home/azureuser/apache-tomcat-8.5.99
Using CATALINA_TMPDIR: /home/azureuser/apache-tomcat-8.5.99/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/azureuser/apache-tomcat-8.5.99/bin/bootstrap.jar:/home/azureuser/apache-tomcat-8.5.99/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
azureuser@T7-2022630278:~/tarea7_backend/Servicio$ |
```

9 Implementación del Back-End

Esta sección detalla la ampliación del servicio REST desarrollado en la Tarea 2 (gestión de usuarios) para construir el prototipo de comercio electrónico solicitado en la Tarea 7. Se incorporan artículos, fotos de artículos, un carrito de compra y operaciones transaccionales, cumpliendo los requerimientos funcionales y no funcionales establecidos.

9.1 Estructura del proyecto

Después de descomprimir Servicio.zip y agregar las nuevas clases, la estructura queda:

Servicio/

 └─ META-INF/

 └─ context.xml

 └─ WEB-INF/

 └─ classes/

 | └─ servicio/ (archivos .class generados tras compilar)

 └─ web.xml

 └─ compila.sh

 └─ compila.bat

 └─ servicio/

 └─ HuboError.java

 └─ Servicio.java

 └─ Usuario.java

 └─ Articulo.java (nuevo)

 └─ CarritoItem.java (nuevo)

Nota: Al ejecutar compila.sh se empaqueta Servicio.war que Tomcat despliega en \$CATALINA_HOME/webapps/Servicio.

- **9.2 Clases añadidas o modificadas**

Clase	Rol	Comentarios
Servicio.java	Controlador REST principal	Se agregan endpoints de artículos y carrito; maneja transacciones.
Usuario.java	POJO de usuario	Sin cambios estructurales salvo uso extendido del token.
Articulo.java	POJO de artículo	Incluye id_articulo, datos básicos y foto (byte[]).
CarritoItem.java	POJO de ítem en carrito	No representa una tabla directa; se arma desde joins y calcula costo.
HuboError.java	Contenedor de error	Permite respuestas uniformes JSON con mensaje.

9.2 Modificación del método login

Para optimizar el flujo se cambió el SELECT (antes devolvía sólo existencia). Ahora se obtiene también id_usuario, evitando consultas posteriores.

Consulta original (Tarea 2):

```
SELECT 1 FROM usuarios WHERE email=? AND password=?
```

Consulta nueva:

```
SELECT id_usuario FROM usuarios WHERE email=? AND password=?
```

La respuesta JSON ahora incluye:

```
{
```

```
    "token": "AB12CD34EF56GH78IJ90",
```

```

    "id_usuario": 5
}

```

9.3 Endpoints añadidos y mapeo a requerimientos funcionales

Endpoint (RF Back-End)	Método	Ruta relativa (/Servicio/res t/ws/...)	Query Params	Body JSON	Autenticaci ón (token)	Descripción / Observaciones
RF1 alta_articulo	POST	alta_articulo	id_usuario, token	Articul o (sin id articulo)	Sí	Inserta artículo en stock y foto en fotos_articulos.
RF2 consulta_articulos	GET	consulta_articulos	palabra, id_usuario , token	—	Sí	Búsqueda con LIKE en nombre o descripción. Devuelve lista con campos requeridos.
RF3 compra_articulo	POST	compra_articulo	id_usuario, token	{id_articulo, cantidad}	Sí	Verifica stock, inserta/actualiza carrito y descuenta stock. Errores RF3.3.
RF4 elimina_articulo_carrito_compra	DELETE	elimina_articulo_carrito_compra	id_usuario, id_articulo, token	—	Sí	Regresa cantidad al stock y elimina ítem del carrito.
RF5 elimina_carrito_compra	DELETE	elimina_carrito_compra	id_usuario, token	—	Sí	Restituye stock de cada artículo y limpia el carrito.
RF6 modifica_carrito_compra	PUT	modifica_carrito_compra	id_usuario, id_articulo, incremento, token	(vacío)	Sí	Incrementa/decremente cantidad en carrito; valida stock o límites.
(Apoyo) consulta_carrito_compra	GET	consulta_carrito_compra	id_usuario, token	—	Sí	Devuelve ítems del carrito y total. Facilita Front-End RF5–RF8.

Imagen 38. Tabla de endpoints en el editor.

9.4 Correspondencia con requerimientos funcionales

Requerimiento original	Implementación en código

RF1 Alta artículo	Método alta_articulo
RF2 Consulta artículos	Método consulta_articulos
RF3 Compra artículo	Método compra_articulo (transacción + validaciones)
RF4 Elimina artículo carrito	Método elimina_articulo_carrito_compra
RF5 Elimina carrito completo	Método elimina_carrito_compra
RF6 Modifica cantidad carrito	Método modifica_carrito_compra
(Front-End soporte) Carrito con total	Método consulta_carrito_compra

Imagen 42. Tabla de correspondencia en el editor.

10 Implementación del Front-end (prueba.html y JS) para la Tarea 7

Se extiende el prueba.html de la Tarea 2 para incorporar las tres nuevas vistas relacionadas con el comercio electrónico: Captura de artículo, Compra de artículos y Carrito de compra, cumpliendo los requerimientos funcionales del front-end (1–8).

10.1 Enfoque y organización

1. Página única (SPA ligera) mediante mostrar/ocultar <div> (patrón ya existente).
2. Reutilización de variables globales: email, token, y nueva id_usuario.
3. Reutilización de WSClient.js para todas las peticiones (uniformidad).
4. Estructura clara de secciones: login, alta_usuario, consulta_usuario, menu, captura_articulo, compra_articulos, carrito.
5. Manejo de imágenes base64 aprovechando la misma función readSingleFile.
(Se podría separar en foto_usuario y foto_articulo como mejora futura.)

Imagen 44. Secciones nuevas en prueba.html.

6. 10.2 Correspondencia con requerimientos front-end

Requerimiento Front- End		Implementación
1 Captura de artículo		Vista captura_articulo + función alta_articulo()
2 Compra de artículos (búsqueda)		Vista compra_articulos + consulta_articulos()
3 Mostrar datos y controles por artículo		Render dinámico: imagen, nombre, descripción, precio, cantidad, botones + / - / Compra

4 Botón Compra	Botón “Compra” → comprar_articulo(id)
5 Carrito de compra con total	Vista carrito + consulta_carrito() calcula y muestra total
6 Eliminar artículo del carrito	Botón “Eliminar artículo” → elimina_articulo_carrito(id)
7 Eliminar carrito completo	Botón “Eliminar carrito” → elimina_carrito()
8 Seguir comprando	Botón “Seguir comprando” regresa a compra_articulos

Imagen 45. Captura de la vista “Compra de artículos” mostrando lista y botones.

7. 10.3 Flujo de interacción

1. Login (obtiene token + id_usuario).
2. Menú principal: acceso a perfil, captura y compra.
3. Captura de artículo: envía POST y regresa al menú.
4. Compra: búsqueda → listado → compras individuales → acceso al carrito.
5. Carrito: ver artículos + total, modificar cantidades, eliminar ítem o vaciar.
6. Seguir comprando retorna a la búsqueda sin perder sesión.

Imagen 46. Diagrama simple del flujo (opcional).

10.2 Funciones JavaScript clave

Función	Endpoint	Propósito	Observaciones

login()	login	Autentica y obtiene token + id_usuario.	Hash SHA-256 en cliente.
alta_articulo()	alta_articulo	Envía datos y foto de nuevo artículo.	Valida campos numéricos.
consulta_articulos()	consulta_articulos	Obtiene artículos filtrados.	Renderiza dinámicamente.
incrementa()/decrementa()	—	Ajusta cantidad antes de compra.	No llaman al servidor.
comprar_articulo()	compra_articulo	Agrega al carrito y descuenta stock.	Maneja mensajes de error.
consulta_carrito()	consulta_carrito_compra	Lista ítems y total.	Reutiliza layout de resultados.
elimina_articulo_carrito()	elimina_articulo_carrito_compra	Borra ítem devolviendo stock.	Refresca carrito.
elimina_carrito()	elimina_carrito_compra	Limpia todo el carrito.	Confirmación previa.
modifica_carrito()	modifica_carrito_compra	Ajusta +1/-1 artículo en carrito.	Valida errores límite y stock.

limpia_captura_articulo()	—	Resetea campos y imagen al entrar.	Evita arrastrar foto previa.
----------------------------------	---	------------------------------------	------------------------------

Imagen 47. Fragmento JS mostrando comprar_articulo().

11 Endpoints REST

Esta sección describe cada endpoint del servicio REST publicado sobre la base URL:

`http:// 4.248.144.161:8080/Servicio/rest/ws`

Incluye: propósito, método HTTP, ruta, parámetros (query / body), ejemplo de petición con curl, respuestas exitosas, errores frecuentes y notas sobre transacciones, autenticación (token) y validaciones. La numeración de imágenes continúa la secuencia del reporte (todas las capturas deben ser completas con fecha y hora, según lineamientos).

11.1 Convenciones generales

- Autenticación: los métodos protegidos verifican token y email o id_usuario mediante las funciones verifica_acceso.
- Códigos HTTP:
 - 200 Éxito.
 - 400 Error de lógica / validación / acceso denegado.
- Formato de errores: { "message": "<Descripción del error>" }
- Base de rutas: todos los endpoints agregan su nombre al final de /Servicio/rest/ws.
- Parámetros:
 - Query string para login, token, ids e incremento.
 - Cuerpo JSON para datos de usuario, artículo o compra.
- Transacciones: Alta/modificación/eliminación que involucran varias tablas usan setAutoCommit(false) y commit/rollback.

Imagen 46: Captura del listado de métodos en el código fuente (scroll mostrando anotaciones @Path).

11.2 Endpoint: login

Aspecto	Detalle
Método	GET
Ruta	/login
Autenticado	No (emite token)
Query Params	email, password (hash SHA-256 ya generado en cliente)
Body	No aplica
Transacción	No

11.3 Endpoint: alta_usuario

Aspecto	Detalle
Método	POST
Ruta	/alta_usuario
Autenticado	No (registro)
Body JSON	Datos del usuario (email, password hash, nombre, apellidos, fecha_nacimiento ISO, teléfono, género, foto base64 opcional)
Transacción	Sí (inserta usuario y foto)

11.4 Endpoint: consulta_usuario

Aspecto	Detalle
Método	GET
Ruta	/consulta_usuario
Autenticado	Sí (email, token)

Query Params	email, token
Transacción	No (lectura)

11.5 Endpoint: modifica_usuario

Aspecto	Detalle
Método	PUT
Ruta	/modifica_usuario
Autenticado	Sí
Query Params	email, token
Body JSON	Campos modificables (password opcional no vacío para cambio)
Transacción	Sí (actualiza usuario y foto)

11.6 Endpoint: borra_usuario

Aspecto	Detalle
Método	DELETE
Ruta	/borra_usuario
Autenticado	Sí
Query Params	email, token
Transacción	Sí (borra fotos, luego usuario)

11.7 Endpoint: alta_articulo

Aspecto	Detalle
Método	POST
Ruta	/alta_articulo?id_usuario&token

Autenticado	Sí
Query Params	id_usuario, token
Body JSON	nombre, descripcion, precio, cantidad, foto (opcional)
Transacción	Sí (inserta stock y foto)

11.8 Endpoint: consulta_articulos

Aspecto	Detalle
Método	GET
Ruta	/consulta_articulos?palabra&id_usuario&token
Autenticado	Sí
Query Params	palabra, id_usuario, token
Body	No
Transacción	No (lectura)

11.9 Endpoint: compra_articulo

Aspecto	Detalle
Método	POST
Ruta	/compra_articulo?id_usuario&token
Autenticado	Sí
Query Params	id_usuario, token
Body JSON	id_articulo, cantidad
Transacción	Sí (carrito + stock)

11.10 Endpoint: elimina_articulo_carrito_compra

Aspecto	Detalle
Método	DELETE
Ruta	/elimina_articulo_carrito_compra?id_usuario&id_articulo&token
Autenticado	Sí
Query Params	id_usuario, id_articulo, token
Transacción	Sí (stock + borrado carrito)

11.11 Endpoint: elimina_carrito_compra

Aspecto	Detalle
Método	DELETE
Ruta	/elimina_carrito_compra?id_usuario&token
Autenticado	Sí
Query Params	id_usuario, token
Transacción	Sí (recorre artículos y revierte cantidades)

11.12 Endpoint: modifica_carrito_compra

Aspecto	Detalle
Método	PUT
Ruta	/modifica_carrito_compra?id_usuario&id_articulo&incremento&token
Autenticado	Sí
Query Params	id_usuario, id_articulo, incremento (1 o -1), token
Body	Vacio ({})

Transacción	Sí (stock y carrito)
--------------------	----------------------

11.13 Endpoint: consulta_carrito_compra

Aspecto	Detalle
Método	GET
Ruta	/consulta_carrito_compra?id_usuario&token
Autenticado	Sí
Query Params	id_usuario, token
Transacción	No (lectura)

12 Pruebas Unitarias del Back-End (curl)

En esta sección se documentan de forma sistemática las pruebas unitarias realizadas sobre cada endpoint REST del back-end. Se incluyen: preparación de datos, comandos curl, respuestas esperadas y ejemplos de errores controlados. Todas las capturas deben ser pantallas completas (sin recortes), con fecha y hora visibles (barra de tareas o terminal con marca temporal). La numeración de imágenes continúa desde la anterior sección (sección 11 terminó en Imagen 61); aquí partimos en la Imagen 62.

12.1 Metodología y Preparación

Las pruebas se ejecutaron directamente contra la VM T7-2022630278 en Azure, usando su IP pública (reemplazar 4.248.144.161 en ejemplos). Se emplearon comandos curl desde:

- Terminal Linux (SSH dentro de la VM) y/o
- Terminal local (Git Bash en Windows 11), asegurando conectividad a puerto 8080.

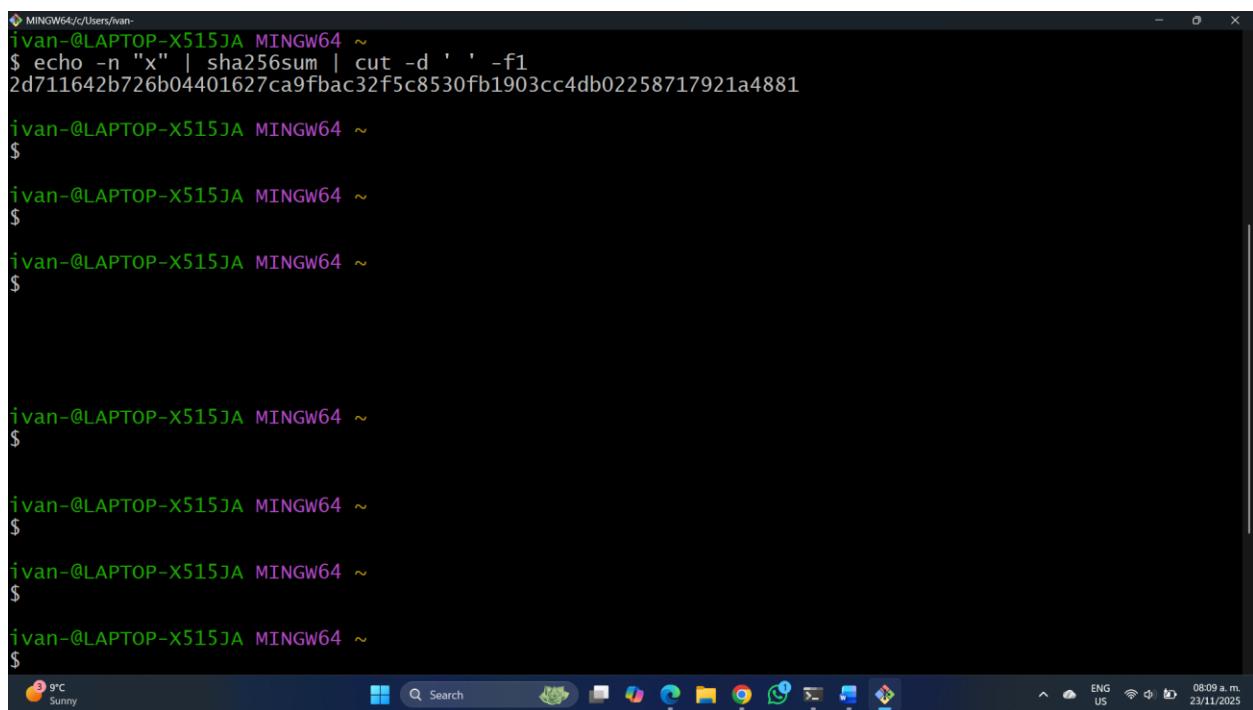
Pasos previos:

1. Verificar que Tomcat y el servicio Servicio.war estén desplegados y funcionando.
2. Confirmar que la base de datos servicio_web contenga las tablas: usuarios, fotos_usuarios, stock, fotos_articulos, carrito_compra.
3. Generar hash SHA-256 de la contraseña en el cliente (ejemplo "ClavePrueba2025"):
 - Uso rápido en Linux:
 - echo -n "x" | sha256sum | cut -d ' ' -f1
4. Utilizar ese hash en las peticiones de registro (alta_usuario) y login.

Consideraciones de captura:

- Mostrar comando ejecutado y respuesta completa JSON.

- Incluir evidencia de casos exitosos y fallidos (códigos 200 y 400).
- Mantener coherencia de email, id_usuario, token y id_articulo entre pruebas (documentar valores reales).
-



The screenshot shows a terminal window titled 'MINGW64/c/Users/ivan-' on a Windows 10 desktop. The terminal displays the following command and its output:

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ echo -n "x" | sha256sum | cut -d ' ' -f1
2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Below the terminal, the Windows taskbar is visible, showing icons for File Explorer, Edge, and other applications. The system tray indicates it's sunny with a temperature of 9°C. The system clock shows 08:09 a.m. on 23/11/2025.

Imagen 62: Terminal mostrando hash generado para la contraseña.

12.2 Evidencias por Endpoint

Para claridad, se usa el siguiente usuario de pruebas (ejemplo):

- Email: x@x.com
- Hash de password: 2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881
- Despues del login: id_usuario = 5 (ejemplo), token = <TOKEN_REAL>

Se describe cada endpoint (12.2.1 a 12.2.12) con:

- Preparación específica si aplica.
- Comandos curl (éxito y error).
- Respuesta esperada y observada.
- Imagen asociada (pantalla completa).

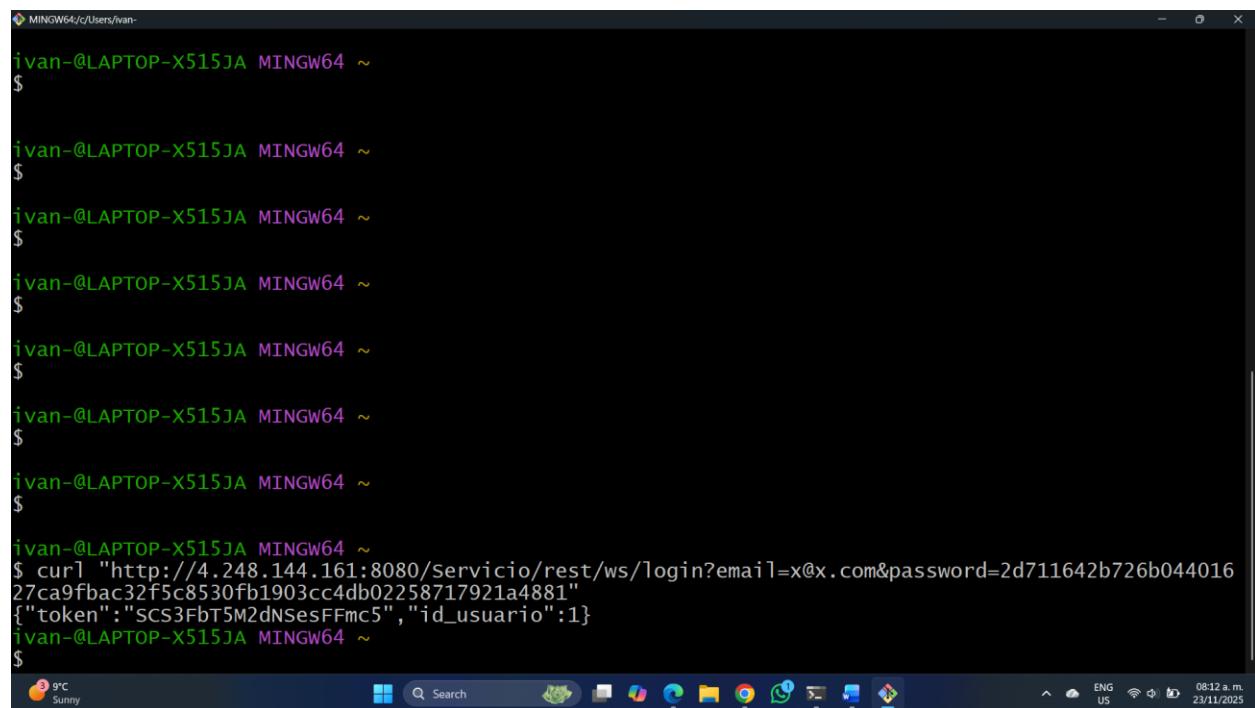
12.2.1 login

Comando (tras haber registrado el usuario):

```
curl
```

```
"http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881"
```

Respuesta esperada (código 200):



The screenshot shows a terminal window titled 'MINGW64/c/Users/ivan-' with a black background and white text. It displays a series of command-line inputs and outputs. The user has entered several blank lines followed by a curl command to log in. The response shows a JSON object containing a token and a user ID. The system tray at the bottom shows weather information ('Sunny'), network status, battery level, and the date and time ('08:12 a.m. 23/11/2025').

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ 

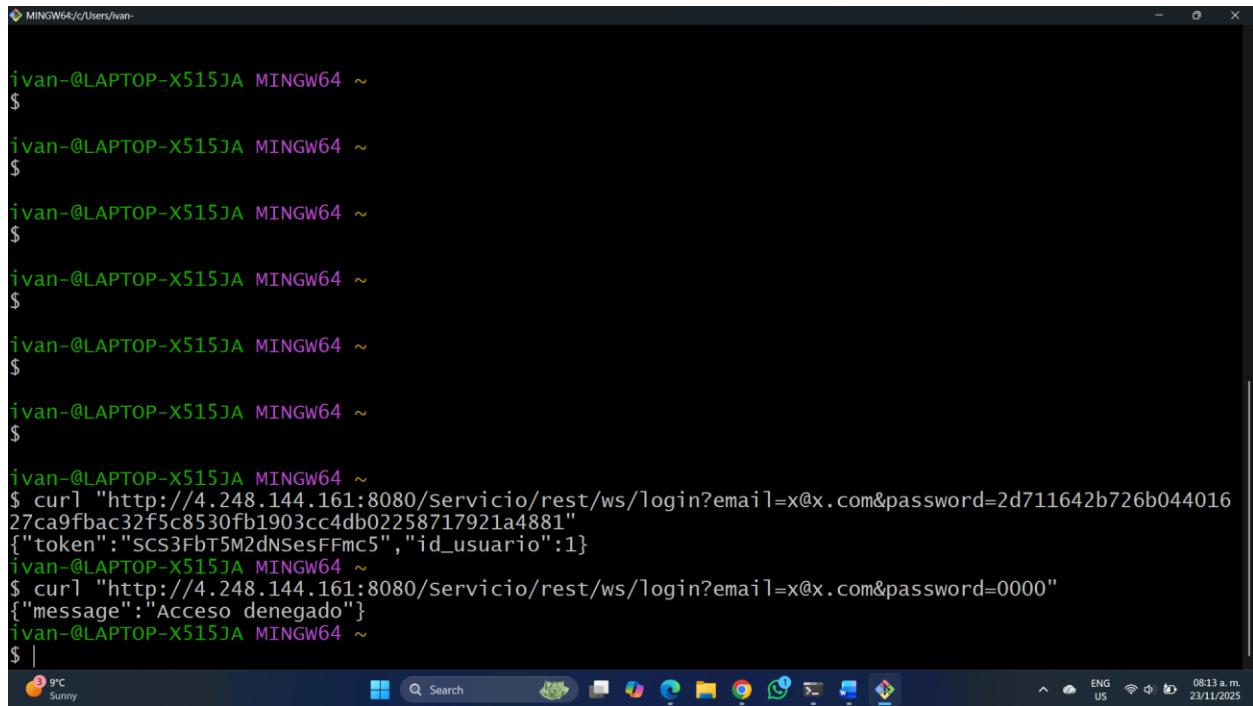
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/servicio/rest/ws/login?email=x@x.com&password=2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881"
{"token":"SCS3FbT5M2dNSesFFmc5","id_usuario":1}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Imagen 63: Login exitoso (con token e id_usuario).

Caso de error (password incorrecto):

```
curl
```

```
"http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=0000"
```



The screenshot shows a terminal window titled 'MINGW64' with the path 'c:/Users/ivan-'. The terminal output is as follows:

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881"
{"token":"SCS3FbT5M2dNseseFFmc5","id_usuario":1}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=0000"
{"message":"Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

The terminal shows two curl commands. The first command is successful, returning a token and user ID. The second command, with the wrong password, returns an 'Acceso denegado' (Access denied) message.

Imagen 64: Login erróneo (mensaje Acceso denegado).

12.2.2 alta_usuario

Comando (registro inicial):

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_usuario" \ -H "Content-Type: application/json" \ -d '{ "email":"x2@x2.com", "password":"x2", "nombre":"Test", "apellido_paterno":"Ejemplo", "apellido_materno":null, "fecha_nacimiento":"2024-05-01T12:00:00.000Z", "telefono":5512345678, "genero":"M", "foto":null }'
```

Respuesta esperada (200):

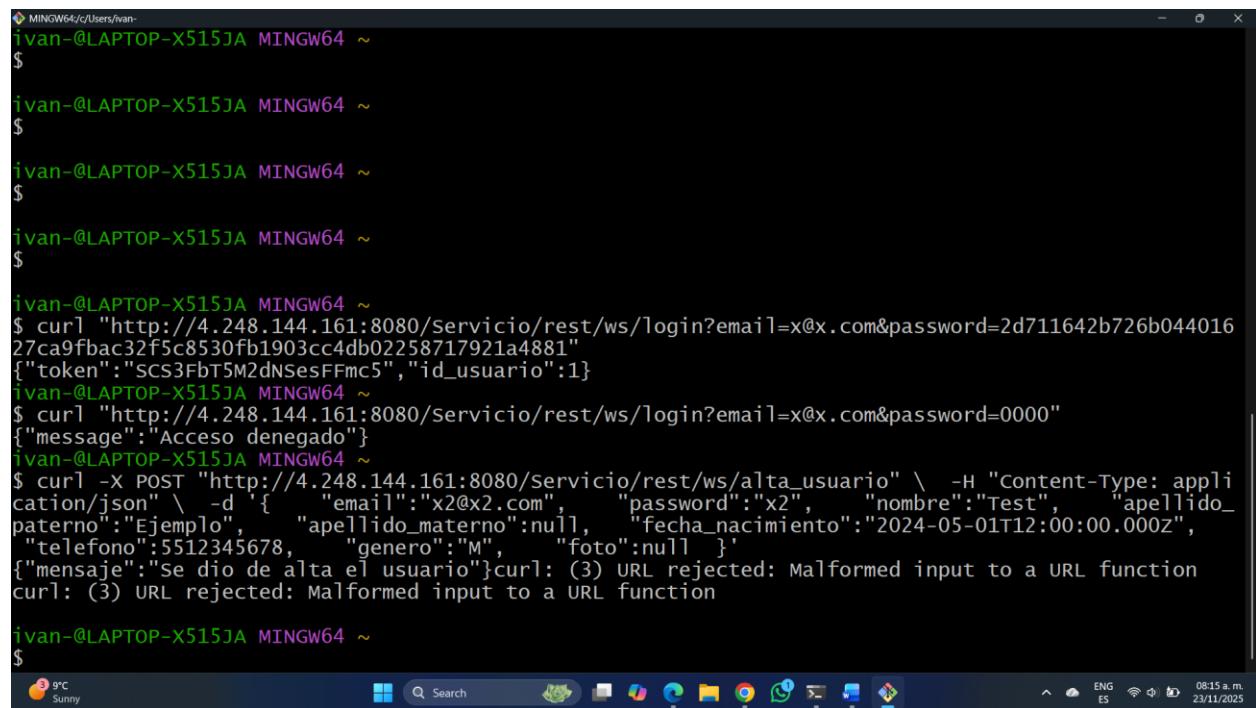
```
{ "mensaje": "Se dio de alta el usuario" }
```

Error (falta nombre):

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_usuario" \ -H "Content-Type: application/json" \ -d '{"email":"u@d.com","password":"<HASH>","nombre":"","apellido_paterno":"X","fecha_nacimiento":"2024-01-01T10:00:00.000Z"}'
```

Respuesta esperada (400):

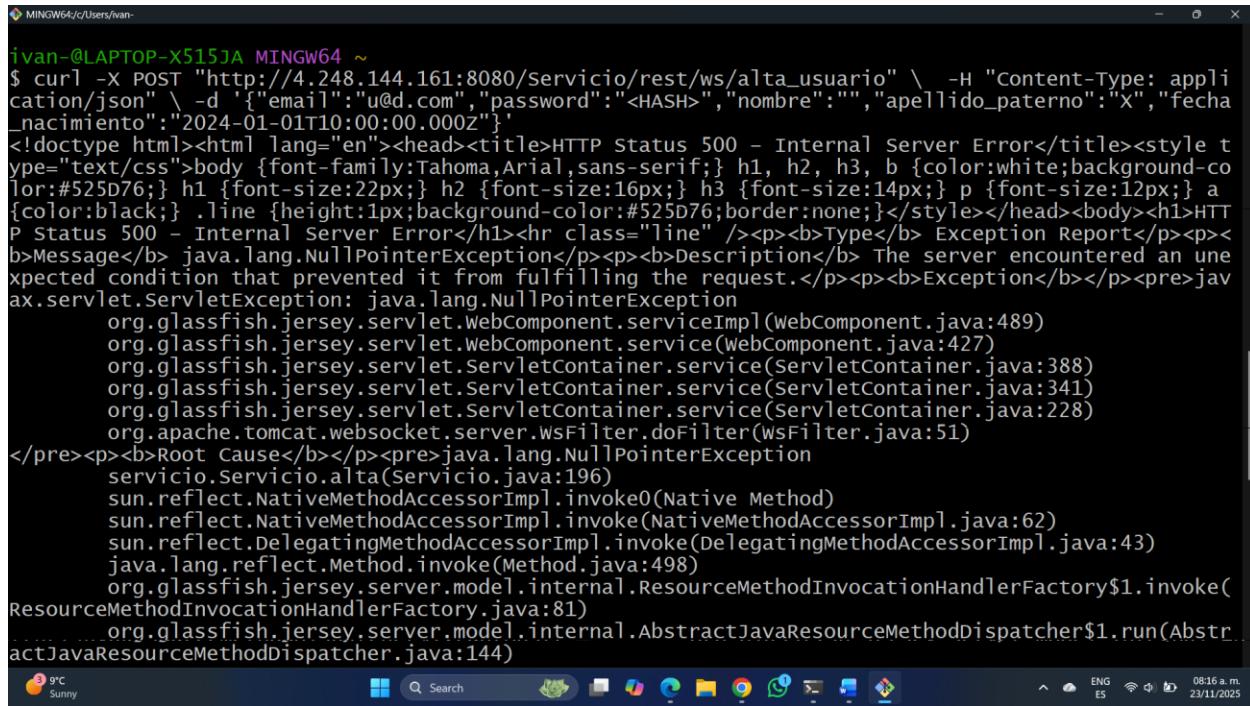
```
{ "message": "Se debe ingresar el nombre" }
```



The screenshot shows a terminal window titled 'MINGW64' with several command-line sessions. The first session shows an empty terminal. The second session starts with a curl command to log in with an empty email and password, resulting in a JSON response with a token and user ID. The third session attempts to log in with an empty email and password, receiving a 'denegado' message. The fourth session shows a curl command to create a new user ('alta_usuario') with an empty name, which fails due to malformed input. The fifth session shows the user creation command again, this time with a valid name ('Test'), which succeeds. The terminal window has a dark background with light-colored text. The bottom of the screen shows a Windows taskbar with various icons and system status indicators.

```
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881"  
{"token":"SCS3FbT5M2dNsEsFFmc5","id_usuario":1}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/login?email=x@x.com&password=0000"  
{"message": "Acceso denegado"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_usuario" \ -H "Content-Type: application/json" \ -d '{ "email": "x2@x2.com", "password": "x2", "nombre": "Test", "apellido_paterno": "Ejemplo", "apellido_materno": null, "fecha_nacimiento": "2024-05-01T12:00:00.000Z", "telefono": 5512345678, "genero": "M", "foto": null }'  
{"mensaje": "Se dio de alta el usuario"}curl: (3) URL rejected: Malformed input to a URL function  
curl: (3) URL rejected: Malformed input to a URL function  
ivan-@LAPTOP-X515JA MINGW64 ~  
$
```

Imagen 65: Alta exitosa.



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_usuario" \ -H "Content-Type: application/json" \ -d '{"email":"u@d.com","password":"<HASH>","nombre":"","apellido_paterno":"X","fecha_nacimiento":"2024-01-01T10:00:00.000Z"}'
<!DOCTYPE html><html lang="en"><head><title>HTTP Status 500 – Internal Server Error</title><style type="text/css">body {font-family:Tahoma,Arial,sans-serif;} h1, h2, h3, b {color:white;background-color:#525D76;} h1 {font-size:22px;} h2 {font-size:16px;} h3 {font-size:14px;} p {font-size:12px;} a {color:black;} .line {height:1px;background-color:#525D76;border:none;}</style></head><body><h1>HTTP Status 500 – Internal Server Error</h1><hr class="line" /><p><b>Type</b> Exception Report</p><p><b>Message</b> java.lang.NullPointerException</p><p><b>Description</b> The server encountered an unexpected condition that prevented it from fulfilling the request.</p><p><b>Exception</b></p><pre>javax.servlet.ServletException: java.lang.NullPointerException
    org.glassfish.jersey.servlet.WebComponent.serviceImpl(WebComponent.java:489)
    org.glassfish.jersey.servlet.WebComponent.service(WebComponent.java:427)
    org.glassfish.jersey.servlet.ServletContainer.service(ServletContainer.java:388)
    org.glassfish.jersey.servlet.ServletContainer.service(ServletContainer.java:341)
    org.glassfish.jersey.servlet.ServletContainer.service(ServletContainer.java:228)
    org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:51)
</pre><p><b>Root Cause</b></p><pre>java.lang.NullPointerException
    servicio.Servicio.alta(Servicio.java:196)
    sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    java.lang.reflect.Method.invoke(Method.java:498)
    org.glassfish.jersey.server.model.internal.ResourceMethodInvocationHandlerFactory$1.invoke(ResourceMethodInvocationHandlerFactory.java:81)
    org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher$1.run(AbstractJavaResourceMethodDispatcher.java:144)
</pre>
```

Imagen 66: Error por campo obligatorio faltante.

12.2.3 consulta_usuario

Con token obtenido del login:

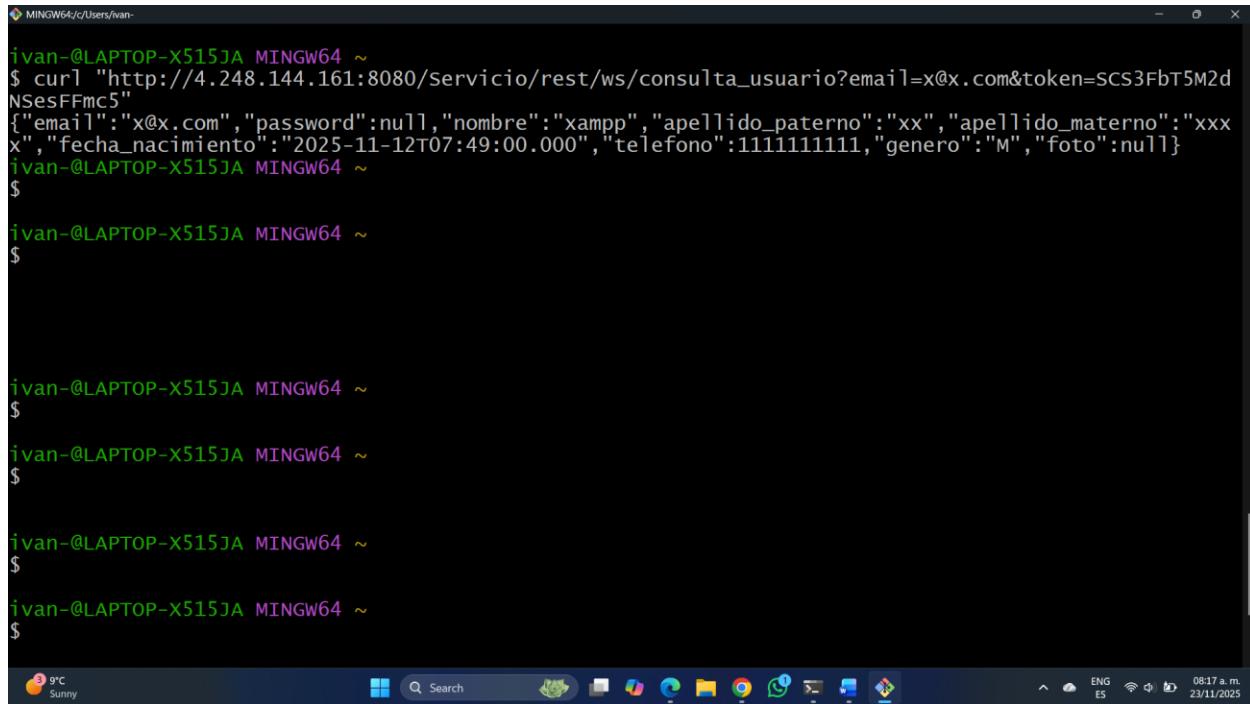
curl

```
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_usuario?email=x@x.com&token=S
CS3FbT5M2dNSesFFmc5"
```

Respuesta esperada (200): Datos del usuario y foto (null o base64). Error (token inválido):

curl

```
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_usuario?email=x@x.com&token=Z
ZZZ"
```

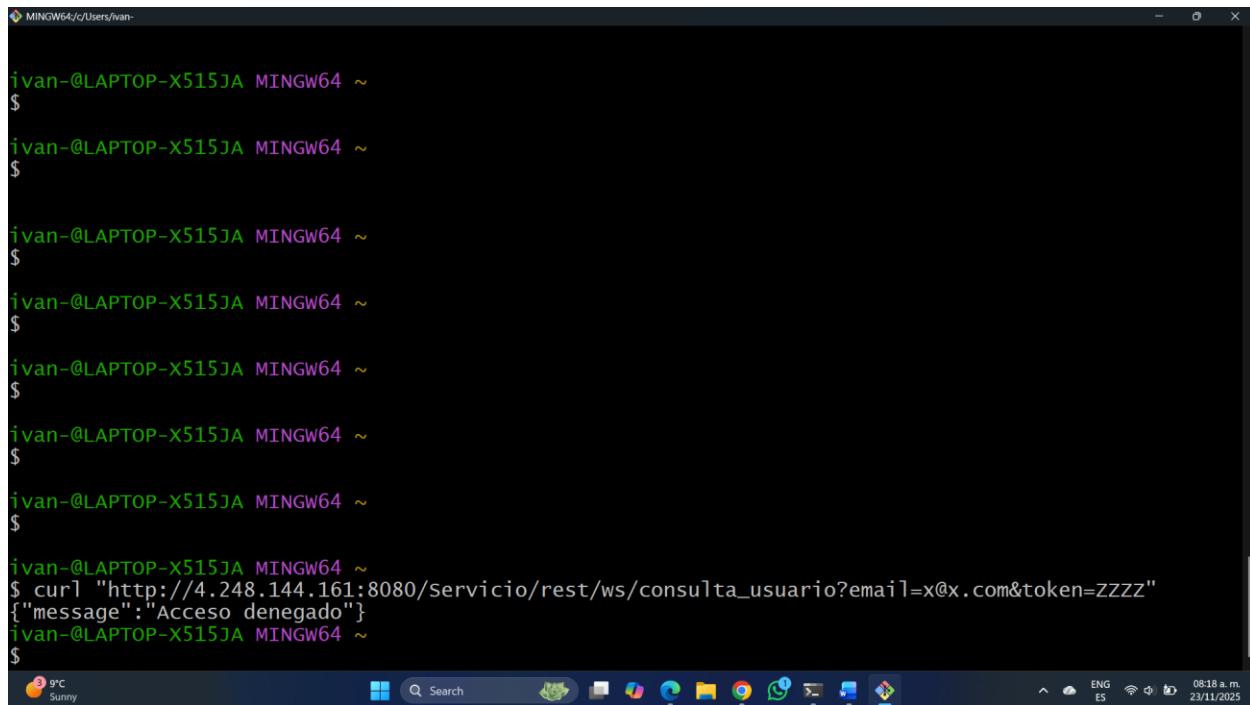


```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_usuario?email=x@x.com&token=SCS3FbT5M2dNSesFFmc5"
[{"email":"x@x.com","password":null,"nombre":"xampp","apellido_paterno":"xx","apellido_materno":"xxx","fecha_nacimiento":"2025-11-12T07:49:00.000","telefono":1111111111,"genero":"M","foto":null}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```



```
9°C Sunny 08:17 a.m. 23/11/2025
```

Imagen 67: Consulta exitosa.



```
ivan-@LAPTOP-X515JA MINGW64 ~
$
```



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_usuario?email=x@x.com&token=ZZZZ"
{"message":"Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```



```
9°C Sunny 08:18 a.m. 23/11/2025
```

Imagen 68: Error acceso denegado (token incorrecto).

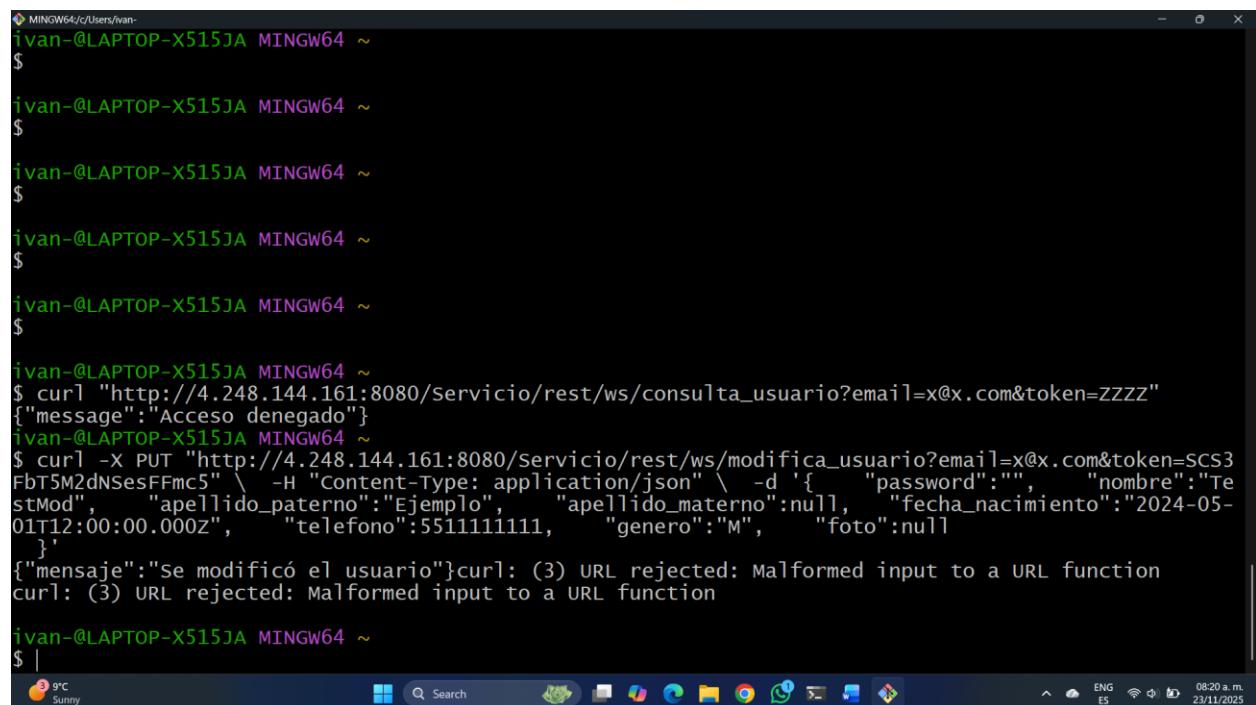
12.2.4 modifica_usuario

Cambiar nombre y teléfono (mantener token vigente):

```
curl -X PUT "http://4.248.144.161:8080/Servicio/rest/ws/modifica_usuario?email=x@x.com&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{ "password": "", "nombre": "TestMod", "apellido_paterno": "Ejemplo", "apellido_materno": null, "fecha_nacimiento": "2024-05-01T12:00:00.000Z", "telefono": "551111111111, "genero": "M", "foto": null }'
```

Respuesta esperada (200):

```
{ "mensaje": "Se modificó el usuario" }
```



The screenshot shows a terminal window titled 'MINGW64' with a black background and white text. The user 'ivan' is logged in at 'LAPTOP-X515JA'. The terminal shows several blank lines followed by a curl command to update a user's information. The command includes the URL, method (PUT), headers (Content-Type: application/json), and a JSON payload with fields like password, nombre, apellido_paterno, apellido_materno, fecha_nacimiento, telefono, genero, and foto. The response from the server is shown as an object with a single key 'mensaje' containing the value 'Se modificó el usuario'. Below the terminal, the Windows taskbar is visible with various icons and system status indicators.

```
MINGW64:/c/Users/ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/modifica_usuario?email=x@x.com&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{ "password": "", "nombre": "TestMod", "apellido_paterno": "Ejemplo", "apellido_materno": null, "fecha_nacimiento": "2024-05-01T12:00:00.000Z", "telefono": "551111111111, "genero": "M", "foto": null }'
{"mensaje": "Se modificó el usuario"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Imagen 69: Modificación exitosa.

12.2.5 borra_usuario (opcional, si se prueba al final)

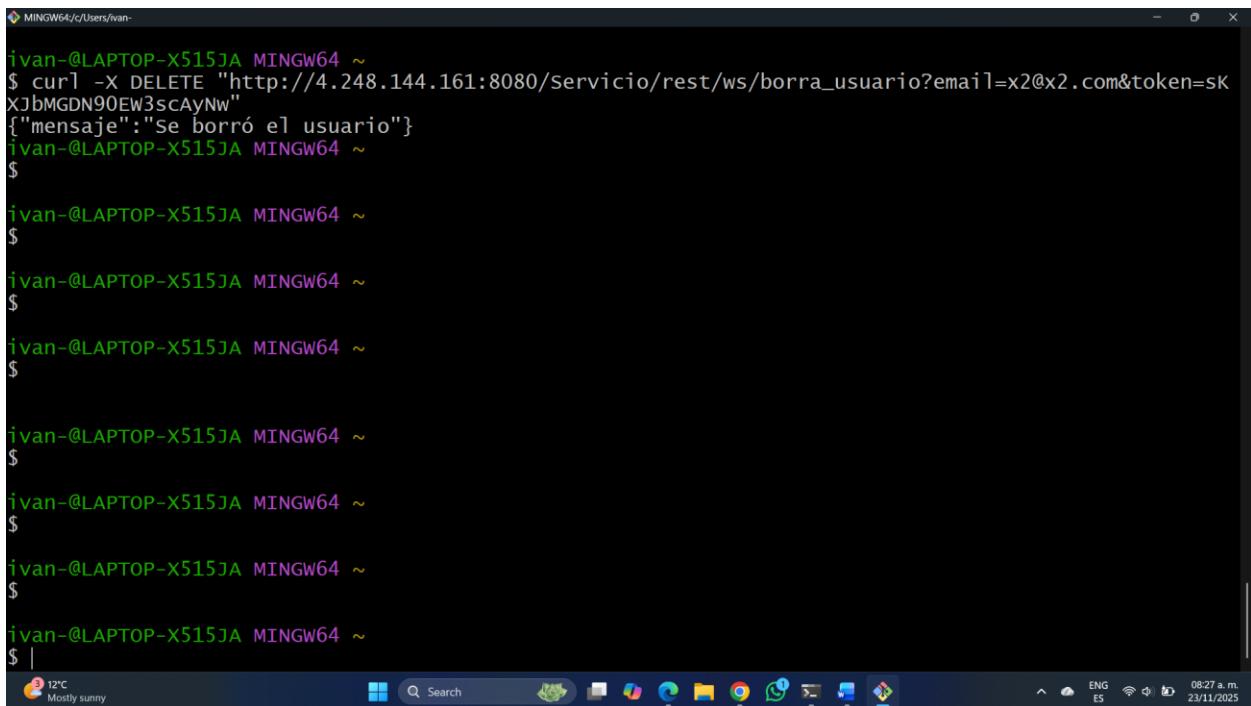
```
curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=x2@x2.com&token=sKXJbMGDN90EW3scAyNw"
```

Respuesta esperada (200):

```
{ "mensaje": "Se borró el usuario" }
```

Error (email no existente):

```
curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5"
```



The screenshot shows a terminal window titled 'MINGW64/c/Users/ivan-' running on a Windows operating system. The user has run a curl command to delete a user from a service. The response is a JSON object indicating the message 'Se borró el usuario' (The user was deleted). The terminal window is dark-themed, and the command history shows several previous empty lines.

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=x2@x2.com&token=sKxJbMGDN90EW3scAyNw"
{"mensaje": "Se borró el usuario"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

ivan-@LAPTOP-X515JA MINGW64 ~
\$
ivan-@LAPTOP-X515JA MINGW64 ~
\$ |

Imagen 71: Borrado exitoso.

```
MINGW64/c/Users/ivan-  
{"mensaje":"se borró el usuario"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5"  
{"message":"Acceso denegado"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$
```

Imagen 72: Error email inexistente.

12.2.6 alta articulo

Registrar artículo:

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{  "nombre":"Mayonesa 500g",  "descripcion":"Envase familiar",  "precio":35.50,  "cantidad":100,  "foto":null}'
```

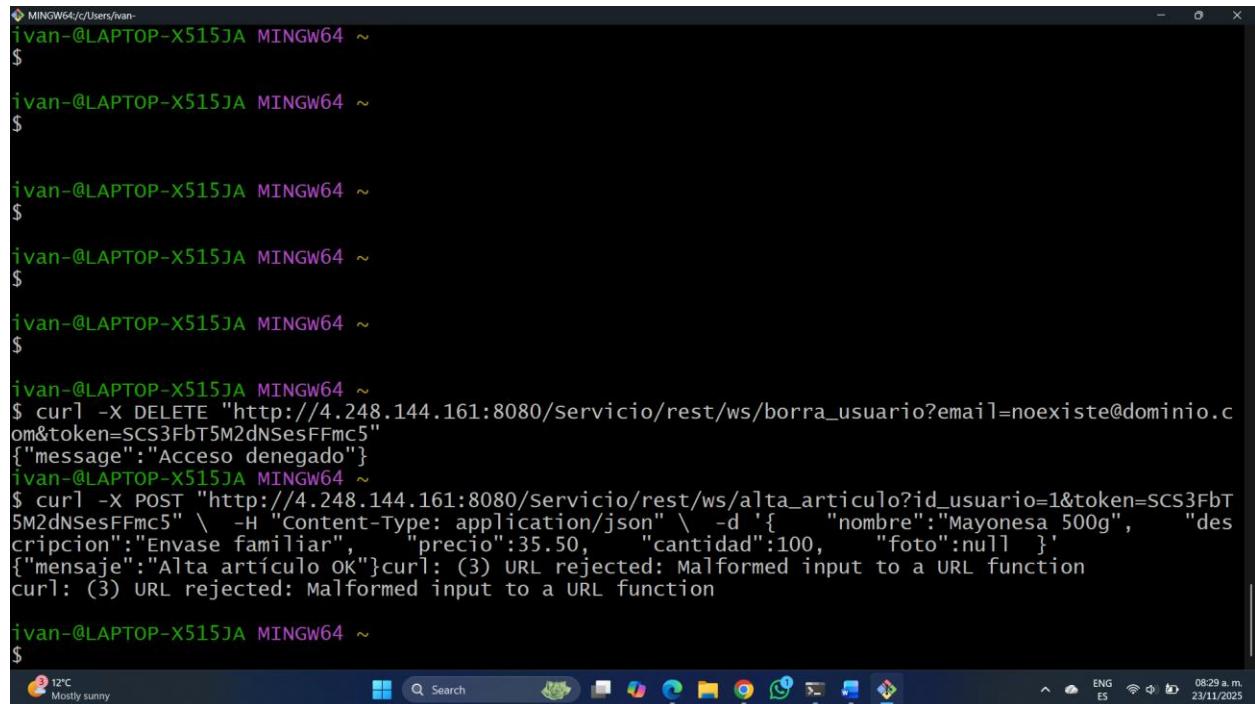
Respuesta esperada (200):

```
{ "mensaje": "Alta artículo OK" }
```

Error (precio inválido):

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3Fb
```

```
T5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d  
'{"nombre":"X","descripcion":"Y","precio":0,"cantidad":10,"foto":null}'
```

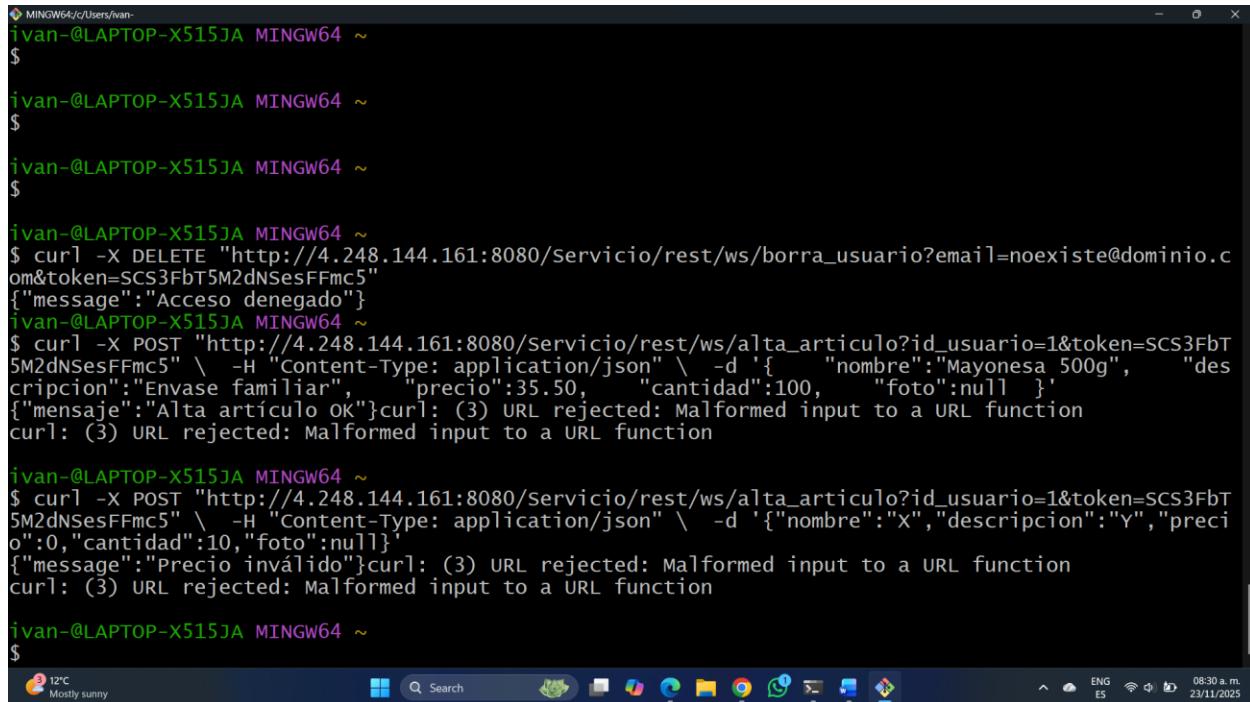


The screenshot shows a terminal window titled 'MINGW64' with a black background and white text. It displays several command-line entries:

- \$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5" {"message": "Acceso denegado"}
- \$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{ "nombre": "Mayonesa 500g", "descripcion": "Envase familiar", "precio": 35.50, "cantidad": 100, "foto": null }' {"mensaje": "Alta artículo OK"}curl: (3) URL rejected: Malformed input to a URL function
- curl: (3) URL rejected: Malformed input to a URL function
- \$

The terminal window has a dark blue header bar with the title 'MINGW64'. At the bottom, there's a taskbar with various icons, including a weather widget showing '12°C Mostly sunny', a search bar, and system status indicators like battery level, signal strength, and date/time ('08:29 a.m. 23/11/2025').

Imagen 73: Alta artículo exitosa.



```
MINGW64/c/Users/ivan-
ivan-@LAPTOP-X515JA MINGW64 ~
$ 

ivan-@LAPTOP-X515JA MINGW64 ~
$ 

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5"
{"message": "Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "Mayonesa 500g",
    "descripcion": "Envase familiar",
    "precio": 35.50,
    "cantidad": 100,
    "foto": null
}'
{"mensaje": "Alta artículo OK"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "X",
    "descripcion": "Y",
    "precio": 0,
    "cantidad": 10,
    "foto": null
}'
{"message": "Precio inválido"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Imagen 74: Error precio inválido.

12.2.7 consulta_articulos

Búsqueda por palabra:

curl

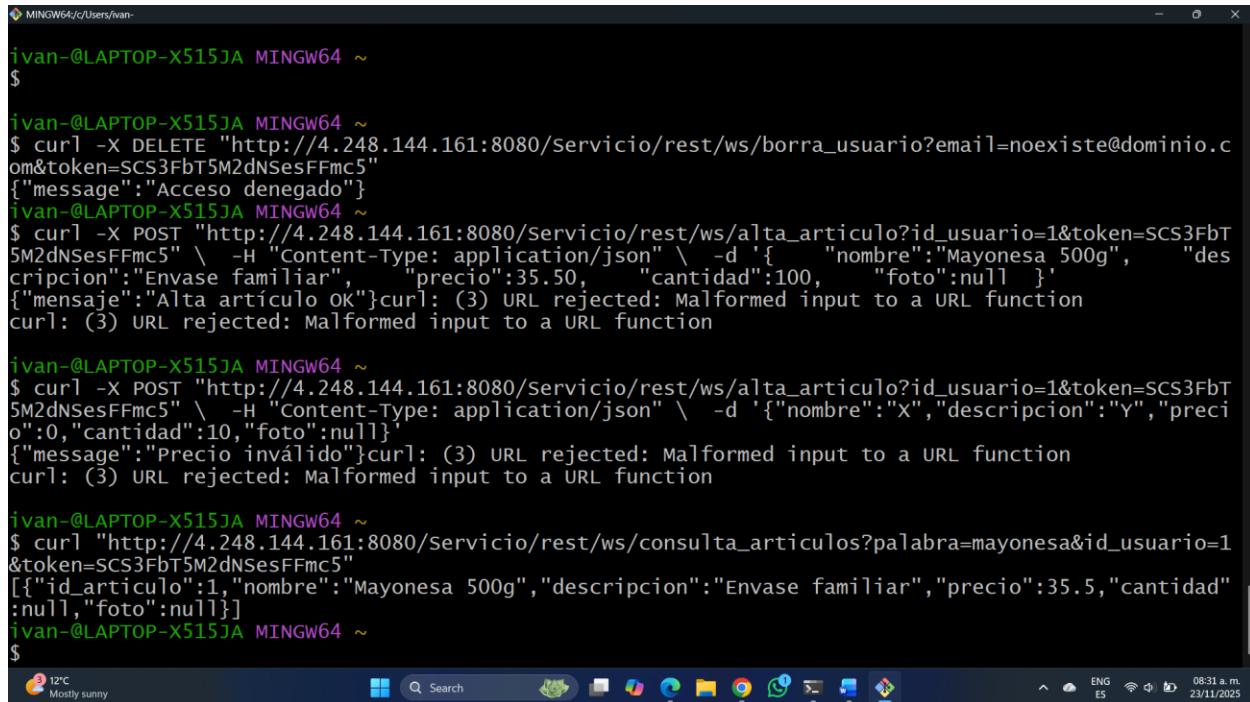
```
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_articulos?palabra=mayonesa&id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"
```

Respuesta esperada (200): Arreglo con artículos que hacen match.

Error (token inválido):

curl

```
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_articulos?palabra=mayonesa&id_usuario=5&token=ZZZZ"
```

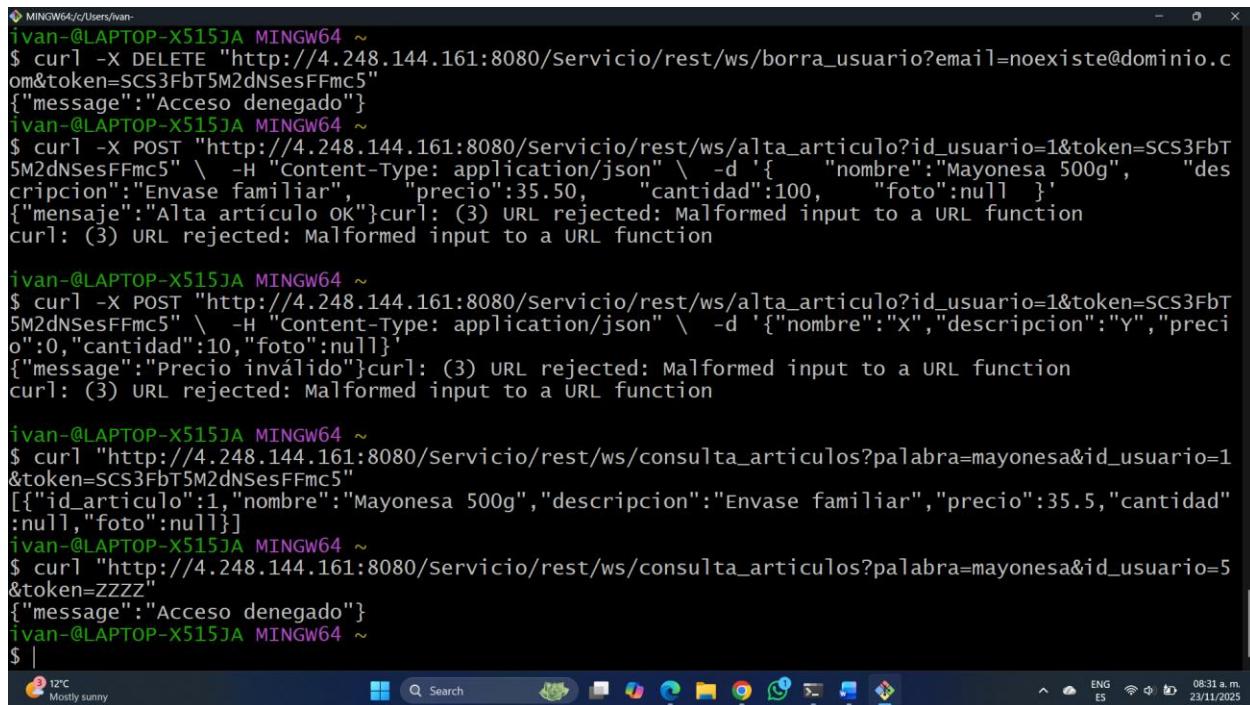


```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5"
{"message":"Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "Mayonesa 500g",
    "descripcion": "Envase familiar",
    "precio": 35.5,
    "cantidad": 100,
    "foto": null
}'
{"mensaje": "Alta artículo OK"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "X",
    "descripcion": "Y",
    "precio": 0,
    "cantidad": 10,
    "foto": null
}'
{"message": "Precio inválido"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_articulos?palabra=mayonesa&id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"
[{"id_articulo": 1, "nombre": "Mayonesa 500g", "descripcion": "Envase familiar", "precio": 35.5, "cantidad": null, "foto": null}]
ivan-@LAPTOP-X515JA MINGW64 ~
```

Imagen 75: Lista con artículo encontrado.



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/borra_usuario?email=noexiste@dominio.com&token=SCS3FbT5M2dNSesFFmc5"
{"message":"Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "Mayonesa 500g",
    "descripcion": "Envase familiar",
    "precio": 35.5,
    "cantidad": 100,
    "foto": null
}'
{"mensaje": "Alta artículo OK"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/alta_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \
-H "Content-Type: application/json" \
-d '{
    "nombre": "X",
    "descripcion": "Y",
    "precio": 0,
    "cantidad": 10,
    "foto": null
}'
{"message": "Precio inválido"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_articulos?palabra=mayonesa&id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"
[{"id_articulo": 1, "nombre": "Mayonesa 500g", "descripcion": "Envase familiar", "precio": 35.5, "cantidad": null, "foto": null}]
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_articulos?palabra=mayonesa&id_usuario=5&token=zzzz"
{"message": "Acceso denegado"}
ivan-@LAPTOP-X515JA MINGW64 ~
```

Imagen 76: Error acceso denegado.

12.2.8 compra_articulo

Comprar 3 unidades (asumiendo id_articulo=1):

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":3}'
```

Respuesta esperada (200):

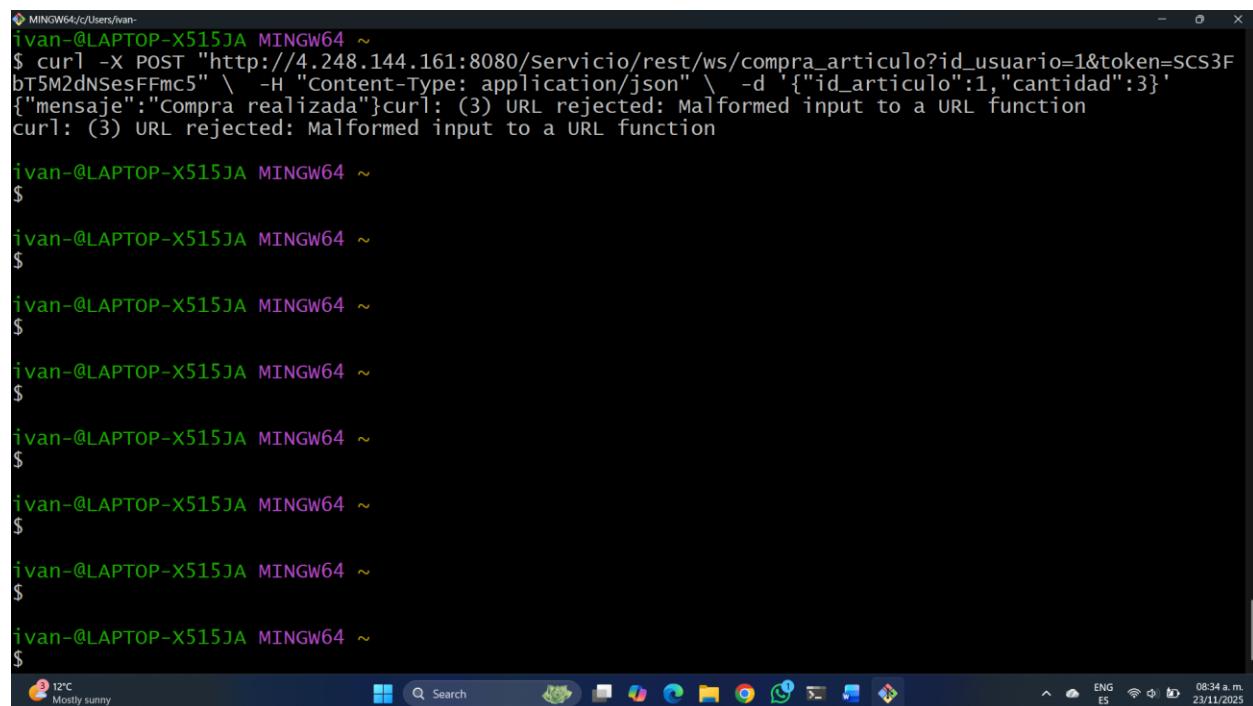
```
{ "mensaje":"Compra realizada" }
```

Error (cantidad mayor al stock remanente):

```
curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}'
```

Respuesta (400):

```
{ "message":"No hay suficientes artículos en stock" }
```



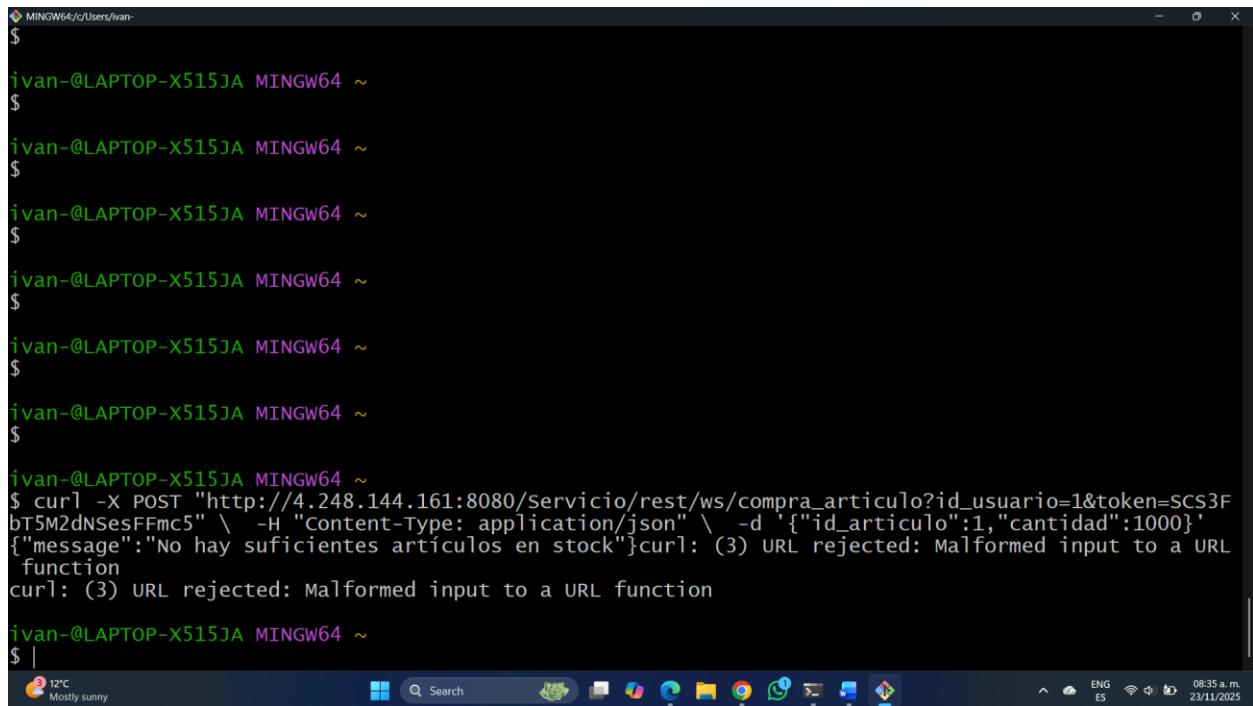
The screenshot shows a terminal window titled 'MINGW64/c/Users/ivan-' with the command 'ivan-@LAPTOP-X515JA MINGW64 ~'. The user runs a curl POST request to purchase 3 units of item ID 1. The response is successful, returning the message 'Compra realizada'. The user then attempts to purchase 1000 units, which fails due to insufficient stock, resulting in a 400 error message: 'No hay suficientes artículos en stock'. The terminal also displays several blank lines and the system tray at the bottom.

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":3}'
{"mensaje":"Compra realizada"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ 

ivan-@LAPTOP-X515JA MINGW64 ~
$ 
```

Imagen 77: Compra exitosa.



A screenshot of a Windows terminal window titled "MINGW64:/c/Users/ivan-". The terminal shows several command-line entries from the user "ivan" on a laptop named "X515JA". The user runs a curl command to make a POST request to "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo" with parameters "id_usuario=1&token=SCS3FbT5M2dNSesFFmc5", "id_articulo=1", and "cantidad=1000". The response indicates success with the message "Compra exitosa".

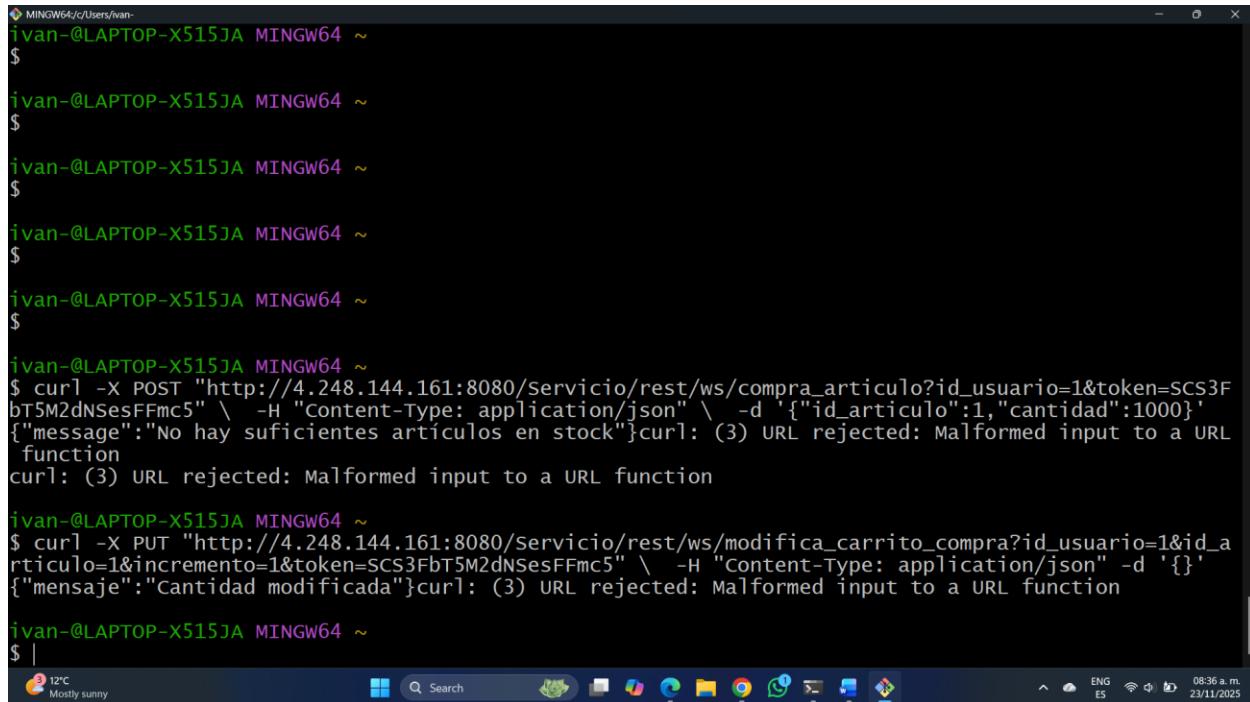
```
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}' {"message":"Compra exitosa"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Imagen 78: Error stock insuficiente.

- **12.2.9 modifica_carrito_compra**

Incrementar 1 unidad:

```
curl -X PUT "http://4.248.144.161:8080/Servicio/rest/ws/modifica_carrito_compra?id_usuario=1&id_articulo=1&incremento=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" -d '{}'
```



```
MINGW64/c/Users/ivan-
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}' {"message":"No hay suficientes artículos en stock"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X PUT "http://4.248.144.161:8080/servicio/rest/ws/modifica_carrito_compra?id_usuario=1&id_articulo=1&incremento=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" -d '{}'
 {"mensaje":"Cantidad modificada"}curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Imagen 79: Incremento exitoso.

12.2.9 consulta_carrito_compra

curl

```
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"
```

Respuesta esperada (200) ejemplo:

```
{
```

```
    "total": 142,
```

```
    "items": [
```

```
        {
```

```
            "id_articulo": 1,
```

```
            "nombre": "Mayonesa 500g",
```

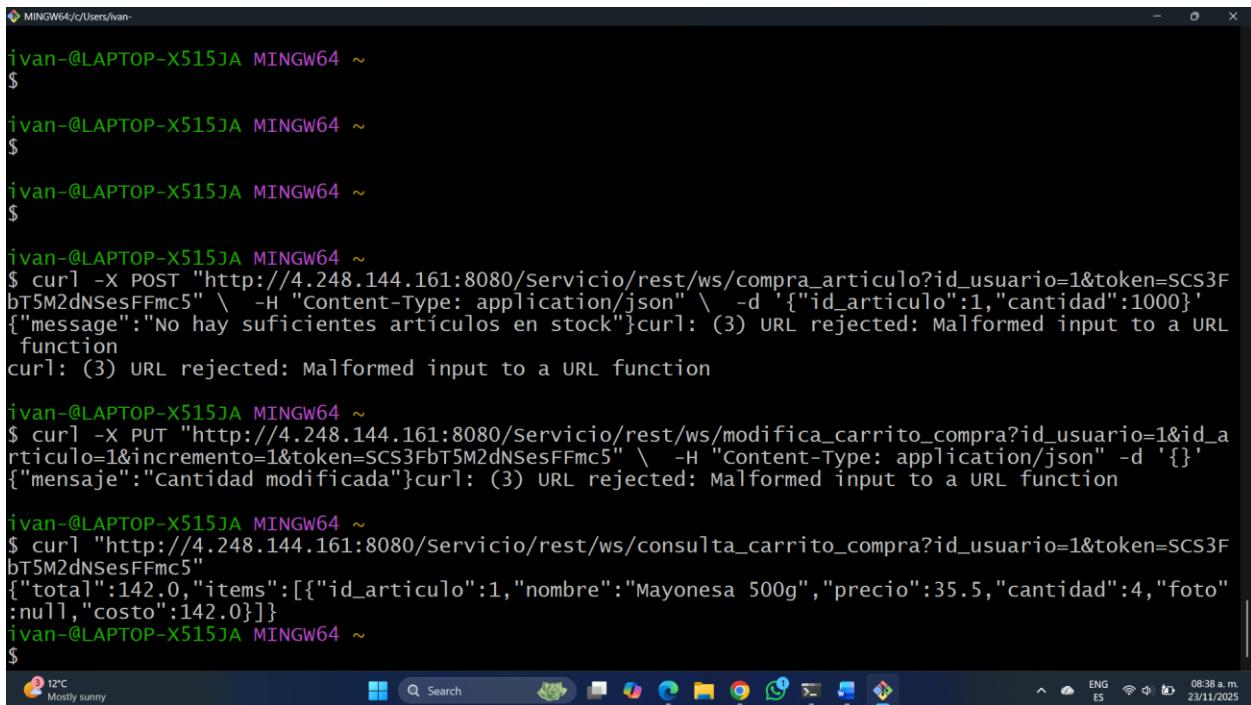
```
            "precio": 35.5,
```

```

    "cantidad": 4,
    "foto": null,
    "costo": 142
  }
]

}

```



The screenshot shows a terminal window titled 'MINGW64/c/Users/ivan-' with the following session:

```

ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/Servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}' {"message":"No hay suficientes artículos en stock"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X PUT "http://4.248.144.161:8080/Servicio/rest/ws/modifica_carrito_compra?id_usuario=1&id_articulo=1&incremento=1&token=SCS3FbT5M2dNSesFFmc5" \ -H "Content-Type: application/json" -d '{}' {"mensaje":"Cantidad modificada"}curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"
{"total":142.0,"items":[{"id_articulo":1,"nombre":"Mayonesa 500g","precio":35.5,"cantidad":4,"foto":null,"costo":142.0}]}
ivan-@LAPTOP-X515JA MINGW64 ~
$ 

```

The terminal shows three curl commands. The first two result in errors due to malformed URLs. The third command successfully retrieves the total and items from the shopping cart.

Imagen 81: Carrito con total y items.

12.2.10 elimina_articulo_carrito_compra

```

curl           -X           DELETE
"http://4.248.144.161:8080/Servicio/rest/ws/elimina_articulo_carrito_compra?id_usuario
=1&id_articulo=1&token=SCS3FbT5M2dNSesFFmc5"

```

Respuesta esperada (200):

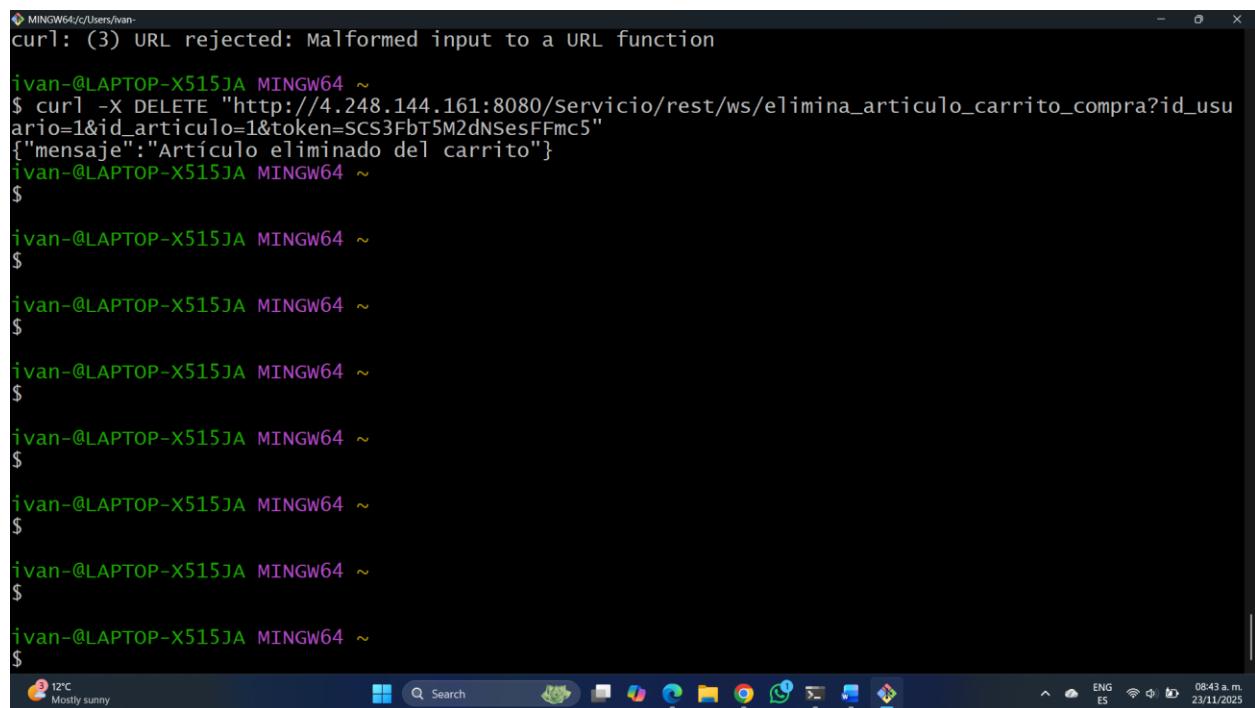
{ "mensaje": "Artículo eliminado del carrito" }

Error (artículo ya no está en el carrito):

```
curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/elimina_articulo_carrito_compra?id_usuario=5&id_articulo=1&token=SCS3FbT5M2dNSesFFmc5"
```

Respuesta (400):

{ "message": "Artículo no está en el carrito" }



```
MINGW64:/c/Users/ivan-ivan@LAPTOP-X515JA MINGW64 ~$ curl -X DELETE "http://4.248.144.161:8080/Servicio/rest/ws/elimina_articulo_carrito_compra?id_usuario=5&id_articulo=1&token=SCS3FbT5M2dNSesFFmc5" {"mensaje": "Artículo eliminado del carrito"} ivan@LAPTOP-X515JA MINGW64 ~$ ivan@LAPTOP-X515JA MINGW64 ~$
```

Imagen 83: Eliminación exitosa.

```

MINGW64/c/Users/ivan-
bt5M2dNSesFFmc5"
{"total":0.0,"items":[]}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/servicio/rest/ws/elimina_articulo_carrito_compra?id_usuario=1&id_articulo=1&token=SCS3FbT5M2dNSesFFmc5"
{"message":"Articulo no está en el carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ 

```

Imagen 84: Error artículo inexistente en carrito.

12.2.11 elimina_carrito_compra

```

curl           -X          DELETE
                "http://4.248.144.161:8080/Servicio/rest/ws/elimina_carrito_compra?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"

```

Respuesta esperada (200):

```
{ "mensaje":"Carrito eliminado" }
```

Consulta posterior (debería devolver total 0 e items vacíos o carrito sin elementos):

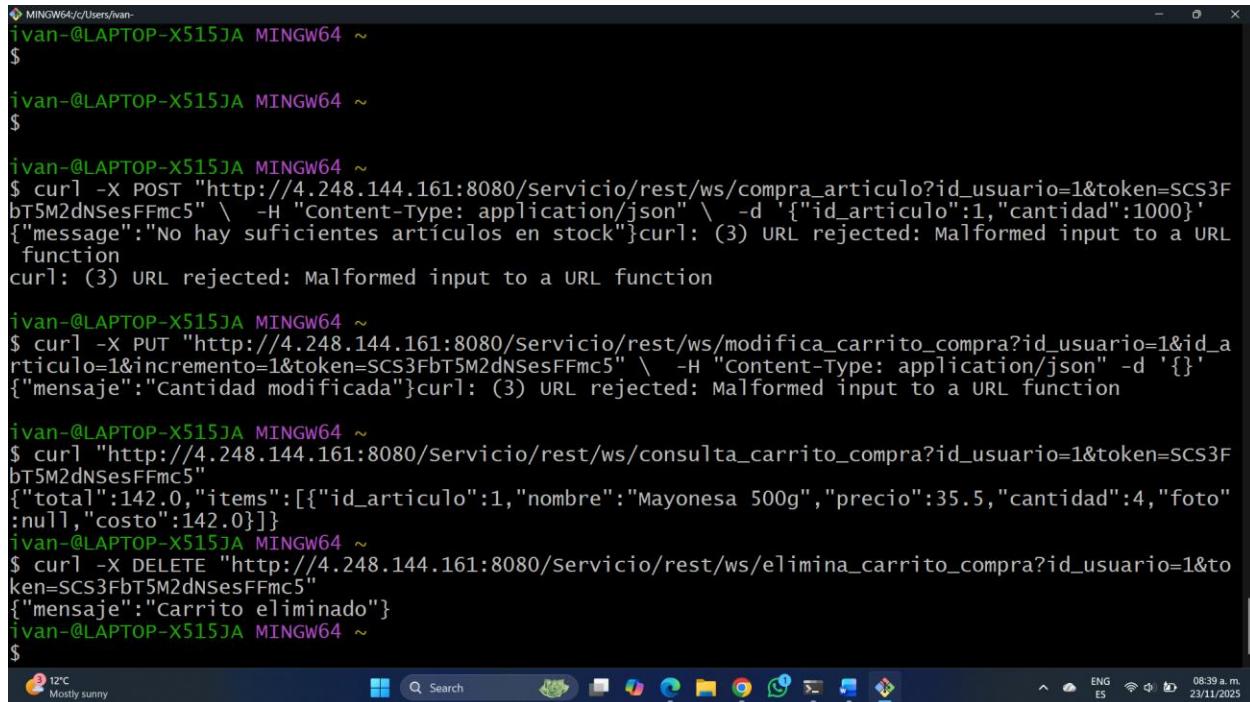
```

curl
"http://4.248.144.161:8080/Servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3FbT5M2dNSesFFmc5"

```

Respuesta ejemplo:

```
{ "total": 0, "items": [] }
```

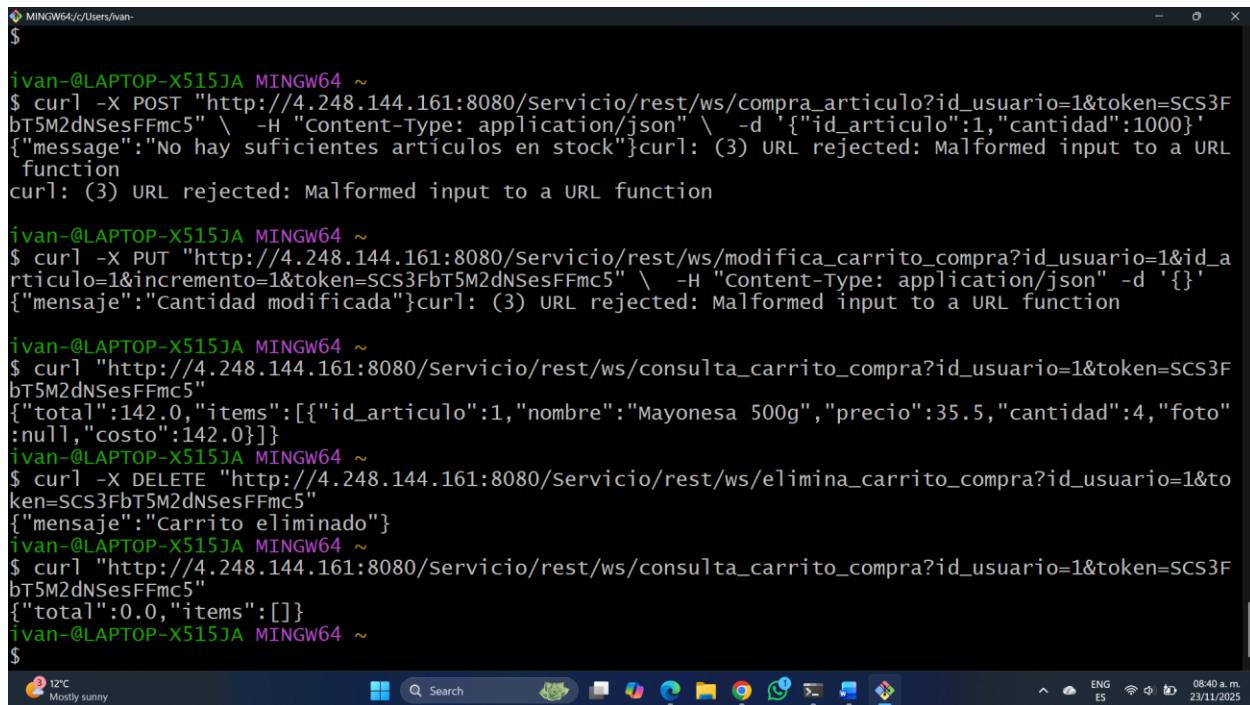


```
MINGW64/c/Users/ivan-
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X POST "http://4.248.144.161:8080/servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}' {"message":"No hay suficientes artículos en stock"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X PUT "http://4.248.144.161:8080/servicio/rest/ws/modifica_carrito_compra?id_usuario=1&id_articulo=1&incremento=1&token=SCS3Fbt5M2dNSesFFmc5" \ -H "Content-Type: application/json" -d '{}' {"mensaje":"Cantidad modificada"}curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5"
{"total":142.0,"items":[{"id_articulo":1,"nombre":"Mayonesa 500g","precio":35.5,"cantidad":4,"foto":null,"costo":142.0}]}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/servicio/rest/ws/elimina_carrito_compra?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5"
{"mensaje":"Carrito eliminado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Imagen 85: Vaciar carrito exitoso.



```
MINGW64/c/Users/ivan-
$ curl -X POST "http://4.248.144.161:8080/servicio/rest/ws/compra_articulo?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5" \ -H "Content-Type: application/json" \ -d '{"id_articulo":1,"cantidad":1000}' {"message":"No hay suficientes artículos en stock"}curl: (3) URL rejected: Malformed input to a URL function
curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X PUT "http://4.248.144.161:8080/servicio/rest/ws/modifica_carrito_compra?id_usuario=1&id_articulo=1&incremento=1&token=SCS3Fbt5M2dNSesFFmc5" \ -H "Content-Type: application/json" -d '{}' {"mensaje":"Cantidad modificada"}curl: (3) URL rejected: Malformed input to a URL function

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/Servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5"
{"total":142.0,"items":[{"id_articulo":1,"nombre":"Mayonesa 500g","precio":35.5,"cantidad":4,"foto":null,"costo":142.0}]}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -X DELETE "http://4.248.144.161:8080/servicio/rest/ws/elimina_carrito_compra?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5"
{"mensaje":"Carrito eliminado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl "http://4.248.144.161:8080/servicio/rest/ws/consulta_carrito_compra?id_usuario=1&token=SCS3Fbt5M2dNSesFFmc5"
{"total":0.0,"items":[]}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Imagen 86: Carrito consultado vacío después de eliminación.

12.3 Tabla Resumen de Pruebas

Endpoint	Caso Éxito	Caso Error Validado	Código Éxito	Código Error
login	Credenciales correctas	Password incorrecta	200	400
alta_usuario	Registro completo	Campo obligatorio vacío	200	400
consulta_usuario	Token válido	Token inválido	200	400
modifica_usuario	Cambios válidos	Fecha nula	200	400
borra_usuario	Usuario existente	Email inexistente	200	400
alta_articulo	Datos válidos	Precio=0	200	400
consulta_articulos	Palabra válida	Token inválido	200	400
compra_articulo	Stock suficiente	Stock insuficiente	200	400
modifica_carrito_compra	Incremento permitido	Incremento sin stock / Decremento a 0	200	400
consulta_carrito_compra	Token válido	Token inválido	200	400
elimina_articulo_carrito_compra	Artículo presente	Artículo ausente	200	400
elimina_carrito_compra	Carrito con artículos	Token inválido	200	400

- **12.5 Verificación de Consistencia**

Además de las respuestas JSON, se verificó el estado real de las tablas tras operaciones clave (compra, modificación, eliminación):

Ejemplo comprobación MySQL:

```
mysql -u usuario_mysql -p -e "SELECT * FROM carrito_compra WHERE id_usuario=5;"
```

```
mysql -u usuario_mysql -p -e "SELECT id_articulo,cantidad FROM stock;"
```

```

azureuser@T7-2022630278: ~ Símbolo del sistema
Database changed
mysql> SELECT * FROM usuarios;
+-----+-----+-----+-----+-----+-----+-----+
| id_usuario | email | password | rno | fecha_nacimiento | telefono | genero |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | x@x.com | 2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881 | fWUdcXU01TEaRkfm6M2g | xampp | xx | xxxx
| 2025-11-12 07:49:00 | 1111111111 | M |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> SELECT * FROM usuarios;
+-----+-----+-----+-----+-----+-----+-----+
| id_usuario | email | password | rno | fecha_nacimiento | telefono | genero |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | x@x.com | 2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881 | fWUdcXU01TEaRkfm6M2g | xampp | xx | xxxx
| 2025-11-12 07:49:00 | 1111111111 | M |
| 2 | x@x2.com | x2 | 2024-05-01 12:00:00 | 5512345678 | M |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.35 sec)

mysql> SELECT * FROM usuarios;
+-----+-----+-----+-----+-----+-----+-----+
| id_usuario | email | password | rno | fecha_nacimiento | telefono | genero |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | x@x.com | 2d711642b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881 | fWUdcXU01TEaRkfm6M2g | TestMod | Ejemplo | NULL
| 2024-05-01 12:00:00 | 5511111111 | M |
| 2 | x@x2.com | x2 | 2024-05-01 12:00:00 | 5512345678 | M |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.35 sec)

```

```

azureuser@T7-2022630278: ~ Símbolo del sistema
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| servicio_web |
| sys |
+-----+
5 rows in set (0.40 sec)

mysql> show tables;
+-----+
| Tables_in_servicio_web |
+-----+
| carro_compra |
| fotos_articulos |
| fotos_usuarios |
| stock |
| usuarios |
+-----+
5 rows in set (0.60 sec)

mysql> SELECT * FROM stock;
+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | precio | cantidad |
+-----+-----+-----+-----+
| 1 | Mayonesa 500g | Envase familiar | 35.50 | 99 |
| 2 | Mayonesa 500g | Envase familiar | 35.50 | 100 |
| 4 | Mayonesa McCormick 100gr | Mayonesa | 100.00 | 100 |
+-----+-----+-----+-----+
3 rows in set (0.45 sec)

mysql> SELECT * FROM carro_compra;
Empty set (0.31 sec)

mysql> SELECT * FROM fotos_articulos;
Empty set (0.24 sec)

mysql> |

```

Imagen 89: Consola MySQL mostrando contenido de `carrito_compra` y `stock` tras varias operaciones.

13 Pruebas Funcionales del Front-End (Móvil)

Las pruebas funcionales del front-end se realizaron empleando un dispositivo móvil (teléfono inteligente) para verificar que cada requerimiento del apartado de comercio electrónico se ejecuta correctamente mediante la interfaz HTML/JavaScript. El objetivo principal fue constatar que las pantallas diseñadas (“Captura de artículo”, “Compra de artículos” y “Carrito de compra”) responden adecuadamente al flujo de usuario previsto: autenticación, gestión de artículos y operaciones sobre el carrito, incluyendo incrementos, decrementos, compras y eliminaciones. Se utilizó el navegador móvil (Chrome) sobre conexión Wi-Fi, accediendo directamente mediante la URL pública: http://<IP_VM>:8080/prueba.html. Las capturas registran el aspecto visual, la secuencia de pasos y los mensajes de retroalimentación (alertas, cambios instantáneos en listado y totales).

La metodología consistió en:

1. Preparar previamente en el back-end un usuario válido, artículos iniciales y estado limpio de carrito.
2. Ingresar al sistema desde el móvil, ejecutar cada acción funcional de manera aislada y luego en combinación (por ejemplo: buscar artículo → cambiar cantidad → comprar → verificar carrito → modificar +/- → eliminar).
3. Registrar capturas antes y después de cada evento relevante (botón presionado y resultado).
4. Validar mensajes de error esperados (stock insuficiente o decremento indebido) manipulando cantidades extremas en la pantalla de compra o carrito.
5. Confirmar la visualización correcta de imágenes (foto por defecto o foto cargada en artículos) y la actualización dinámica del total en el carrito.

Además, se comprobó la adaptabilidad mínima de la interfaz en orientación vertical (portrait), descartando problemas de desplazamiento horizontal y confirmando que el tamaño restringido (250px de ancho central) facilitó el uso táctil sin superposiciones. Aunque no se aplicó un framework de diseño responsivo avanzado, el prototipo cumple la accesibilidad básica de interacción para la tarea.

13.1 Dispositivo y Entorno

- Dispositivo de prueba: Teléfono inteligente.
- Navegador: Chrome móvil actualizado.
- Conectividad: Wi-Fi estable.

- URL utilizada: <http://4.248.144.161:8080/prueba.html>.

The screenshot shows a Linux desktop environment. On the left, a terminal window titled "Simbolo del sistema - scrypy" displays MySQL queries and their results:

```

+-----+-----+-----+
| 1 | x@x.com | 2d71164b726b04401627ca9fbac32f5c8530fb1903cc4db02258717921a4881 | SCS3FbT5M2dNSesFFmc5 | Te
+-----+-----+-----+
1 row in set (0.49 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| servicio_web |
| sys |
+-----+
5 rows in set (0.95 sec)

mysql> show tables;
+-----+
| Tables_in_servicio_web |
+-----+
| carro_compra |
| fotos_articulos |
| fotos_usuarios |
| stock |
| usuarios |
+-----+
5 rows in set (0.80 sec)

mysql> SELECT * FROM stock;
+-----+-----+-----+-----+
| id_articulo | nombre | descripcion | precio | cantidad |
+-----+-----+-----+-----+
| 1 | Mayonesa 500g | Envase familiar | 35.50 | 100 |
+-----+-----+-----+-----+
1 row in set (0.49 sec)

mysql>
mysql>

```

On the right, a browser window titled "Acceso al sistema" shows a login form with fields for "Email" and "Password", and buttons for "Aceptar" and "Registro de usuario". The browser's address bar shows the URL "4.248.144.161:8080/c".

13.2 Casos de Prueba por Requerimiento Funcional

A continuación se listan los casos funcionales del front-end y las evidencias requeridas. Cada captura sugerida posee un identificador de imagen incremental para mantener coherencia con la numeración previa (continuando después de la imagen 35 de la sección anterior).

1. Registro y Login de usuario

- Paso: Introducir email, contraseña, datos obligatorios, registrar y luego iniciar sesión.
- Verificación: Aparición de menú principal con nuevas opciones (“Captura de artículo” y “Compra de artículos”).

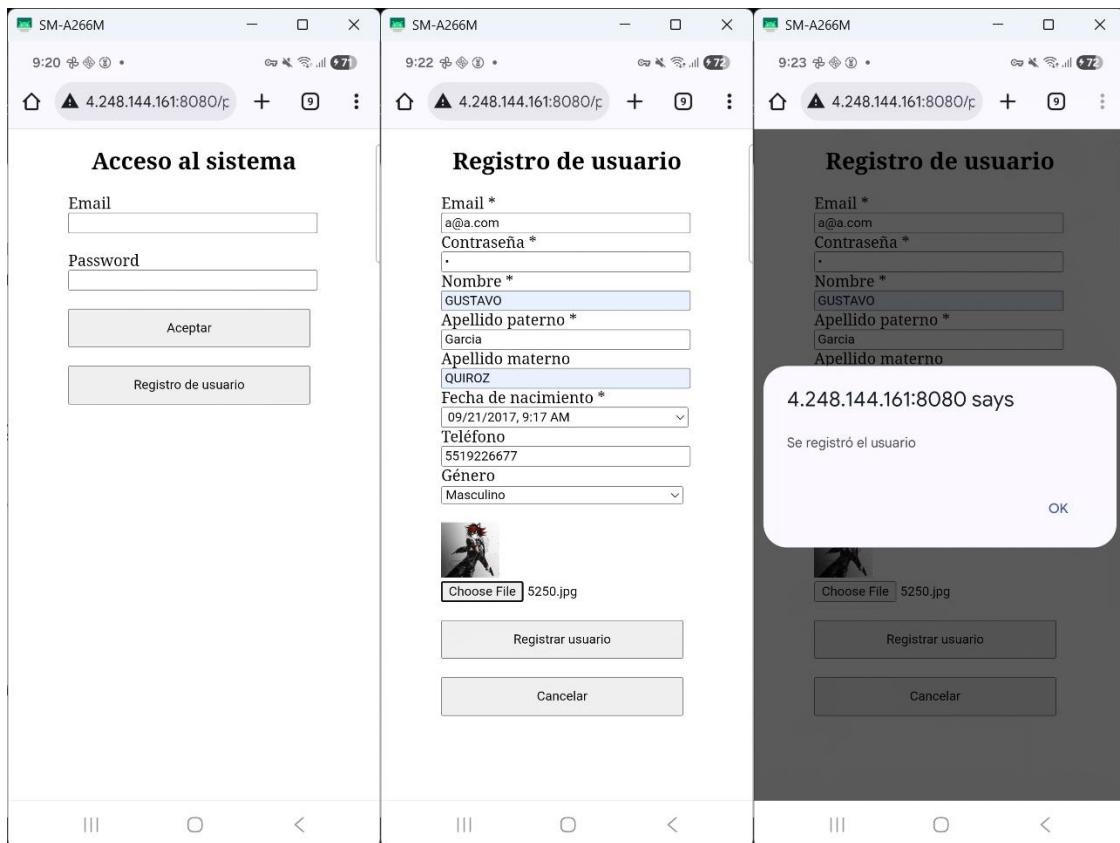


Imagen 36: Pantalla de registro completa antes de enviar.

Imagen 37: Pantalla de login mostrando resultado exitoso (ingreso al menú).

2. Acceso al Menú Principal

- Verificación de botones: Perfil, Captura de artículo, Compra de artículos, Salir.

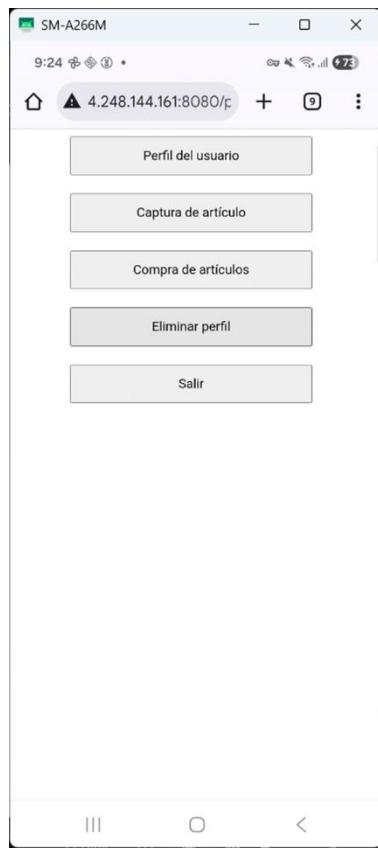


Imagen 38: Menú principal en dispositivo móvil (vista clara de opciones).

3. Captura de artículo

- Paso: Abrir pantalla, llenar nombre, descripción, precio, cantidad, cargar foto opcional y guardar.
- Resultado: Alerta de confirmación y retorno a menú (opcional).

The screenshot shows a mobile application interface titled "Captura de artículo". The form fields are as follows:

- Nombre:** Mayonesa McCormick 500mg
- Descripción:** Mayonesa
- Precio:** 50
- Cantidad:** 100

Below the form is a thumbnail image of a product and a file input field labeled "Choose File 315.jpg". At the bottom of the screen are two buttons: "Guardar artículo" and "Regresar".

Imagen 39: Formulario “Captura de artículo” completo antes de presionar “Guardar artículo”.

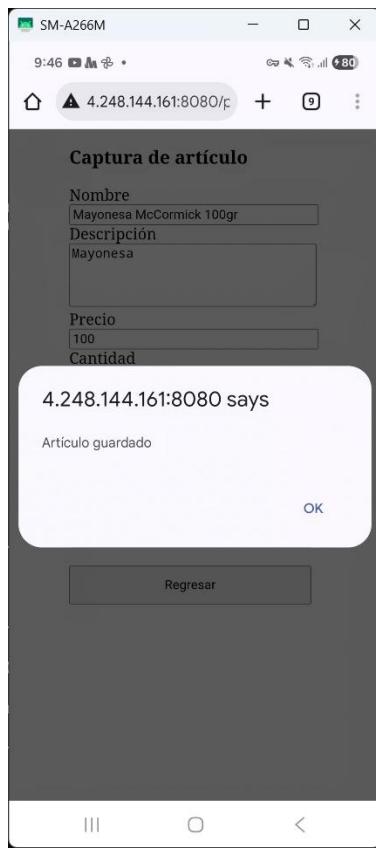


Imagen 40: Mensaje de confirmación (alert) o retorno al menú tras alta.

4. Búsqueda de artículos

- Paso: Ingresar palabra clave y ejecutar búsqueda.
- Verificación: Listado de resultados con miniatura, nombre, descripción, precio, controles de cantidad y compra.

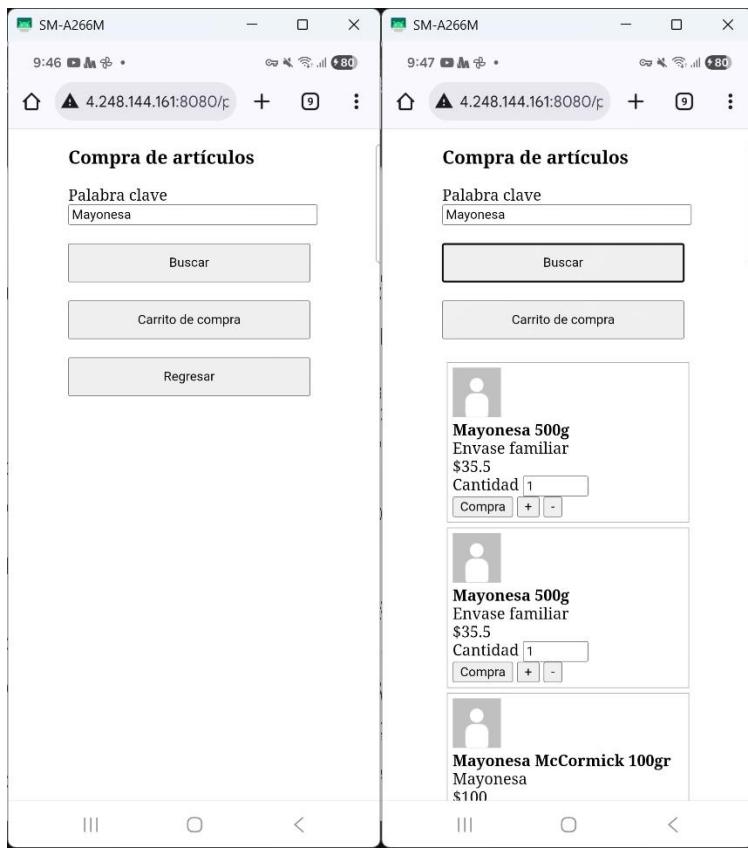


Imagen 41: Pantalla “Compra de artículos” con resultados tras la búsqueda.

5. Ajuste de cantidad previo a compra

- Paso: Usar botones +/- o campo numérico para cambiar cantidad inicial (por defecto 1).

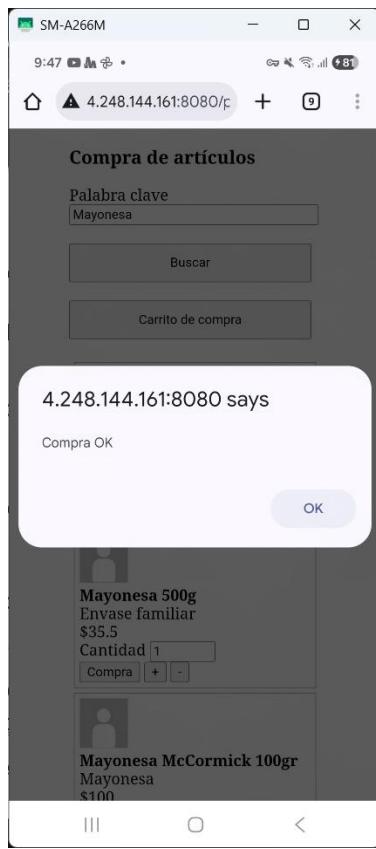


Imagen 42: Ejemplo de artículo con cantidad modificada (campo numérico actualizado).

6. Compra de artículo

- Paso: Presionar “Compra” (cantidad > 0).
- Resultado: Confirmación y actualización interna del stock (no visible directamente pero reflejada en futuras operaciones).

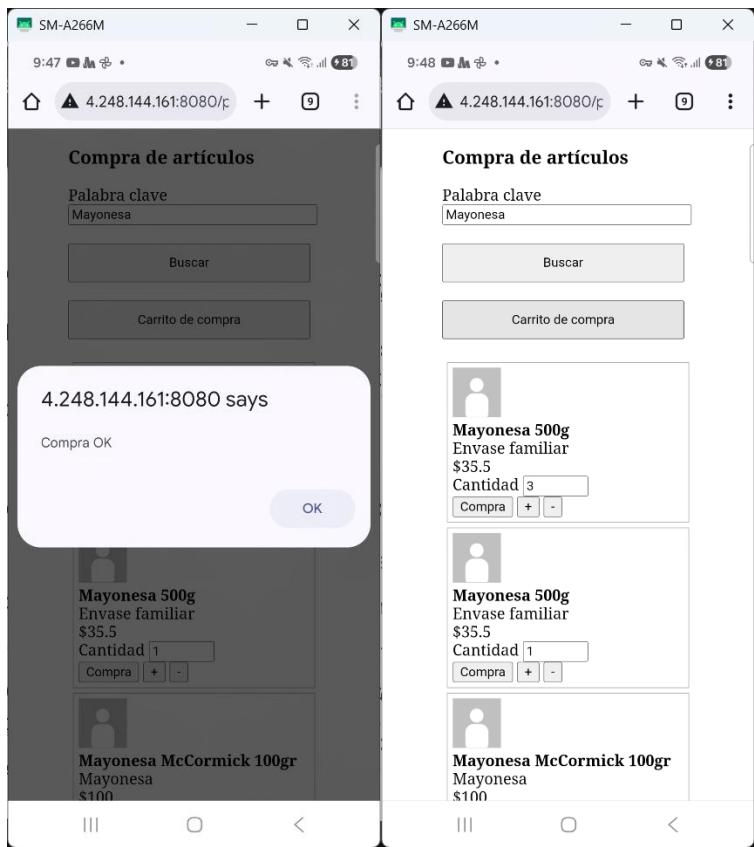


Imagen 43: Alerta de compra exitosa (mensaje de confirmación).

7. Visualización del carrito de compra

- Paso: Presionar botón “Carrito de compra”.
- Verificación: Lista de artículos con miniatura, precio, cantidad, costo (cantidad × precio) y total acumulado.

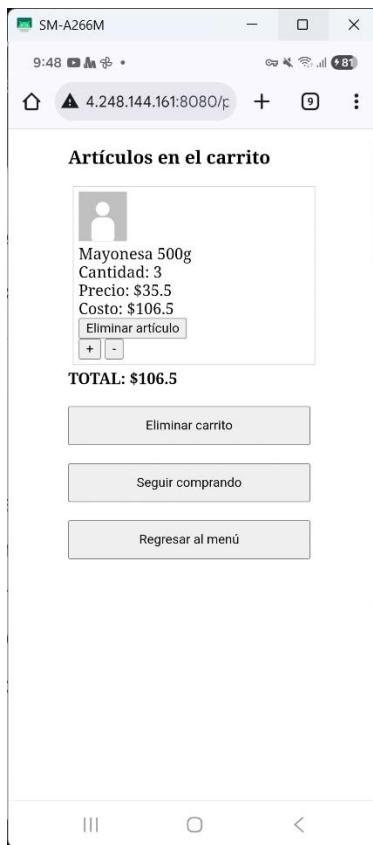


Imagen 44: Pantalla “Artículos en el carrito” mostrando uno o más artículos.

8. Incremento de cantidad dentro del carrito

- Paso: Botón + en un artículo del carrito (llama a modifica_carrito_compra).
- Resultado: Cantidad y costo actualizados, total recalculado.

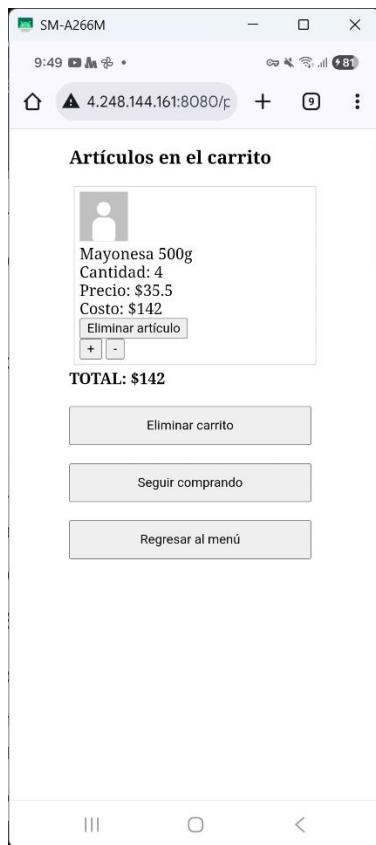


Imagen 45: Carrito después de incrementar la cantidad (resaltando cambio numérico).

9. Decremento de cantidad dentro del carrito

- Paso: Botón – en un artículo (sin llegar a invalidar las reglas, es decir no provocar mensaje de error).
- Resultado: Cantidad disminuida, costo recalculado.

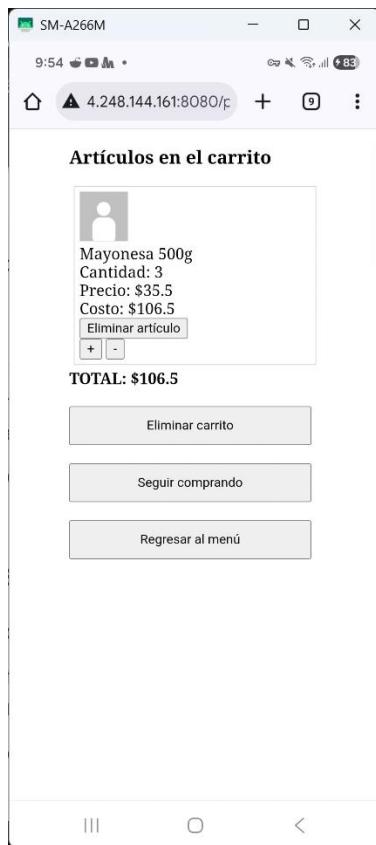


Imagen 46: Carrito tras decremento válido de cantidad.

10. Decremento invalidado (opcional para error)

- Paso: Intentar reducir la cantidad por debajo del mínimo permitido (según lógica del prototipo).
- Resultado: Alerta con mensaje “No hay más artículos en el carrito”.

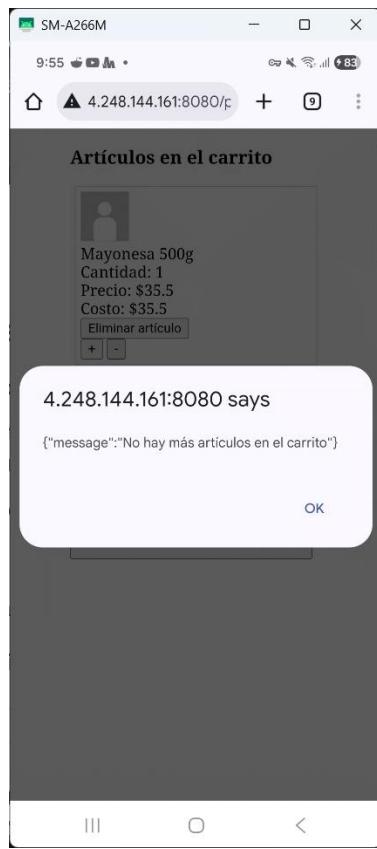


Imagen 47: Alerta de error de decremento.

11. Eliminación de artículo específico

- Paso: Presionar “Eliminar artículo” sobre uno de los ítems.
- Resultado: Ítem desaparece y el total se ajusta.

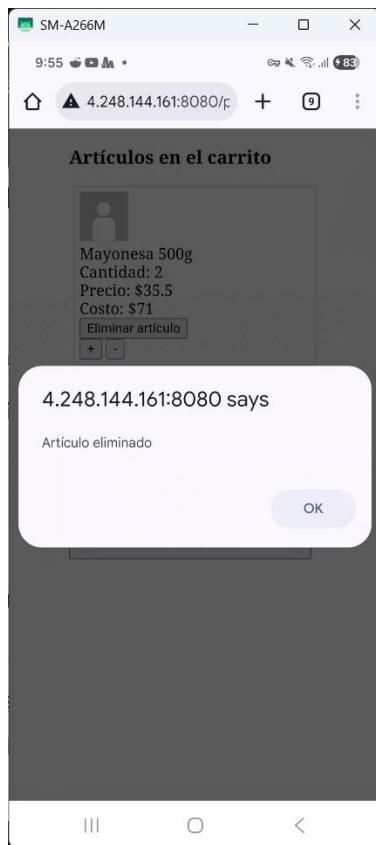


Imagen 48: Carrito tras eliminar un artículo (lista reducida).

12. Eliminación total del carrito

- Paso: Press “Eliminar carrito”, confirmar.
- Resultado: Lista vacía y total igual a 0.

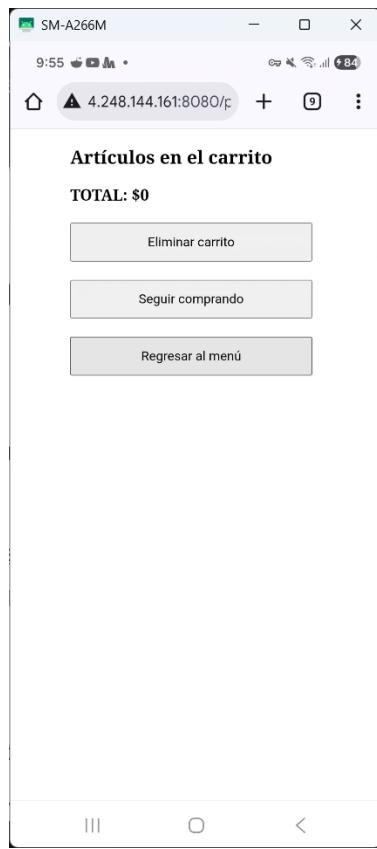


Imagen 49: Carrito vacío después de la operación.

13. Regresar a la pantalla de compra

- Paso: “Seguir comprando” desde el carrito.
- Resultado: Retorno a la pantalla de búsqueda sin perder token.



Imagen 50: Pantalla “Compra de artículos” reabierta tras navegar desde carrito.

14. Carga y visualización de imagen de artículo

- Paso: Capturar nuevo artículo con foto; luego buscar y verificar miniatura en resultados y en carrito.



Imagen 51: Resultado de búsqueda mostrando artículo con foto personalizada.



Imagen 52: Carrito mostrando la miniatura del artículo comprado.

15. Flujo completo encadenado (opcional para evidencia global)

- Paso: Registro → Login → Captura artículo → Compra → Carrito → Modificar → Eliminar.

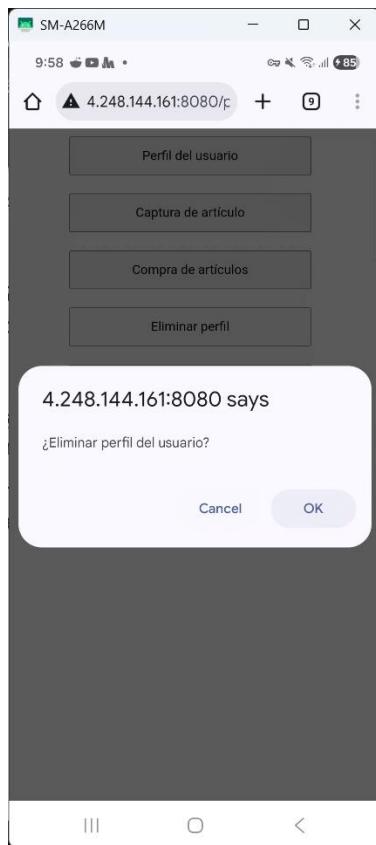


Imagen 53: Collage o secuencia final demostrando el recorrido completo (puede ser un montaje o captura extendida).

13.3 Validación de Mensajes y Errores

Se forzaron condiciones para generar mensajes de error:

- Stock insuficiente (compra de cantidad exagerada) desde pantalla de compra: alerta “No hay suficientes artículos en stock”.
- Decremento indebido en carrito: alerta “No hay más artículos en el carrito”.
- Token incorrecto (simulación manual en código no mostrada en interfaz común): interfaz muestra alert genérico de acceso denegado cuando se intercepta respuesta 400 con dicho mensaje.

Estos mensajes confirman que el front-end interpreta correctamente la estructura uniforme de errores JSON retornados por el back-end.

14 Enlace del chat de la IA generativa

Enlace: <https://github.com/copilot/share/80155284-4a44-8c87-a112-324f043a68d3>

15 Conclusiones

El desarrollo del prototipo de comercio electrónico logró integrar de forma coherente las funcionalidades solicitadas: alta y consulta de artículos, manejo completo del carrito de compra (agregar, modificar, eliminar y vaciar), y ampliación del servicio REST previo de usuarios sin romper compatibilidad con la arquitectura inicial de la Tarea 2. La adopción de transacciones en las operaciones críticas aseguró la consistencia entre el inventario (stock) y las reservas temporales (carrito), evitando desbalances comunes en escenarios de compra simultánea. La simplicidad estructural (un único WAR y un front-end HTML/JavaScript plano) permitió un aprendizaje enfocado en los principios esenciales de servicios REST, persistencia relacional y comunicación cliente-servidor.

El uso de GitHub Copilot aceleró la generación de plantillas repetitivas (endpoints, POJOs y patrones de manejo de errores), pero la validación manual fue imprescindible para garantizar la correcta aplicación de reglas de negocio y el empleo adecuado de transacciones y bloqueos (SELECT ... FOR UPDATE). El enfoque de seguridad aplicado (token + hash SHA-256 en cliente) es suficiente a nivel académico, aunque se identificaron varias oportunidades de robustecimiento para una versión productiva (expiración de token, hashing con sal adaptativa, HTTPS obligatorio, control de roles, rate limiting).

En el front-end se confirmó que cada requerimiento funcional se cumple de forma clara desde un dispositivo móvil: la navegación por pantallas, la captura de artículos, la compra y el ajuste de cantidades en el carrito resultaron intuitivos y consistentes. Las pruebas con curl reforzaron la verificación de cada método del back-end, mostrando respuestas uniformes (códigos 200 y 400) y mensajes de error explícitos. Esta dualidad de evidencia (capturas móviles + consola) robustece el reporte y cumple con los lineamientos de entrega.

El prototipo cumple los objetivos académicos y deja un terreno fértil para modularizar componentes, integrar capas adicionales (historial de pedidos, pagos simulados) y elevar el estándar de seguridad. Se alcanzó así un equilibrio entre simplicidad y funcionalidad

que facilita la comprensión del ciclo completo “usuario autenticado → artículo → carrito → consistencia transaccional”.

16 Referencias (Formato IEEE)

- [1] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Univ. California, Irvine, 2000.
- [2] Oracle, "Java Platform Standard Edition 17 API Specification – Class SecureRandom," 2025. [Online]. Available: <https://docs.oracle.com/>
- [3] ISO/IEC 9075-1:2016, "Information technology — Database languages — SQL — Part 1: Framework (SQL/Framework)."
- [4] Oracle, "InnoDB Storage Engine," MySQL 8.0 Reference Manual, 2025. [Online]. Available: <https://dev.mysql.com/doc/>
- [5] Apache Software Foundation, "Apache Tomcat 10 Documentation," 2025. [Online]. Available: <https://tomcat.apache.org/>
- [6] Eclipse Foundation, "Jakarta RESTful Web Services (JAX-RS) Specification," 2025. [Online]. Available: <https://jakarta.ee/specifications/>
- [7] FasterXML, "Jackson Databind and Core Libraries," 2025. [Online]. Available: <https://github.com/FasterXML/jackson>
- [8] GitHub, "GitHub Copilot: AI Pair Programmer," 2025. [Online]. Available: <https://github.com/features/copilot>
- [9] NIST, "Secure Hash Standard (SHS)," FIPS PUB 180-4, Aug. 2015.
- [10] W3C, "File API," W3C Recommendation, 2015. [Online]. Available: <https://www.w3.org/TR/FileAPI/>
- [11] W3C, "Web Cryptography API," W3C Recommendation, 2017. [Online]. Available: <https://www.w3.org/TR/WebCryptoAPI/>
- [12] OWASP Foundation, "OWASP Top 10: Web Application Security Risks," 2023. [Online]. Available: <https://owasp.org/>
- [13] D. Thomas et al., "Bcrypt Password Hashing," OpenBSD Project Documentation, 2025. [Online]. Available: <https://man.openbsd.org/>
- [14] P. Wuille et al., "Argon2: The Memory-Hard Function for Password Hashing," RFC Draft (argon2), 2025.
- [15] Google, "Fetch API Living Standard," WHATWG, 2025. [Online]. Available: <https://fetch.spec.whatwg.org/>

- [16] Mozilla, “Content Security Policy (CSP) Guide,” MDN Web Docs, 2025. [Online]. Available: <https://developer.mozilla.org/>
- [17] Git, “Distributed Version Control System,” Git Project, 2025. [Online]. Available: <https://git-scm.com/>
- [18] Azure, “Azure Virtual Machines Documentation,” Microsoft Learn, 2025. [Online]. Available: <https://learn.microsoft.com/azure/virtual-machines/>