

- Se verificó que la Function App pueda leer archivos via File.ReadAllText(ROOT + path).

Tabla de configuración principal:

Variable	Valor	Uso
----------	-------	-----

ROOT	/home/site/wwwroot/front-end	Raíz del front-end para Get
Server	t8mysql2022630278.mysql.database.azure.com	Conexión a MySQL PaaS
UserID	x	Usuario para MySQL
Password	Aaaaaaaaaaa0	Secreto para MySQL
Database	servicio_web	Base de datos del sistema

Tabla 12 Configuración de montaje /t8pm2022630278 y variables de entorno en la Function App.

8.4 Publicación de Azure Functions (t8ap2022630278)

Se realizó la publicación del proyecto de Azure Functions desde Visual Studio Code a la Function App t8ap2022630278 en la región Canada. Se accedió al panel de despliegue, se autenticó la suscripción de Azure for Students y se seleccionó la Function App correcta. Se validó la presencia de los endpoints (/api/login, /api/alta_usuario, /api/Get, etc.) y el correcto enlazado con MySQL PaaS.

- Se instaló la extensión de Azure Functions para VS Code (si no estaba instalada).
- Se accedió a “Deploy to Function App” y se seleccionó t8ap2022630278.
- Se verificó que las funciones respondieran con 200/400 según casos.

The screenshot displays two views of the Azure Functions extension in VS Code:

- Top View:** Shows the function code for the 'front-end' project. The code is an HTML file with a container div containing a header, a total div, and several buttons for managing a shopping cart.
- Bottom View:** Shows the deployment logs for the function app 't8ap2022630278'. The logs detail the deployment process, including Kudu steps like RemoveWorkersstep, SyncTriggerstep, CleanUpStep, and the completion of the deployment pipeline.

Figura 32. Panel de publicación en VS Code y confirmación de endpoints en la Function App.

8.5 Pruebas de acceso a la aplicación: URL [/api/Get?nombre=/prueba.html](https://t8ap2022630278.azurewebsites.net/api/Get?nombre=/prueba.html)

Se accedió al front-end mediante la URL pública de la Function App:

<https://t8ap2022630278.azurewebsites.net/api/Get?nombre=/prueba.html>

Se realizó la navegación por las distintas vistas (“Captura de artículo”, “Compra de artículos”, “Artículos en el carrito”) y se efectuaron llamadas al back-end desplegado.

Se verificó que las operaciones de carrito y stock funcionaran como en local, incluyendo validaciones y transacciones.

- Se accedió desde un dispositivo móvil y desde escritorio para validar disponibilidad.
- Se realizó la prueba de login para obtener token, y la recuperación de id_usuario.
- Se confirmó el flujo completo: alta, búsqueda, compra, modificación y eliminación del carrito.

Tabla de validaciones de producción:

Prueba	Resultado	Observaciones
Carga de prueba.html	OK	Servido vía Get desde Azure Files
Login (token)	OK	Actualiza usuarios.token
Alta de artículo	OK	Inserta en stock y opcional fotos_articulos
Consulta por palabra	OK	LIKE en nombre y descripcion
Compra y descuentos	OK	Transacción en carrito_compra y stock
Eliminar artículo/carrito	OK	Restaura stock y borra entradas
Modificar cantidad	OK	Valida stock y límites de carrito

Tabla 13 Tabla de validaciones de producción

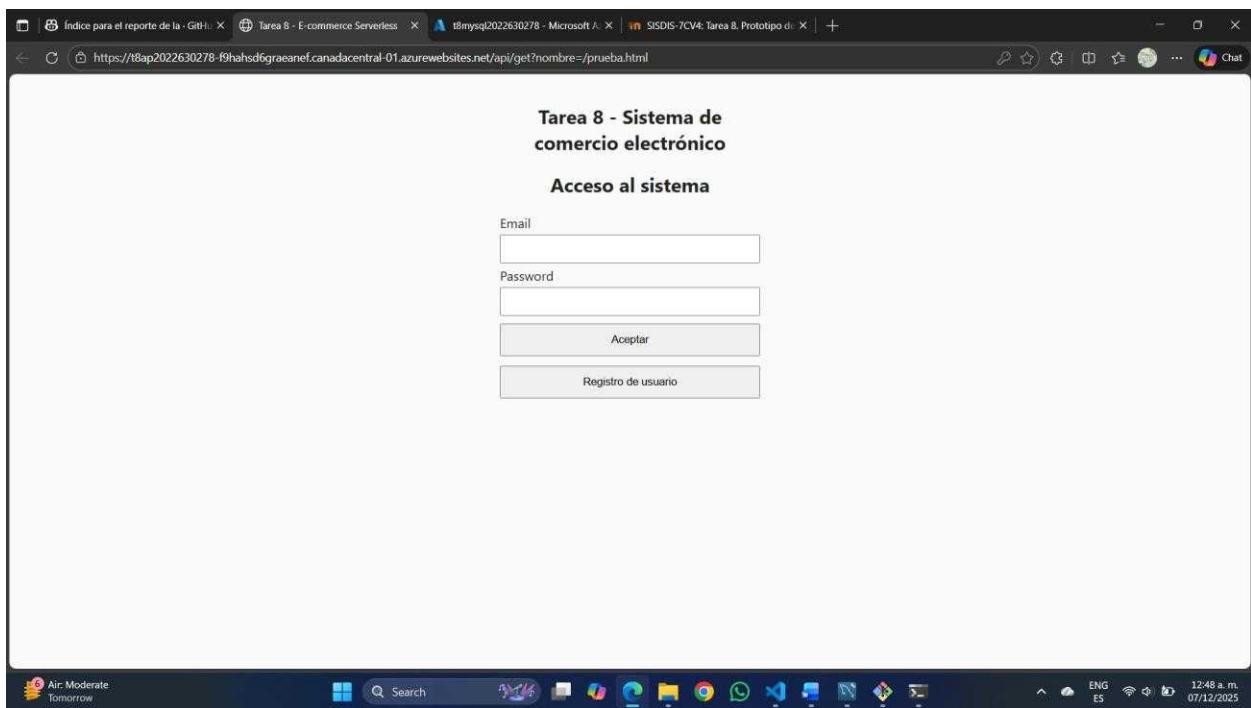


Figura 33. Acceso a la URL pública del front-end y ejecución de flujos en producción (Azure – Canada).

9 Evidencias requeridas (capturas de pantalla completas con fecha y hora)

Se realizaron y documentaron las evidencias solicitadas para el reporte, asegurando que todas las capturas sean pantallas completas, legibles y con barra de tareas mostrando fecha y hora. Las evidencias se organizaron por tipo: configuración en Azure, pruebas en dispositivo móvil del front-end y pruebas unitarias locales del back-end con curl. A continuación se indica dónde colocar cada imagen y el contenido mínimo que debe mostrar.

9.1 Variables de entorno en Azure (Function App: Configuration)

Se accedió a la Function App t8ap2022630278 y se realizó la captura de la sección “Configuration”, mostrando las variables Server, UserID, Password, Database y ROOT. Se aseguró que ROOT apunte a /t8pm2022630278 y que la región sea Canada, acorde a la nomenclatura indicada.

- Se realizó la verificación de valores y su activación.
- Se accedió al apartado de “Path mounts” para corroborar el montaje del File Share.

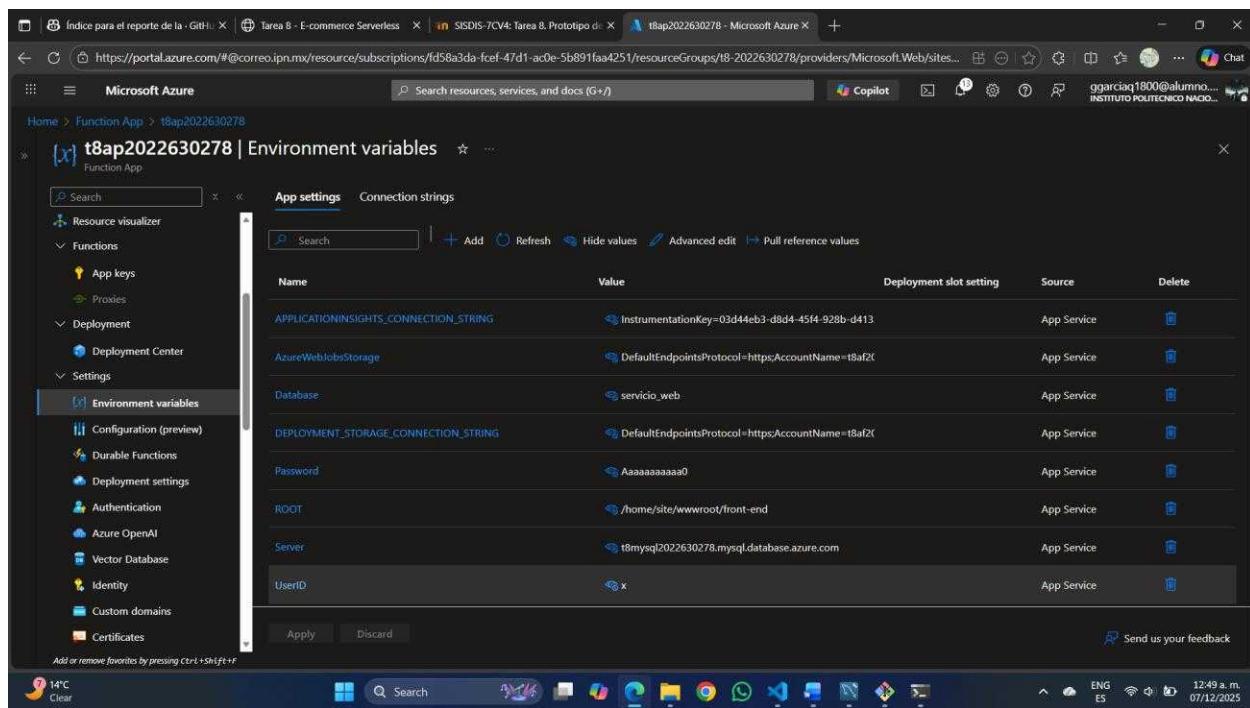


Figura 34. Configuración de variables de entorno y montaje de ruta en t8ap2022630278 (pantalla completa con fecha/hora).

9.2 Pruebas en dispositivo móvil del front-end

Se accedió al front-end desde un teléfono o tableta, y se realizaron las pruebas de cada requerimiento funcional. Cada captura debe mostrar la vista correspondiente, los controles y los resultados visibles, además de la fecha y hora del dispositivo.

Tabla de evidencias por vista:

Vista / Acción	Qué debe mostrar la captura	Endpoint implicado
Captura de artículo	Formulario con nombre, descripción, precio, cantidad y foto; confirmación de alta	alta_articulo
Búsqueda y listado	Campo de palabra clave; lista con mini-foto, nombre, descripción, precio; botones +/- y “Compra”	consulta_articulos, modifica_carrito_compra, compra_articulo
Compra de artículo	Confirmación visible de la compra y ajuste de stock	compra_articulo

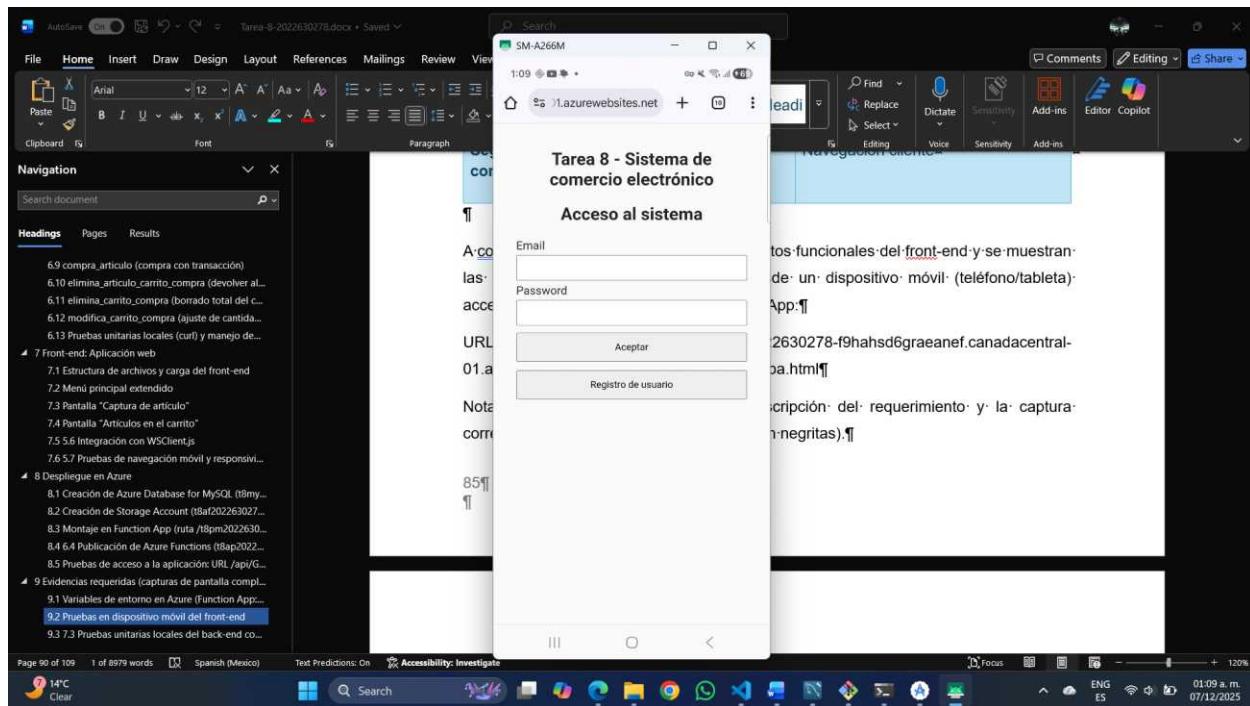
Carrito de compra	Lista de artículos, cantidad, precio, costo por artículo y total	(consulta/refresh del carrito)
Eliminar artículo	Botón y confirmación de eliminación; stock restaurado	elimina_articulo_carrito_compra
Eliminar carrito	Vaciar el carrito y restauración del stock	elimina_carrito_compra
Seguir comprando	Navegación de regreso a la vista de “Compra de artículos”	Navegación cliente

Tabla 14 Tabla de evidencias por vista

A continuación se describen los requerimientos funcionales del front-end y se muestran las evidencias de prueba ejecutadas desde un dispositivo móvil (teléfono/tableta) accediendo a la URL pública de la Function App:

URL de acceso:

<https://t8ap2022630278-f9hahsd6graeanef.canadacentral-01.azurewebsites.net/api/get?nombre=/prueba.html>



9.2.1 Requerimiento 1: Pantalla “Captura de artículo”

- Descripción: Al seleccionar “Captura de artículo”, se muestra una pantalla para capturar nombre, descripción, precio, cantidad en existencia y la fotografía del artículo. El botón “Guardar” envía la alta al back-end (función alta_articulo), incluyendo id_usuario y token.

Tarea 8 - Sistema de comercio electrónico

Acceso al sistema

Email
x1@x1.com

Password
.....

Aceptar

Registro de usuario

Tarea 8 - Sistema de comercio electrónico

Captura de artículo

Nombre *
Leche Alpura 1L

Descripción *
Leche

Precio *
47.99

Cantidad *
10

Fotografía

Choose Files 6347.jpg

Guardar

Regresar



Figura 1. Pantalla “Captura de artículo” en móvil (campos visibles y carga de imagen)



Figura 2. Confirmación de artículo registrado (toast o alerta de éxito en móvil)

9.2.2 Requerimiento 2: Pantalla “Compra de artículos” con búsqueda por palabra clave

- Descripción: La pantalla permite ingresar una palabra clave, que se busca con LIKE en las columnas nombre y descripción de la tabla stock. Se llama a la función consulta_articulos del back-end.
- Evidencia:



Figura 3. Búsqueda por palabra clave y resultados con imagen, nombre, descripción y precio en móvil

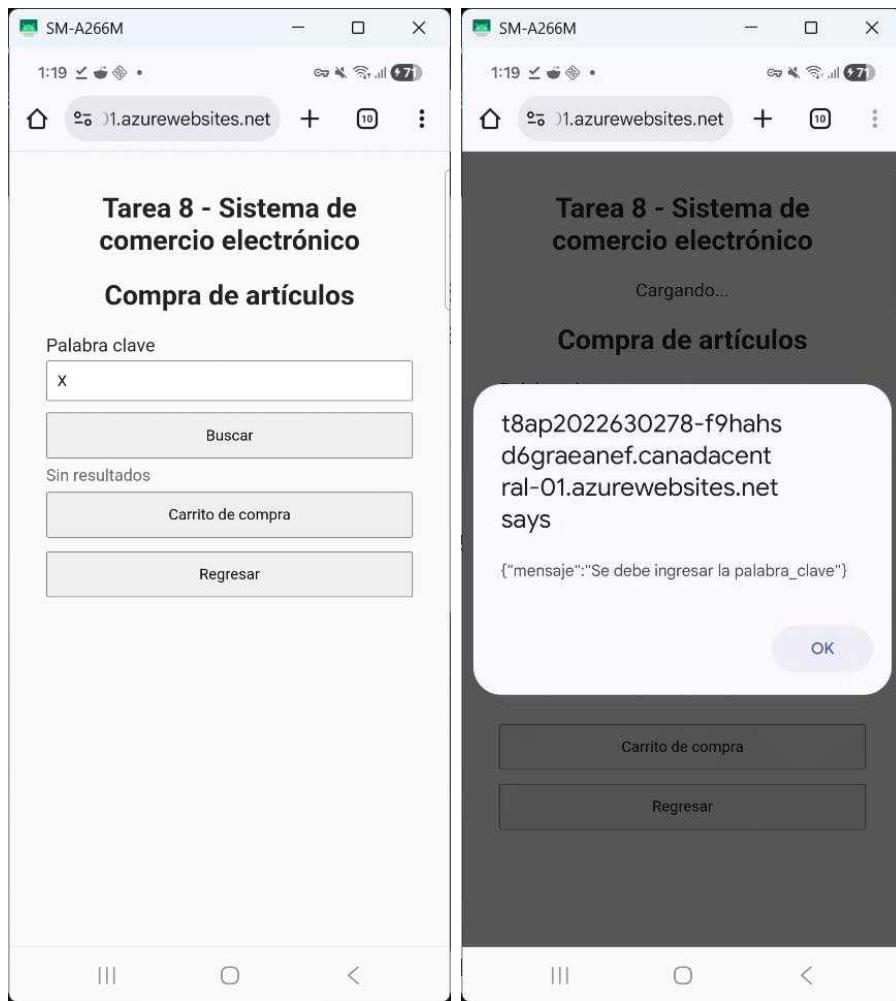


Figura 4. Indicador de “Sin resultados” cuando la búsqueda no devuelve artículos

9.2.3 Requerimiento 3: Agregar artículo al carrito desde “Compra de artículos”

- Descripción: Cada resultado muestra un campo “Cantidad” y el botón “Agregar al carrito”. Al presionar, se invoca compra_articulo (back-end), que inserta/actualiza en carrito_compra y descuenta del stock bajo transacción.
- Evidencia:



Figura 5. Botón “Agregar al carrito” con cantidad especificada y mensaje de agregado



Figura 6. Confirmación de agregado al carrito (toast o alerta de éxito)

9.2.4 Requerimiento 5: Pantalla “Artículos en el carrito” con total

- Descripción: El botón “Carrito de compra” despliega los artículos en carrito_compra con mini-imagen, nombre, cantidad, precio y costo (cantidad x precio), así como el total. Se invoca consulta_carrito (back-end).
- Evidencia:



Figura 7. Pantalla “Artículos en el carrito” mostrando líneas con imagen, nombre, cantidad, precio y costo

9.2.5 Requerimiento 6: Modificación de cantidad en el carrito (+/-)

- Descripción: En “Artículos en el carrito” existen botones “+” y “-” por línea para modificar la cantidad (función `modifica_carrito_compra`). La operación ajusta carrito y stock dentro de una transacción. Maneja casos de error (stock insuficiente o cantidad ya en 0) mostrando mensajes.
- Evidencia:



Figura 9. Incremento de cantidad con botón “+” y actualización de costo/total



Figura 10. Decremento de cantidad con botón “-” y actualización de costo/total



Figura 11. Mensaje de error “No hay suficientes artículos en stock” al intentar exceder existencias