



Figura 12. Mensaje de error “No hay más artículos en el carrito” al intentar decrementar por debajo de 0

9.2.6 Requerimiento 6 (extensión coherente) y 7: Eliminar artículo y eliminar carrito

- Descripción:
 - “Eliminar artículo”: Invoca `elimina_articulo_carrito_compra`, regresa la cantidad al stock y borra el artículo dentro de una transacción.
 - “Eliminar carrito”: Invoca `elimina_carrito_compra`, regresa todas las cantidades al stock y borra todos los registros del carrito dentro de una transacción.

- Evidencia:



Figura 13. Eliminación de artículo individual del carrito y actualización del total



Figura 14. Eliminación completa del carrito y confirmación de carrito vacío

9.2.7 Botón adicional “Comprar todo” (checkout)

- Descripción: El botón “Comprar todo” realiza el checkout creando una orden (ordenes) y detalles (orden_detalle) a partir del carrito, y lo vacía sin regresar stock. Se invoca la función finaliza_compra (back-end). Devuelve id_orden y total.
- Evidencia:



Figura 15. Ejecución de “Comprar todo” y confirmación con número de orden y total



Figura 16. Carrito vacío tras finalizar la compra

9.2.8 Navegación entre pantallas y login/registro

- Descripción:
 - Desde el menú principal se accede a Perfil, Captura de artículo y Compra de artículos.
 - El login genera token y se obtiene id_usuario con consulta_usuario.
 - El registro de usuario y modificación de perfil usan las funciones existentes (alta_usuario, consulta_usuario, modifica_usuario, borra_usuario).
- Evidencia:

Figura 17. Pantalla de login en móvil

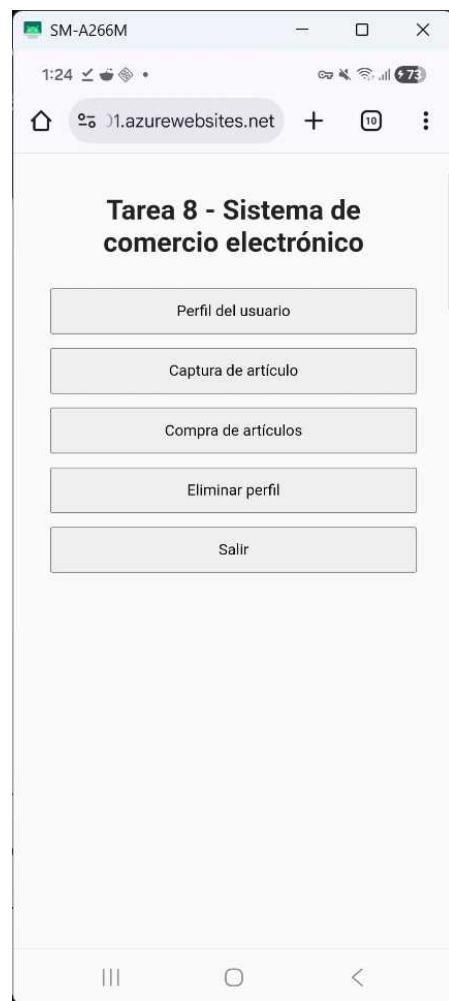


Figura 18. Menú principal con opciones “Perfil del usuario”, “Captura de artículo”, “Compra de artículos”

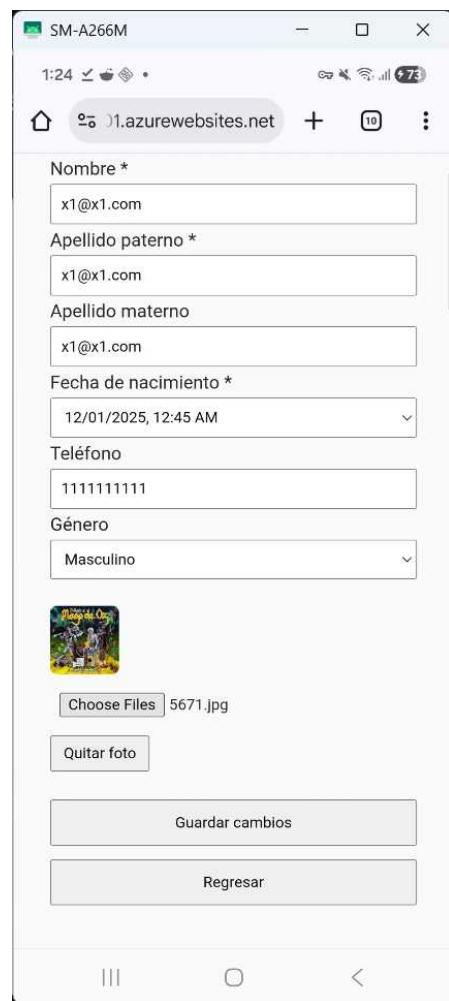


Figura 19. Pantalla “Perfil del usuario” con edición de datos e imagen

9.3 Pruebas unitarias del despliegue del back-end con curl

Se realizaron pruebas locales invocando cada endpoint del back-end con curl, capturando la consola con los comandos y las respuestas en JSON, además del código HTTP (cuando sea visible). Las capturas deben ser de la terminal en pantalla completa, con barra de tareas y fecha/hora.

Tabla de endpoints a evidenciar con ejemplo de llamada:

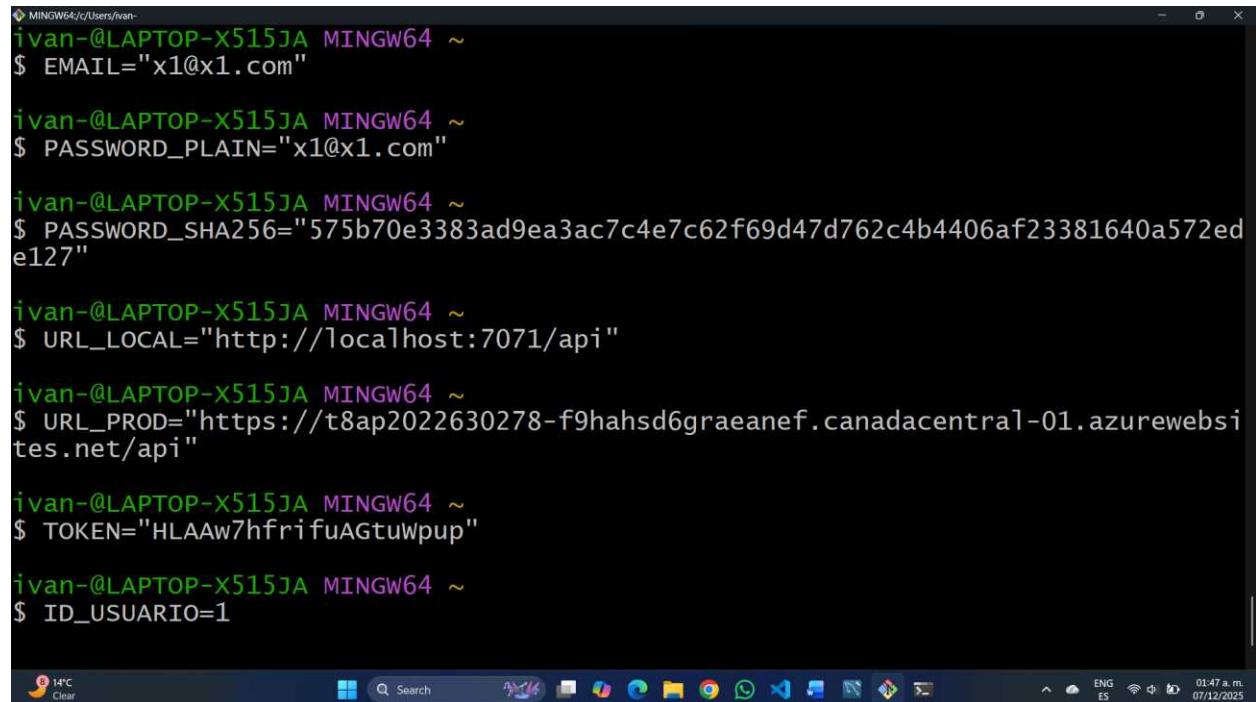
Función	Método	Ejemplo de curl (local)
login	GET	curl "http://localhost:7071/api/login?email=...&password=...sha256..."
alta_usuario	POST	curl -X POST -H "Content-Type: application/json" -d "..." http://localhost:7071/api/alta_usuario
consulta_usuario	GET	curl "http://localhost:7071/api/consulta_usuario?email=...&token=..."
modifica_usuario	PUT	curl -X PUT -H "Content-Type: application/json" "http://localhost:7071/api/modifica_usuario?email=...&token=..." -d "..."
borra_usuario	DELETE	curl -X DELETE "http://localhost:7071/api/borra_usuario?email=...&token=..."
alta_articulo	POST	curl -X POST -H "Content-Type: application/json" -d "..." http://localhost:7071/api/alta_articulo
consulta_articulos	GET	curl "http://localhost:7071/api/consulta_articulos?palabra_clave=...&id_usuario=...&token=..."
compra_articulo	POST	curl -X POST -H "Content-Type: application/json" -d "..." http://localhost:7071/api/compra_articulo
elimina_articulo_carrito_compra	DELETE	curl -X DELETE "http://localhost:7071/api/elimina_articulo_carrito_compra?id_usuario=...&id_articulo=...&token=..."
elimina_carrito_compra	DELETE	curl -X DELETE "http://localhost:7071/api/elimina_carrito_compra?id_usuario=...&token=..."
modifica_carrito_compra	PUT	curl -X PUT -H "Content-Type: application/json" "http://localhost:7071/api/modifica_carrito_compra?id_usuario=...&token=..." -d "{"id_articulo":..., incremento:...}"

Tabla 15 Tabla de endpoints

9.4 Pruebas unitarias locales del back-end con curl (detalladas)

Las siguientes pruebas se ejecutan en Git Bash, contra el entorno local (<http://localhost:7071/api>) y/o producción según se requiera evidencia. Sustituye los valores de ejemplo por tus datos reales.

Variables de ejemplo a usar en Git Bash:



The screenshot shows a terminal window titled 'MINGW64' running on a Windows 10 desktop. The terminal displays several curl commands with placeholder variables:

```
ivan-@LAPTOP-X515JA MINGW64 ~
$ EMAIL="x1@x1.com"

ivan-@LAPTOP-X515JA MINGW64 ~
$ PASSWORD_PLAIN="x1@x1.com"

ivan-@LAPTOP-X515JA MINGW64 ~
$ PASSWORD_SHA256="575b70e3383ad9ea3ac7c4e7c62f69d47d762c4b4406af23381640a572ed
e127"

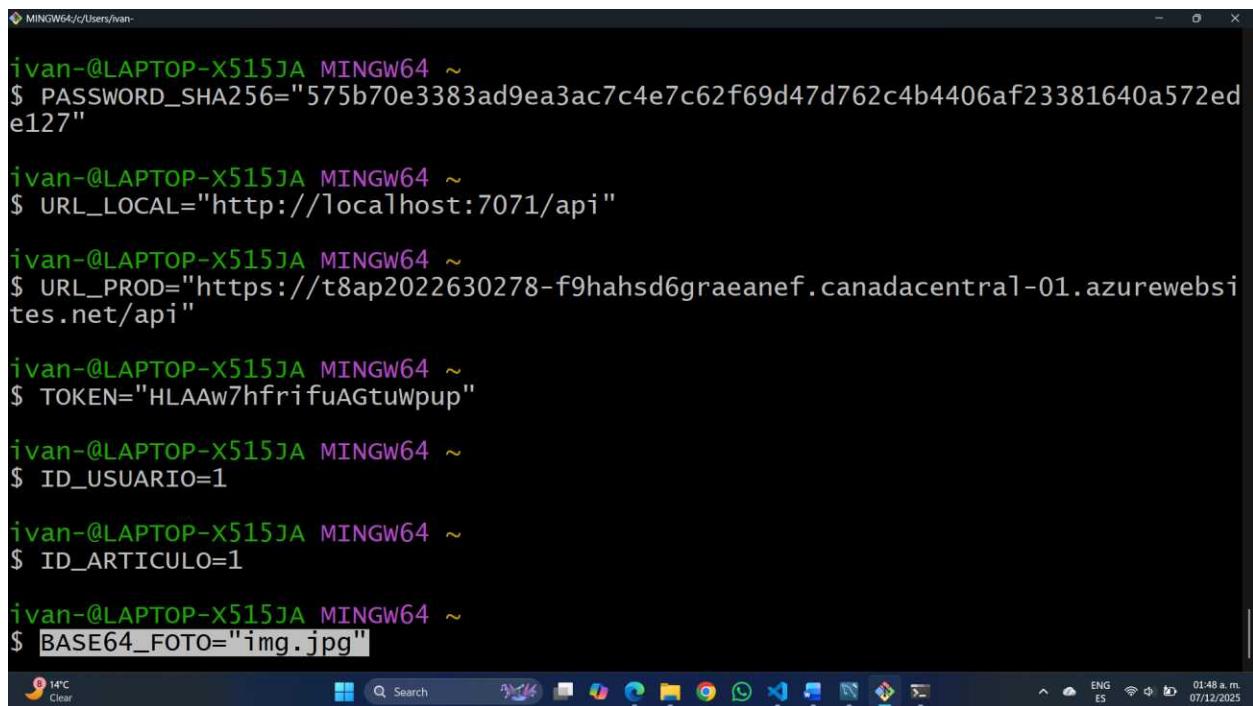
ivan-@LAPTOP-X515JA MINGW64 ~
$ URL_LOCAL="http://localhost:7071/api"

ivan-@LAPTOP-X515JA MINGW64 ~
$ URL_PROD="https://t8ap2022630278-f9hahsd6graeane.f.canadacentral-01.azurewebsi
tes.net/api"

ivan-@LAPTOP-X515JA MINGW64 ~
$ TOKEN="HLAAw7hfrijuAGtuwpup"

ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_USUARIO=1
```

The desktop taskbar at the bottom shows icons for weather (14°C), search, and various system applications. The system tray indicates the date and time as 01:47 a.m. 07/12/2025.



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ PASSWORD_SHA256="575b70e3383ad9ea3ac7c4e7c62f69d47d762c4b4406af23381640a572ed
e127"

ivan-@LAPTOP-X515JA MINGW64 ~
$ URL_LOCAL="http://localhost:7071/api"

ivan-@LAPTOP-X515JA MINGW64 ~
$ URL_PROD="https://t8ap2022630278-f9hahsd6graeanef.canadacentral-01.azurewebsi
tes.net/api"

ivan-@LAPTOP-X515JA MINGW64 ~
$ TOKEN="HLAAw7hfrifuAGtuwpup"

ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_USUARIO=1

ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_ARTICULO=1

ivan-@LAPTOP-X515JA MINGW64 ~
$ BASE64_FOTO="img.jpg"
```

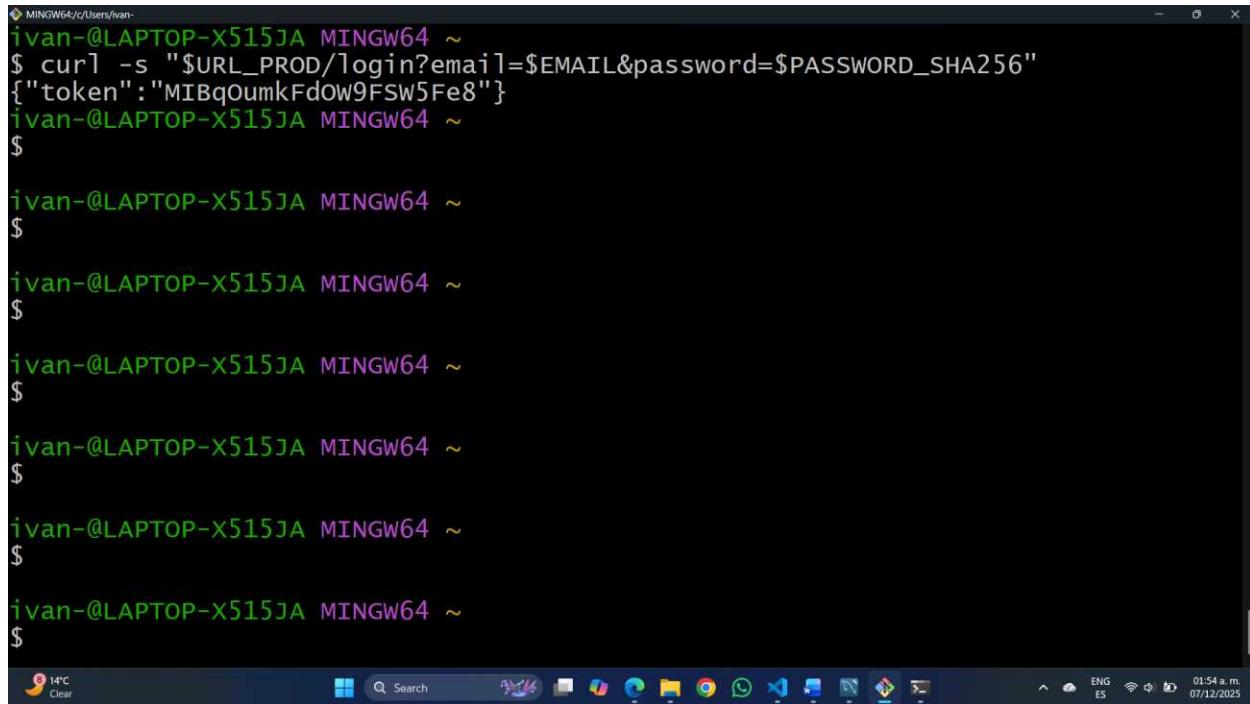
Figura 20. Preparación de variables

9.4.1 Prueba 0. Login y obtención de token (previo a las demás)

- Descripción: Verifica login y obtención de token.

```
curl -s "$URL_PROD/login?email=$EMAIL&password=$PASSWORD_SHA256"
```

- Evidencia:



```
MINGW64/c/Users/van-ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s "$URL_PROD/login?email=$EMAIL&password=$PASSWORD_SHA256"
>{"token": "MIBqOumkFdOW9FSW5Fe8"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
14°C Clear
```

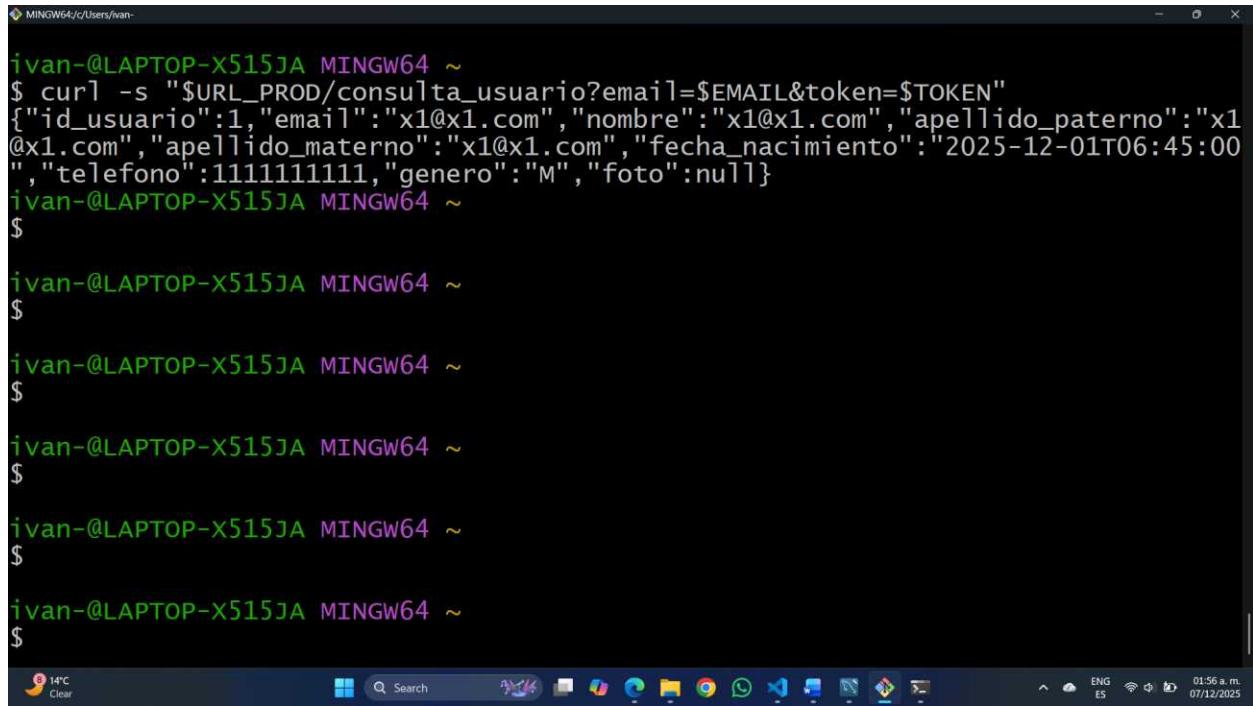
Figura 20. Respuesta JSON de login con token

9.4.2 Prueba 0.1 Consulta de usuario para obtener id_usuario

- Descripción: Verifica consulta_usuario y lectura de id_usuario.

```
curl -s "$URL_PROD/consulta_usuario?email=$EMAIL&token=$TOKEN"
```

- Evidencia:



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s "$URL_PROD/consulta_usuario?email=$EMAIL&token=$TOKEN"
{"id_usuario":1,"email":"x1@x1.com","nombre":"x1@x1.com","apellido_paterno":"x1@x1.com","apellido_materno":"x1@x1.com","fecha_nacimiento":"2025-12-01T06:45:00","telefono":1111111111,"genero":"M","foto":null}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

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

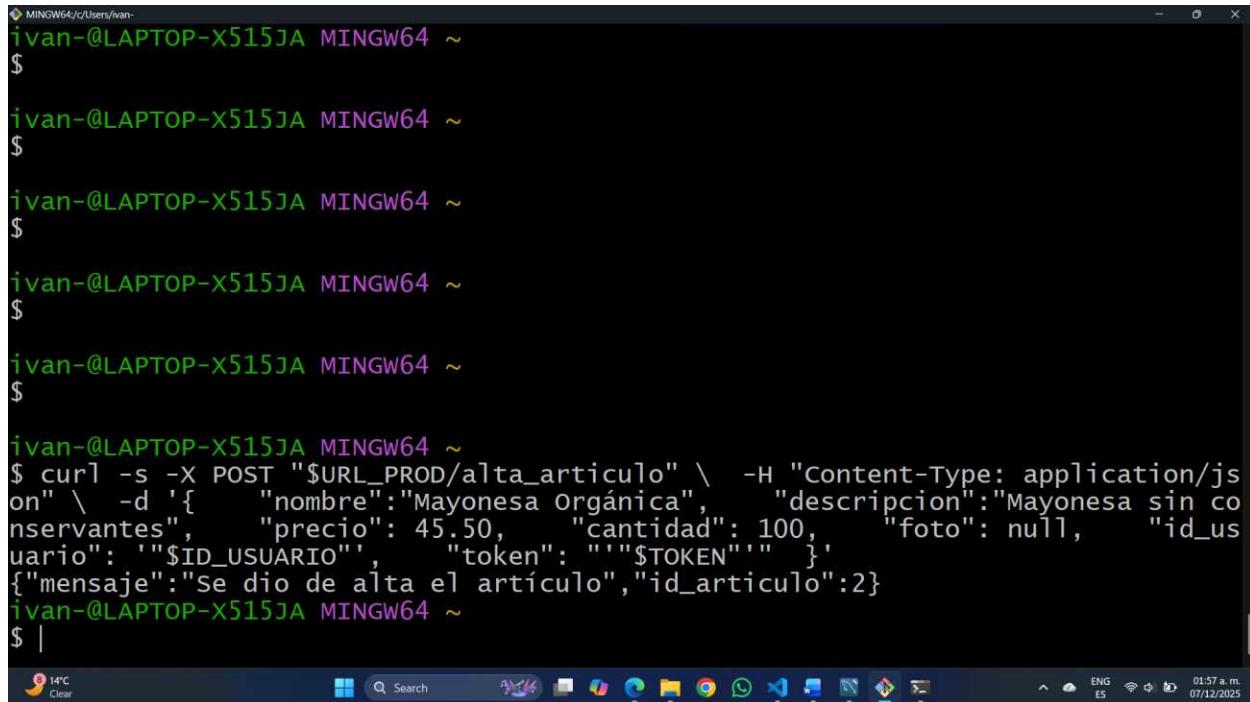
Figura 21. Respuesta JSON con id_usuario y perfil

9.4.3 Requerimiento 1 (back-end): alta_articulo

- Descripción: Alta de artículos en stock y opcionalmente foto en fotos_articulos.

```
curl -s -X POST "$URL_PROD/alta_articulo" \ -H "Content-Type: application/json" \ -d '{ "nombre": "Mayonesa Orgánica", "descripcion": "Mayonesa sin conservantes", "precio": 45.50, "cantidad": 100, "foto": null, "id_usuario": "'"$ID_USUARIO"', "token": "'"$TOKEN"''" }'
```

- Evidencia:



A screenshot of a Windows desktop environment showing a terminal window. The terminal window has a dark background with white text. It displays a series of command-line entries starting with 'ivan-@LAPTOP-X515JA MINGW64 ~' followed by several '\$' symbols indicating command prompts. The user runs a 'curl' command with various options and a JSON payload. The payload describes a new article with a name ('Mayonesa Orgánica'), price (45.50), quantity (100), and a message ('Se dio de alta el artículo'). The command ends with a closing brace. The terminal window is positioned above a taskbar which contains icons for various applications like File Explorer, Edge, and FileZilla. The system tray shows the date and time (01:57 a.m. 07/12/2025) and battery status.

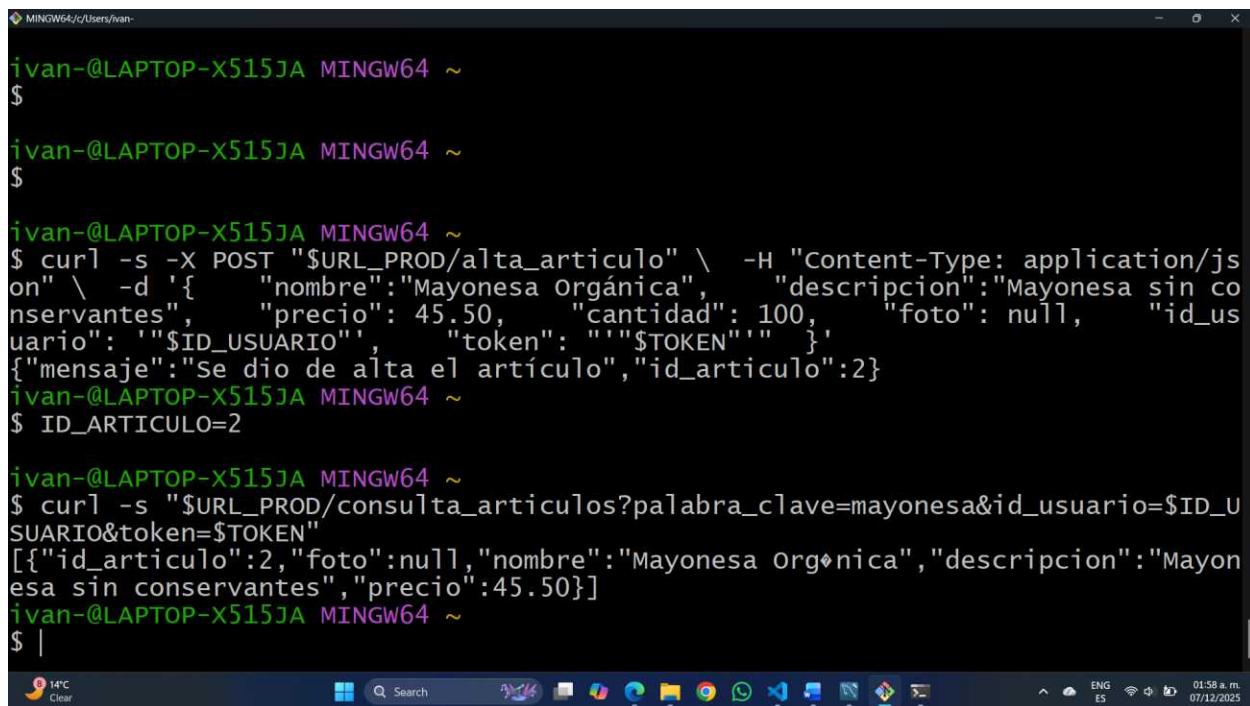
```
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ 
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/alta_articulo" \ -H "Content-Type: application/json" \ -d '{   "nombre":"Mayonesa Orgánica",   "descripcion":"Mayonesa sin co nservantes",   "precio": 45.50,   "cantidad": 100,   "foto": null,   "id_us uario": "'.$ID_USUARIO'",   "token": "'.$TOKEN'"}' 
{"mensaje":"Se dio de alta el artículo","id_articulo":2}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Figura 22. Respuesta de alta de artículo con id_articulo

9.4.4 Requerimiento 2: consulta_articulos (búsqueda por palabra clave)

```
curl -s
"$URL_PROD/consulta_articulos?palabra_clave=mayonesa&id_usuario=$ID_USUARI O&token=$TOKEN"
```

- Evidencia:



```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/alta_articulo" \ -H "Content-Type: application/json" \ -d '{ "nombre":"Mayonesa Orgánica", "descripcion":"Mayonesa sin conservantes", "precio": 45.50, "cantidad": 100, "foto": null, "id_usuario": "'.$ID_USUARIO'", "token": "'".$TOKEN"' }' {"mensaje":"se dio de alta el artículo","id_articulo":2}
ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_ARTICULO=2

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s "$URL_PROD/consulta_articulos?palabra_clave=mayonesa&id_usuario=$ID_USUARIO&token=$TOKEN"
[{"id_articulo":2,"foto":null,"nombre":"Mayonesa Orgánica","descripcion":"Mayonesa sin conservantes","precio":45.50}]
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

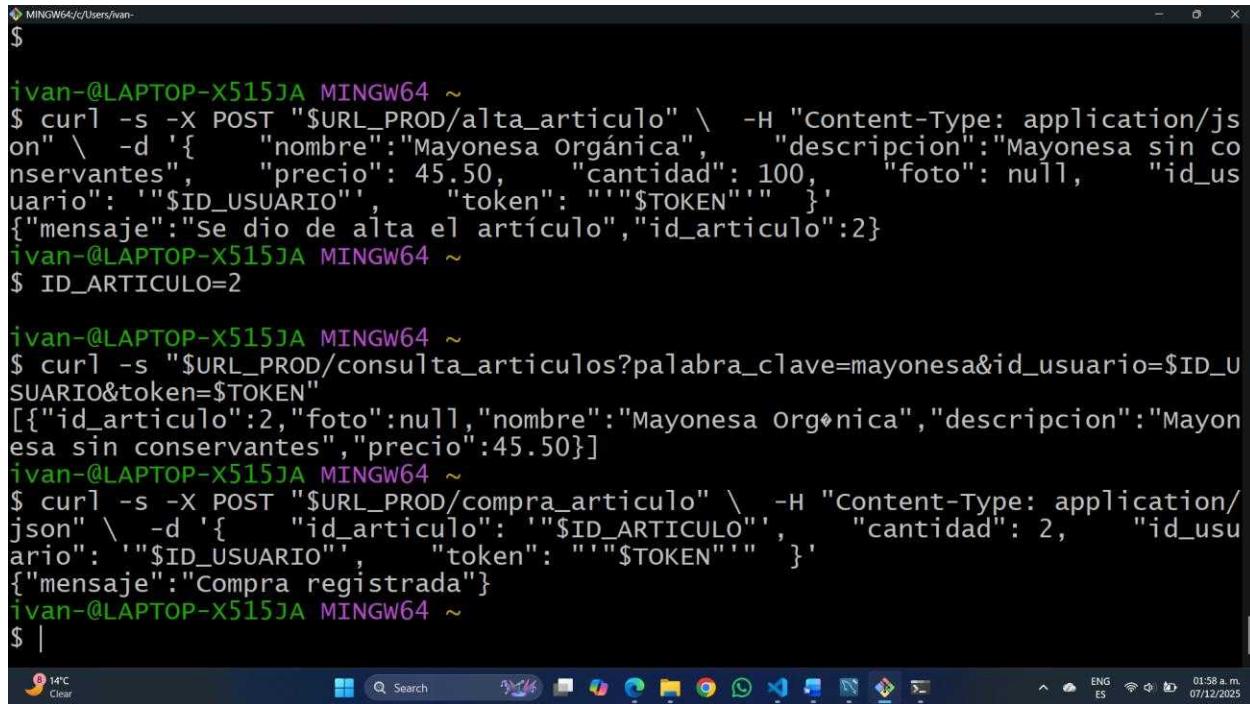
Figura 23. Respuesta JSON con lista de artículos filtrados por palabra clave

9.4.5 Requerimiento 3: compra_articulo

- Compra 2 unidades:

```
curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 2, "id_usuario": "'.$ID_USUARIO'", "token": "'".$TOKEN"' }'
```

- Evidencia:



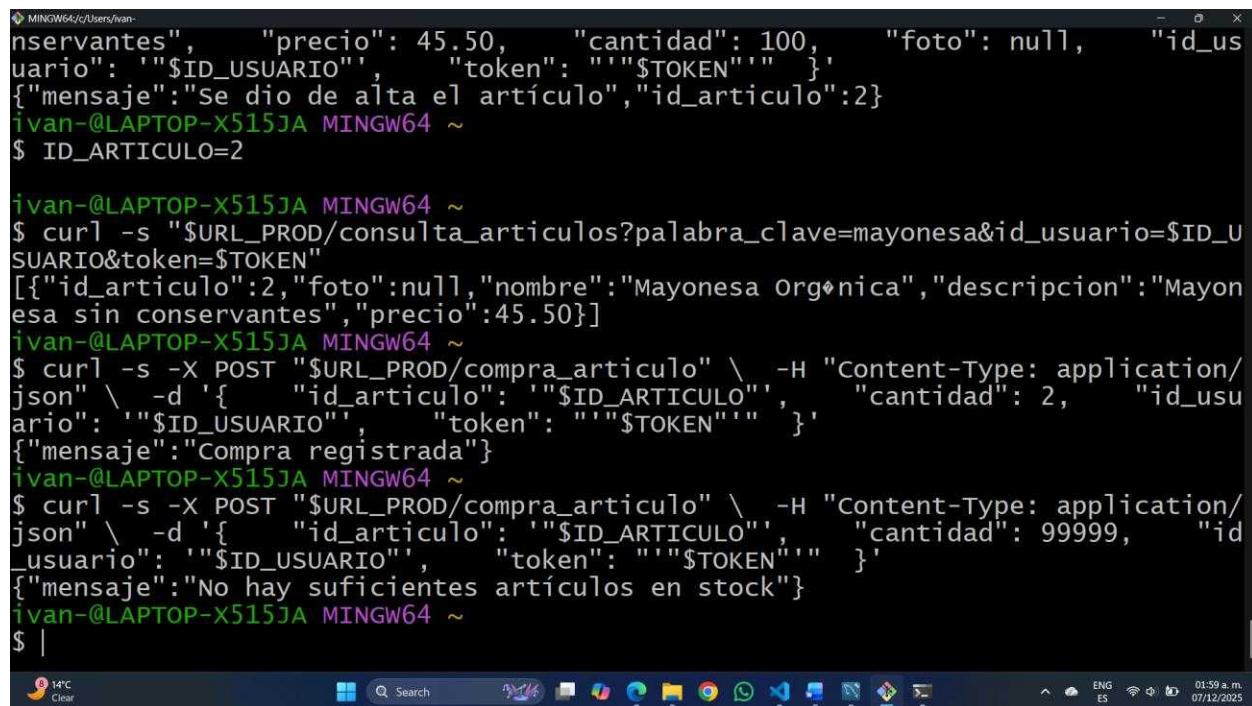
```
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/alta_articulo" \ -H "Content-Type: application/json" \ -d '{ "nombre":"Mayonesa Orgánica", "descripcion":"Mayonesa sin conservantes", "precio": 45.50, "cantidad": 100, "foto": null, "id_usuario": "'.$ID_USUARIO'", "token": "'".$TOKEN"" }'
{"mensaje":"Se dio de alta el artículo","id_articulo":2}
ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_ARTICULO=2

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s "$URL_PROD/consulta_articulos?palabra_clave=mayonesa&id_usuario=$ID_USUARIO&token=$TOKEN"
[{"id_articulo":2,"foto":null,"nombre":"Mayonesa Orgánica","descripcion":"Mayonesa sin conservantes","precio":45.50}]
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 2, "id_usuario": "'.$ID_USUARIO'", "token": "'".$TOKEN"" }'
{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |
```

Figura 24. Respuesta de compra registrada

- Error de stock insuficiente (ejemplo compra de 99999 unidades):

```
curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \
-d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 99999, "id_usuario": "'.$ID_USUARIO'", "token": "'".$TOKEN"" }'
```



```

MINGW64/c/Users/ivan-
nservantes", "precio": 45.50, "cantidad": 100, "foto": null, "id_us
uario": "$ID_USUARIO", "token": "$TOKEN"}'
>{"mensaje":"Se dio de alta el artículo","id_articulo":2}
ivan-@LAPTOP-X515JA MINGW64 ~
$ ID_ARTICULO=2

ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s "$URL_PROD/consulta_articulos?palabra_clave=mayonesa&id_usuario=$ID_U
SUARIO&token=$TOKEN"
[{"id_articulo":2,"foto":null,"nombre":"Mayonesa Orgánica","descripcion":"Mayon
esa sin conservantes","precio":45.50}]
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \
-H "Content-Type: application/
json" \
-d '{
    "id_articulo": "$ID_ARTICULO",
    "cantidad": 2,
    "id_usu
ario": "$ID_USUARIO",
    "token": "$TOKEN"}'
>{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \
-H "Content-Type: application/
json" \
-d '{
    "id_articulo": "$ID_ARTICULO",
    "cantidad": 99999,
    "id_
usuario": "$ID_USUARIO",
    "token": "$TOKEN"}'
>{"mensaje":"No hay suficientes artículos en stock"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |

```

Figura 25. Error de stock insuficiente (HTTP 400 con mensaje)

9.4.6 Requerimiento 4: elimina_articulo_carrito_compra

```

curl           -s           -X           DELETE
"$URL_PROD/elimina_articulo_carrito_compra?id_usuario=$ID_USUARIO&id_articulo
=$ID_ARTICULO&token=$TOKEN"

```

- Evidencia:

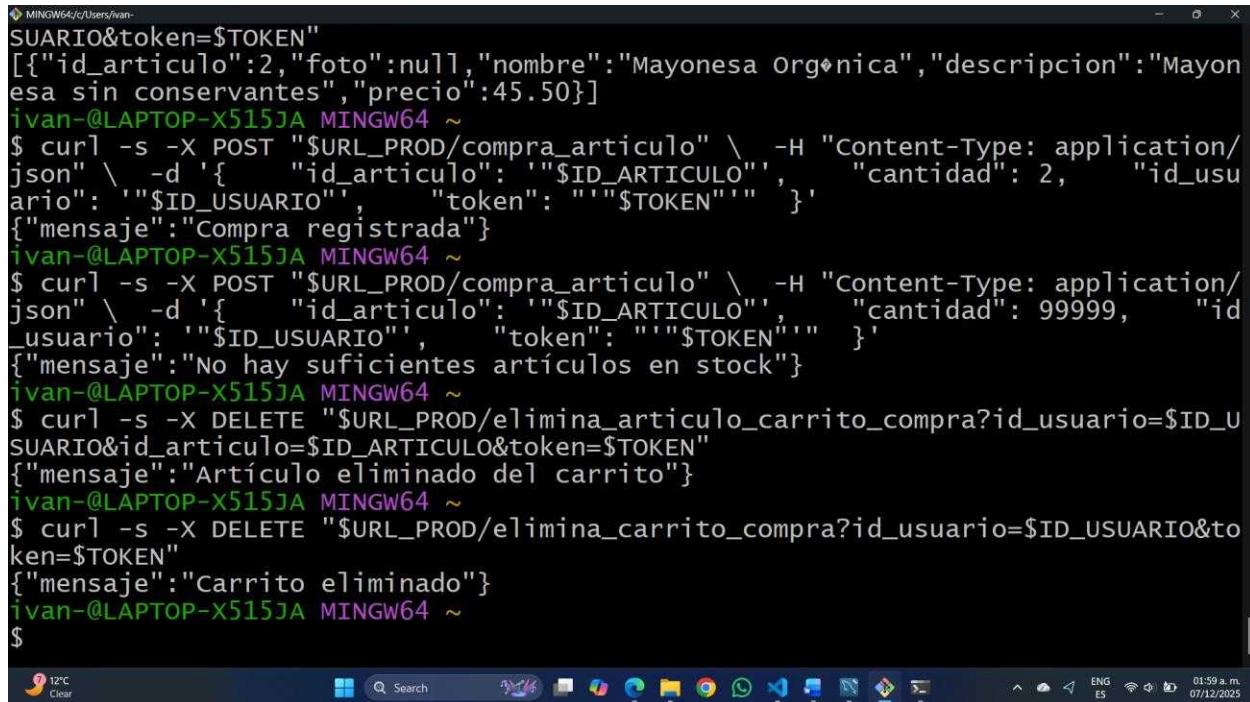
```
MINGW64:/c/Users/ivan-  
$ ID_ARTICULO=2  
  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -s "$URL_PROD/consulta_articulos?palabra_clave=mayonesa&id_usuario=$ID_USUARIO&token=$TOKEN"  
[{"id_articulo":2,"foto":null,"nombre":"Mayonesa Orgánica","descripcion":"Mayonesa sin conservantes","precio":45.50}]  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'"$ID_ARTICULO"'", "cantidad": 2, "id_usuario": "'"$ID_USUARIO"'", "token": "'"$TOKEN"'"}'  
{"mensaje":"Compra registrada"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'"$ID_ARTICULO"'", "cantidad": 99999, "id_usuario": "'"$ID_USUARIO"'", "token": "'"$TOKEN"'"}'  
{"mensaje":"No hay suficientes artículos en stock"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ curl -s -X DELETE "$URL_PROD/elimina_articulo_carrito_compra?id_usuario=$ID_USUARIO&id_articulo=$ID_ARTICULO&token=$TOKEN"  
{"mensaje":"Artículo eliminado del carrito"}  
ivan-@LAPTOP-X515JA MINGW64 ~  
$ |
```

Figura 26. Respuesta de eliminación de artículo del carrito

9.4.7 Requerimiento 5: elimina_carrito_compra

```
curl           -s           -X           DELETE  
"$URL_PROD/elimina_carrito_compra?id_usuario=$ID_USUARIO&token=$TOKEN"
```

- Evidencia:



```

MINGW64/c/Users/ivan-
SUARIO&token=$TOKEN"
[{"id_articulo":2,"foto":null,"nombre":"Mayonesa Orgánica","descripcion":"Mayonesa sin conservantes","precio":45.50}]
ivan@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 2, "id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
{"mensaje":"Compra registrada"}
ivan@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 99999, "id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
{"mensaje":"No hay suficientes artículos en stock"}
ivan@LAPTOP-X515JA MINGW64 ~
$ curl -s -X DELETE "$URL_PROD/elimina_articulo_carrito_compra?id_usuario=$ID_USUARIO&id_articulo=$ID_ARTICULO&token=$TOKEN"
{"mensaje":"Artículo eliminado del carrito"}
ivan@LAPTOP-X515JA MINGW64 ~
$ curl -s -X DELETE "$URL_PROD/elimina_carrito_compra?id_usuario=$ID_USUARIO&token=$TOKEN"
{"mensaje":"Carrito eliminado"}
ivan@LAPTOP-X515JA MINGW64 ~
$
```

Figura 27. Respuesta de eliminación completa del carrito

9.4.8 Requerimiento 6: modifica_carrito_compra (+1/-1)

- Prepara el carrito con una compra mínima:

```
curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/json" \
-d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 1, "id_usuario": "'.$ID_USUARIO'", \
"token": "'.$TOKEN'"}'
```

- Incremento (+1):

```
curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type: application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": 1, \
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
```

```

MINGW64/c/Users/ivan-
json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 99999, "id_
_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
{"mensaje":"No hay suficientes artículos en stock"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X DELETE "$URL_PROD/elimina_articulo_carrito_compra?id_usuario=$ID_U
SUARIO&id_articulo=$ID_ARTICULO&token=$TOKEN"
>{"mensaje":"Artículo eliminado del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X DELETE "$URL_PROD/elimina_carrito_compra?id_usuario=$ID_USUARIO&to
ken=$TOKEN"
>{"mensaje":"Carrito eliminado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/
json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 2, "id_usu
ario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
>{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type: appli
cation/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": 1,
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'
>{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ |

```

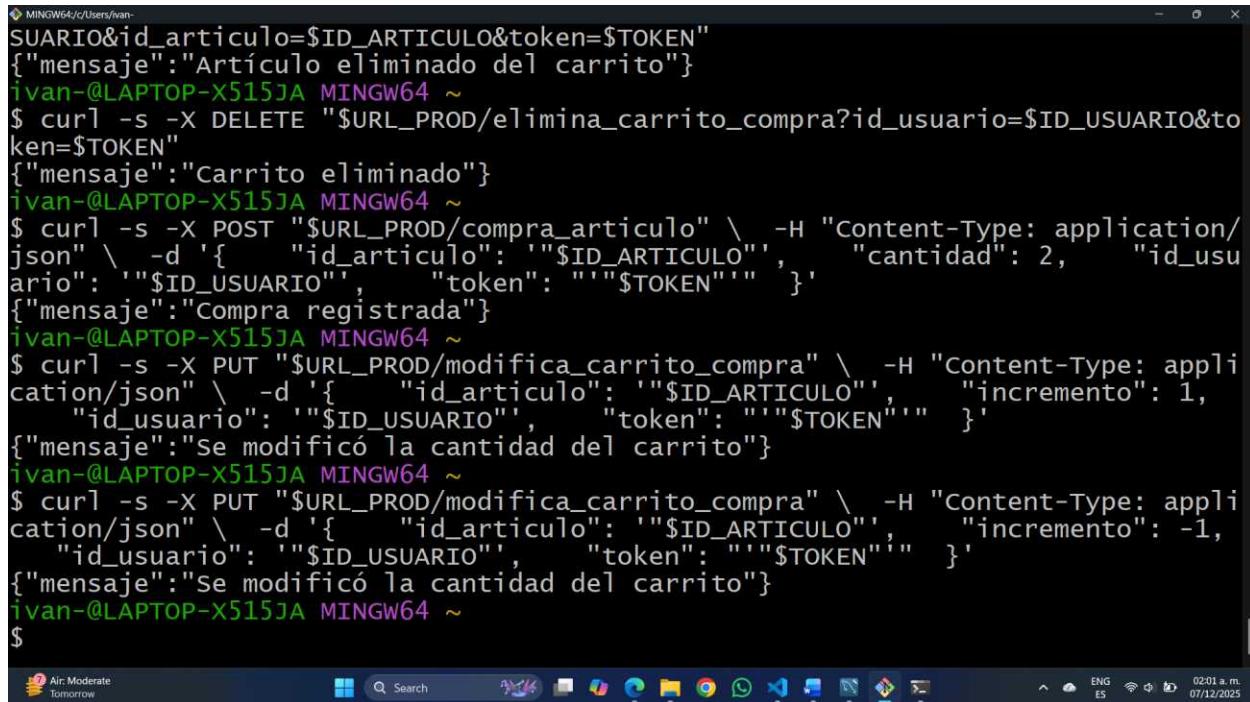
Figura 28. Respuesta de modificación +1

- Decremento (-1):

```

curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type:
application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": -1,
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN'"}'

```



```
MINGW64/c/Users/ivan-
SUARIO&id_articulo=$ID_ARTICULO&token=$TOKEN"
{"mensaje":"Artículo eliminado del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X DELETE "$URL_PROD/elimina_carrito_compra?id_usuario=$ID_USUARIO&to
ken=$TOKEN"
{"mensaje":"Carrito eliminado"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X POST "$URL_PROD/compra_articulo" \ -H "Content-Type: application/
json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "cantidad": 2, "id_usu
ario": "'.$ID_USUARIO'", "token": "'.$TOKEN.'" }'
{"mensaje":"Compra registrada"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type: appli
cation/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": 1,
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN.'" }'
{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$ curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type: appli
cation/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": -1,
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN.'" }'
{"mensaje":"Se modificó la cantidad del carrito"}
ivan-@LAPTOP-X515JA MINGW64 ~
$
```

Figura 29. Respuesta de modificación -1

- Error “No hay suficientes artículos en stock” (si stock = 0 y se intenta +1):

```
curl -s -X PUT "$URL_PROD/modifica_carrito_compra" \ -H "Content-Type:
application/json" \ -d '{ "id_articulo": "'.$ID_ARTICULO'", "incremento": 1,
"id_usuario": "'.$ID_USUARIO'", "token": "'.$TOKEN.'" }'
```