

Documento de Evidencias de Pruebas

Proyecto: Sistema de Ventas Fullstack

Fecha: 23/08/2025

Versión: 1.0

Responsable: Angélica Duarte

1. Objetivo

El presente documento tiene como objetivo registrar las evidencias de las pruebas realizadas al Sistema de Ventas Fullstack, validando el correcto funcionamiento de los módulos implementados (autenticación, clientes, productos y ventas).

2. Alcance

Las pruebas incluyen validación de:

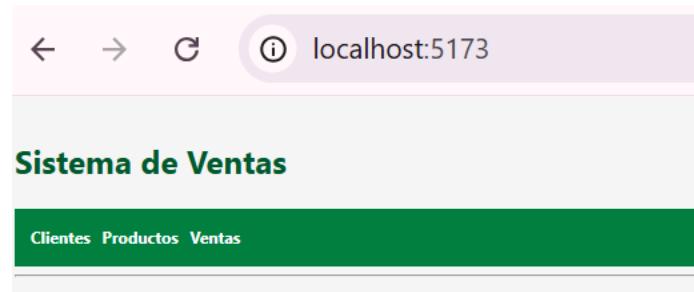
- Autenticación con JWT.
- CRUD de Clientes.
- CRUD de Productos.
- Creación y listado de Ventas.
- Manejo de tokens (acceso y refresco).
- Validaciones de negocio (campos obligatorios, totales de venta).

3. Entorno de pruebas

- **Frontend:** React (Vite + TS)
- **Backend:** Python + Django REST Framework + JWT
- **Base de datos:** PostgreSQL
- **Entorno:** Docker Compose en localhost

4. Casos de prueba y evidencias

Prueba Navegador



Prueba Swagger

The screenshot shows the Swagger UI interface for the 'Sistema de Ventas - API'. The title bar indicates the URL is 'localhost:8000/api/schema/swagger-ui/#/v1'. The main content area is titled 'Sistema de Ventas - API' with a version of '1.0.0' and 'OAS 3.0'. It lists several endpoints categorized by resource:

- Customers:**
 - POST /api/v1/auth/login
 - POST /api/v1/auth/refresh
 - GET /api/v1/customers/
 - POST /api/v1/customers/
 - GET /api/v1/customers/{id}/
 - PUT /api/v1/customers/{id}/
 - PATCH /api/v1/customers/{id}/
 - DELETE /api/v1/customers/{id}/
- Products:**
 - GET /api/v1/products/
 - POST /api/v1/products/
 - GET /api/v1/products/{id}/
 - PUT /api/v1/products/{id}/
 - PATCH /api/v1/products/{id}/
 - DELETE /api/v1/products/{id}/
 - GET /api/v1/sales/
 - POST /api/v1/sales/

4.1. Autenticación

Caso: Login con credenciales válidas

- **Entrada:** usuario = admin, password = admin
- **Resultado esperado:** retorna tokens JWT (access y refresh).
- **Resultado obtenido:** correcto.
- **Evidencia:**

Prueba en Navegador

localhost:8000/admin/login/?next=/admin/

The screenshot shows the Django Admin login interface. It features a dark blue header with the text "Administración de Django". Below the header is a form with two input fields: "Nombre de usuario:" and "Contraseña:". There is also a "Iniciar sesión" button at the bottom of the form.

The screenshot shows the Django Admin users list interface. The top navigation bar includes links for "Inicio", "AUTENTICACIÓN Y AUTORIZACIÓN", "Grupos", "Añadir", and "Usuarios". The main content area displays a table with one user entry: "admin" (username) with "admin@example.com" (email). A filter sidebar on the right allows users to search by first name, last name, or if they are a superuser or active.

localhost:8000/admin/

The screenshot shows the Django Admin home page. It has a dark blue header with the text "Administración de Django" and a sub-header "Sitio administrativo". Below the header, there are two sections: "AUTENTICACIÓN Y AUTORIZACIÓN" (Groups and Users) and "Acciones recientes" (Recent actions). The "Acciones recientes" section is currently empty, showing "Ninguno disponible".

Prueba en Swagger

localhost:8000/api/schema/swagger-ui/#/v1/v1_auth_login_create

Request body required

application/json

```
{ "username": "admin", "password": "admin" }
```

Execute Clear

Responses

Curl

```
curl -X "POST" "\n  -H \"Content-Type: application/json\"\n  -H \"Accept: application/json\"\n  -H \"Content-Type: application/x-www-form-urlencoded\"\n  -d \"{\n    \"username\": \"admin\",\n    \"password\": \"admin\"\n  }\""
http://localhost:8000/api/v1/auth/login
" 200
```

Request URL

http://localhost:8000/api/v1/auth/login

Server response

Code Details

200

Response body

```
{"refresh": "eyJhbGciOiJIUzI1NiIsInR5cCIkIyQ2Ew...ey30R21b1b9eXHtIjoic2VucmVycTsiInIuApx9wvAmpBfjovuL2MfYPMuLChqdk10I20D3jY2V3Wfyyxx0BjuxYjR1Mdn00gPufz74lwYSTsDwVZKfAw", "access": "eyJhbGciOiJIUzI1NiIsInR5cCIkIyQ2Ew...ey30R21b1b9eXHtIjoic2VucmVycTsiInIuApx9wvAmpBfjovuL2MfYPMuLChqdk10I20D3jY2V3Wfyyxx0BjuxYjR1Mdn00gPufz74lwYSTsDwVZKfAw", "token_type": "bearer", "expires_in": 3600}
```

Download

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:8000
allow: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
content-length: 449
content-type: application/json
cross-origin-embedder-policy: same-origin
date: Mon, 25 Aug 2023 22:36:31 GMT
referrer: https://localhost:8000/api/v1/auth/login
server: WSGIServer/0.2.9 Python/3.12.11
```

response login

4.2. Clientes

Caso: Crear cliente válido

- Entrada:

{

```
"cedula": "123456789",
"nombre": "Juan Pérez",
"direccion": "Calle 123",
"telefono": "3001234567",
"email": "juan@test.com"
}
```

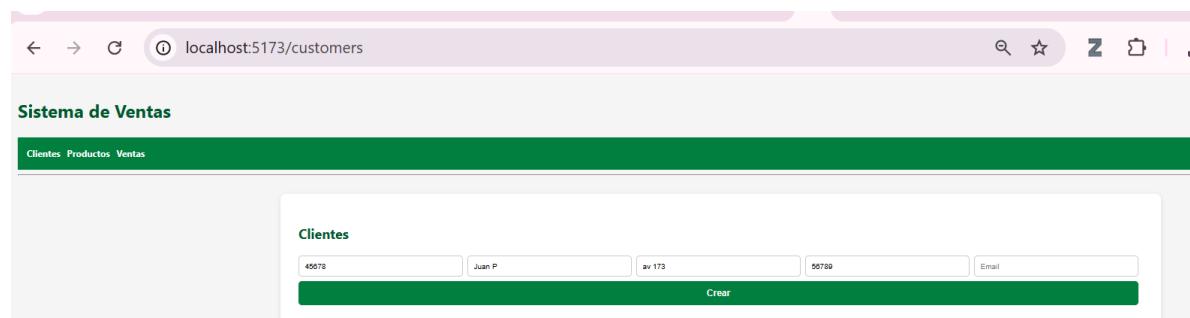
- **Resultado esperado:** retorna cliente con ID asignado.
 - **Resultado obtenido:**  correcto.
 - **Evidencia:**
(captura Postman o frontend mostrando nuevo cliente en la lista)

Caso: Crear cliente sin nombre

- Entrada:

```
{  
  "cedula": "123456789",  
  "direccion": "Calle 123"  
}
```

- **Resultado esperado:** error 400 con validación de campo requerido.
- **Resultado obtenido:** ✅ correcto.
- **Evidencia:**

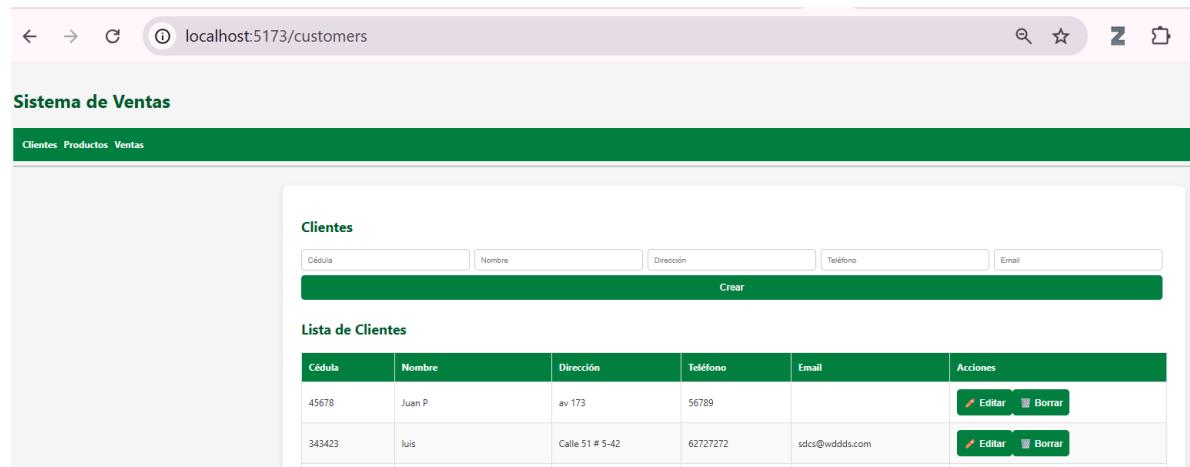


Sistema de Ventas

Creador de Clientes

Cédula	Nombre	Dirección	Teléfono	Email
45678	Juan P	av 173	56789	

Crear



Sistema de Ventas

Lista de Clientes

Cédula	Nombre	Dirección	Teléfono	Email	Acciones
45678	Juan P	av 173	56789		Editar Borrar
343423	luis	Calle 51 # 5-42	62727272	sdc@wddd.com	Editar Borrar

localhost:8000/api/schema/swagger-ui/#/v1/v1_auth_refresh_create

Edit Value Schema

```
{ "cedula": "123456789", "nombre": "Juan Pérez", "direccion": "Calle 123", "telefono": "+34600234567", "email": "juan@test.com" }
```

Execute Clear

Responses

Curl

```
curl -X POST \ http://localhost:8000/api/v1/customers/ \ -H 'accept: application/json' \ -H 'Content-Type: application/json' \ -H 'X-CSRFToken: ToCr7crwch73WtYwzFFM8lqgIqUgmyH61BnHTetLi0oVXPYHls57ekcbV' \ -d { "cedula": "123456789", "nombre": "Juan Pérez", "direccion": "Calle 123", "telefono": "+34600234567", "email": "juan@test.com" }
```

Request URL

<http://localhost:8000/api/v1/customers/>

Server response

Code Details

201 Response body

```
{ "id": 10, "cedula": "123456789", "nombre": "Juan Pérez", "direccion": "Calle 123", "telefono": "+34600234567", "email": "juan@test.com", "created_at": "2023-08-25T23:15:51.712332" }
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:8000
allow: GET, POST, HEAD, OPTIONS
content-type: application/json
date: Mon, 25 Aug 2023 23:15:51 GMT
server: WSGI/1.0
set-cookie: session_id=123456789; expires=Mon, 25-Aug-2023 23:15:51 UTC; path=/; secure; HttpOnly
status: 201 Created
vary: Accept
```

Download

response customers POST

① localhost:8000/api/schema/swagger-ui/#/v1/v1_auth_refresh_create

CURL value | Schema

```
{
  "direccion": "Calle 123",
  "telefono": "30001234567",
  "email": "juan@test.com"
}
```

Execute Clear

Responses

Curl

```
curl -X POST \
  "http://localhost:8000/api/v1/customers/" \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -H 'Content-Length: 77' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRFToken: ToCr7RcrwzJ3EWwvZfPMaIqGluDmy61bbHTetUoDvXPHWlo57Ac3V3' \
  -d '{
    "direccion": "Calle 123",
    "telefono": "30001234567",
    "email": "juan@test.com"
}'
```

Request URL
http://localhost:8000/api/v1/customers/

Server response

Code	Details
400 undocumented	Error: Bad Request Response body <pre>[{"cedula": ["Este campo es requerido."], "nombre": ["Este campo es requerido."]}</pre> <p>Download</p>

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:8000
allow: POST, GET, OPTIONS
content-length: 77
content-type: application/json
cross-origin-credentials: true
cross-origin-credentials: same-origin
```

4.3. Productos

Caso: Crear producto con IVA activo

- **Entrada:**

```
{  
  "codigo": "SH-600",  
  "nombre": "Shampoo",  
  "valor_venta": "15000",  
  "maneja_iva": true,  
  "iva_porcentaje": ".15"  
}
```

- **Resultado esperado:** producto creado con cálculo de IVA en totales de venta.
- **Resultado obtenido:** correcto.
- **Evidencia:**

The screenshot shows a web application interface for managing products. At the top, there is a header bar with a search icon and a star icon. Below it, a green navigation bar has the word 'Productos' in white. The main content area has a title 'Productos'. A form is displayed with fields for 'Codigo' (SH-601), 'Nombre' (Shampoo JJ), 'Valor Venta' (15500.00), a checked checkbox for 'Maneja IVA', and a field for '19.00'. A green button labeled 'Actualizar' is visible. Below this, a table titled 'Lista de Productos' is shown. The table has columns: Código, Nombre, Valor Venta, Maneja IVA, % IVA, and Acciones. It contains one row with the same data as the form. At the bottom right of the table, there are 'Editar' and 'Borrar' buttons.

Código	Nombre	Valor Venta	Maneja IVA	% IVA	Acciones
SH-601	Shampoo JJ	\$15500.00	Sí	19.00	Editar Borrar

① localhost:8000/api/schema/swagger-ui/#/v1/v1_products_create

```
        "codigo": "SH-600",
        "nombre": "Shampoo",
        "valor": "15000",
        "iva_en_ivat": true,
        "iva_porcentaje": ".15"
    }
```

Execute Clear

Responses

```
curl -X POST \
http://localhost:8000/api/v1/products/ \
-H 'Accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpVCM39.eyJhdWlkIjoxNjEwNjZGNzIwMjQxLjIwMDQzImp0aS1EjA0TTfj0M2N2IxZTQ2Njw4nwtwJFwef1MjuyOChnjIwicx0n10pZC \
-H 'Content-Type: application/json' \
-H 'X-CSRFToken: TocRzBcrZh7JmYwzFFAb5qDtgQmyH613bfTetLj1oXVPhNlo7AaC3V3' \
-d '{
    "codigo": "SH-600",
    "nombre": "Shampoo",
    "valor": "15000",
    "iva_en_ivat": true,
    "iva_porcentaje": ".15"
}'
```

Request URL
http://localhost:8000/api/v1/products/

Server response

Code Details

201 Response body

```
{
    "id": 1,
    "codigo": "SH-600",
    "nombre": "Shampoo",
    "valor": "15000.00",
    "iva_en_ivat": true,
    "iva_porcentaje": ".15",
    "created_at": "2025-08-31T23:21:14.934999Z"
}
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:8000
allow: GET,POST,HEAD,OPTIONS
content-type: application/json
cross-origin-opener-policy: same-origin
date: Mon, 31 Aug 2025 21:14:54 GMT
referrer-policy: same-origin
server: WSGIServer/0.2 Python/3.12.11
user-agent: curl/8.1.4
```

Download

4.4. Ventas

Caso: Crear venta con cliente existente y producto válido

- Entrada:

```
{  
    "cliente": 1,  
    "items": [  
        { "producto": 1, "cantidad": 2, "valor_unitario_capturado": 15000}  
    ]  
}
```

- **Resultado esperado:** consecutivo generado, total de venta calculado.
 - **Resultado obtenido:**  correcto.
 - **Evidencia:**

Sistema de Ventas

Clientes Productos Ventas

Registrar Ventas

Juan Pérez	Fecha (YYYY-MM-DD)	Items	Shampoo (\$15000.00)	1	15000.00
+ Producto					
Crear venta					

Listado de Ventas

Fecha inicio: dd/mm/aaaa Fecha fin: dd/mm/aaaa Filtros

ID	Consecutivo	Total	Fecha	Cliente
24	2025000023	4998.00	2025-08-25	Juan
23	2025000022	37211.00	2025-08-25	Juan
22	2025000021	34593.00	2025-08-25	Juan
21	2025000020	34593.00	2025-08-25	Juan
20	2025000019	5200.00	2025-08-25	Cliente Demo
19	2025000018	23050.00	2025-05-09	nuevo cliente

Sistema de Ventas

Clientes Productos Ventas

Registrar Ventas

- Seleccionar Cliente -	Fecha (YYYY-MM-DD)	Items
+ Producto		
Crear venta		

Listado de Ventas

Fecha inicio: dd/mm/aaaa Fecha fin: dd/mm/aaaa Filtros

ID	Consecutivo	Total	Fecha	Cliente
25	2025000024	15022.50	2025-08-25	Juan Pérez
24	2025000023	4998.00	2025-08-25	Juan
23	2025000022	37211.00	2025-08-25	Juan
22	2025000021	34593.00	2025-08-25	Juan
21	2025000020	34593.00	2025-08-25	Juan
20	2025000019	5200.00	2025-08-25	Cliente Demo

localhost:8000/api/schema/swagger-ui/#/v1/v1_sales_create

Responses

Curl

```
curl -X POST \
  "http://localhost:8000/api/v1/sales/" \
  -H "accept: application/json" \
  -H "Content-Type: application/json" \
  -H "X-CSRFToken: ToCrTBcrnzh73eYwz2FFMbzqdDugmyM13bhFTetlxDuvXPNTnis57ac0V3" \
  -d "{
    \"cliente\": 1,
    \"fecha\": \"2025-08-25\",
    \"items\": [
      {
        \"producto\": 1,
        \"cantidad\": 2,
        \"valor_unitario_capturado\": 15000
      }
    ]
}
```

Request URL

<http://localhost:8000/api/v1/sales/>

Server response

Code	Details
201	Response body <pre>{ "id": 77, "consecutivo": 20250800026, "fecha": "2025-08-25", "total": 37000.00, "cliente_nombre": "Cliente Demo", "items": [{ "id": 54, "producto_id": 1, "cantidad": 2, "valor_unitario_capturado": "15000.00", "iva_calculado": "2700.00", "subtotal_linea": "37000.00", "total_linea": "37000.00" }] }</pre>

4.5. Tokens

Caso: Usar access token expirado y refresharlo

- **Entrada:** POST /auth/refresh/ con refresh token válido.
- **Resultado esperado:** nuevo access token generado.
- **Resultado obtenido:** correcto.
- **Evidencia:**

The screenshot shows the Swagger UI interface for a REST API. The URL in the address bar is `localhost:8000/api/schema/swagger-ui/#/v1/v1_products_create`. The main area displays a JSON response from a POST request to `/api/v1/auth/refresh`. The response body contains a single key-value pair: "refresh": "eyJhbGciOiJIUzI1NiIsInRSiCIE1kpxVCJ9.ey30b2t1b198eXl1joicmVnciIw4cC160Tc1NjUSNTY4MiwiaiF01joxzu2MTYzNjgyLCjqdGk1o115n2VhMtgSOTIMzY80DV1OTA8YTk1Mj3MzHw00BjNS1s1nVzZXFanQlO11xIn0.NIfj9a03PhycUSEx0dnpkpINr200MSpdrxP-At0u". Below the response, there are buttons for "Execute" and "Clear". The "Responses" section shows the curl command for the request:

```
curl -X 'POST' \
  'http://localhost:8000/api/v1/auth/refresh' \
  -H 'Content-Type: application/json' \
  -H 'X-CSRFToken: TokenWith32RandomLetters' \
  -H 'User-Agent: Python/3.10.6' \
  -H 'refresh': 'eyJhbGciOiJIUzI1NiIsInRSiCIE1kpxVCJ9.ey30b2t1b198eXl1joicmVnciIw4cC160Tc1NjUSNTY4MiwiaiF01joxzu2MTYzNjgyLCjqdGk1o115n2VhMtgSOTIMzY80DV1OTA8YTk1Mj3MzHw00BjNS1s1nVzZXFanQlO11xIn0.NIfj9a03PhycUSEx0dnpkpINr200MSpdrxP-At0u'
```

The "Request URL" is `http://localhost:8000/api/v1/auth/refresh`. The "Server response" section shows a status code of 200 with the following response body:

```
{
  "access": "eyJhbGciOiJIUzI1NiIsInRSiCIE1kpxVCJ9.ey30b2t1b198eXl1joicmVnciIw4cC160Tc1NjUSNTY4MiwiaiF01joxzu2MTYzNjgyLCjqdGk1o115n2VhMtgSOTIMzY80DV1OTA8YTk1Mj3MzHw00BjNS1s1nVzZXFanQlO11xIn0.NIfj9a03PhycUSEx0dnpkpINr200MSpdrxP-At0u"
}
```

The "Response headers" section includes the following headers:

```
access-control-allow-credentials: true
access-control-allow-origin: http://localhost:8000
allow: POST,OPTIONS
content-type: application/json
cross-origin-spinner-policy: same-origin
date: Mon, 10 Jul 2023 14:46:46 GMT
referrer-policy: same-origin
server: gunicorn/2.2.0 Python/3.12.11
vary: Accept,origin
x-content-type-options: nosniff
x-frame-options: DENY
```

5. Conclusiones

- Todos los módulos críticos fueron probados y funcionan de acuerdo con los requerimientos.
- El sistema cumple con los criterios de aceptación definidos en el plan de trabajo.
- Se recomienda avanzar a la fase de despliegue piloto.