

API Directory – POS

Base URL (local): `http://localhost:4000`

Formato: JSON

Auth: Bearer JWT en header

`Authorization: Bearer <token>`

1) Auth

POST `/auth/login` (sin auth)

Body:

```
{ "email": "admin@tienda.com", "password": "123456" }
```

200 → { "token": "...", "user": { "id": 1, "name": "Admin", "email": "...", "role": "ADMIN" } }

401 → { "error": "Credenciales inválidas" }

GET `/auth/me`

Header: `Authorization: Bearer <token>`

200 → { "id": 1, "name": "...", "email": "...", "role": "ADMIN" }

2) Users

GET `/api/users`

200 → [{ "id":1,"name":"...","email":"...","role":"ADMIN","active":true }, ...]

POST `/api/users`

```
{ "name":"Juan", "email":"juan@x.com", "password":"123456", "role":"STAFF", "active":true }
```

201 → usuario creado

PUT `/api/users/:id`

```
{ "name":"Juan P", "role":"MANAGER", "active":false }
```

DELETE /api/users/:id

204

3) Customers

GET /api/customers

200 → [{ "id":1,"name":"Alicia","email":"...","phone":"..." }, ...]

POST /api/customers

```
{ "name":"Alicia", "email":"alicia@x.com", "phone":"555-123-4567" }
```

PUT /api/customers/:id

```
{ "phone":"555-000-1111" }
```

DELETE /api/customers/:id

204

4) Products

Campos: sku (único), name, price, stock, min, image_url (URL opcional)

GET /api/products

200 → [{ "id":1,"sku":"WPI-001","name":"Proteína","price":799,"stock":10,"min":3,"image_url":"" }, ...]

POST /api/products

```
{  
  "sku":"WPI-001","name":"Proteína Whey 1kg",  
  "price":799,"stock":10,"min":3,  
  "image_url":"https://..."  
}
```

201 → producto

PUT /api/products/:id

```
{ "price":749, "stock":12, "image_url":"/uploads/WPI-001.jpg" }
```

DELETE /api/products/:id

204

5) Stock (Ajustes + Movimientos)

GET /api/stock/movements

Query opcional: ?product_id=1

```
200 → [ {  
  "id":10,"product_id":1,"type":"IN","qty":5,"note":"Recepción","created_at":"2  
025-08-27 14:32:10","sku":"WPI-001","name":"Proteína" }, ... ]
```

POST /api/stock/adjust

Ajuste IN/OUT:

```
{ "product_id":1, "type":"IN", "qty":10, "note":"OC-123" }
```

```
200 → { "ok":true, "product": { ... stock actualizado ... } }
```

POST /api/stock/set

Fijar stock exacto:

```
{ "product_id":1, "stock":120, "note":"Conteo cíclico" }
```

6) Sales (Ventas)

Totales se calculan en servidor (IVA 16% por default en el demo).

GET /api/sales

```
200 → [ { "id":1,"folio":"V-0001","date":"2025-08-27  
12:03:11","customer":"Venta al  
público","method":"EFECTIVO","subtotal":249,"tax":39.84,"total":288.84 }, ...  
]
```

GET /api/sales/:id

200 →

```
{
  "id":1,"folio":"V-0001","date":"2025-08-27 12:03:11",
  "customer":"Venta al público","method":"EFECTIVO",
  "items":[ { "product_id":1,"sku":"WPI-001",
  "name":"Proteína","price":249,"qty":1,"total":249 } ],
  "subtotal":249,"tax":39.84,"total":288.84
}
```

POST /api/sales

Crea venta y **descuenta** stock:

```
{
  "customer_name":"Venta al público",
  "payment_method":"EFECTIVO",
  "items":[ { "product_id":1, "qty":2 }, { "product_id":2, "qty":1 } ]
}
```

201 → venta

PUT /api/sales/:id

Recalcula y ajusta stock diferencial:

```
{
  "customer_name":"Alicia Gómez",
  "payment_method":"TARJETA",
  "items":[ { "product_id":1, "qty":1 } ]
}
```

200 → venta actualizada

DELETE /api/sales/:id

Revierte stock y elimina venta.

204

Errores comunes

- **400** Validación: faltan campos, stock insuficiente, etc. → { "error": "mensaje" }
- **401** Sin token/expirado → { "error":"Unauthorized" }
- **404** No encontrado

- **409** Conflicto (por ejemplo, sku duplicado)