

1. AUTH — auth.py

Endpoint: `/login`

Método HTTP: POST

Formato Serialización: JSON

Cabecera entrada:

`Content-Type: application/json`

Estructura de datos IN:

```
{
  "nombre": "string",
  "apellido": "string",
  "contraseña": "string",
  "captcha_code": "string"
}
```

Estructura de datos OUT (200):

```
{
  "status": "ok",
  "user_id": 1
}
```

Estructura de datos ERR (400 / 401):

```
{
  "error": "Credencial invalida"
}
```

o

```
{
  "error": "Captcha Invalido"
}
```

Endpoint: **/logout**

Método HTTP: POST

Formato Serialización: JSON

Cabecera entrada: Ninguna

Estructura de datos IN: *(no aplica)*

Estructura de datos OUT (200):

```
{  
  "status": "logged_out"  
}
```

Estructura de datos ERR: *(no aplica)*

Endpoint: **/captcha/new**

Método HTTP: GET

Formato Serialización: JSON

Estructura de datos OUT (200):

```
{  
  "captcha": "AB123"  
}
```

Estructura de datos ERR: *(no aplica)*

2. HEALTH — health.py

Endpoint: **/health/app**

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
{  
  "app": "ok"  
}
```

Estructura ERR: *(no aplica)*

Endpoint: `/health/db`

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
{  
  "db": "ok"  
}
```

Estructura ERR (500):

```
{  
  "db": "down"  
}
```

3. FEED — feed.py

Endpoint: `/feed`

Método HTTP: GET

Tipo: HTML (NO REST)

→ No se documenta JSON.

Endpoint: `/feed/data`

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
[
  {
    "id": 1,
    "nombre": "Producto X",
    "precio": 2500,
    "image_url": "...",
    "vendor": "Panadería Sol"
  }
]
```

Estructura ERR (500):

```
{
  "error": "No se puede obtener la fuente"
}
```

4. CATALOG — catalog.py

Endpoint: `/catalog/create`

Método HTTP: POST

Formato Serialización: JSON o multipart/form-data

Cabecera entrada:

- JSON: `Content-Type: application/json`
- Form: `Content-Type: multipart/form-data`

Estructura de datos IN (JSON):

```
{
  "nombre": "string",
  "descripcion": "string",
  "precio": 1200,
  "imagen": "base64 | url | file",
  "vendor_id": 5
}
```

Estructura OUT (200 / 201):

```
{
  "status": "created",
  "item_id": 22
}
```

Estructura ERR (400):

```
{
  "error": "Missing fields"
}
```

Endpoint: **/catalog/<id>**

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
{
  "id": 22,
  "nombre": "Producto X",
  "descripcion": "Algo",
  "precio": 1200,
  "image_url": "...",
  "vendor_name": "Juan Pérez"
}
```

Estructura ERR (404):

```
{
```

```
"error": "Artículo no encontrado"
}
```

Endpoint: **/catalog/<id>/delete**

Método HTTP: DELETE

Formato Serialización: JSON

Estructura OUT (200):

```
{
  "status": "deleted"
}
```

Estructura ERR (404):

```
{
  "error": "Artículo no encontrado"
}
```

Endpoint: **/catalog/vendor/<vendor_id>**

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
[
  {
    "id": 5,
    "nombre": "Producto 1",
    "precio": 800
  }
]
```

Estructura ERR (404):

```
{
  "error": "Emprendedor no encontrado"
}
```

```
}
```

Endpoint: `/catalog/list`

Método HTTP: GET

Formato Serialización: JSON

Estructura OUT (200):

```
[  
  {  
    "id": 1,  
    "nombre": "Producto X",  
    "precio": 2000  
  }  
]
```

Estructura ERR: *(no aplica)*

5. VENDOR — vendor.py

Endpoint: `/complete_profile`

Método HTTP: GET

Tipo: *HTML — NO REST / NO JSON*

→ No recibe ni devuelve JSON.

→ Renderiza `profile.html`.

Endpoint: `/complete_profile`

Método HTTP: POST

Cabecera entrada:

Content-Type: application/json

Estructura de datos IN (POST):

```
{  
  "bio": "string",  
  "hubicaion": "string",  
  "avatar_url": "string"  
}
```

Estructura de datos OUT (200):

```
{  
  "status": "profile_created"  
}
```

Estructura de datos ERR (400):

```
{  
  "error": "No se pudo guardar el perfil"  
}
```

Endpoint: **/profile/<vendor_id>**

Método HTTP: GET

Formato Serialización: JSON

Cabecera entrada: *(no aplica)*

Estructura OUT (200):

```
{  
  "id": 8,  
  "nombre": "Juan",  
  "apellido": "Pérez",  
  "bio": "Panadero hace 20 años",  
  "ubicacion": "Córdoba",  
  "avatar_url": "https://example.com/avatar.png"  
}
```

Estructura ERR (404):

```
{  
  "error": "Emprendedor no encontrado"  
}
```