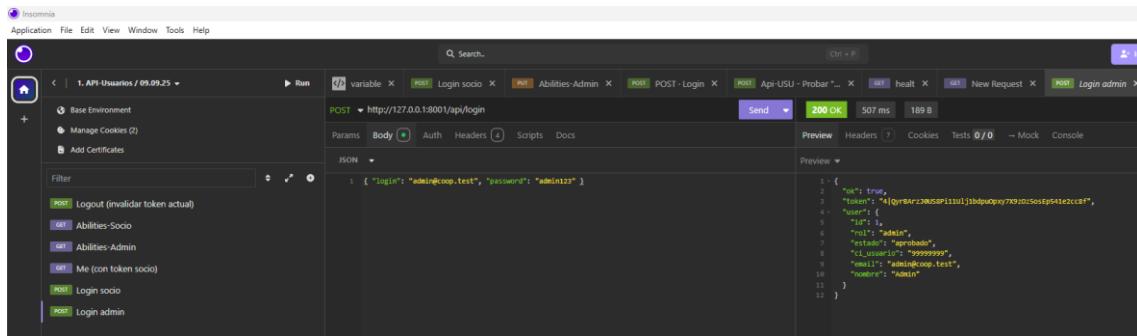


## API USUARIOS

Figura

Evidencia extraída de Insomnia - Prueba de inicio de sesión del usuario administrador en la API de Usuarios.



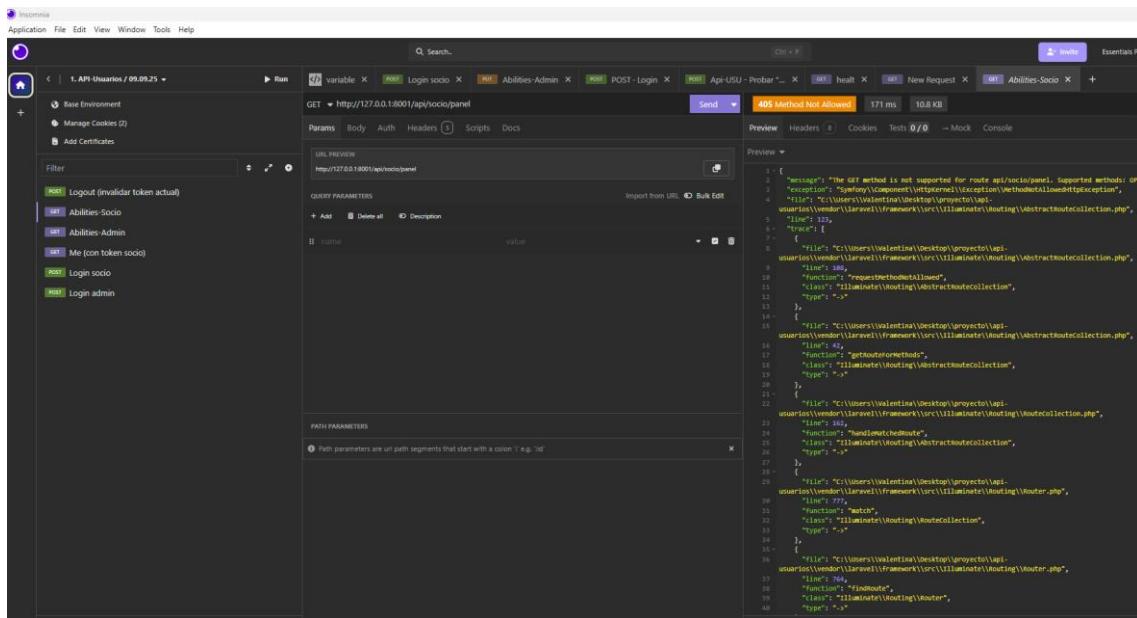
```
POST http://127.0.0.1:8001/api/login
{
    "login": "admin@coop.test",
    "password": "admin123"
}

200 OK
507 ms
189 B
{
    "id": 1,
    "token": "4jqyBarz3WUSP1lUJ1hpuopxy7X3j05oSp54t2ccB",
    "user": {
        "id": 1,
        "name": "admin",
        "role": "admin",
        "estado": "aprobado",
        "ci_usuario": "99999999",
        "email": "admin@coop.test",
        "telefono": "1234567890"
    }
}
```

Esta evidencia corresponde a la v1.0 donde login admin estaba implementada desde api-usuarios por error. En ese esquema socios y administradores estaban integrados, diferenciados por roles.

Figura

Evidencia extraída de Insomnia - Manejo de error por método no permitido – /api/socio/panel



```
GET http://127.0.0.1:8001/api/socio/panel
405 Method Not Allowed
171 ms
10.8 KB
{
    "message": "The GET method is not supported for route api/socio/panel. Supported methods: POST",
    "exception": "Symfony\\Component\\HttpKernel\\Exception\\MethodNotAllowedHttpException",
    "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\AbstractRouteCollection.php",
    "line": 125,
    "trace": [
        {
            "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\AbstractRouteCollection.php",
            "line": 126,
            "function": "requestMethodNotAllowed",
            "class": "Illuminate\\Routing\\AbstractRouteCollection",
            "type": ">"
        },
        {
            "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\AbstractRouteCollection.php",
            "line": 142,
            "function": "getRoutesForMethod",
            "class": "Illuminate\\Routing\\AbstractRouteCollection",
            "type": ">"
        },
        {
            "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\Router.php",
            "line": 160,
            "function": "handleMethodRoute",
            "class": "Illuminate\\Routing\\AbstractRouteCollection",
            "type": ">"
        },
        {
            "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\Router.php",
            "line": 77,
            "function": "match",
            "class": "Illuminate\\Routing\\AbstractRouteCollection",
            "type": ">"
        },
        {
            "file": "C:\\Users\\Valentina\\Desktop\\proyecto\\api\\usuarios\\vendor\\laravel\\framework\\src\\Illuminate\\Routing\\Router.php",
            "line": 76,
            "function": "findRoute",
            "class": "Illuminate\\Routing\\Router",
            "type": ">"
        }
    ]
}
```

Evidencia del control de errores ante un método HTTP incorrecto.

Se invoca el endpoint /api/socio/panel con método **GET** y la API responde **405 Method Not Allowed**, indicando que solo se admite **POST**. Esto demuestra la validación de métodos permitidos en las rutas.

## Figura

### Evidencia extraída de Insomnia - Login de usuario no aprobado (403 Forbidden)

The screenshot shows the Insomnia interface with a failed POST request to `/api/login`. The response status is **403 Forbidden** with the message "usuario no aprobado aún". The request body contains the JSON object: { "ok": false, "error": "usuario no aprobado aún.", "estado": "pendiente" }.

Evidencia de la lógica de negocio que bloquea usuarios pendientes.

En el entorno de pruebas v2 test api-usuarios se ejecuta POST /api/login con un usuario cuyo estado es *pendiente* y la API responde **403 Forbidden**, con el mensaje “usuario no aprobado aún”, confirmando que no puede acceder hasta ser aprobado por el backoffice.

## Figura

### Evidencia extraída de Insomnia - Actualización de perfil de socio – PUT /api/perfil

The screenshot shows a successful PUT request to `/api/perfil` with a response status of **200 OK**. The request body is a JSON object with the key `"segundo_nombre"` set to `"Pepe"`. The response body is a JSON object containing updated user data, including the new second name and other fields like email and role.

```
1: {
2:   "ok": true,
3:   "data": {
4:     "ci_usuario": "111111",
5:     "primer_nombre": "PRUEBA",
6:     "segundo_nombre": "Pepe",
7:     "primer_apellido": "Uno",
8:     "segundo_apellido": null,
9:     "email": "prueba@example.com",
10:    "rol": "socio",
11:    "estado_registro": "Aprobado",
12:    "created_at": "2025-09-07T14:51:52.000000Z",
13:    "updated_at": "2025-09-07T14:51:52.000000Z"
14:  }
15: }
```

Evidencia de actualización de datos de perfil mediante endpoint protegido.

Con un token válido se llama al endpoint PUT /api/perfil modificando campos del usuario (por ejemplo el segundo nombre) y la API responde **200 OK** devolviendo los datos actualizados, lo que confirma que la modificación se guarda correctamente en la base de datos.

## Figura

Evidencia extraída de Insomnia - Registro de nuevo socio – POST /api/register (solicitud pendiente)

The screenshot shows the Insomnia API client interface. The top navigation bar includes Application, File, Edit, View, Window, Tools, Help, and a search bar. Below the navigation is a tab bar with several tabs: variable, Login socio, Abilities-Admin, POST - Login, Api-USU - Probar..., health, New Request, and others. The active tab is 'POST - Login'. The main workspace shows a POST request to 'http://127.0.0.1:8001/api/register'. The 'Preview' tab displays a JSON response: { "ok": true, "message": "Solicitud recibida. Queda pendiente de aprobación.", "solicitud\_id": 2 }. The 'Params' tab shows URL PREVIEW: http://127.0.0.1:8001/api/register. The 'Body' tab contains the JSON body of the request. The 'Headers' tab shows the response headers: Status: 201 Created, Content-Type: application/json, Date: Mon, 13 Mar 2023 14:45:11 GMT, Server: Apache/2.4.41 (Ubuntu), Connection: keep-alive, Content-Length: 96. The status bar at the bottom indicates 201 Created, 331 ms, 96 B.

Evidencia del proceso de registro desde la lógica de “nuevo socio”.

Se ejecuta POST /api/register y la API responde 201 Created, con el mensaje “solicitud recibida, queda pendiente de aprobación” y un solicitud\_id, demostrando que la solicitud se crea correctamente en estado pendiente para ser gestionada desde el backoffice.

## Figura

Evidencia extraída de Insomnia - Login socio aprobado – POST /api/login

The screenshot shows the Insomnia API client interface. The top navigation bar includes Application, File, Edit, View, Window, Tools, Help, and a search bar. Below the navigation is a tab bar with several tabs: variable, Login socio, Abilities-Admin, POST - Login, Api-USU - Login Us..., health, New Request, and others. The active tab is 'POST - Login'. The main workspace shows a POST request to 'http://127.0.0.1:8001/api/login'. The 'Preview' tab displays a JSON response: { "token": "18101234567890123456789012345678", "user": "valentino", "solicitud\_id": 2, "rol": "socio", "correo": "valentino@correo.com", "ci\_usuario": "1234567890", "email": "valentino.aprobado@correo.test", "nombre": "Valentino" }. The 'Params' tab shows URL PREVIEW: http://127.0.0.1:8001/api/login. The 'Body' tab contains the JSON body of the request. The 'Headers' tab shows the response headers: Status: 200 OK, Content-Type: application/json, Date: Mon, 13 Mar 2023 14:45:11 GMT, Server: Apache/2.4.41 (Ubuntu), Connection: keep-alive, Content-Length: 207. The status bar at the bottom indicates 200 OK, 807 ms, 207 B.

Evidencia del inicio de sesión exitoso de un socio aprobado.

Se envían las credenciales de un socio con estado aprobado al endpoint POST /api/login y la API responde 200 OK, devolviendo token y datos básicos del usuario, habilitando el acceso a los recursos protegidos.

## Figura

## Evidencia extraída de Insomnia - Login con credenciales inválidas (API Usuarios – Respuesta 422)

The screenshot shows the Insomnia REST client interface. The request path is `POST /api/login`. The JSON body contains:

```
1: {  
2:   "login": "22222222",  
3:   "password": "11111111"  
4: }
```

The response status is **422 Unprocessable Entity**. The preview pane shows the JSON response:

```
1: {  
2:   "message": "Credenciales inválidas.",  
3:   "error": 1,  
4:   "login": "",  
5:   "password": ""  
6: }
```

Figura

## Evidencia extraída de Insomnia - Consulta de usuario autenticado – GET /api/me

The screenshot shows the Insomnia REST client interface. The request path is `GET /api/me`. The response status is **200 OK**. The preview pane shows the JSON response:

```
1: {  
2:   "id": 1,  
3:   "name": "Valentín",  
4:   "email": "valentin.sanchez@cyber.test",  
5:   "rol": "socio",  
6:   "rol_id": 1,  
7:   "rol_nombre": "Socio",  
8:   "rol_descripcion": "Socio autorizado",  
9:   "rol_descripcion_id": 1,  
10:  "rol_descripcion_nombre": "Socio autorizado",  
11:  "rol_descripcion_email": "socio.autorizado@cyber.test",  
12:  "rol_descripcion_result": "socio.autorizado@cyber.test",  
13:  "rol_descripcion_nombre": "Valentín"  
14: }
```

Evidencia del funcionamiento del endpoint que devuelve los datos del usuario logueado.

Utilizando el token obtenido en el login, se invoca `GET /api/me` y la API responde 200 OK con la información del usuario autenticado, validando el uso correcto del token en cabecera `Authorization`.

Figura

### Evidencia extraída de Insomnia - Consulta de perfil completo del socio – GET /api/perfil

The screenshot shows the Insomnia interface with a successful response (200 OK) for a GET request to `http://127.0.0.1:8001/api/perfil`. The response body is a JSON object containing user profile information:

```
1 - {  
2 -   "ok": true,  
3 -   "user": {  
4 -     "id": null,  
5 -     "rol": "socio",  
6 -     "nombre": "Valentino",  
7 -     "cuenta": "valentino",  
8 -     "email": "socio.aprobado@cyber.test",  
9 -     "telefono": "23222222",  
10 -   }  
11 - }
```

Evidencia de acceso a los datos de perfil del socio autenticado.

Con el mismo token se llama al endpoint protegido GET /api/perfil y la API responde 200 OK con el detalle del perfil (rol, email, estado\_registro, etc.), demostrando el acceso restringido a información sensible.

### Figura

### Evidencia extraída de Insomnia - Cierre de sesión – POST /api/logout

The screenshot shows the Insomnia interface with a successful response (200 OK) for a POST request to `http://127.0.0.1:8001/api/logout`. The response body is a JSON object indicating the token was invalidated:

```
1 - {  
2 -   "ok": true  
3 - }
```

Evidencia del proceso de cierre de sesión y revocación del token.

Se ejecuta POST /api/logout con un token activo y la API responde **200 OK** con {"ok": true}, indicando que el token fue invalidado y que el usuario queda desautenticado.

### Figura

### Evidencia extraída de Insomnia - Health check de la API – GET /api/health

The screenshot shows the Insomnia API client interface. The top navigation bar includes Application, File, Edit, View, Window, Tools, Help, and a user icon. The main workspace has tabs for variable, Login socio, Abilities-Admin, POST - Login, and Api-COP - GET api... (which is the active tab). The URL field shows "GET http://127.0.0.1:8002/api/health". The status bar indicates "200 OK", "310 ms", and "11 B". Below the URL field, there are sections for Params, Body, Auth, Headers, Scripts, Docs, Preview, Headers, Cookies, Tests, Mock, and Console. The preview pane shows a JSON response: {"ok": true}. On the left sidebar, there is a tree view of API endpoints under "v2 test api-usuarios-13.09.2025", including "Base Environment", "Add Cookies", "Add Certificates", and various endpoint entries like "GET /api/comprobantes/estado", "POST /api/comprobantes", "GET /api/horas/mias", "POST /api/horas", "GET /api/health", etc.

Evidencia del endpoint de monitoreo de salud del servicio.

Se invoca GET /api/health y la API responde **200 OK** con {"ok": true}, utilizándose como prueba rápida de disponibilidad y correcto funcionamiento general de la API.