

Informe de documentación API REST

POST /register

- Registra un nuevo usuario usando su correo.
- Body: email: T
- Ejemplo:

```
{"email": "user@example.com"}
```

POST /login

- Inicia sesión con nombre de usuario y contraseña.
- Body: username: T, password: T
- Ejemplo:

```
{"username": "usuario1", "password": "miPasswordSegura"}
```

GET /user/:userId

- Obtiene información de un usuario.
- Params: userId: N
- Ejemplo:

```
{"success": true, "user": {"id": 123, "username": "usuario1", "email": "user@example.com", "points": 150, "active_hours": 10.5, "total_distance": 25.3, "town_hall": {"name": "Madrid", "province": "Madrid"}, "role": "walker"}}
```

PUT /user/:userId

- Actualiza el perfil de un usuario.
- Params: userId: N
- Body: username: T, email: T, current_password: T, new_password: T
- Ejemplo:

```
{"username": "nuevoUsuario", "email": "nuevo@example.com", "current_password": "OldPassword", "new_password": "NewPassword"}
```

PUT /user/activity

- Actualiza la actividad del usuario.
- Body: userId: N, time: R, distance: R
- Ejemplo:

```
{"userId": 123,"time": 1.5,"distance": 3.2}
```

POST /user/daily-stats

- Registra estadísticas diarias del usuario.
- Body: userId: N, activeHours: R, distance: R, points: N
- Ejemplo:

```
{"userId": 123,"activeHours": 2,"distance": 5,"points": 10}
```

POST /node/link

- Vincula un nodo a un usuario.
- Body: userId: N, nodeName: T
- Ejemplo:

```
{"userId": 123,"nodeName": "GTI"}
```

GET /node/ofUser/:userId

- Devuelve el nodo vinculado a un usuario.
- Params: userId: N
- Ejemplo:

```
{"success": true,"node": {"id": 5,"name": "NODE-01","status": "active","lastStatusUpdate": "2025-01-01T10:00:00Z"}}
```

DELETE /node/ofUser/:userId

- Desvincula el nodo de un usuario.
- Params: userId: N
- Ejemplo:

```
{"success": true,"message": "Nodo desvinculado correctamente"}
```

GET /usuario/calidad-aire-resumen

- Devuelve resumen de calidad del aire del usuario.
- Query: userId: N
- Ejemplo:

```
{"success": true,"status": "buena","summaryText": "La calidad del aire ha sido buena","timeHours": 2.5,"distanceKm": 5.2,"points": 20,"graph": {"timestamps": ["10:00","11:00"],"o3": [0.02,0.03],"no2": [0.01,0.015],"co": [0.3,0.4]}}
```

POST /measurements

- Inserta mediciones de un nodo.
- Body: nodeId: N, co: R, o3: R, no2: R, latitude: R, longitude: R
- Ejemplo:

```
{"nodeId": 5,"co": 0.3,"o3": 0.02,"no2": 0.01,"latitude": 40.4168,"longitude": -3.7038}
```

GET /getAyuntamientos

- Obtiene la lista de ayuntamientos.
- Sin parámetros
- Ejemplo:

```
{"success": true,"data": [{"id": "1","name": "Ayuntamiento A"}, {"id": "2","name": "Ayuntamiento B"}]}
```

POST /apply

- Crea una solicitud previa para registro.
- Body: firstName: T, lastName: T, email: T, dni: T, phone: T, townHallId: N
- Ejemplo:

```
{"firstName": "Ana","lastName": "Pérez","email": "ana@example.com","dni": "12345678A","phone": "600123123","townHallId": 1}
```

DELETE /application/:applicationId

- Elimina una solicitud por id.
- Params: applicationId: N
- Ejemplo:

```
{"success": true,"message": "Solicitud eliminada"}
```

GET /points/:userId

- Obtiene puntos totales de usuario.
- Params: userId: N
- Ejemplo:

```
{"success": true,"points": 150}
```

PUT /points

- Suma puntos al total de usuario.
- Body: userId: N, points: N
- Ejemplo:

```
{"userId": 123, "points": 20}
```

GET /prizes

- Obtiene todos los premios activos.
- Sin parámetros
- Ejemplo:

```
{"success": true, "prizes": [{"id": 1, "name": "Camiseta", "description": "Camiseta oficial", "points_required": 100, "quantity_available": 10, "initial_quantity": 50, "active": 1}]} 
```

POST /redeem

- Canjea puntos por premio.
- Body: userId: N, prizeId: N
- Ejemplo:

```
{"userId": 123, "prizeId": 1}
```

GET /redemptions/:userId

- Obtiene historial de canjes.
- Params: userId: N
- Ejemplo:

```
{"success": true, "redemptions": [{"id": 1, "coupon_code": "ABC-1234-XY9Z", "redemption_date": "2025-01-01T12:00:00Z", "prize_name": "Camiseta", "description": "Camiseta oficial", "points_required": 100, "image_url": null}]} 
```

GET /informeNodos/:tipo

- Devuelve info nodos por tipo.
- Params: tipo: T ("todos" | "inactivos" | "erroneos")
- Ejemplo:

```
{"success": true, "nodos": [{"id": 1, "name": "NODE-01", "username": "usuario1", "status": "active", "lastStatusUpdate": "2025-01-01T12:00:00Z"}]}
```

```
01T10:00:00Z", "horas_desde_actualizacion": 2, "tiene_mediciones_recientes":  
1}]}}
```