

API de Reportes

Listar Reportes

Endpoint: GET /reports

Obtiene todos los reportes con paginación.

Permiso necesario: view_reports

Request HTTP

```
GET /reports?_start=0&_end=10&_sort=tiempo_fecha&_order=DESC HTTP/1.1
Host: localhost:3000
Authorization: Bearer tu-token-aqui
```

Response HTTP

```
HTTP/1.1 200 OK
Content-Type: application/json
X-Total-Count: 150

[
  {
    "_id": "507f1f77bcf86cd799439011",
    "folio": "1001",
    "tiempo_fecha": "2025-10-23T10:30:00.000Z",
    "ubi": "Av. Insurgentes Sur 123",
    "modo_de_activacion": "C4",
    "gravedad_emergencia": 3,
    "tipo_servicio": ["medical"],
    "dictamen": "Traslado a hospital",
    "usuario_reportando": "Carlos Hernández"
  }
]
```

Ejemplo JavaScript

```
async function obtenerReportes() {
  const token = localStorage.getItem('token');

  const response = await fetch('http://localhost:3000/reports?_start=0&_end=10', {
    headers: {
      'Authorization': `Bearer ${token}`
    }
  });

  const total = response.headers.get('X-Total-Count');
  const reportes = await response.json();
}
```

```
console.log(`Total de reportes: ${total}`);  
return reportes;  
}
```

Obtener Reporte por ID

Endpoint: GET /reports/:id

Permiso necesario: view_reports

Request HTTP

```
GET /reports/507f1f77bcf86cd799439011 HTTP/1.1  
Host: localhost:3000  
Authorization: Bearer tu-token-aqui
```

Ejemplo JavaScript

```
async function obtenerReporte(id) {  
  const token = localStorage.getItem('token');  
  
  const response = await fetch(`http://localhost:3000/reports/${id}`, {  
    headers: {  
      'Authorization': `Bearer ${token}`  
    }  
  });  
  
  return await response.json();  
}
```

Crear Reporte

Endpoint: POST /reports

Permiso necesario: create_reports

Request HTTP

```
POST /reports HTTP/1.1  
Host: localhost:3000  
Authorization: Bearer tu-token-aqui  
Content-Type: application/json  
  
{  
  "folio": "1050",  
  "tiempo_fecha": "2025-10-23T14:30:00.000Z",  
  "tiempo_fecha_atencion": "2025-10-23T14:45:00.000Z",  
  "ubi": "Calle Reforma 456, Cuajimalpa",  
}
```

```

"codigoPostal": "05000",
"modo_de_activacion": "C4",
"gravedad_emergencia": 2,
"tipo_servicio": ["accident", "medical"],
"tiempo_translado": 25,
"kilometros_recorridos": 8.5,
"dictamen": "Traslado a hospital por fractura",
"trabaja_realizado": [
  { "trabajo": "Inmovilización de extremidad" },
  { "trabajo": "Signos vitales estables" }
],
"nombres_afectados": [
  { "nombre": "Pedro Martínez" }
],
"dependencias_participantes": [
  { "dependencia": "Policía Municipal" }
],
"observaciones": "Paciente consciente y cooperativo",
"otros": ""
}

```

Ejemplo JavaScript

```

async function crearReporte(datos) {
  const token = localStorage.getItem('token');

  const response = await fetch('http://localhost:3000/reports', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${token}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(datos)
  });

  return await response.json();
}

// Usar
const nuevoReporte = {
  folio: '1050',
  tiempo_fecha: new Date().toISOString(),
  tiempo_fecha_atencion: new Date().toISOString(),
  ubi: 'Calle Reforma 456, Cuajimalpa',
  codigoPostal: '05000',
  modo_de_activacion: 'C4',
  gravedad_emergencia: 2,
  tipo_servicio: ['accident', 'medical'],
  tiempo_translado: 25,
  kilometros_recorridos: 8.5,
  dictamen: 'Traslado a hospital por fractura',

```

```

trabaja_realizado: [
  { trabajo: 'Inmovilización de extremidad' }
],
nombres_afectados: [
  { nombre: 'Pedro Martínez' }
],
dependencias_participantes: [
  { dependencia: 'Policía Municipal' }
],
observaciones: 'Paciente consciente',
otros: ''
};

crearReporte(nuevoReporte).then(reporte => {
  console.log('Reporte creado:', reporte._id);
});

```

Actualizar Reporte

Endpoint: PUT /reports/:id

Permiso necesario: edit_reports

Request HTTP

```

PUT /reports/507f1f77bcf86cd799439011 HTTP/1.1
Host: localhost:3000
Authorization: Bearer tu-token-aqui
Content-Type: application/json

{
  "observaciones": "Se actualizó el estado del paciente",
  "dictamen": "Alta médica"
}

```

Ejemplo JavaScript

```

async function actualizarReporte(id, cambios) {
  const token = localStorage.getItem('token');

  const response = await fetch(`http://localhost:3000/reports/${id}`, {
    method: 'PUT',
    headers: {
      'Authorization': `Bearer ${token}`,
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(cambios)
  });

  return await response.json();
}

```

```
}

// Usar
actualizarReporte('507f1f77bcf86cd799439011', {
  observaciones: 'Paciente recuperado',
  dictamen: 'Alta médica'
});
```

Eliminar Reporte

Endpoint: DELETE /reports/:id

Permiso necesario: delete_reports

Request HTTP

```
DELETE /reports/507f1f77bcf86cd799439011 HTTP/1.1
Host: localhost:3000
Authorization: Bearer tu-token-aqui
```

Ejemplo JavaScript

```
async function eliminarReporte(id) {
  const token = localStorage.getItem('token');

  const response = await fetch(`http://localhost:3000/reports/${id}`, {
    method: 'DELETE',
    headers: {
      'Authorization': `Bearer ${token}`
    }
  });

  return await response.json();
}
```

Mis Reportes

Endpoint: GET /my-reports

Obtiene solo los reportes creados por ti.

Permiso necesario: view_my_reports

Ejemplo JavaScript

```
async function obtenerMisReportes() {
  const token = localStorage.getItem('token');

  const response = await fetch('http://localhost:3000/my-reports', {
```

```
    headers: {
      'Authorization': `Bearer ${token}`
    }
  });

  return await response.json();
}
```

Reportes de Mi Turno

Endpoint: GET /turn-reports

Obtiene reportes de tu turno asignado.

Permiso necesario: view_turn_reports

Ejemplo JavaScript

```
async function obtenerReportesTurno() {
  const token = localStorage.getItem('token');

  const response = await fetch('http://localhost:3000/turn-reports', {
    headers: {
      'Authorization': `Bearer ${token}`
    }
  });

  return await response.json();
}
```

Modos de Activación

- C3 - Centro de Control C3
- C4 - Centro de Control C4
- C5 - Centro de Control C5
- Police - Policía
- Direct - Directo

Tipos de Servicio

- accident - Accidente
- rescue - Rescate
- medical - Médico
- fire - Incendio
- transfer - Traslado