



Índice

-
- 1. Autenticación
 - 2. Usuarios
 - 3. Roles y Permisos
 - 4. Sensores
 - 5. Dispositivos
 - 6. Lecturas
 - 7. Dashboard
 - 8. Códigos de Estado HTTP
 - 9. Filtros y Búsqueda
-

Autenticación

1. Registro de Usuario

Endpoint: POST /api/auth/register/

Permisos: Público (AllowAny)

Request Body:

```
{  
    "username": "string",  
    "email": "string",  
    "password": "string",  
    "password_confirm": "string",  
    "first_name": "string",  
    "last_name": "string",  
    "tipo_usuario": "interno|externo",  
    "rol": 1 // ID del rol (opcional)  
}
```

Response (201 Created):

```
{
  "user": {
    "id": 1,
    "username": "nuevo_usuario",
    "email": "usuario@example.com",
    "first_name": "Nombre",
    "last_name": "Apellido",
    "full_name": "Nombre Apellido",
    "tipo_usuario": "externo",
    "tipo_usuario_display": "Externo",
    "is_active": true,
    "rol": 2,
    "rol_detail": {...}
  },
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "access": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "message": "Usuario registrado exitosamente"
}
```

2. Login

Endpoint: POST /api/auth/login/

Permisos: Público (AllowAny)

Request Body:

```
{
  "username": "admin",
  "password": "admin123"
}
```

Response (200 OK):

```
{
  "user": {
    "id": 1,
    "username": "admin",
    "email": "admin@iotplatform.com",
    "full_name": "Super Admin",
    "tipo_usuario": "interno",
    "is_active": true,
    "is_superuser": true,
    "rol": 1,
    "rol_detail": {
      "id": 1,
      "nombre": "superusuario",
      "nombre_display": "Superusuario",
      "permisos": [...]
    }
  },
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "access": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "message": "Login exitoso"
}
```

3. Refresh Token

Endpoint: POST /api/auth/refresh/

Permisos: Público (AllowAny)

Request Body:

```
{
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGc..."
}
```

Response (200 OK):

```
{
  "access": "eyJ0eXAiOiJKV1QiLCJhbGc...",
  "refresh": "eyJ0eXAiOiJKV1QiLCJhbGc..."
}
```

4. Perfil del Usuario Actual

Endpoint: GET /api/users/me/

Permisos: Autenticado

Headers: Authorization: Bearer {access_token}

Response (200 OK):

```
{
  "id": 1,
  "username": "admin",
  "email": "admin@iotplatform.com",
  "first_name": "Super",
  "last_name": "Admin",
  "full_name": "Super Admin",
  "tipo_usuario": "interno",
  "tipo_usuario_display": "Interno",
  "is_active": true,
  "is_staff": true,
  "is_superuser": true,
  "rol": 1,
  "rol_detail": {...},
  "created_at": "2024-12-01T10:00:00Z",
  "updated_at": "2024-12-01T10:00:00Z",
  "last_login": "2024-12-04T08:30:00Z"
}
```

Usuarios

1. Listar Usuarios

Endpoint: GET /api/users/

Permisos: Superusuario

Headers: Authorization: Bearer {access_token}

Query Parameters:

- search : Búsqueda por username, email, first_name, last_name
- ordering : Ordenar por username, created_at, last_login
- page : Número de página

Response (200 OK):

```
{
  "count": 10,
  "next": "http://localhost:8000/api/users/?page=2",
  "previous": null,
  "results": [
    {
      "id": 1,
      "username": "admin",
      "email": "admin@iotplatform.com",
      "full_name": "Super Admin",
      "tipo_usuario": "interno",
      "is_active": true,
      "rol": 1,
      "rol_detail": {...}
    }
  ]
}
```

2. Crear Usuario

Endpoint: POST /api/users/**Permisos:** Superusuario**Headers:** Authorization: Bearer {access_token}**Request Body:**

```
{
  "username": "operador1",
  "email": "operador1@example.com",
  "password": "password123",
  "first_name": "Operador",
  "last_name": "Uno",
  "tipo_usuario": "interno",
  "rol": 2,
  "is_active": true
}
```

Response (201 Created):

```
{
  "id": 5,
  "username": "operador1",
  "email": "operador1@example.com",
  "first_name": "Operador",
  "last_name": "Uno",
  "tipo_usuario": "interno",
  "is_active": true,
  "rol": 2
}
```

3. Actualizar Usuario

Endpoint: PUT/PATCH /api/users/{id}/

Permisos: Superusuario

Headers: Authorization: Bearer {access_token}

Request Body (PATCH - parcial):

```
{
  "is_active": false,
  "rol": 3
}
```

4. Activar Usuario

Endpoint: POST /api/users/{id}/activate/

Permisos: Superusuario

Headers: Authorization: Bearer {access_token}

Response (200 OK):

```
{
  "message": "Usuario operador1 activado exitosamente"
}
```

5. Desactivar Usuario

Endpoint: POST /api/users/{id}/deactivate/

Permisos: Superusuario

Headers: Authorization: Bearer {access_token}

Response (200 OK):

```
{
  "message": "Usuario operador1 desactivado exitosamente"
}
```

Roles y Permisos

1. Listar Roles

Endpoint: GET /api/roles/

Permisos: Superusuario

Response (200 OK):

```
{  
  "count": 3,  
  "results": [  
    {  
      "id": 1,  
      "nombre": "superusuario",  
      "nombre_display": "Superusuario",  
      "descripcion": "Tiene acceso completo a todas las funcionalidades del sistema",  
      "permisos": [  
        {  
          "id": 1,  
          "nombre": "Gestionar Usuarios",  
          "codigo": "gestionar_usuarios",  
          "descripcion": "Permite crear, editar y eliminar usuarios"  
        }  
      ],  
      "created_at": "2024-12-01T10:00:00Z"  
    }  
  ]  
}
```

2. Listar Permisos

Endpoint: GET /api/permisos/

Permisos: Superusuario

Response (200 OK):

```
{  
  "count": 13,  
  "results": [  
    {  
      "id": 1,  
      "nombre": "Gestionar Usuarios",  
      "codigo": "gestionar_usuarios",  
      "descripcion": "Permite crear, editar y eliminar usuarios",  
      "created_at": "2024-12-01T10:00:00Z"  
    }  
  ]  
}
```

Sensores

1. Listar Sensores

Endpoint: GET /api/sensors/

Permisos: Autenticado

Headers: Authorization: Bearer {access_token}

Query Parameters:

- search : Búsqueda por nombre, tipo, descripción
- tipo : Filtrar por tipo de sensor
- estado : Filtrar por estado (activo, inactivo, mantenimiento)
- ordering : Ordenar por nombre, tipo, created_at

Response (200 OK):

```
{
  "count": 6,
  "results": [
    {
      "id": 1,
      "nombre": "DHT22 - Temperatura",
      "tipo": "temperatura",
      "tipo_display": "Temperatura",
      "unidad_medida": "°C",
      "rango_min": -40.0,
      "rango_max": 80.0,
      "estado": "activo",
      "estado_display": "Activo",
      "descripcion": "",
      "created_by": 1,
      "created_by_username": "admin",
      "created_at": "2024-12-01T10:00:00Z",
      "updated_at": "2024-12-01T10:00:00Z"
    }
  ]
}
```

2. Crear Sensor

Endpoint: POST /api/sensors/

Permisos: Superusuario o Operador

Headers: Authorization: Bearer {access_token}

Request Body:

```
{
  "nombre": "BMP280 - Presión",
  "tipo": "presión",
  "unidad_medida": "hPa",
  "rango_min": 300,
  "rango_max": 1100,
  "estado": "activo",
  "descripcion": "Sensor de presión barométrica"
}
```

Response (201 Created):

```
{
  "id": 7,
  "nombre": "BMP280 - Presión",
  "tipo": "presión",
  "tipo_display": "Presión",
  "unidad_medida": "hPa",
  "rango_min": 300.0,
  "rango_max": 1100.0,
  "estado": "activo",
  "estado_display": "Activo",
  "created_by": 1,
  "created_by_username": "admin",
  "created_at": "2024-12-04T10:00:00Z"
}
```

3. Sensores Disponibles

Endpoint: GET /api/sensors/available/

Permisos: Autenticado

Response (200 OK):

```
[
  {
    "id": 1,
    "nombre": "DHT22 - Temperatura",
    "tipo": "temperatura",
    "estado": "activo",
    ...
  }
]
```

4. Tipos de Sensores

Endpoint: GET /api/sensors/tipos/

Permisos: Autenticado

Response (200 OK):

```
[
  {"value": "temperatura", "label": "Temperatura"},
  {"value": "humedad", "label": "Humedad"},
  {"value": "presión", "label": "Presión"},
  {"value": "luz", "label": "Luz"},
  {"value": "movimiento", "label": "Movimiento"},
  {"value": "gas", "label": "Gas"}
]
```

Dispositivos

1. Listar Dispositivos

Endpoint: GET /api/devices/

Permisos: Autenticado (Operadores ven solo sus dispositivos)

Headers: Authorization: Bearer {access_token}

Query Parameters:

- search : Búsqueda por nombre, tipo, identificador_unico, ubicacion
- tipo : Filtrar por tipo de dispositivo
- estado : Filtrar por estado
- operador : Filtrar por ID de operador

Response (200 OK):

```
{
  "count": 4,
  "results": [
    {
      "id": 1,
      "nombre": "Raspberry Pi - Sala",
      "tipo": "raspberry_pi",
      "tipo_display": "Raspberry Pi",
      "identificador_unico": "RPI-SALA-001",
      "ubicacion": "Sala Principal",
      "estado": "activo",
      "estado_display": "Activo",
      "descripcion": "",
      "operador_asignado": 2,
      "operador_username": "operador1",
      "sensores_asignados": [
        {
          "id": 1,
          "sensor": 1,
          "sensor_nombre": "DHT22 - Temperatura",
          "sensor_detail": {...},
          "configuracion_json": {},
          "activo": true,
          "fecha_asignacion": "2024-12-01T10:00:00Z"
        }
      ],
      "cantidad_sensores": 3,
      "created_at": "2024-12-01T10:00:00Z"
    }
  ]
}
```

2. Crear Dispositivo

Endpoint: POST /api/devices/

Permisos: Superusuario o Operador

Headers: Authorization: Bearer {access_token}

Request Body:

```
{
  "nombre": "ESP32 - Oficina",
  "tipo": "esp32",
  "identificador_unico": "ESP32-OFICINA-001",
  "ubicacion": "Oficina Principal",
  "estado": "activo",
  "descripcion": "ESP32 en oficina con DHT22"
}
```

Response (201 Created):

```
{
  "id": 5,
  "nombre": "ESP32 - Oficina",
  "tipo": "esp32",
  "tipo_display": "ESP32",
  "identificador_unico": "ESP32-OFICINA-001",
  "ubicacion": "Oficina Principal",
  "estado": "activo",
  "operador_asignado": null,
  "sensores_asignados": [],
  "cantidad_sensores": 0,
  "created_at": "2024-12-04T10:00:00Z"
}
```

3. Asignar Sensor a Dispositivo

Endpoint: POST /api/devices/{id}/assign-sensor/

Permisos: Superusuario o Operador

Headers: Authorization: Bearer {access_token}

Request Body:

```
{
  "sensor_id": 1,
  "configuracion_json": {
    "intervalo": 60,
    "umbral_alerta": 30
  }
}
```

Response (201 Created):

```
{
  "message": "Sensor asignado exitosamente",
  "asignacion": {
    "id": 10,
    "dispositivo": 5,
    "dispositivo_nombre": "ESP32 - Oficina",
    "sensor": 1,
    "sensor_nombre": "DHT22 - Temperatura",
    "configuracion_json": {
      "intervalo": 60,
      "umbral_alerta": 30
    },
    "activo": true,
    "fecha_asignacion": "2024-12-04T10:00:00Z"
  }
}
```

4. Asignar Operador a Dispositivo

Endpoint: POST /api/devices/{id}/assign-operator/

Permisos: Superusuario

Headers: Authorization: Bearer {access_token}

Request Body:

```
{
  "operador_id": 2
}
```

Response (200 OK):

```
{
  "message": "Operador asignado exitosamente",
  "dispositivo": {
    "id": 5,
    "nombre": "ESP32 - Oficina",
    "operador_asignado": 2,
    "operador_username": "operador1",
    ...
  }
}
```

5. Remover Sensor de Dispositivo

Endpoint: DELETE /api/devices/{id}/remove-sensor/?sensor_id=1

Permisos: Superusuario o Operador

Headers: Authorization: Bearer {access_token}

Response (200 OK):

```
{
  "message": "Sensor removido exitosamente"
}
```

6. Tipos de Dispositivos

Endpoint: GET /api/devices/tipos/

Permisos: Autenticado

Response (200 OK):

```
[
  {"value": "raspberry_pi", "label": "Raspberry Pi"},  

  {"value": "esp32", "label": "ESP32"},  

  {"value": "arduino", "label": "Arduino"},  

  {"value": "esp8266", "label": "ESP8266"}  

]
```

Lecturas

1. Listar Lecturas

Endpoint: GET /api/readings/

Permisos: Autenticado (Operadores ven solo lecturas de sus dispositivos)

Headers: Authorization: Bearer {access_token}

Query Parameters:

- dispositivo : Filtrar por ID de dispositivo
- sensor : Filtrar por ID de sensor
- fecha_inicio : Filtrar desde fecha (YYYY-MM-DDTHH:MM:SS)
- fecha_fin : Filtrar hasta fecha (YYYY-MM-DDTHH:MM:SS)
- ordering : Ordenar por timestamp

Response (200 OK):

```
{
  "count": 120,
  "results": [
    {
      "id": 1,
      "dispositivo": 1,
      "dispositivo_nombre": "Raspberry Pi - Sala",
      "sensor": 1,
      "sensor_nombre": "DHT22 - Temperatura",
      "sensor_unidad": "°C",
      "valor": 25.5,
      "timestamp": "2024-12-04T10:30:00Z",
      "metadata_json": {}
    }
  ]
}
```

2. Crear Lectura

Endpoint: POST /api/readings/

Permisos: Superusuário o Operador

Headers: Authorization: Bearer {access_token}

Request Body:

```
{  
    "dispositivo": 1,  
    "sensor": 1,  
    "valor": 25.5,  
    "metadata_json": {  
        "calidad": "buena",  
        "bateria": 85  
    }  
}
```

Response (201 Created):

```
{  
    "id": 121,  
    "dispositivo": 1,  
    "dispositivo_nombre": "Raspberry Pi - Sala",  
    "sensor": 1,  
    "sensor_nombre": "DHT22 - Temperatura",  
    "sensor_unidad": "°C",  
    "valor": 25.5,  
    "timestamp": "2024-12-04T10:35:00Z",  
    "metadata_json": {  
        "calidad": "buena",  
        "bateria": 85  
    }  
}
```

3. Crear Lecturas en Bulk

Endpoint: POST /api/readings/bulk/

Permisos: Superusuário o Operador

Headers: Authorization: Bearer {access_token}

Request Body:

```
{
  "lecturas": [
    {
      "dispositivo": 1,
      "sensor": 1,
      "valor": 25.5
    },
    {
      "dispositivo": 1,
      "sensor": 2,
      "valor": 60.2
    },
    {
      "dispositivo": 2,
      "sensor": 3,
      "valor": 1013.25
    }
  ]
}
```

Response (201 Created):

```
{
  "message": "3 lecturas creadas exitosamente",
  "count": 3
}
```

4. Estadísticas de Lecturas

Endpoint: GET /api/readings/estadisticas/

Permisos: Autenticado

Headers: Authorization: Bearer {access_token}

Query Parameters:

- dispositivo : ID de dispositivo
- sensor : ID de sensor

Response (200 OK):

```
{
  "total": 120,
  "promedio": 24.8,
  "maximo": 32.5,
  "minimo": 18.2
}
```

Dashboard

1. Estadísticas del Sistema

Endpoint: GET /api/dashboard/stats/

Permisos: Autenticado

Headers: Authorization: Bearer {access_token}

Response para Superusuario (200 OK):

```
{
  "total_usuarios": 10,
  "total_sensores": 15,
  "sensores_activos": 12,
  "total_dispositivos": 8,
  "dispositivos_activos": 7,
  "total_lecturas": 1250
}
```

Response para Operador (200 OK):

```
{
  "total_usuarios": 10,
  "total_sensores": 15,
  "sensores_activos": 12,
  "total_dispositivos": 8,
  "dispositivos_activos": 7,
  "total_lecturas": 1250,
  "mis_dispositivos": 2,
  "mis_lecturas": 380
}
```

Códigos de Estado HTTP

Código	Significado	Uso
200	OK	Petición exitosa
201	Created	Recurso creado exitosamente
204	No Content	Eliminación exitosa
400	Bad Request	Datos inválidos
401	Unauthorized	No autenticado
403	Forbidden	Sin permisos
404	Not Found	Recurso no encontrado
500	Internal Server Error	Error del servidor

Filtros y Búsqueda

Búsqueda (search)

Disponible en endpoints de listado. Busca en múltiples campos.

Ejemplo:

```
GET /api/sensors/?search=temperatura
GET /api/users/?search=operador
```

Ordenamiento (ordering)

Ordena los resultados por un campo específico. Usar `-` para orden descendente.

Ejemplo:

```
GET /api/readings/?ordering=-timestamp
GET /api/devices/?ordering=nombre
```

Paginación

Todos los endpoints de listado soportan paginación.

Parámetros:

- `page` : Número de página
- `page_size` : Elementos por página (máx 50)

Ejemplo:

```
GET /api/sensors/?page=2&page_size=20
```

Formato de Errores

Error de Validación (400)

```
{
  "campo": [
    "Este campo es requerido."
  ],
  "password": [
    "Las contraseñas no coinciden."
  ]
}
```

Error de Autenticación (401)

```
{
    "detail": "Las credenciales de autenticación no se proveyeron."
}
```

Error de Permisos (403)

```
{
    "detail": "No tiene permisos para realizar esta acción."
}
```

Error no encontrado (404)

```
{
    "detail": "No encontrado."
}
```

Notas Importantes

1. **Autenticación:** Todos los endpoints (excepto register, login, refresh) requieren el header `Authorization: Bearer {access_token}`
2. **Tokens JWT:**
 - Access token expira en 60 minutos
 - Refresh token expira en 24 horas
3. **Permisos por Rol:**
 - **Superusuario:** Acceso completo
 - **Operador:** Gestionar dispositivos asignados, sensores y lecturas
 - **Solo Lectura:** Solo visualización
4. **Validaciones:**
 - Los sensores tienen rangos min/max que se validan en las lecturas
 - Los dispositivos tienen identificadores únicos
 - Las lecturas validan que el sensor esté asignado al dispositivo
5. **Filtros de Operador:** Los operadores solo ven sus propios dispositivos y lecturas relacionadas

Para más información, consulta la documentación interactiva en: <http://localhost:8000/api/docs/>