

Documentación CMDB API

- Steven Josue González Monroy
- 201903974

Link del REPOSITORIO

- https://github.com/Tiwue/SA_T3_201903974.git

Tabla de Contenidos

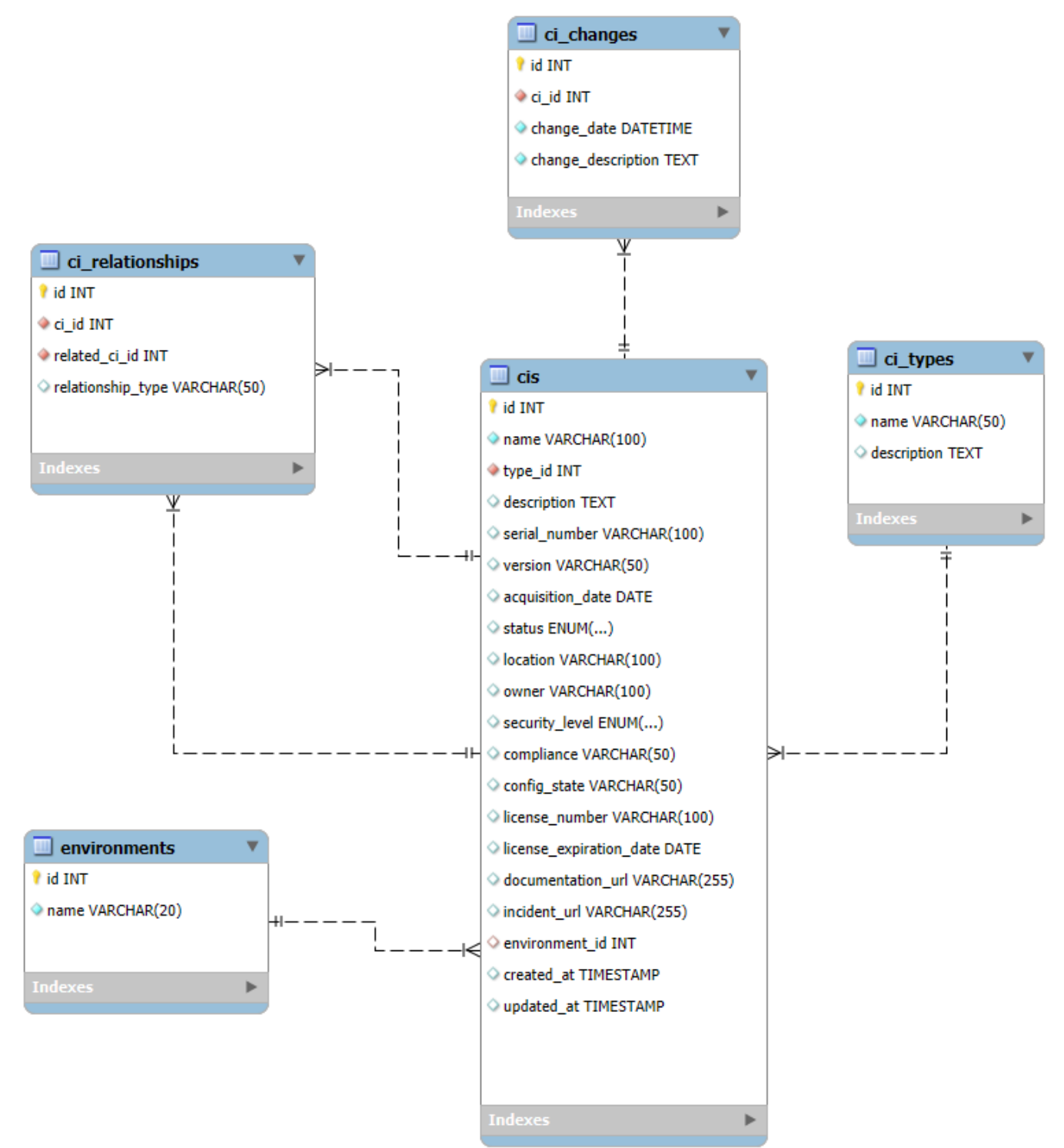
1. [Introducción](#)
2. [Esquema de Base de Datos](#)
 1. [Tablas](#)
 2. [Relaciones](#)
3. [API RESTful](#)
 1. [Configuración General](#)
 2. [Recursos y Endpoints](#)
 - [Tipos de CI \(ci_types\)](#)
 - [Ambientes \(environments\)](#)
 - [Configuration Items \(cis\)](#)
 - [Cambios de CI \(ci_changes\)](#)
 - [Relaciones de CI \(ci_relationships\)](#)

Introducción

Este documento describe de forma detallada el modelo relacional en MySQL de la CMDB y la API RESTful construida con Node.js y Express. Incluye:

- Diseño de tablas y relaciones.
- Endpoints para operaciones CRUD sobre cada recurso.
- Ejemplos de peticiones (curl).

Esquema de Base de Datos



Tablas

Tabla	Descripción
ci_types	Catálogo de tipos de Configuration Item (CI).
environments	Catálogo de ambientes (DEV, QA, PROD, etc.).
cis	Configuration Items con sus atributos principales.
ci_changes	Historial de cambios (auditoría) de cada CI.
ci_relationships	Relaciones entre CIs (dependencias, alojamientos, etc.).

1. ci_types

```
CREATE TABLE ci_types (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(50) NOT NULL UNIQUE,  
  description TEXT  
);
```

2. environments

```
CREATE TABLE environments (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(20) NOT NULL UNIQUE  
);
```

3. cis

```
CREATE TABLE cis (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  type_id INT NOT NULL,  
  description TEXT,  
  serial_number VARCHAR(100),  
  version VARCHAR(50),  
  acquisition_date DATE,  
  status ENUM('Activo', 'Inactivo', 'Mantenimiento', 'Retirado') DEFAULT 'Activo',  
  location VARCHAR(100),  
  owner VARCHAR(100),  
  security_level ENUM('Alto', 'Medio', 'Bajo') DEFAULT 'Medio',  
  compliance VARCHAR(50),  
  config_state VARCHAR(50),  
  license_number VARCHAR(100),  
  license_expiration_date DATE,  
  documentation_url VARCHAR(255),  
  incident_url VARCHAR(255),  
  environment_id INT,  
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  FOREIGN KEY (type_id) REFERENCES ci_types(id),  
  FOREIGN KEY (environment_id) REFERENCES environments(id)  
);
```

4. ci_changes

```
CREATE TABLE ci_changes (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  ci_id INT NOT NULL,  
  change_date DATETIME NOT NULL,  
  change_description TEXT NOT NULL,  
  FOREIGN KEY (ci_id) REFERENCES cis(id) ON DELETE CASCADE  
);
```

5. ci_relationships

```
CREATE TABLE ci_relationships (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  ci_id INT NOT NULL,  
  related_ci_id INT NOT NULL,  
  relationship_type VARCHAR(50),  
  FOREIGN KEY (ci_id) REFERENCES cis(id) ON DELETE CASCADE,  
  FOREIGN KEY (related_ci_id) REFERENCES cis(id) ON DELETE CASCADE  
);
```

Relaciones

- **cis.type_id** → **ci_types.id**
- **cis.environment_id** → **environments.id**
- **ci_changes.ci_id** → **cis.id**
- **ci_relationships.ci_id, related_ci_id** → **cis.id**

API RESTful

Configuración General

- **Base URL:** `http://localhost:3000/api`
- **Formato de intercambio:** JSON
- **Headers comunes:**

```
Content-Type: application/json
```

Recursos y Endpoints

Tipos de CI (`/api/types`)

Método	Ruta	Descripción
GET	/types	Listar todos los tipos de CI.
POST	/types	Crear un nuevo tipo de CI.
GET	/types/:id	Obtener un tipo por su ID.
PUT	/types/:id	Actualizar un tipo existente.
DELETE	/types/:id	Eliminar un tipo de CI.

Ejemplos de peticiones

- **Listar todos**

```
GET http://localhost:3000/api/types
```

- **Crear**

```
POST http://localhost:3000/api/types \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "name": "Middleware",  
  "description": "Servicios intermedios"  
'
```

- **Obtener por ID**

```
GET http://localhost:3000/api/types/1
```

- **Actualizar**

```
PUT http://localhost:3000/api/types/1 \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "name": "Hardware",  
  "description": "Servidores y switches (actualizado)"  
'
```

- **Eliminar**

```
DELETE http://localhost:3000/api/types/1
```

Ambientes (/api/envs)

Método	Ruta	Descripción
GET	/envs	Listar todos los ambientes.
POST	/envs	Crear un nuevo ambiente.
GET	/envs/:id	Obtener un ambiente por ID.
PUT	/envs/:id	Actualizar un ambiente.
DELETE	/envs/:id	Eliminar un ambiente.

Ejemplos de peticiones

- **Listar todos**

```
GET http://localhost:3000/api/envs
```

- **Crear**

```
POST http://localhost:3000/api/envs \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "name": "STAGING"  
'
```

- **Obtener por ID**

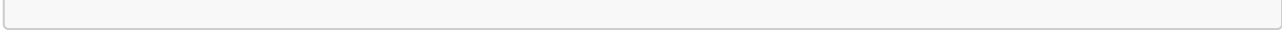
```
GET http://localhost:3000/api/envs/2
```

- **Actualizar**

```
PUT http://localhost:3000/api/envs/2 \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "name": "QA"  
'
```

- **Eliminar**

```
DELETE http://localhost:3000/api/envs/3
```



Configuration Items (/api/cis)

Método	Ruta	Descripción
GET	/cis	Listar todos los Configuration Items.
POST	/cis	Crear un nuevo CI.
GET	/cis/:id	Obtener un CI por su ID.
PUT	/cis/:id	Actualizar un CI existente.
DELETE	/cis/:id	Eliminar un CI.

Ejemplos de peticiones

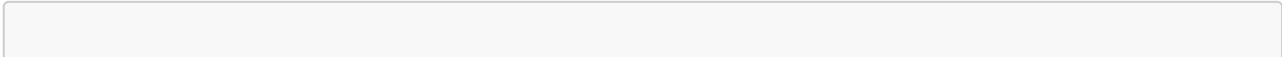
- Listar todos

```
GET http://localhost:3000/api/cis
```

- Crear

```
POST http://localhost:3000/api/cis \
HEADERS: "Content-Type: application/json" \
DATA: '{
  "name": "Servidor-Web-01",
  "type_id": 1,
  "description": "Servidor web principal",
  "serial_number": "SN-WEB-0001",
  "version": "Ubuntu 20.04",
  "acquisition_date": "2023-05-10",
  "status": "Activo",
  "location": "CPD-1",
  "owner": "Infraestructura",
  "documentation_url": "http://docs.empresa/servidor-web-01",
  "incident_url": "http://jira.empresa/browse/INC-123",
  "security_level": "Alto",
  "compliance": "PCI-DSS",
  "config_state": "Aprobado",
  "license_number": "LIC-1234",
  "license_expiration_date": "2024-05-10",
  "environment_id": 3
}'
```

- Obtener por ID



```
GET http://localhost:3000/api/cis/5
```

- **Actualizar**

```
PUT http://localhost:3000/api/cis/5 \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "status": "Mantenimiento",  
  "owner": "Soporte"  
}'
```

- **Eliminar**

```
DELETE http://localhost:3000/api/cis/5
```

Cambios de CI (/api/cis/:ciId/changes)

Método	Ruta	Descripción
GET	/cis/:ciId/changes	Listar historial de cambios de un CI.
POST	/cis/:ciId/changes	Registrar un nuevo cambio (auditoría).

Ejemplos de peticiones

- **Listar cambios**

```
GET http://localhost:3000/api/cis/5/changes
```

- **Crear cambio**

```
POST http://localhost:3000/api/cis/5/changes \  
HEADERS: "Content-Type: application/json" \  
DATA: '{  
  "change_date": "2025-06-22T15:30:00Z",  
  "change_description": "Actualización de kernel"  
}'
```

Relaciones de CI (/api/cis/:ciId/relationships)

Método	Ruta	Descripción
GET	<code>/cis/:ciId/relationships</code>	Listar relaciones de un CI.
POST	<code>/cis/:ciId/relationships</code>	Crear una nueva relación para un CI.
DELETE	<code>/cis/:ciId/relationships/:relId</code>	Eliminar una relación específica de un CI.

Ejemplos de peticiones

- **Listar relaciones**

```
GET http://localhost:3000/api/cis/5/relationships
```

- **Crear relación**

```
POST http://localhost:3000/api/cis/5/relationships \
HEADERS: "Content-Type: application/json" \
DATA: '{
  "related_ci_id": 3,
  "relationship_type": "depende de"
}'
```

- **Eliminar relación**

```
DELETE http://localhost:3000/api/cis/5/relationships/2
```