Universidad San Carlos de Guatemala

Ingeniería Ciencias en Sistemas

Software Avanzado

PROYECTO : API CMDB

Nombre : Andy Ezequiel Sanic Tiul

Carnet : 202006699

Link a repositorio : https://github.com/AndyST70/Practicas-SA-A-202006699/blob/main/Tareas%20vacas/tarea%203/cmdb_api/routes/ci_routes.py

ESTRUCTURA BÁSICA DEL CONCEPTO

Nombre del CI	Tipo de CI	Descripción	Número de Serie	Versión	Fecha de Adquisición	Estado Actual	Relaciones	Padres/Hijos	Ubicación Física	Propietario/Responsable	Fecha de Cambio	Descripción del Cambio	Documentación relacionada
Servidor1	Hardware	Servidor de Aplicaciones	SN123456	v1.0	2022-01-01	Activo	Base de Datos1		Sala de Servidores 1	Equipo de Infraestructura	2022-02-01	Actualización de Software	[Enlace a Manual](url)
Aplicación	Software	Aplicación de contabilidad		v2.5	2022-03-15	Activo	Base de Datos1		Servidor1	Equipo de Desarrollo	2022-04-01	Parche de Seguridad	[Enlace a Documentaci√≥n T√©cnica](url)

Documentación relacionada	Enlaces a Incidentes y Problemas	Niveles de Seguridad	Cumplimiento	Estado de Configuraci√≥n	N√∫mero de Licencia	Fecha de Vencimiento
[Enlace a Manual](url)	[Enlace a Incidente](url)	Alto	Cumple	Aprobado	ABC123	2023-01-01
[Enlace a Documentaci√≥n T√©cnica](url)	[Enlace a Incidente](url)	Medio	Cumple	Aprobado	XYZ456	2024-01-01

TECNOLOGIAS

Se utilizo flask por ser ligero, fácil de configurar y suficiente para una API REST académica.

Tener instalado lo siguiente :

Tecnologias

- JWT
- mysql-connector-python
- python-dotenv
- pandas
- Flask

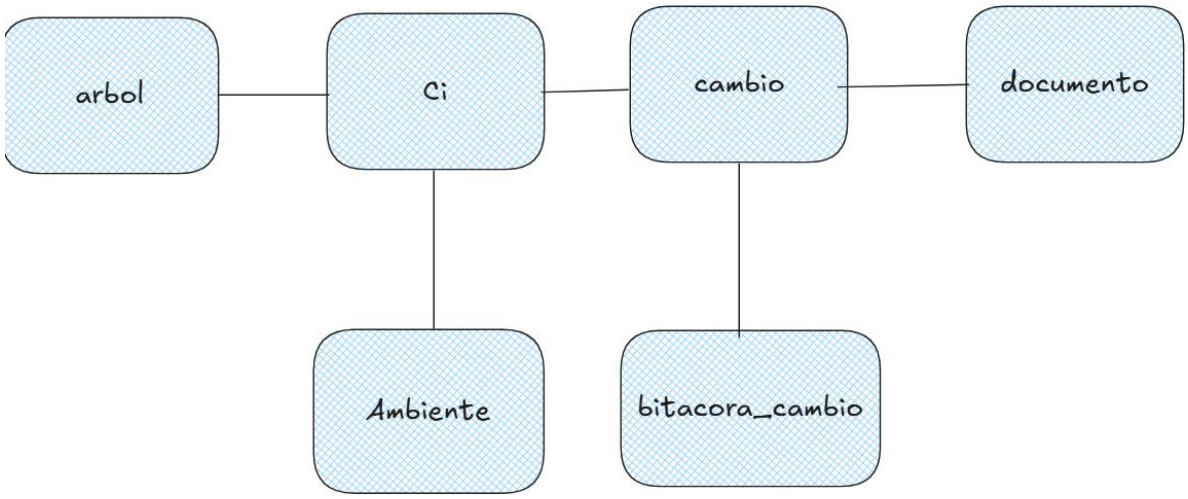
BASE DE DATOS - MYSQL

Se utilizo un modelo relacional que incluye lo siguiente:

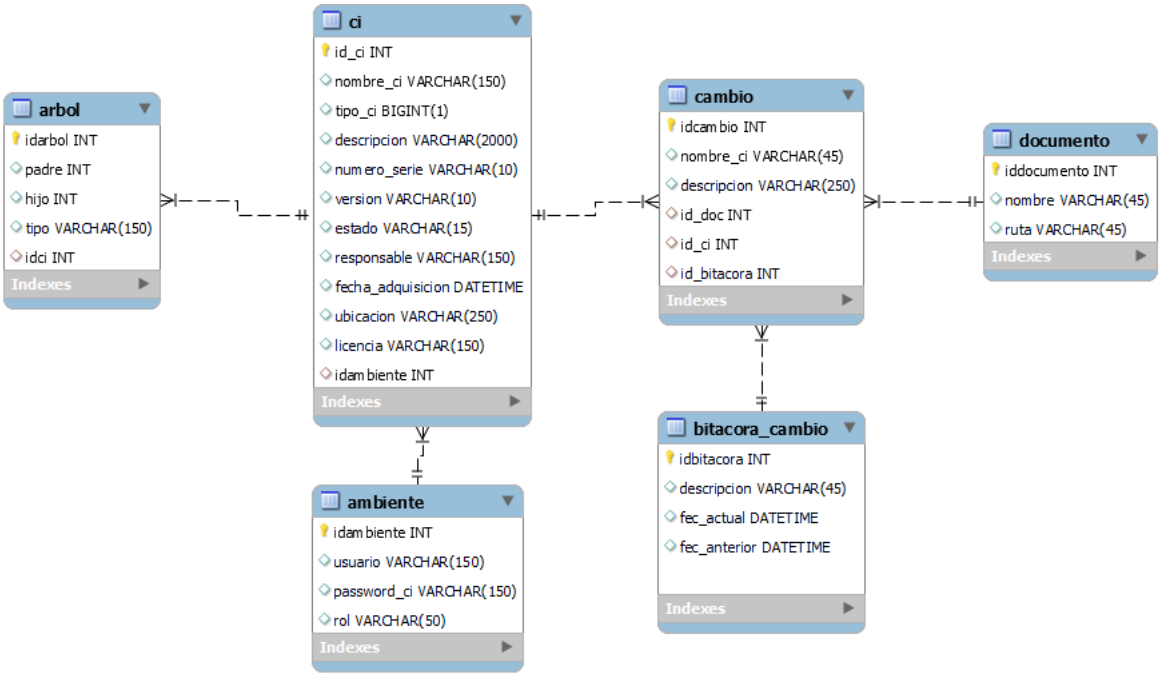
- ci: Información base de los items de configuración.
- ambiente: Etiquetado de los CIs por entorno (requisito funcional).
- arbol: Para representar relaciones tipo padre-hijo entre CIs.
- cambio, bitacora_cambio, documento: Para registrar auditoría básica de cambios, cumpliendo con el requisito de trazabilidad.

Se incluyo un /reset_tablas para pruebas rápidas durante el desarrollo.

DIAGRAMA LÓGICO



ENTIDAD RELACIÓN



API

Método	Ruta	Descripción
POST	/login	Autenticación y generación de token
GET	/ci	Listado de todos los CIs
POST	/ci	Crear nuevo CI
GET	/ci/<id>	Obtener un CI por ID
DELETE	/ci/<id>	Eliminar CI
PUT	/ci/<int:id>	Actualizar ci
POST	/cambio	Crear cambio con bitácora y doc
PUT	/cambio/<id>	Actualizar un cambio
POST	/reset_tablas	Limpia todas las tablas (dev)
POST	/cargar_ci	Carga masiva desde Excel

ESTRUCTURA DE ARQUITECTURA

```
cmdb_api/
|— config/
| └— db.py
|— routes/
| └— ci_routes.py
|— utils/
| └— utils.py
|— main.py
|— .env
|— requirements.txt
|— CMDB.xlsx
|— README.md
```

COMO EJECUTAR

- Clona el repo
- Crear .env
- Instalar dependencias pip install -r requeriments.txt
- Corre con pyton main.py
- Acceder con postman o navegador

ROLES POR AMBIENTE

Ambiente	Descripción	Responsabilidad Principal
DEV	Desarrollo	Crear, probar y modificar CIs en fase inicial
QA	Calidad / Testing	Validar que los CIs funcionen correctamente antes de producción
PROD	Producción	Ejecutar CIs en ambiente estable y controlado

EXPLICACIÓN DE CONFIGURACIÓN

La base de datos se levantará en un contenedor de Docker mediante las siguientes variables de entorno.

DB_HOST
DB_USER
DB_PASS

INICIALIZACIÓN DB

Con el entorno virtual de la DB de docker configurada es necesario levantar el contenedor, los siguientes comandos son los recomendados:

Levantar el proyecto en segundo plano y hacer el build.

- docker compose up -d --build

Eliminar los contenedores y volumenes

- docker compose down -v --rmi all

DEPLIEGUE DE ENTORNO

Pasos para correr son:

- python -m venv venv
- venv\Scripts\activate
- pip install -r requirements.txt

Esto ejecutara este api:

- http://127.0.0.1:5000/