

Para ejecutar

1. Crear el virtual environment

```
python -m venv venv
```

2. Activar el virtual environment

```
.\venv\Scripts\Activate.ps1
```

3. Instalar dependencias

```
pip install -r requirements.txt
```

4. Desde la carpeta dns-api:

```
py run.py
```

Queda corriendo en localhost:8080

Para documentacion de la API

localhost:8080/docs

Postman

<https://backend-7188.postman.co/workspace/My-Workspace~e4f81b51-5466-4cb1-8798-525845f98c0d/request/38941369-95a269ce-945d-418a-9239-39c6b09a5f7b?action=share&creator=38941369&ctx=documentation>

Explicacion breve

/api/exists: Nos dice si el hostname está registrado en la BD. Si está registrado, se manda a /resolve para que se resuelva según la estrategia (single, multi, weight, geo, rtt)

/dns_resolver: Aquí se manda la solicitud dns cuando, o no existe en la BD, o es de tipo no query

Y ahí deje unas crud basicas para records: donde estan los hostnames con sus ip, targets y datos.

BD

Colección **records**

Estructura de documentos DNS con algoritmos de balanceo de carga:

Campos básicos:

- **fqdn**: Nombre de dominio completo (ej: "single.example.com")
- **type**: Algoritmo de resolución ("single", "multi", "weight", "geo", "roundtrip")
- **ttl**: Tiempo de vida en segundos para cache
- **targets**: Array de servidores disponibles con id e ip

Tipos de algoritmos:

single: Un solo target fijo

```
{
  "fqdn": "single.example.com",
  "type": "single",
  "ttl": 60,
  "targets": [{ "id": "t1", "ip": "198.135.127.5" }]
}
```

multi: Round-robin entre targets saludables

- **rr_index**: Índice para rotación circular

weight: Distribución ponderada

- Cada target tiene **weight** (peso relativo)

geo: Selección por geolocalización

- Targets automáticamente geolocalizados por IP

roundtrip: Selección por menor latencia (RTT)

- Usa datos de health para medir tiempos de respuesta

Health data (subcollection):

- **status**: "healthy" o "unhealthy"
- **rtt.last_ms**: Última latencia medida
- **rtt.by_region**: RTT específico por región geográfica

Coleccion ip_to_country

- Esta colección contiene información sobre rangos de direcciones IP organizados por país.
- Cada documento representa un bloque de direcciones IP consecutivas con sus valores numéricos correspondientes para facilitar consultas de rango.
-
- @field country - Código de país ISO de 2 letras (ej: "IN" para India)
- @field start_ip - Dirección IP inicial del rango en formato string
- @field end_ip - Dirección IP final del rango en formato string

- @field range_start - Valor numérico de la IP inicial para consultas eficientes
- @field range_end - Valor numérico de la IP final para consultas eficientes

Importante, al registrar un record

Cuando se registra un record, y cada target, nada mas se tiene que registrar su ip, ya la funcion create record agarra la ip, y obtiene su pais y region de la BD (coleccion ip_to_country)

Nota, los endpoints de crud no estan testeados pero deberian funcionar