API de Países — Express.js + PostgreSQL

Esta API entrega información de países utilizando una base de datos local.
Replica los datos esenciales de la API usada en el Taller 1, pero no consulta RestCountries en tiempo real.

Los datos se inicializan desde `database.sql` y `countries-seed.json`.

Ejecución con Docker Compose

Desde la raíz del repositorio:

```
docker compose up --build
```

La API quedará disponible en:

http://localhost:4000

El puerto interno es 4000, pero puede sobrescribirse usando la variable PORT.

La base de datos se crea automáticamente con database.sql.
Al iniciar, server.js revisa cuántos países hay; si son menos de 30, vuelve a cargar datos desde countries-seed.json.

Endpoints

GET /countries

Devuelve el listado completo de países.

Ejemplo de respuesta:

```
[
  {
    "name": { "common": "Argentina", "official": "Argentine Republic" },
    "flags": {
      "png": "https://flagcdn.com/w320/ar.png",
      "svg": "https://flagcdn.com/ar.svg"
    },
    "region": "Americas",
    "capital": ["Buenos Aires"],
    "population": 45376763,
    "cca2": "AR"
  }
]
```

GET /countries/search?name=

Busca países cuyo name.common contenga el texto indicado (búsqueda case-insensitive).

Ejemplo de uso:

http://localhost:4000/countries/search?name=chile

Ejemplo de respuesta:


```
[
  {
    "name": { "common": "Chile", "official": "Republic of Chile" },
    "flags": {
      "png": "https://flagcdn.com/w320/cl.png",
      "svg": "https://flagcdn.com/cl.svg"
    },
    "region": "Americas",
    "capital": ["Santiago"],
    "population": 19116201,
    "cca2": "CL"
  }
]
```

Datos utilizados

database.sql → crea la tabla y carga datos base

countries-seed.json → semilla usada como fallback

Campos almacenados:

```
common_name

official_name

cca2

region

capital

population

flag_png

flag_svg
```