

BackEndChallenge

[Enunciado](#)

[Contratos entrada](#)

[Contrato salida](#)

[Base de Datos](#)

[Consideraciones](#)

Enunciado

Dada la siguiente API <https://earthquake.usgs.gov/fdsnws/event/1/> la cual obtiene información sobre terremotos, se desea crear un servicio en Python utilizando **Flask** con los siguientes endpoints, con contratos de entrada y salida formato json, donde cada endpoint deberá hacer uso la API mencionada para obtener la información solicitada (llamar a API según especificación de la misma).

Contratos entrada

1. **getEarthquakesByDates**: Consultar terremotos entre dos fechas y una magnitud mínima

- a. Contrato de entrada:

```
{
  "fechaInicio": "2020-01-01",
  "fechaFin": "2020-01-02",
  "magnitudMinima": "6.5"
}
```

2. **getEarthquakesByMagnitudes**: Consultar terremotos entre dos magnitudes

- a. Contrato de entrada

```
{
  "magnitudMinima": "6.5",
  "magnitudMaxima": "7.1"
}
```

Contrato salida

En ambos casos deberá retornar el siguiente contrato de salida:

```
{
  "mag": 1.1,
  "place": "117km NW of Talkeetna, Alaska",
  "time": "Wednesday, January 1, 2014 11:47:26.501 PM",
  "updated": "Monday, May 20, 2019 10:45:30.681 PM",
  "alert": null,
  "status": "reviewed",
  "tsunami": 0,
  "magType": "ml",
  "type": "earthquake",
  "title": "M 1.1 - 117km NW of Talkeetna, Alaska"
}
```

Base de Datos

El resultado debe ser almacenado en una base de datos relacional, cuya tabla debe ser (se recomienda BD embebida):

earthquake	
campo	descripción
create_at	Llave primaria, fecha de insercion registro, valor por defecto
fecha_inicio	Fecha inicio búsqueda
fecha_fin	Fecha fin búsqueda
magnitud_min	Magnitud minima busqueda
magnitud_max	Magnitud maxima busqueda
salida	Resultado busqueda (json)

Consideraciones

- Tu primer commit debe ser un código fuente que haga funcionar tu sistema
- Tu segundo commit debe ser un refactoring que mejora el primer código
- Deberás validar datos de entrada para cada endpoint
- Manejo de errores
- Manejo de excepciones
- Manejo de logs: info y debug
- Deberás realizar test unitarios que consideres necesarios para tus endpoints
- Deberás subir tu código hacia algun repositorio (Github, Bitbucket, etc) en una rama llamada feature/creating-endpoints
 - Debe contener el archivo Readme.md especificando los requerimientos, construcción, empaquetado y despliegue.
- Opcionalmente el servicio debe poder correr en el puerto 8099 y path inicial searchEarthquake, por ejemplo: <http://localhost:8099/searchEarthquake/><endpoint>

Éxito!!