

## Documentación API biblioteca

Lo primero que tenemos que hacer es instalar Flask en nuestra máquina, se hace ejecutando los siguientes comandos:

```
>> py -3 -m venv venv
```

```
>> venv\Scripts\activate
```

```
>> pip install Flask
```

La primera función pública que nos encontramos en el main.py es “*creaDB*”, que nos servirá para crear la base de datos dinámicamente. No recibe ningún parámetro, pero será necesario que la estructura de nuestros archivos se corresponda a la siguiente imagen:

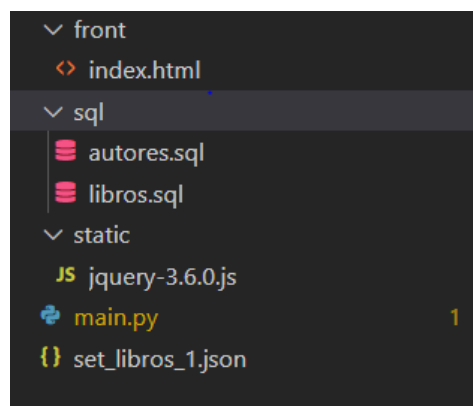


Ilustración 1: Estructura de archivos

La función anterior se encargará de llamar a las funciones “*creaLibros*” y “*creaAutores*”, que se encargarán de insertar los libros y los autores respectivamente en nuestra base de datos.

Para ejecutar una función independientemente de la ruta usaremos jquery.

```
function enviaInfo(){
    $.ajax({
        type: 'POST',
        url: '/test2', //Metodo a ejecutar
        async: false,
        data: {mydata: "dataSended"} //Datos a enviar
    })
}
```

Ilustración 2: Función del HTML para enviar parámetros a las funciones

Será necesario adjuntar este script a nuestro HTML

```
<script src="{{ url_for('static',filename='jquery-3.6.0.js')}}"></script>
```

Ilustración 3: Script enlazado al HTML

A continuación, tenemos la función *"nuevoRegistroPorArchivo"*. A esta función le mandamos como parámetro un archivo con formato .sql y este archivo será ejecutado en nuestra base de datos. Si el archivo que le pasamos no es un .sql nos dará una advertencia y no ejecutará nada.

La siguiente función que nos encontramos es *"nuevoRegistroPorTexto"*, a esta función se le manda como parámetro una cadena de texto que será ejecutada como una sentencia sql en nuestra BD. Si queremos ejecutar varias sentencias tendremos que separarlas por *";"*. Por ejemplo: *nuevoRegistroPorTexto (select \* from autores ; select \* from libros)*

La siguiente función es *"eliminarLibro"*. A esta se le manda como parámetro una cadena que debe ser el id de algún libro almacenado en nuestra base de datos. Lo que hace esta función es eliminar de la bd el libro con el id que le hayamos pasado.

Después tendremos la función *"realizarBusqueda"*. Como su nombre indica, esta función sirve para encontrar un libro guardado en la bd. Se le manda como parámetro un string que puede ser el id de un libro o el nombre de un autor. Si mandamos el id de un libro nos va a devolver toda la información de ese libro (autor, título, editorial, lugar, fecha y descripción) y si mandamos un autor nos va a devolver el título, la editorial, el lugar, la fecha y la descripción de todos los libros de ese autor.

A continuación, encontramos la función *"addRegistro"*. Esta función se encarga de insertar un libro creado por un autor. Tendrá que recibir como parámetros un nombre para el autor, lugar de nacimiento, id del libro (–Letra a en minúscula seguida de 7 números–, por ejemplo: *"a1234567"*) título del libro, editorial, lugar, fecha y descripción.

Por último, tenemos la función *"editarLibro"*. Esta función como su nombre indica es para editar un libro. Recibe como parámetro el id del libro a editar y los nuevos datos del libro que serían autor, título, editorial, lugar, fecha y descripción