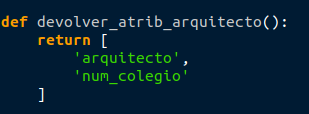
Documentación APIRes en Python

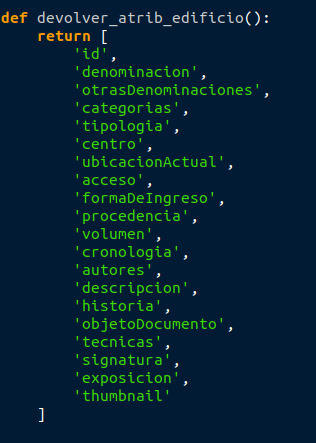
Tecnologías utilizadas:

* Flask
* Sqlite3
* Json (módulo importado).

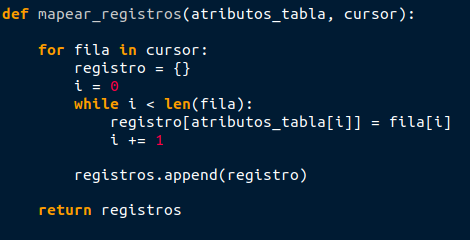
\* Base de datos utilizada es de arquitectos y edificios. \*



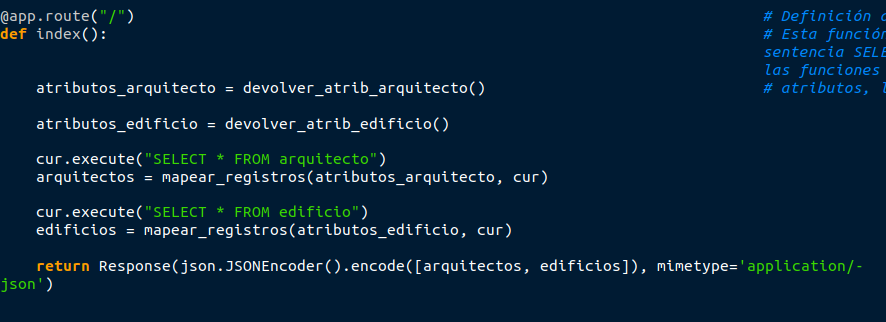
Función dedicada a la devolución de los atributos de la tabla “arquitecto”.



Función dedicada a la devolución de los atributos de la tabla “edificio”.

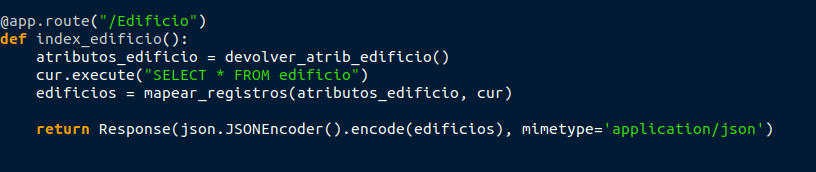


Función dedicada a mapear o mostrar en la web la información de los registros o tablas.



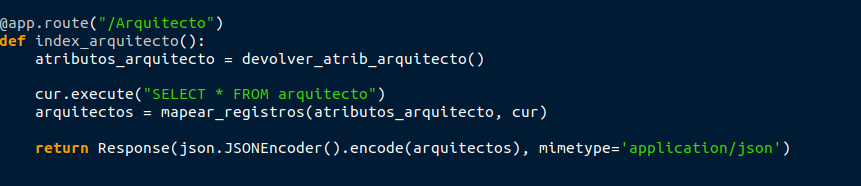
Definimos la ruta raíz, o path. (@app.route(“/”) )

La función index() va a mostrarnos mediante la sentencia Sql SELECT y asistencia de las funciones previamente explicadas, los datos de los registros de “arquitecto” y “edificio”.



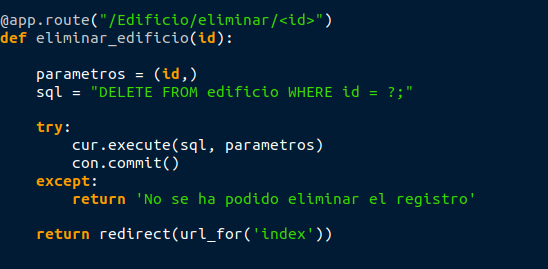
Definimos la ruta Edificio, la cual nos va a contener toda la información de los registros en referencia.

La función index\_edificio nos va a mostrar los datos e información del edificio, usando la sentencia Sql SELECT y las funciones de devolución de atributos.



Definimos la ruta Arquitecto, va a contener toda la información de la tabla arquitecto.

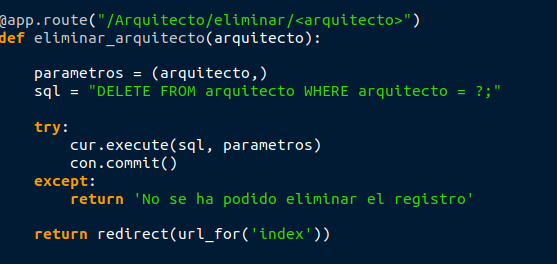
La función index\_arquitecto nos va a mostrar los datos e información del arquitecto, con el uso de la sentencia Sql SELECT y las funciones de devolución de atributos.



Definimos la ruta Eliminar Edificio, aquí vamos a eliminar los registros por filas.

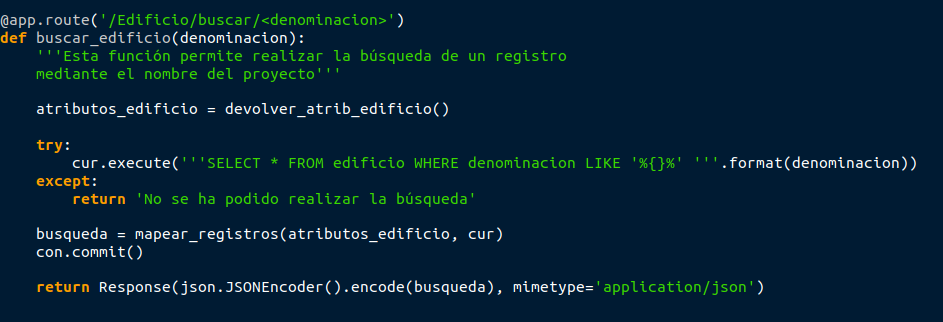
La función eliminar\_edificio eliminará los registros por id, empleando la sentencia Sql DELETE.

Por último, se ofrecerá un feedback mínimo mediante el uso de un sistema de excepciones.



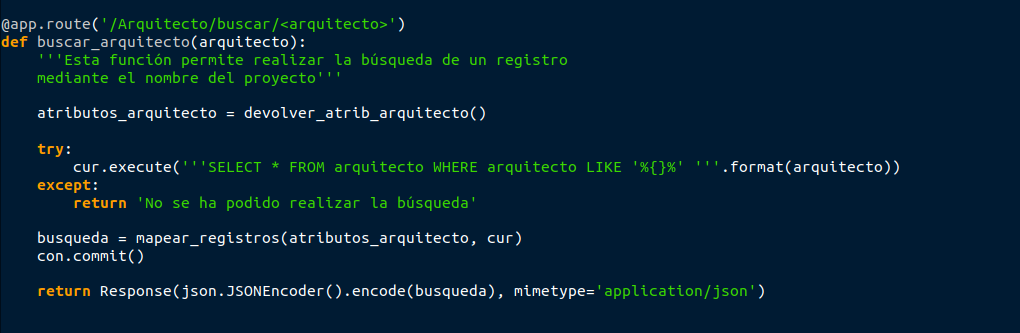
Definimos la ruta Eliminar Arquitecto, aquí vamos a eliminar los registros por filas, tal y como lo planteamos en la función anterior.

La función eliminar\_arquitecto va a eliminar los registros por id, empleando nuevamente la sentencia Sql DELETE.



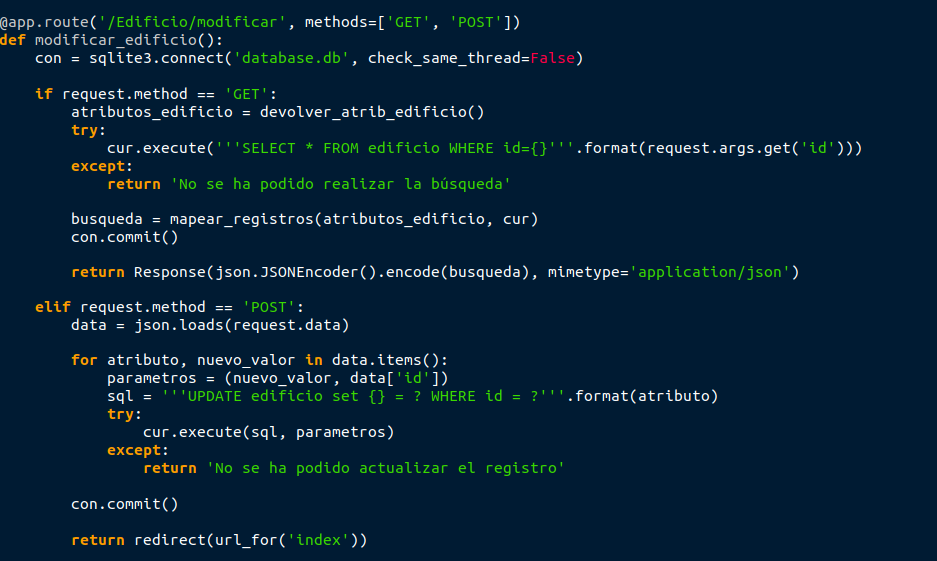
Definimos la ruta Buscar Edificio, que se va a encargar de propiamente buscarnos el edificio que queramos.

La función buscar\_edificio nos va buscar un registro mediante el nombre del proyecto. Se hará uso de la sentencia Sql SELECT para indizar más específicamente.



Definimos la ruta Buscar Arquitecto, que se va a encargar de propiamente buscar registros de arquitecto.

La función buscar\_arquitecto nos va buscar un registro mediante el nombre del proyecto. Se empleará la sentencia Sql SELECT para indizar más específicamente, casi al igual que la función anterior.

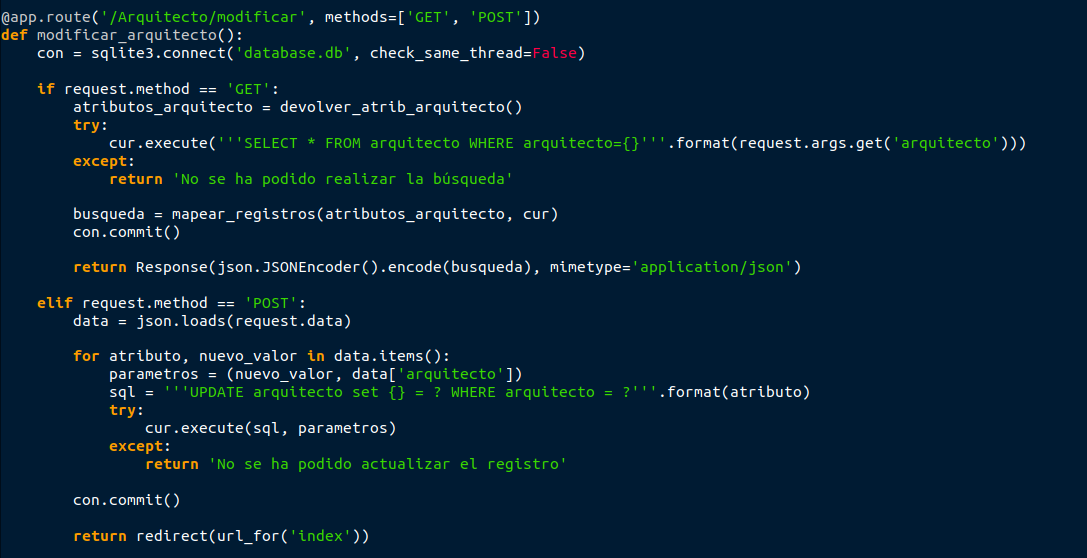


Definimos la ruta Modificar Edificio, esta ruta nos va a permitir cambiar los valores de los atributos de una fila de edificio.

Nos apoyaremos en los verbos del protocolo HTTP, usando GET y POST.

GET nos añadirá los datos codificados a la URL, y POST añadirá los datos al cuerpo de la web, no a la URL.

Cuando se ejecute la sentencia Sql UPDATE, se actualizarán los datos de la BD y nos va a redirigir al index.



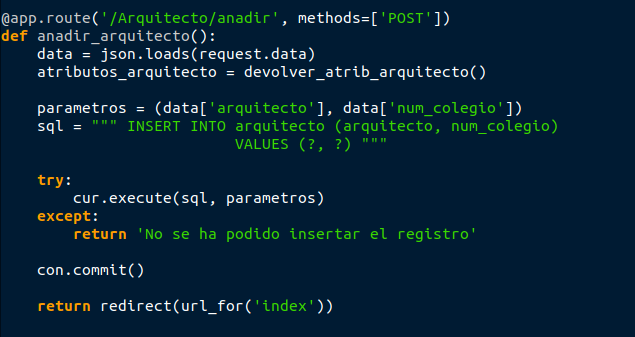
Definimos la ruta Modificar Arquitecto, esta ruta nos va a permitir cambiar los valores de los atributos de una fila de arquitecto.

Nos apoyaremos en los verbos del protocolo HTTP, usando GET y POST.

GET nos añadirá los datos codificados a la URL, y POST añadirá los datos al cuerpo de la web, no a la URL.

Cuando se ejecute la sentencia Sql UPDATE, se actualizarán los datos de la BD. Nos devolverá una vista del JSON con los valores actualizados. Nos va a modificar los valores en el cuerpo, pero no en la URL.

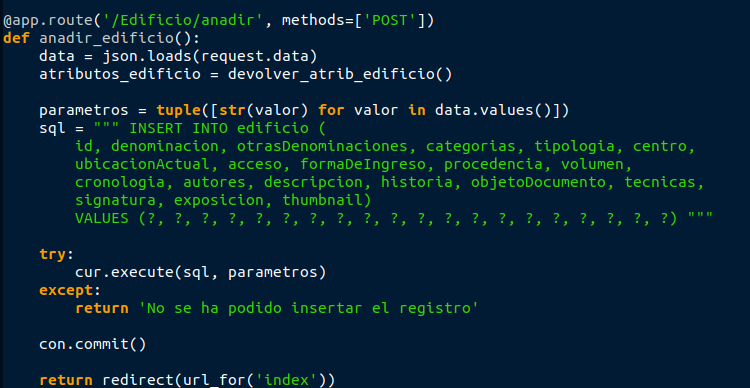
Nos va a redirigir al index.



Definimos la ruta Añadir Arquitecto, la cual nos permitirá realizar adiciones de filas de la tabla arquitecto.

Ejecutaremos la sentencia Sql INSERT para darle valor a los atributos de arquitecto.

Se nos redirigirá al index.



Definimos la ruta Añadir Edificio, la cual nos permitirá realizar adiciones de filas de la tabla edificio.

Ejecutaremos la sentencia Sql INSERT para darle valor a los atributos de edificio.

Se nos redirigirá al index.