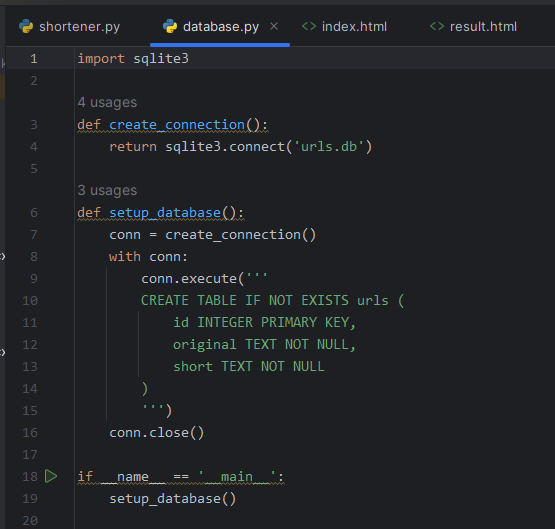
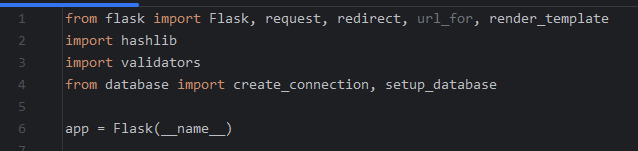
Acortador de URLS Proyecto:

database.py  


Creando un archivo de base de datos (SQL) y creando una tabla llamada urls, con columnas id, original y short (acortada). Esto se hace primero importando el módulo sqlite3 y luego creando la tabla utilizando las funciones que se observan.

shortener.py  


Importamos varios módulos. El módulo flask es un framework web, las funciones que se importan son:

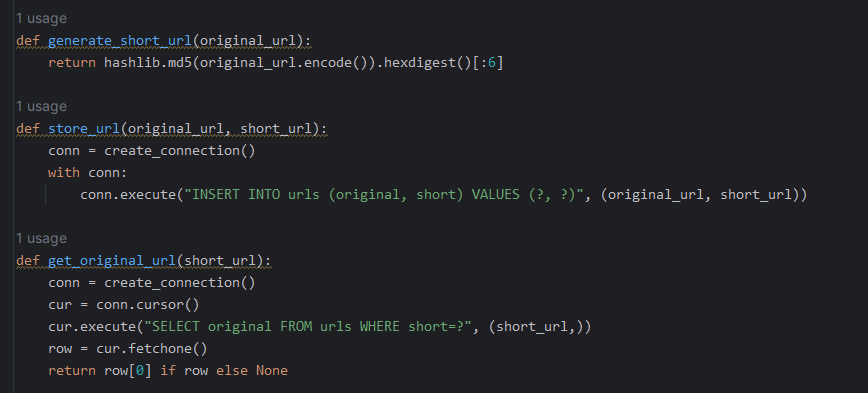
* Flask: Clase principal de Flask, se utiliza para crear instancias de apps web.
* Request: un objeto que contiene los datos de solicitud HTTP entrante, como los datos del formulario, encabezados, etc.
* Redirect: una función que redirige la solicitud a una URL diferente, usada para este caso específicamente donde se buscaba acortar URLS.
* url\_for: función que genera URLS para las rutas definidas en la aplicación Flask.
* Render\_template: Una función que renderiza plantillas HTML con datos dinámicos.

Módulo hashlib: proporciona una interfaz segura para hashing de datos, lo que se utiliza para almacenar contraseñas de forma segura en una base de datos.

Módulo validators: este módulo contiene funciones personalizadas para validar los datos de entrada del usuario, aunque no es común en Python.

From database import: aquí importamos funciones del archivo database que creamos anteriormente. Es para crear una conexión con la base de datos y configurarla.

App = Flask(\_\_name\_\_) crea una instancia de la aplicación Flask, pasando \_\_name\_\_ como argumento. \_\_name\_\_ es una variable incorporada de Python que representa el módulo actual, lo que permite que Flask localice los recursos de la aplicación correctamente.



La primera función generate\_short\_url(original\_url) utiliza el argumento original\_url, que serían las URL original que vienen del input del usuario y genera una URL acortada utilizando el algoritmo de hash MD5 de hashlib. Haslib.md5(original\_url.encode()) calcula el hash MD5 de la URL original codificada en bytes y .hexdigest()[0:6] convierte el hash en una representación hexadecimal y toma los primeros 6 caracteres, que se utilizará para la URL corta.

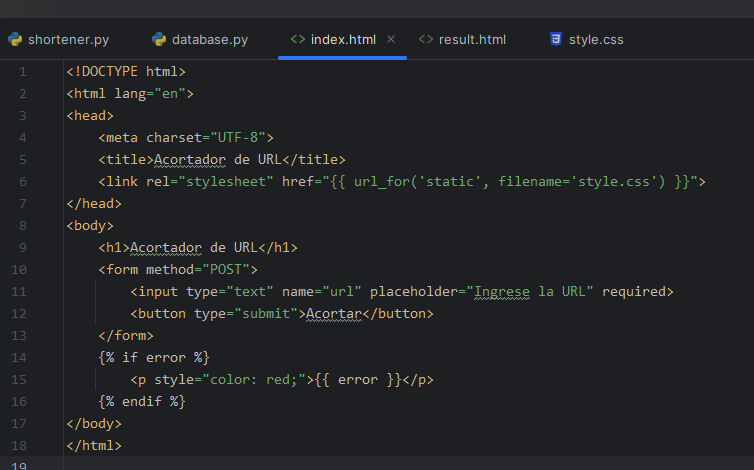
La función store\_url(original\_url, short\_url): almacena una URL original y su URL acortada en la base de datos con los argumentos original\_url, short\_url. conn = create\_connection() establece una conexión a la base de datos utilizando la función create\_connection. with con: abre un contexto para ejecutar consultas en la conexión de forma segura. conn.execute… inserta unanueva fila en la tabla urls con las nuevas URL, la ingresada y la acortada.

La función get\_original\_url(short\_url) recupera la URL original apartir de una URL acortada dada. Primero se conecta a la base de datos y luego se crea un cursor para crear consultas, luego con cur.execute se consulta la talba urls para obtener la URL original correspondiente a la clave (URL acortada) dada. row = cur.fetchone() recupera la primera fila del resultado de la consulta y return row[0] if row else None devuelve la URL original si se encontró una coincidencia o None si no se encontró. 

En @app.route(‘/’, methods=[‘GET’, ‘POST’]) se define una ruta en la aplicación Flask. Cualquier solicitud HTTP a la raíz / será manejada por la función index.

La función index maneja las solicitudes a la raíz. Si request.method == ‘POST’, significa que se envío un formulario, original\_url = request.form[‘url’] obtiene la URL original del formulario. Si if not validators.url(original\_url): verifica si la URL original es válida utilizando validators.url. Si la URL no es válida, se renderiza el template ‘index.html’ con un mensaje de error, si la URL es válida, se genera una URL acortada (short\_url) utilizando generate\_short\_url(original\_url). Se almacena la URL original y la acortada en la base de datos con store\_url(original\_url, short\_url). Se renderiza el template result.html con la URL acortada. Si request.method == ‘GET’, simplemente se renderiza el template ‘index.html’.

@app.route(‘/<short\_url>’): esta ruta maneja las solicitudes a URLS acortadas. La función redirect\_to\_url(short\_url) redirige a la URL original apartir de la URL acortada. original\_url = get\_original\_url(short\_url) obtiene la URL original a partir de la URL acortada. Si se encuentra una URL original, se redirige a esa URL utilizando return redirect(original\_url). Si no se encuentra una URL original, se devuelve un error 404. if \_\_name\_\_ == ‘\_\_main\_\_’: Esta comprobación se utiliza para ejecutar el código solo cuando el script se ejecuta directamente. Setup\_database() configura la base de datos y app.run(debug=True) inicia el servidor de desarrollo de Flask en modo de depuración.

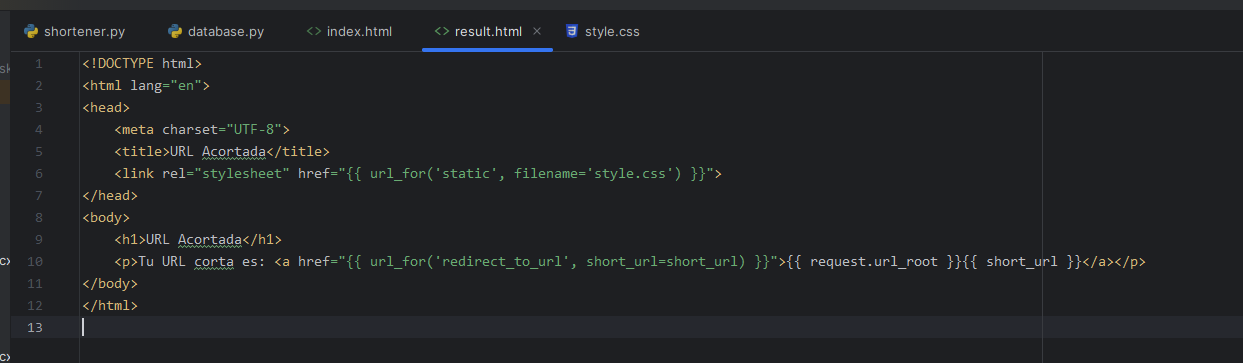


<meta charset = “UTF-8”> define la codificación de caracteres del documento como UTF-8.

<link rel=”stylesheet” href=”{{ url\_for(‘static’, filename= ‘style.css’)}}”>: Esta línea enlaza una hoja de CSS ubicada en la carpeta static. La función url\_for de Flask se utiliza para generar la URL correcta para el archivo CSS.

<form method=”POST”>: Inicia un formulario HTML con el método POST. <input type=”text” name=”url” placeholder=”Ingrese la URL” required>: Un campo de entrada de texto donde el usuario ingresa la URL deseada. El atributo required indica que este campo es obligatorio. <button type= “submit”>Acortar</button>: Un botón de envío para el formulario.

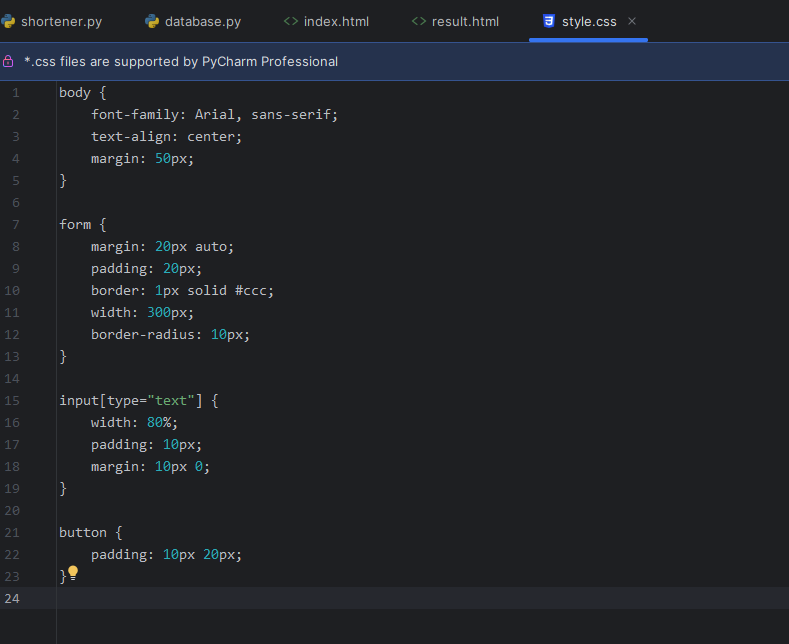
{% if error %}: Esta es una declaración de plantilla Jinja2 que se utiliza en Flask. Verifica si existe una variable error. <p style”color: red;”>{{ error }} </p>: Si error existe, muestra un párrafo con el valor de error en color rojo. {% endif %} cierra la declaración condicional de Jinja2.



Se enlaza el Código de style.css.

Luego <p> Tu URL corta es: <a href=”{{ url\_for(‘redirect\_to\_url’, short\_url=short\_url) }}”>{{ request.url\_root }}{{ short\_url }}</a></p> este párrafo muestra la URL acortada generada. {{ url\_for(‘redirect\_to\_url’, short\_url= short\_url) }} Esta expression Jinja2 genera la URL para la ruta redirect\_to\_url pasando la variable short\_url como argumento.

{{ request.url\_root }}: muestra la raíz de la URL de la aplicación Flask; {{short\_url }} Muestra la URL acortada generada. Toda esta expresión está envuelta en un enlace <a> que permitirá al usuario hacer clic en la URL acortada y redirigirlo.



El estilo de la página en CSS.