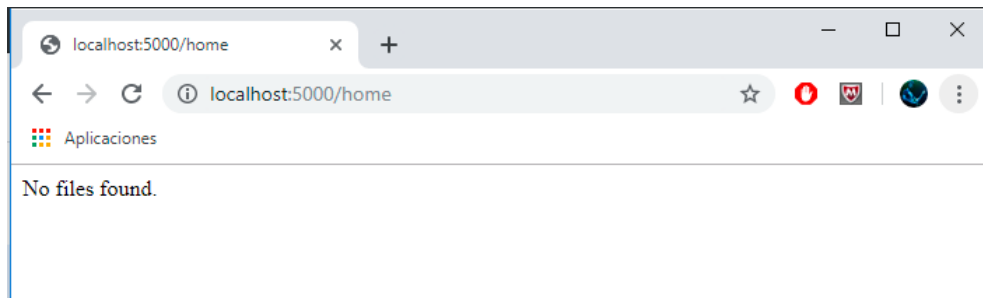


Caso	Accion	Input	Output esperado	Output real
1	Ejecutar API	python api.py	Running on http://0.0.0.0:5000/	Running on http://0.0.0.0:5000/
2	Listar documentos en drive vacio	GET http://localhost:5000/home	GET /home HTTP/1.1 200 y mensaje 'No files found' en el explorador	GET /home HTTP/1.1 200 y mensaje 'No files found' en el explorador
3	Intentar crear documento con parametros erroneos	curl -i -X POST -H Content-Type:application/json -d {"name":"TITULO", "description":""} http://localhost:5000/create	'404 Not Found' y 'HTTP/1.1" 500'	'404 Not Found' y 'HTTP/1.1" 500'
4	Crear documento con contenido	curl -i -X POST -H Content-Type:application/json -d {"name":"Test1", "description":"Contenido de prueba"} http://localhost:5000/create	{"id":"1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc", "titulo":"Test1", "descripcion":"Contenido de prueba"} y "POST /create HTTP/1.1" 200	{"id":"1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc", "titulo":"Test1", "descripcion":"Contenido de prueba"} y "POST /create HTTP/1.1" 200
5	Listar documentos	GET http://localhost:5000/home	Test1 (1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc)	Test1 (1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc)
6	Buscar palabra que existe	GET http://localhost:5000/search-in-doc/1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc?word=Contenido	Keyword found in requested file' y 'HTTP/1.1" 200 '	Keyword found in requested file' y 'HTTP/1.1" 200 '
7	Buscar palabra que no existe	GET http://localhost:5000/search-in-doc/1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1fkcc?word=asd	HTTP/1.1" 404	HTTP/1.1" 404
8	Buscar palabra con parametros erroneos	GET http://localhost:5000/search-in-doc/1eLDRtdS-FPnLOvZQGIUZ-fMbdMwElcPnaWeQke1	Error - Enter the parameters correctly	Error - Enter the parameters correctly

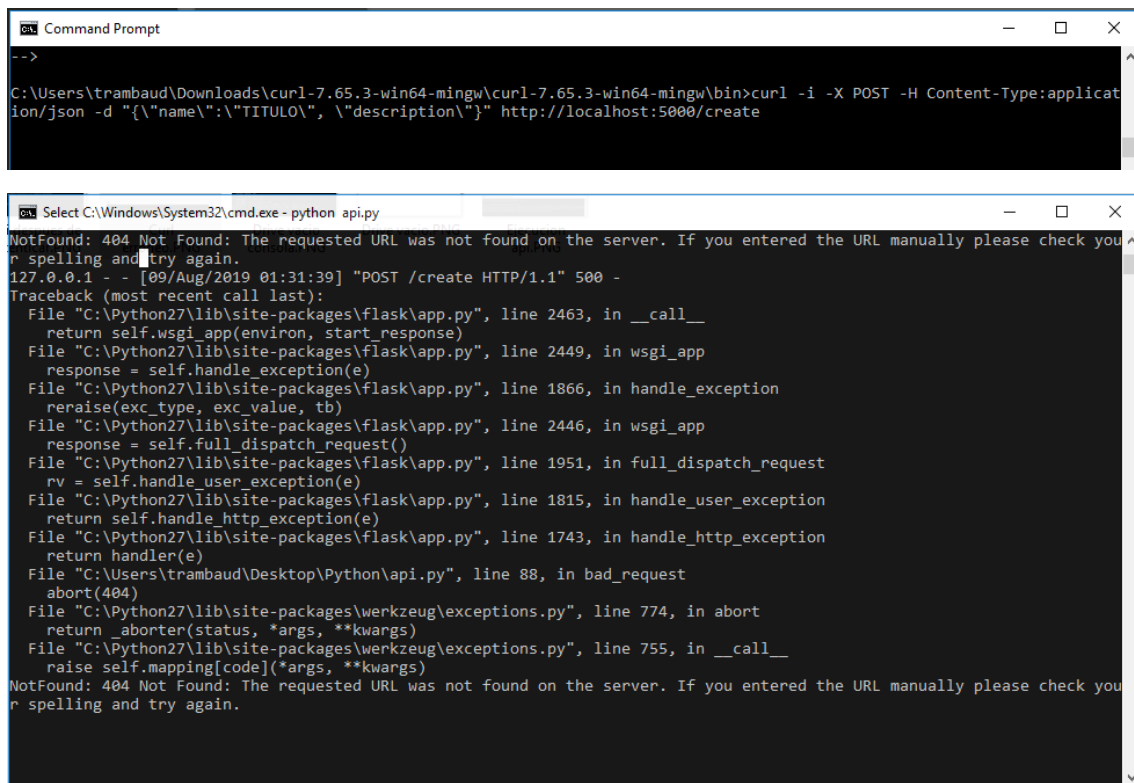
Caso 1)

```
C:\Users\trambaud\Desktop\Python>python api.py
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 116-483-120
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Caso 2)



Caso 3)



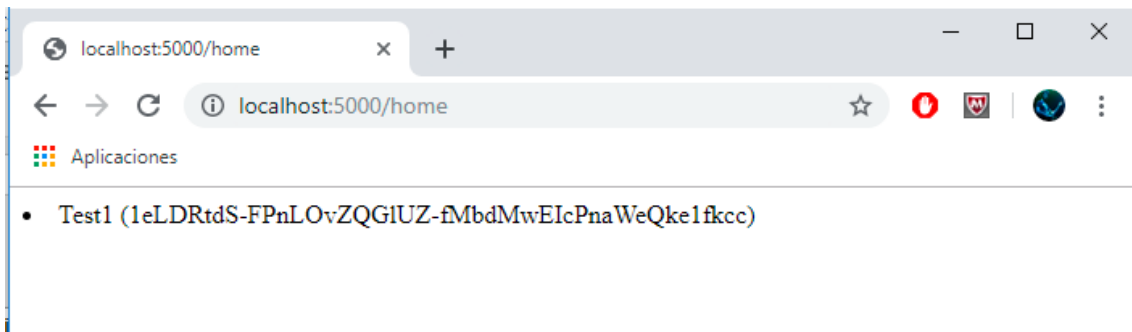
Caso 4)

```
Command Prompt
C:\Users\trambaud\Downloads\curl-7.65.3-win64-mingw\curl-7.65.3-win64-mingw\bin>curl -i -X POST -H Content-Type:application/json -d '{"name":"Test1", "description":"Contenido de prueba"}' http://localhost:5000/create
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 108
Server: Werkzeug/0.15.5 Python/2.7.15
Date: Fri, 09 Aug 2019 04:38:44 GMT

{"id":"1eLDRtdS-FPnLOvZQGIUZ-fMbdMwEicPnaWeQke1fkcc", "titulo":"Test1", "descripcion":"Contenido de prueba"}
C:\Users\trambaud\Downloads\curl-7.65.3-win64-mingw\curl-7.65.3-win64-mingw\bin>
```

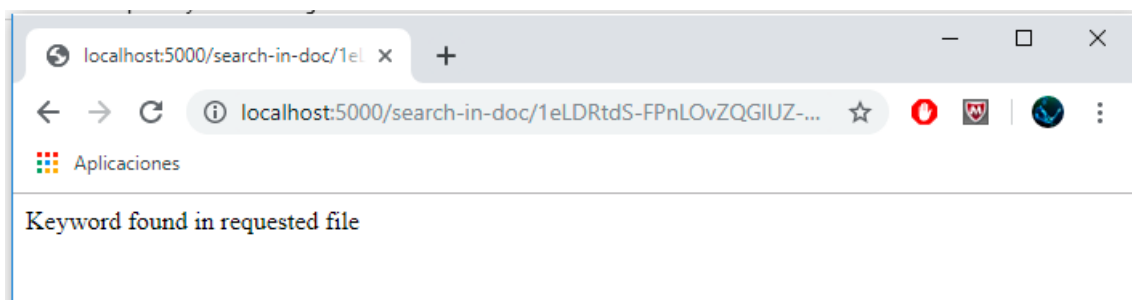
```
C:\Users\trambaud\Desktop\Python>python api.py
* Serving Flask app "api" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 116-483-120
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [09/Aug/2019 01:38:44] "POST /create HTTP/1.1" 200 -
```

Caso 5)

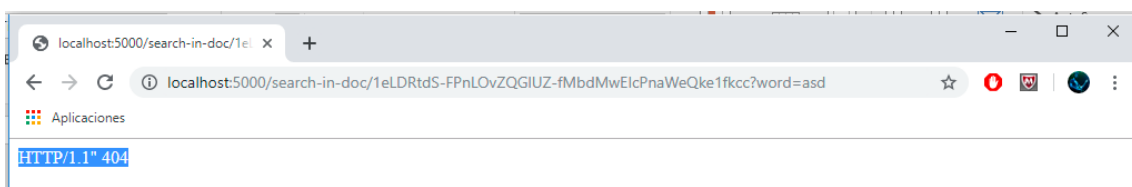


Caso 6)

```
127.0.0.1 - - [09/Aug/2019 01:41:38] "GET /home HTTP/1.1" 200 -
127.0.0.1 - - [09/Aug/2019 01:46:21] "GET /search-in-doc/1eLDRtdS-FPnLOvZQGIUZ-fMbdMwEicPnaWeQke1fkcc?word=Contenido HTTP/1.1" 200 -
```



Caso 7)



Caso 8)

