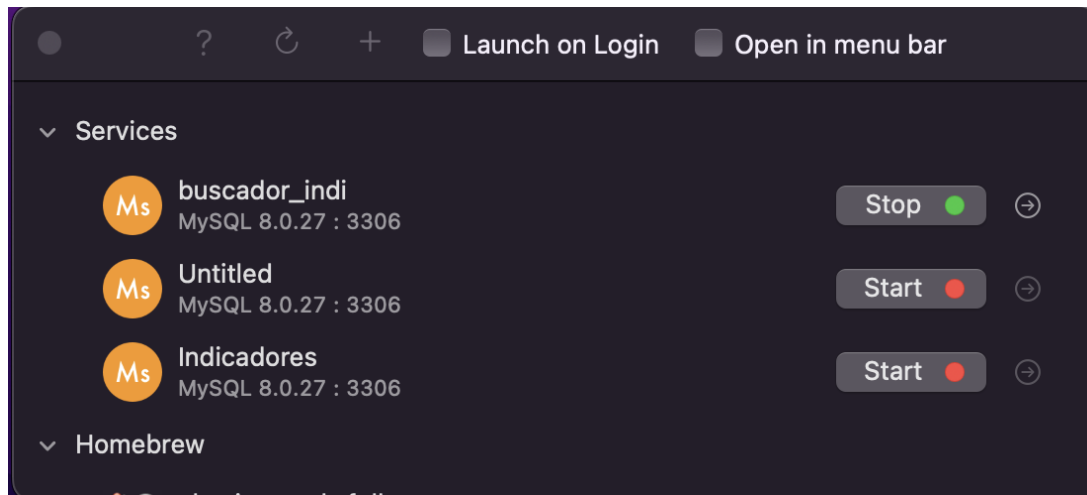


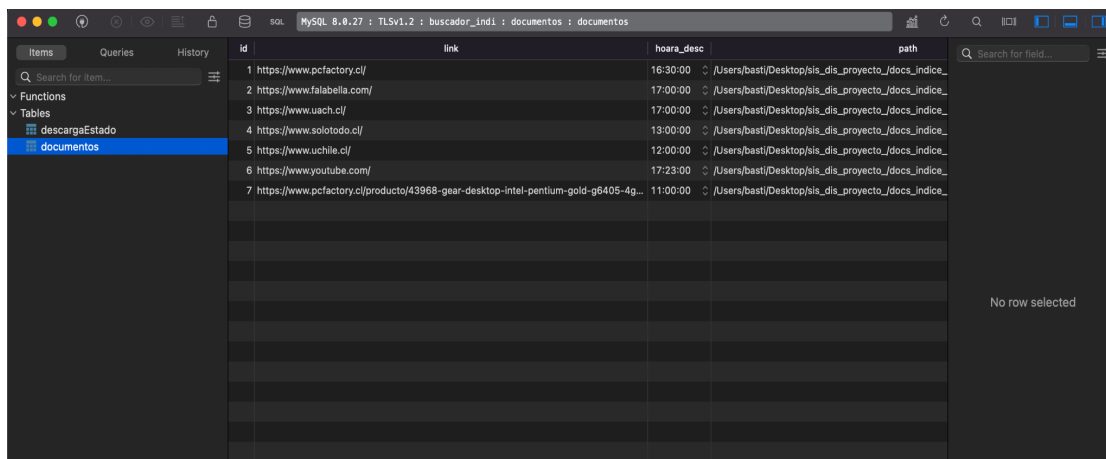
despliegue manual

Base de datos

- 1) tener nuestra base de datos corriendo en el puerto:3306, ip:localhost



- 2) tener datos en la tabla documentos



Nota: en la carpeta **Maquina_virtual_3** se encuentra el archivo.sql para cargar los datos

elasticsearch

- 1) abrimos una terminal independiente.
- 2) nos dirigimos a la máquina virtual 2 donde se encuentra la carpeta elasticsearch-8.7.1.
- 3) una vez, ya posicionado escribimos el siguiente comando `./bin/elasticsearch`

```
{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1} completing election], term: 14, version: 174, delta: master node changed {previous [], current [{MacBook-Pro-de-Bastian.local}{9Se6YX5ySs-B2eAZpWxKrQ}{doS0c8TsQ0K2eeU0Z4A_8A}{MacBook-Pro-de-Bastian.local}{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1}}]
[2023-05-19T15:55:36,128][INFO ][o.e.c.s.ClusterApplierService] [MacBook-Pro-de-Bastian.local] master node changed {previous [], current [{MacBook-Pro-de-Bastian.local}{9Se6YX5ySs-B2eAZpWxKrQ}{doS0c8TsQ0K2eeU0Z4A_8A}{MacBook-Pro-de-Bastian.local}{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1}}], term: 14, version: 174, reason: Publication{term=14, version=174}
[2023-05-19T15:55:36,141][INFO ][o.e.r.s.FileSettingsService] [MacBook-Pro-de-Bastian.local] starting file settings watcher ...
[2023-05-19T15:55:36,143][INFO ][o.e.r.s.FileSettingsService] [MacBook-Pro-de-Bastian.local] file settings service up and running [tid=77]
[2023-05-19T15:55:36,145][INFO ][o.e.c.c.NodeJoinExecutor ] [MacBook-Pro-de-Bastian.local] node-join: [{MacBook-Pro-de-Bastian.local}{9Se6YX5ySs-B2eAZpWxKrQ}{doS0c8TsQ0K2eeU0Z4A_8A}{MacBook-Pro-de-Bastian.local}{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1}] with reason [completing election]
[2023-05-19T15:55:36,146][INFO ][o.e.h.AbstractHttpServerTransport] [MacBook-Pro-de-Bastian.local] publish_address {10.244.177.221:9200}, bound_addresses {[::]:9200}
[2023-05-19T15:55:36,146][INFO ][o.e.n.Node ] [MacBook-Pro-de-Bastian.local] started [MacBook-Pro-de-Bastian.local]{9Se6YX5ySs-B2eAZpWxKrQ}{doS0c8TsQ0K2eeU0Z4A_8A}{MacBook-Pro-de-Bastian.local}{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1}{ml.max_jvm_size=4294967296, ml.allocated_processors_double=8.0, xpack.installed=true, ml.machine_memory=8589934592, ml.allocated_processors=8}
[2023-05-19T15:55:36,301][INFO ][o.e.l.LicenseService ] [MacBook-Pro-de-Bastian.local] license [57b60f11-6e62-4b62-9832-d62368dd8e73] mode [basic] - valid
[2023-05-19T15:55:36,303][INFO ][o.e.g.GatewayService ] [MacBook-Pro-de-Bastian.local] recovered [2] indices into cluster state
[2023-05-19T15:55:36,537][INFO ][o.e.h.n.s.HealthNodeTaskExecutor] [MacBook-Pro-de-Bastian.local] Node [{MacBook-Pro-de-Bastian.local}{9Se6YX5ySs-B2eAZpWxKrQ}] is selected as the current health node.
[2023-05-19T15:55:36,539][INFO ][o.e.c.r.a.AllocationService] [MacBook-Pro-de-Bastian.local] current.health="YELLOW" message="Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[db_scrapper][0], [.security-7][0]]])." previous.health="RED" reason="shards started [[db_scrapper][0], [.security-7][0]]"
```

Nota: el *elasticsearch* está corriendo en el puerto definido.

BackEnd

- 1) abrimos una terminal
- 2) nos dirigimos a la máquina virtual 2 donde se encuentra la carpeta backEnd
- 3) para levantar en backEnd se escribe el segundo comando `python3 main.py`

```
[2023-05-19T15:55:36,146][INFO ][o.e.h.AbstractHttpServerTransport] [MacBook-Pro-de-Bastian.local] publish_address {10.244.177.221:9200}, bound_addresses {[::]:9200}
[2023-05-19T15:55:36,146][INFO ][o.e.n.Node ] [MacBook-Pro-de-Bastian.local] started [MacBook-Pro-de-Bastian.local]{9Se6YX5ySs-B2eAZpWxKrQ}{doS0c8TsQ0K2eeU0Z4A_8A}{MacBook-Pro-de-Bastian.local}{127.0.0.1}{127.0.0.1:9300}{cdfhilmrstw}{8.7.1}{ml.max_jvm_size=4294967296, ml.allocated_processors_double=8.0, xpack.installed=true, ml.machine_memory=8589934592, ml.allocated_processors=8}
[2023-05-19T15:55:36,301][INFO ][o.e.l.LicenseService ] [MacBook-Pro-de-Bastian.local] license [57b60f11-6e62-4b62-9832-d62368dd8e73] mode [basic] - valid
[2023-05-19T15:55:36,303][INFO ][o.e.g.GatewayService ] [MacBook-Pro-de-Bastian.local] recovered [2] indices into cluster state
[2023-05-19T15:55:36,537][INFO ][o.e.h.n.s.HealthNodeTaskExecutor] [MacBook-Pro-de-Bastian.local] Node [{MacBook-Pro-de-Bastian.local}{9Se6YX5ySs-B2eAZpWxKrQ}] is selected as the current health node.
[2023-05-19T15:55:36,539][INFO ][o.e.c.r.a.AllocationService] [MacBook-Pro-de-Bastian.local] current.health="YELLOW" message="Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[db_scrapper][0], [.security-7][0]]])." previous.health="RED" reason="shards started [[db_scrapper][0], [.security-7][0]]"

basti@MacBook-Pro-de-Bastian backEnd % python3 main.py
INFO: Started server process [26428]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
```

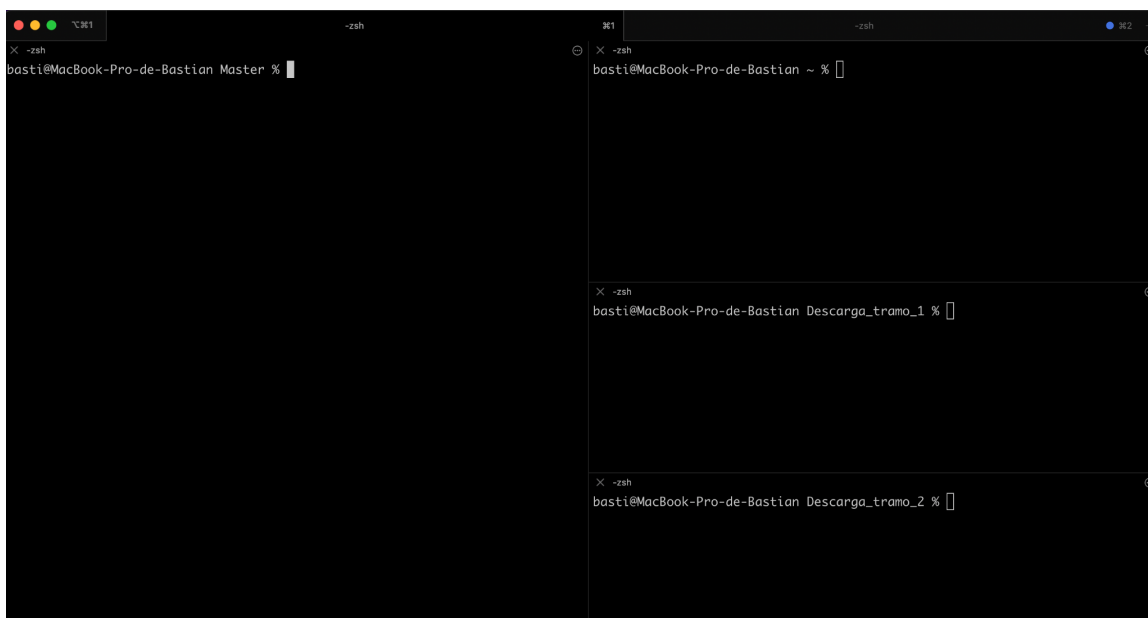
Nota: se levanta el backend exitosamente. cabe mencionar que esta corriendo en el puerto 8000, ya que en maquina donde se esta realizando el levantamiento esta ocupado el puerto 5000.

se necesita hacer un paso extra para activar la base de datos del *elasticsearch*

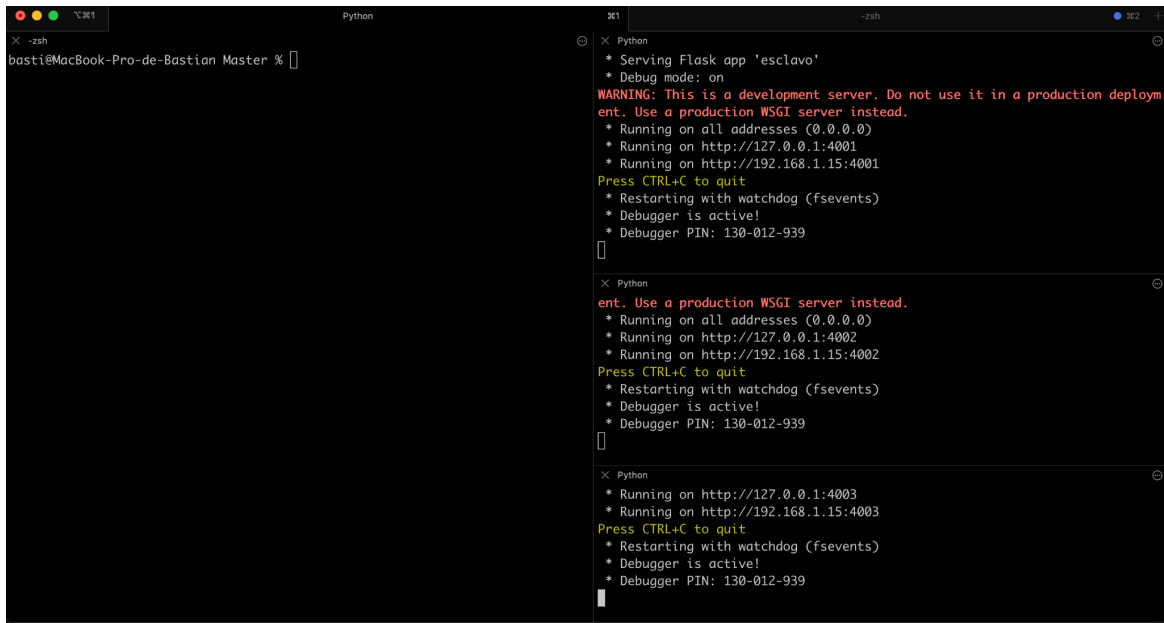
- 1) es a un navegador y en la barra de busqueda escribir la siguiente petición:
<http://0.0.0.0:8000/api/elasticsearch/create>

Master Descarga más Descarga tramos

- 1) abrir 4 terminales las cuales tiene que estar en la siguientes rutas
 - Master <- /docs_indice_invertido/Maquina_virtual_4/Master
 - esclavo1 <- /docs_indice_invertido/Maquina_Virtual_5.0/Descarga_tramo_1
 - esclavo2 <- /docs_indice_invertido/Maquina_Virtual_5.1/Descarga_tramo_2
 - esclavo3 <- /docs_indice_invertido/Maquina_Virtual_5.2/Descarga_tramo_3



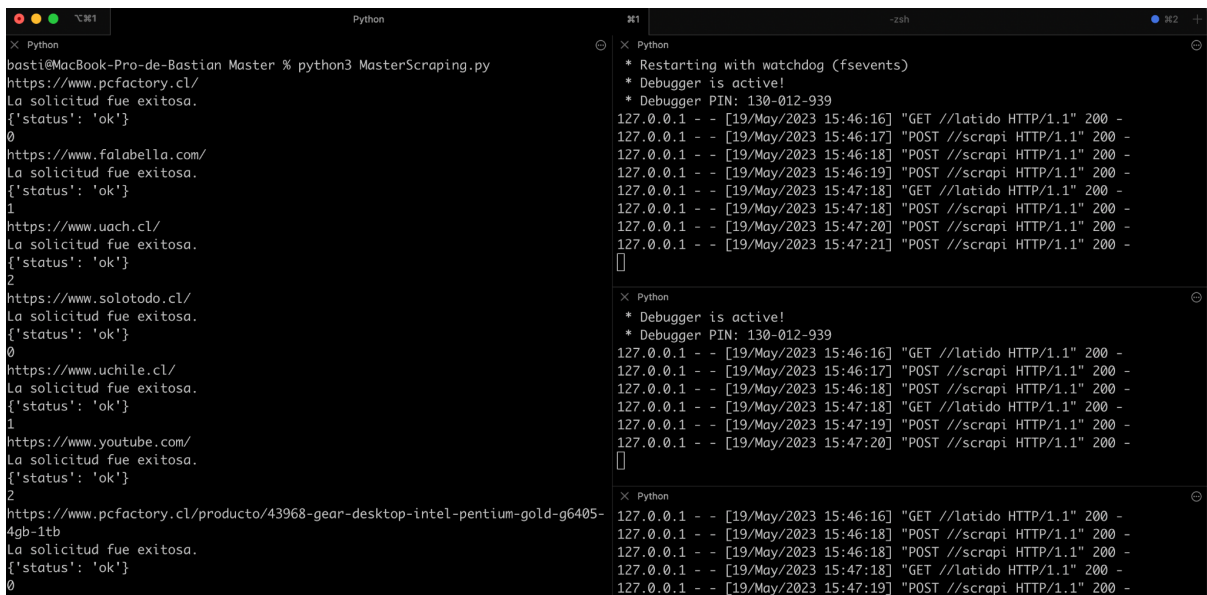
- 2) levantar a los esclavos con el siguiente comando `python3 esclavo.py`. Esto se tiene que repetir en las 3 terminales



```
basti@MacBook-Pro-de-Bastian Master %  
Python  
* Serving Flask app 'esclavo'  
* Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:4001  
* Running on http://192.168.1.15:4001  
Press CTRL+C to quit  
* Restarting with watchdog (fsevents)  
* Debugger is active!  
* Debugger PIN: 130-012-939  
Python  
ent. Use a production WSGI server instead.  
* Running on all addresses (0.0.0.0)  
* Running on http://127.0.0.1:4002  
* Running on http://192.168.1.15:4002  
Press CTRL+C to quit  
* Restarting with watchdog (fsevents)  
* Debugger is active!  
* Debugger PIN: 130-012-939  
Python  
* Running on http://127.0.0.1:4003  
* Running on http://192.168.1.15:4003  
Press CTRL+C to quit  
* Restarting with watchdog (fsevents)  
* Debugger is active!  
* Debugger PIN: 130-012-939
```

Nota: los esclavos están levantados en los puertos definidos por la arquitectura.

3) levantar el descarga master con el siguiente comando *python3 MasterScraping.py*

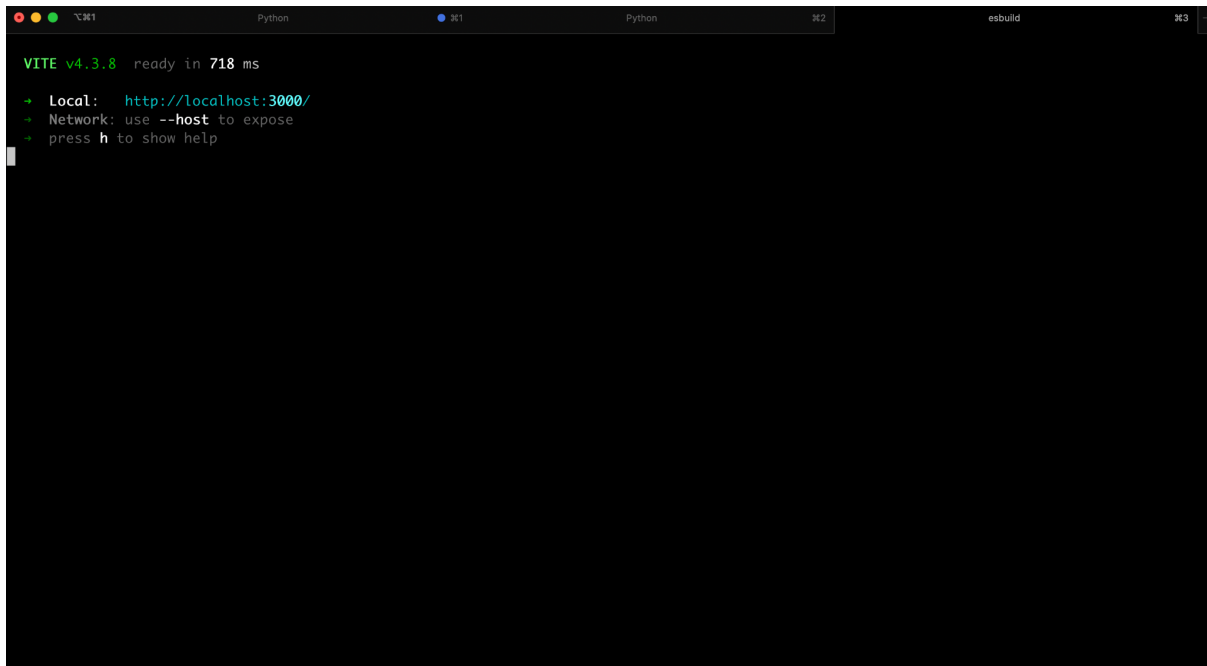


```
basti@MacBook-Pro-de-Bastian Master % python3 MasterScraping.py  
https://www.pcfactory.cl/  
La solicitud fue exitosa.  
{'status': 'ok'}  
0  
https://www.falabella.com/  
La solicitud fue exitosa.  
{'status': 'ok'}  
1  
https://www.uach.cl/  
La solicitud fue exitosa.  
{'status': 'ok'}  
2  
https://www.solatodo.cl/  
La solicitud fue exitosa.  
{'status': 'ok'}  
0  
https://www.uchile.cl/  
La solicitud fue exitosa.  
{'status': 'ok'}  
1  
https://www.youtube.com/  
La solicitud fue exitosa.  
{'status': 'ok'}  
2  
https://www.pcfactory.cl/producto/43968-gear-desktop-intel-pentium-gold-g6405-4gb-1tb  
La solicitud fue exitosa.  
{'status': 'ok'}  
0  
Python  
* Restarting with watchdog (fsevents)  
* Debugger is active!  
* Debugger PIN: 130-012-939  
127.0.0.1 - - [19/May/2023 15:46:16] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:17] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:18] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:19] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:18] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:18] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:20] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:21] "POST //scrapi HTTP/1.1" 200 -  
Python  
* Debugger is active!  
* Debugger PIN: 130-012-939  
127.0.0.1 - - [19/May/2023 15:46:16] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:17] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:18] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:18] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:19] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:20] "POST //scrapi HTTP/1.1" 200 -  
Python  
127.0.0.1 - - [19/May/2023 15:46:16] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:18] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:46:18] "POST //scrapi HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:18] "GET //latido HTTP/1.1" 200 -  
127.0.0.1 - - [19/May/2023 15:47:19] "POST //scrapi HTTP/1.1" 200 -
```

Nota: podemos notar cómo se realiza es scraping con éxito al momento de levantar el master

FrontEnd

- 1) abrir una terminal
- 2) dirigimos a la carpeta Maquina_virtual_1 dentro de esta carpeta ir a frontend
- 3) ejecutar el siguiente comando `npm run dev`



```
VITE v4.3.8 ready in 718 ms
➔ Local:   http://localhost:3000/
➔ Network: use --host to expose
➔ press h to show help
```

Nota: podemos ver que nuestro frontEnd se levantó correctamente

Si todavía tienes alguna duda sobre cómo levantar los distintos módulos, puedes consultar los archivos README ubicados en cada carpeta. En ellos se especifica detalladamente cómo poner en marcha cada componente.