

Reportes Cut

Propósito

Aplicación Web que busca facilitar la administración y organización de las evidencias que se les socilitan a los maestros a lo largo del semestre.



índice:

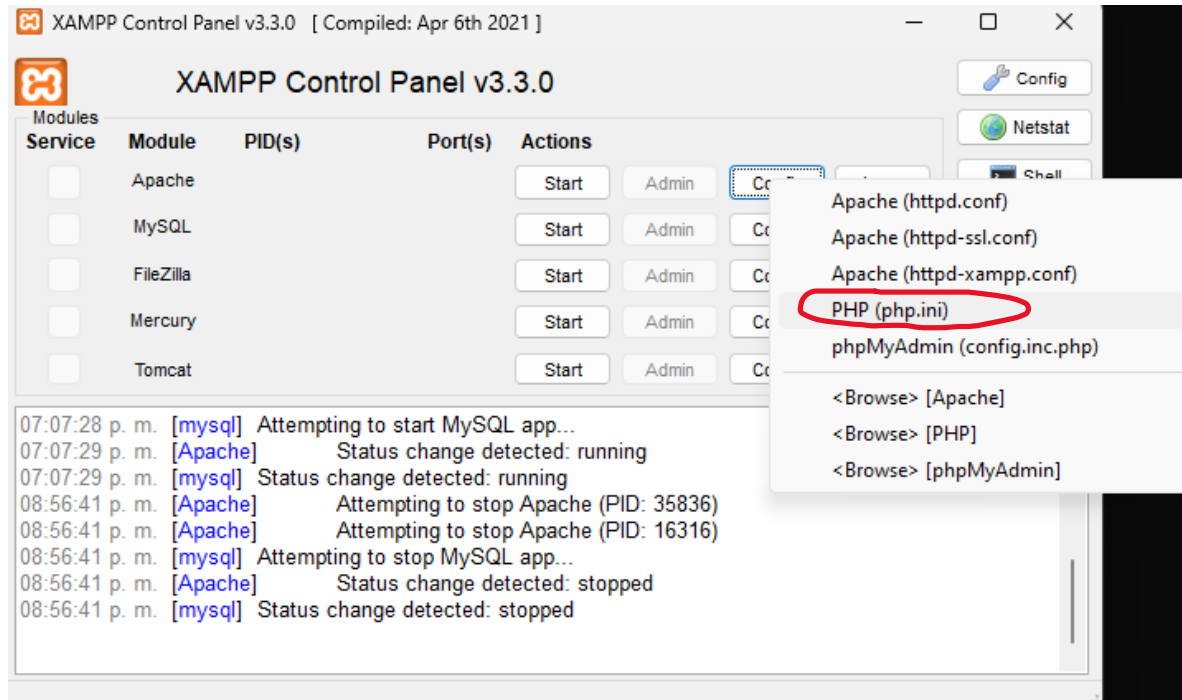
Contenido

| | |
|--------------------------------------|-----------|
| Importar base de datos: | 3 |
| Backend: | 7 |
| Frontend: | 10 |

Importar base de datos:

Necesitamos tener instalado "XAMPP", con eso nos instala lo necesario para correr la base de datos (MySQL, apache, etc), nomás tenemos que especificarlo en la instalación (no desmarcar ninguna castilla durante la instalación).

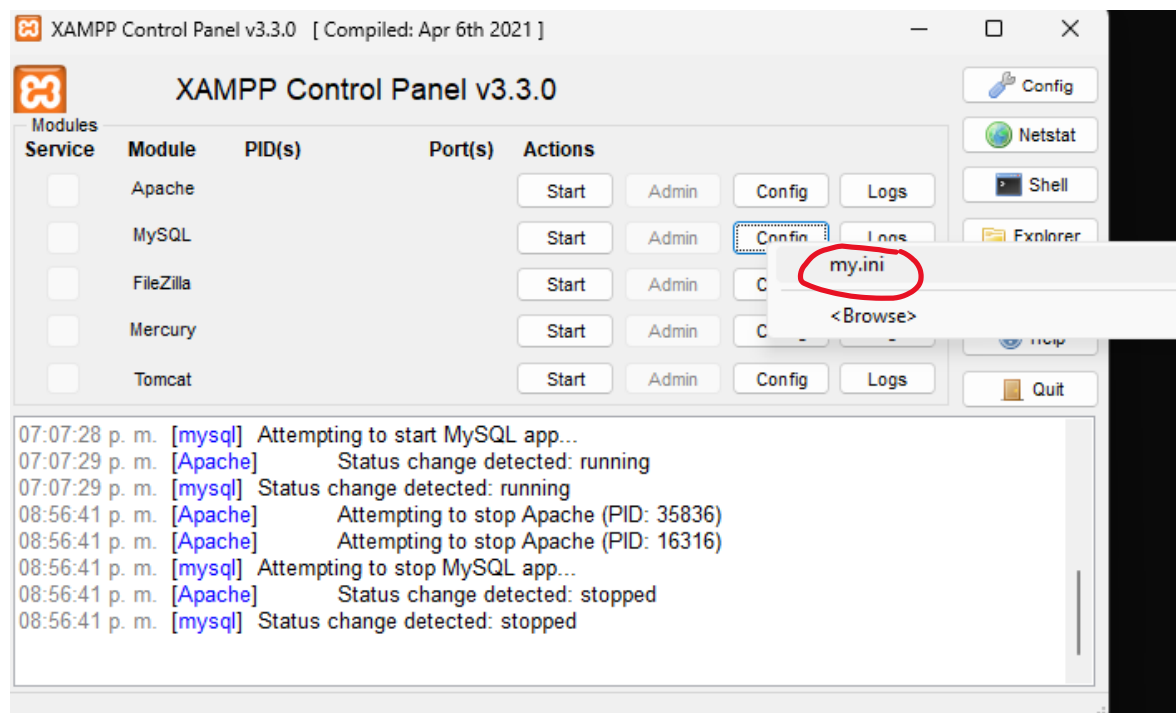
Nos movemos a "XAMPP" y damos click en la configuración de "apache" y abrimos el "PHP (php.ini)":



En el archivo presionamos "Ctrl + F" (en caso de tener la PC en ingles es con "Ctrl + B") y cambiamos los siguientes parámetros por estos:

```
max_execution_time = 600  
max_input_time = 600  
memory_limit = 1024M  
post_max_size = 1024M  
upload_max_filesize = 1024M
```

Guardamos los cambios, salimos y cambiamos a “MySQL” y entramos al archivo “my.ini”:

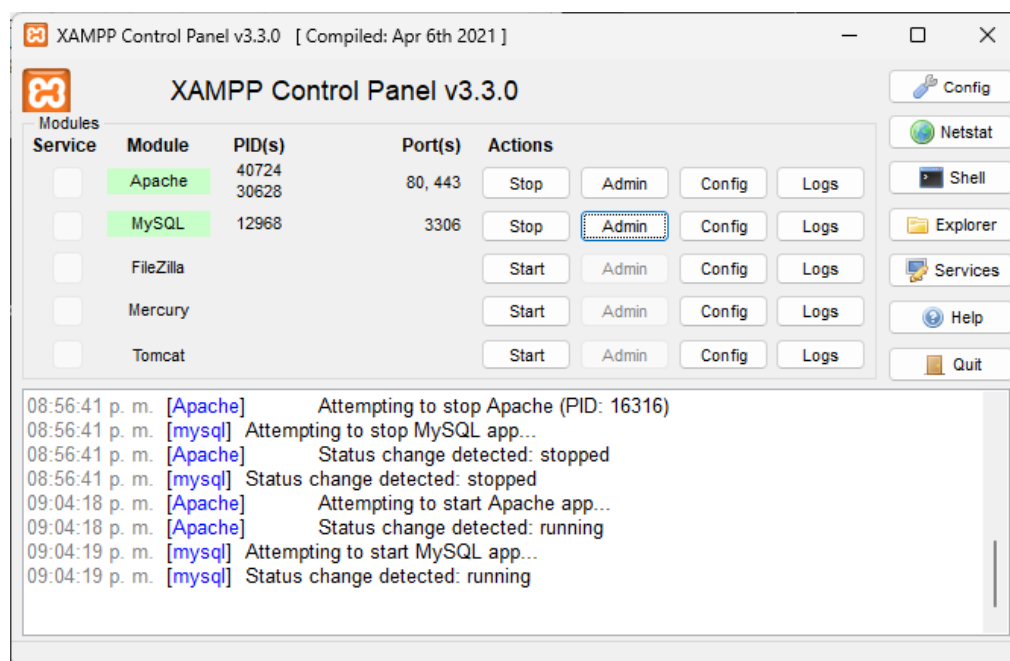


Cambiamos el siguiente valor:

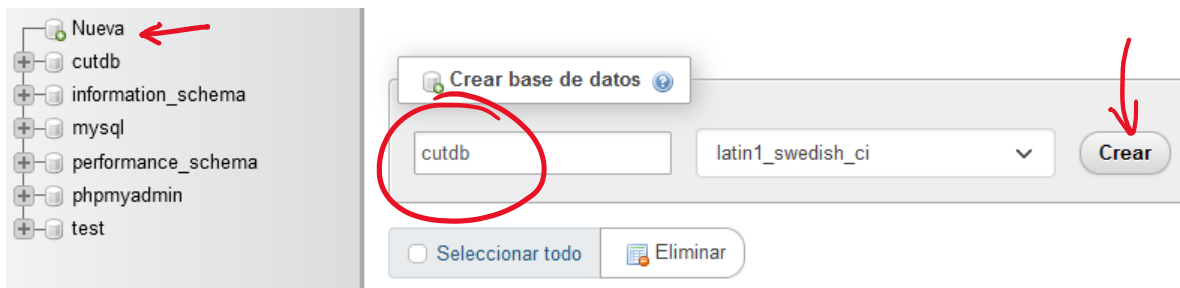
`max_allowed_packet = 1024M`

Igual guardamos y salimos.

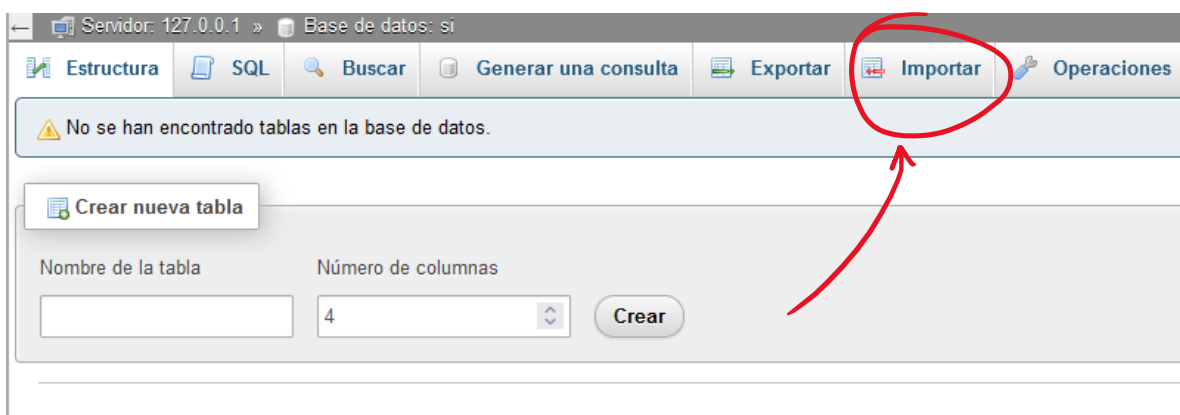
Ahora iniciamos ambos servicios y entramos al “administrador de “MySQL” para movernos a phpMyAdmin:



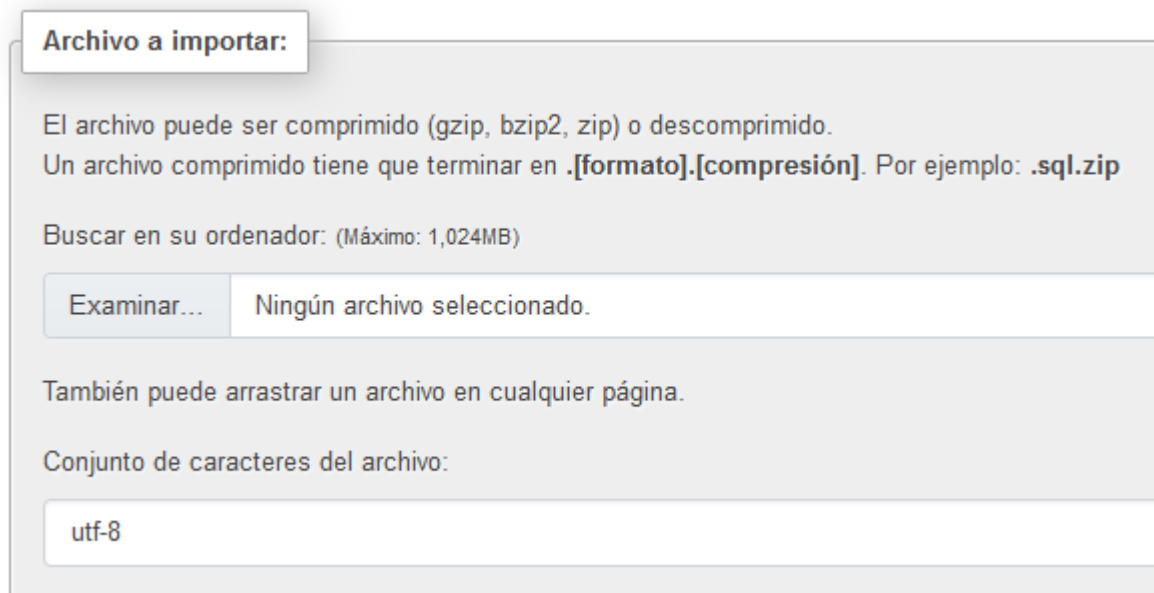
Para tener la base de datos creamos una nueva con el nombre de “cutdb” y damos en crear:



Nos posicionamos dentro de la base recién creada y damos a “importar” en la parte superior:

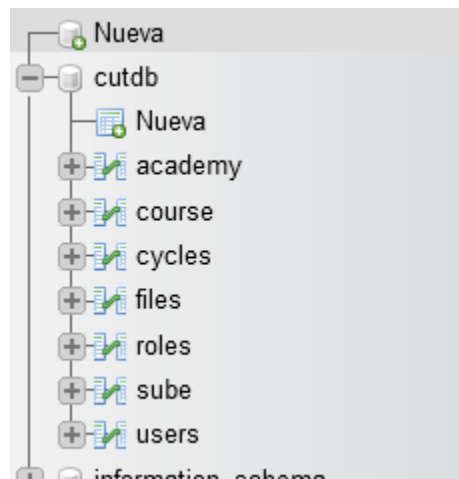


Elegimos el archivo “cutdb.sql” que tenemos dentro de la carpeta “backend”:



Una vez seleccionada, nos vamos hasta abajo y damos a “importar”

Nos deberían salir las siguientes tablas dentro de nuestra database:



Y con eso ya tenemos importada nuestra database.

Backend:

Necesitamos tener instalado Python 3.9.5 o alguna versión 3.9.x

Vemos si tenemos la carpeta “venv”:

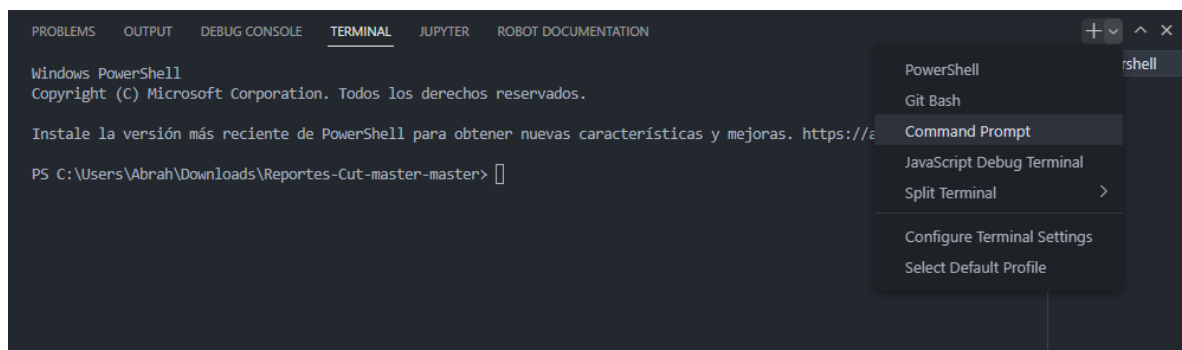
| Nombre | Fecha de modificación | Tipo | Tamaño |
|--------------------|------------------------|-----------------------|-----------|
| ▼ La semana pasada | | | |
| cutdb | 12/10/2022 05:02 p. m. | Archivo de origen ... | 42,025 KB |
| requirements | 12/10/2022 05:02 p. m. | Documento de te... | 1 KB |
| .gitignore | 12/10/2022 05:02 p. m. | Archivo de origen ... | 1 KB |
| venv | 12/10/2022 05:09 p. m. | Carpeta de archivos | |
| src | 12/10/2022 05:02 p. m. | Carpeta de archivos | |

Si es así, la borramos

Abrimos el proyecto en Visual Studio o nos posicionamos en la carpeta desde CMD

```
C:\Users\Abrah>cd Downloads
C:\Users\Abrah\Downloads>cd Reportes-Cut-master-master
C:\Users\Abrah\Downloads\Reportes-Cut-master-master>
```

Es necesario (en caso de estar en Visual Studio) estar en modo “command prompt”, ya que si estamos en la terminal de Visual Studio en modo “powershell” no podremos ejecutar bien el proyecto.



Antes de crear la ventana virtual, es necesario tener instalado flask, entonces ejecutamos en “cmd” el siguiente comando:

```
pip install flask
```

Una vez instalado “flask” nos movemos a la carpeta de “backend” con:

```
cd backend
```

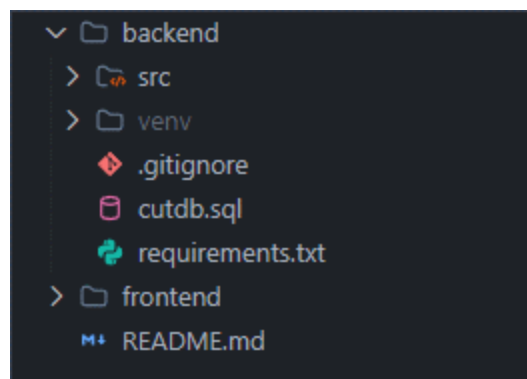
Ejecutamos:

```
pip install virtualenv
```

Después tenemos 2 opciones, si uno no funciona, usar el otro:

1. `virtualenv venv`
2. `python3 -m venv venv`

Con cualquiera de esos 2 se nos tiene que crear la carpeta “venv”:



Nos movemos a la carpeta venv, y después Scripts, con:

```
cd venv\Scripts\
```

```
C:\Users\Abrah>cd Downloads
C:\Users\Abrah\Downloads>cd Reportes-Cut-master-master
C:\Users\Abrah\Downloads\Reportes-Cut-master-master>cd backend
C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend>cd venv\Scripts\
C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv\Scripts>
```


Dentro de Scripts ejecutamos:

`activate`

```
C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv\Scripts>activate
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv\Scripts>
```

Regresamos a la carpeta “backend” y ejecutamos:

`pip install -r requirements.txt`

```
C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv\Scripts>activate
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv\Scripts>cd..
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\venv>cd..
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend>pip install -r requirements.txt_
```

Con eso se nos instalan las librerías necesarias.

Para correr el “backend”, estando en “backend” nos movemos a la carpeta “src” y ejecutamos:

`python app.py`

```
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend>cd src
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\src>python app.py
```

Sería todo, con eso ya estamos corriendo el “backend”

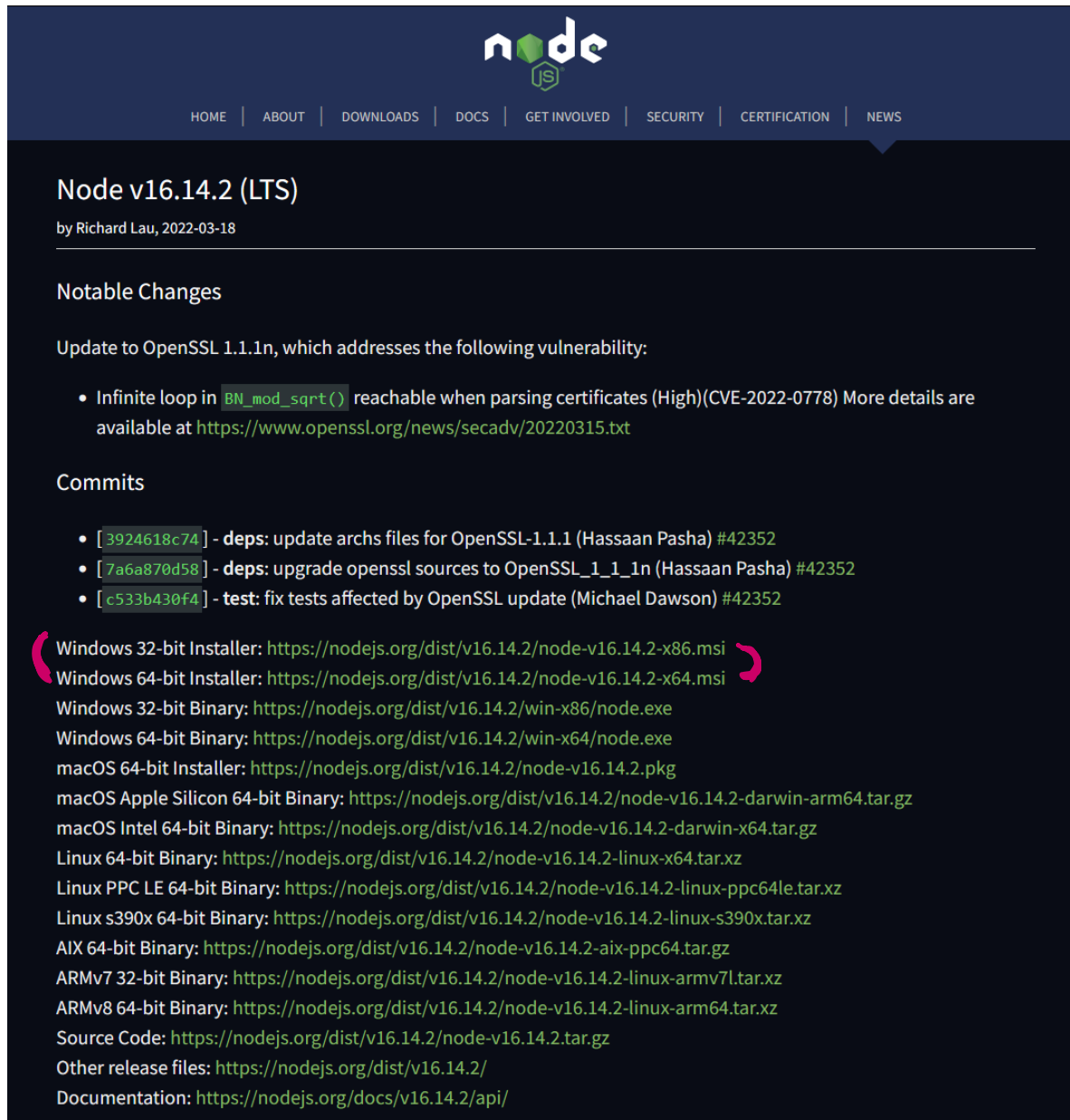
```
(venv) C:\Users\Abrah\Downloads\Reportes-Cut-master-master\backend\src>python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 102-885-295
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

La siguiente vez que queramos iniciarlo solo tenemos que posicionarnos en nuestra carpeta con el proyecto “backend\venv\Scripts”, poner “activate”, regresar a “backend”, movernos a src “backend\src” y escribir “python app.py”.

Frontend:

Para esto necesitamos tener instalado el “[node.js](#)” en su versión “[16.14.2](#)” y el “[npm](#)” en su versión “[8.5.0](#) u [8.5.4](#)”, para ello vamos al siguiente link, pulsamos sobre la versión de Windows que tengamos ([64-bit](#) o [32-bit](#)) y lo instalamos normalmente:

<https://nodejs.org/en/blog/release/v16.14.2/>



The screenshot shows the Node.js v16.14.2 LTS release page. At the top is the Node.js logo and a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. The main heading is "Node v16.14.2 (LTS)" by Richard Lau, dated 2022-03-18. Below this is a section titled "Notable Changes" which mentions an update to OpenSSL 1.1.1n to address a vulnerability (CVE-2022-0778). A list of commits follows, including updates to archs files, openssl sources, and test fixes. The bottom section lists download links for various operating systems and architectures, including Windows 32-bit and 64-bit installers and binaries, macOS 64-bit installer and binaries, Linux 64-bit and s390x binaries, AIX 64-bit binary, and ARMv7 and ARMv8 binaries. It also provides links for source code, other release files, and documentation.

Node v16.14.2 (LTS)
by Richard Lau, 2022-03-18

Notable Changes

Update to OpenSSL 1.1.1n, which addresses the following vulnerability:

- Infinite loop in `BN_mod_sqrt()` reachable when parsing certificates (High)(CVE-2022-0778) More details are available at <https://www.openssl.org/news/secadv/20220315.txt>

Commits

- [[3924618c74](#)] - deps: update archs files for OpenSSL-1.1.1 (Hassaan Pasha) #42352
- [[7a6a870d58](#)] - deps: upgrade openssl sources to OpenSSL_1_1_1n (Hassaan Pasha) #42352
- [[c533b430f4](#)] - test: fix tests affected by OpenSSL update (Michael Dawson) #42352

Windows 32-bit Installer: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-x86.msi>

Windows 64-bit Installer: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-x64.msi>

Windows 32-bit Binary: <https://nodejs.org/dist/v16.14.2/win-x86/node.exe>

Windows 64-bit Binary: <https://nodejs.org/dist/v16.14.2/win-x64/node.exe>

macOS 64-bit Installer: <https://nodejs.org/dist/v16.14.2/node-v16.14.2.pkg>

macOS Apple Silicon 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-darwin-arm64.tar.gz>

macOS Intel 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-darwin-x64.tar.gz>

Linux 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-x64.tar.xz>

Linux PPC LE 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-ppc64le.tar.xz>

Linux s390x 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-s390x.tar.xz>

AIX 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-aix-ppc64.tar.gz>

ARMv7 32-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-armv7l.tar.xz>

ARMv8 64-bit Binary: <https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-arm64.tar.xz>

Source Code: <https://nodejs.org/dist/v16.14.2/node-v16.14.2.tar.gz>

Other release files: <https://nodejs.org/dist/v16.14.2/>

Documentation: <https://nodejs.org/docs/v16.14.2/api/>

Una vez instalado ingresamos el siguiente comando:

```
npm cache clean --force
```

```
C:\Users\Abrah\Downloads>cd Reportes-Cut-master-master  
C:\Users\Abrah\Downloads\Reportes-Cut-master-master>npm cache clean --force  
npm WARN using --force Recommended protections disabled.  
C:\Users\Abrah\Downloads\Reportes-Cut-master-master>_
```

Para correr el “frontend” nos posicionamos sobre la carpeta “frontend” y escribimos “npm start”:

```
C:\Users\Abrah\Downloads\Reportes-Cut-master-master\frontend>npm start  
  
> frontend@0.1.0 start  
> react-scripts start
```

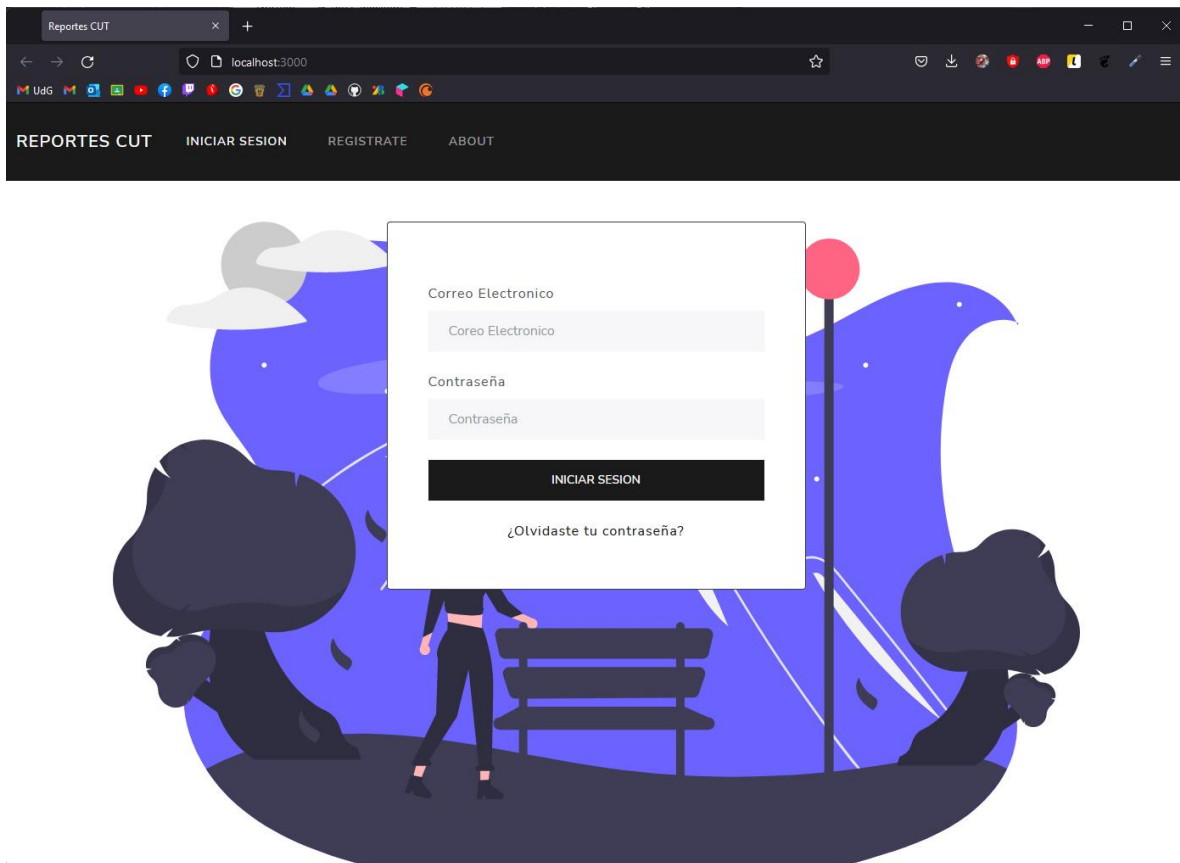
En caso de que debajo de eso nos de un error llamado “react-scripts no se reconoce como un comando interno o externo”

```
"react-scripts" no se reconoce como un comando interno o externo,  
programa o archivo por lotes ejecutable.  
npm ERR! code ELIFECYCLE  
npm ERR! errno 1  
npm ERR! todo-app-react@0.1.0 start: `react-scripts start`  
npm ERR! Exit status 1  
npm ERR!  
npm ERR! Failed at the todo-app-react@0.1.0 start script.  
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.  
  
npm ERR! A complete log of this run can be found in:  
npm ERR! C:\Users\HP-PC\AppData\Roaming\npm-cache\_logs\2021-11-11T22_01_06_861Z-debug.log
```

Ingresamos el siguiente comando para arreglar eso:

```
npm install react-scripts
```

Y volvemos a correr el proyecto con “npm start” y debería correr normalmente:



Ya cuando hayamos terminado todo eso, para abrir el frontend nos posicionamos sobre la carpeta, nos movemos a “frontend” y escribimos “npm start”.

Para iniciar todo el proyecto como tal:

1. Activar apache y MySQL en XAMPP
2. Correr el backend
3. Correr el frontend
4. Y ya, fin :D