

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE BUSCADOR WEB BASADO EN LINUX PARA GARANTIZAR LA PRIVACIDAD DE LOS DATOS DEL USUARIO

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

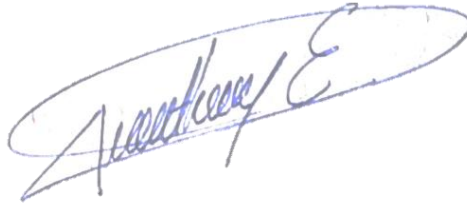
ANTHONY BRYAN ENCALADA DÁVILA

DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ

DMQ, marzo 2023

CERTIFICACIONES

Yo, Anthony Bryan Encalada Dávila declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Anthony Bryan Encalada Dávila

anthony.encalada01@epn.edu.ec

anthonyenca.2002@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Anthony Bryan Encalada Dávila, bajo mi supervisión.



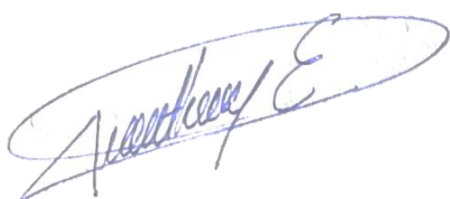
Leandro Antonio Pazmiño Ortiz

DIRECTOR

leandro.pazmino@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



Anthony Bryan Encalada Dávila

CI: 1724755689

DEDICATORIA

Le dedico el resultado de este trabajo de titulación a mi hermano, quien supo darme consejos, ayudarme a ser una mejor persona cada día y darme su apoyo incondicional, también a mi abuela quién hizo el papel de madre y me ayudó con todo lo que estuvo a su alcance para darme una profesión, finalmente a mi familia quienes me han apoyado desde el inicio hasta la finalización de mi carrera.

AGRADECIMIENTO

Agradezco a Dios por darme salud y vida para culminar mi carrera, a los maestros quienes supieron guiarme a través de esta parte tan importante de mi vida, a mi tutor de tesis quien me ayudó a culminar este proyecto de titulación con éxito, finalmente a mi abuela, hermano y a mi familia por el apoyo y cariño que me brindaron en esta etapa de mi vida.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN.....	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	1
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	2
2 METODOLOGÍA	4
3 RESULTADOS	5
3.1 Identificación de los requerimientos para el diseño del prototipo	5
Uso de <i>software</i> libre	5
<i>Software</i> del servidor	5
<i>Hardware</i> del servidor.....	5
Conectividad con Internet y la red.....	5
3.2 Selección de software y hardware acorde a los requerimientos establecidos .	6
Selección de software.....	6
Selección del <i>hardware</i>	7
3.3 Diseño del prototipo.....	8
Esquema general.....	8
Consulta del direccionamiento en la red	9
Configuración de la dirección <i>IP</i> del servidor	9
Reservación de la dirección <i>IP</i>	10

3.4	Implementación del prototipo	11
	Configuración de la máquina virtual	11
	Instalación de paquetes	13
	Creación del usuario y directorio.....	13
	Instalación de <i>SearXNG</i>	14
	Configuración de <i>SearXNG</i>	15
	Comprobación	15
3.5	Pruebas de funcionamiento del prototipo	16
	Acceso desde cualquier dispositivo	16
	Búsquedas realizadas con más de un motor de búsqueda	17
	Datos de navegación almacenados de manera local	18
	Vídeo demostrativo del funcionamiento	18
4	CONCLUSIONES	18
5	RECOMENDACIONES.....	20
6	REFERENCIAS BIBLIOGRÁFICAS.....	20
7	ANEXOS.....	23
	ANEXO I: Certificado de Originalidad	i
	ANEXO II: Archivo de configuración	iii

RESUMEN

En este documento se detalla el procedimiento y los requerimientos tanto de software como de hardware los cuales son necesarios para la implementación de un prototipo de buscador web dicho prototipo se lo realizó mediante el metabuscador SearXNG en Linux que garantice la privacidad.

En la sección inicial se encuentra la introducción del proyecto esta a su vez contiene la descripción del componente, adicionalmente se detallan el alcance y objetivos propuestos para la implementación de este prototipo.

En la segunda sección se encuentra desarrollada la metodología usada con la finalidad de cumplir con cada uno de los objetivos que se encuentran planteados para el proyecto, además, contiene los procesos realizados para cada punto solicitado con el fin de cumplir dichos objetivos.

En la tercera sección se encuentran los resultados que se obtuvieron en función del resultado final de la implementación, además, en dicha sección se muestran cuáles fueron los parámetros que se tomaron en cuenta para la elección tanto de software como de hardware, para la elección se toma en cuenta los requerimientos planteados para este prototipo, de igual manera se encuentran las configuraciones de instalación y en el metabuscador las mismas que son necesarias para el correcto funcionamiento del mismo y las pruebas del prototipo mediante la realización de búsquedas de manera local, desde otro dispositivo dentro de la red y consulta del lugar de almacenamiento de los datos generados por el usuario.

Finalmente, en la cuarta sección se encuentran las recomendaciones y conclusiones obtenidas al realizar la implementación del prototipo con sus respectivas pruebas, además, en la última sección se muestran las referencias usadas para el desarrollo este proyecto, así mismo se muestran los anexos.

PALABRAS CLAVE: buscador web, Linux, privacidad, prototipo, SearXNG.

ABSTRACT

This document details the procedure and requirements of both software and hardware which are necessary for the implementation of a prototype of a web search engine, this prototype was made using the meta search engine SearXNG in Linux to ensure privacy.

In the initial section is the introduction of the project, which in turn contains the description of the component, additionally the scope and objectives proposed for the implementation of this prototype are detailed.

In the second section is developed the methodology used in order to meet each of the objectives that are set for the project, also contains the processes carried out for each point requested in order to meet these objectives.

In the third section are the results that were obtained based on the final result of the implementation, in addition, this section shows what were the parameters that were taken into account for the choice of both software and hardware, for the choice is taken into account the requirements set out for this prototype, in the same way are the installation configurations and in the meta-search the same that are necessary for the proper functioning of the same and testing of the prototype by performing searches locally, from another device within the network and consultation of the storage location of the data generated by the user.

Finally, in the fourth section are the recommendations and conclusions obtained from the implementation of the prototype with its respective tests, in addition, the last section shows the references used for the development of this project, as well as the annexes.

KEYWORDS: web search engine, Linux, privacy, prototype, SearXNG.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Una de las principales desventajas al momento de realizar búsquedas en los navegadores web convencionales es que las instancias en las que estos se encuentran instalados son públicas, es decir que, están a disposición de cualquier usuario que posea la URL, al usar dichas instancias no se tiene la certeza de que es lo que sucede con la información que se encuentra en las solicitudes al realizar búsquedas, ya que, éstas pueden ser registradas o a su vez vendidas a terceros.

Con este proyecto se implementará un prototipo de navegador web privado, este tendrá características similares a los buscadores convencionales, así como un mayor nivel de seguridad, este prototipo será capaz de presentar resultados tomados de varios motores de búsqueda como son: *Google*, *Bing*, *duckduckgo*, entre otros. Los datos generados por el usuario al momento de realizar consultas no serán almacenados por el navegador, ya que se guardarán de forma local brindando un nivel mayor de seguridad al usuario, finalmente, los dispositivos conectados de forma cableada e inalámbrica que se encuentren en la misma red en la que se implementó este buscador podrán tener acceso al mismo. El objetivo principal del prototipo es reemplazar a los buscadores convencionales teniendo seguridad adicional.

La implementación se realizó en base al metabuscador *SearXNG*, la razón por la cual se ha elegido este es debido al nivel de protección que brinda a los usuarios ya que usa instancias privadas y posee una variedad de configuraciones a nivel de seguridad siendo estas: niveles de restricción al momento de realizar búsquedas, configuración de los motores de búsqueda con los que se desea trabajar, almacenamiento de los datos, no muestra anuncios por lo que la información del usuario no será enviada a terceros, entre otras.

Para comprobar el funcionamiento se realizaron búsquedas dentro de *SearXNG* y se verificaron que los resultados mostrados pertenezcan a más de un motor de búsqueda, adicionalmente se configuró una restricción al momento de realizar las consultas para brindar mayor seguridad.

1.1 Objetivo general

Implementar un prototipo de buscador web basado en Linux para garantizar la privacidad de los datos del usuario.

1.2 Objetivos específicos

- Identificar los requerimientos para el diseño del prototipo.
- Seleccionar el software y hardware acorde a los requerimientos identificados.
- Diseñar el prototipo del buscador web.
- Implementar el prototipo.
- Realizar pruebas del funcionamiento del prototipo.

1.3 Alcance

Por medio del presente proyecto se busca implementar un prototipo de buscador web que permita realizar cualquier consulta en Internet con la finalidad de reemplazar a los buscadores tradicionales como: Google, Bing, Yahoo, etc. La diferenciación principal es que la privacidad de los datos del usuario estará protegida. Por tal motivo, tendrán la capacidad de realizar las siguientes acciones:

- Desplegar un sistema que permita contar con un buscador web que pueda ser accedido desde cualquier dispositivo.
- Las búsquedas web se podrán realizar empleando más de un motor de búsqueda a la vez como por ejemplo Google y Bing permitiendo diversificar las respuestas a las consultas realizadas.
- Los datos de navegación de los usuarios serán mantenidos de manera local lo que permitirá proteger la privacidad de estos.

1.4 Marco Teórico

Metabuscador

Uno de los mayores inconvenientes al momento de realizar búsquedas en motores de búsqueda único es que no muestra la información necesaria y tampoco toma toda las posibles respuestas para ser mostrado al usuario, por el contrario, un metabuscador ofrece mayor diversidad al realizar consultas en comparación a usar un solo motor de búsqueda, el motor de búsqueda único muestra solamente el 16% de todas las páginas que se muestran en Internet, mientras que un metabuscador puede mostrar aproximadamente el 42%, esto se debe a que combina varios motores únicos para realizar las consultas, para que un metabuscador sea eficiente usa una interfaz consistente, es importante que exista esta interfaz ya que la misma garantiza la unión de varios motores de búsqueda [1].

Metabuscador SearXNG

Es un metabuscador gratuito que brinda privacidad a los datos del usuario negando el rastreo y creación de perfiles de usuario, este tiene la capacidad usar Tor para tener aún más privacidad y es capaz de proteger la información de los usuarios sin importar si la instancia que se usa es privada o pública. Además, elimina los datos privados que son solicitados por las páginas por lo que estos no son enviados a terceros, la eliminación de la información privada tiene relación con las cookies ya que, al eliminar dicha información, las cookies no se envían a los motores de búsqueda, esta información es usada por dichos motores para crear perfiles de los usuarios acerca de los tipos de búsquedas más frecuentes que realizan los usuarios [2].

Entre las características más relevantes de SearXNG se encuentran: auto hospedado, no realiza un seguimiento a los usuarios que lo usan, no crea perfiles de usuario, es compatible con 70 motores de búsqueda y posee fácil integración con los mismos, las cookies generadas se almacenan de forma local y las conexiones al momento de realizar las búsquedas son seguras usando los protocolos HTTPS y SSL [2].

Instancia privada

En una instancia privada se tiene control total de lo que se implemente sobre ella, dentro de la instancia se puede modificar el código fuente, configuraciones de los registros y control de los datos, de esta forma no es necesario confiar la información a administradores de otras instancias que no se conocen, por lo que cualquier configuración se puede editar. Otro aspecto es que en caso de restablecer las cookies las configuraciones dentro del buscador no volverán a los datos de fábrica ya que todo se almacena de forma local [3].

Indexación de archivos

La indexación se produce al momento de recopilar documentos, los cuales son clasificados y posteriormente pueden ser accedidos con la ayuda de palabras clave, se trata de un factor importante al momento de trabajar con motores de búsqueda, ya que con usar una palabra aparecerá el contenido relacionado e importante de la palabra en cuestión [4].

Entorno virtual

Se trata de un sistema aislado, es decir que se encuentra separado del sistema operativo o de cualquier otro entorno que se cree, dentro de este se encuentra el intérprete de Python, adicionalmente se pueden agregar librerías o descargar paquetes,

las herramientas comunes para la instalación de estos paquetes son: *setuptools* y *pip* [5].

2 METODOLOGÍA

Para el primer paso se establecieron los requerimientos necesarios de *hardware* dentro de los cuales se encuentran los requisitos físicos y *software* en este caso el sistema operativo basados en el alcance, ya que de este se obtienen las características con las que cumple el prototipo, en base estos requerimientos se investigó el equipo necesario.

En la selección de hardware y software se eligieron las características necesarias para el funcionamiento correcto del prototipo que van desde la elección del sistema operativo libre el cual presenta una versión compatible de *Python* importante para el uso de librerías, hasta las características de almacenamiento y adaptador de red que se necesita.

Dentro del diseño del prototipo se desarrollaron los pasos necesarios para su funcionamiento, en este se encuentran: el esquema en el que se muestran las conexiones de los equipos, la máquina que contendrá el metabuscador y será capaz de brindar el servicio, así como las configuraciones necesarias dentro del *router* para el direccionamiento de la red.

Para la implementación del prototipo se creó una máquina virtual, está tendrá instalado el sistema operativo y el metabuscador, se verificó que el funcionamiento esté correcto y que se exista conexión a la máquina virtual desde otro equipo de la red mediante un ping.

Las pruebas de funcionamiento se basaron en los objetivos del prototipo para comprobar que su desempeño es el correcto, los resultados que se obtuvieron al momento de realizar las pruebas son esenciales para realizar correcciones de posibles errores que pudieran aparecer durante la ejecución y de esta forma se garantizó el cumplimiento de todos los requisitos para el prototipo.

3 RESULTADOS

3.1 Identificación de los requerimientos para el diseño del prototipo

Los requerimientos para la implementación se analizaron en función de la documentación de *SearXNG*, además se tuvieron en cuenta los objetivos planteados a continuación, almacenamiento local de los datos de navegación, procesamiento de las peticiones, y el acceso al prototipo desde cualquier dispositivo por lo cual se identificaron los siguientes requerimientos:

Uso de *software* libre

La característica principal que tiene el sistema operativo para el servidor es que sea de *software* libre puesto que se realizaran configuraciones al momento de implementar este prototipo, por lo que es necesario disponer de una distribución de Linux que cuente con facilidad de manejo y pueda brindar entornos visuales y gráficos de forma gratuita, finalmente el uso de las aplicaciones debe ser sencillo.

***Software* del servidor**

El *software* sobre el cual se desplegará el prototipo debe contar con la versión de *Python* 3.8, debido a que, para el funcionamiento del prototipo son necesarias librerías de *Python* esto con la finalidad de no tener problemas de compatibilidad entre estas y el metabuscador.

***Hardware* del servidor**

Para el funcionamiento del prototipo es necesario que este cuente con librerías pertenecientes a *Python*, así como sus dependencias por lo que es importante que se cuente con el espacio suficiente para almacenar las mismas, así como la cantidad de memoria *RAM* que este disponga, la cual se usará al momento de ejecutar el buscador, además, en base a que el lugar donde se implementará el prototipo posee un número reducido de usuarios el procesador que se elegirá no presentará características elevadas.

Conectividad con Internet y la red

Uno de los objetivos que tiene el proyecto es el acceso a este prototipo desde cualquier dispositivo por lo que el servidor requiere de conectividad con todos los dispositivos que se encuentren en la misma red, otro requerimiento es la conexión a Internet el cual es

indispensable para la descarga de las librerías y dependencias necesarias, así como para brindar el servicio de búsquedas en Internet, además, para la capacidad de este se toma en cuenta que el ambiente en el cual se implementará tiene un número reducido de usuarios.

3.2 Selección de software y hardware acorde a los requerimientos establecidos

Selección de software

Selección del sistema operativo libre

Ubuntu 20.04 Desktop

La distribución de Linux Ubuntu 20.04 *desktop* cuenta con interfaz gráfica, esta es necesaria para acceder a *SearXNG* y realizar consultas para comprobar el funcionamiento de este, otro aspecto por el cual se decidió usar esta distribución es que tiene instalado por defecto la versión de *Python 3.8* lo cual permitirá tener compatibilidad con paquetes y librerías, necesarios para la implementación del prototipo [6].

Python 3.8

Es imprescindible que se encuentre instalada la versión 3.8, puesto que, las dependencias y paquetes usados para la implementación del prototipo deben ser compatibles, los paquetes y versiones correspondientes requeridos por *SearXNG* se muestran a continuación [7] [8]:

- certifi==2022.12.7
- babel==2.11.0
- flask-babel==2.0.0
- flask==2.2.2
- jinja2==3.1.2
- lxml==4.9.2
- pygments==2.13.0
- python-dateutil==2.8.2
- pyyaml==6.0
- httpx[http2]==0.21.2
- Brotli==1.0.9
- uvloop==0.17.0
- httpx-socks[asyncio]==0.7.2
- setproctitle==1.3.2

- redis==4.4.0
- markdown-it-py==2.1.0
- typing_extensions==4.4.0
- fasttext-wheel==0.9.2

Selección del *hardware*

Para la selección de *hardware* se consideraron las especificaciones del sistema operativo Ubuntu sobre la cual se implementa el prototipo, así como las características que posee máquina física en la que se instalará la máquina virtual y, además, se tiene en cuenta el lugar de implementación seleccionado, ya que el *hardware* deberá cumplir con los requerimientos establecidos.

Procesador

Los motores de búsqueda realizan indexación tanto al momento de realizar consultas como con los archivos en tiempo real, es decir que realiza el procesamiento de información de la base de datos que poseen los buscadores, por ello se requiere de un procesador que permita tener un rendimiento óptimo para trabajar con dicha información, puesto que el lugar en el que se implementará el prototipo no posee un número elevado de usuarios se usó el procesador del ordenador sobre el cual se encuentra implementada la máquina virtual para este caso fue un Intel Core i3 de segunda generación con una velocidad de 2.20 (GHz) con 4 núcleos [9] [10].

Memoria RAM

La memoria RAM se encargará de almacenamiento y ejecución temporal de los procesos del sistema operativo, así como del servicio, la asignación insuficiente tiene un impacto negativo en el funcionamiento del sistema operativo, tomando en cuenta el requerimiento mínimo para la distribución que se va a usar, este prototipo cuenta con 3 (GB) de RAM [9] [10].

Capacidad de almacenamiento

Tomando en cuenta el requisito mínimo para la distribución el cual es de 25 (GB), y dependencias que son necesarias descargar y que la cantidad de almacenamiento de la máquina dependerá de los archivos y componentes que harán la indexación el espacio en disco es de 50 (GB) [11] [12].

Adaptador de red

Puesto que el metabuscador realiza las consultas a través de varios motores de búsqueda, este debe contar con conexión a Internet, además, uno de los objetivos

planteados es el acceso al prototipo desde cualquier dispositivo de la red, por ello se decidió usar el adaptador de red del ordenador físico.

La máquina virtual se encuentra configurada con adaptador tipo puente gracias a ello es reconocida como una máquina física dentro de la red, permitiendo así tener conexión bidireccional, teniendo en cuenta que el número de usuarios que usarán el servicio es reducido la capacidad del adaptador de red es el que tiene el ordenador donde se encuentra la máquina virtual, siendo este un Realtek Express Gigabit Ethernet 1 (Gbps) [13] [14].

3.3 Diseño del prototipo

Esquema general

Este prototipo se desplegó en una máquina virtual que cuenta con el sistema operativo Ubuntu 20.04, el cual se encuentra configurado como servidor brindando el servicio de búsqueda a los dispositivos que estén conectados en la misma red. El prototipo se diseñó de tal forma que cualquier dispositivo dentro de la red pueda acceder al metabuscador, en la Figura 3.1 se observa el esquema general del prototipo.

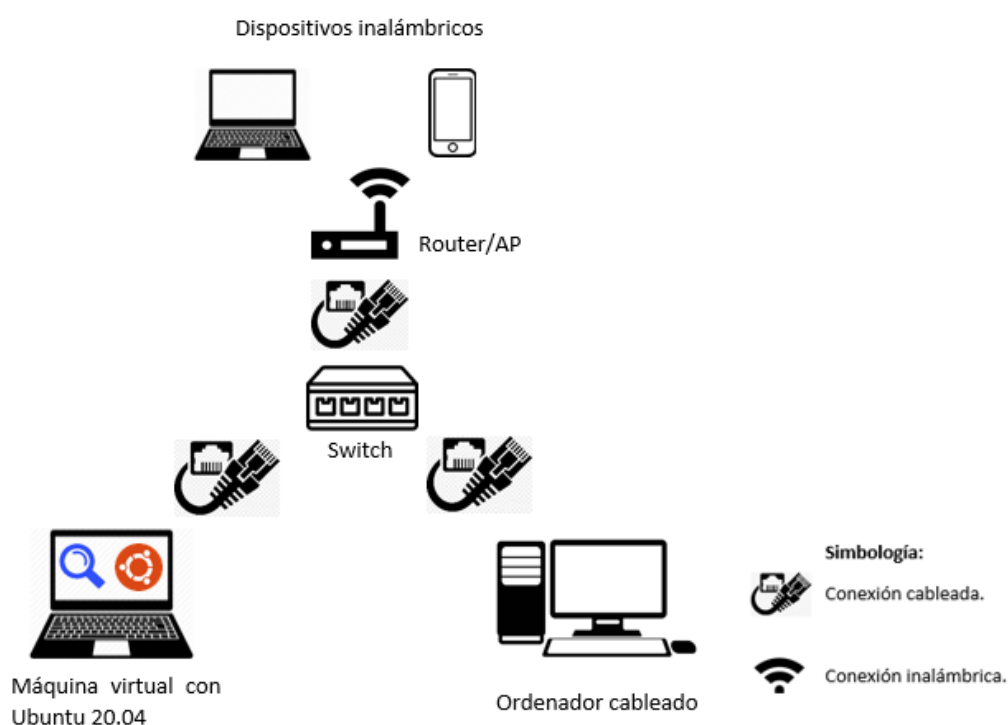


Figura 3.1 Esquema general del prototipo

Para la creación de la máquina virtual se usó *VirtualBox*, en dicha máquina se instaló el sistema operativo Ubuntu 20.04 con las características de hardware y software seleccionadas anteriormente, esta máquina virtual se encuentra en una laptop, y esta a

su vez está conectada de forma cableada. Adicionalmente se encuentra conectado un ordenador de escritorio ambos equipos están dentro de la red por medio de un *switch* y este a su vez se conecta al router.

En la máquina virtual se encuentra instalado el metabuscador *SearXNG*, el cual da servicio a todos los dispositivos que estén conectados a la misma red, este posee configuraciones adicionales para permitir dicho acceso, así como la agregación de varios motores de búsqueda.

El router se encarga de realizar el direccionamiento mediante el protocolo DHCP, esta asignación de direcciones se realiza tanto a los equipos que se encuentran conectados de forma cableada como a los equipos que están conectados de forma inalámbrica, ya que el router tiene la función de *access point*.

Consulta del direccionamiento en la red

Para conocer las direcciones que se asignan en la red fue necesario ingresar a la configuración del router en el apartado de LAN y en la configuración del servidor DHCP como se muestra en la siguiente Figura 3.2.

The image shows a configuration window for a DHCP server, specifically the 'Primary Address Pool' section. It contains several fields and checkboxes. The 'Enable Primary DHCP Server' checkbox is checked. Below it, 'Enable DHCP Relay' and 'Enable Option125' are also checked. The 'LAN Host IP Address' is set to 192.168.100.1, and the 'Subnet Mask' is 255.255.255.0. The 'Start IP Address' is 192.168.100.2, with a note indicating it must be in the same subnet as the LAN host. The 'End IP Address' is 192.168.100.254. The 'Lease Time' is set to 3 days. There are empty fields for 'Primary DNS Server' and 'Secondary DNS Server'. Below this section is the 'Secondary Address Pool' section, which has an unchecked 'Enable Secondary DHCP Server' checkbox. At the bottom are 'Apply' and 'Cancel' buttons.

Primary Address Pool	
Enable Primary DHCP Server:	<input checked="" type="checkbox"/>
Enable DHCP Relay:	<input checked="" type="checkbox"/>
Enable Option125:	<input checked="" type="checkbox"/>
LAN Host IP Address:	192.168.100.1
Subnet Mask:	255.255.255.0
Start IP Address:	192.168.100.2 <small>*(It must be in the same subnet as the IP address of the LAN host.)</small>
End IP Address:	192.168.100.254 *
Lease Time:	3 days
Primary DNS Server:	
Secondary DNS Server:	
Secondary Address Pool	
Enable Secondary DHCP Server:	<input type="checkbox"/>
<input type="button" value="Apply"/> <input type="button" value="Cancel"/>	

Figura 3.2 Consulta del direccionamiento para la red

Configuración de la dirección *IP* del servidor

Una vez consultado el rango de direcciones que se está asignando en la red, se configuró la dirección 192.168.100.38 de forma estática, para este cambio se ingresó a las configuraciones de Ubuntu en el **apartado de red**, en la opción de cableado se muestra un símbolo de configuración mostrado en la Figura 3.3.



Figura 3.3 Apartado de configuración para la dirección *IP*

En la Figura 3.4 se observa la ventana para la configuración de la dirección, en la cual se debe seleccionar la opción manual y luego ingresar la dirección que se eligió anteriormente.

Cancelar Cableada Aplicar

Detalles Identidad **IPv4** IPv6 Seguridad

Método IPv4

☐ Automático (DHCP) ☐ Sólo enlace local

☒ Manual ☐ Desactivar

☐ Compartida con otros equipos

Direcciones

Dirección	Máscara de red	Puerta de enlace
192.168.100.38	255.255.255.0	192.168.100.1

DNS Automático

8.8.8.8,4.4.4.4

Direcciones IP separadas por comas

Figura 3.4 Configuración de *IP* estática para el servidor

Reservación de la dirección *IP*

Luego de haber configurado la dirección *IP* en el servidor, se ingresó a la configuración del *router* y se reservó la misma para que el servidor no sufra cambios de direccionamiento, es decir que esta dirección no se asignará a otro equipo, la configuración se observa en la Figura 3.5.

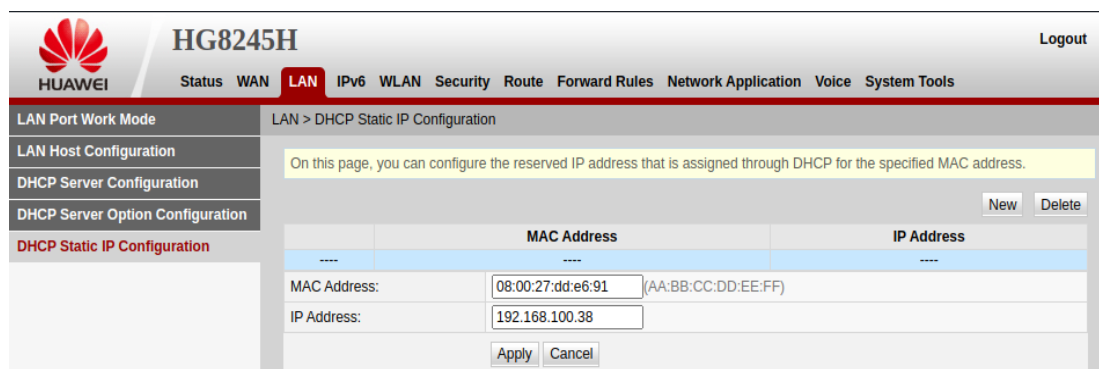


Figura 3.5 Reservación de la dirección *IP* en el router

3.4 Implementación del prototipo

El prototipo se implementó a través de una máquina virtual la misma que cuenta con la distribución Ubuntu 20.04 en la versión de escritorio, adicional a esto el ambiente en el que funciona el prototipo no cuenta con un número elevado de equipos.

Configuración de la máquina virtual

Para crear y configurar la máquina virtual se usó *VirtualBox*, adicionalmente, es necesario contar con el archivo de imagen que generalmente se encuentra en formato *ISO*, en la interfaz de *VirtualBox* se encuentra la opción “nueva” para empezar con el proceso de creación, en la Figura 3.6 se visualiza la ventana inicial de configuración en la que se ingresó el nombre, el tipo de sistema operativo y se cargó el archivo imagen de Ubuntu.

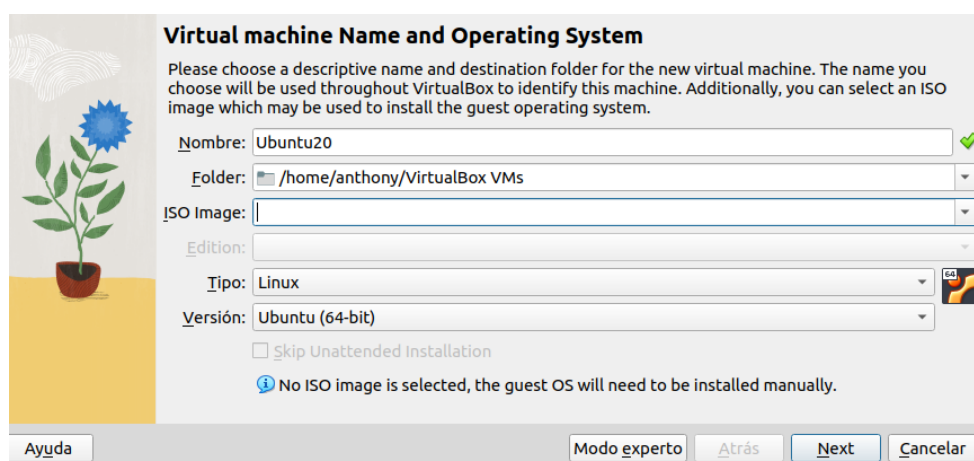


Figura 3.6 Selección de nombre, imagen y sistema operativo

La cantidad de procesamiento al realizar las consultas no es exigente puesto que en el lugar donde se implementó no existe un gran número de dispositivos que hagan uso de

este por ello en la Figura 3.7 se muestra que la memoria RAM asignada es de aproximadamente 3 (GB) y el procesador configurado es de un núcleo.

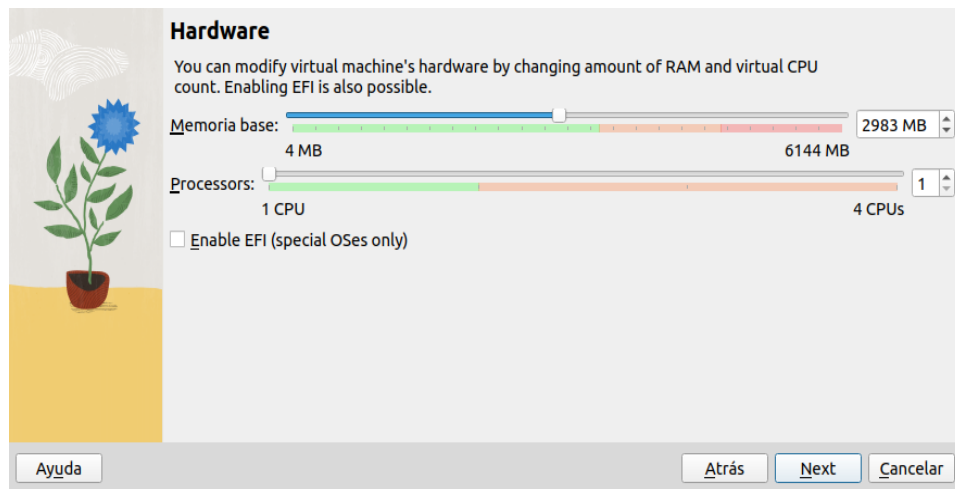


Figura 3.7 Selección de memoria RAM y núcleos de procesamiento

En el siguiente paso se seleccionó la capacidad de almacenamiento de la máquina virtual acorde a la selección de hardware, en este prototipo se asignaron 50 (GB) ya que se deben descargar las librerías de *Python* y el metabuscador, adicional a esto se tomó en consideración el almacenamiento mínimo que necesita el sistema operativo como indica en la Figura 3.8.

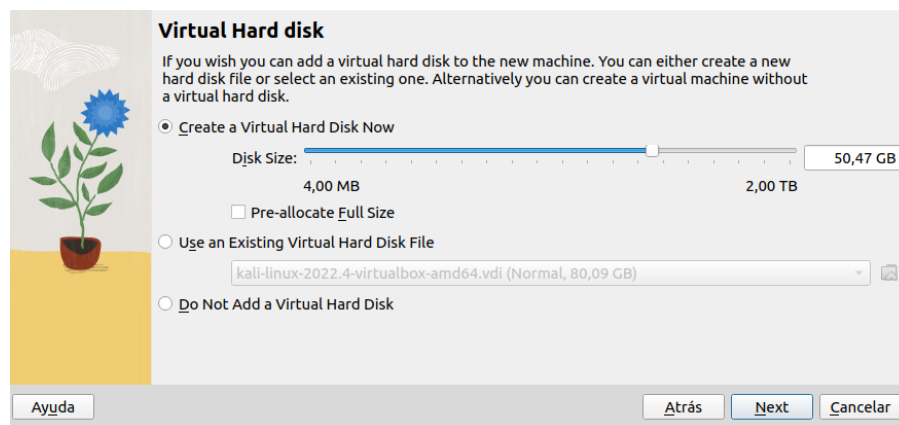


Figura 3.8 Selección de almacenamiento

Una vez creada la máquina virtual se ingresó a las configuraciones y a la opción de Red en la cual se encuentran los adaptadores, para este prototipo se eligió el adaptador puente, como se aprecia en la Figura 3.9.

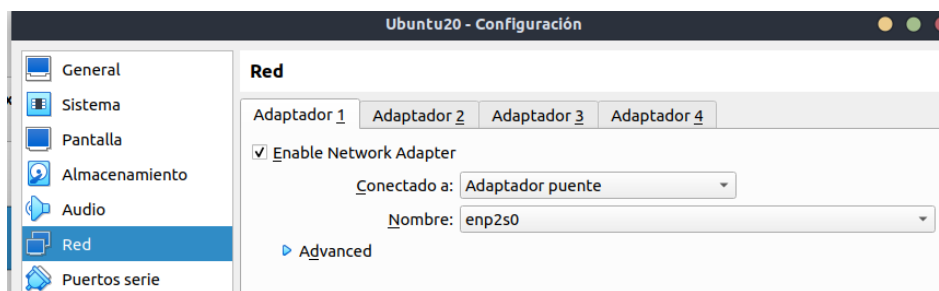


Figura 3.9 Configuración del adaptador de red

Instalación de paquetes

Es importante disponer los paquetes necesarios antes de la instalación de *SearXNG*, estos paquetes contienen archivos de encabezado y bibliotecas que son útiles para encontrar aplicaciones de *Python* que generalmente están basadas en la web, también se encuentran los módulos que permiten crear entornos virtuales y el uso de protocolos criptográficos SSL y TLS, para dicha instalación se usó el comando que se indica a continuación en la Figura 3.10.

```
server@Ubuntu20:~$ sudo -H apt-get install -y \
> python3-dev python3-babel python3-venv \
> uwsgi uwsgi-plugin-python3 \
> git build-essential libxslt-dev zlib1g-dev libffi-dev libssl-dev
```

Figura 3.10 Instalación de paquetes y librerías de Python

Creación del usuario y directorio

Se configuró el usuario *SearXNG* con *shell* predeterminado es decir en el directorio */bin/bash*, para la creación no se solicitó información adicional como la contraseña ya que se creó en el sistema, para finalizar se agregó un comentario el cual hace referencia a que dicho usuario se creó para el metabuscador, ver Figura 3.11.

```
server@Ubuntu20:~$ sudo -H useradd --shell /bin/bash --system \
> --home-dir "/usr/local/searxng" \
> --comment 'Privacy-respecting metasearch engine' \
> searxng
```

Figura 3.11 Creación del usuario y directorio para la instalación

En la Figura 3.12 se muestra la creación del directorio donde se encuentra el usuario *SearXNG*, para ello se usó el comando *mkdir* y asignó como propietario del directorio al usuario creado mediante el comando *chown*.

```
:-$ sudo -H mkdir "/usr/local/searxng"
:-$ sudo -H chown -R "searxng:searxng" "/usr/local/searxng"
:-$
```

Figura 3.12 Creación del directorio y cambio del propietario

Instalación de SearXNG

Para realizar la instalación se inició un *shell* interactivo con el nombre del usuario y dentro de este se clonó el repositorio donde se encuentra SearXNG mostrado en la Figura 3.13.

```
server@Ubuntu20:~$ sudo -H -u searxng -i
searxng@Ubuntu20:~$ git clone "https://github.com/searxng/searxng" \
> "/usr/local/searxng/searxng-src"
Clonando en '/usr/local/searxng/searxng-src'...
remote: Enumerating objects: 57782, done.
remote: Counting objects: 100% (322/322), done.
remote: Compressing objects: 100% (132/132), done.
remote: Total 57782 (delta 156), reused 299 (delta 153), pack-reused 57460
Recibiendo objetos: 100% (57782/57782), 34.90 MiB | 1024.00 KiB/s, listo.
Resolviendo deltas: 100% (35665/35665), listo.
```

Figura 3.13 Clonación del repositorio y replicación del contenido

Luego se creó un entorno virtual con la ayuda de *Python3* en el que posteriormente se replicó el contenido entre los directorios con el comando *echo*, como se puede visualizar en la Figura 3.14.

```
searxng@Ubuntu20:~$ python3 -m venv "/usr/local/searxng/searx-pyenv"
searxng@Ubuntu20:~$ echo ". /usr/local/searxng/searx-pyenv/bin/activate" \
> >> "/usr/local/searxng/.profile"
```

Figura 3.14 Entorno virtual y transferencia de archivos en los directorios

A continuación, se comprobó que la versión de *Python* es la 3.8 mostrada en la Figura 3.15, adicionalmente se deben actualizar las siguientes herramientas: *pip*, *setuptools*, *wheel*, *pyyaml*, esta actualización se la realizó mediante el comando *pip* como se observa en la Figura 3.16.

```
server@Ubuntu20:~$ sudo -H -u searxng -i
(searx-pyenv) searxng@Ubuntu20:~$ command -v python && python --version
/usr/local/searxng/searx-pyenv/bin/python
Python 3.8.10
(searx-pyenv) searxng@Ubuntu20:~$ /usr/local/searxng/searx-pyenv/bin/python
Python 3.8.10 (default, Nov 14 2022, 12:59:47)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> Python 3.8.10
```

Figura 3.15 Comprobación de la versión instalada de Python

```
~$ pip install -U pip setuptools wheel pyyaml
```

Figura 3.16 Actualización de herramientas

La instalación fue realizada en el entorno virtual, adicionalmente se ingresó al directorio que se muestra en la Figura 3.17 para instalar el metabuscador debido a que en esa carpeta se encuentran archivos necesarios para este proceso, finalmente se instaló SearXNG a través del comando *pip*.

```
(searx-pyenv) searxng@Ubuntu20:~$ cd "/usr/local/searxng/searxng-src"
(searx-pyenv) searxng@Ubuntu20:~/searxng-src$ pip install -e .
```

Figura 3.17 Instalación de SearXNG

Configuración de SearXNG

Para realizar las configuraciones fue necesario crear un directorio y copiar el contenido en el archivo *settings.yml* como se muestra en la Figura 3.18.

```
server@Ubuntu20:~$ sudo -H mkdir -p "/etc/searxng"
server@Ubuntu20:~$ sudo -H cp "/usr/local/searxng/searxng-src/utils/templates/e
tc/searxng/settings.yml" \
> "/etc/searxng/settings.yml"
```

Figura 3.18 Creación del directorio y creación del archivo de configuración

El siguiente paso se configuró el metabuscador para ello se debe modificar el archivo *settings.yml* el cual se indica en el ANEXO 7.1, en este caso se modificó:

- El nombre de la instancia el cual es “AnthonyXNG” (opcional).
- Cambio de *secret_key*.
- El puerto por el cual se realizan las peticiones.
- Una dirección por defecto.
- Agregación de motores de búsqueda.

Una vez finalizadas las configuraciones se modificó el apartado de *secret_key* por el texto que se ingresó en el archivo de configuración como se puede observar en la siguiente Figura 3.19.

```
server@Ubuntu20:~$ sudo -H sed -i -e "s/ultrasecretkey/${openssl rand -hex 16}/
g" \
> "/etc/searxng/settings.yml"
server@Ubuntu20:~$
```

Figura 3.19 Cambio de *secret_key* en el archivo de configuración

Comprobación

Para comprobar el funcionamiento se habilitó la depuración, esto se realiza fuera del entorno virtual y es importante para detectar fallos en caso de existir, para ello se usó el comando mostrado en la Figura 3.20.

```
server@Ubuntu20:~$ sudo -H sed -i -e "s/debug : False/debug : True/g" "/etc/sear
xng/settings.yml"
server@Ubuntu20:~$
```

Figura 3.20 Habilitación de la depuración para SearXNG

Una vez activada la depuración se ingresa nuevamente al entorno virtual y posteriormente al directorio en el que se instaló *SearXNG*, mediante el comando *export* se marcan las funciones que pasan a procesos en segundo plano, para este caso las configuraciones que se realizaron mediante el archivo *settings.yml*, para ejecutar el servicio es necesario de *Python* como se observa en la Figura 3.21.

```
server@Ubuntu20:~$ sudo -H -u searxng -i
(searx-pyenv) searxng@Ubuntu20:~$ cd /usr/local/searxng/searxng-src
(searx-pyenv) searxng@Ubuntu20:~/searxng-src$ export SEARXNG_SETTINGS_PATH="/etc/searxng/settings.yml"
(searx-pyenv) searxng@Ubuntu20:~/searxng-src$ python searx/webapp.py
```

Figura 3.21 Exportación de las configuraciones y ejecución de *SearXNG*

Finalmente se ingresó a un navegador y se usó *localhost* junto con el puerto establecido anteriormente con ello se puede visualizar en pantalla una ventana con el logo de *SearXNG* como se puede observar en la Figura 3.22 y de esta forma se pueden realizar las consultas.

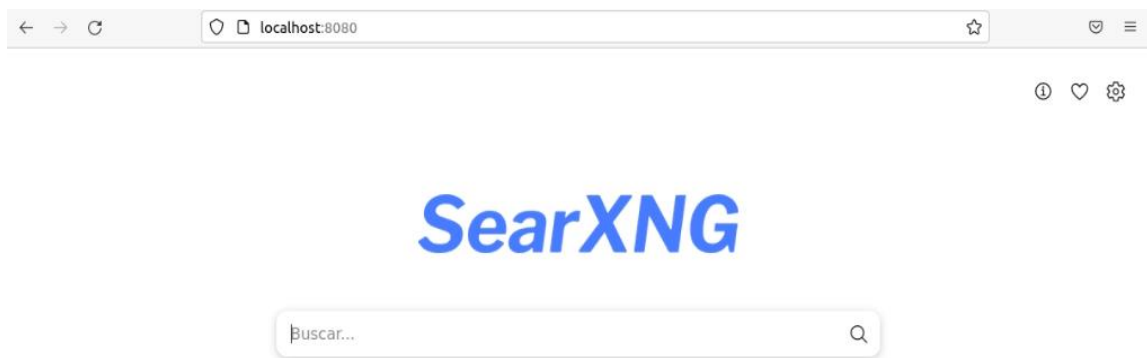


Figura 3.22 Acceso a *SearXNG* usando *localhost*

3.5 Pruebas de funcionamiento del prototipo

Para comprobar que el prototipo se encuentra funcionando correctamente se realizaron las pruebas mostradas a continuación en función del alcance planteado para este proyecto.

Acceso desde cualquier dispositivo

Para comprobar este punto se accede desde un ordenador como se muestra en la Figura 3.23 y desde un dispositivo móvil indicado en la Figura 3.24 para ingresar al metabuscador se debe ingresar la dirección *IP* y el número del puerto.

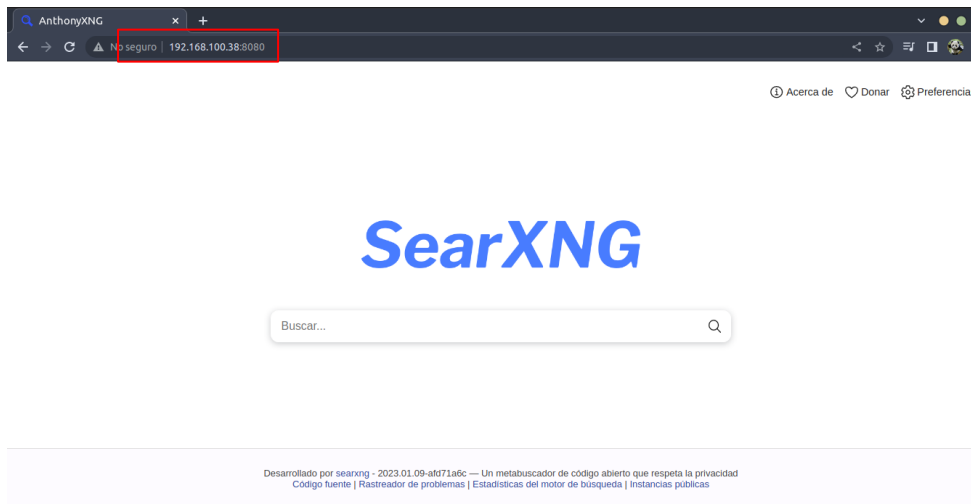


Figura 3.23 Acceso a *SearXNG* desde un ordenador

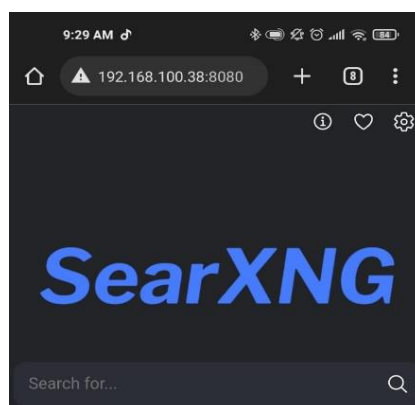


Figura 3.24 Acceso a *SearXNG* desde un dispositivo móvil

Búsquedas realizadas con más de un motor de búsqueda

Al realizar una consulta, en la parte inferior derecha del resultado se pueden observar los motores de búsqueda que se utilizaron para mostrar dicho resultado como se indica en la Figura 3.25.

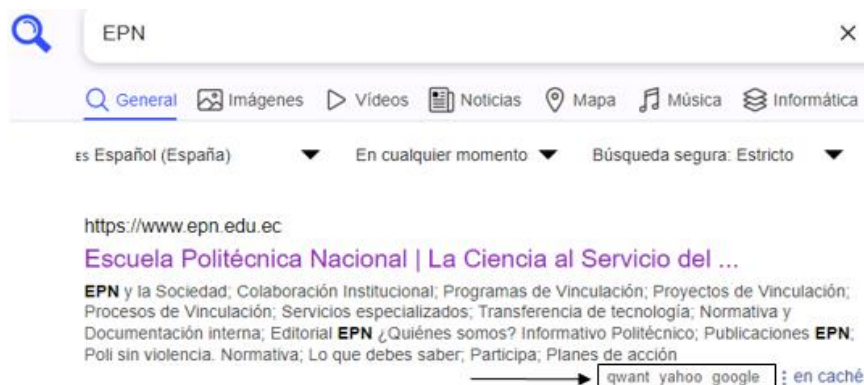


Figura 3.25 Resultado usando los motores qwant, Google y Yahoo

Datos de navegación almacenados de manera local

En la Figura 3.26 se muestra la configuración de *SearXNG*, en el apartado de *cookies* se encuentra la dirección en la que se almacenan las preferencias creadas, siendo este en *localhost*, además, en la parte inferior izquierda se puede observar que esta información solo será usada por el usuario más no por el navegador.



Figura 3.26 Almacenamiento de los datos de navegación

Vídeo demostrativo del funcionamiento

En la Figura 3.27 se muestra el código QR que pertenece al vídeo donde se muestra el funcionamiento del prototipo.



Figura 3.27 QR del vídeo demostrativo

4 CONCLUSIONES

- Como uno de los criterios principales para la selección del software se consideró el análisis de la documentación oficial de *SearXNG*, en la que se especifica la versión de *Python* que se necesita para la instalación, en base a ello se eligió la distribución Ubuntu 20.04 ya que esta cuenta con *Python* 3.8, dicha versión es

necesaria para trabajar con los paquetes y dependencias, necesarios para la instalación del metabuscador *SearXNG*.

- Como parte del diseño se estableció que la computadora portátil funcionará como servidor, debido a esto la dirección *IP* para esta máquina es estática, para esta configuración se consultó el direccionamiento dentro de la red cableada e inalámbrica a través del *router*, además, se verificó que ningún otro dispositivo dentro de esta red tenga asignada la dirección 192.168.100.38 la cual se configuró tanto en la máquina virtual y además fue reservada en el servidor DHCP del *router*.
- Un aspecto importante para la selección de *SearXNG* fue que admite que las consultas sean realizadas con más de un motor de búsqueda, permitiendo que tanto en las configuraciones del navegador, como en las configuraciones manuales desde el archivo se pueden modificar los motores de búsqueda que serán usados para dichas consultas. Cabe recalcar que la selección del motor de búsqueda a emplear se debe basar en sus indicadores claves de desempeño como su tiempo de respuesta y disponibilidad, esta información es proporcionada por *SearXNG* y se encuentra disponible en el apartado de motores de búsqueda en las configuraciones.
- Se concluyó que las características de hardware de la máquina sobre la cual se implementa el prototipo van a cambiar en función del entorno sobre el cual se está implementando el prototipo, puesto que la cantidad de dispositivos que usarán el prototipo tendrá relación directa sobre el almacenamiento, el procesamiento y la capacidad del adaptador de red ya que en un ambiente más grande el número de peticiones que el prototipo tendrá que procesar será mucho mayor.
- Para garantizar que el buscador pueda ser accedido desde cualquier dispositivo dentro de la red, la configuración del adaptador de red en la máquina virtual es el punto clave a considerar y se debe configurar en tipo puente para que trabaje con el adaptador del equipo físico y de esta forma la máquina virtual sea reconocida como un equipo más dentro de la red.
- Para que el prototipo pueda ser accedido desde cualquier dispositivo de la red y que el funcionamiento de este sea correcto es necesario realizar las siguientes modificaciones dentro del archivo *settings.yml*: el puerto por el que se da el servicio, una dirección *IP* por defecto y adicionalmente se debe cambiar el apartado de *secret_key* este último es obligatorio ya que de no hacerlo se mostrará un mensaje indicando que se debe realizar dicho cambio.

- Se evidenció que el prototipo implementado permite garantizar la privacidad del usuario, es por esta razón que las configuraciones y los datos de navegación que se generan al realizar las consultas se almacenan de forma local en la máquina virtual donde se encuentra el metabuscador, esto se visualiza en el apartado de preferencias, en cookies en esta pestaña se indica la URL donde se guarda dicha información.

5 RECOMENDACIONES

- Al momento de iniciar el servidor es necesario ingresar una serie de comandos, los cuales llevan tiempo debido a su extensión en alguno de ellos, por lo que para ahorrar tiempo se recomienda realizar un *script* el cual contenga los comandos de inicio que van desde la depuración hasta la ejecución del servidor y que de esta manera sea más sencilla la puesta en marcha del servicio, para ello se debe tener conocimiento en la creación y ejecución de *scripts* dentro de *Linux*.
- Ya que el metabuscador proporciona una lista de los motores de búsqueda con información acerca de la disponibilidad y tiempo de respuesta se recomienda siempre tener activado únicamente los motores que posean tiempos de respuesta bajos y porcentajes de disponibilidad altos, además, debido a que estos se encuentran cambiando se debe realizar una comprobación antes de realizar cualquier tipo de consulta.
- Para implementaciones futuras de este prototipo se recomienda revisar la documentación oficial de *SearXNG*, puesto que si los requisitos de Python o librerías cambian se debe realizar un nuevo análisis para elegir un sistema operativo compatible.
- En la *safe_search* se sugiere tener siempre el nivel 2 para aumentar el nivel de seguridad del buscador, esto permite hacer un filtrado en los resultados mostrados.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. J. Glover, S. Lawrence, W. P. Birmingham y L. Giles, «ResearchGate,» Febrero 2000. [En línea]. Available: https://www.researchgate.net/publication/2596239_Architecture_of_a_Metasear

ch_Engine_that_Supports_User_Information_Needs. [Último acceso: 18 Diciembre 2022].

- [2] E. SearXNG, «Welcome to SearXNG,» 2021. [En línea]. Available: <https://docs.SearXNG.org/index.html>. [Último acceso: 16 Diciembre 2022].
- [3] E. SearXNG, «Why use a private instance?,» 2021. [En línea]. Available: <https://docs.SearXNG.org/own-instance.html>. [Último acceso: 17 Diciembre 2022].
- [4] R. Wiki, «Indexación,» RYTE Wiki, 2021. [En línea]. Available: <https://es.ryte.com/wiki/Indexaci%C3%B3n>.
- [5] Python, «Python,» Python, 9 Febrero 2023. [En línea]. Available: <https://docs.python.org/es/3.8/library/venv.html>. [Último acceso: 28 January 2023].
- [6] S. S.L., «stackscale,» 04 30 2020. [En línea]. Available: <https://www.stackscale.com/es/blog/ubuntu-20-04->. [Último acceso: 19 Diciembre 2022].
- [7] «Github,» 16 Diciembre 2022. [En línea]. Available: <https://github.com/SearXNG/SearXNG/blob/master/requirements.txt>. [Último acceso: 19 Diciembre 2022].
- [8] E. SearXNG, «Step by step installation,» 2021. [En línea]. Available: <https://docs.SearXNG.org/admin/installation-SearXNG.html>. [Último acceso: 19 Diciembre 2022].
- [9] Dirigentesdigital, «Dirigentesdigital,» 20 Junio 2022. [En línea]. Available: <https://dirigentesdigital.com/tecnologia/como-funciona-la-indexacion-de-los-motores-de-busqueda#:~:text=La%20indexaci%C3%B3n%20consiste%20en%20a%C3%B1adir,registran%20millones%20de%20p%C3%A1ginas%20web..> [Último acceso: 19 Diciembre 2022].
- [10] profesionalreview, «profesionalreview,» 2021. [En línea]. Available: <https://www.profesionalreview.com/intel/intel-core-i5/>. [Último acceso: 21 Diciembre 2022].

- [11] P. Requisitos, «PC Requisitos,» [En línea]. Available: <https://pcrequisitos.com/ubuntu-20-04/>. [Último acceso: 21 Diciembre 2022].
- [12] IBM, «IBM,» 05 Marzo 2021. [En línea]. Available: <https://www.ibm.com/docs/es/elm/6.0.1?topic=samples-installing-context-aware-search-tool>. [Último acceso: 21 Diciembre 2022].
- [13] H. Rodríguez, «¿Qué son los motores de búsqueda? ¡Hay vida más allá de Google!,» 20 Julio 2021. [En línea]. Available: <https://www.crehana.com/blog/negocios/que-son-los-motores-de-busqueda/>. [Último acceso: 21 Diciembre 2022].
- [14] danitic, «DANITIC,» 24 Octubre 2018. [En línea]. Available: <https://danitic.wordpress.com/2018/10/24/diferencias-entre-nat-red-nat-adaptador-puente-internal-y-solo-anfitrion-en-virtualbox/#:~:text=El%20adaptador%20puente%20hace%20que,el%20anfitri%C3%B3n%2C%20y%20a%20Internet%20bidireccionalmente..> [Último acceso: 21 Diciembre 2022].
- [15] vmware, «vmware,» 26 Julio 2021. [En línea]. Available: <https://docs.vmware.com/es/VMware-Horizon/2106/horizon-architecture-planning/GUID-10ED49A9-56B6-4BD7-8425-94900F53AB02.html>. [Último acceso: 21 Diciembre 2022].
- [16] Altare, «Altare,» [En línea]. Available: <https://altaresp.es/que-es/memoria-ram/>. [Último acceso: 21 Diciembre 2022].

7 ANEXOS

La lista de los anexos se muestra a continuación:

ANEXO I. Certificado de originalidad.

ANEXO II. Archivo de configuración.

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de febrero de 2023

De mi consideración:

Yo, LEANDRO ANTONIO PAZMIÑO ORTIZ, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE BUSCADOR WEB BASADO EN LINUX PARA GARANTIZAR LA PRIVACIDAD DE LOS DATOS DEL USUARIO elaborado por ANTHONY BRYAN ENCALADA DÁVILA estudiante de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 12 %.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

[Reporte de similitud - AEncalada.pdf](#)

Atentamente,



Leandro Antonio Pazmiño Ortiz

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Archivo de configuración

```
#SearXNG settings

use_default_settings: true

general:

    debug: false

    instance_name: "AnthonyXNG"

search:

    safe_search: 2

    autocomplete: "duckduckgo"

server:

    port: 8080

    bind_address: "0.0.0.0"

    base_url: false

    limiter: false

    secret_key: "anthonysear"

    limiter: true

    image_proxy: true

engines:

    - name: google
      use_mobile_ui: true

    - name: yahoo
      engine: yahoo
      shortcut: yh
      disabled: false

    - name: qwant
      engine: qwant
      shortcut: qw
      disabled: false
```

ANEXO 7.1 Archivo de configuración *settings.yml*