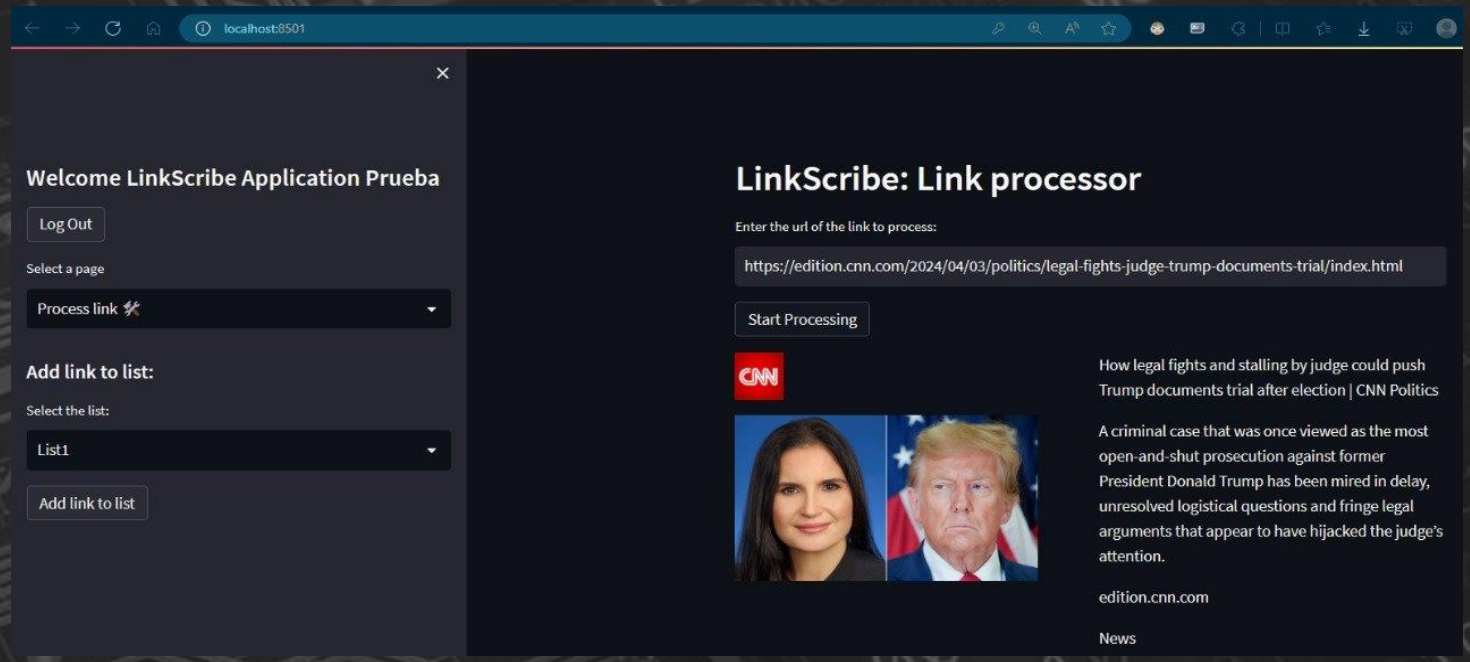
LINKSCRIBE

YGALVES/WEBSCRAPPING (GITHUB.COM)



- 2235650 GUILLERMO LEON ZAPATA ÁLVAREZ
- 2235326 JOAQUIN ANDRES ALARCON GUEVARA
- 2235918 JAIRO ALBERTO VÉLEZ GIRALDO
- 2237389 YONILIMAN GALVIS AGUIRRE

NECESIDAD DEL CLIENTE

La empresa SearchInt, quien ofrece servicios de desarrollo de software dirigido a centros de investigación y universidades, requiere desarrollar una aplicación web de nombre LinkScribe, que utilice NLP, agilizando los procesos de investigación al categorizar y resumir el contenido de páginas web que podrán servir de referencia para las diferentes necesidades de los equipos.

OBJETIVOS DEL PROYECTO

- Desarrollar una aplicación intuitiva y fácil de usar para la creación y organización de listas de enlaces.
- Implementar técnicas de Procesamiento del Lenguaje Natural (NLP) para la extracción de información de los enlaces y su categorización automática.
- Proporcionar a los usuarios la capacidad de crear y gestionar categorías personalizadas para sus listas de enlaces.
- Ofrecer una experiencia de usuario atractiva y fluida en la APP.

REQUERIMIENTOS

NECESIDADES DEL CLIENTE

1. Fácil uso: Los usuarios pueden simplemente copiar y pegar un enlace web.
2. Procesamiento automático extrayendo información sobre el contenido de la página y clasificándolos de acuerdo a la información obtenida, por ejemplo, news, technology, education, travel, etc.
3. LinkScribe utiliza NLP para extraer automáticamente información relevante de los enlaces, incluyendo el título, la descripción y la imagen de vista previa.
4. Acceso con credenciales de usuario con password encriptado.

RESTRICCIONES TÉCNICAS

1. Solo en Idioma Inglés.
2. Caractéres del alfabeto internacional.
3. Sin autoscaling
4. No hay variables de entorno

EQUIPO DE TRABAJO

Líder de Proyecto

- Vigilancia de cumplimiento de los Requisitos de la App.
- Control de Cambios e hitos.
- Control de Pruebas.
- Análisis de experiencia de Usuario

Infraestructura, Base de Datos, Gobernabilidad

Full Stack Eng.
- Planeación y Diseño de la infraestructura de la app

DBA Engineer.
- Planeación y Diseño del modelo adecuado para el manejo de los datos y sus relaciones.

Cyber - security Engineer:
- Diseño e implementación de metodos de seguridad informática eficientes para garantizar la integridad y confidencialidad de la información en línea.

Back-end

Data Scientist
- Encargado de realizar el tratamiento adecuado a los datos usados para entrenamiento y pruebas, adaptar la infraestructura para obtener datos futuros para re-entrenamiento

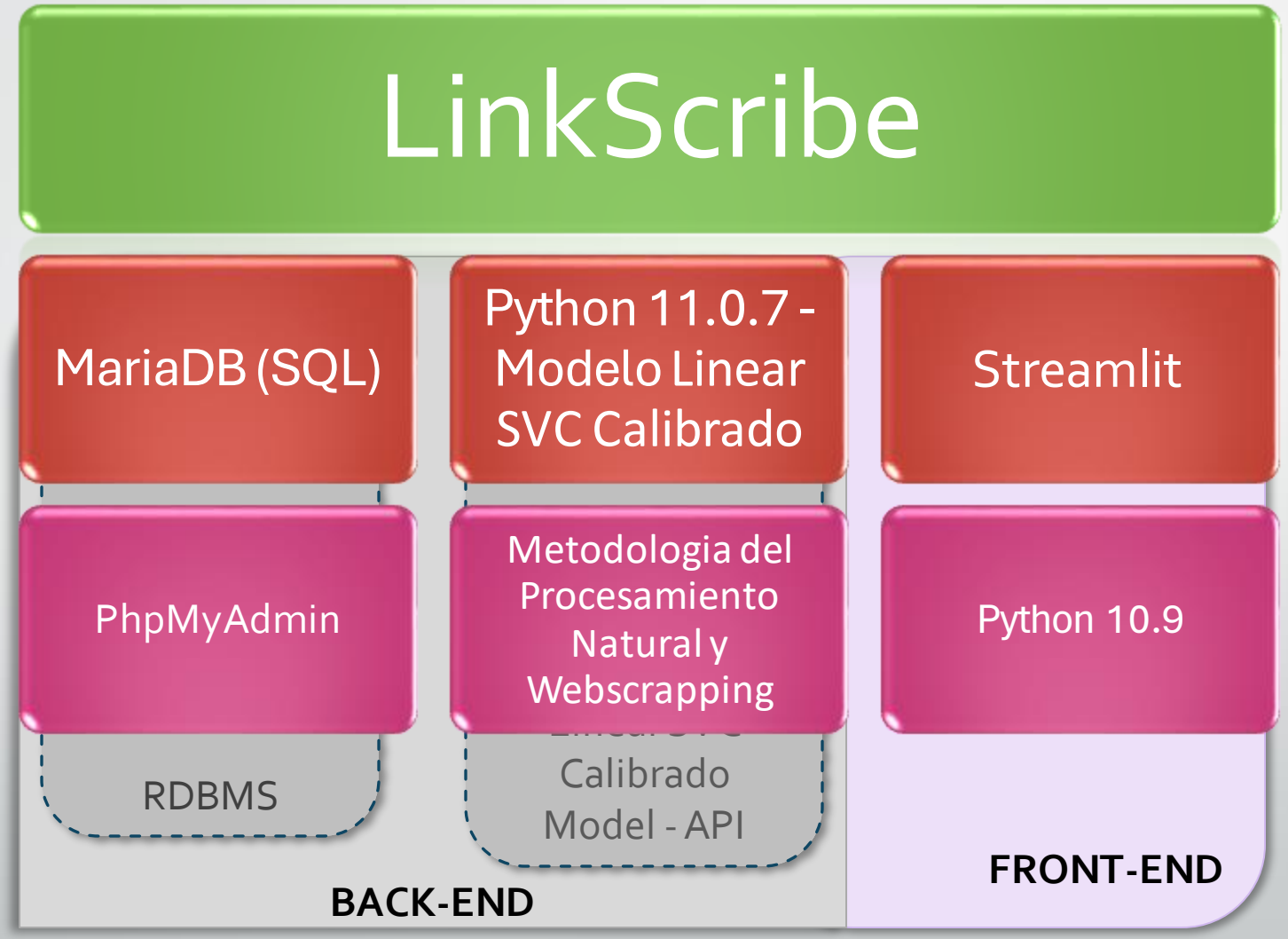
Machine Learning Eng.
- Encargado de la seleccion adecuado del modelo, entrenar, realizar pruebas y verificar los resultados

Front-End

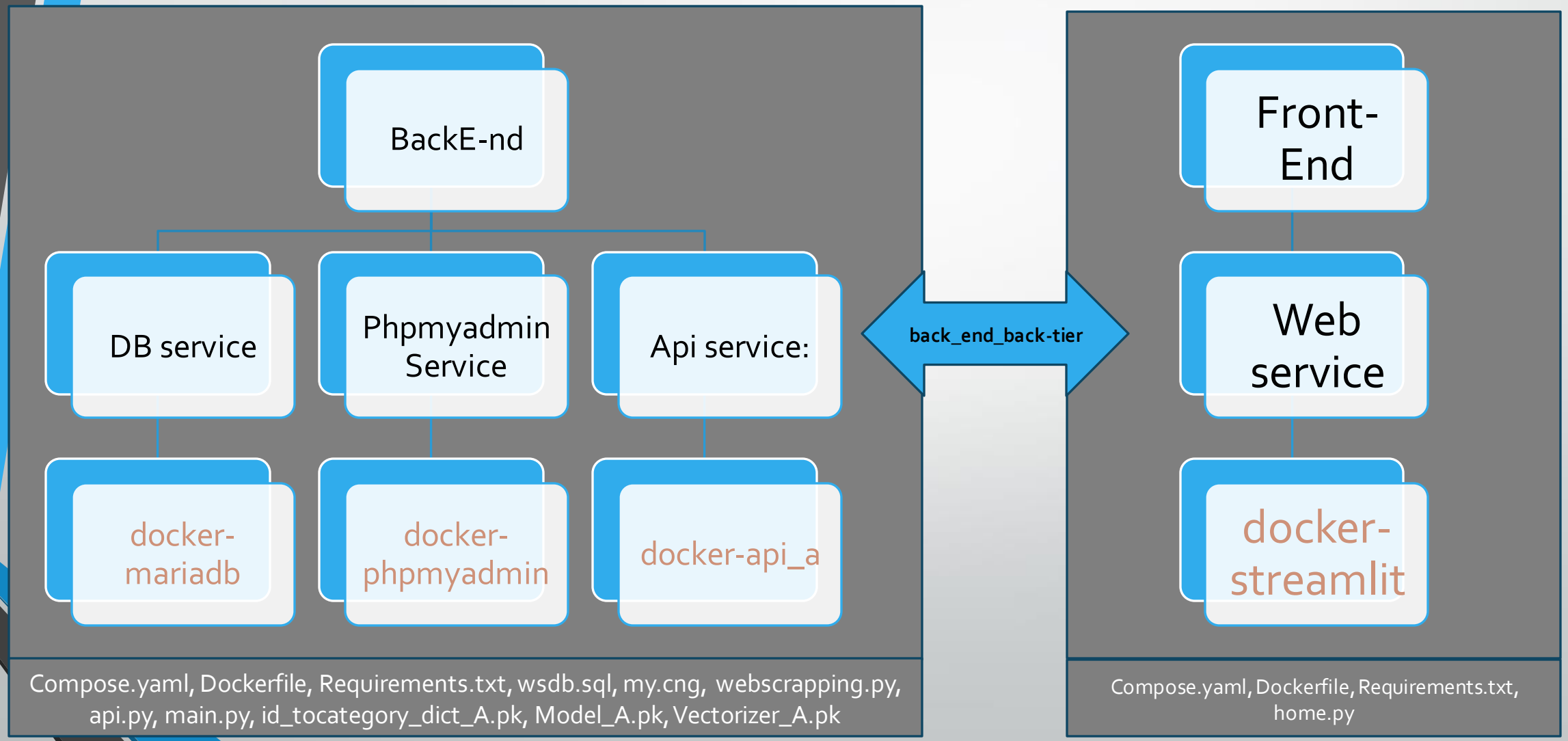
Frontend Eng.
- Crear la interface gráfica (WEB) de forma multiplataforma y asegurar la experiencia de usuario.

Analista de Pruebas

ARQUITECTURA DEL PROYECTO



STACK DEL PROYECTO



TECNOLOGÍAS Y LIBRERÍAS UTILIZADAS

Gestión de Proyecto

- GitHub: Sistema de gestión de versiones, tareas y grupos de trabajo
- Pyenv: Manejador de versiones.
- Poetry: Herramienta de gestión utilizada para gestionar las dependencias y las versiones del proyecto.

Stack

- Docker: Contenerizar la aplicación
- Docker Compose: facilitador de experimentos y despliegue.
- MariaDB/MySQL: Bases de datos relacional
- PhpMyAdmin: Gestor Web Base de datos

BackEnd

- Spacy : Biblioteca de código abierto para procesamiento de lenguaje natural (NLP)
- EncoreWeb: Modelo NLP Spacy en texto en ingles
- Beautiful soup (BS4): Biblioteca de codigo abierto para realizar WebScrapping

API

- FastAPI: Framework web de alto rendimiento para desarrollar API con Python
- Uvicorn: Framework web utilizado para desarrollar la API RESTful del backend.
- Lxml: Enlace de Python con libxml2 y libxslt (lenguaje xml / html)

FrontEnd

- Streamlit: Biblioteca utilizada para desarrollar la interfaz de usuario web usando lenguaje Python
- MySQL connector: Librería que provee coneccion a MariaDB/MySQL server para programas cliente.

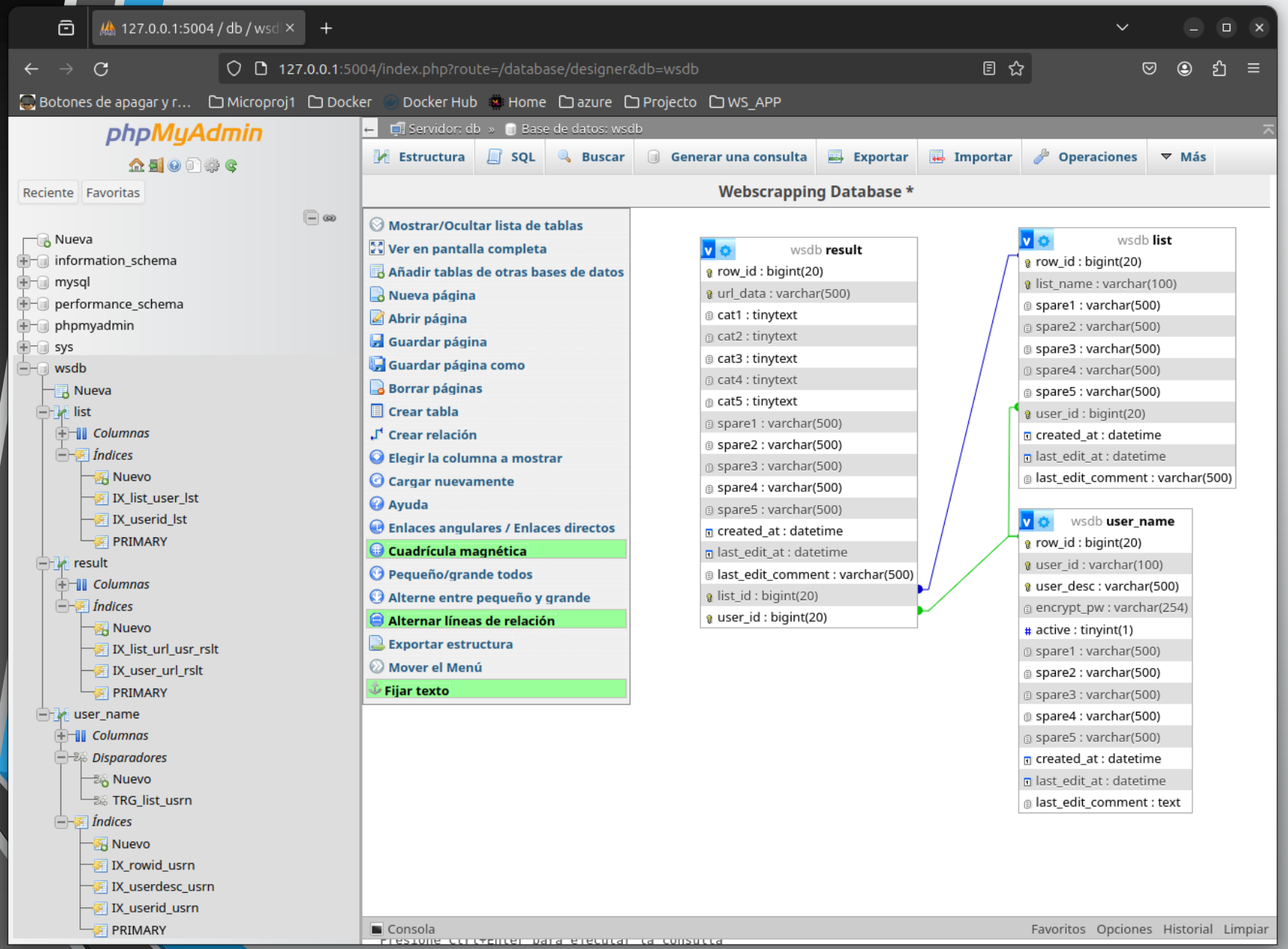
- Otras librerías importantes: Pydantic, Contextlib, OS, Numpy, Scikit-learn, Pathlib, Typing, Pandas, Requests, Numba, Json, Net-tools, Netcat etc.
- Vs Studio extensions: Github, Docker, Azure, Prettier, usuarios Windows: WSL, Ubuntu

DESPLIEGUE Y EJECUCIÓN

1. Clonar el repositorio del proyecto desde GitHub.
2. Instalar Docker y Docker Compose en su sistema si aún no están instalados.
3. Ejecutar docker-compose up en el directorio raíz del proyecto para construir y ejecutar la aplicación.
4. Acceder a la aplicación desde su navegador web utilizando la URL proporcionada por Docker.

BASE DE DATOS RELACIONAL

- Estructura de los Datos Definida
- Integridad de los Datos
- Lenguaje SQL estandar.
- Tareas automatizadas (Stored Procedures)
- Relaciones entre tablas (index)
- Disparadores que ejecutan acciones automáticas (actualizacion, escritura o borrado de registros)
- Permite conservar la seguridad de datos ocultandolos o encriptandolos
- Gestion del sistema para respaldos, reindexado, reorganizacion sobre las tablas de registros
- Tablas con relaciones cruzadas con multiples bases de datos
- Uso de analisis estadistico para mejorar el rendimiento
- Seguimiento a Base de Datos (control de cambios)
- Implementacion de DB Espejo para Alta disponibilidad



MILESTONES BASES DE DATOS

REQUISITOS DE INSTALACIÓN

- Si tienes windows debes instalar WSL para tener una terminal ubuntu en tu sistema, esta creará una unidad Ubuntu donde luego podrás clonar este repositorio.
- Instala Visual Studio + la extension de WSL, recomendado instalar las extensiones de Docker, GitHub (no es necesario Github-copilot ya que requiere cuenta de pago y no es necesaria para este ejemplo), recomendado instalar las extenciones de prettier para yaml, toml, php, html
- Una vez en la terminal ubuntu es necesario installar pip3, Docker, Docker Compose
- clona este repositorio desde la teminal de ubuntu, por defecto la carpeta debera crearse en ~/home/Webscrapping
- Usando Visual Studio abre area de trabajo y busca la carpeta (en la unidad de ubuntu) donde clonaste este git y abre webscrapping.code-workspace
- herramienas que puedes instalar y usar para verificar las redes en ubuntu: netstat, netcat, nmap
- test

Microsoft Azure Estimate

Su presupuesto

Service category	Service type	Custom name	Region	Description	Estimated monthly cost	Estimated upfront cost
Contenedores	Azure Container Registry		Korea Central	Nivel Basic, registro 1 x 30 días, 0 GB Almacenamiento adicional, Compilación de contenedor- 1 CPU x 1 Segundos - Tipo de transferencia entre regiones, 5 GB de transferencia de datos de salida desde Korea Central a East Asia	$5,00	$0,00
Compute	Azure Container Instances		Korea Central	1 grupos de contenedores x 2.592.000 segundos, Linux sistema operativo, Pago por uso, 4 GB memoria, 2 vCPU	$81,82	$0,00
Support			Support		$0,00	$0,00
			Licensing Program	Microsoft Customer Agreement (MCA)		
			Billing Account			
			Billing Profile			
			Total		$86,82	$0,00

BIBLIOGRAFIA

- **Imodel:**

- hetulmehta, Hetul Mehta, Kaggle Expert, Mumbai, Maharashtra, India, Technical Head at DataZen https://www.kaggle.com/code/hetulmehta/classification-of-websites

- pagutierrez, Pedro Antonio Gutiérrez, Ph.D Computer Science and Artificial Intelligence, Spain, University of Córdoba https://notebook.community/pagutierrez/tutorial-sklearn/notebooks-spanish/11-extraccion_caracteristicas_texto - https://jsonlink.io/

- **Dataset:**

- website_classification.csv

- 1408 samples

- 3 columns ['website_url','cleaned_website_text','Category']

- 16 categories