

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS



COMPUTACIÓN TOLERANTE A FALLAS

I7036

SECCIÓN D06

PROYECTO FINAL

BAUTISTA MARTINEZ EDEN ABDIEL

CÓDIGO: 221350524

PROFESOR: MICHEL EMANUEL LOPEZ FRANCO

Introducción

El siguiente documento tiene como objetivo proporcionar una descripción detallada del proyecto final de la materia, abordando sus objetivos, alcance, requerimientos, arquitectura, y otros aspectos relevantes para su desarrollo y despliegue.

El proyecto tiene como objetivo implementar un proyecto con ciertas características, como el uso de contenedores, kubernetes, microservicios. Este documento del proyecto establecerá los límites y alcances específicos de las diferentes fases y componentes del proyecto.

Para el proyecto se utilizaron las siguientes tecnologías:

- Python (PySide2 para la interfaz grafica, Requests)
- Php
- Dockers
- Xming
- Kubernetes

Mismas que serán necesarios para su despliegue.

A continuación el proceso que se tomó para su realización

Desarrollo

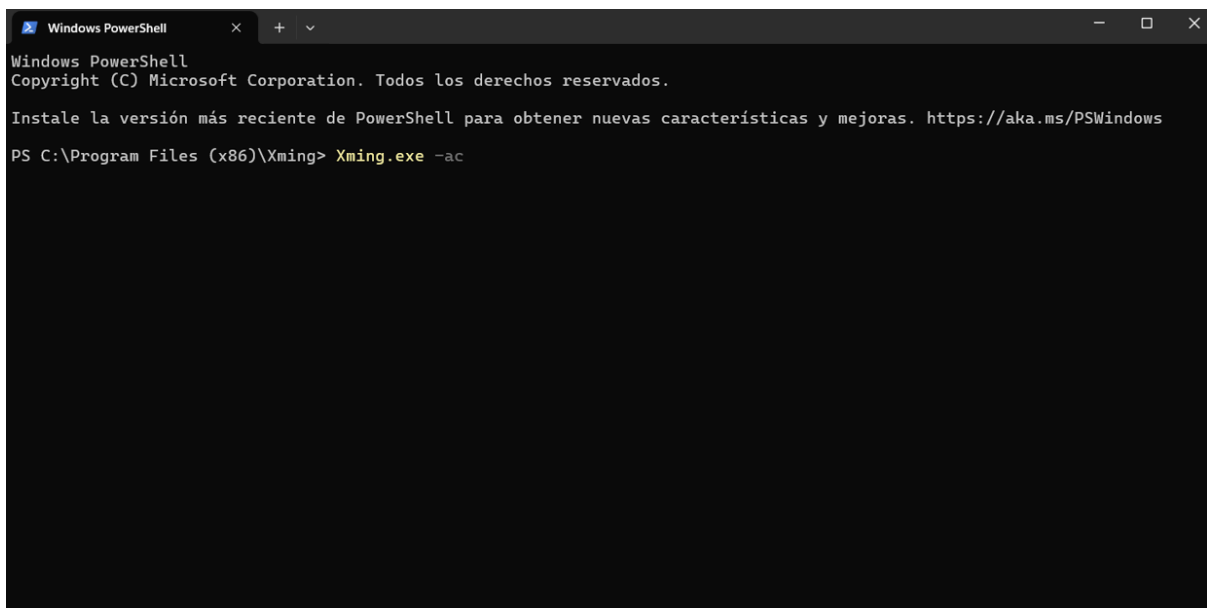
Dockerfile para el proyecto

```
1  FROM python:3.10
2  WORKDIR /app
3  COPY . /app
4
5  COPY pip.txt ./
6
7  RUN apt-get update && apt-get install -y \
8      libgl1-mesa-glx \
9      libxkbcommon-x11-0 \
10     libxcb-icccm4 \
11     libxcb-image0 \
12     libxcb-keysyms1 \
13     libxcb-randr0 \
14     libxcb-render-util0 \
15     libxcb-render0 \
16     libxcb-shape0 \
17     libxcb-sync1 \
18     libxcb-xfixes0 \
19     libxcb-xinerama0 \
20     libxcb-xkb1 \
21     libxcb1 \
22     libxrender1 \
23     libxi6 \
24     libdbus-1-3 \
25     libxcb-cursor0 \
26     libegl1 \
27     && apt-get clean \
28     && rm -rf /var/lib/apt/lists/*
29  RUN pip install -r pip.txt
30
31  COPY . .
32  ENV PYTHONDONTWRITEBYTECODE 1
33  ENV PYTHONUNBUFFERED 1
34
35  CMD ["python3", "main.py"]
36
```

Docker-compose para orquestar los microservicios

```
1  version: '3.8'
2
3  services:
4    frontend:
5      build:
6        context: ./frontend
7        dockerfile: Dockerfile
8      ports:
9        - "5000:5000"
10
11    backend:
12      build:
13        context: ./backend
14        dockerfile: Dockerfile
15      ports:
16        - "5001:5001"
17      depends_on:
18        - frontend
19
```

Una vez que se construye el proyecto, se debe de preparar el entorno para ejecutarlo, para ello se necesita de Xming para mostrar la GUI

A screenshot of a Windows PowerShell terminal window. The title bar reads "Windows PowerShell" with standard window controls. The terminal text includes: "Windows PowerShell", "Copyright (C) Microsoft Corporation. Todos los derechos reservados.", a message about installing the latest PowerShell version with a link to <https://aka.ms/PSWindows>, and the command "PS C:\Program Files (x86)\Xming> Xming.exe -ac" at the prompt.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Program Files (x86)\Xming> Xming.exe -ac
```

Se debe de preparar el DISPLAY

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Adaptador de LAN inalámbrica Conexión de área local* 2:

Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . :

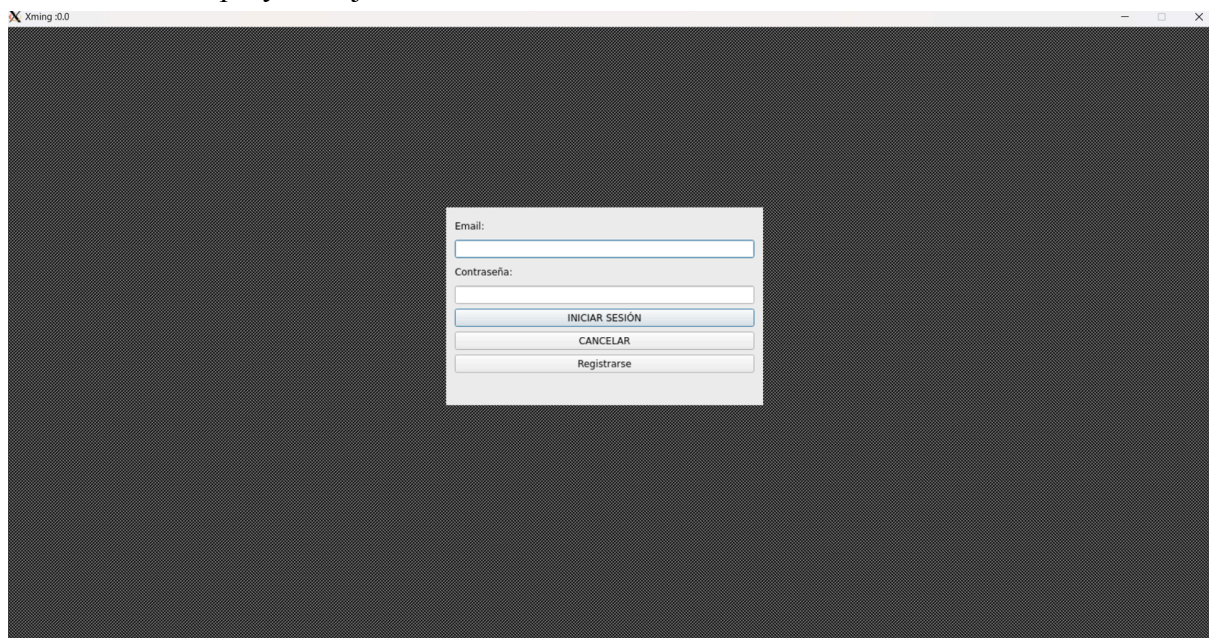
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . :
Dirección IPv6 . . . . . : fd54:5284:1081:b200:658d:c795:e097:748d
Dirección IPv6 temporal. . . . . : fd54:5284:1081:b200:5cb5:2ab4:c096:99ac
Vínculo: dirección IPv6 local. . . : fe80::f55a:3ef7:aef2:9875%7
Dirección IPv4. . . . . : 192.168.3.28
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . : fe80::5652:84ff:fe10:81b2%7
                                         192.168.3.1
PS C:\Users\edenb\Desktop\Servicio Automotriz> set DISPLAY=192.168.3.28:0.0
```

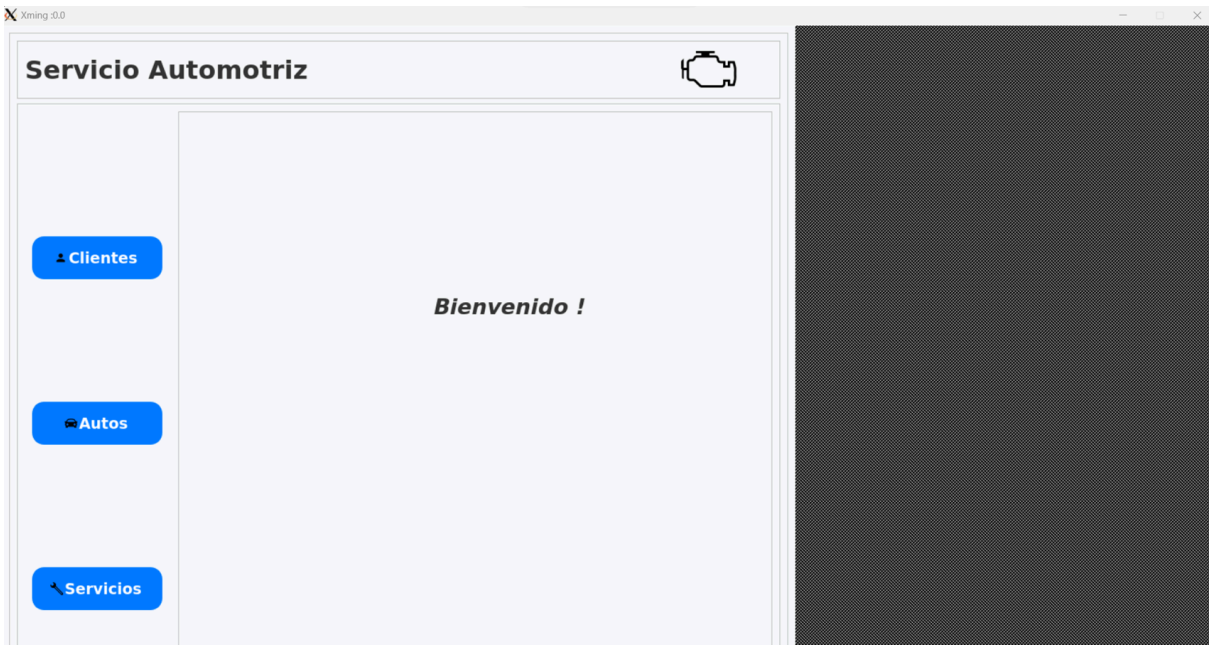
Y ejecutar

```
PS C:\Users\edenb\Desktop\Servicio Automotriz> docker run -e DISPLAY=host.docker.internal:0 -v /tmp/.X11-unix:/tmp/.X11-unix servicio
qt.qpa.xcb: X server does not support XInput 2
failed to get the current screen resources
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
qt.qpa.xcb: QXcbConnection: XCB error: 1 (BadRequest), sequence: 165, resource id: 90, major code: 130 (Unknown), minor code: 47
qt.qpa.xcb: QXcbConnection: XCB error: 170 (Unknown), sequence: 180, resource id: 90, major code: 146 (Unknown), minor code: 20
```

Podremos ver el proyecto ejecutandose



Y funcionando



<input type="checkbox"/>		de25d17340d1	servicio	Running	0%	2 minutes ago			
<input type="checkbox"/>		k8s_POD_kube-apiserver-docker-5af70380dd7b	registry.k8s.io/pause:3.9	Running	0%	2 minutes ago			
<input type="checkbox"/>		k8s_POD_etcd-docker-desktop-18e324b6c26f3	registry.k8s.io/pause:3.9	Running	0%	2 minutes ago			
<input type="checkbox"/>		k8s_POD_kube-controller-manager-e11268569718	registry.k8s.io/pause:3.9	Running	0%	2 minutes ago			
<input type="checkbox"/>		k8s_POD_kube-scheduler-docker-fbd2a716aedf	registry.k8s.io/pause:3.9	Running	0%	2 minutes ago			
<input type="checkbox"/>		k8s_kube-apiserver_kube-apiserver-369affa672dd	sha256:8a9000f98a528cbf5c11adee	Running	3.8%	2 minutes ago			
<input type="checkbox"/>		k8s_etcd_etcd-docker-desktop-19972883d4852	sha256:a0eed15eed4498c145ef2f1883f	Running	2.64%	2 minutes ago			
<input type="checkbox"/>		k8s_kube-scheduler_kube-scheduler-b583bc299394	sha256:6fc5e6b7218c753ea7ebbd03205	Running	0.62%	2 minutes ago			
<input type="checkbox"/>		k8s_kube-controller-manager_kube-controller-manager-de25d17340d1	sha256:138fb5a3a2e3d1df1ef83e14e56	Running	3.11%	2 minutes ago			

Conclusión

Durante el desarrollo de este proyecto, existieron una serie de desafíos. Desde el diseño inicial hasta la implementación final, cada etapa del proyecto proporcionó valioso conocimiento y oportunidades de aprender.

La integración de múltiples tecnologías y componentes dentro de un solo sistema fue compleja. Incorporar tecnologías como Python y sus respectivas librerías fue la parte más sencilla puesto que se tiene experiencia previa.

La implementación de contenedores Docker y orquestación con Kubernetes representó un desafío significativo. Docker permitió encapsular y distribuir cada microservicio de manera eficiente, mientras que Kubernetes nos brindó la escalabilidad y la gestión automatizada necesarias para desplegar y mantener la aplicación en entornos distintos.

Este proyecto fue complejo pero aun así se logró completarlo y al final es una gran satisfacción poder completarlo.