



UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

Introducción a las bases de datos

Ingeniería en ciencias de la computación

RECSIN

Manual técnico

GRUPO: 6CC2

ISAAC GONZÁLEZ AGUILERA 368048

ANGEL ALONSO LOPEZ QUIÑONEZ 355302

LUIS ARTURO HERNANDEZ CASTILLO 367685

Chihuahua, Chihuahua 28 de Mayo del 2025

Índice


Manual técnico.....	1
Introducción.....	3
Requisitos del sistema.....	3
Instalación.....	4
Inicio del programa.....	4
Guía de uso.....	5

Introducción

El programa es una aplicación web que se desarrolló para cubrir la necesidad de digitalizar todos los requisitos en un local donde se generan servicios técnicos a computadoras e impresoras, en este lugar por curioso que parezca no tienen una computadora tan potente en el mostrador por lo que se tuvo que adaptar a la necesidad de no tener tanto poder computacional y con una base de datos en la nube para apaciguar más esta necesidad.

Se basó en la simpleza del programa para que cualquier persona en el mostrador pueda hacer el servicio de manera sencilla y no se confunda, a continuación se podrá saber el cómo es que se usa.

Requisitos del sistema

Componente	Requisito Mínimo	Recomendado	
Procesador (CPU)	Dual Core (Intel i3 o AMD equivalente)	Quad Core (Intel i5 o superior)	
Memoria RAM	4 GB	8 GB o más	
Almacenamiento	128 GB HDD o SSD (espacio libre para navegador y caché)	256 GB SSD (para mejor rendimiento)	
Tarjeta de red	Ethernet o Wi-Fi (compatible con red local o internet)	Wi-Fi 5/6 o Ethernet Gigabit	
Tarjeta gráfica	Integrada (Intel HD Graphics o similar)	Cualquier GPU integrada o dedicada moderna	
Monitor	Resolución mínima de 1366x768	Resolución Full HD (1920x1080) o superior	
Periféricos	Teclado, ratón, bocinas (si la app tiene audio)	Opcionales: micrófono, cámara web (si la app lo pide)	
Sistema operativo	Windows 10+, macOS 10.14+, Ubuntu 18.04+	Última versión estable del sistema operativo	
Navegador web	Chrome, Firefox, Edge, Safari (actualizado)	Chrome o Firefox en su última versión	

Instalación

La instalación de la aplicación para el uso no es más que descargar el repositorio que tenemos de git, donde el único condicionamiento que se tiene que hacer es la ruta de donde se guardan las credenciales de la base de datos que están alojadas en la nube de la siguiente manera:

```
server-flask > functions > db.py > conectar_db
1  import oracledb
2
3  def conectar_db():
4      return oracledb.connect(
5          user="ADMIN",
6          password="contraRECSIN61%",
7          dsn="recsin_high",
8          config_dir= r"C:\Users\Insqn\Downloads\Wallet_RECSIN",
9          wallet_location=r"C:\Users\Insqn\Downloads\Wallet_RECSIN",
10         wallet_password="contraRECSIN61%",
11     )
12
```

La ruta que se muestra es la ruta absoluta sin formato para que no genere conflictos, las credenciales se les otorgarán directamente por temas de confidencialidad y seguridad.

Además de esto se tienen que descargar aparte las dependencias de npm y en un entorno virtual de python descargar las librerías que están especificadas en los 3 archivos readme que tenemos en el repositorio.

Inicio del programa

El inicio del programa de manera local se hará de la siguiente manera, muy sencillo y fácil para todos.

Nos iremos a la carpeta en la que está ubicado el frontend, y correremos el siguiente comando, claro que todo desde la terminal, no importa el sistema operativo, aunque está contemplado verse en windows

```
PS C:\Users\lnsqn\Desktop\ProyectoFinalDB> cd .\RECSIN\
PS C:\Users\lnsqn\Desktop\ProyectoFinalDB\RECSIN> npm run dev

> recsin@0.0.0 dev
> vite

VITE v6.3.5 ready in 694 ms

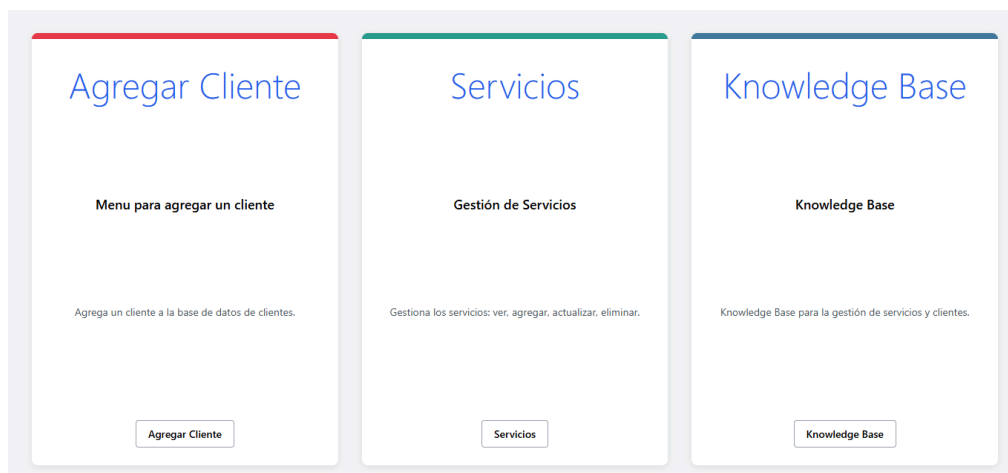
→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

y ahora nos iremos a la carpeta del backend para poder correr el servidor local donde la api se conectara con el frontend (Se debe de tomar en cuenta que ya se descargaron las librerías correspondientes)

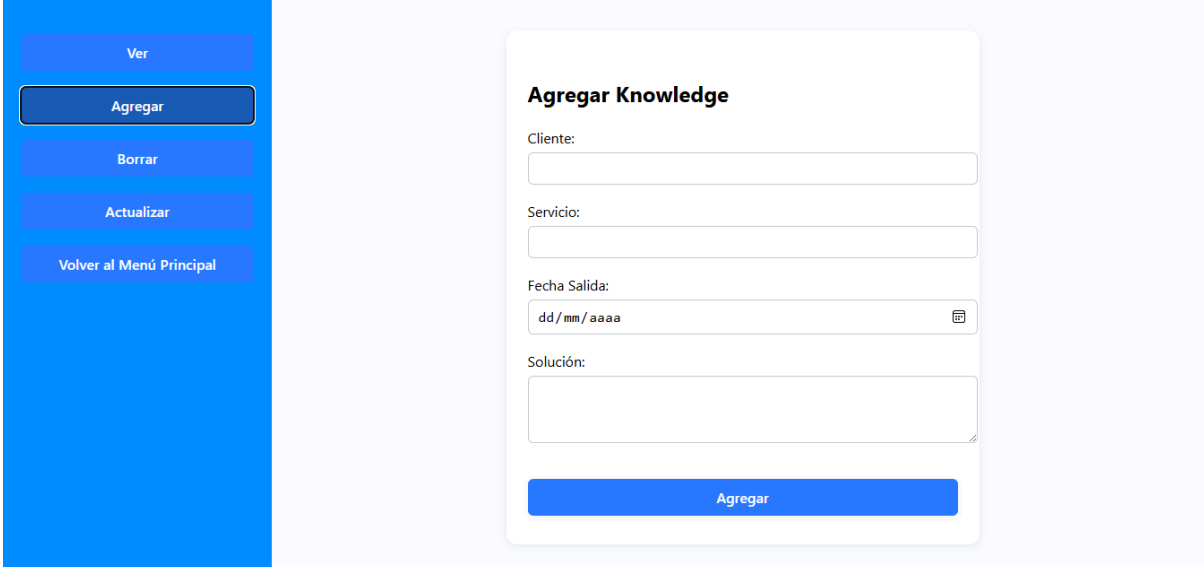
```
PS C:\Users\lnsqn\Desktop\ProyectoFinalDB> cd .\server-flask\
PS C:\Users\lnsqn\Desktop\ProyectoFinalDB\server-flask> .\venv\Scripts\activate
(venv) PS C:\Users\lnsqn\Desktop\ProyectoFinalDB\server-flask> python .\main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 264-626-550
```

Guia de uso

La aplicación se usará de la siguiente manera, es una manera verdaderamente intuitiva y a expensas de actualizaciones y mantenimiento estas son las opciones que se están abordando y el cómo se usarán



Como se muestra ahí tenemos un menú en donde tenemos que queremos hacer si administrar/gestionar la información respectiva de nuestros clientes, servicios o la knowledge base que es donde tenemos toda la información sobre las soluciones que hemos encontrado a las problemáticas de los servicios, veremos por ejemplo cómo manipular la knowledge base:



The image shows a web application interface. On the left is a blue sidebar menu with five buttons: 'Ver', 'Agregar', 'Borrar', 'Actualizar', and 'Volver al Menú Principal'. The 'Agregar' button is highlighted. To the right of the sidebar is a light blue background containing a white form titled 'Agregar Knowledge'. The form has four input fields: 'Cliente:', 'Servicio:', 'Fecha Salida:' (with a date picker icon and placeholder 'dd/mm/aaaa'), and 'Solución:'. At the bottom of the form is a blue button labeled 'Agregar'.

como se ve de manera intuitiva aquí están las opciones para la knowledge base, ver la tabla completa, agregar, actualizar y borrar.

El funcionamiento es sencillo se llenan los campos con la información que se recopiló a la hora de hacer el servicio, y esta se valida de manera automática