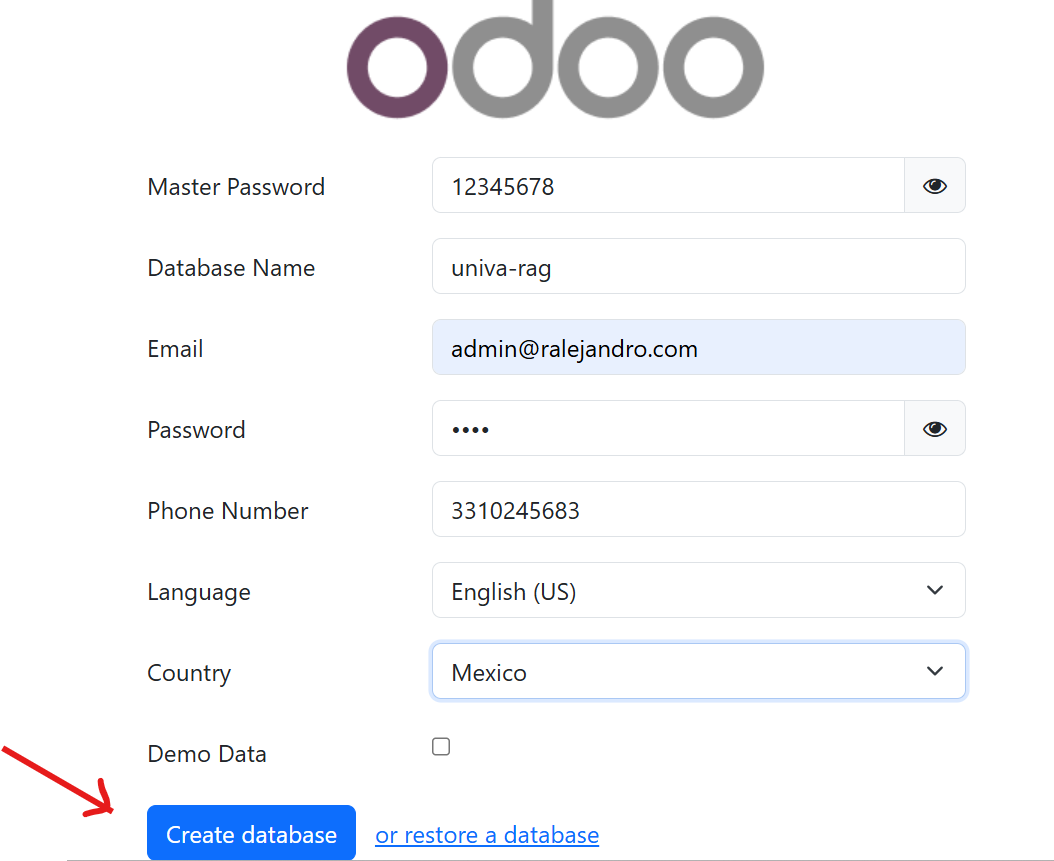
Pasos a seguir para instalar sistema RAG:

1. Explicación inicial con diapositivas de que es lo que se realizará.
2. Instalar WSL en Windows (si se tiene Ubuntu no es necesario).
3. Instalar el sistema Docker Desktop en windos o Docker en Linux.
4. Clonar el repositorio dentro de las rutas del Ubuntu WSL en Windows o en cualquier directorio en linux.
5. Instalar el comando “make” con “sudo apt-get install make”.
6. Explicar que es el comando make y para qué sirve.
7. Explicar la carpeta odoo.
8. Explicar cual es el funcionamiento del archivo doker-compose.yml y Dokerfile.
9. Duplicar el archivo .env.example de la carpeta setup y renombrarlo a .env
10. Duplicar el archivo example.odoo.conf en la carpeta setup/config y renombrarlo a odoo.conf.
11. Darle permisos a la carpeta setup “sudo chmod -R 777 setup”
12. Ejecutar Docker y crear una red de Docker necesaria expresada en el Docker-compose con el comando “docker network créate web”
13. Ejecutar Docker y Correr el comando “make dev” (Si estamos en Linux asegurarnos de que el proceso de Docker este corriendo).
14. Explicar cuántos Docker están corriendo y cuál es su funcionamiento. Se pueden ver los dockers desde Docker Desktop o con el comando Docker ps.
15. Abrir odoo en la ruta <http://localhost:8069/>, en la pantalla que sale para crear o reestablecer una base de datos la contraseña maestra es la contraseña “12345678” de administrador que se encuentra en el archivo odoo.conf como admin password, el nombre de la base de datos “rag-univa” y el password el que se decida.

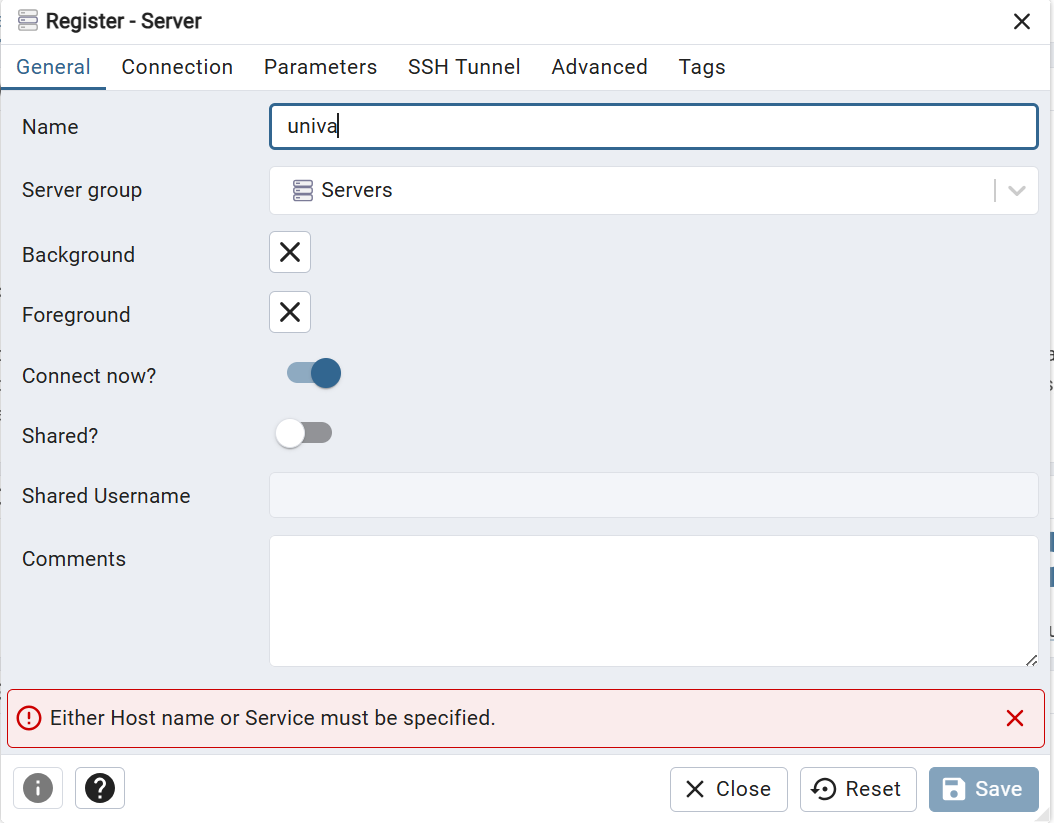
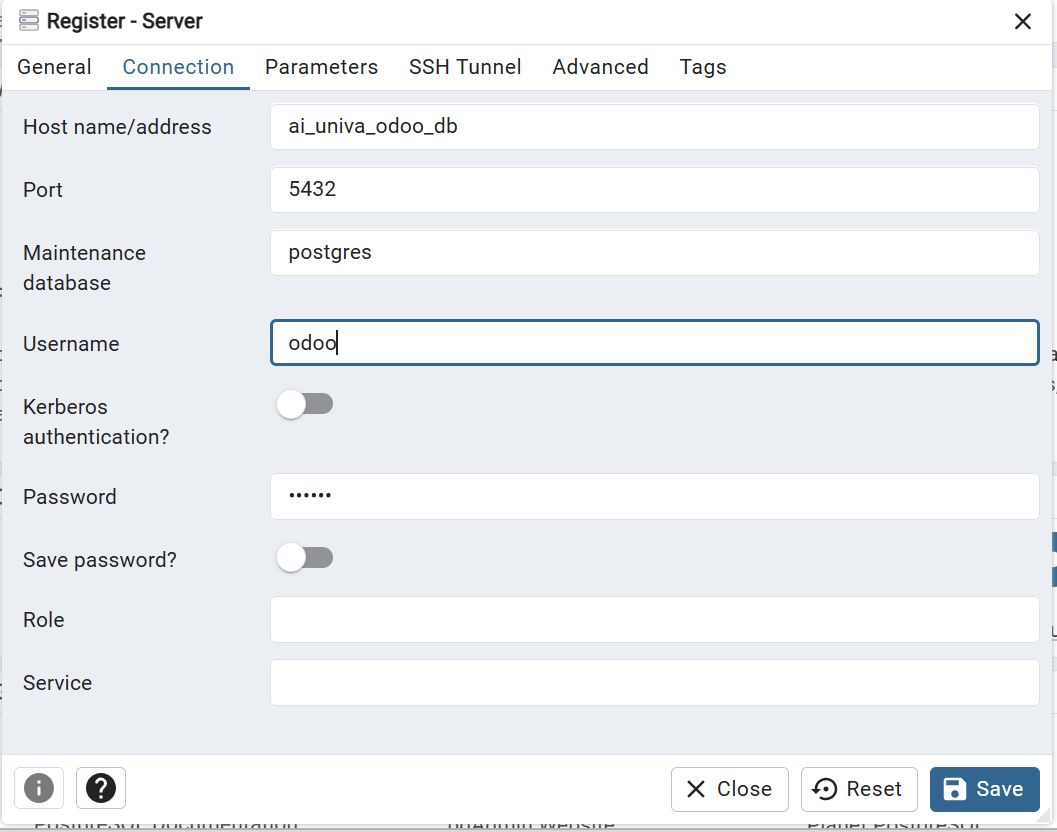


1. Explicar las carpetas de la carpeta rag como pluguins de odoo.
2. Copiar las carpetas internas de la carpeta rag en la la ubicación custom-addons cuando el contenedor no esté en ejecución.
3. Agregar la línea del api key de openIA al .env en la parte de # AI Models. OPENAI\_API\_KEY=sk-proj-GBGEiHLYQRr14e26HQxg\_RgdinPqbE214Io9KJ7uv1ZFl4GJ2yiFduco-7v7JWJUlfdYP3hOK4T3BlbkFJeknH\_xmlLw02TpXxTJe-5wPIOy410fnUHyEp0W3FVF\_0Vem2uvhMeXrtgrMfEzDKUkvxfWCnUA
4. Ejecutar nuevamente los contenedores con el comando make dev.
5. En odoo habilitar la opción de desarrollador en ajustes.
6. En odoo en apps actualizar aplicaciones disponibles.
7. En odoo en apps instalar los addons que agregamos (buscar por univa quitando el filtro apps).
8. En odoo agregar vectorizados, partes interesadas y agregar un par de avisos. Si al momento de agregar avisos mando un error verificar de que trata, Si en un error de memoria o un error de sentence transformer agregar las siguiente líneas al odoo.conf :

**limit\_memory\_hard = 8589934592**

**limit\_memory\_soft = 6442450944**

**limit\_request = 8192**

1. Configurar pgadmin está en <http://localhost:5050>
   1. Agregar un server con las siguientes especificaciones (mismas que están en el archivo .env).  
2. Ver los registros de embeddings en la base de datos en la tablas de pgVector “select \* from langchain\_pg\_embedding”, “select \* from res\_users”.
3. Revisar el “prompt engineering” preguntar a los alumnos donde esta la indicación que se le da a OpenIA.
4. Lanzar petición tipo post por medio de postman para probar el chatbot, la petición se hace a la ruta <http://localhost:8069/api/v1/whatsapp/answers> con un json como content :

{

  "from\_phone": "3310262600",

  "message": "dime cuando se fundó la UNIVA"

}



1. Explicación de la temperatura en la petición a openAI.
2. Explicación final de cómo funciona una red Neuronal y un LLM.