¡Claro, John! Aquí tienes el **paso a paso final y completo**, con **todas las correcciones actualizadas** que incluyen:

✅ App Flask con Bootstrap  
✅ Conexión a MySQL mediante Docker  
✅ Variables de entorno seguras  
✅ Docker Compose  
✅ Servida por Nginx con SSL (Let's Encrypt)  
✅ Listo para producción en https://ventas.incolpan.com

**🧱 PASO A PASO FINAL (ACTUALIZADO)**

**1️⃣ ACCEDER Y CREAR USUARIO**

ssh root@IP\_DEL\_VPS

**Crear usuario nuevo:**

adduser john

usermod -aG sudo john

exit

Volver a entrar como el nuevo usuario:

ssh john@IP\_DEL\_VPS

**2️⃣ INSTALAR PAQUETES BÁSICOS**

sudo apt update && sudo apt upgrade -y

sudo apt install git curl ufw nginx software-properties-common ca-certificates lsb-release gnupg -y

**3️⃣ INSTALAR DOCKER Y COMPOSE**

curl -fsSL https://get.docker.com | sudo bash

sudo usermod -aG docker $USER

newgrp docker

sudo apt install docker-compose-plugin -y

**4️⃣ (Opcional) INSTALAR COCKPIT**

sudo apt install cockpit -y

sudo systemctl enable --now cockpit.socket

**5️⃣ CONFIGURAR FIREWALL**

sudo ufw allow OpenSSH

sudo ufw allow 80,443,3306,9090/tcp

sudo ufw enable

**6️⃣ CLONAR TU APLICACIÓN**

cd /srv/

git clone https://github.com/JohnFRave90/sistema\_ventas.git

cd sistema\_ventas

**7️⃣ CREAR .env**

📄 Ruta: /srv/sistema\_ventas/.env

FLASK\_DEBUG=1

SECRET\_KEY=Incolpan12624+

DB\_USER=incolpan

DB\_PASSWORD=Incolpan12624+

DB\_HOST=mysql

DB\_NAME=sistema\_ventas

DB\_PORT=3306

**8️⃣ AÑADIR .env AL .gitignore**

📄 Añade esta línea a tu .gitignore:

.env

**9️⃣ CREAR Dockerfile**

📄 /srv/sistema\_ventas/Dockerfile

FROM python:3.11-slim

WORKDIR /app

COPY . /app

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 5000

CMD ["python", "run.py"]

**🔟 CREAR docker-compose.yml**

📄 /srv/sistema\_ventas/docker-compose.yml

version: '3.8'

services:

mysql:

image: mysql:8

container\_name: sistema\_mysql

restart: always

environment:

MYSQL\_DATABASE: sistema\_ventas

MYSQL\_USER: incolpan

MYSQL\_PASSWORD: Incolpan12624+

MYSQL\_ROOT\_PASSWORD: Incolpan12624+

volumes:

- mysql\_data:/var/lib/mysql

ports:

- "3306:3306"

web:

build: .

container\_name: sistema\_web

ports:

- "5000:5000"

volumes:

- .:/app

env\_file:

- .env

depends\_on:

- mysql

restart: always

phpmyadmin:

image: phpmyadmin/phpmyadmin

container\_name: phpmyadmin

restart: always

ports:

- "8080:80"

environment:

PMA\_HOST: mysql

PMA\_PORT: 3306

MYSQL\_ROOT\_PASSWORD: Incolpan12624+

depends\_on:

- mysql

volumes:

mysql\_data:

**1️⃣1️⃣ ACTUALIZA LA CONEXIÓN EN FLASK**

En \_\_init\_\_.py o config.py, usa las variables de entorno:

import os

app.config['SQLALCHEMY\_DATABASE\_URI'] = (

f"mysql+pymysql://{os.getenv('DB\_USER')}:{os.getenv('DB\_PASSWORD')}@{os.getenv('DB\_HOST')}:{os.getenv('DB\_PORT')}/{os.getenv('DB\_NAME')}"

)

Y asegúrate de tener en requirements.txt:

Flask

SQLAlchemy

pymysql

**1️⃣2️⃣ CONSTRUIR Y LEVANTAR APP**

docker compose up -d --build

Verifica:

docker ps

docker logs sistema\_web

**1️⃣3️⃣ CONFIGURAR NGINX**

sudo nano /etc/nginx/sites-available/ventas

Contenido:

server {

listen 80;

server\_name ventas.incolpan.com;

location / {

proxy\_pass http://127.0.0.1:5000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

Activar y recargar:

sudo ln -s /etc/nginx/sites-available/ventas /etc/nginx/sites-enabled/

sudo nginx -t

sudo systemctl reload nginx

**1️⃣4️⃣ SSL CON CERTBOT**

sudo apt install certbot python3-certbot-nginx -y

sudo certbot --nginx -d ventas.incolpan.com

Selecciona **redirigir HTTP a HTTPS**.

**✅ 1️⃣5️⃣ LISTO**

Abre en tu navegador:

https://ventas.incolpan.com

Tu app Flask + MySQL debe funcionar con Docker, Nginx, y SSL en producción.

¿Quieres agregar phpMyAdmin, backups automáticos o monitoreo con Uptime Kuma o Portainer? Puedo ayudarte también con eso.

Perfecto, John. Si ya tienes un archivo llamado crear\_tablas.py que define y crea las tablas de la base de datos usando SQLAlchemy, puedes ejecutarlo fácilmente **dentro del contenedor web** donde corre Flask.

**✅ PASOS PARA EJECUTAR crear\_tablas.py EN DOCKER**

**1️⃣ Accede al contenedor de la app Flask**

Desde tu servidor:

docker exec -it sistema\_web bash

Esto te deja dentro del contenedor con una terminal tipo Linux.

**2️⃣ Ejecuta el script de creación de tablas**

Asegúrate de estar en el directorio correcto (/app por defecto):

ls

Si ves crear\_tablas.py, ejecútalo:

python crear\_tablas.py

✅ Si todo está bien configurado (y el script conecta usando variables del .env), las tablas serán creadas dentro de MySQL automáticamente.

**3️⃣ Salir del contenedor**

exit

**🧪 OPCIONAL: Verifica que se hayan creado las tablas**

Puedes usar un cliente MySQL o instalar phpMyAdmin, o simplemente usar:

docker exec -it sistema\_mysql mysql -u incolpan -p

Y luego:

USE sistema\_ventas;

SHOW TABLES;

¿Quieres que el script crear\_tablas.py se ejecute automáticamente si la base de datos está vacía? También te puedo ayudar a integrarlo al flujo del contenedor.

Perfecto, si ya hiciste el cambio en GitHub, puedes actualizar tu aplicación en el VPS con estos pasos:

**🔁 PASO A PASO PARA ACTUALIZAR CÓDIGO DESDE GITHUB**

**1️⃣ Ve al directorio del proyecto en tu VPS**

cd /srv/sistema\_ventas

**2️⃣ Descargar los últimos cambios del repositorio**

git pull origin main

✅ Asegúrate de que la rama se llama main. Si es master u otra, usa ese nombre:

git pull origin master

**3️⃣ Reconstruir los contenedores para aplicar los cambios**

docker compose down

docker compose up -d --build

Esto:

* Apaga los contenedores
* Reconstruye la imagen con el nuevo run.py
* Levanta el servicio actualizado

**4️⃣ Verificar que Flask esté respondiendo**

curl http://localhost:5000

O accede desde tu navegador:

https://ventas.incolpan.com

¿Quieres automatizar este paso en el futuro con GitHub Webhooks o un botón de "actualizar" en Cockpit? Te puedo ayudar con eso también.