Para desplegar tu proyecto en una Orange Pi y mantenerlo actualizado automáticamente cuando realices cambios en desarrollo desde tu equipo Windows, sigue estos pasos:

### 1. Despliegue inicial en Orange Pi

1. **Instala Python, pip y Node.js** en la Orange Pi (si no están instalados).
2. **Copia tu proyecto** a la Orange Pi (puedes usar scp, rsync, o subirlo a un repositorio Git y clonarlo).
3. **Instala las dependencias**:

pip install flask flask-cors apscheduler pygame

cd scheduler-app

npm install

npm run build

1. **Configura el backend** para servir los archivos estáticos del frontend (puedes usar Flask para esto o un servidor como Nginx).
2. **Ejecuta el backend** (por ejemplo, con python3 schedule-controller.py).

### 2. Sincronización automática de cambios

La forma más robusta y profesional es usar **Git** y un pequeño script de actualización automática:

#### Opción A: Pull automático con Git y systemd

1. **Sube tu proyecto a un repositorio Git** (GitHub, GitLab, etc.).
2. **Clona el repositorio** en la Orange Pi.
3. **Crea un servicio systemd** para tu backend (para que siempre esté corriendo y se reinicie si hay cambios).
4. **Crea un script de actualización automática** que haga git pull y reinicie el servicio si hay cambios.

Ejemplo de script (actualizar\_y\_reiniciar.sh):

#!/bin/bash

cd /ruta/a/tu/proyecto

git pull

cd scheduler-app

npm install

npm run build

cd ..

sudo systemctl restart tu-servicio-backend

Puedes programar este script con cron para que se ejecute cada cierto tiempo, o ejecutarlo manualmente tras cada push.

#### Opción B: Sincronización con rsync

Si no usas Git, puedes usar rsync desde tu PC para sincronizar los archivos:

rsync -avz --delete /ruta/local/usuario/proyecto/ usuario@ip\_orangepi:/ruta/remota/proyecto/

Luego reinicia el backend en la Orange Pi.

### 3. Auto-actualización tras cambios

* Si usas **systemd** para correr el backend, puedes reiniciarlo automáticamente tras cada actualización.
* Si usas **pm2** (para Node.js), también puedes reiniciar el proceso tras cada build.

### 4. Consejos extra

* Usa .env o variables de entorno para separar configuraciones de desarrollo y producción.
* Haz backup regular de la base de datos [alarmas.db](vscode-file://vscode-app/c:/Users/ferna/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") si es importante.

¿Quieres un ejemplo de archivo de servicio systemd para tu backend Flask? ¿O un script de actualización listo para copiar?