

GUÍA DE INSTALACIÓN DETALLADA AEROSYS:AEROPUERTOS

KASSIMCITO

GUÍA DE INSTALACIÓN DETALLADA - SISTEMA DE GESTIÓN DE AEROPUERTOS

TABLA DE CONTENIDOS

1. **Introducción**
2. **Requisitos del Sistema**
3. **Instalación en Windows**
4. **Instalación en Linux**
5. **Instalación en macOS**
6. **Configuración de Base de Datos**
7. **Configuración de Producción**
8. **Solución de Problemas**
9. **Mantenimiento**

INTRODUCCIÓN

Esta guía proporciona instrucciones detalladas para instalar y configurar el Sistema de Gestión de Aeropuertos en diferentes sistemas operativos y entornos.

REQUISITOS DEL SISTEMA

REQUISITOS MÍNIMOS

- **Sistema Operativo:** Windows 10, Ubuntu 18.04+, macOS 10.14+
- **Python:** 3.8 o superior
- **RAM:** 4GB mínimo, 8GB recomendado
- **Espacio en Disco:** 2GB libres
- **Navegador:** Chrome 80+, Firefox 75+, Safari 13+, Edge 80+

REQUISITOS RECOMENDADOS

- **Sistema Operativo:** Windows 11, Ubuntu 20.04+, macOS 12+
- **Python:** 3.9 o superior
- **RAM:** 8GB o más
- **Espacio en Disco:** 5GB libres
- **Procesador:** 4 núcleos o más

INSTALACIÓN EN WINDOWS

PASO 1: INSTALAR PYTHON

1. Descarga Python desde python.org
2. Ejecuta el instalador
3. **IMPORTANTE:** Marca “Add Python to PATH”
4. Selecciona “Install Now”
5. Verifica la instalación:

```
python --version  
pip --version
```

PASO 2: CLONAR EL REPOSITORIO

1. Abre PowerShell o CMD como administrador
2. Navega al directorio deseado
3. Clona el repositorio:

```
git clone https://github.com/KassimCITO/aeroSys.git  
cd aeroSys
```

PASO 3: INSTALACIÓN AUTOMÁTICA

1. Ejecuta el script de instalación:
2. Sigue las instrucciones en pantalla
3. El script creará automáticamente:

```
python install.py
```

- Entorno virtual
- Instalación de dependencias
- Base de datos inicial
- Usuario administrador

PASO 4: INSTALACIÓN MANUAL (ALTERNATIVA)

1. Crea el entorno virtual:

```
python -m venv venv
```

2. Activa el entorno virtual:

```
venv\Scripts\activate
```

3. Instala las dependencias:

```
pip install -r requirements.txt
```

4. Configura la base de datos:

```
python -m flask db upgrade
```

5. Pobra la base de datos:

```
python seed.py
```

PASO 5: EJECUTAR LA APLICACIÓN

1. Activa el entorno virtual:

```
venv\Scripts\activate
```

2. Ejecuta la aplicación:

```
python app.py
```

3. Abre el navegador en <http://localhost:5000>

INSTALACIÓN EN LINUX

PASO 1: ACTUALIZAR EL SISTEMA

```
sudo apt update  
sudo apt upgrade -y
```

PASO 2: INSTALAR PYTHON Y DEPENDENCIAS

```
sudo apt install python3 python3-pip python3-venv git -y
```

PASO 3: CLONAR EL REPOSITORIO

```
git clone https://github.com/KassimCITO/aeroSys.git  
cd aeroSys
```

PASO 4: INSTALACIÓN AUTOMÁTICA

```
python3 install.py
```

PASO 5: INSTALACIÓN MANUAL (ALTERNATIVA)

1. Crear entorno virtual:

```
python3 -m venv venv
```

2. Activar entorno virtual:

```
source venv/bin/activate
```

3. Instalar dependencias:

```
pip install -r requirements.txt
```

4. Configurar base de datos:

```
python -m flask db upgrade
```

5. Poblar base de datos:

```
python seed.py
```

PASO 6: EJECUTAR LA APLICACIÓN

```
source venv/bin/activate  
python app.py
```

INSTALACIÓN EN MACOS

PASO 1: INSTALAR HOMEBREW (SI NO ESTÁ INSTALADO)

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

PASO 2: INSTALAR PYTHON

```
brew install python3
```

PASO 3: CLONAR EL REPOSITORIO

```
git clone https://github.com/KassimCITO/aeroSys.git  
cd aeroSys
```

PASO 4: INSTALACIÓN AUTOMÁTICA

```
python3 install.py
```

PASO 5: INSTALACIÓN MANUAL (ALTERNATIVA)

1. Crear entorno virtual:

```
python3 -m venv venv
```

2. Activar entorno virtual:

```
source venv/bin/activate
```

3. Instalar dependencias:

```
pip install -r requirements.txt
```

4. Configurar base de datos:

```
python -m flask db upgrade
```

5. Poblar base de datos:

```
python seed.py
```

PASO 6: EJECUTAR LA APLICACIÓN

```
source venv/bin/activate
python app.py
```

CONFIGURACIÓN DE BASE DE DATOS

SQLITE (DESARROLLO)

La aplicación usa SQLite por defecto para desarrollo. No requiere configuración adicional.

MYSQL (PRODUCCIÓN)

1. Instala MySQL:

```
# Ubuntu/Debian
sudo apt install mysql-server

# CentOS/RHEL
sudo yum install mysql-server

# macOS
brew install mysql
```

2. Crea la base de datos:

```
CREATE DATABASE aeropuertos;
CREATE USER 'aerosys'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON aeropuertos.* TO 'aerosys'@'localhost';
FLUSH PRIVILEGES;
```

3. Configura la aplicación:

```
cp config.env.example .env
# Edita .env con la configuración de MySQL
```

POSTGRESQL (PRODUCCIÓN)

1. Instala PostgreSQL:

```
# Ubuntu/Debian
sudo apt install postgresql postgresql-contrib

# CentOS/RHEL
sudo yum install postgresql postgresql-server

# macOS
brew install postgresql
```

2. Crea la base de datos:

```
CREATE DATABASE aeropuertos;
CREATE USER aerosys WITH PASSWORD 'password';
GRANT ALL PRIVILEGES ON DATABASE aeropuertos TO aerosys;
```

3. Configura la aplicación:

```
cp config.env.example .env
# Edita .env con la configuración de PostgreSQL
```

CONFIGURACIÓN DE PRODUCCIÓN

USANDO GUNICORN

1. Instala Gunicorn:

```
pip install gunicorn
```

2. Crea un archivo `wsgi.py`:

```
from app import app

if __name__ == "__main__":
    app.run()
```

3. Ejecuta con Gunicorn:

```
gunicorn -w 4 -b 0.0.0.0:8000 wsgi:app
```

USANDO NGINX COMO PROXY REVERSO

1. Instala Nginx:

```
# Ubuntu/Debian
sudo apt install nginx

# CentOS/RHEL
sudo yum install nginx
```

2. Configura Nginx:

```
server {
    listen 80;
    server_name tu-dominio.com;

    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

3. Reinicia Nginx:

```
sudo systemctl restart nginx
```

CONFIGURACIÓN DE HTTPS

1. Instala Certbot:

```
sudo apt install certbot python3-certbot-nginx
```

2. Obtén certificado SSL:

```
sudo certbot --nginx -d tu-dominio.com
```

CONFIGURACIÓN DE SERVICIO SYSTEMD

1. Crea el archivo de servicio:

```
sudo nano /etc/systemd/system/aerosys.service
```

2. Contenido del archivo:

```
[Unit]
Description=AeroSys Web Application
After=network.target
```

```
[Service]
User=www-data
Group=www-data
WorkingDirectory=/path/to/aeroSys
Environment="PATH=/path/to/aeroSys/venv/bin"
ExecStart=/path/to/aeroSys/venv/bin/gunicorn --workers 3 --bind unix:/path/to/aeroSys/aerosys.sock -m 007 wsgi:app
ExecReload=/bin/kill -s HUP $MAINPID
Restart=always

[Install]
WantedBy=multi-user.target
```

3. Habilita y inicia el servicio:

```
sudo systemctl enable aerosys
sudo systemctl start aerosys
```

SOLUCIÓN DE PROBLEMAS

ERROR: "PYTHON NO ENCONTRADO"

- Verifica que Python esté instalado y en el PATH
- En Windows, reinstala Python marcando "Add to PATH"

ERROR: "PIP NO ENCONTRADO"

- Instala pip:

```
python -m ensurepip --upgrade
```

ERROR: "MÓDULO NO ENCONTRADO"

- Verifica que el entorno virtual esté activado
- Reinstala las dependencias:

```
pip install -r requirements.txt
```

ERROR: "BASE DE DATOS NO ENCONTRADA"

- Ejecuta las migraciones:

```
python -m flask db upgrade
```

ERROR: "PUERTO EN USO"

- Cambia el puerto en **app.py**:

```
app.run(host='0.0.0.0', port=5001, debug=True)
```

ERROR: “PERMISOS DENEGADOS”

- En Linux/macOS, usa sudo si es necesario
- Verifica permisos del directorio:

```
chmod -R 755 /path/to/aeroSys
```

MANTENIMIENTO

BACKUP DE BASE DE DATOS

1. **SQLite:**

```
cp aeropuertos.db backup_aeropuertos_$(date +%Y%m%d).db
```

2. **MySQL:**

```
mysqldump -u aerosys -p aeropuertos > backup_aeropuertos_$(date +%Y%m%d).sql
```

3. **PostgreSQL:**

```
pg_dump -U aerosys aeropuertos > backup_aeropuertos_$(date +%Y%m%d).sql
```

ACTUALIZACIÓN DEL SISTEMA

1. Haz backup de la base de datos
2. Descarga la nueva versión:

```
git pull origin main
```

3. Actualiza las dependencias:

```
pip install -r requirements.txt
```

4. Ejecuta las migraciones:

```
python -m flask db upgrade
```

MONITOREO

- Revisa los logs en **app.log**
- Monitorea el uso de CPU y memoria

- Verifica el espacio en disco regularmente

LIMPIEZA

- Limpia archivos temporales regularmente
 - Elimina logs antiguos
 - Optimiza la base de datos periódicamente
-

Versión de la Guía: 1.0

Última Actualización: Octubre 2025

Sistema: Sistema de Gestión de Aeropuertos v1.0