Table of Contents

Guía de Instalación Detallada - Sistema de Gestión de Aeropuertos

Tabla de Contenidos

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:

```
python install.py
```

2. Sigue las instrucciones en pantalla
3. El script creará automáticamente:
    - 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. Pobla 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/ins
tall/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:

```sql
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:

```sql
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`:

```python
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 uni
x:/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