Manual del usuario aeroSys:Aeropuertos

KassimCITO

# Manual de Usuario - Sistema de Gestión de Aeropuertos

## Tabla de Contenidos

1. [Introducción](#introducción)
2. [Instalación y Configuración](#instalación-y-configuración)
3. [Acceso al Sistema](#acceso-al-sistema)
4. [Dashboard Principal](#dashboard-principal)
5. [Gestión de Aeronaves](#gestión-de-aeronaves)
6. [Gestión de Pilotos](#gestión-de-pilotos)
7. [Gestión de Vuelos](#gestión-de-vuelos)
8. [Gestión de Confirmaciones](#gestión-de-confirmaciones)
9. [Gestión de Usuarios](#gestión-de-usuarios)
10. [Reportes](#reportes)
11. [Configuración del Sistema](#configuración-del-sistema)
12. [Solución de Problemas](#solución-de-problemas)

## Introducción

El Sistema de Gestión de Aeropuertos es una aplicación web desarrollada en Flask que permite gestionar de manera integral las operaciones de un aeropuerto, incluyendo la administración de aeronaves, pilotos, vuelos, confirmaciones y usuarios.

### Características Principales

* **Dashboard intuitivo** con estadísticas en tiempo real
* **Gestión completa de aeronaves** con imágenes
* **Administración de pilotos** con información detallada
* **Control de vuelos** con asignación de aeronaves y pilotos
* **Sistema de confirmaciones** para seguimiento de vuelos
* **Gestión de usuarios** con diferentes roles
* **Reportes en PDF y Excel**
* **Interfaz responsive** compatible con dispositivos móviles

## Instalación y Configuración

### Requisitos del Sistema

* Python 3.8 o superior
* pip (gestor de paquetes de Python)
* Navegador web moderno

### Instalación Automática

1. Descarga el proyecto desde el repositorio
2. Ejecuta el script de instalación:

* python install.py

1. Sigue las instrucciones en pantalla
2. El sistema creará automáticamente:
   * Entorno virtual
   * Instalación de dependencias
   * Base de datos inicial
   * Usuario administrador

### Instalación Manual

1. Crea un entorno virtual:

* python -m venv venv

1. Activa el entorno virtual:
   * Windows: venv\Scripts\activate
   * Linux/Mac: source venv/bin/activate
2. Instala las dependencias:

* pip install -r requirements.txt

1. Configura la base de datos:

* python -m flask db upgrade

1. Pobla la base de datos con datos de ejemplo:

* python seed.py

1. Ejecuta la aplicación:

* python app.py

### Opciones de Ejecución

El sistema ofrece diferentes formas de ejecutar la aplicación según tus necesidades:

#### Opción 1: Ejecución Estándar (Desarrollo)

python app.py

* Servidor de desarrollo de Flask
* Recarga automática al detectar cambios
* Modo debug activado
* Acceso: http://localhost:5000

#### Opción 2: Ejecución con Flask CLI

flask run

* Servidor de desarrollo de Flask
* Configuración desde variables de entorno
* Acceso: http://localhost:5000

#### Opción 3: Ejecución en Modo Producción

# Usando Gunicorn (Linux/Mac)  
gunicorn -w 4 -b 0.0.0.0:5000 app:app  
  
# Usando Waitress (Windows)  
waitress-serve --host=0.0.0.0 --port=5000 app:app

* Servidor WSGI de producción
* Múltiples workers para mejor rendimiento
* Acceso desde red local: http://<tu-ip>:5000

#### Opción 4: Ejecución con Puerto Personalizado

# Modificar en app.py o usar variable de entorno  
export FLASK\_RUN\_PORT=8080  
flask run

* Cambia el puerto por defecto
* Útil para evitar conflictos de puertos

#### Opción 5: Ejecución en Red Local

python app.py --host=0.0.0.0

* Accesible desde otros dispositivos en la red
* Acceso: http://<ip-del-servidor>:5000

### Variables de Entorno Importantes

Configura estas variables antes de ejecutar:

# Windows  
set FLASK\_APP=app.py  
set FLASK\_ENV=development  
set SECRET\_KEY=tu-clave-secreta  
  
# Linux/Mac  
export FLASK\_APP=app.py  
export FLASK\_ENV=development  
export SECRET\_KEY=tu-clave-secreta

### Ejecución como Servicio (Producción)

#### Systemd (Linux)

Crea el archivo /etc/systemd/system/aerosys.service:

[Unit]  
Description=AeroSys - Sistema de Gestión de Aeropuertos  
After=network.target  
  
[Service]  
User=www-data  
WorkingDirectory=/ruta/a/aeroSys  
Environment="PATH=/ruta/a/aeroSys/venv/bin"  
ExecStart=/ruta/a/aeroSys/venv/bin/gunicorn -w 4 -b 0.0.0.0:5000 app:app  
  
[Install]  
WantedBy=multi-user.target

Comandos:

sudo systemctl start aerosys  
sudo systemctl enable aerosys  
sudo systemctl status aerosys

#### Windows Service (NSSM)

# Descargar NSSM y ejecutar:  
nssm install AeroSys "C:\ruta\a\venv\Scripts\python.exe" "C:\ruta\a\app.py"  
nssm start AeroSys

### Configuración de Nginx (Proxy Reverso)

Para producción con Nginx:

server {  
 listen 80;  
 server\_name tu-dominio.com;  
  
 location / {  
 proxy\_pass http://127.0.0.1:5000;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  
 }  
  
 location /static {  
 alias /ruta/a/aeroSys/static;  
 }  
}

## Acceso al Sistema

### Credenciales por Defecto

* **Usuario:** admin
* **Contraseña:** 123456

### Primer Acceso

1. Abre tu navegador web
2. Navega a http://localhost:5000
3. Ingresa las credenciales de administrador
4. Cambia la contraseña por defecto en la primera sesión

### Roles de Usuario

* **Administrador:** Acceso completo al sistema
* **Operador:** Gestión de vuelos y confirmaciones
* **Piloto:** Consulta de vuelos asignados
* **Invitado:** Acceso de solo lectura

## Dashboard Principal

El dashboard es la pantalla principal del sistema y muestra:

### Estadísticas Generales

* Total de aeronaves registradas
* Número de pilotos activos
* Vuelos programados
* Confirmaciones pendientes

### Accesos Rápidos

* Botones para acceder a cada módulo
* Enlaces a reportes más utilizados
* Acceso a configuración del sistema

### Información del Sistema

* Estado de la base de datos
* Última actualización
* Usuario conectado

## Gestión de Aeronaves

### Agregar Nueva Aeronave

1. Navega a “Aeronaves” en el menú principal
2. Haz clic en “Agregar Aeronave”
3. Completa el formulario:
   * **Matrícula:** Identificador único de la aeronave
   * **Modelo:** Modelo de la aeronave
   * **Fabricante:** Empresa fabricante
   * **Capacidad:** Número de pasajeros
   * **Tipo:** Comercial, privado, etc.
   * **Imagen:** Foto de la aeronave (opcional)
4. Haz clic en “Guardar”

### Editar Aeronave

1. En la lista de aeronaves, haz clic en el botón “Editar”
2. Modifica los campos necesarios
3. Haz clic en “Actualizar”

### Eliminar Aeronave

1. Haz clic en el botón “Eliminar”
2. Confirma la eliminación
3. La aeronave se eliminará del sistema

### Filtros y Búsqueda

* Filtra por fabricante
* Filtra por tipo de aeronave
* Busca por matrícula o modelo

## Gestión de Pilotos

### Agregar Nuevo Piloto

1. Navega a “Pilotos” en el menú principal
2. Haz clic en “Agregar Piloto”
3. Completa el formulario:
   * **Nombre:** Nombre completo del piloto
   * **Licencia:** Número de licencia
   * **Tipo de Licencia:** ATPL, CPL, PPL, etc.
   * **Horas de Vuelo:** Experiencia del piloto
   * **Nacionalidad:** País de origen
4. Haz clic en “Guardar”

### Editar Piloto

1. En la lista de pilotos, haz clic en “Editar”
2. Modifica la información necesaria
3. Haz clic en “Actualizar”

### Eliminar Piloto

1. Haz clic en el botón “Eliminar”
2. Confirma la eliminación

### Filtros

* Filtra por tipo de licencia
* Filtra por nacionalidad
* Busca por nombre o licencia

## Gestión de Vuelos

### Crear Nuevo Vuelo

1. Navega a “Vuelos” en el menú principal
2. Haz clic en “Agregar Vuelo”
3. Completa el formulario:
   * **Número de Vuelo:** Código único del vuelo
   * **Origen:** Aeropuerto de salida
   * **Destino:** Aeropuerto de llegada
   * **Fecha de Salida:** Fecha y hora de partida
   * **Fecha de Llegada:** Fecha y hora de llegada
   * **Aeronave:** Selecciona la aeronave
   * **Piloto:** Piloto al mando
   * **Copiloto:** Piloto secundario (opcional)
   * **Observaciones:** Notas adicionales
4. Haz clic en “Guardar”

### Editar Vuelo

1. En la lista de vuelos, haz clic en “Editar”
2. Modifica la información necesaria
3. Haz clic en “Actualizar”

### Confirmar Vuelo

1. Haz clic en el botón “Confirmar” junto al vuelo
2. Selecciona el estado:
   * **Confirmado:** Vuelo aprobado
   * **Pendiente:** Esperando confirmación
   * **Cancelado:** Vuelo cancelado
3. Agrega notas si es necesario
4. Haz clic en “Guardar Confirmación”

### Filtros

* Filtra por estado del vuelo
* Filtra por fecha
* Filtra por aeronave o piloto

## Gestión de Confirmaciones

### Ver Confirmaciones

1. Navega a “Confirmaciones” en el menú principal
2. Visualiza todas las confirmaciones de vuelos
3. Filtra por estado o vuelo específico

### Agregar Confirmación

1. Haz clic en “Agregar Confirmación”
2. Selecciona el vuelo
3. Establece el estado
4. Agrega notas si es necesario
5. Haz clic en “Guardar”

### Editar Confirmación

1. Haz clic en “Editar” junto a la confirmación
2. Modifica el estado o las notas
3. Haz clic en “Actualizar”

## Gestión de Usuarios

### Agregar Usuario

1. Navega a “Usuarios” (solo administradores)
2. Haz clic en “Agregar Usuario”
3. Completa el formulario:
   * **Nombre de Usuario:** Identificador único
   * **Contraseña:** Contraseña segura
   * **Rol:** Administrador, Operador, Piloto, Invitado
4. Haz clic en “Guardar”

### Editar Usuario

1. Haz clic en “Editar” junto al usuario
2. Modifica la información necesaria
3. Haz clic en “Actualizar”

### Eliminar Usuario

1. Haz clic en “Eliminar”
2. Confirma la eliminación

## Reportes

### Generar Reportes

1. Navega a “Reportes” en el menú principal
2. Selecciona el tipo de reporte:
   * **Aeronaves:** Lista de aeronaves con filtros
   * **Pilotos:** Información de pilotos
   * **Vuelos:** Detalles de vuelos
   * **Confirmaciones:** Estado de confirmaciones
   * **Dashboard:** Estadísticas generales

### Filtros de Reportes

* Aplica filtros según el tipo de reporte
* Selecciona formato: PDF o Excel
* Haz clic en “Generar Reporte”

### Exportar Reportes

* Los reportes se descargan automáticamente
* Formato PDF para impresión
* Formato Excel para análisis de datos

## Configuración del Sistema

### Configuración del Aeropuerto

1. Navega a “Configuración” en el menú principal
2. Completa la información del aeropuerto:
   * **Nombre:** Nombre oficial del aeropuerto
   * **Códigos:** IATA e ICAO
   * **Dirección:** Ubicación física
   * **Contacto:** Teléfono y email
3. Haz clic en “Guardar”

### Configuración de Usuario

1. Haz clic en tu nombre de usuario
2. Selecciona “Configuración”
3. Modifica tu información personal
4. Cambia tu contraseña si es necesario

## Solución de Problemas

### Problemas Comunes y Soluciones

#### 1. Error de Conexión a Base de Datos

**Síntoma:** OperationalError: no such table o database is locked

**Soluciones:**

# Verificar que la base de datos existe  
ls -la instance/  
  
# Recrear la base de datos  
python -m flask db upgrade  
  
# Si persiste, resetear completamente  
python reset\_db.py  
python seed.py

#### 2. Error de Módulos No Encontrados

**Síntoma:** ModuleNotFoundError: No module named 'flask'

**Soluciones:**

# Verificar que el entorno virtual está activado  
# Windows  
venv\Scripts\activate  
  
# Linux/Mac  
source venv/bin/activate  
  
# Reinstalar dependencias  
pip install -r requirements.txt

#### 3. Error de Puerto en Uso

**Síntoma:** OSError: [Errno 48] Address already in use

**Soluciones:**

# Encontrar el proceso usando el puerto 5000  
# Windows  
netstat -ano | findstr :5000  
taskkill /PID <PID> /F  
  
# Linux/Mac  
lsof -i :5000  
kill -9 <PID>  
  
# O usar otro puerto  
python app.py --port=8080

#### 4. Error de Permisos de Usuario

**Síntoma:** 403 Forbidden o acceso denegado

**Soluciones:** - Verificar que el usuario tenga el rol correcto - Cerrar sesión y volver a iniciar - Contactar al administrador para verificar permisos - Revisar la tabla usuarios en la base de datos

#### 5. Error de Archivos y Uploads

**Síntoma:** Error al subir imágenes de aeronaves

**Soluciones:**

# Crear directorio de uploads si no existe  
mkdir -p static/uploads  
  
# Verificar permisos (Linux/Mac)  
chmod 755 static/uploads  
  
# Windows: Verificar permisos de escritura en propiedades

#### 6. Error de Imágenes

**Síntoma:** Imágenes no se muestran o error al subir

**Soluciones:** - Tamaño máximo: 16MB - Formatos permitidos: JPG, JPEG, PNG, GIF - Verificar que la ruta static/uploads existe - Limpiar caché del navegador (Ctrl + F5)

#### 7. Error de Migraciones de Base de Datos

**Síntoma:** alembic.util.exc.CommandError

**Soluciones:**

# Verificar estado de migraciones  
python -m flask db current  
  
# Aplicar migraciones pendientes  
python -m flask db upgrade  
  
# Si hay conflictos, resetear  
rm -rf migrations/versions/\*.py  
python -m flask db stamp head  
python -m flask db migrate -m "reset"  
python -m flask db upgrade

#### 8. Error de Sesión Expirada

**Síntoma:** Redireccionamiento constante al login

**Soluciones:** - Limpiar cookies del navegador - Verificar SECRET\_KEY en configuración - Reiniciar el servidor - Usar modo incógnito para probar

#### 9. Error de Reportes PDF/Excel

**Síntoma:** Error al generar reportes

**Soluciones:**

# Verificar instalación de ReportLab  
pip install reportlab  
  
# Verificar instalación de openpyxl  
pip install openpyxl  
  
# Reinstalar todas las dependencias  
pip install -r requirements.txt --force-reinstall

#### 10. Error de Rendimiento Lento

**Síntoma:** La aplicación responde lentamente

**Soluciones:** - Verificar cantidad de registros en base de datos - Usar índices en tablas grandes - Limpiar datos antiguos - Aumentar workers en producción: bash gunicorn -w 8 -b 0.0.0.0:5000 app:app

#### 11. Error de CORS en API

**Síntoma:** Errores de CORS en consola del navegador

**Soluciones:**

# Agregar en app.py si es necesario  
from flask\_cors import CORS  
CORS(app)

#### 12. Error de Encoding/Caracteres Especiales

**Síntoma:** Caracteres especiales se muestran incorrectamente

**Soluciones:** - Verificar que la base de datos use UTF-8 - Asegurar que los archivos Python tengan encoding UTF-8 - Agregar al inicio de archivos Python: python # -\*- coding: utf-8 -\*-

### Comandos Útiles de Diagnóstico

# Verificar versión de Python  
python --version  
  
# Verificar paquetes instalados  
pip list  
  
# Verificar estado de la aplicación  
python -c "import app; print('OK')"  
  
# Verificar base de datos  
python -c "from app import db; print(db.engine.table\_names())"  
  
# Verificar configuración  
python -c "from config import Config; print(Config.SQLALCHEMY\_DATABASE\_URI)"

### Logs del Sistema

#### Ubicación de Logs

* **Aplicación:** app.log
* **Errores Flask:** Consola/terminal
* **Servidor:** Según configuración (Gunicorn, Nginx, etc.)

#### Revisar Logs

# Ver últimas líneas del log  
tail -f app.log  
  
# Buscar errores específicos  
grep -i "error" app.log  
  
# Ver logs de Nginx (si aplica)  
tail -f /var/log/nginx/error.log

#### Habilitar Debug Detallado

# En app.py  
app.config['DEBUG'] = True  
app.config['SQLALCHEMY\_ECHO'] = True # Ver queries SQL

### Mantenimiento Preventivo

#### Respaldo de Base de Datos

# SQLite  
cp instance/aeropuerto.db instance/aeropuerto\_backup\_$(date +%Y%m%d).db  
  
# Programar respaldos automáticos (cron Linux)  
0 2 \* \* \* cp /ruta/instance/aeropuerto.db /ruta/backups/aeropuerto\_$(date +\%Y\%m\%d).db

#### Limpieza de Archivos Temporales

# Limpiar archivos .pyc  
find . -type f -name "\*.pyc" -delete  
  
# Limpiar \_\_pycache\_\_  
find . -type d -name "\_\_pycache\_\_" -exec rm -rf {} +

#### Actualización del Sistema

# Actualizar dependencias  
pip install --upgrade -r requirements.txt  
  
# Aplicar nuevas migraciones  
python -m flask db upgrade

### Contacto de Soporte

* **Email:** KassimCITO@gmail.com
* **Teléfono:** +52 (443) 505.1882
* **Horario:** Lunes a Viernes, 8:00 AM - 6:00 PM

**Versión del Manual:** 1.0  
**Última Actualización:** Octubre 2025  
**Sistema:** Sistema de Gestión de Aeropuertos v1.0