

Documentación Técnica: Sistema de Gestión de Tráfico Aéreo



Daniel Zuñiga, Ivan Escandón, Ruben Parra, Alvaro Cano

Índice

1. Objetivo General.....	3
2. Arquitectura del Sistema.....	3
2.1. Estructuras de Datos Implementadas.....	3
2.2. Flujo de Datos Principal.....	3
3. Funcionalidades Detalladas.....	4
3.1. Alta de Vuelos (Punto 1).....	4
3.2. Gestión de Pistas (Punto 2).....	4
3.3. Reloj Simulado (Punto 3).....	5
3.4. Gestión de Fluxos (Punto 4).....	6
3.5. Asignación de Pistas (Punto 5).....	6
3.6. Sistema de Logs (Punto 6).....	7
3.7. Generación de Informes (Punto 7).....	7
3.8. Interfaz de Consola (Punto 8).....	8
3.9. Gestión de Casos Particulares (Punto 9).....	9
3.10. Persistencia y Recuperación.....	9
4. Ventajas del Diseño Implementado.....	10
4.1. Eficiencia Operativa.....	10
4.2. Escalabilidad.....	10
4.3. Fiabilidad.....	10
4.4. Usabilidad.....	10
5. Aplicaciones Prácticas.....	10
6. Limitaciones y Consideraciones.....	11
6.1. Supuestos del Modelo.....	11
6.2. Extensiones Futuras.....	11
7. Conclusión.....	11

1. Objetivo General

El sistema desarrollado es una aplicación de consola en Python que simula la gestión integral de operaciones aeroportuarias. Su propósito principal es modelar de manera realista el flujo de despegues y aterrizajes, optimizando la asignación de recursos (pistas) según criterios de seguridad y eficiencia operativa, manteniendo un registro histórico completo de todos los eventos mediante archivos de log.

2. Arquitectura del Sistema

2.1. Estructuras de Datos Implementadas

- ❖ El sistema utiliza las siguientes estructuras de datos fundamentales:
- ❖ Listas enlazadas: Para gestionar las colas de vuelos pendientes (aterrizaje y despegue)
- ❖ Diccionarios: Para almacenar y acceder rápidamente a la información de vuelos y pistas
- ❖ Colas prioritarias: Implementadas manualmente para ordenar los vuelos según la política de prioridad establecida
- ❖ Archivos de texto plano: Para persistencia de datos y registro de eventos

2.2. Flujo de Datos Principal

- ❖ Carga inicial de configuración desde archivos CSV
- ❖ Inicialización del reloj simulado y estructuras de control
- ❖ Procesamiento cíclico por minuto simulado
- ❖ Generación de informes y guardado de estado

3. Funcionalidades Detalladas

3.1. Alta de Vuelos (Punto 1)

- ❖ Propósito: Incorporar vuelos al sistema de simulación desde fuentes externas y manuales.
- ❖ Funcionalidades implementadas:
 - ❖ Carga automática inicial: Al iniciar el programa, se lee automáticamente el archivo vuelos.csv y se cargan todos los vuelos definidos
 - ❖ Estructura de almacenamiento: Los vuelos se almacenan en una estructura combinada que permite acceso rápido por ID y ordenamiento por prioridad
 - ❖ Actualización dinámica: Los estados de los vuelos se actualizan en tiempo real según avanza la simulación
 - ❖ Validación de datos: Verificación de formatos, IDs únicos y coherencia entre campos
 - ❖ Alta manual: Capacidad para agregar vuelos adicionales durante la ejecución mediante interfaz de consola
 - ❖ Campos gestionados por vuelo:
 - ID único del vuelo (clave primaria)
 - Tipo (ATERRIZAJE/DESPEGUE)
 - Tiempo estimado (ETA para aterrizajes, ETD para despegues)
 - Nivel de prioridad (0-2)
 - Minutos de combustible restante (solo aterrizajes)
 - Estado actual en el sistema

3.2. Gestión de Pistas (Punto 2)

- ❖ Propósito: Administrar los recursos físicos del aeropuerto (pistas) y su disponibilidad.
- ❖ Funcionalidades implementadas:

- Carga desde archivo: Lectura inicial de pistas.csv con toda la configuración
- Seguimiento de estado: Monitoreo constante de cada pista (libre, ocupada, deshabilitada)
- Asignación temporal: Registro de qué vuelo utiliza cada pista y hasta qué minuto
- Control de categorías: Diferenciación entre pistas cortas, estándar y largas
- Gestión de mantenimiento: Manejo de pistas deshabilitadas temporal o permanentemente
- Atributos por pista:
 - Identificador único
 - Categoría (determina tipos de vuelos compatibles)
 - Tiempo de uso estándar por operación
 - Estado de habilitación
 - Historial de uso

3.3. Reloj Simulado (Punto 3)

- ❖ Propósito: Sincronizar todas las operaciones del sistema en una línea temporal virtual.
- ❖ Funcionalidades implementadas:
 - Avance discreto: El tiempo avanza en minutos enteros simulados
 - Conversión temporal: Relación configurable entre tiempo real y simulado (por defecto 5 segundos reales = 1 minuto simulado)
 - Activación de rutinas: Por cada incremento del reloj se ejecutan automáticamente:
 - Actualización de combustible de vuelos en espera
 - Liberación de pistas cuyas operaciones han finalizado

- Revisión y asignación de nuevas operaciones
- Registro de eventos significativos

3.4. Gestión de Flujos (Punto 4)

- ❖ Propósito: Ordenar y priorizar el tráfico aéreo según criterios de seguridad y eficiencia.
- ❖ Funcionalidades implementadas:
 - Flujos paralelos: Separación independiente de colas para aterrizajes y despegues
 - Política de prioridad: Algoritmo de selección que considera:
 - Nivel de emergencia (prioridad 2)
 - Combustible crítico en aterrizajes
 - Tiempo de atraso respecto a hora prevista
 - Orden alfabético por ID como desempate
 - Detección automática de emergencias: Elevación a prioridad 2 cuando combustible ≤ 5 minutos
 - Balance operacional: Equilibrio entre aterrizajes y despegues según disponibilidad

3.5. Asignación de Pistas (Punto 5)

- ❖ Propósito: Optimizar la utilización de recursos asignando pistas a vuelos según disponibilidad y prioridad.
- ❖ Funcionalidades implementadas:
 - Detección automática: Identificación de pistas habilitadas y libres
 - Asignación inteligente: Selección del vuelo más prioritario para cada pista disponible

- Bloqueo temporal: Reserva de pista por el tiempo especificado en su configuración
- Liberación automática: Desbloqueo de pista al completarse el tiempo de uso
- Actualización de estados: Cambio automático de vuelo a estado COMPLETADO
- Reintentos: Reconsideración en siguiente minuto para pistas deshabilitadas

3.6. Sistema de Logs (Punto 6)

- ❖ Propósito: Mantener un registro histórico completo y recuperable de todas las operaciones.
- ❖ Funcionalidades implementadas:
 - Registro cronológico: Archivo eventos.log con timestamp de cada evento
 - Modo append: Preservación de historial entre ejecuciones
 - Eventos registrados:
 - Carga inicial del sistema
 - Cambios de estado de vuelos
 - Asignaciones y liberaciones de pistas
 - Emergencias detectadas
 - Finalización de operaciones
 - Cierre de simulación
 - Formato estandarizado: Estructura uniforme para fácil procesamiento posterior

3.7. Generación de Informes (Punto 7)

- ❖ Propósito: Proporcionar análisis estadísticos y resúmenes operativos.
- ❖ Funcionalidades implementadas:

- Informe final: Generación de informe.log con métricas clave
- Métricas calculadas:
- Tiempo total simulado
- Número de vuelos atendidos
- Tiempo medio de espera
- Utilización por pista
- Emergencias gestionadas
- Detalle cronológico por vuelo
- Salida dual: Generación simultánea a archivo y pantalla
- Formato legible: Presentación clara para análisis humano

3.8. Interfaz de Consola (Punto 8)

- ❖ Propósito: Proporcionar control interactivo sobre la simulación.
- ❖ Funcionalidades implementadas:
 - Menú principal: Navegación intuitiva por opciones numeradas
 - Operaciones disponibles:
 - Carga de archivos de datos
 - Adición manual de vuelos
 - Avance controlado del tiempo (1 o N minutos)
 - Visualización del estado actual del sistema
 - Generación de informe en cualquier momento
 - Guardado de estado y salida ordenada

- Validación de entrada: Control de errores en todas las interacciones
- Feedback inmediato: Confirmación de operaciones realizadas

3.9. Gestión de Casos Particulares (Punto 9)

- ❖ Propósito: Manejo robusto de situaciones excepcionales.
- ❖ Funcionalidades implementadas:
 - Resolución de empates: Criterio alfabético por ID de vuelo
 - Combustible crítico: Elevación automática a prioridad 2 y notificación
 - Pistas deshabilitadas: Reintentó automático en siguiente ciclo
 - Cierre temporal: Manejo de pistas deshabilitadas por N minutos
 - Cancelaciones: Eliminación ordenada de vuelos y registro en log
 - Robustez: Tolerancia a fallos en archivos de entrada

3.10. Persistencia y Recuperación

- ❖ Propósito: Mantener consistencia entre las ejecuciones del programa.
- ❖ Funcionalidades implementadas:
 - Guardado de estado: Exportación a estado.csv de la situación actual
 - Recuperación: Capacidad de reiniciar desde último estado guardado
 - Integridad: Validación de consistencia entre archivos relacionados
 - Backup implícito: Los logs sirven como trazabilidad completa

4. Ventajas del Diseño Implementado

4.1. Eficiencia Operativa

- ❖ Algoritmos para ordenamiento de prioridades
- ❖ Acceso a información frecuentemente consultada
- ❖ Memoria optimizada: Estructuras solo con datos necesarios

4.2. Escalabilidad

- ❖ Diseño modular: Facilita adición de nuevas funcionalidades
- ❖ Parámetros configurables: Tiempos, prioridades, umbrales
- ❖ Formatos extensibles: CSV admiten columnas adicionales

4.3. Fiabilidad

- ❖ Validación exhaustiva: De todos los datos de entrada
- ❖ Manejo de errores: Graceful degradation en fallos
- ❖ Trazabilidad completa: Capacidad de auditoría mediante logs

4.4. Usabilidad

- ❖ Interfaz intuitiva: Menús claros y opciones autoexplicativas
- ❖ Feedback constante: Información de progreso en pantalla
- ❖ Documentación embebida: Ayuda disponible dentro del programa

5. Aplicaciones Prácticas

- ❖ Este sistema simula escenarios reales que permiten:
 - Análisis de capacidad: Determinar límites operativos del aeropuerto

- Pruebas de políticas: Evaluar diferentes algoritmos de priorización
- Entrenamiento: Formación de controladores aéreos en entorno simulado
- Planificación: Optimización de recursos ante diferentes volúmenes de tráfico
- Investigación: Estudio de efectos de contingencias (pistas fuera de servicio, emergencias)

6. Limitaciones y Consideraciones

6.1. Supuestos del Modelo

- ❖ Tiempos de operación fijos por pista
- ❖ Prioridades discretas (0, 1, 2)
- ❖ Combustible decreciente linealmente
- ❖ Independencia entre operaciones

6.2. Extensiones Futuras

- ❖ Interfaz gráfica para visualización en tiempo real
- ❖ Simulación meteorológica y su impacto
- ❖ Tipos adicionales de emergencias
- ❖ Optimización multi-objetivo
- ❖ Integración con sistemas externos

7. Conclusión

El sistema desarrollado constituye una herramienta completa para la simulación y análisis de operaciones aeroportuarias, implementando todas las funcionalidades solicitadas con un diseño robusto, eficiente y extensible. La arquitectura modular y el uso de estructuras de datos apropiadas garantizan tanto el correcto funcionamiento actual como la capacidad de

evolución futura, mientras que el sistema completo de logs e informes proporciona la trazabilidad y capacidad de análisis requeridas para aplicaciones profesionales y educativas.