

**FlightOnTime**  
**Sistema de Predicción de Retrasos de Vuelos**  
**Manual Técnico y Documento de Referencia del Proyecto**

## **Capítulo 1. Introducción y propósito del proyecto**

FlightOnTime es un sistema de predicción de retrasos de vuelos diseñado para anticipar, con base en datos históricos y técnicas de Machine Learning, si un vuelo despegará de forma puntual o con retraso significativo. El proyecto nace de una necesidad concreta de la industria aérea: los retrasos no solo afectan la experiencia del pasajero, sino que generan ineficiencias operativas, sobrecostos y problemas de planificación que podrían mitigarse si el riesgo se conoce con antelación.

El objetivo principal del sistema no es explicar retrospectivamente por qué ocurrió un retraso, sino **estimar el riesgo de que ocurra antes del despegue**, cuando aún es posible tomar decisiones preventivas. Para lograrlo, FlightOnTime combina un modelo de Data Science robusto con una arquitectura pensada para su uso real en producción, exponiendo las predicciones mediante una API REST.

Este enfoque convierte al proyecto en una solución completa, que va más allá del análisis exploratorio o el modelado experimental, y se posiciona como una herramienta práctica de apoyo a la toma de decisiones.

## **Capítulo 2. Enfoque de Data Science y definición del problema**

Desde el punto de vista de Data Science, el problema se formuló como un **problema de clasificación binaria**, donde el modelo debe predecir si un vuelo pertenecerá a una de las siguientes clases:

- Vuelo puntual
- Vuelo con retraso igual o superior a 15 minutos

La variable objetivo utilizada es DEP\_DEL15, ampliamente empleada en datasets aeronáuticos, lo que asegura coherencia con estándares reales del sector. Esta definición permite que la predicción tenga un significado operativo claro y directamente interpretable por usuarios técnicos y no técnicos.

El desafío principal radica en la alta variabilidad del sistema aéreo: múltiples factores interactúan simultáneamente, desde condiciones climáticas hasta patrones operacionales específicos de aerolíneas, aeropuertos y franjas horarias. Por ello, el enfoque del proyecto privilegia modelos capaces de capturar relaciones complejas sin sacrificar robustez.

### **Capítulo 3. Datos utilizados y enriquecimiento del dataset**

El modelo fue entrenado utilizando un dataset de gran escala compuesto por aproximadamente **35,7 millones de vuelos domésticos de Estados Unidos**. Este volumen de datos permite capturar patrones estructurales del sistema aéreo que no serían visibles en datasets pequeños o acotados.

Además de la información básica de los vuelos, el proyecto incorpora un proceso de **enriquecimiento de datos**, integrando variables externas relevantes para la predicción de retrasos. En particular, se añadió información climática histórica y datos de geolocalización obtenidos mediante APIs especializadas y fuentes evaluadas por su confiabilidad.

Este enriquecimiento es una de las principales fortalezas del proyecto, ya que introduce factores que influyen directamente en la puntualidad de los vuelos y que suelen ser omitidos en aproximaciones simplificadas.

### **Capítulo 4. Variables y features consideradas por el modelo**

El conjunto final de features utilizadas por el modelo fue cuidadosamente seleccionado para cumplir dos criterios fundamentales: relevancia predictiva y disponibilidad previa al vuelo.

#### **Variables temporales**

Estas variables permiten capturar estacionalidad y patrones horarios:

- Año (YEAR)
- Mes (MONTH)
- Día de la semana (DAY\_OF\_WEEK)
- Minuto programado del día (sched\_minute\_of\_day)

#### **Variables operacionales**

Representan características propias del vuelo:

- Aerolínea (OP\_UNIQUE\_CARRIER)
- Aeropuerto de origen (ORIGIN)
- Aeropuerto de destino (DEST)
- Distancia del vuelo (DISTANCE, DIST\_MET\_KM)

#### **Variables climáticas**

Introducen el impacto del entorno meteorológico:

- Temperatura (TEMP)
- Velocidad del viento (WIND\_SPD)
- Precipitación horaria (PRECIP\_1H)

- Índice de severidad climática (CLIMATE\_SEVERITY\_IDX)

### **Variables descartadas**

Se eliminaron variables que podían generar fuga de información, no estaban disponibles en tiempo real o no aportaban valor predictivo claro, como:

- DEP\_DELAY
- FL\_DATE
- CRS\_DEP\_TIME
- Nombres de ciudades en lugar de códigos normalizados

Esta depuración refuerza el carácter realista y productivo del modelo.

## **Capítulo 5. Modelo de Machine Learning**

Tras evaluar distintos enfoques, el modelo final seleccionado fue **XGBoost**, un algoritmo de gradient boosting ampliamente utilizado en problemas de clasificación con datos tabulares.

La elección de XGBoost se fundamenta en su capacidad para:

- Capturar relaciones no lineales complejas
- Manejar interacciones entre múltiples variables
- Escalar eficientemente con grandes volúmenes de datos
- Mantener un excelente desempeño predictivo

El modelo fue entrenado, evaluado y optimizado, alcanzando las siguientes métricas:

- Accuracy: 72.32%
- Recall: 54.30%
- ROC-AUC: 0.7194
- Threshold optimizado: 0.5591

Estas métricas reflejan un equilibrio adecuado entre precisión global y capacidad para identificar vuelos con riesgo real de retraso, priorizando la detección temprana de los casos más críticos.

## **Capítulo 6. Arquitectura del sistema e integración en producción**

FlightOnTime fue diseñado con una arquitectura modular y desacoplada, separando claramente las responsabilidades de cada componente:

- Capa de Data Science: entrenamiento, evaluación y serialización del modelo
- Capa de Back-End: exposición del modelo mediante una API REST
- Capa de Front-End: consumo de la API por usuarios finales

El modelo fue serializado utilizando **Joblib** y cargado dinámicamente por la API desarrollada con **FastAPI**. La comunicación entre el Back-End y el modelo se basa en un **contrato de features definido previamente**, lo que asegura coherencia entre entrenamiento y predicción en producción.

Esta arquitectura permite que el modelo sea reutilizable, escalable y fácilmente integrable en distintos contextos.

## Capítulo 7. Funcionamiento del sistema y datos de entrada

Desde el punto de vista del usuario o sistema cliente, el funcionamiento del proyecto es sencillo y transparente.

Para solicitar una predicción, se deben proporcionar a la API los datos básicos del vuelo, entre ellos:

- Aerolínea
- Aeropuerto de origen y destino
- Fecha y hora programada de salida
- Información necesaria para reconstruir las variables temporales y espaciales

La API valida la información, aplica las transformaciones necesarias y ejecuta el modelo predictivo.

## Capítulo 8. Resultados entregados e interpretación

Como resultado, el sistema devuelve:

- Una clasificación: vuelo puntual o vuelo con retraso
- Una probabilidad asociada al retraso

Esta probabilidad no debe interpretarse como una certeza absoluta, sino como un **indicador de riesgo**. Valores altos indican que, dadas condiciones históricas similares, existe una probabilidad elevada de retraso. Esto permite tomar decisiones informadas, como reforzar la planificación operativa, anticipar comunicaciones o priorizar el monitoreo de determinados vuelos.

## Capítulo 9. Valor y relevancia del proyecto

FlightOnTime demuestra cómo la Data Science genera impacto real cuando se integra en sistemas utilizables. El proyecto combina rigor técnico, uso de datos reales, modelado robusto y una arquitectura pensada para producción.

No promete eliminar los retrasos, pero sí **reducir la incertidumbre**, que es uno de los mayores problemas operativos de la industria aérea. En ese sentido, el proyecto se posiciona como una solución necesaria, escalable y alineada con prácticas profesionales reales.