

# Informe EELISA

## Práctica final Big Data

### Índice

- [1\) Introducción](#)
- [2\) Descripción del reto a resolver](#)
- [3\) Arquitectura física del sistema](#)
- [4\) Arquitectura lógica del sistema](#)
- [5\) Decisiones de diseño](#)
- [6\) Flujo de datos en el sistema](#)
- [7\) Descripción de cambios respecto a la práctica de vuelos](#)
- [8\) \(EXTRA\) Ejemplo práctico del funcionamiento del sistema](#)
- [9\) Conclusiones](#)

### 1) Introducción

El intercambiador de Plaza Elíptica enfrenta problemas críticos relacionados con la afluencia masiva de personas durante los horarios pico. Estos problemas incluyen saturación de accesos, largas esperas y riesgos de seguridad, que afectan tanto la experiencia de los usuarios como la operación del intercambiador. Este informe propone un sistema basado en Big Data que utiliza datos históricos, datos en tiempo real provenientes de cámaras y sensores, y datos de otras fuentes para predecir la afluencia de público en momentos específicos. Con estas predicciones, los operadores podrán planificar y gestionar mejor los recursos, optimizando la comodidad y la seguridad del espacio.

### 2) Descripción del reto a resolver

El desafío principal consiste en predecir la afluencia de personas en diferentes momentos del día, considerando tanto días regulares como eventos especiales, días festivos y patrones históricos. La solución debe integrar datos de diversas fuentes para ofrecer predicciones precisas y en tiempo real. Además, el sistema debe ser capaz de responder a preguntas específicas de los operadores a través de una aplicación web, proporcionando información relevante para la toma de decisiones y mejorando la experiencia de los usuarios.

### 3) Arquitectura física del sistema

La arquitectura física del sistema incluye sensores IoT y cámaras instalados en puntos estratégicos del intercambiador, responsables de capturar datos en tiempo

real sobre la afluencia y distribución de personas. Estos dispositivos están conectados a una red que transmite los datos a servidores locales o en la nube. Los servidores ejecutan tecnologías como Kafka y Spark para procesar los datos, mientras que una base de datos MongoDB almacena los resultados. La interacción de los operadores con el sistema se realiza a través de una aplicación web desarrollada con Flask, que se comunica directamente con Kafka para generar predicciones.

#### **4) Arquitectura lógica del sistema**

El sistema opera mediante un flujo de datos continuo. Los datos se dividen en tres fuentes principales: datos históricos almacenados, datos en tiempo real provenientes de cámaras y sensores, y datos obtenidos de otras fuentes externas (como información meteorológica, horarios de eventos y días festivos). Estas fuentes se combinan para entrenar un modelo de machine learning desarrollado con Spark MLlib. Cuando un operador realiza una consulta a través de la aplicación web, esta se comunica con Kafka, que transmite la solicitud a Spark. Spark utiliza el modelo predictivo para generar una respuesta que se almacena en MongoDB y se devuelve a la aplicación web, ofreciendo información precisa sobre la afluencia esperada en el día consultado.

#### **5) Decisiones de diseño**

El diseño del sistema se centra en garantizar la escalabilidad, la modularidad y la precisión de las predicciones. Kafka fue elegido para manejar el flujo de datos debido a su capacidad para gestionar mensajes en tiempo real con baja latencia. Spark se empleó para procesar datos históricos y actuales de manera eficiente, integrando el modelo de machine learning que predice la afluencia de personas. MongoDB fue seleccionado como base de datos por su flexibilidad y escalabilidad, permitiendo almacenar tanto los datos históricos como las predicciones generadas. Finalmente, la interfaz basada en Flask ofrece a los operadores una herramienta intuitiva y eficiente para interactuar con el sistema, facilitando la consulta de predicciones en tiempo real.

#### **6) Flujo de datos en el sistema**

El flujo de datos comienza con la captura de información de tres fuentes: los datos históricos almacenados en bases de datos, los datos en tiempo real obtenidos a través de cámaras y sensores, y datos externos que incluyen información sobre días festivos, eventos y otras variables contextuales. Estos datos se integran en el sistema, donde Kafka actúa como un intermediario que organiza las solicitudes y distribuye los datos hacia Spark. Spark utiliza un modelo de machine learning entrenado previamente para procesar la información y generar predicciones. Los resultados se almacenan en MongoDB y son accesibles para la aplicación web.

basada en Flask. Este flujo asegura que los operadores reciban respuestas rápidas y precisas a sus consultas, mientras se actualizan continuamente los datos históricos para mejorar el modelo predictivo.

## **7) Descripción de cambios respecto a la práctica de vuelos**

En comparación con la práctica de vuelos, este sistema adapta las variables de entrada y los procesos para enfocarse en la predicción de afluencia de personas en un entorno urbano. Se integraron sensores IoT y cámaras para capturar datos en tiempo real, y se añadieron datos externos como días festivos y eventos locales, que no estaban presentes en el modelo original. La aplicación web también fue modificada para permitir consultas específicas sobre la afluencia en días y horarios particulares. Además, se estableció un flujo de datos que conecta Kafka con Spark y MongoDB, permitiendo que las predicciones sean generadas y entregadas en tiempo real. Estas adaptaciones aseguran que el sistema sea capaz de enfrentar los retos específicos del intercambiador.

## **8) (EXTRA) Ejemplo práctico del funcionamiento del sistema**

Escenario: Un operador necesita saber cuánta gente se espera en el intercambiador de Plaza Elíptica durante el próximo jueves, considerando que es un día festivo con un evento deportivo en la zona.

### **Paso 1: Datos iniciales**

El sistema dispone de tres fuentes principales de datos:

1. Datos históricos almacenados: Incluyen registros de afluencia en días festivos previos, horarios pico, y eventos similares.
2. Datos en tiempo real: Sensores IoT y cámaras capturan información actual sobre el flujo de personas en el intercambiador.
3. Datos externos: Información proveniente de calendarios festivos, horarios de eventos y condiciones meteorológicas se obtiene de APIs y fuentes externas.

### **Paso 2: Consulta desde la aplicación web**

El operador utiliza la aplicación web basada en Flask para realizar una consulta: "¿Cuál será la afluencia estimada el próximo jueves, de 17:00 a 20:00 horas?" La interfaz muestra un campo para seleccionar la fecha y el rango horario, junto con una descripción opcional del contexto (por ejemplo, la presencia de un evento).

### **Paso 3: Comunicación entre componentes**

1. La aplicación Flask envía la solicitud al sistema.
2. Flask se comunica con Kafka, que actúa como intermediario, asegurándose de que la consulta sea organizada y transmitida correctamente.

### **Paso 4: Procesamiento de la predicción**

1. Kafka distribuye la solicitud y los datos relevantes hacia Spark Streaming, que accede al modelo predictivo entrenado previamente con Spark MLlib.
2. Spark utiliza los datos históricos, actuales y externos para generar una predicción ajustada al contexto del día consultado.
  - Por ejemplo, considera que el jueves es festivo y hay un evento deportivo cerca, por lo que anticipa un aumento significativo en la afluencia de público durante la tarde.

#### Paso 5: Almacenamiento y respuesta

1. Los resultados de la predicción, como el número estimado de personas por hora y las métricas de confianza, se almacenan en MongoDB en formato JSON.
2. Flask recupera los datos desde MongoDB y los presenta en un formato gráfico en la interfaz web.

#### Paso 6: Visualización en la web

El operador ve en la pantalla:

- Una gráfica de barras mostrando la afluencia estimada de 17:00 a 20:00, desglosada por horas.
- Una nota que indica la probabilidad de saturación en ciertos accesos, recomendando medidas como aumentar el personal disponible o redirigir el flujo en áreas congestionadas.

#### Paso 7: Actualización del modelo

1. Los datos capturados en tiempo real por sensores y cámaras se almacenan en un sistema como HDFS o S3 para su posterior análisis.
2. Periódicamente, el modelo de machine learning se re-entrena con estos datos, mejorando su precisión y adaptándose a patrones emergentes.

#### Resultado:

El operador cuenta con información precisa y en tiempo real que le permite tomar decisiones como redistribuir recursos, planificar horarios del personal o ajustar la señalización para gestionar mejor la afluencia de público durante el evento y garantizar una experiencia segura para los usuarios.

## 9) Conclusiones

La implementación de un sistema basado en Big Data para predecir la afluencia de personas en el intercambiador de Plaza Elíptica representa una solución innovadora y efectiva para gestionar espacios públicos. La integración de datos históricos, datos en tiempo real y datos externos en un modelo de machine learning permite generar



POLITÉCNICA

UNIVERSIDAD  
POLITÉCNICA  
DE MADRID



European University

predicciones precisas y en tiempo real. Estas predicciones facilitan la planificación y gestión de recursos, mejorando la experiencia de los usuarios y aumentando la seguridad en el intercambiador. En el futuro, el sistema podría incluir funcionalidades adicionales, como el análisis de sentimientos a partir de redes sociales, para enriquecer las predicciones y optimizar aún más la toma de decisiones.