

# Diseño y desarrollo de un microservicio para la gestión de información de monitorización y predicciones de tráfico en red

28 de noviembre de 2022

**TRABAJO FIN DE MÁSTER**

Máster Universitario en Ingeniería de  
Telecomunicación

**Autor:** Enrique Fernández Sánchez

**Tutor:** Pablo Pavón Mariño



# Índice general

Índice de figuras . . . . .	3
Listado de ejemplos . . . . .	4
<b>1 Introducción</b>	<b>5</b>
1.1 Contexto del trabajo . . . . .	5
1.2 Motivación . . . . .	5
1.3 Descripción Global . . . . .	5
1.4 Objetivos . . . . .	5
1.5 Resumen capítulos de la memoria . . . . .	5
<b>2 Tecnologías empleadas</b>	<b>6</b>
2.1 Arquitectura y microservicios . . . . .	6
2.2 Bases de datos . . . . .	7
2.3 Modelo de predicción . . . . .	7
2.4 Lenguajes de programación y frameworks . . . . .	7
2.5 Tecnologías utilizadas en un despliegue en producción . . . . .	7
<b>3 Diseño e implementación del sistema</b>	<b>8</b>
3.1 Descripción REST API . . . . .	8
3.2 Estructura de la aplicación . . . . .	8
3.3 Modelos de datos . . . . .	8
3.4 Endpoints . . . . .	8
3.5 Implementación del sistema . . . . .	8
<b>4 Pruebas y validación del sistema</b>	<b>9</b>
<b>5 Conclusiones</b>	<b>10</b>
5.1 Propuestas futuras . . . . .	10
<b>6 Bibliografía</b>	<b>11</b>
<b>Anexos</b>	<b>12</b>
Anexo I. Generación dataset sintético . . . . .	12

# Índice de figuras

# Listado de ejemplos

# Capítulo 1

## Introducción

1.1 Contexto del trabajo

1.2 Motivación

1.3 Descripción Global

1.4 Objetivos

1.5 Resumen capítulos de la memoria

## Capítulo 2

# Tecnologías empleadas

En este capítulo, se van a presentar las diferentes tecnologías utilizadas para la implementación de la aplicación.

### 2.1 Arquitectura y microservicios

En primer lugar, se va a comentar acerca de la arquitectura escogida. En este caso, se decide realizar el sistema siguiendo la premisa de ser una **API** (*Application programming interface*), ya que uno de los objetivos propuestos es el de poder realizar ejecuciones de la aplicación, desde otras aplicaciones.

Una API permite a dos componentes comunicarse entre sí mediante una serie de reglas. Además, supone un "contrato" en el que se establecen las solicitudes y respuestas esperadas en la comunicación.

Dependiendo de la implementación de la API que se realice, distinguimos cuatro tipos de API:

- API de SOAP. Utilizan un protocolo de acceso a objetos. Los interlocutores intercambian mensajes XML. En general, es una solución poco flexible.
- API de RPC. Basado en llamadas de procedimientos remotos. El cliente ejecuta una función en el servidor, y este responde con la salida de la función.
- API de WebSocket. Solución moderna de desarrollo de API, que utiliza objetos JSON y un canal bidireccional para realizar la comunicación entre el cliente y el servidor.
- API de REST. Solución más popular. El cliente envía solicitudes al servidor como datos, utilizando métodos HTTP. Es una opción muy flexible.

En el caso de nuestra aplicación, se decidió utilizar el tipo REST API, ya que permite una sencilla implementación de cara al cliente que quiera utilizar dicha interfaz.

## **2.2 Bases de datos**

## **2.3 Modelo de predicción**

## **2.4 Lenguajes de programación y frameworks**

## **2.5 Tecnologías utilizadas en un despliegue en producción**

## Capítulo 3

# Diseño e implementación del sistema

### 3.1 Descripción REST API

### 3.2 Estructura de la aplicación

### 3.3 Modelos de datos

### 3.4 Endpoints

### 3.5 Implementación del sistema



## Capítulo 4

# Pruebas y validación del sistema

# Capítulo 5

## Conclusiones

### 5.1 Propuestas futuras

# Capítulo 6

## Bibliografía

1. [¿Qué es una API?](#)

# Anexos

## Anexo I. Generación dataset sintético