

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

*Autor:* Enrique Fernández Sánchez

*Tutor:* Pablo Pavón Mariño

Universidad Politécnica de Cartagena (UPCT)



4 de enero de 2023

1. Introducción
2. Tecnologías empleadas
3. Implementación del sistema
4. Validación del sistema
5. Conclusiones
6. Bibliografía

- *Abstract*: Aplicación que permite almacenar muestras de monitorización de tráfico en red, y a su vez, generar predicciones futuras del tráfico de red, en función de la información almacenada.

## Objetivos del proyecto

- Diseñar una aplicación siguiendo la metodología de microservicio.
- Investigar herramientas de predicción de series temporales.
- Investigar opciones de almacenamiento para muestras temporales.
- Utilizar herramientas de documentación que permitan conocer la estructura de la aplicación.

Definición microservicio

content...

API

content...

Tipo relacional

asdasdasda

Tipo serie temporal

content...

## Python

Lenguaje de programación orientado a objetos, interpretado y de alto nivel. Muy popular en los siguientes ambitos:

- Aplicaciones web.
- Data Science
- Inteligencia Artificial



Figura: asd

## FastAPI

Framework moderno y rápido para construir APIs. Rápido, intuitivo y robusto, además de basado en estándares de la industria.

- Rápido
- Intuitivo
- Robusto



Figura: asd

## Prophet

Framework del lenguaje de programación Python, desarrollado por Meta (Facebook)



Figura: asd

# Despliegue en producción

---

content...



# Descripción API REST (I)

---

content...

# Descripción API REST (II)

---

content...

content...

# Estructura (II)

---

content...

content...

- asdasd

Networks
id_network: Int, Public Key, Unique
name: String
description: String
ip_red: String
influx_net: String

Interfaces
id_interface: Int, Public Key, Unique
name: String
description: String
influx_if_rx: String
influx_if_tx: String
network: Int, Foreign Key

content...

# Predicción de tráfico de red

---

content...



## CRUD: networks (/networks)

- **Información de todas** las redes:  
*GET* - /networks
- **Crear** una red:  
*POST* - /networks
- **Información de una** red:  
*GET* - /networks/<net\_id>
- **Eliminar** una red:  
*DELETE* - /networks/<net\_id>
- **Actualizar** una red:  
*PATCH* - /networks/<net\_id>

## CRUD: interfaces (/networks/<id1>/interfaces)

- **Información de todas** las interfaces:  
*GET* - /networks/id/interfaces
- **Crear** una interfaz:  
*POST* - /networks/id/interfaces
- **Información de una** interfaz:  
*GET* - ../interfaces/<id2>
- **Eliminar** una interfaz:  
*DELETE* - ../interfaces/<id2>
- **Actualizar** una interfaz:  
*PATCH* - ../interfaces/<id2>

# Resumen rutas HTTP (II)

---

content...

# Validación del sistema (I)

---

content...

Demo

content...

# Validación del sistema (II)

---

content...

- asdsad
- asdasd

## Propuestas futuras

- asd
- asd

- La contenida en la memoria del proyecto: páginas 53 - 54

**Muchas gracias por su atención**

¿Preguntas?

Enlace a la aplicación:

`https://tfm-api.ranii.pro:8443/`