

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 **microservicios**.
- 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 de microservicio

Sistemas que cumplen las siguientes premisas:

- Sistemas pequeños, independientes y poco “acoplados”.
- Código fuente separado entre los diferentes servicios, no necesariamente mismo lenguaje.
- Los servicios se comunican entre sí utilizando APIs.
- Cada sistema es independiente, y responsable de su persistencia de datos.

Importancia microservicios

content...

- Una API permite a dos **componentes comunicarse** entre sí **mediante** una serie de **reglas**.
- Supone un “**contrato**” en el que **se establecen las solicitudes** y respuestas **esperadas en la comunicación**.

Tipos de API

Dependiendo de la implementación, distinguimos entre cuatro tipos:

- **SOAP.**
- **RPC.**
- **Web Socket.**
- **REST.**

- En función del tipo de dato a almacenar, se distinguen dos bases de datos dentro de la aplicación:

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



FastAPI

Framework moderno y rápido para construir APIs. Características:

- Rápido: rendimiento equivalente a otros lenguajes (NodeJS o Go).
- Intuitivo: soporta autocompletado.
- Robusto: herramienta Swagger/ReDoc automática.



Prophet

Framework del lenguaje de programación Python, desarrollado por Meta (Facebook). Agrupa una serie de procedimientos que permiten realizar predicciones en un dataset temporal. Características:

- Permite encontrar y tener en cuenta efectos no lineales (tendencias diarias, semanales, mensuales...).
- Permite predecir datos en días vacacionales.
- Basado en inferencia estadística, más eficiente que si utilizáramos técnicas de *Machine Learning*.

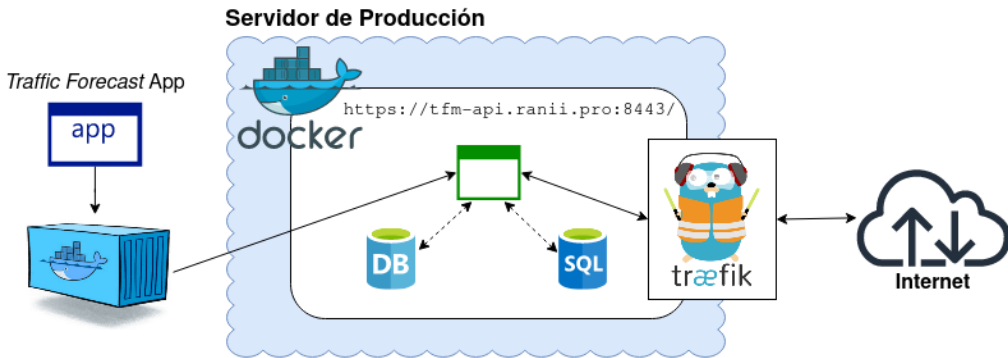


PROPHET

Despliegue en producción

Herramientas utilizadas para el despliegue del sistema en un entorno de producción:

- Docker
- docker-compose
- Traefik



content...

Agentes

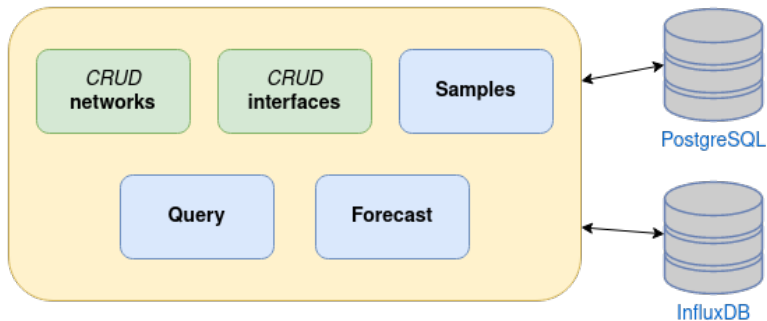
Se identifican los diferentes “agentes” presentes en el sistema:

- **Redes** (`networks`), corresponde con una red que contiene interfaces a monitorizar.
- **Interfaces** (`interfaces`), corresponde con las interfaces de red que queremos monitorizar.
- **Muestras de monitorización** (`samples`). Valor de tráfico asociado a una interfaz en un intervalo determinado.

Descripción API REST (II)

Esquema de la aplicación implementada

Traffic Forecast Microservice



Implementación (I)

content...

Implementación (II)

content...

content...

- Definimos los datos y su **representación dentro** de nuestra **aplicación**.
- **En función del tipo de dato**, almacenaremos en **base de datos tipo relacional** (SQL) o **base de datos tipo serie temporal** (InfluxDB).

Tablas utilizadas en la base de datos SQL

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

- En InfluxDB almacenamos las muestras de tráfico en red.

Configuración InfluxDB para el sistema

InfluxDB	Descripción
measurement	Red a monitorizar, valor almacenado en <code>Networks::influx_net</code>
fields	Nombre del valor a monitorizar (<i>link_count</i>)
tags	Solo disponemos un tag, llamado interface, contiene la información del identificador de una interfaz
points	Corresponde con el valor numérico del campo field.

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>

Samples

- **Importar** datos de topología con **más de una interfaz**:
POST -
`/samples/id/import_topology`
- **Importar** datos de **una interfaz**:
POST -
`/samples/id/import_interface/id`

Query Samples

- **Consultar datos** de monitorización almacenados:
GET - `/query/`

Forecast

- **Ejecutar predicción** de tráfico en red:
POST - `/forecast/`

Validación del sistema (I)

content...

Demo

content...

Validación del sistema (II)

content...

- Alcanzados todos los objetivos propuestos.
- Se ha desarrollado una **microservicio completo**, permitiendo ser implementado por otras aplicaciones.
- El sistema es **capaz** de **almacenar** muestras de monitorización, además de poder **generar predicciones** de tráfico en red en la escala temporal que el usuario solicite.

Propuestas futuras

- Permitir la **importación** de **datos de monitorización** de **herramientas específicas** de planificación de red.
- Añadir **funcionalidad de SSE** (Server Side Event) para **tareas** que requieran un **largo tiempo de ejecución**.
- Extender la funcionalidad de las **predicciones**, permitiendo hacer selección **más selectiva**, o añadir **filtrados extra**.

- 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/>