



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

Monitorización sensores IOT



Presentado por Daniel Mellado Hurtado
en Universidad de Burgos — 19 de septiembre
de 2021

Tutor: Bruno Baruque Zanon



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre Bruno Baruque Zanon, profesor del departamento Ingeniería Informática, área de Ciencia de la computación e inteligencia.

Expone:

Que el alumno D. Daniel Mellado Hurtado, con DNI 71363864Q, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Monitorización sensores IOT.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 19 de septiembre de 2021

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

El tratamiento, monitorización y gestión de datos procedentes de dispositivos inteligentes es más prioritario que nunca, poder reconocer patrones y tendencias en estos datos puede ayudar a predecir fallos o eventos no deseados con suficiente tiempo de antelación como para poder ser evitados.

Con el fin de desarrollar un sistema capaz de mantener monitorizados y gestionados diversos dispositivos IOT se ha desarrollado esta aplicación.

Descriptores

IoT, internet de las cosas, sensores, monitorización, inteligencia artificial, aprendizaje automático, aprendizaje incremental, predicciones, PRTG, Elasticsearch.

Abstract

The processing, monitoring and management of data from smart devices is more of a priority than ever, being able to recognise patterns and trends in data can help to predict failures or unwanted events sufficiently in advance so that they can be avoided.

In order to develop a system capable of keeping various IOT devices monitored and managed, this application has been developed.

Keywords

IoT, internet of things, sensors, monitoring, artificial intelligence, machine learning, incremental learning, predictions, PRTG, Elasticsearch.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Material adjunto	1
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. El Internet de las Cosas (IOT)	5
3.2. Big Data	6
3.3. Anomalías	6
3.4. Time series	7
3.5. Data stream	7
3.6. Inteligencia artificial	7
3.7. NoSQL	8
3.8. Daemons	9
3.9. Curl	9
Técnicas y herramientas	11
4.1. Visualización y análisis de datos	11
4.2. Motor de búsquedas	12
4.3. Control de versiones	13
4.4. Lenguajes de programación	13

4.5. librerías Python	14
4.6. Archivos de texto	15
4.7. Sistemas Operativos	15
4.8. IDEs	15
4.9. Virtualización	16
4.10. Comunicación	16
4.11. Cliente SSH	16
4.12. ELK Stack	16
Aspectos relevantes del desarrollo del proyecto	19
5.1. Motivación en la elección del proyecto	19
5.2. Desarrollo del proyecto	19
5.3. Problemas y resoluciones	22
Trabajos relacionados	25
6.1. Elasticsearch Machine Learning	25
6.2. SkySpark	25
6.3. Datadog IoT Monitoring	25
6.4. AWS IoT Device Management	26
Conclusiones y Líneas de trabajo futuras	27
7.1. Conclusión	27
7.2. Líneas de trabajo futuras	27
Bibliografía	29

Índice de figuras

4.1. ELK Stack	17
5.2. monitorización de sensores	21

Índice de tablas

Introducción

Vivimos en un mundo en el que todo está conectado y parametrizado, miles de dispositivos inteligentes capaces de captar y transmitir datos, desde los relojes que miden nuestras pulsaciones hasta los sensores que miden el nivel de oxígeno en la Estación Espacial Internacional.

Por ello el tratamiento de los datos procedentes de todos estos sensores es primordial hoy en día y poder reconocer patrones y tendencias en estos datos puede ser de gran utilidad, no solo para empresas sino para todo el mundo.

Con el fin de mantener monitorizados y gestionados diversos dispositivos IOT, los cuales han sido cedidos por un cliente interesado en el proyecto, se ha desarrollado esta aplicación.

A lo largo de esta memoria y sus anexos se verá con detalle cómo ha sido el desarrollo del proyecto, su diseño y como utilizar el producto final.

1.1. Material adjunto

Los materiales que se encuentran adjuntos en la entrega de este proyecto son los siguientes:

- Máquina virtual ubuntu server con la aplicación ya instalada.
- Memoria del Trabajo de Fin de Grado más los documentos de anexos.
- Script de instalación *install.sh*.
- Paquete debian *Monitorizacion_IOT_1.0_all.deb*.

- Fichero Jupyter Notebook con las pruebas de regresión realizadas.
- Ficheros con los datasets usados para las pruebas.
- El código del programa.

Objetivos del proyecto

El objetivo de este proyecto consta de la captación dinámica de diversos datos provenientes de sensores IOT. Los datos, cedidos por el cliente, se han de tratar y almacenar en una base de datos para poder ser consultados y monitorizados por el usuario en cualquier momento.

Se implementará también un algoritmo de aprendizaje automático que ha de aprender con los datos obtenidos a identificar patrones que puedan llevar a situaciones no deseadas y que esto pueda ayudar al usuario a tomar decisiones con la suficiente antelación.

Objetivos

1. La instalación y configuración de un sistema capaz de recoger, almacenar y gestionar datos de sensores.
2. La implementación de motores de búsqueda para facilitar encontrar los datos deseados con la mayor precisión posible.
3. Que la monitorización de los datos se muestre lo más clara y accesible posible para el usuario.
4. Realizar un algoritmo de aprendizaje automático que pueda predecir comportamientos y patrones en los datos de los sensores.

Objetivos personales

1. Utilizar un repositorio en github y aplicar conocimientos sobre gestión de proyectos.

2. Adentrarme y aprender sobre tratamiento, procesado y gestión de datos.
3. Tratar con dispositivos IoT.

Conceptos teóricos

En este apartado se desarrollarán los principales conceptos teóricos que facilitarán el entendimiento del trabajo.

3.1. El Internet de las Cosas (IOT)

El Internet de las Cosas (en inglés *Internet of things* IOT) es la red de objetos físicos que incorpora sensores, software y otras tecnologías con la finalidad de aportar datos a otros dispositivos a través de Internet.[21]

En las últimas décadas se ha visto incrementada esta tecnología debido a la rápida evolución de internet, la aparición de servicios de almacenamiento de tipo nube y el avance en el hardware que ha permitido que estos dispositivos recopilen datos con una intervención humana mínima.

Esta tecnología es especialmente interesante para las empresas, tener una medición de sus sistemas facilita el estudio de balance de beneficio, así como ayuda a detectar fallos o eventos no deseados.

Sensores

Los sensores son uno de los principales componentes que tienen los sistemas para la captación de datos. Detectan cambios en el entorno, convierten un fenómeno físico en un voltaje analógico medible y los transmiten para su lectura. [13]

Para este proyecto se han utilizado tres tipos de sensores, a continuación, se describirán brevemente:

Sensores de presión de agua

Son instrumentos capaces de captar la presión que el agua ejerce sobre el dispositivo y transforma en una señal eléctrica donde la cantidad depende de la presión aplicada sobre este. [7]

Sensores de humedad

Los sensores de humedad, también conocidos como higrómetros, detectan el contenido de humedad en un área dada, pudiendo ser usados tanto en interiores como exteriores, se basan en la detección de agua o vapor en el entorno.

En este proyecto se monitorizan dos sensores de humedad, uno que mide la humedad en el ambiente y otro que lo mide en una parcela de una bodega.

Sensores de Consumo de bombas

Las bombas de agua son máquinas hidráulicas que utilizan energía para mover el agua, los sensores se encargan de medir el consumo energético de estas, transforman dicha energía que en señales digitales.

3.2. Big Data

El Big Data está formado por conjuntos complejos de datos de un elevado tamaño y con una gran velocidad de crecimiento. Estos conjuntos de datos tienen tal volumen que es imposible de almacenarlos, gestionarlos, procesarlos y/o analizarlos con software de procesamiento de datos convencionales.

Pese a su dificultad, estos volúmenes tan masivos son de gran utilidad para abordar problemas de todo tipo que no habrían sido posibles de solucionar antes. [22]

3.3. Anomalías

En Big Data las anomalías son datos que no siguen el patrón natural de la mayoría de los datos, esto puede ser síntoma de algún tipo de fallo o de algún evento no esperado.

Para detectar estas anomalías o comportamientos atípicos es necesario distinguir en un conjunto de datos cuales son anomalías. Esta puede ser una tarea extremadamente difícil en conjuntos de gran tamaño como se da en

Big Data, para ello es necesario la incorporación de modelos de machine learning que identifiquen estas anomalías.[16]

3.4. Time series

Time series o serie temporal es una sucesión de datos ordenados de forma cronológica. Estos datos pueden estar separados tanto por intervalos iguales de tiempo o desiguales

Uno de los usos de time series es el de análisis de predicciones. A la hora de realizar predicciones el tiempo es normalmente la variable independiente y el objetivo es realizar predicciones a futuro [18]

3.5. Data stream

Los data stream son datos que se generan de forma continua y de diversas fuentes. Estos datos suelen ser utilizados para análisis de datos y son procesados de manera secuencial y gradual.

Los datos que se generan pueden ser de cualquier tipo, en el caso de este proyecto los datos generados son de tipo time series.[4]

3.6. Inteligencia artificial

La inteligencia artificial o IA es la simulación de inteligencia humana por parte de las máquinas, incluyendo el aprendizaje, el razonamiento y la auto corrección.

Dicha disciplina alberga muchos campos como pueden ser el aprendizaje automático (Machine learning) o el aprendizaje profundo (Deep Learning).

Actualmente la inteligencia artificial se aplica a cualquier campo, pudiendo realizar tareas que por otros medios serían imposibles.[19]

Machine learning

Machine learning es una rama derivada de la inteligencia artificial que se centra en el uso de datos y algoritmos para imitar el aprendizaje humano. [15]

Se puede dividir los algoritmos de machine learning en tres partes:

- **Proceso de decisión:** por lo general los algoritmos de machine learning se utilizan para problemas de predicción o clasificación con la ayuda de datos (los cuales pueden estar etiquetados o no) el algoritmo será capaz de estimar patrones en los datos proporcionados.
- **Función de error:** una función de error que sirva para evaluar las predicciones del modelo y medir la precisión del mismo.
- **Proceso de optimización del modelo:** si el modelo puede mejorar la precisión con los datos de entrenamiento entonces se ajustan los pesos. El algoritmo repitiera este proceso hasta que se haya llegado al umbral de precisión.

Aprendizaje supervisado

El aprendizaje supervisado utiliza datos etiquetados, es decir, datos para los que se conoce la respuesta para entrenar el modelo que será capaz de clasificar o predecir cuándo se le presenten datos de los que se desconoce la respuesta.

Incremental/online learning

El aprendizaje incremental son todos aquellos algoritmos escalables que aprenden de forma secuencial y van mejorando el modelo de forma constante con un flujo infinito de datos (data streams).

En el aprendizaje incremental no se tiene acceso a todo el conjunto de datos cuando se crea el modelo, sino que se tiene que crear dicho modelo para que se adapte y aprenda según tenga acceso a los datos.

Esta técnica de machine learning es muy útil cuando se manejan datos a tiempo real donde cada minuto se generan nuevos datos, cómo puede ser el caso de sensores.^[11]

3.7. NoSQL

Not Only SQL, estas bases de datos, son diseñadas para modelos de datos específicos y cuentan con esquemas flexibles. Están optimizadas para aplicaciones que requieren una gran cantidad de datos, baja latencia y un modelo de datos flexibles.

Existen una gran cantidad de bases de datos NoSQL: clave-valor, Documentos, Gráficos, etc...

Documentos

En este tipo de bases de datos, los datos se representan como un objeto o documento, este puede ser de distintos tipo cómo JSON, XML, YAML o BSON.[3]

Los documentos no requieren que se ajusten a ningún esquema o estructura específica.

3.8. Daemons

En informática, un daemon es un programa que se ejecuta cómo un proceso en segundo plano. En Linux los daemons son administrados por *systemd* y se administran mediante el comando *systemctl*. Estos leen los archivos con el nombre *nombre.Service* que contiene información sobre cómo se ha de inicializar. Estos archivos se almacenan en */etc,usr/lib,run/systemd/system*[6]

3.9. Curl

El comando Curl (Cliente URL) proporciona una forma de verificar la conexión a las URL y transferir datos. El comando curl es compatible con diversos protocolos cómo: HTTP, HTTPS, FTP, FTPS, IMAP, POP3, SMB, SCP, TELNET, SMTP entre otros.[14]

Técnicas y herramientas

4.1. Visualización y análisis de datos

Para monitorizar nuestros datos debemos hacer uso de herramientas de Network monitoring que nos permita detectar fallos y anomalías en los datos así cómo alertarnos si alguno de los datos monitorizados pasa por debajo de un umbral.

PRTG Network Monitor

PRTG Network Monitor es una herramienta de monitorización desarrollada por Paessler, que permite supervisar y monitorizar datos de sensores, la versión gratuita está limitada a 100 sensores, una de las ventajas de PRTG es la cantidad de variedad de sensores que existen de forma predefinida. Permite mostrar alertas al encontrar problemas o métricas inusuales.[23]

ThingSpeak

ThingSpeak es una plataforma open source orientada al Internet de las Cosas (IOT) que permite recoger, almacenar datos de sensores en la nube usando el protocolo HTTP y visualizar los datos. También permite realizar análisis de los datos usando MATLAB.

ThingSpeak consta de aplicaciones complementarias con diversas funcionalidades. [30]

Grafana

Grafana permite visualizar y representar métricas de datos sin importar donde estén almacenados, la visualización es rápida y flexible con multitud de opciones. En Grafana se pueden definir alertas sobre las métricas que se desee y manda notificaciones de estas, también cuenta con cientos de plugins oficiales.

Contiene una opción de pago en la que ellos proporcionan el servidor y otro completamente gratuita en la que permite ejecutarlo en cualquiera de los sistemas operativos principales.[12]

Splunk

Splunk es una plataforma que permite monitorizar, buscar y analizar datos en tiempo real, esta herramienta consta de cientos de aplicaciones y es compatible con casi cualquier formato de logs. Splunk es gratis con un volumen de datos diarios de menos de 500 MB. [29]

Uno de sus atractivos es que consta de una herramienta machine learning que aprende de tus datos detectando anomalías y prediciéndolas.

Elección final

Debido a que el cliente que va a facilitar los datos de los sensores utiliza PRTG se he optado por esta opción.

4.2. Motor de búsquedas

Algolia

Algolia es un motor de búsqueda accesible vía API, la cual se puede integrar en web y aplicaciones móviles. Funciona exclusivamente con datos en formato JSON. Algolia posee una inteligencia artificial que aprende del usuario para mejorar su experiencia de uso. Consta de una versión gratuita, la cual tiene limitada el número de resultados de búsqueda.[1]

Elasticsearch

Elasticsearch es uno de los motores de búsqueda más valorados del mercado[9] , de código abierto que proporciona análisis y búsqueda de datos

a tiempo real. Esta desarrollado en Java y sólo soporta JSON cómo tipos de respuesta.

Posee una arquitectura distribuida siendo la escalabilidad horizontal, una de sus ventajas es su rapidez a la hora de realizar búsquedas de textos complejos para facilitar esto funciona mediante índices invertidos.[10]

Solr

Apache Solr es un motor de búsqueda open source escrito en java con comandos escritos en HTTP y guarda los archivos utilizando XML. Esta optimizado para tráfico de altos volúmenes de datos y todo en tiempo real.[5]

Elección final

Debido a que el entorno de Elasticsearch facilita la ingesta de datos, almacenamiento y la posterior visualización de estos se ha elegido esa opción.

4.3. Control de versiones

Herramientas consideradas: GitHub y GitLab.

Para organizar el proyecto se ha elegido la plataforma de control de versiones de GitHub debido a ser el software de versiones Git más utilizado.

4.4. Lenguajes de programación

Python

Python es un lenguaje de programación multiplataforma y multiparadigma, soporta tanto programación orientación a objetos cómo programación imperativa. Es un lenguaje cuya sintaxis se centra en un código legible y sintaxis clara. [25]

Bash

GNU Bash es una interfaz de usuario de línea de comandos y un lenguaje de programación de scripting, se creó originalmente para el sistema operativo GNU. Actualmente disponible en la mayoría de distribuciones GNU/Linux, Mac OS X y también tiene versiones para Windows 10 y Android.

4.5. librerías Python

Scikit-learn

Scikit-learn es un módulo de programación en python para machine learning, permite realizar algoritmos de clasificación, regresión, Análisis de grupos, Máquinas de vectores de soporte, Árboles de decisión, etc...

Scikit-learn trabaja a su vez con otras librerías como NumPy, SciPy, pandas, SymPy, Matplotlib, Ip[y].[\[27\]](#)

scikit-multiflow

Scikit-multiflow es una librería de machine learning para python dedicada a streaming data y multi-output learning que permite generar y evaluar data streams.[\[20\]](#)

NumPy

NumPy es una librería de Python especializada en el cálculo numérico y computación de datos.

Pandas

Pandas es una librería open source de programación para Python y utilizando como extensión NumPy, se utiliza para análisis de datos, ofreciendo estructuras de datos y operaciones con tablas numéricas y series temporales.[\[31\]](#)

River

River es una librería de machine learning en streaming para python, juntando las librerías creme y scikit-multiflow.

Es una librería reciente por lo que la documentación no está completa y aún no hay muchos proyectos que hayan utilizado la librería.[\[26\]](#)

4.6. Archivos de texto

JSON

El formato de texto JSON (JavaScript Object Notation) se utiliza para el intercambio de datos. Resulta sencillo tanto de leerlo cómo de escribirlo para los humanos y sencillo de interpretarlo y generarlo para las máquinas.

El formato JSON es independiente del lenguaje JavaScript.

En este trabajo he decidido utilizar JSON para el intercambio de datos sobre otros formatos como CSV por su sencillez y su compatibilidad con PRTG y Elasticsearch.^[17]

4.7. Sistemas Operativos

Windows 10

Debido a que el cliente que va a facilitar los datos utiliza PRTG para monitorizar los sensores y este es exclusivo de Windows no se tiene más alternativa que utilizar este sistema operativo para albergar un servidor PRTG.

Ubuntu Server 20.04.2

Para albergar tanto Elasticsearch como el programa que conecte todos los sistemas se a escogido la última versión de ubuntu server debido a su ligera y por la facilidad que este tendría para implementarlo en un servidor real.

4.8. IDEs

L^AT_EX

Herramientas consideradas: Overleaf y Textmaker

La herramienta que se ha decidido utilizar para redactar la memoria y los anexos de este proyecto ha sido Overleaf dividido a que es un sencillo editor L^AT_EX basado en la nube y no requiere de ninguna instalación previa.

Vim

Para la edición de archivos de texto en el servidor de Ubuntu se ha decidido utilizar vim debido a que es un editor que en los sistemas ubuntu ya viene preinstalado además de su gran versatilidad y ligereza.

Jupyter Notebook

Es una aplicación cliente-servidor que permite crear y compartir documentos web con formato JSON. Jupyter soporta los lenguajes de programación Julia, Python y R.

En este proyecto se utiliza para documentar las pruebas realizadas.[\[24\]](#)

4.9. Virtualización

VirtualBox

Para albergar las máquinas virtuales se ha utilizado VirtualBox debido a ser la herramienta open source más famosa de virtualización.

4.10. Comunicación

Microsoft Teams

Microsoft Teams es una plataforma de comunicación y colaboración para reuniones telemáticas.

4.11. Cliente SSH

PuTTY

PuTTY es un cliente SSH y telnet con licencia libre.

Este software permite conectarse de forma remota a al Ubuntu server así cómo intercambiar ficheros entre el server y el host.

4.12. ELK Stack

ELK es la combinación de Elasticsearch, Logstash y Kibana.

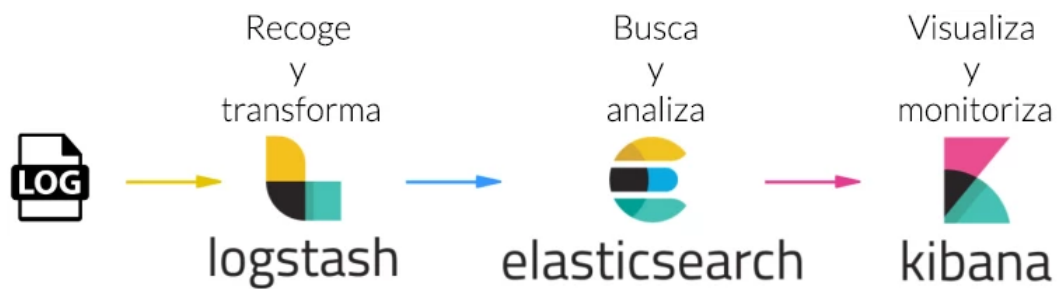


Figura 4.1: ELK Stack

Elasticsearch

Cómo se comentó con anterioridad en el apartado de motores de búsqueda Elasticsearch se encarga del almacenamiento de datos y aporta un potente motor de búsqueda.

Logstash

Logstash es un pipeline de procesamiento de datos capaz de ingresar datos de múltiples fuentes y enviarlos a una variedad de salidas, pero en el caso de nuestro programa será Elasticsearch.

Kibana

Kibana es una interfaz de usuario que permite la visualización de datos de Elasticsearch. Esta herramienta nos facilita la creación de histogramas, gráficos, mapas y demás formas gráficas de representación de los datos.

Aspectos relevantes del desarrollo del proyecto

En este apartado se detallarán los aspectos más relevantes del proyecto.

5.1. Motivación en la elección del proyecto

La motivación a la hora de elegir este proyecto surgió por lo interesante de la premisa, tratar con datos de sensores reales y diseñar un sistema que no solo mantenga monitorizados dichos sensores, sino que aprenda de ellos resultado ser suficiente para la elección de este proyecto.

5.2. Desarrollo del proyecto

Inicio

Las primeras semanas tras la asignación del proyecto se destinaron en asentar la idea general del proyecto y a establecer una serie de objetivos, así como la ruta que había que llevar para lograrlos con éxito.

Durante esta primera etapa, la mayoría del tiempo se destinó al aprendizaje y a la investigación de posibles herramientas y tecnologías que podrían ser de utilidad.

Se eligió Elasticsearch como principal motor de búsqueda y como base de datos para los sensores. A partir de ahí comenzó el desarrollo del proyecto.

Reuniones con el cliente

A lo largo del proyecto se celebraron una serie de reuniones con el cliente, el cual dispone de una serie de sensores encargados de medir ciertos parámetros de una bodega. Estos sensores los tenía conectados al software de monitorización PRTG, por lo tanto, se estableció como punto de partida y cómo uno de los primeros objetivos el recopilar los datos de PRTG.

Conexión entre servidores

El primer gran reto que se presentó fue realizar la conexión entre el servidor PRTG con los sensores del cliente configurados y el servidor Ubuntu, el cual contiene Elasticsearch que se encargaría de la monitorización de dichos sensores.

A la hora de descargar los datos de PRTG surgieron una serie de adversidades que se fueron solucionando, estos inconvenientes se encuentran detallados en el apartado 5.3. Una vez se logró que se fueran descargando los datos procedentes de PRTG de forma periódica llegó la hora de configurar las herramientas de *ELK Stack* para poder tratar los datos obtenidos y poder indexarlos correctamente en Elasticsearch.

Una vez conseguida la conexión entre los dos sistemas y logrado un flujo continuo de datos también se logró uno de los objetivos propuestos, monitorizar los datos. Gracias a Elasticsearch y Kibana se podía mantener un control sobre los datos de los sensores pudiendo representarlo de diversas formas gráficas.

Aplicación de modelos machine learning

El siguiente gran reto fue la realización de un modelo capaz de poder aprender de los datos recibidos y así poder realizar predicciones a futuro.

Durante esta etapa se llevó a cabo la investigación sobre que métodos y algoritmos utilizar. Se llegó a la conclusión de que la mejor idea y la más interesante, sería implementar algoritmos de aprendizaje incremental, de esta forma el modelo que se cree podrá aprender según obtenga los datos. Para ello se decidió la utilización de la librería de Python River, la cual, pese a seguir en desarrollo y contar con una documentación aún por terminar, es una librería prometedora.

Se diseñó un sistema con el cual tras descargarse de los datos de PRTG y almacenarlos en Elasticsearch se volvían a extraer para poder ser entrenados

por un modelo. Se decidió que se almacenaran y luego se volvieran a extraer en vez de utilizar los datos obtenidos directamente de PRTG porque de esta forma el entrenamiento no dependía de PRTG y si se desactiva esta funcionalidad durante un tiempo y se vuelve a activar después, poder tener la opción de entrenar el modelo con datos pasados.

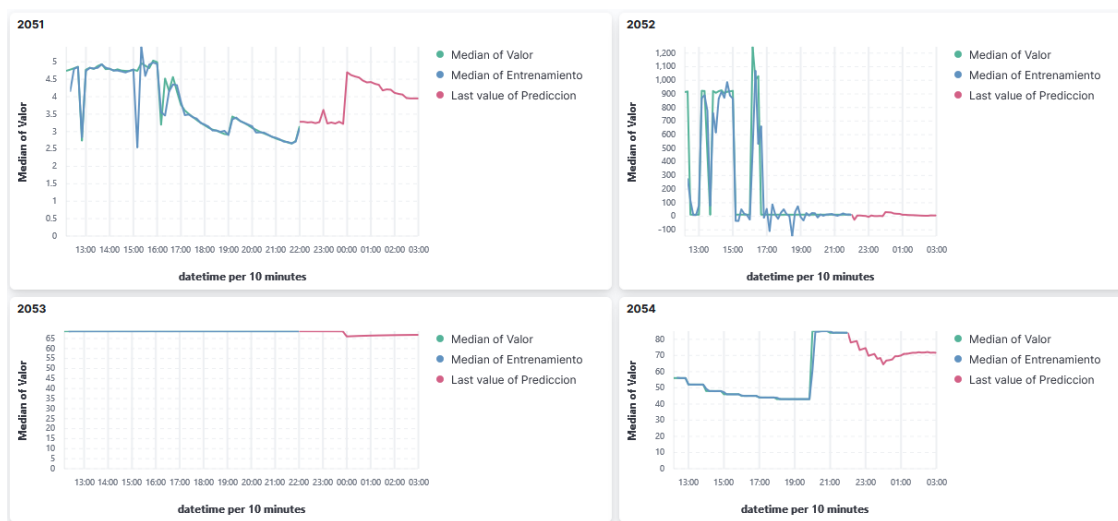


Figura 5.2: monitorización de sensores

Pruebas y resultados

Tras realizar las pruebas sobre los modelos se llegó a una desilusionante conclusión y es que sus predicciones no eran lo suficientemente buenas.

Debido a la poca experiencia y conocimientos en este campo no se pudo optimizar los modelos para que hicieran predicciones aceptables.

Facilitar la instalación

Para terminar, debido a la compleja configuración de todos los sistemas involucrados se decidió realizar un instalador que facilitará al usuario poner en marcha el programa.

5.3. Problemas y resoluciones

A continuación, se mostrarán una serie de problemas que se fueron presentando durante todo el desarrollo del proyecto, así como la solución a la que se llegó.

- El primer inconveniente que se presentó fue a la hora de descargar los datos de PRTG ya que no permite descargar datos en intervalos menores a media hora. Para solucionar este problema y poder descargar los datos de forma más continua, se decidió descargar desde PRTG en intervalos de media hora pero realizar esta acción con una periodicidad de un minuto, de esta forma la primera vez que se ejecuta el programa se suben los datos descargados en su totalidad pero en los ciclos posteriores el nuevo conjunto de datos descargados de PRTG (con un rango de media hora) es comparado con el del ciclo anterior eliminando los datos repetidos, de esta forma se logra sacar los datos correspondientes al último minuto.
- Al descargarse los datos desde PRTG estos se encontraban con un formato difícil de procesar por logstash (ya que este está pensado principalmente para trabajar con logs). Por ello se decidió realizar un sencillo programa en flex que transformara las líneas de datos en un formato legible para logstash.

Tras unos meses, se vio necesario realizar un tratamiento del campo fecha, Las horas descargadas por PRTG vienen en un formato H:mm:ss mientras que Logstash lee las horas con un formato HH:mm:ss por lo que al encontrarse horas de un solo dígito logstash no lo reconocía y saltaba una excepción.

se utilizó flex en un primer lugar por su sencillez, pero tras esta nueva necesidad se vio necesario migrar esta funcionalidad a Python y prescindir del programa realizado en flex.

- Elasticsearch almacena las variables de tiempo en formato UTC pero a la hora de visualizar los datos en kibana se utiliza por defecto la zona horaria del navegador.

Una forma de solucionarlo es configurar Kibana para que cambie la zona horaria a UTC.

- PRTG solo recoge los datos de los sensores cuando este se encuentre en un equipo encendido y con acceso a internet, si no, los datos procedentes

de los sensores se pierden, haciendo que nunca se guardarán datos de varios días seguidos.

Para solucionar esto se debería migrar el sistema a un servidor que se mantenga funcional siempre.

Trabajos relacionados

En este apartado se mostrarán algunas aplicaciones y trabajos similares relacionados con la monitorización, análisis y predicción de datos procedentes de tecnologías del internet de las cosas.

6.1. Elasticsearch Machine Learning

Elasticsearch, la herramienta que se utiliza como motor de monitorización y base de datos para nuestro proyecto cuenta con una herramienta que permite realizar predicciones y detección de anomalías, sobre los datos almacenados, sin embargo, esta funcionalidad es de pago.[\[10\]](#)

6.2. SkySpark

SkySpark es una plataforma de análisis de datos procedentes de dispositivos inteligentes IOT desarrollada por la compañía *SkyFoundry*, este software está diseñado para la identificación de fallos, anomalías y tendencias sobre los datos almacenados. Se encuentra disponible en cualquier plataforma que soporte java, como pueden ser los sistemas operativos de Windows, Linux o macOS.

Esta plataforma es de pago, aunque ofrece de una demo gratuita. [\[28\]](#)

6.3. Datadog IoT Monitoring

Datadog es una herramienta de monitorización y análisis de tecnologías IOT para aplicaciones en la nube enfocada a empresas. También consta

de herramientas de predicción que ayudan al reconocimiento de patrones y seguimiento de tendencias en los datos.

Se encuentra disponible en los sistemas operativos Windows, Linux y Mac, es una herramienta de pago, aunque cuenta de un periodo de prueba durante 14 días sin limitación en el número de sistemas que se desee monitorizar.[8]

6.4. AWS IoT Device Management

AWS permite la conexión de miles de dispositivos IOT a la nube de AWS, facilitando su seguimiento, monitorización y administración de forma remota. Este servicio Esta desarrollado por la compañía Amazon. [2]

Conclusiones y Líneas de trabajo futuras

Para concluir, en este último apartado se compartirán las conclusiones a las que se ha llegado tras la realización de este proyecto, así como unas líneas de trabajo futuras que servirán para mejorar el proyecto.

7.1. Conclusión

Tras la finalización del proyecto se entrega un producto totalmente funcional, capaz de monitorizar diversos sensores y lograr realizar predicciones a futuro, pese a que las predicciones no sean tan fiables como cabría de esperar, este es sin duda un apartado en el que habría que mejorar, modificar los algoritmos o buscar nuevas soluciones que pudieran realizar predicciones fiables.

Durante el transcurso del proyecto no solo se han logrado los objetivos generales sino los personales también, el realizar un proyecto como este para lo bueno y para lo malo ha sido una experiencia enriquecedora tanto profesionalmente como personalmente.

7.2. Líneas de trabajo futuras

Esta aplicación se podría mejorar con la implementación nuevas funcionalidades, así como la mejora de las ya existentes.

A continuación, se listan algunas de ellas:

- Mover el sistema a un servidor real, de esta manera se podrá captar los datos de los sensores de forma continuada y sin pérdidas.
- Implementar una interfaz de usuario para facilitar la experiencia del usuario. Mediante dicha interfaz se propone la implementación de los siguientes apartados:
 - Añadir y borrar sensores de manera gráfica.
 - Poder crear modelos para los sensores de forma intuitiva.
- Guardar metadatos de los sensores que se vayan a introducir en Elasticsearch como pueden ser el nombre del sensor, localización, etc.
- Mejorar los algoritmos de aprendizaje incremental así cómo añadir nuevos modelos capaces de realizar mejores predicciones.

Bibliografía

- [1] Algolia. Algolia. <https://www.algolia.com/>, 2021.
- [2] Amazon. AWS IoT Device Management. <https://aws.amazon.com/es/iot-device-management/?nc=sn&loc=2&dn=4>, 2021.
- [3] Amazon. ¿Qué es NoSQL? <https://aws.amazon.com/es/nosql/>, 2021.
- [4] Amazon. ¿Qué son los datos de streaming? <https://aws.amazon.com/es/streaming-data/>, 2021.
- [5] Apache Software Foundation. Apache Solr. <https://lucene.apache.org/solr/>, 2021.
- [6] Arch Linux. Daemons (Español). [https://wiki.archlinux.org/title/Daemons_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/title/Daemons_(Espa%C3%B1ol)), 2021.
- [7] Claudia Salazar. Tipos de sensores y sus aplicaciones. <https://www.industriasgsl.com/blog/post/tipos-de-sensores-y-sus-aplicaciones>, 2020.
- [8] Datadog. Datadog: Modern monitoring & security. <https://www.datadoghq.com/>, 2021.
- [9] DB-Engines. DB-Engines Ranking of Search Engines. <https://db-engines.com/en/ranking/search+engine>, 2020.
- [10] Elasticsearch. Elasticsearch. <https://www.elastic.co/es/>, 2021.
- [11] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2016.

- [12] Grafana. Grafana. <https://grafana.com/>, 2021.
- [13] Grant Maloy Smith. ¿Qué es un Sensor y Qué Hace? <https://dewesoft.com/es/daq/que-es-un-sensor#current-transducer>, 2020.
- [14] Gustavo B. ¿Qué es un comando Curl y cómo usarlo? <https://www.hostinger.es/tutoriales/comando-curl>, 2020.
- [15] IBM Cloud Education. Machine Learning. <https://www.ibm.com/cloud/learn/machine-learning>, 2020.
- [16] Jonathan Johnson. Anomaly Detection with Machine Learning: An Introduction. <https://www.bmc.com/blogs/machine-learning-anomaly-detection/>, 2020.
- [17] json.org. Introducción a JSON. <https://www.json.org/json-es.html>, 2021.
- [18] Marco Peixeriro. The Complete Guide to Time Series Analysis and Forecasting. <https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775>, 2019.
- [19] Margaret Rouse. Inteligencia artificial o IA. <https://searchdatacenter.techtarget.com/es/definicion/Inteligencia-artificial-o-AI>, 2019.
- [20] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdessalem. Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5, 2018.
- [21] Oracle. ¿Qué es el IoT? <https://www.oracle.com/es/internet-of-things/what-is-iot/>, 2020.
- [22] Oracle España. ¿Qué es el big data? <https://www.oracle.com/es/big-data/what-is-big-data/>, 2021.
- [23] PAESSLER. PRTG Network Monitor. <https://www.es.paessler.com/prtg>, 2021.
- [24] Proyecto Jupyter. Jupyter. <https://jupyter.org/index.html>, 2021.
- [25] Python. General Python FAQ ¿Que es Python? <https://docs.python.org/3.11/faq/general.html#what-is-python>, 2021.
- [26] River. River. <https://riverml.xyz/latest/>, 2021.

- [27] scikit-learn. scikit-learn Machine Learning in Python. <https://scikit-learn.org/stable/>, 2021.
- [28] SkyFoundry. SkySpark. <https://skyfoundry.com/product>, 2021.
- [29] Splunk. Splunk: The data-to.Everything Platform. <https://www.splunk.com/>, 2021.
- [30] ThingSpeak. ThingSpeak for IoT Projects. <https://thingspeak.com/>, 2021.
- [31] Wikipedia. Pandas (software). [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software)), 2021.