



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 — 15 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 de nombre departamento, área de nombre área.

Expone:

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

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, 15 de septiembre de 2021

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

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. Daemons	8
Técnicas y herramientas	9
4.1. Stream Processing Framework	9
4.2. Visualización y análisis de datos	10
4.3. Motor de búsquedas	11
4.4. Control de versiones	12
4.5. Lenguajes de programación	12
4.6. librerías Python	13
4.7. Archivos de texto	14
4.8. Sistemas Operativos	14
4.9. IDEs	14
4.10. Virtualización	15
4.11. Procesadores de lenguaje	15
4.12. Comunicación	15

4.13. Cliente SSH	15
4.14. ELK Stack	15
Aspectos relevantes del desarrollo del proyecto	17
5.1. Motivación en la elección del proyecto	17
5.2. Desarrollo del proyecto	17
5.3. Problemas y resoluciones	19
Trabajos relacionados	21
6.1. Elasticsearch	21
6.2. SkySpark	21
6.3. Datadog IoT Monitoring	21
6.4. AWS IoT Device Management	22
Conclusiones y Líneas de trabajo futuras	23
7.1. Conclusión	23
7.2. Líneas de trabajo futuras	23
Bibliografía	25

Índice de figuras

Índice de tablas

Introducción

El Internet de las Cosas es un campo que cada vez avanza más rápido, vivimos en un mundo en el que todo está conectado, por ello el tratamiento de los datos procedentes de todos estos sensores es primordial 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 diversos dispositivos IOT y lograr mediante machine learning la realización de un modelo capaz de aprender de dichos dispositivos y poder lograr una predicción en el futuro, con el objetivo de ver tendencias y detectar fallos o anomalías en los datos procedentes de los sensores 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, cómo ha sido diseñado 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.

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. Realizar un algoritmo de aprendizaje automático que pueda predecir comportamientos y patrones en los datos de los 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.

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. Reforzar mis conocimientos en el campo de la inteligencia artificial.

Conceptos teóricos

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

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.^[17]

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

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. ^[9]

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.

Data stream

Data stream son series de datos ordenados por tiempo que se generan de forma continua y desde diversas fuentes.

Time series

Time series o serie temporal es una sucesión de datos ordenados de forma cronológica.

A la hora de realizar predicciones el tiempo es normalmente la variable independiente y el objetivo es realizar predicciones a futuro [19]

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 cómo 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.[22]

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. [7]

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.

En el aprendizaje incremental no tenemos acceso a todo el conjunto de datos cuando creamos el modelo, sino que tenemos 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.

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 de tipo JSON (debido a que para los desarrolladores es un modelo de datos eficiente e intuitivo).^[3]

3.1. 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*^[16]

Técnicas y herramientas

4.1. Stream Processing Framework

Stream processing es una tecnología que procesa los datos de forma continua y secuencial, para realizar esta tarea se utilizan flujos de datos infinitos y sin límites de tiempo. Son extremadamente útiles a la hora de monitorizar sistemas, redes, aplicaciones, dispositivos, etc...

algunas de las alternativas más populares que he considerado son las siguientes:

Apache Kafka

Apache Kafka es un sistema de intermediación de mensajes que responde al patrón "Publish/Subscribe Messaging" que se utiliza para la comunicación entre aplicaciones. Entre sus principales características se encuentran que es un sistema escalable y persistente, con gran tolerancia a fallos y gran velocidad tanto de escritura como de lectura.

Para transmitir estos datos Kafka crea *Topics*, flujos de datos que a su vez se dividen en particiones, cada mensaje se almacena en una de estas particiones.

Debido a su popularidad es una herramienta bien documentada [15] tanto por la propia compañía como los usuarios. [14]

Apache Airflow

Apache Airflow es una plataforma open-source que nos permite automatizar tareas con la ayuda de scripts y mediante el uso de un planificador

para llevar a cabo numerosas tareas. Permite la utilización de DAGs (Grafo acíclico dirigido) y crear pipelines dinámicos, mediante Python.

Apache Airflow tiene una curva de aprendizaje elevada y puede resultar complicado para los usuarios[23]

Apache Samza

Es un procesador de flujo a tiempo real, usa Apache Kafka para la mensajería y Apache Hadoop YARN para tolerancia a fallos, seguridad, independencia de procesos y gestión de recursos para su utilización es necesario Hadoop. Samza utiliza streams inmutables e implementa el procesamiento de flujo sin y con estado. [24]

Apache Flume

Apache Flume es un servicio distribuido capaz de mover grandes cantidades de datos y logs, este forma parte del ecosistema de Hadoop. Su arquitectura es tanto flexible como sencilla. aunque presenta cierta dificultad a la hora de escalar. Está basado en flujo de datos en Streaming permitiendo múltiples flujos.[10]

4.2. 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 como alertarnos si alguno de los datos monitorizados pasa por debajo de un umbral.

PRTG Network Monitor

PRTG Network Monitor es una herramienta de monitorización de Paessler, nos 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. Nos permite mostrar alertas al encontrar problemas o métricas inusuales y estas nos pueden llegar de diversas formas.[18]

ThingSpeak

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

ThingSpeak consta de aplicaciones complementarias con diversas funcionalidades. [28]

Grafana

Grafana nos 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 deseemos y mandarnos notificaciones de las mismas, también cuenta con cientos de plugins oficiales.

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

Splunk

Splunk es una plataforma que nos permite monitorizar, buscar y analizar nuestros datos en tiempo real, esta herramienta consta con cientos de aplicaciones y es compatible con casi cualquier formato de logs. Splunk es gratis con un volumen de datos diarios de menos de 500MB. [27]

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

Debido a que la empresa que nos iba a facilitar los datos de los sensores utiliza PRTG he optado por esta opción.

4.3. 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[6], de código abierto que proporciona análisis y búsqueda de datos a tiempo real. Está desarrollado en Java y sólo soporta JSON como 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.[8]

Solr

Apache Solr es un motor de búsqueda open source escrito en Java con comandos escritos en HTTP y guarda los archivos utilizando XML. Está optimizado para tráfico de altos volúmenes de datos y todo en tiempo real.[11]

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

4.4. Control de versiones

Herramientas consideradas: GitHub y GitLab.

Para organizar el proyecto he elegido la plataforma de control de versiones de GitHub debido a ser el software de versiones Git más utilizado y con el cual tengo más experiencia.

4.5. Lenguajes de programación

Python

Python es un lenguaje de programación multiplataforma y multiparadigma, soporta tanto programación orientada a objetos como programación imperativa. Es un lenguaje cuya sintaxis se centra en un código legible y sintaxis clara. [20]

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.

4.6. librerías Python

Scikit-learn

Scikit-learn es un módulo de programación en python para machine learning, nos 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].[\[4\]](#)

scikit-multiflow

Scikit-multiflow es una librería de machine learning para python dedicada a streaming data y multi-output learning que nos permite generar y evaluar data streams.[\[25\]](#)

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.[\[29\]](#)

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.[21]

4.7. 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.[13]

4.8. Sistemas Operativos

Windows 10

Debido a que en la empresa que nos iba a facilitar los datos utilizan PRTG para monitorizar los sensores y este es exclusivo de Windows no tenía más alternativa que utilizar este sistema operativo para albergar mi servidor PRTG.

Ubuntu Server 20.04.2

Para albergar tanto Elasticsearch como el he escogido la última versión de ubuntu server debido a que es muy ligera

4.9. IDEs

L^AT_EX

Herramientas consideradas: Overleaf y Textmaker

La herramienta que he decidido utilizar para redactar la memoria ha sido Overleaf dividido a que es un 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 he decidido utilizar vim debido a su gran versatilidad y ligereza.

4.10. Virtualización

VirtualBox

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

4.11. Procesadores de lenguaje

Herramientas consideradas: Flex, Bison, JavaCC

Debido a la baja complejidad y el poco uso que le iba a dar a esta herramienta he decidido utilizar Flex, un analizador de lenguaje open source.

4.12. Comunicación

Microsoft Teams

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

4.13. Cliente SSH

PuTTY

PuTTY es un cliente SSH y telnet con licencia libre.

Este software nos permitirá conectarnos de forma remota a nuestro Ubuntu server así cómo intercambiar ficheros.

4.14. ELK Stack

ELK es la combinación de Elasticsearch, Logstash y Kibana,

Elasticsearch

Cómo se comentado 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 donde el cliente tiene albergados los sensores, PRTG y el servidor Ubuntu , el cual contenía 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 ser indexados correctamente en Elasticsearch.

Una vez conseguida la conexión entre los dos sistemas y logrado el mantener 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 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.

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 cambiar 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

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 sobre los datos almacenados, sin embargo, esta funcionalidad es de pago.

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 consta de una demo gratuita. [\[26\]](#)

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.[5]

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.

Durante el transcurso del proyecto no solo se han logrado los objetivos generales sino los personales también, el realizar un proyecto cómo 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 como pueden ser el nombre del sensor, localización, etc.
- Mejorar los algoritmos de aprendizaje incremental así cómo añadir nuevos modelos.

Bibliografía

- [1] Algolia. Algolia, 2020.
- [2] Amazon. Aws iot device management, 2021.
- [3] Amazon. ¿qué es nosql?, 2021.
- [4] David Cournapeau. scikit-learn machine learning in python, 2020.
- [5] Datadog. Datadog: Modern monitoring & security, 2021.
- [6] DB-Engines. Db-engines ranking of search engines, 2020.
- [7] IBM Cloud Education. Machine learning, 2020.
- [8] ElasticSearch. Elasticsearch, 2020.
- [9] Oracle España. ¿qué es el big data?, 2021.
- [10] Apache Flume. Apache flume, 2020.
- [11] Apache Software Foundation. Apache solr, 2020.
- [12] Grafana. Grafana, 2020.
- [13] json.org. Introducción a json, 2020.
- [14] Apache Kafka. Apache kafka a distributed streaming plataform, 2020.
- [15] Apache Kafka. Kafka 2.7 documentation, 2020.
- [16] Arch Linux. Daemons (español), 2021.
- [17] Oracle. ¿qué es el iot?, 2020.

- [18] PAESSLER. Prtg network monitor, 2020.
- [19] Marco Peixeriro. The complete guide to time series analysis and forecasting, 2019.
- [20] Python. General python faq ¿que es python?, 2021.
- [21] River. River, 2020.
- [22] Margaret Rouse. Inteligencia artificial o ia, 2019.
- [23] Nicholas Samuel. Apache kafka vs airflow: A comprehensive guide, 2020.
- [24] Apache Samza. Apache samza a distributed streaming processing framework, 2020.
- [25] The scikit-mutliflow development team and the open research community. scikit-mutliflow a machine learning package for streaming data in python, 2020.
- [26] SkyFoundry. Skyspark, 2021.
- [27] Splunk. Splunk: The data-to.everything platform, 2020.
- [28] ThingSpeak. Thingspeak for iot projects, 2020.
- [29] Wikipedia. Pandas (software), 2021.