



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**
título del TFG



Presentado por Nombre del alumno
en Universidad de Burgos — 7 de febrero
de 2021

Tutor: nombre tutor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, 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, 7 de febrero 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
Objetivos del proyecto	3
Conceptos teóricos	5
3.1. Secciones	5
3.2. Referencias	5
3.3. Imágenes	6
3.4. Listas de ítems	6
3.5. Tablas	7
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. librerías	12
Aspectos relevantes del desarrollo del proyecto	13
Trabajos relacionados	15
Conclusiones y Líneas de trabajo futuras	17

Bibliografía

19

Índice de figuras

3.1. Autómata para una expresión vacía	6
--	---

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	8
---	---

Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

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 y a tiempo real.

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 felicitar 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 mediante tecnologías como Stream Processing Frameworks.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de L^AT_EX¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [16]. Para citar webs, artículos o libros [10].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

▪

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

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 [9] tanto por la propia compañía como los usuarios. [8]

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 dinámicas, mediante Python.

Apache Airflow tiene una curva de aprendizaje elevada y puede resultar complicado para los usuarios[12]

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. [13]

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. Esta basado en flujo de datos en Streaming permitiendo múltiples flujos.[5]

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 pasan 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 esta 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.[11]

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. [15]

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.[7]

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 motor de búsqueda mas valorados del mercado[3] , 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.[4]

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 trafico de altos volúmenes de datos y todo en tiempo real.[6]

4.4. librerías

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 cómo NumPy, SciPy, pandas, SymPy, Matplotlib, Ip[y].[2]

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.[14]

Aspectos relevantes del desarrollo del proyecto

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros³, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Algolia. Algolia, 2020.
- [2] David Cournapeau. scikit-learn machine learning in python, 2020.
- [3] DB-Engines. Db-engines ranking of search engines, 2020.
- [4] ElasticSearch. Elasticsearch, 2020.
- [5] Apache Flume. Apache flume, 2020.
- [6] Apache Software Foundation. Apache solr, 2020.
- [7] Grafana. Grafana, 2020.
- [8] Apache Kafka. Apache kafka a distributed streaming plataform, 2020.
- [9] Apache Kafka. Kafka 2.7 documentation, 2020.
- [10] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [11] PAESSLER. Prtg network monitor, 2020.
- [12] Nicholas Samuel. Apache kafka vs airflow: A comprehensive guide, 2020.
- [13] Apache Samza. Apache samza a distributed streaming processing framework, 2020.
- [14] The scikit-mutliflow development team and the open research community. scikit-mutliflow a machine learning package for streaming data in python, 2020.

- [15] ThingSpeak. Thingspeak for iot projects, 2020.
- [16] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].