



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

**TECNOLOGÍA ESPECÍFICA DE
COMPUTACIÓN**

TRABAJO FIN DE GRADO

Sense:
**Plataforma de alto rendimiento para captura y
procesado de eventos IoT**

David Martín García

Julio, 2017

SENSE:
PLATAFORMA DE ALTO RENDIMIENTO PARA CAPTURA Y PROCESADO DE
EVENTOS IoT



UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

Tecnologías y Sistemas de Información

**TECNOLOGÍA ESPECÍFICA DE
COMPUTACIÓN**

TRABAJO FIN DE GRADO

Sense:

**Plataforma de alto rendimiento para captura y
procesado de eventos IoT**

Autor: David Martín García

Director: David Villa Alises

Director: María José Santofimia Romero

Julio, 2017

David Martín García

Ciudad Real – Spain

E-mail: davidmartingarcia0@gmail.com

Teléfono: 601 407 688

© 2017 David Martín García

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Se permite la copia, distribución y/o modificación de este documento bajo los términos de la Licencia de Documentación Libre GNU, versión 1.3 o cualquier versión posterior publicada por la *Free Software Foundation*; sin secciones invariantes. Una copia de esta licencia esta incluida en el apéndice titulado «GNU Free Documentation License».

Muchos de los nombres usados por las compañías para diferenciar sus productos y servicios son reclamados como marcas registradas. Allí donde estos nombres aparezcan en este documento, y cuando el autor haya sido informado de esas marcas registradas, los nombres estarán escritos en mayúsculas o como nombres propios.

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

El presente documento es un ejemplo de memoria del Trabajo de Fin Grado según el formato y criterios de la Escuela Superior de Informática de Ciudad Real. La intención es que este texto sirva además como una serie de consejos sobre tipografía, \LaTeX , redacción y estructura de la memoria que podrían resultar de ayuda. Por este motivo, se aconseja al lector consultar también el código fuente de este documento.

Este documento utiliza la clase \LaTeX *esi-tfg*, disponible como paquete Debian/Ubuntu, consulta:

<https://bitbucket.org/arco.group/esi-tfg>.

Si encuentra cualquier error o tiene alguna sugerencia, por favor, utilice el *issue tracker* del proyecto *esi-tfg* en:

<https://bitbucket.org/arco.group/esi-tfg/issues>

El resumen debería estar formado por dos o tres párrafos resaltando lo más destacable del documento. No es una introducción al problema, es decir, debería incluir los logros más importantes del proyecto. Suele ser más sencillo escribirlo cuando la memoria está prácticamente terminada. Debería caber en esta página (es decir, esta cara).

Abstract

English version of the previous page.

Agradecimientos

Escribe aquí algunos chascarrillos simpáticos. Haz buen uso de todos tus recursos literarios porque probablemente será la única página que lean tus amigos y familiares. Debería caber en esta página (esta cara de la hoja).

Juan¹

¹Sí, los agradecimientos se firman

A alguien querido y/o respetado

Índice general

Resumen	V
Abstract	VII
Agradecimientos	IX
Índice general	XIII
Índice de cuadros	XVII
Índice de figuras	XIX
Índice de listados	XXI
Listado de acrónimos	XXIII
1. Introducción	1
2. Objetivos	3
2.1. Objetivo general	3
2.2. Objetivos específicos	3
2.2.1. Análisis y selección del conjunto de elementos del proyecto:	3
2.2.2. Identificación y análisis de los datos ligados al internet de las cosas (IoT):	3
2.2.3. Crear un entorno para desarrolladores:	4
2.2.4. Desarrollo de punto de entrada de información:	4
2.2.5. Desarrollo de la aplicación web:	4
2.2.6. Despliegue y configuración de la infraestructura básica:	4
2.2.7. Creación y gestión de la documentación:	4
2.2.8. Monitorización de la plataforma:	5
2.2.9. Integración continua y despliegue automático:	5
2.2.10. Pruebas de estrés:	5

2.2.11. Búsqueda de partners:	5
3. Antecedentes	7
3.1. Sensorización: Definición.	7
3.1.1. Internet of Things: Definición y estado.	7
3.1.2. Hub de datos: Definición, características principales, y su estado . .	7
3.1.3. Ciudades Inteligentes: Estructura y servicios.	7
3.2. Protocolos de comunicación: Tipos y principales características	7
3.2.1. Interfaces de programación de aplicaciones: Definición y principales usos.	7
3.2.2. Transferencia de Estado Representacional: Definición, característi- cas y principales detalles en su arquitectura.	7
3.3. Minería de datos: Procesos de extracción de conocimiento y sus distintos tipos.	7
3.3.1. Análisis de resultados: Analisis masivo de tendencias y detección de valores anormales.	7
3.4. Análisis de servicios a utilizar	7
3.4.1. Lenguaje y frameworks web	7
3.4.2. Bases de datos	7
3.5. Aprovisionamiento de la plataforma	7
3.5.1. Entorno para desarrolladores	7
3.5.2. Entorno de pruebas	7
3.5.3. Entorno de producción	7
4. Método y fases de trabajo	9
4.1. Metodología de trabajo.	9
4.2. Sprints del desarrollo.	9
4.2.1. Sprint 0: Evaluación y elección de herramientas.	9
4.2.2. Sprint 1: Definición del modelo de datos y sus interfaces	9
4.2.3. Sprint 2: Creación del entorno para desarrolladores y PMV.	9
4.2.4. Sprint 3: Configuración de la integración Continua y Despliegue Continuo.	9
4.2.5. Sprint 4: Desarrollo de la interfaz de gestión.	9
4.2.6. Sprint 5: Algoritmos de análisis de datos.	9
4.2.7. Sprint 6: Gestion de integraciones con terceros.	9
5. Arquitectura de la aplicación	11
5.1. Diagrama de comunicaciones	11

6. Resultados	13
6.1. Caso de estudio I: Rendimiento en condiciones de carga optimas.	13
6.2. Caso de estudio II: Rendimiento en condiciones de carga superiores a las soportadas	13
6.3. Caso de estudio III: Analisis de los resultados obtenidos por el analizador de datos	13
7. Trabajos futuros	15
8. Conclusiones	17
Un anexo de ejemplo	21
Referencias	23

Índice de cuadros

Índice de figuras

Índice de listados

Listado de acrónimos

IoT	Internet of Things
API	Application Programming Interface
REST	Representational State Transfer
M2M	Machine to Machine

Capítulo 1

Introducción

EL ser humano siempre ha tenido la curiosidad de explorar su entorno, conocerlo más de cerca y analizar los sucesos que ocurren en este, se podría decir que se ve atraído por lo que no conoce, e intenta clasificarlo y sacar conclusiones, en el pasado, todo este proceso de obtención de información se realizaba de manera manual, pero por suerte, a día de hoy tenemos elementos que se encargan de esta labor, son los sensores, que nos permiten obtener datos de una forma programática para procesarlos posteriormente.

En los últimos años podemos presenciar un creciente interés en la temática relacionada con la sensorización, existe una gran cantidad de comunidad en torno a esta y esta surgiendo infinidad de plataformas hardware que ven en la sensorización su mayor fuerza, como hemos dicho anteriormente, estos dispositivos tienen la capacidad de ofrecernos información del entorno, pudiendo esta servir en infinidad de casos, como por ejemplo:

- Análisis del clima
- Detección temprana de catástrofes
- Gestión de una cadena de montaje.
- Monitorización de los niveles de radiación en una central nuclear

Todos estos sensores son cada vez mas comunes en nuestro entorno, tanto que, ciudades enteras se nutren de estos para la toma de decisiones críticas, un ejemplo de esto es la capital de España, Madrid, cuyos sensores de polución han causado más de una noticia en televisión, con los datos recogidos por estos sensores se deciden temas tan importantes que afectan a la salud de estos ciudadanos.

Toda esta tendencia no hace nada más que crecer, la cantidad de sensores se ve incrementada de forma exponencial, tanto que recientemente podemos haber escuchado hablar de conceptos como el Internet de las Cosas (en inglés Internet of Things o IoT), M2M (Machine to machine) o de las Smart Cities. Todos estos conceptos tienen en común la necesidad de la toma de datos por parte de sensores y la posterior comunicación para notificar del valor obtenido, todo esto hace que millones de lecturas tengan que ser manejadas de una forma eficiente y sobre todo a un bajo coste.

1. INTRODUCCIÓN

Por otra parte, toda esta tendencia ha venido acompañada de una serie de plataformas, estas tratan de solucionar los problemas de comunicación así como de almacenamiento hablados anteriormente, en concreto hemos podido ver como el número de plataformas ligadas al almacenamiento de datos relacionados con IoT ha crecido de forma también exponencial. La mayoría de estos solo te permiten, dentro del plan gratuito, unas cuotas de datos irrisorias para cualquier pequeño conjunto de sensores, teniendo que sacrificar resolución, además de tener una baja persistencia, ya que los datos son borrados pasado un tiempo. Una vez salimos del plan gratuito, los precios pueden llegar a dispararse.

El conjunto de todos estos detalles es lo que da forma a este proyecto, donde se busca obtener una solución eficiente, con el mínimo coste y con una máxima tasa de disponibilidad. Aunar todos estos requisitos es complicado, por lo que, se requerirá de un exhaustivo análisis de las diferentes opciones existentes eligiendo la mejor opción posible de todas las existentes dentro de cada categoría. Este proyecto llevará a cada tecnología involucrada a su máximo exponente, conceptos como el uso híbrido de bases de datos, que nos permiten seleccionar para cada momento donde almacenar la información en función del mejor tiempo de respuesta, tecnologías web bien probadas durante años en el campo de las telecomunicaciones formarán parte de nuestras herramientas para poner solución al problema de tener millones de sensores conectados.

Otro punto a destacar es el que todo lo desarrollado será software libre, donde cualquier persona podrá consultar como funciona, mejorarlo, o incluso, realizar su propia solución. Con todo esto se busca crear una nueva plataforma open-source, uno de los pocos casos en este campo, buscando poner un punto de partida donde, en un futuro, exista un proyecto colaborativo que se maneje por la propia comunidad.

Como punto final, el proyecto será planificado mediante metodologías ágiles, mediante la planificación de pequeños sprints, desarrollo guiado por test y integración continua, así como despliegue continuo, pudiendo controlar cada una de las fases del software de una forma exhaustiva.

Con todo esto, este proyecto pretende ser una alternativa respecto a las actuales plataformas de gestión de datos relacionados con la sensorización. «Sense» nace como un nuevo candidato en un mundo en el que, sus actuales competidores, no son lo suficiente eficientes para poder mantener los costes a un nivel bajo. Partimos con la ventaja de aprender de sus errores, y con la posibilidad de mejorar lo actualmente existente.

Capítulo 2

Objetivos

2.1 Objetivo general

Este proyecto tiene el objetivo de poner en marcha una aplicación web, para esto debemos llevar a cabo los principales procesos involucrados en la creación de cualquier servicio web, desde el análisis inicial hasta su puesta en marcha y conexión con el mundo, teniendo en cuenta el mantenimiento posterior a tal momento.

Se busca desarrollar una plataforma de gestión de sensorización dentro del ámbito de IoT con un rendimiento por encima de todo lo actualmente visto, teniendo tasas de tiempos de respuesta cercanos a la milésima de segundo en los puntos vitales, además de aguantar un gran número de conexiones concurrentes, esto hace que se aproveche al máximo los recursos de la máquina huésped, además crearé un proyecto open-source y un entorno favorable para que se genere una comunidad entorno a este.

Esta plataforma puede posicionarse como alternativa a otros servicios gratuitos, gracias al bajo coste derivado de su alto rendimiento. No existe algo similar en términos de simplicidad y rendimiento en este momento.

2.2 Objetivos específicos

Podemos destacar varios apartados, que serán piezas básicas para la realización de tal proyecto:

2.2.1 Análisis y selección del conjunto de elementos del proyecto:

Comparativa del rendimiento de los actuales frameworks web así como de la infraestructura necesaria para este proyecto. Dado que una de las máximas de este proyecto será un elevado rendimiento a un coste bajo, se seleccionará entre todos los candidatos el que mejor se adapte ante esta situación para cada una de las partes involucradas en este proyecto.

2.2.2 Identificación y análisis de los datos ligados al internet de las cosas (IoT):

Identificar posibles estándares ligados a este concepto, además de desarrollar la arquitectura básica que tomarán los datos en nuestra aplicación. Se requerirá analizar plataformas existentes, así como indagar sobre posibles estándares a la hora de enviar información.

2. OBJETIVOS

2.2.3 Crear un entorno para desarrolladores:

Este entorno debe facilitar a cualquier desarrollador el poder aterrizar en el proyecto con la mínima dificultad posible, debe ser totalmente estable y totalmente aislado de la máquina del desarrollador, siendo este una réplica exacta del encontrado en los servidores al que acceden los usuarios. Con esto conseguimos poder generar un entorno limpio, con todos y cada uno de los elementos involucrados así como un conjunto de datos mínimo con el que poder empezar a trabajar. Como hito, un desarrollador debe poder empezar a trabajar en este proyecto en menos de un cuarto de hora.

2.2.4 Desarrollo de punto de entrada de información:

Desarrollo de una API REST como punto de entrada y salida de información proveniente de sensores. Debe estar debidamente protegida para que usuarios no autorizados no puedan acceder a dichos recursos. Se debe maximizar el rendimiento de esta parte, dado que es crucial a la hora de evitar cuellos de botella y posibles problemas ante altas tasas de conexiones.

2.2.5 Desarrollo de la aplicación web:

Desarrollo de una aplicación web donde poder gestionar los datos recibidos por el usuario, tal aplicación deberá poder hacer búsquedas dentro de estos datos así como permitirnos un mínimo análisis.

2.2.6 Despliegue y configuración de la infraestructura básica:

Desplegar la infraestructura necesaria para hacerlo accesible a cualquier usuario, desde el dominio hasta los servidores donde se alojara la aplicación. Se intentara hacerlo del modo más transparente, dejando la posibilidad de, en un futuro, poder mejorar la infraestructura en caso de ser necesario.

2.2.7 Creación y gestión de la documentación:

Dado que esta aplicación será usada por personas externas al proyecto, debemos generar los documentos necesarios para informarles de cómo hacer uso de los distintos puntos de acceso a la aplicación. Esta documentación también incluye algún ejemplo con alguna plataforma hardware de sensorización. Por otro lado, deberemos disponer de una /acsAPI interna para el frontend, por lo que debemos tener información para sincronizar los cambios entre la aplicación y su interfaz web. Por otro lado, también será necesario documentar casos básicos dentro del lenguaje seleccionado, desde como crear cualquier tipo de elemento, hasta como generar un test automático para tal, se debe ilustrar con algún ejemplo interno y explicar cada una de las partes, dado que el desarrollador no tiene el deber de conocer el lenguaje de programación.

2.2.8 Monitorización de la plataforma:

Tener conocimiento del estado de toda la plataforma en cada momento, así como recibir alertas en caso de fallo ó bajo rendimiento. Esto es importante, dado que debemos anticiparnos a posibles picos de tráfico además de validar la integridad de los mismos. Siendo una aplicación para terceros, la disponibilidad es clave. Si no obtenemos una disponibilidad mínima, los usuarios rechazarán nuestra plataforma.

2.2.9 Integración continua y despliegue automático:

Siempre que un desarrollador termine un trabajo, este se comprobará mediante una serie de tests, si todos estos tests son válidos y obtienen un resultado favorable, se llevará a cabo un proceso automático de despliegue a los servidores. Con esto conseguimos facilitar tal proceso, así como aportar un control de calidad previo a mover estos cambios a cualquier entorno.

2.2.10 Pruebas de estrés:

Generar un modelo de pruebas de estrés que nos permita evaluar el resultado final del rendimiento de la plataforma así como su estabilidad y la corrección de cada uno de los puntos.

2.2.11 Búsqueda de partners:

Busqueda de socios tecnológicos con los que probar la aplicación una vez éste en el mercado, permitiendo un posicionamiento más rápido así como una prueba inicial de las cualidades del servicio en un entorno real.

Antecedentes

3.1 Sensorización: Definición.

3.1.1 Internet of Things: Definición y estado.

3.1.2 Hub de datos: Definición, características principales, y su estado

3.1.3 Ciudades Inteligentes: Estructura y servicios.

3.2 Protocolos de comunicación: Tipos y principales características

3.2.1 Interfaces de programación de aplicaciones: Definición y principales usos.

3.2.2 Transferencia de Estado Representacional: Definición, características y principales detalles en su arquitectura.

3.3 Minería de datos: Procesos de extracción de conocimiento y sus distintos tipos.

3.3.1 Análisis de resultados: Análisis masivo de tendencias y detección de valores anormales.

3.4 Análisis de servicios a utilizar

3.4.1 Lenguaje y frameworks web

3.4.2 Bases de datos

3.5 Aprovisionamiento de la plataforma

3.5.1 Entorno para desarrolladores

3.5.2 Entorno de pruebas

3.5.3 Entorno de producción

Capítulo 4

Método y fases de trabajo

4.1 Metodología de trabajo.

4.2 Sprints del desarrollo.

4.2.1 Sprint 0: Evaluación y elección de herramientas.

4.2.2 Sprint 1: Definición del modelo de datos y sus interfaces

4.2.3 Sprint 2: Creación del entorno para desarrolladores y PMV.

4.2.4 Sprint 3: Configuración de la integración Continua y Despliegue Continuo.

4.2.5 Sprint 4: Desarrollo de la interfaz de gestión.

4.2.6 Sprint 5: Algoritmos de análisis de datos.

4.2.7 Sprint 6: Gestion de integraciones con terceros.

Capítulo 5

Arquitectura de la aplicación

5.1 Diagrama de comunicaciones

Capítulo 6

Resultados

- 6.1 Caso de estudio I: Rendimiento en condiciones de carga optimas.**
- 6.2 Caso de estudio II: Rendimiento en condiciones de carga superiores a las soportadas**
- 6.3 Caso de estudio III: Analisis de los resultados obtenidos por el analizador de datos**

Capítulo 7

Trabajos futuros

Capítulo 8

Conclusiones

ANEXOS

Un anexo de ejemplo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Referencias

Este documento fue editado y tipografiado con \LaTeX empleando la clase **esi-tfg** (versión 0.20170731) que se puede encontrar en:
https://bitbucket.org/arco_group/esi-tfg

[respeta esta atribución al autor]

