



INGENIERÍA EN TECNOLOGÍAS DE LA  
TELECOMUNICACIÓN

Curso Académico 2017/2018

Trabajo Fin de Grado/Máster

SAKURA  
ANGULAR 4 - ELASTICSEARCH - DASHBOARD  
INTERFACE

Autor : Ismael Slimane Zubillaga

Tutor : Dr. Jesús M. González-Barahona



# Trabajo Fin de Grado/Máster

Título del Trabajo con Letras Capitales para Sustantivos y Adjetivos

**Autor :** Nombre del Alumno/a

**Tutor :** Dr. Gregorio Robles Martínez

La defensa del presente Proyecto Fin de Carrera se realizó el día            de  
de 20XX, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a            de            de 20XX



*Dedicado a  
mi familia / mi abuelo / mi abuela*



# Agradecimientos

Aquí vienen los agradecimientos... Aunque está bien acordarse de la pareja, no hay que olvidarse de dar las gracias a tu madre, que aunque a veces no lo parezca disfrutará tanto de tus logros como tú... Además, la pareja quizás no sea para siempre, pero tu madre sí.





# Resumen

Aquí viene un resumen del proyecto. Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Contents

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Descripción del Problema . . . . .	1
1.2	Objetivo Principal y Estructura . . . . .	1
1.3	Disponibilidad del Software . . . . .	1
<b>2</b>	<b>State of Art</b>	<b>3</b>
2.1	ElasticSearch . . . . .	3
2.1.1	History . . . . .	3
2.1.2	Main Concepts . . . . .	3
2.2	GitHub . . . . .	5
2.3	Webpack . . . . .	5
2.4	Angular 4 . . . . .	5
2.5	TypeScript . . . . .	5
2.6	JavaScript . . . . .	6
2.7	Lodash . . . . .	6
2.8	ChartJS . . . . .	6
2.9	Canvas . . . . .	6
2.10	SASS . . . . .	6
2.11	Flexbox . . . . .	6
<b>3</b>	<b>Desarrollo</b>	<b>7</b>
3.1	Metodología SCRUM . . . . .	7
3.2	Arquitectura general . . . . .	7
<b>4</b>	<b>Resultados</b>	<b>9</b>

<b>5 Conclusiones</b>	<b>11</b>
5.1 Consecución de objetivos . . . . .	11
5.2 Aplicación de lo aprendido . . . . .	11
5.3 Lecciones aprendidas . . . . .	11
5.4 Trabajos futuros . . . . .	12
<b>A Manual de usuario</b>	<b>13</b>
<b>Bibliography</b>	<b>15</b>

# List of Figures

3.1	Estructura del parser básico . . . . .	8
-----	--	---





# Chapter 1

## Introducción

En este capítulo se introduce el proyecto. Debería tener información general sobre el mismo, dando la información sobre el contexto en el que se ha desarrollado.

No te olvides de echarle un ojo a la página con los cinco errores de escritura más frecuentes<sup>1</sup>.

Aconsejo a todo el mundo que mire y se inspire en memorias pasadas. Las más están todas almacenadas en mi web del GSyC<sup>2</sup>.

### 1.1 Descripción del Problema

### 1.2 Objetivo Principal y Estructura

En esta sección se debería introducir la estructura de la memoria. Así:

- En el primer capítulo se hace una intro al proyecto.
- En el capítulo ?? (ojo, otra referencia automática) se muestran los objetivos del proyecto.
- A continuación se presenta el estado del arte.
- ...

### 1.3 Disponibilidad del Software

---

<sup>1</sup><http://www.tallerdeescritores.com/errores-de-escritura-frecuentes>

<sup>2</sup><https://gsyc.urjc.es/~grex/pfcs/>



# Chapter 2

## State of Art

### 2.1 ElasticSearch

*Elasticsearch* is an open-source, *broadly-distributable, readily-scalable, enterprise-grade* search engine based on Lucene<sup>1</sup>. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.

#### 2.1.1 History

**Shay Banon** created the precursor to Elasticsearch, called *Compass*, in 2004. While thinking about the third version of Compass he realized that it would be necessary to rewrite big parts of Compass to "create a scalable search solution". So he created "a solution built from the ground up to be distributed" and used a common interface, *JSON* over *HTTP*, suitable for programming languages other than *Java* as well. Shay Banon released the first version of Elasticsearch in February 2010.

#### 2.1.2 Main Concepts

All the following concepts definitions are extracted from the Elasticsearch documentation web page<sup>2</sup>.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Apache\\_Lucene](https://en.wikipedia.org/wiki/Apache_Lucene)

<sup>2</sup>[https://www.elastic.co/guide/en/elasticsearch/reference/current/\\_basic\\_concepts.html](https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html)

- **Near Realtime (NRT):** Elasticsearch is a near real time search platform. What this means is there is a slight latency (normally one second) from the time you index a document until the time it becomes searchable.
- **Index:** An index is a collection of documents that have somewhat similar characteristics. For example, you can have an index for customer data, another index for a product catalog, and yet another index for order data. An index is identified by a name (that must be all lowercase) and this name is used to refer to the index when performing indexing, search, update, and delete operations against the documents in it. In a single cluster, you can define as many indexes as you want.

- **Shards Replicas:** An index can potentially store a large amount of data that can exceed the hardware limits of a single node.

To solve this problem, Elasticsearch provides the ability to subdivide your index into multiple pieces called **shards**. When you create an index, you can simply define the number of shards that you want. Each shard is in itself a fully-functional and independent "index" that can be hosted on any node in the cluster.

In a network/cloud environment where failures can be expected anytime, it is very useful and highly recommended to have a failover mechanism in case a shard/node somehow goes offline or disappears for whatever reason. To this end, Elasticsearch allows you to make one or more copies of your index's shards into what are called *replica shards*, or **replicas** for short.

- **Type:** A type in Elasticsearch represents a class of similar documents. A type consists of a *name*—such as user or blogpost—and a *mapping*. The *mapping*, like a database schema, describes the fields or properties that documents of that type may have, the datatype of each field—such as string, integer, or date—and how those fields should be indexed and stored by *Lucene*.
- **Document:** A document is a basic unit of information that can be indexed. For example, you can have a document for a single customer, another document for a single product, and yet another for a single order. This document is expressed in JSON (JavaScript Object Notation) which is an ubiquitous internet data interchange format.

- **Node:** A node is a single server that is part of your cluster, stores your data, and participates in the cluster's indexing and search capabilities.
- **Cluster:** A cluster is a collection of one or more nodes (servers) that together holds your entire data and provides federated indexing and search capabilities across all nodes.

## 2.2 GitHub

Los objetivos específicos se pueden entender como las tareas en las que se ha desglosado el objetivo general. Y, sí, también vienen en infinitivo.

## 2.3 Webpack

## 2.4 Angular 4

## 2.5 TypeScript

A mí me gusta que aquí pongáis una descripción de lo que os ha llevado realizar el trabajo. Hay gente que añade un diagrama de GANTT. Lo importante es que quede claro cuánto tiempo llevas (tiempo natural, p.ej., 6 meses) y a qué nivel de esfuerzo (p.ej., principalmente los fines de semana).

## **2.6 JavaScript**

## **2.7 Lodash**

## **2.8 ChartJS**

## **2.9 Canvas**

## **2.10 SASS**

## **2.11 Flexbox**

# Chapter 3

## Desarrollo

Aquí viene todo lo que has hecho tú (tecnológicamente). Puedes entrar hasta el detalle. Es la parte más importante de la memoria, porque describe lo que has hecho tú. Eso sí, normalmente aconsejo no poner código, sino diagramas.

### 3.1 Metodología SCRUM

### 3.2 Arquitectura general

Si tu proyecto es un software, siempre es bueno poner la arquitectura (que es cómo se estructura tu programa a “vista de pájaro”).

Por ejemplo, puedes verlo en la figura 3.1.

Si utilizas una base de datos, no te olvides de incluir también un diagrama de entidad-relación.

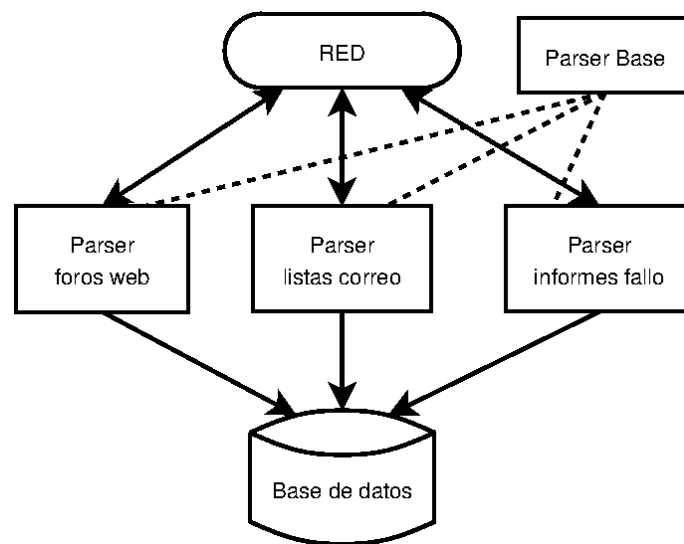


Figure 3.1: Estructura del parser básico



# **Chapter 4**

## **Resultados**

En este capítulo se incluyen los resultados de tu trabajo fin de grado.

Si es una herramienta de análisis lo que has realizado, aquí puedes poner ejemplos de haberla utilizado para que se vea su utilidad.



# Chapter 5

## Conclusiones

### 5.1 Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

### 5.2 Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM.

Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

### 5.3 Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. a

2. b

## **5.4 Trabajos futuros**

Ningún software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFM.

## **Apéndice A**

### **Manual de usuario**



# **Bibliography**