



Recuperación da Información e Web Semántica
Máster Universitario en Ingeniería Informática - MUEI
2020/2021

Sistema de recuperación de información mediante Twitter

Autora: Jennifer Dubra Rey

A Coruña, novembro de 2020.

Índice Xeral

1	Introducción	1
1.1	Motivación	1
1.2	Obxectivos	1
2	Conceptos	3
2.1	Twitter	3
3	Tecnoloxías, ferramentas e librerías	5
3.1	Linguaxe e tecnoloxías	5
3.1.1	Java	5
3.1.2	Spring Boot	6
3.2	Ferramentas	6
3.2.1	IntelliJ IDEA	6
3.2.2	Docker	6
3.2.3	Postman	6
3.2.4	Apache Maven	7
3.2.5	Git	7
3.2.6	Apache Tomcat	7
3.3	Librerías	8
3.3.1	Twitter4J	8
3.3.2	RestHighLevelClient	8
3.3.3	Thymeleaf	8
3.3.4	Leaflet	8
4	Configuración	9
4.1	Configuración dos entornos	9
4.1.1	ElasticSearch 7.9.2	9
4.1.2	Apache Tomcat 8.5.58	9

4.1.3	Docker 19.03.6	9
4.2	Configuración necesaria da aplicación	11
4.2.1	Twitter4J	11
4.2.2	RestHighLevelClient	12
5	Desenvolvemento	13
5.1	Análise de requisitos	13
5.1.1	Requisitos funcionais	13
5.2	Desenvolvemento	18
5.2.1	Recuperación de Tweets	18
5.2.2	Persistir Tweets en ElasticSearch	18
5.2.3	Procurar Tweets en ElasticSearch	19
5.2.4	Deseño da interfaz gráfica	19
6	Interfaz gráfica e resultados	21
6.0.1	Gardar Tweets en tempo real	21
6.1	Páxina principal	21
6.2	Procura por consulta	22
6.3	Procura por usuario	22
6.4	Procura por consulta e usuario	23
6.5	Visualización no mapa da procura por consulta e usuario	23
6.6	Visualización no mapa de todos os Tweets	24
6.6.1	Gardar Tweets en tempo real	24
7	Entregables	25

Índice de Figuras

5.1	Casos de uso do sistema.	17
6.1	Página principal.	21
6.2	Procura por consulta.	22
6.3	Procura por usuario.	22
6.4	Procura por usuario e consulta.	23
6.5	Procura por usuario e consulta visualizados en mapa.	23
6.6	Ver todos os Tweets en mapa.	24

Introducción

Neste capítulo explicaranse os obxectivos do traballo pertencente á asignatura de Representación de Información e Web Semántica.

1.1 Motivación

A motivación deste exercicio está presentada pola necesidade de representar a información dunha forma comprensible para o ser humano e de forma máis gráfica.

Búscase extrapolar datos que poidan ser significativos e indicadores para posibles investigacións.

1.2 Obxectivos

Os obxectivos desta práctica consisten en recuperar os Tweets da rede social Twitter para un posterior almacenamento nun servidor de Elasticsearch.

Os Tweets que se gardarán serán os recuperados en tempo real de forma similar á que se faría nun proxecto real. Os atributos gardados serán aqueles que conteñan información relevante e de posible interés para satisfacer os requisitos funcionais definidos, por exemplo: o contido, os datos do usuario, o lugar dende onde se publicaron, etc.

O software realizará as procuras no servidor de Elasticsearch mediante queries booleanas.

Considerarase entón, como obxectivos desta práctica, a capacidade de poder buscar, a través de unha interfaz gráfica, os Tweets indexados en función das variables "Consulta" e "Usuario". A variable "Consulta" será definida como os destintos termos que é imprescindible que conteñan os tweets que se recuperan. Da mesma maneira, a variable "Usuario" será o conxunto de palabras que os usuarios dos Tweets que se recuperan terán no seu nome.

O usuario/a do software poderá visualizar os resultados da súa procura (por "Consulta", "Usuario" ou ambos) nun mapa de calor interactivo que mostrará só aqueles que pertencen á

rexión do mapa que se visualiza.

Igualmente, o usuario/a deberá ser capaz de ver todos os documentos indexados en función da rexión do mapa que está visible.

A opción do mapa interactivo xunto coa búsqueda dos Tweets da rexión que se visualiza pode dar ao usuario unha idea de en qué zonas do mundo se comenta máis sobre certo tema. O cal pode ser relevante para futuras investigacións.

Capítulo 2

Conceptos

Describiranse neste apartado os conceptos que se manexarán durante o desenvolvemento do documento e do proxecto.

2.1 Twitter

Twitter¹ é unha rede social que está activa dende 2006 e con 328 millóns de usuarios activos en Setembro do 2019.

Esta rede social permite enviar mensaxes *Tweets* de texto plano de corta lonxitude, máximo 280 caracteres, que se mostran no perfil do usuario/a. As persoas usuarias pódense suscribir aos perfís para que se mostren no *feed* os últimos *Tweets* publicados.

Por defecto, estas mensaxes son públicas o que habilita a opción de *Retweet* que fai referencia a compartir unha aplicación entre os seguidores da persoa usuaria logueada. Os mesmos *Tweets* poden ser comentados e marcados como "me gusta".

Entre outras funcionalidades, Twitter tamén permite visualizar cales son as tendencias en tempo real nunha determinada zona xeográfica e enviar mensaxes privadas entre os diferentes usuarios/as.

¹<https://twitter.com/>

Tecnoloxías, ferramentas e librerías

Neste capítulo explicarase e xustificarase o uso das distintas tecnoloxías, ferramentas e librerías utilizadas na elaboración do traballo.

3.1 Linguaxe e tecnoloxías

Nesta sección entenderase a escolla do linguaxe de programación como as distintas tecnoloxías utilizadas.

3.1.1 Java

Java destaca polas características de ser seguro, rápido e fiable. É un dos linguaxes de programación máis populares particularmente para aquelas aplicacións ca estrutura de cliente-servidor.

É un linguaxe de programación orientado a obxectos onde a idea chave é deseñar o software de forma que os distintos tipos de datos que se usen estean unidos as súas operacións de tal forma que os datos e as súas funcións combinadas formen obxectos. Estes obxectos ofrecen unha base máis estable para o deseño de un sistema software pois grandes proxectos serán máis fáciles de xestionar e manexar, mellorando como consecuencia a súa calidade.

En adición do favorable que é para o proxecto un linguaxe de programación orientado a obxectos, escolleuse Java pola súa independencia da plataforma e polo posterior uso de Apache Maven para a xeración de proxectos que serán despregados no servidor de aplicacións Apache Tomcat. Java é un linguaxe de alto nivel que permite unha abstracción dos detalles menores dos códigos de baixo nivel. Este linguaxe cando se compila xera un código intermedio (bytecode) que logo será executado e interpretado nunha máquina virtual propia de Java (JVM) que identifica o hardware. Debido a isto, calquera programa en Java poderase executar igualmente en calquera tipo de hardware (tal e como predica o axioma deste linguaxe *"write once, run anywhere"*).

3.1.2 Spring Boot

Spring Boot é o framework de código aberto para Java encargado de facilitar o desenvolvemento de aplicacións web. Entre as vantaxes que aporta están a súa capacidade de autoconfiguración, unha xestión de dependencias sinxela e a non necesidade de instalar un servidor web pois xa prové, entre outros, un servidor Tomcat embebido.

É moi útil en este proxecto pois permite tamén dividir a estrutura deste segundo o patrón de arquitectura software MVC (Modelo Vista Controlador), que permite unha mellor xestión do control dos datos e a súa representación.

3.2 Ferramentas

A continuación detállanse as especificacións das ferramentas que se empregaron entendendo por ferramenta como un programa informático utilizado durante ou nalgunha das partes do ciclo de vida dun produto software ca intención de crealo, depuralo, xestionalo ou mantelo.

3.2.1 IntelliJ IDEA

IntelliJ é un software libre escrito en Java que foi desenvolto por JetBrains e que está dispoñible dende 2001.

Dispón, entre outras, de un editor de texto con un analizador sintáctico, compilación en tempo real e integración con distintos sistemas de control de versións como pode ser Git ou Subversion. Tamén contén probas unitarias a través de JUnit, un depurador de código e integración con Maven.

Neste proxecto usouse a edición Community.

3.2.2 Docker

Docker consiste nun software de código aberto que automatiza o despregue de aplicacións dentro de contenedores software. Usa recursos do kernel de Linux a través de máquinas virtuais, dando soporte aos sistemas operativos Windows, Linux e macOS.

Serve para levantar máquinas independentes con sistemas operativos lixeiros en moi pouco tempo.

3.2.3 Postman

Postman fundouse en 2014 é a ferramenta de software libre que permite realizar peticións web sobre APIs REST de forma simple.

Serve para levantar máquinas independentes con sistemas operativos lixeiros en moi pouco tempo.

3.2.4 Apache Maven

Apache Maven é un software de código aberto e multiplataforma que serve para simplificar os procesos de compilación e xeración de executables a partir de un código fonte cun modelo de configuración simple baseado nun formato XML. Fomenta a reutilización de código e librerías a través dos POMs (Project Object Models) que serven para describir o proxecto de software a construír e as dependencias con outros módulos e compoñentes externos. Tamén se poden incluír en el mecanismos como *plugins* customizables para facelo extensible.

As tarefas principais de Apache Maven consisten na compilación de código e o empaquetado deste e ocúpase incluso das tarefas de despregue. A maiores executa as probas e xera informes e documentación.

Emprégase Apache Maven neste proxecto pola súa comodidade á hora de estruturar o proxecto e de xestionar dependencias con outras librerías pois é el mesmo quen se encarga de descargalas grazas a súa cualidade de estar listo na rede. Tamén é Apache Maven o que crea o .war que será despregado no servidor de aplicacións.

3.2.5 Git

Git é a ferramenta escollida para levar a cabo un control de versións. Permite un aloxamento de repositorios e xestionar a liña de desenvolvemento destes.

O sistema de versións funciona ao levar un rexistro dos cambios realizados no na estrutura do proxecto. Ofrece tamén un apoio ao desenvolvemento non lineal con gran rapidez na xestión de ramas e a mestura de distintas versións.

Git conta dunha plataforma web con ferramentas específicas para navegar e visualizar o historial de desenvolvemento así como os propios arquivos en calquera dos seus estados rexistrados.

3.2.6 Apache Tomcat

Apache Tomcat é o software de código libre que proporciona un servidor HTTP web dispoñible para os distintos sistemas operativos que contén con unha máquina virtual Java (JVM). A principal función dun servidor web, neste caso Apache Tomcat, é almacenar, procesar e entregar ao software as peticións para que este poida responder en función das necesidades dos usuarios.

3.3 Librerías

Nesta sección indícanse cales foron as implementacións importadas no proxecto para os procesos de recuperación de información e indexación dos documentos recuperados.

3.3.1 Twitter4J

Twitter4J consiste nunha librería de código aberto que proporciona unha API en Java para acceder á API de Twitter.

Dende esta librería pódese: crear *tweets*, enviar e recibir mensaxes directas, buscar *tweets*, etc.

Asegura tamén a seguridade e privacidade do usuario, o cal é necesario para poder usar a API.

Escolleuse esta librería por ser a máis popular entre os usuarios desenvolvedores en Java e porque proporciona un *Listener* a través do cal se poden recuperar os *tweets* en tempo real.

3.3.2 RestHighLevelClient

RestHighLevelClient é a librería cliente de alto nivel que permite crear peticións e ler as respostas de Elasticsearch. O principal obxectivo desta API é expoñer métodos específicos, aceptar obxectos peticións e devolver a resposta tamén como un obxecto.

Os métodos poden ser executados de forma síncrona e asíncrona.

3.3.3 Thymeleaf

Thymeleaf é unha biblioteca adaptada para traballar coa arquitectura Modelo-Vista-Controlador, modelándose coa capa vista. Propón un motor de plantillas XML/ XHTML/ HTML5 moi útiles para traballar en entornos web, aínda que tamén son aplicables para entornos non web. Dentro das súas vantaxes está a característica de que é integrable con Springboot e Java EE.

3.3.4 Leaflet

Leaflet é a principal librería de código libre de JavaScript con mapas interactivos. Foi lanzada por primeira vez en 2011 e é compatible coa maioría de plataformas móbiles e de escritorio.

Leaflet admite customizacións con HTML5 e CSS3. Soporta de forma nativa capas *Web Map Service* (WMS), geoJSON, vectoriales e capas de mosaico. A maiores pódense engadir soporte a máis tipos de capas a través de plugins.

Configuración

4.1 Configuración dos entornos

4.1.1 Elasticsearch 7.9.2

Versión	7.9.2
Hosts	localhost
Portos	9200, 9300

Táboa 4.1: Características da configuración de Elasticsearch.

4.1.2 Apache Tomcat 8.5.58

Versión	8.5.58
Hosts	localhost
Portos	8080

Táboa 4.2: Características da configuración de Apache Tomcat.

4.1.3 Docker 19.03.6

A versión de Docker empregada neste traballo é a 19.03.6.

Para o levantamento do servidor de Elasticsearch mencionado anteriormente e do Apache Tomcat é necesario crear un arquivo *docker-compose.yml* co contido da Listaxe [4.1](#).

Unha vez gardado o *docker-compose.yml*, executarase dende o directorio do arquivo o comando da Listaxe [4.2](#).

O contido do arquivo *docker-compose.yml* configura na mesma rede o contenedor co servidor de ElasticSearch e o contenedor de Apache Tomcat para que poida haber unha comunicación entre eles.

Cabe destacar que, na configuración do contenedor de Tomcat mapéanse os arquivos do contenedor */usr/local/tomcat/tomcat* e */usr/local/tomcat/webapps* cos arquivos */tomcat* e */deploy* respectivamente na carpeta onde se executou o comando. Isto é interesante porque facilita os despregues da aplicación xa que ao copiar no directorio local */deploy* os *.war* desexados, estes son directamente despregados no contenedor de Tomcat asociado.

```
1 version: '3'
2 services:
3   es:
4     image: docker.elastic.co/elasticsearch/elasticsearch:7.9.2
5     container_name: elastic
6     environment:
7       - xpack.security.enabled=false
8       - xpack.monitoring.enabled=false
9       - xpack.ml.enabled=false
10      - xpack.graph.enabled=false
11      - xpack.watcher.enabled=false
12      - cluster.name=docker-cluster
13      - bootstrap.memory_lock=true
14      - ES_JAVA_OPTS=-Xms512m -Xmx512m
15      - discovery.zen.minimum_master_nodes=1
16      - discovery.type=single-node
17     ulimits:
18       memlock:
19         soft: -1
20         hard: -1
21       nofile:
22         soft: 65536
23         hard: 65536
24     restart: always
25     ports:
26       - "9200:9200"
27       - "9300:9300"
28     networks:
29       - elastic
30
31   tomcat:
32     image: unidata/tomcat-docker:latest
33     container_name: tomcat
34     restart: always
35     volumes:
36       - ./tomcat:/usr/local/tomcat/tomcat
```



```
37     - ./deploy:/usr/local/tomcat/webapps
38     ports:
39       - "8080:8080"
40     depends_on:
41       - es
42     networks:
43       - elastic
44
45     networks:
46       elastic:
47         driver: bridge
48 networks:
49     elk:
```

Listaxe 4.1: docker-compose.yml

```
1 docker-compose up
```

Listaxe 4.2: Comando para executar o docker-compose.yml

4.2 Configuración necesaria da aplicación

4.2.1 Twitter4J

Como ben se mencionou anteriormente, é necesaria unha autenticación para conectarse á API de Twitter.

Neste traballo créase un Bean (véxase a Listaxe 4.3) que conteña a propia instancia de `TwitterStream`.

```
1     @Bean(name="twitterStream")
2     public TwitterStream twitterStream() {
3         final String CK = XXXX;
4         final String CS = ZZZZ;
5         final String AT = YYYY;
6         final String ATS = TTTT;
7
8         ConfigurationBuilder configurationBuilder = new
9         ConfigurationBuilder();
10        configurationBuilder.setOAuthConsumerKey(CK)
11        .setOAuthConsumerSecret(CS)
12        .setOAuthAccessToken(AT)
13        .setOAuthAccessTokenSecret(ATS)
14        .setJSONStoreEnabled(true);
15
16        return (new
17        TwitterStreamFactory(configurationBuilder.build())).getInstance();
```

```
15 | }
```

Listaxe 4.3: Configuración en Java de Twitter4J

4.2.2 RestHighLevelClient

De maneira similar á configuración de Twitter4J, configúrase un Bean que recolla o cliente de ElasticSearch, tal e como se pode ver na Listaxe 4.4.

Como se configurou no *docker-compose.yml* o Tomcat e o ElasticSearch na mesma rede "es" enténdese que o host ao que se debe conectar a aplicación é "es" e non "localhost".

```
1  @Bean(name="elasticSearchClient")
2  public RestHighLevelClient elasticSearchClient() {
3
4      return new RestHighLevelClient(
5          RestClient.builder(
6              new HttpHost("es", 9200, "http")));
7  }
```

Listaxe 4.4: Configuración en Java de RestHighLevelClient

Desenvolvemento

5.1 Análise de requisitos

Segundo o IEEE, un requisito é a condición ou capacidade que debe ter un sistema para satisfacer un contrato, estándar, especificación ou outra documentación formalmente imposta. Establecen o *qué* se debería obter dun produto pero non o *cómo* obtelo.

5.1.1 Requisitos funcionais

Un requisito funcional está caracterizado como unha función cuns parámetros de entrada que se procesarán para xerar unha saída. Están definidos polas iteracións que terán os usuarios ou usuarias co software final e pódense representar a través de casos de uso, Figura 5.1.

Casos de uso

A continuación mostrase o detalle dos distintos casos de uso do software:

CU-<02>	Ver Tweets en mapa	
Descrición	A persoa usuaria debe ser capaz de visualizar os Tweets recuperados nun mapa de calor xunto o seu listado.	
Secuencia normal	1	O sistema ofrece a opción de buscar os Tweets dos usuarios a través dun campo de texto xunto coa opción de realizar unha consulta pola que filtrar ou a opción de "Ver tódolos chíos en mapa".
	2	[OPCIONAL] A persoa usuaria introduce o nome do usuario ou/e consulta e preme "Ver chíos en mapa".
	3	A persoa usuaria preme "Ver tódolos chíos en mapa".
	4	O sistema mostra a lista de Tweets recuperados da zona xeográfica que se visualiza e que satisfacen a consulta. De non haber unha insertada mostramos todos.
Importancia	Media.	

Táboa 5.1: Caso de uso CU-<02>.

CU-<03>	Buscar Tweets por consulta	
Descrición	A persoa usuaria debe ser capaz de buscar os Tweets a través dunha consulta especificada.	
Secuencia normal	1	O sistema ofrece a opción de buscar os Tweets a través dun campo de texto.
	2	A persoa usuaria introduce a consulta.
	3	O sistema mostra os Tweets recuperados que satisfacen a consulta.
Importancia	Alta.	
Comentarios	A consulta buscará as palabras completas que se indicaron aínda que pode variar o orde de estas.	

Táboa 5.2: Caso de uso CU-<03>.

CU-<04>	Buscar Tweets por usuario	
Descrición	A persoa usuaria debe ser capaz de buscar os Tweets a través do nome do usuario.	
Secuencia normal	1	O sistema ofrece a opción de buscar os Tweets dos usuarios a través dun campo de texto.
	2	A persoa usuaria introduce o nome do usuario.
	3	O sistema mostra os Tweets recuperados que satisfacen a consulta.
Importancia	Alta.	
Comentarios	A consulta buscará os usuarios que conteñan as palabras completas que se indicaron aínda que pode variar o orde de estas.	

Táboa 5.3: Caso de uso CU-<04>.

CU-<05>	Buscar Tweets por consulta e usuario	
Descrición	A persoa usuaria debe ser capaz de buscar os Tweets a través do nome de usuario cunha consulta predeterminada.	
Secuencia normal	1	O sistema ofrece a opción de buscar os Tweets dos usuarios a través dun campo de texto xunto coa opción de insertar a consulta pola que filtrar.
	2	A persoa usuaria introduce o nome do usuario maila consulta.
	3	O sistema mostra os Tweets recuperados que satisfacen a consulta.
Importancia	Alta.	
Comentarios	A consulta buscará os usuarios e os Tweets que conteñan as palabras completas que se indicaron aínda que pode variar o orde de estas.	

Táboa 5.4: Caso de uso CU-<05>.

CU-<06>	Ordear por data	
Descrición	A persoa usuaria debe ser capaz de ordear os Tweets recuperados pola data de creación dos mesmos.	
Secuencia normal	1	O sistema ofrece a opción de buscar os Tweets dos usuarios a través dun campo de texto xunto coa opción de insertar a consulta pola que filtrar.
	2	A persoa usuaria introduce o nome do usuario ou/e consulta.
	3	O sistema mostra os Tweets recuperados que satisfacen a consulta.
	4	O usuario escolle ordear de forma ascendente ou descendente os Tweets en función da data.
	5	O sistema devolve a lista ordeada.
Importancia	Alta.	
Comentarios	A consulta buscará os usuarios e os Tweets que conteñan as palabras completas que se indicaron aínda que pode variar o orde de estas.	

Táboa 5.5: Caso de uso CU-<06>.

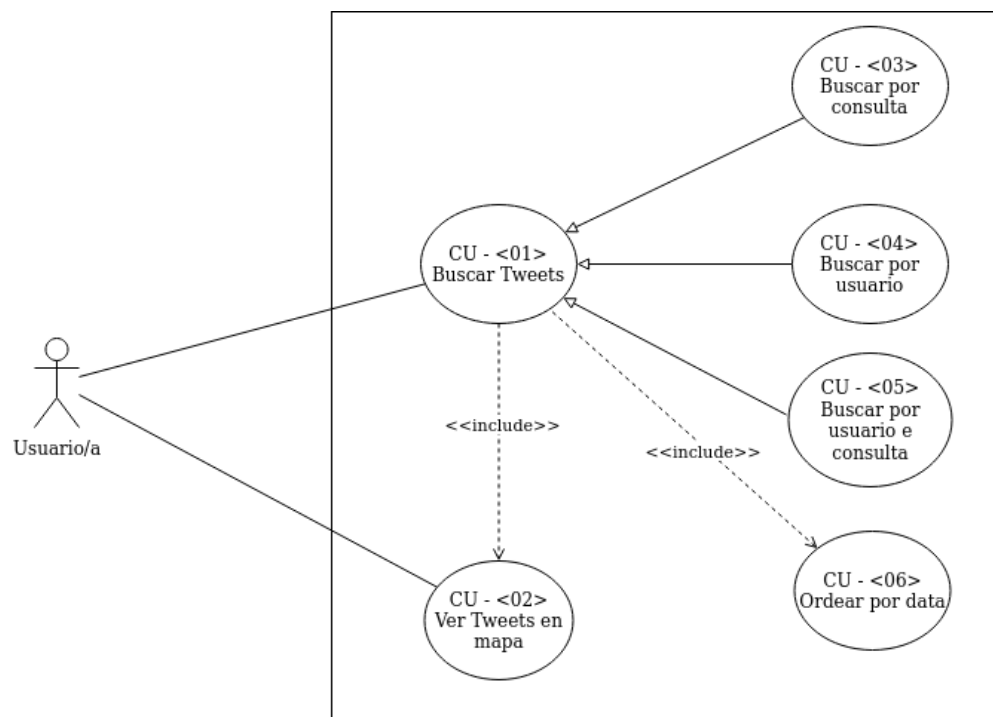


Figura 5.1: Casos de uso do sistema.

5.2 Desenvolvemento

5.2.1 Recuperación de Tweets

Na primeira iteración desta práctica creouse un servizo para a recuperación de Tweets en tempo real. Para isto foi necesaria a configuración mencionada no apartado 4.2.1.

TwitterStream proporciona un *listener* que se activa cando se crean, borran Tweets entre outros.

Nesta práctica só utilizamos o método que detecta a creación de Tweets debido a que serán os que posteriormente se indexen en ElasticSearch.

5.2.2 Persistir Tweets en ElasticSearch

Unha vez analizados os obxectos que devolven os métodos de TwitterStream e diferenciados os atributos que se queren engadir ao servidor de ElasticSearch, establécese a configuración do mesmo como se indica na sección 4.2.2.

Defínese para persistir os datos como un servizo separado.

O *id* do propio Tweet recuperado será o *id* do documento en elasticsearch por se, en proxectos futuros, se realizan cambios sobre o mesmo ou se realiza algún *bulk* o propio Tweet recolla os cambios sen necesidade de crear duplicados.

Debido a que a principal funcionalidade deste software é visualizar os Tweets recuperados dunha consulta (ou velos todos) no mapa de calor, tómase a decisión de só gardar aqueles que conteñan información de localización da publicación do Tweet (atributo "Place" distinto de nulo). Esta decisión é tomada debido a que pode chegar a crear confusión ao realizar unha búsqueda simple e que logo, no mapa, non se recuperen os Tweets que deberían por ese feito de non ter unha xeolocalización asignada.

Nota: Para poder realizar as procuras de xeolocalización é necesario lanzar a petición que se menciona na Listaxe 5.1 en Postman. Onde *place.geoLocation* contén as coordenadas xeográficas donde se publicou o Tweet. Neste caso o índice onde se están a gardar os documentos está bautizado como "chiador".

```
1 GET http://localhost:9200/chiador
2 {
3   "mappings": {
4     "properties": {
5       "place": {
6         "properties": {
7           "geoLocation": {
8             "type": "geo_point"
9           }
10        }
11      }
12    }
13  }
```



```
11     }  
12   }  
13 }  
14 }
```

Listaxe 5.1: Establecer punto xeográfico en Elasticsearch

5.2.3 Procurar Tweets en Elasticsearch

Unha vez que se consegue que o índice conteña elementos procédese a crear un servizo que se encargue de realizar as distintas búsquedas no servidor de Elasticsearch. Créase dende outro servizo independente para que a búsqueda e indexación estean separadas. É dicir, que se poida buscar nun índice e á mesma vez indexar noutro distinto.

Na Listaxe 5.2 móstrase un exemplo onde se procuran os Tweets que se encontren dentro dunha zona determinada definida por o punto noreste e o punto suroeste que, posteriormente, serán proporcionados dende a interfáz gráfica.

```
1      BoolQueryBuilder qb = QueryBuilders.boolQuery();  
2  
3      qb.must(QueryBuilders.geoBoundingBoxQuery("place.geoLocation")  
4              .setCorners(northEast.getLatitude(),  
5                          southWest.getLongitude(), southWest.getLatitude(),  
                          northEast.getLongitude())  
6              .setValidationMethod(GeoValidationMethod.IGNORE_MALFORMED)  
7              .type(GeoExecType.INDEXED));
```

Listaxe 5.2: Consulta para buscar por xeolocalización

5.2.4 Deseño da interfaz gráfica

Unha vez probados os servizo de procura e indexación procédese ca realización da interfaz gráfica para que o software sexa usable por un usuario/a tal e como se mencionou na motivación e obxectivos deste proxecto.

Interfaz gráfica e resultados

Como se mencionou anteriormente, a aplicación estará despregada nun Apache Tomcat co porto 8080 visible. Ao mesmo tempo, o contexto da aplicación será *chiador*.

6.0.1 Gardar Tweets en tempo real

Para activar o servicio de persistencia de Twitter é necesario realizar unha petición GET á URL `http://localhost:8080/chiador/start` e inmediatamente comezaranse a indexar os Tweets en tempo real que conteñan a localización.

6.1 Páxina principal

A URL para acceder á páxina principal é a seguinte: `http://localhost:8080/chiador`. E visualizarase o correspondente á imaxe 6.1.

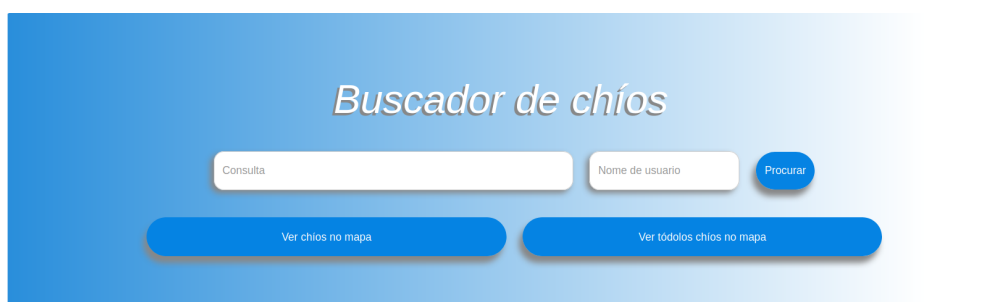


Figura 6.1: Páxina principal.

6.2 Procura por consulta

Visualízanse os Tweets que satisfacen á consulta abaixo do buscador tal e como se mostra na imaxe 6.2. Os resultados mostrados poderanse ordear de forma ascendente ou descendente en función da data.

The screenshot shows the 'Buscador de chíos' search interface. At the top, there's a search bar with 'tweet' entered and a 'Procurar' button. Below the search bar, there are two buttons: 'Ver chíos no mapa' and 'Ver todos os chíos no mapa'. The search results are displayed in a table with three columns: 'Contido', 'Data de creación', and 'Mais información'.

Contido	Data de creación	Mais información
@ my last tweet lol	Sat Nov 21 17:45:40 UTC 2020	Usuario Martin Localización Dallas, TX
My tweet is disputed!	Sat Nov 21 18:54:11 UTC 2020	Usuario sam Vyas Localización Charlotte, NC

Figura 6.2: Procura por consulta.

6.3 Procura por usuario

Visualízanse os Tweets que satisfacen á consulta por usuario abaixo do buscador tal e como se mostra na imaxe 6.3.

The screenshot shows the 'Buscador de chíos' search interface. At the top, there's a search bar with 'Consulta' entered and a 'Procurar' button. Below the search bar, there are two buttons: 'Ver chíos no mapa' and 'Ver todos os chíos no mapa'. The search results are displayed in a table with three columns: 'Contido', 'Data de creación', and 'Mais información'.

Contido	Data de creación	Mais información
@ my last tweet lol	Sat Nov 21 17:45:40 UTC 2020	Usuario Martin Localización Dallas, TX
This tho	Sat Nov 21 19:03:39 UTC 2020	Usuario Martin Localización Seaforth, England
6.6.6.6 http://cc5CV7Bk7cG6	Sat Nov 21 21:20:44 UTC 2020	

Figura 6.3: Procura por usuario.

6.4 Procura por consulta e usuario

Visualízanse os Tweets que satisfacen á consulta e por usuario abaixo do buscador tal e comos e mostra na imaxe 6.4.

Contido	Data de creación	Máis información
@ my last tweet lol	Sat Nov 21 17:45:40 UTC 2020	Usuario Martin Localización Dallas, TX

Figura 6.4: Procura por usuario e consulta.

6.5 Visualización no mapa da procura por consulta e usuario

A través do botón "Visualizar chíos en mapa" visualízanse nun mapa de calor os Tweets que satisfacen á calquera das consultas mencionadas anteriormente. Mostraranse aqueles Tweets que se encontren na zona visible do mapa tal e comos e mostra na imaxe 6.5.

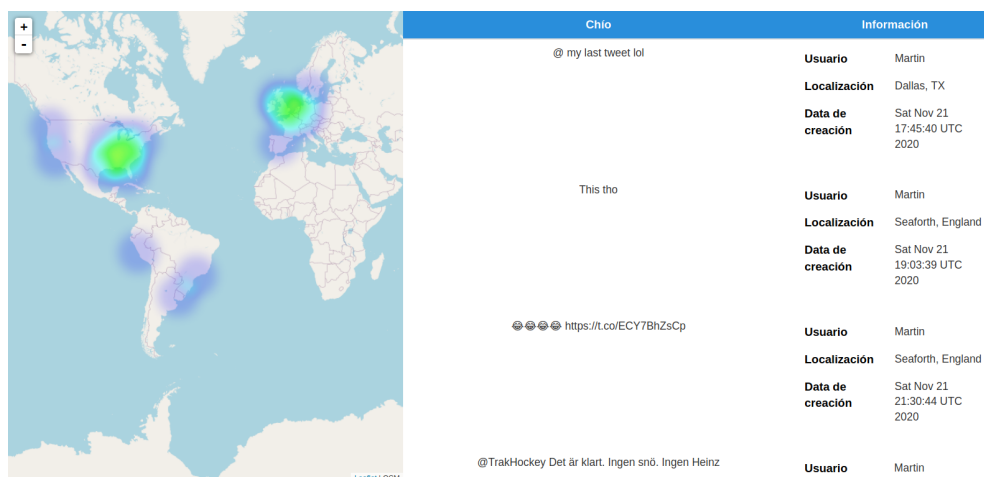


Figura 6.5: Procura por usuario e consulta visualizados en mapa.

6.6 Visualización no mapa de todos os Tweets

A través do botón "Visualizar tódolos chíos en mapa" visualízanse nun mapa de calor os Tweets que satisfacen á calquera das consultas mencionadas anteriormente. Mostraranse aqueles Tweets que se encontren na zona visible do mapa tal e comos e mostra na imaxe 6.6.

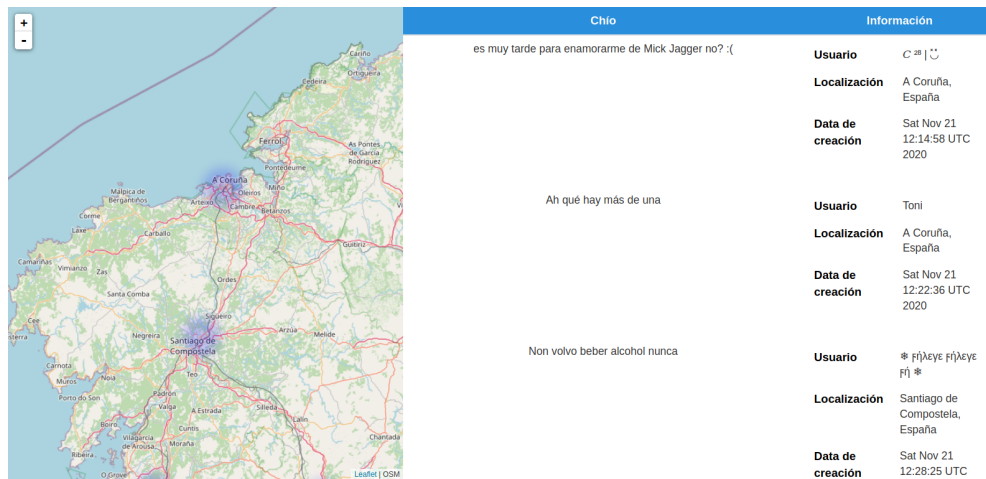


Figura 6.6: Ver tódos os Tweets en mapa.

6.6.1 Gardar Tweets en tempo real

Para activar o servizo de persistencia de Twitter é necesario realizar unha petición GET á URL <http://localhost:8080/chiador/start> e inmediatamente comezan a indexar os Tweets en tempo real que conteñan a localización.

Entregables

A continuación enuméranse os distintos entregables da práctica

1. Memoria: RIWS - Jennifer Dubra Rey.
2. Código fonte: <https://git.fic.udc.es/jennifer.dubra/RIWS> ou no arquivo codigo-fonte-Jennifer-Dubra.zip
3. Desplegable: chiador.war