Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Plataforma software para análise textual de desordes mentais en Internet

Estudante: Jennifer Dubra Rey
Dirección: Álvaro Barreiro García
Javier Parapar López

A Coruña, febreiro de 2020.

A Chus

Agradecementos

Moitas grazas a Álvaro e Javier, por guiarme e orientarme con paciencia nestes meses de arduo traballo facilitando que culmine o meu transcurso polo grao de informática, podendo rematar sactisfactoriamente unha etapa da miña vida.

Chus, a ti por creer en min cando máis o necesitei. Aos meus pais e avoa polo cariño omnipresente e incondicional recibido.

Sempre agradecida aos mellores compañeiros da facultade polo seu apego auténtico e leal.

Resumo

A importancia dos métodos de prevención secundaria para os trastornos mentais é crucial xa que disto depende a calidade de vida futura das persoas que a sofren no presente. Os medios sociais poden favorecer un gran avance no desenvolvemento destes métodos debido a que son canles de comunicación nos que os usuarios e usuarias da Internet participan de forma activa e prevese que isto vaia aumentando co paso dos anos.

O obxectivo principal deste proxecto é deseñar e implementar unha plataforma software que sirva de soporte aos especialistas da saúde mental, como psicólogos ou psiquiatras, para etiquetar aos suxeitos de maneira máis áxil, rápida e sinxela. A aplicación posibilitará a persoa usuaria procesar coleccións de documentos para ver as súas estadísticas, buscar documentos e realizar consultas a través delas. Tamén permitirá ao usuario ou usuaria analizar os resultados dos agrupamentos por similitude dos documentos ou termos que as conforman.

Para poder alcanzar os obxectivos marcados eficientemente, decidiuse usar unha metodoloxía iterativa e incremental debido á flexibilidade e adaptación que aporta ás distintas situacións polas que un proxecto atravesa. Finalmente, conseguiuse unha plataforma software que cumpre cos requisitos especificados.

Abstract

Methods of secondary prevention for mental disorders are of crucial relevance because these depend the quality of future life of the people who are suffering mental diseases at the moment. Social media can improve develop these methods for the reason that it is a channel of communication where a lot of Internet users are involved and it is expected to increase over the years.

The main goal of this project is to design and implement a software plataform to give support to mental health specialists, as psychologists or psychiatrists, to tag subjects agilely, quickly and easily. The application will allow the user process document collections to check their statistics, search documents and search queries in them. It will also allow the user to analyze the clustering results by similarity of the documents or terms that make them up.

In order to achieve the goals set efficiently, it has been decided to use an iterative and incremental methodology for its flexibility and adaptation to the different situations that a project goes through. Eventually, it has been achieved an application that meets with the mentioned requirements.

Palabras chave:

Recuperación de información

Trastornos mentais

eRisk

Coleccións

Medios sociais

LSI

SVD

Keywords:

Information Retrieval

Mental deseases

eRisk

Collections

Social Media

LSI

SVD

Índice Xeral

Índice de Figuras

Índice de Táboas

Capítulo 1

Introdución

Segundo a Organización Mundial da Saúde (OMS) na súa última publicación sobre a saúde mental dos mozos e mozas [1], de cada 6 persoas do mundo que padecen de trastornos mentais, unha delas ten entre 10 e 19 anos. Isto indícanos que un 16% da poboación que sofre de trastornos mentais son adolescentes. Nalgúns casos (en especial a depresión), a consecuencia destas enfermidades desembocan en suicidios, sendo esta a terceira causa de morte entre os mozos e mozas de entre 15 e 19 anos de todo o mundo.

É fundamental a detección destas enfermidades nunha fase temperá debido a que entre as estadísticas que proporciona a OMS, a metade dos trastornos mentais comezan a partir dos 14 anos e a maioría deles non se detectan nin se tratan podendo prolongarse ata as fases adultas dos homes e mulleres. Isto terá un impacto na saúde física e mental destas persoas impedíndolles levar unha vida plena e elevando o suicidio ata o segundo posto entre as principais causas de morte das persoas entre 15 e 29 anos.

Os medios sociais poden ter un papel fundamental no descubrimento das enfermidades mentais porque son canles de comunicación onde os usuarios e usuarias participan de forma activa creando, modificando e intercambiando contido. Co paso dos anos a participación da poboación na comunicación vía Internet foi incrementando rapidamente (Figura 1.1), especialmente nos grupos mozos.

Entre as persoas que sofren os trastornos mentais máis frecuentes (trastornos de ansiedade en xeral e trastornos de depresión maior), pódense observar certos comportamentos característicos de introversión. As condutas introvertidas están caracterizadas polas dificultades a hora de manifestar os pensamentos e sentimentos de forma espontánea e, debido a isto, prodúcese unha exclusión do propio individuo ao resto da sociedade.

A comunicación a través de Internet ofrece múltiples vantaxes, entre elas o feito de que non é necesaria a presenza física para que a comunicación se estableza. Da mesma maneira, o intercambio de información do usuario ou usuaria co resto do mundo pode ser en calquera momento e sobre o tema que el ou ela desexe, de forma contraria do que nos permiten os

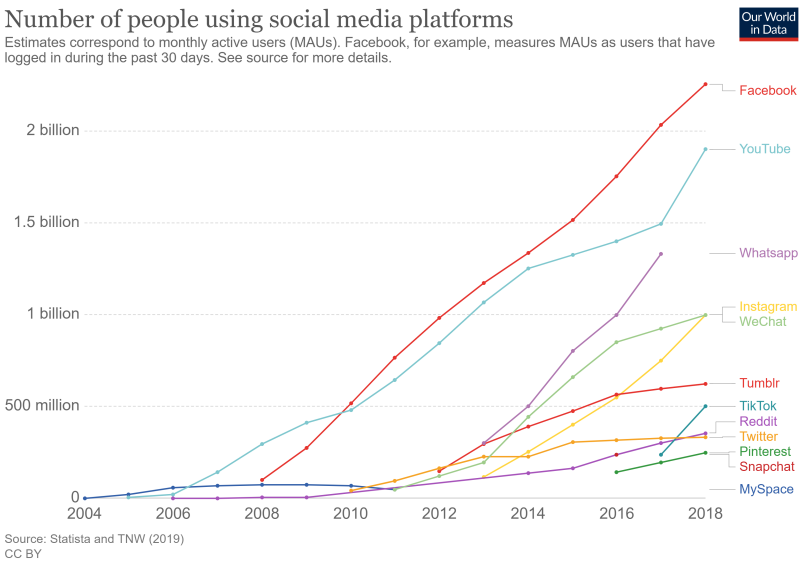

Figura 1.1: Número de usuarios activos por plataforma ao longo dos anos. Datos obtidos de Our World in Data[2].

canles convencionais. Para a gran parte das persoas usuarias, sobre todo para aquelas que son máis introvertidas, é un lugar cómodo para poder expresarse libremente xa que lles evita ter que enfrontarse a un cara a cara co resto de membros de Internet.

1.1 Motivación

A importancia da detección dos trastornos mentais nas fases temperás do seu desenvolvemento é crucial xa que disto depende a calidade de vida futura das persoas que as sofren no presente. Unha detección a tempo pode significar unha maior probabilidade de éxito no seu tratamento e incluso unha restauración completa da saúde mental sen secuelas no paciente.

Os trastornos mentais, a maiores de ter unha carga social e psicolóxica no individuo, tamén teñen un impacto económico moi elevado. Concretamente en España, segundo o estudo publicado en Plos One sobre o custe dos tratamentos dos trastornos cerebrais en España [3], os tratamentos cerebrais (entre os que se inclúen os tratamentos por trastornos mentais) supuxeron un custo de case o equivalente ao 8% do PIB do país. Tense en conta tamén que, segundo a Organización Mundial da Saúde na publicación de 2019 sobre os trastornos mentais [4], nos países con ingresos altos entre o 35% e o 50% enfermidades mentais graves non son tratadas e esta cifra sobe nos países de ingresos baixos e medios ata un porcentaxe entre o 76% e 85%.

Hoxe en día, son poucos os métodos de prevención secundaria en canto se remite as enfermidades mentais, pese a ser moi necesarios. Os recursos existentes para a detección mani-

festan un custo moi alto de tempo per se e isto pode provocar que os trastornos transcorran a fases máis avanzadas durante o seu diagnóstico.

Debido a súa gran significación xurdiu a idea de levar a cabo este proxecto para ofrecer soporte aos especialistas adicados a esta área da saúde.

1.2 Obxectivos

O obxectivo principal deste proxecto é desenvolver unha plataforma software de análise de textos para dar soporte aos especialistas da saúde mental e así poder levar a cabo os métodos de prevención secundaria de maneira máis áxil, rápida e sinxela.

Este proxecto procesará as coleccións de rexistros dos suxeitos, recollidas a través de Reddit[1], para posteriormente mostrar de maneira gráfica as distintas relacións de conceptos ou documentos que se poden encontrar no conxunto a axuizar.

Os datos que se utilizarán para procesar as coleccións serán o identificador do suxeito e os distintos textos que publicou, que se almacenarán nun índice invertido do que se implementarán as funcións de acceso e procura.

A maiores, realizaranse métodos de agrupamento. Contemplarase agrupar os termos que forman parte do índice para poder extraer as características xerais que estes conteñen segundo a distancia entre termo e termo, así como agrupar os documentos segundo os espazos entre eles. As distancias variarán segundo a representación dos termos ou documentos que virán configuradas pola persoa usuaria da aplicación.

A representación dos termos será un vector do tamaño da colección onde cada posición corresponderá a un documento:

- Tf: O valor do termo será a frecuencia deste en cada documento da lista.
- Tfidf: Esta representación indícanos que tan importante é cada termo para cada documento do corpus. A relevancia da palabra crece de forma proporcional ao número de veces que aparece o termo no documento entre o número de documentos que o conteñen.
- Binaria: Cada posición do vector será binaria, valendo 1 se o termo existe no documento e 0 do contrario.

A representación dos documentos será un vector onde cada posición se corresponderá a un termo:

- Tf: O valor que terá cada posición será a frecuencia no documento do termo correspondente a cada índice.

[1] https://www.reddit.com/

- Tfidf: Cada índice do vector será o resultado do tfidf do respectivo termo no documento.
- Tfidf:Binaria: O valor de cada índice será binario, valendo 1 se o documento contén o termo correspondente a posición e 0 do contrario.

Para realizar os métodos de agrupamentos mencionados anteriormente de forma máis eficaz poranse en práctica técnicas propias do análise semántico latente (Sección 2.4).

Os resultados servirán de guía para os expertos para valorar de maneira crítica o etiquetado final de cada suxeito, permitíndolles ser máis rápidos e eficaces.

Contemplaranse tamén como obxectivos:

- Flexibilidade: A plataforma deberá soportar diferentes formatos de entrada e representación de documentos de maneira axeitada sen que afecte ao seu rendemento.
- Eficacia: Será prioritario uns resultados acordes ao que se procura pois serán a base dos etiquetados futuros presentados polos especialistas da rama da psicoloxía.
- Eficiencia: A plataforma terá que ser capaz de traballar con grandes volumes de datos de maneira axeitada.

1.3 Estrutura da memoria

A continuación farase unha breve introdución aos diferentes apartados da memoria dos que se profundará en cada capítulo:

1. **Introdución**: Explícase o contexto e a motivación deste proxecto así como os obxectivos do mesmo.
2. **Conceptos**: Ofrécense os coñecementos básicos utilizados de forma resumida para ter un marco teórico para a comprensión da memoria e proxecto.
3. **Tecnoloxías, ferramentas e librerías**: Detállanse as características das tecnoloxías, ferramentas e librerías utilizadas para o desenvolvemento do software.
4. **Metodoloxía e xestión do proxecto**: Neste capítulo procederanse coas técnicas da enxeñaría do software postos en práctica.
5. **Desenvolvemento**: Explicarase paso a paso a execución do proxecto.
6. **Resultados**: Farase un resumo de algúns resultados obtidos do software así como a eficiencia do mesmo en algoritmos como o de indexación, agrupamento e LSI.
7. **Conclusión e liñas futuras**: Avalíase o produto global e metas alcanzadas como posibles traballos futuros do proxecto.

1.4 Plan de traballo

A grandes risocs o desenvolvemento do proxecto pódese dividir nas seguintes fases:

1. Estudo dos requisitos funcionais e non funcionais.
2. Estudo e elección das tecnoloxías, ferramentas e librerías.
3. Desenvolvemento iterativo e incremental do software, onde se tomarán as medidas de corrección de desviacións de alcance necesarias.
4. Creación da memoria.

Capítulo 2

Conceptos

Este capítulo está adicado a introdución dos conceptos básicos sobre os que se funda o proxecto. Explicarase de que trata ERisk e as características propias das técnicas empregadas neste traballo.

2.1 eRisk

Early Risk Prediction on the Internet. O grupo de investigación IRLab da UDC[1] abordou no ano 2012 a detección de predadores sexuais na Internet[5, 6] e posteriormente nos anos 2017, 2018 e 2019 organizou o laboratorio de eRisk[2] no eido de CLEF[7, 8, 9]. En este momento estase levando adiante eRisk 2020. Esta iniciativa explora a metodoloxía de avaliación, as métricas de efectividad e as aplicacións prácticas da procura de riscos en fases temperás en Internet. Aínda que as tecnoloxías de eRisk se poden aplicar a distintas áreas, están enfocadas maioritariamente nas áreas da saúde e da seguridade. Entre os exemplos de aplicacións que propoñen están a detección de posibles pedófilos, acosadores, persoas con tendencias suicidas ou susceptibles a depresión.

Este proxecto está relacionado con eRisk polo feito de que pretende resaltar as posibles palabras chave que poidan ser de utilidade para os psicólogos para detectar trastornos mentais en fases temperás, pois encontrar sinais deste tipo de enfermidades pode evitar males maiores como autolesións ou, incluso, suicidios. A maiores, eRisk proporciona ao proxecto os rexistros dos suxeitos utilizados para a investigación deste.

2.2 Sistemas de Recuperación de Información

Os sistemas de recuperación de información son aqueles que permiten recoller a información dos documentos electrónicos. É unha ciencia que nace da necesidade de información do

[1] http://irlab.org/
[2] https://early.irlab.org/

usuario a través de consultas ao sistema co obxetivo de recibir un *ranking* de textos, imaxes ou outro tipo de datos de forma relevante.

Entre as aplicación dos sistema de recuperación máis coñecidas encóntranse os motores de procura como Google ou Bing.

2.2.1 Web crawling

Un *crawler* é un bot de Internet que se usa para a recuperación e preservación de documentos de Internet para un posterior preprocesado dos sistemas de recuperación de información

A este "rastreador" especifícanselle unhas URLs sementes das que identificará os hipervínculos e engadiraos a unha lista de páxinas para visitar chamada *fronteira de seguimento*. É un proceso iterativo no que se examinarán as URLs da fronteira de seguimento e iranse engadindo os hipervínculos que se encontren á lista. Cada páxina que visita queda gardada de tal maneira que se poida ver, ler e navegar como se fora na Web.

2.2.2 Preprocesado

As técnicas de preprocesado son moi importantes porque melloran significativamente o rendemento do sistema e a eficacia deste. A continuación explicaranse uns exemplos de preprocesado utilizados neste proxecto.

Converter a minúsculas

Esta técnica converte todos os caracteres alfabéticos do documento a minúsculas. Isto fai que se aforre espazo en memoria porque as distintas combinacións de minúsculas e maiúsculas dunha palabra estarán mapeadas a unha mesma (Figura 2.1), co cal é suficiente engadir outra entrada na *posting list* do termo en minúsculas gardado no índice. En consecuencia, ao aumentar as *posting list* e non o tamaño do índice, o tempo de indexación e procura tamén diminúe.

Eliminar palabras con pouco significado

En todos os textos existen palabras que se repiten con frecuencia pero non teñen significado per se, de gardalas aumentaría significativamente o custo de indexación, procura e o espazo en memoria. Para evitar isto, existen listas que filtrarán as palabras que conteñan, as *stopwords*.

2.2.3 Indexación

A indexación é a acción encargada de gardar os datos recollidos do preprocesado en índices invertidos. Os índices invertidos son unha lista de termos onde cada un destes contén unha

Termos	Mapeo no índice
Indice InDice INDICE	indice
CaSA CASA casa	casa

Táboa 2.1: Exemplo de conversión a minúsculas

lista de documentos nos que aparecen (*posting lists*) tal e como se mostra na Táboa 2.2 para os documentos:

- Documento 1: A vaca di mu.
- Documento 2: Di mu, di mu.
- Documento 3: Vaca di mu mu mu.
- Documento 4: Vaca mu.

Palabra	Documentos
a	1
vaca	1, 3, 4
di	1, 2, 3
mu	1, 2, 3, 4

Táboa 2.2: Exemplo de índice invertido

As listas onde se mostran os documentos nos que aparece un determinado termo tamén poden ter recollido as posicións nas que aparece en cada documento, Táboa 2.3, ou as súas frecuencias nestes.

Sen un índice, un motor de procura tería que escanear cada documento dun corpus, o cal requiría un considerable custo en tempo e rendemento. A pesar de que un índice require espazo en memoria e as operacións de actualizacións teñen un maior custo, para os motores de procura compensa polo tempo aforrado durante a recuperación de información.

2.2.4 Procesado de consultas

Existen diferentes técnicas para o procesado de consultas nos sistemas de RI entre as que destacan as propostas por Turtle and Flood, DAAT, TAAT [10].

Palabra	Documentos e posicións
a	1: 1
vaca	1: 2 3: 1 4: 1
di	1: 3 2: 1, 3 3: 2
mu	1: 4 2: 2, 5, 6 3: 3, 4, 5 4: 2

Táboa 2.3: Exemplo de índice cas posicións dos termos nos documentos.

Term-at-a-time (TAAT)

Os algoritmos term-at-a-time teñen a característica de procesar cada termo da consulta de forma individual, recorrendo as *posting lists* tantas veces como palabras na *query* haxa, Listaxe 2.1.

É necesario un *array* acumulador para almacenar os *scores* (puntuacións) de cada documento para, ao rematar, ordealos descendentemente e retornalo como resultado.

```
Inicializar acumulador A[] a 0

Repetir para cada termo t da consulta:
    Repetir para cada documento d da postinglist do termo t:
        tf = frecuencia do termo t no documento d
        A[d] +=  f(d,tf) sendo f unha función de ranking da que se
    obtén un score para un documento e un termo
    Fin
Fin

Ordear A[] de maior a menor
```

Listaxe 2.1: Pseudocódigo TAAT

Document-at-a-time (DAAT)

Document-at-a-time, ao contrario que term-at-a-time, procesa todas as *postinglist* dos termos dunha consulta de forma simultánea e debido a isto, non hai necesidade de un acumulador e o espazo de memoria que usa depende do número máximo de resultados a devolver.

Sitúase un cursor en cada entrada das *postinglists* nas que apareza o documento a examinar e súmanse as puntuacións individuais obtendo a súa puntuación final, tal e como se mostra na Figura 2.1, onde se representa a esquerda na consulta "a b c" coas súas *postinglists* cos *scores* asociados a cada entrada e na dereita o resultado obtido logo de dúas iteracións.

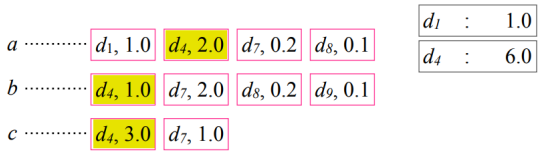

Figura 2.1: Exemplo de algoritmo DAAT obtido do Instituto Max Planck de Informática [3].

2.2.5 Modelos de Recuperación de Información

Un modelo de recuperación especifica a forma na que se declaran os documentos, consultas e as funcións de recuperación. Pese que destacan o modelo booleano e o modelo vectorial nos sistemas de recuperación de información, tamén se encontra o modelo probabilístico nos textos de RI ao igual que os Language Models [11, 12].

Modelo booleano

Baseado na teoría de conxuntos e na lóxica booleana nas cales a consulta e os documentos están representados por un conxunto de termos e a recuperación de información efectúase comprobando se os documentos conteñen ou non os vocábulos da consulta. Os documentos recuperados carecen dunha orde concreta.

As consultas con fáciles de entender polo usuario do sistema e fáciles de implementar polo desenvolvedor.

A continuación mostraranse algúns exemplos de consultas con operadores lóxicos para os documentos de exemplo da Sección 2.2.3 e os respectivos resultados:

- a AND di: Documento 1.
- a OR di: Documentos 1, 2 e 3.
- a NOT di: Sin resultados.

En cambio, as consultas máis complexas con expresións lóxicas poden ser dificultosas para usuarios inexpertos.

[3] http://resources.mpi-inf.mpg.de/departments/d5/teaching/ws13_14/irdm/slides/irdm-5-3.pdf

Modelo vectorial

O modelo vectorial é a forma de representar os documentos e consultas de forma alxebrica a través dun vector onde cada dimensión corresponde a un dos termos que os conforma, Ecuacións 2.1 e 2.2. Esta representación facilita as medidas de similitude entre un documento e unha consulta mencionadas na Sección 2.3. Permite controlar o número de resultados según un umbral e ordear os documentos recuperados en función do grao de relevancia que teñan. A pesar das vantaxes aporta, o seu uso implica unha pérdida semántica e sintáctica da información pois non é capaz de detectar casos de polisemia e sinonimia.

$$documento_i =< termo_{1,i}, termo_{2,i}, ..., termo_{n,i} > \tag{2.1}$$

$$consulta = q = (termo_{1,q}, termo_{2,q}, ..., termo_{n,q}) \tag{2.2}$$

Este modelo é especialmente útil para as coleccións grandes debido a súa facilidade implementación e eficiencia desta. Existen diferentes formas de representar cada termo de forma cuantitativa, entre as que se destacan:

- Representación binaria: Cada vector de cada documento terá tantos termos como haxa no índice e o valor de cada elemento será 1 ou 0 dependendo se o termo aparece ou non no documento. As consultas trataranse da mesma forma.

  Para os documentos de exemplos da Sección 2.2.3 e a consulta ”a vaca”, aplicando unha función de similitude entre documentos e consulta de +1 por cada termo do documento que coincida ca consulta, os resultados serían os da Táboa 2.4 sendo o documento 1 o máis similar a consulta.

	A	vaca	di	mu	
Documento 1	1	1	1	1	similitude = 2
Documento 2	0	0	1	1	similitude = 0
Documento 3	0	1	1	1	similitude = 1
Documento 4	0	1	0	1	similitude = 1
Consulta ”a vaca”	1	1	0	0	

Táboa 2.4: Exemplo de consulta nun modelo binario.

- Representación de pesos: Serve para medir que tan representativo é un termo no documento ou consulta e este valor será o seu peso.

  Os elementos distintivos das representacións por pesos son o tf (frecuencia do termo t nun documento), o df (frecuencia do documento na colección N) e a normalización

aplicada aos documentos da colección. Tanto o tf como o df e a normalización son os responsables das variacións dos pesos entre un método e outro.

As distintas combinacións de formas de calcular cada elemento chámanse esquemas de pesado e seguen un sistema de nomeado SMART *(System for the Mechanical Analysis and Retrieval of Text).* Pódese observar na Figura 2.2 a notación SMART das distintas aproximacións para o pesado de termos presentadas por Salton e Buckley.

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}, \alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

Figura 2.2: Notación SMART para as distintas fórmulas de tf, df e normalización. Imaxe obtida do grupo de procesado de linguaxe natural de Stanford [4].

No sistema de notación SMART cada esquema noméase a través dun triplete de letras onde cada unha representa o tf, o df e a normalización equivalentemente. A representación binaria mencionada anteriormente representaríase en SMART como *bnn.*

Neste proxecto úsase o sistema de representación de documentos e consultas por pesos, tal e como se detalla na Sección 1.2.

2.3 Agrupamento

Clústering. É o procedemento de separación de un conxunto de datos en grupos con características semellantes. Estas características veñen dadas normalmente pola distancia ou similitude.

A distancia euclidiana, Ecuación 2.3, é o método comunmente usado nos sistemas de recuperación de información para medir como de lonxe están dous vectores en un espazo K-dimensional. Trátase da raíz cadrada da suma da diferencia ao cadrado das compoñentes (a_i e b_i) que conforman os vectores A e B.

$$DistanciaEuclidiana(A,B)=\sqrt{\sum_{i=1}^{k}(a_i-b_i)^2} \tag{2.3}$$

[4] https://nlp.stanford.edu/IR-book/html/htmledition/document-and-query-weighting-schemes-1.html

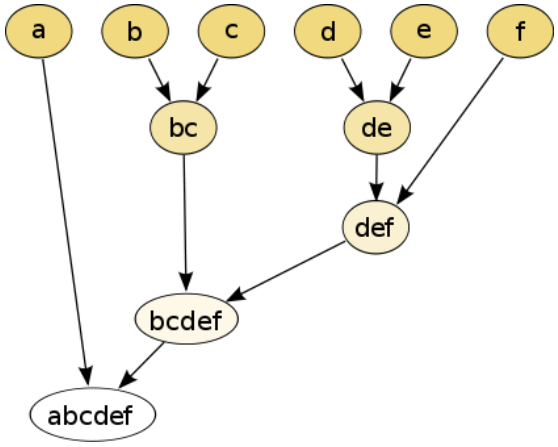

Figura 2.3: Exemplo de agrupamento xerárquico aglomerativo obtido de Wikipedia [5].

A similitude mídese nun espazo vectorial como o coseno do ángulo que forman dous vectores de palabras, como se pode ver representada na Ecuación 2.4 entre a representación vectorial das palabras A e B sendo |A| e |B| os seus respectivos módulos. Considérase entón que canto menor sexa esta medida, maior será a semellanza.

$$Similitude(A, B) = \frac{A \cdot B}{|A| \cdot |B|} \tag{2.4}$$

O *clústering* forma parte do aprendizaxe non supervisado, o que quere decir que para esta metodoloxía só temos datos de entrada.

2.3.1 Técnicas de agrupamento

Existen dúas grandes vertentes entre as metodoloxías de agrupamento de datos como son o agrupamento xerárquico e o particionado.

Agrupamento xerárquico

O agrupamento xerárquico pode ser aglomerativo ou divisivo. A diferencia entre o agrupamento aglomerativo e o divisivo encóntrase en que o aglomerativo é ascendente, os pares de grupos mestúranse mentres se soben na xerarquía, tal e como se mostra no exemplo da Figura 2.3. En cambio, o divisivo é descendente, os grupos vanse dividindo en pares mentres se baixa na xerarquía.

[5] https://en.wikipedia.org/wiki/Hierarchical_clustering

Agrupamento particionado

Nesta vertente o número de grupos nos que se van a dividir os datos sábense de antemán. O obxetivo está en dividir o conxunto en k grupos que optimicen o criterio de partición (similitude).

Para a procura do óptimo global, requírese unha procura exhautiva de todas as particións. Existen algoritmos que con métodos heurísticos encontran os óptimos locales como son os algoritmos de K-Means e K-Medioids.

2.3.2 K-Means

O proceso de agrupación do algoritmo particionado K-Means consiste en organizar os datos de forma iterativa en K grupos onde se busca que cada grupo teña a mínima distancia euclidiana entre puntos que o conforman e a máxima varianza co resto de grupos, Listaxe 2.2.

```
k = número de grupos

Asignar os k centroides de cada grupo de forma aleatoria

Repetir ata converxencia ou un número fixo de iteracións:

    Para cada punto que non sexa centroide:
        Buscar o grupo cuxo centroide que estea máis cerca
        Asignar o punto a ese grupo

    Para cada grupo:
        centroide = media de todos os puntos que forman o grupo.

Fin
```

Listaxe 2.2: Pseudocódigo do algoritmo K-Means.

A desvantaxe deste algoritmo é a súa inicialización, pois escolle os centros iniciais de cada grupo de maneira aleatoria e isto pode provocar agrupamentos febles pouco similares aos grupos óptimos.

2.3.3 K-Means++

Para solucionar a deficiencia do K-Means, propúxose o algoritmo K-Means++ como alternativa onde se especifica un procedemento para inicializar os centros dos conxuntos antes de proceder cas sucesivas iteracións.

Esta inicialización consiste en escoller o primeiro centro de forma aleatoria e logo o resto de centros elíxense en función de unha probabilidade proporcional a súa distancia ao cadrado

dende o centro do conxunto existente máis próximo ao dato.

Con esta inicialización prodúcese unha mellora considerable do erro final de K-Means. Malia que se toma un tempo extra en escoller os centros iniciais, o algoritmo converxe de ata o dobre de rápido co tradicional despois desta selección inicial de centros.

2.4 LSI e LSA

Latent Semantic Indexing, *Latent Semantic Analysis*. O análise latente semántico, tamén coñecido como LSI, é unha técnica de procesado do linguaxe natural e de minería de datos que analiza as relacións entre un conxunto de documentos e os termos que conteñen para producir os conceptos que os relacionan.

Ao non haber un método que recolla a sinonimia dos termos na representación por pesos, o valor da similitude dos documentos que conteñan termos semellantes aos da consulta será menor en comparación ca similitude que tería para o usuario, terán por tanto unha menor relevancia. Esta representación tampouco ten en conta a polisemia das palabras, un mesmo vocabulo pode aparecer no mesmo documento múltiples veces pero con distintos significados, isto causa unha similitude maior da real.

LSI asome que aqueles termos de significado similar aparecerán en documentos que traten sobre temas parecidos, pódese considerar entón como unha agrupación suave de termos a cal incrementa a precisión nas consultas.

Para obter as relacións mencionadas anteriormente constrúese unha matriz C, Táboa 2.5, cuxas filas representan aos termos presentes na colección e as columnas aos documentos, sendo a relación entre filas e columnas a representación escollida (tf, tf-idf, binaria...) de cada termo en cada un dos documentos da matriz.

	Documento 1	Documento 2	Documento 3	Documento 4
a	1	0	0	0
vaca	1	0	1	1
di	1	2	1	0
mu	1	3	3	1

Táboa 2.5: Matriz $C_{m \times n} dondice 2.3 cost f sdostermos.$

2.4.1 SVD

Singular Value Decomposition é a técnica de reducción de dimensionalidade de matrices *sparse* con valores non negativos utilizada en LSI que as representa como un produto de matrices de dimensións menores. A descomposición dunha matriz C, como a do exemplo da Táboa

2.5, formularíase según a Ecuación 2.5, onde Σ é a matriz que contén os valores singulares de C, U a matriz de termos e V^T a matriz de documentos sendo r é o rango da matriz a reducir C.

$$C_{m\times n} = U_{m\times r}\Sigma_{r\times r}V^T_{n\times r} \tag{2.5}$$

Isto permite eliminar ruído da matriz para logo obter o conxunto de conceptos relacionados cos termos e documentos como a sinonimia ou a polisemia usando menos dimensións ca no conxunto de datos orixinal.

O custo computacional desta descomposición é significativo, o cal é o maior obstáculo para unha aplicación máis estendida de LSI na minería de datos.

A continuación móstrase un exemplo práctico obtido do grupo de procesado de linguaxe natural de Stanford[6].

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Táboa 2.6: Matriz C.

Para a matriz C, Táboa 2.6, logo de aplicarlle o SVD obterase a matriz diagonal, Táboa 2.7, cos valores singulares de C ordeados de maior a menor.

2.16	0.00	0.00	0.00	0.00
0.00	1.59	0.00	0.00	0.00
0.00	0.00	1.28	0.00	0.00
0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.39

Táboa 2.7: Matriz $\Sigma_{5\times 5.}$

A multiplicación da matriz de termos, Táboa 2.8, con Σ e a matriz de documentos, Táboa 2.9, será entón unha aproximación da matriz inicial C.

[6] https://nlp.stanford.edu/IR-book/html/htmledition/latent-semantic-indexing-1.html

	d_1	d_2	d_3	d_4	d_5
ship	-0.44	-0.30	0.57	0.58	0.25
boat	-0.13	-0.33	-0.59	0.00	0.73
ocean	-0.48	-0.51	-0.37	0.00	-0.61
voyage	-0.70	0.35	0.15	-0.58	0.16
trip	-0.26	0.65	-0.41	0.58	-0.09

Táboa 2.8: Matriz $U_{5\times5.}$

	d_1	d_2	d_3	d_4	d_5	d_6
1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
2	-0.29	-0.53	-0.19	0.63	0.22	0.41
3	0.28	-0.75	0.4	-0.20	0.12	-0.33
4	0.00	0.00	0.58	0.00	-0.58	0.58
5	-0.53	0.29	0.63	0.19	0.41	-0.22

Táboa 2.9: Matriz $V_{6\times5transposta.}$

Capítulo 3

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 por ser o linguaxe no que está implementado a ferramenta utilizada para a recuperación de información Lucene (Sección 3.3.1) . 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 Springboot

Springboot 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 Eclipse IDE

Eclipse é un software de código aberto e multiplataforma que ofrece un entorno de desenvolvemento Java. Dispón 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.

Fíxose esta escolla porque é unha ferramenta intuitiva e fácil de usar da cal xa se conta con unha experiencia previa do seu uso.

3.2.2 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 construir e as dependencias con outros módulos e compoñentes externos. Tamén se poden incluir 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.

3.2.3 GitLab e Git

Gitlab é a ferramenta escollida para levar a cabo un control de versións. Consiste nun software web baseado en Git que 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.

3.2.4 Taiga

Taiga é a plataforma de código aberto de xestión de proxectos intuitiva e simple para metodoloxías áxiles como, a usada neste proxecto, Scrum. Permite xestionar historias e tarefas así como a asignación de recursos nestas.

3.2.5 Apache Tomcat

Apache Tomcat é o software de código libre que proporciona un servidor HTTP web dispoñible para os distintos sistemas operativos que conten 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 tratamento de coleccións de documentos, agrupamento de datos, visualización dos mesmos, etc..

3.3.1 Apache Lucene

Apache Lucene é a API Java de código aberto baixo a licencia de Apache utilizada para a recuperación de información. Crea índices invertidos que mapean termos con documentos o cal permite que as consultas sexan máis rápidas xa que se procura a través do índice e non da colección.

A implementación desta API estandariza os distintos datos de entrada en documentos de Lucene (Document) formados por un conxunto de campos (Fields) para a construción e manexo do índice. Proceso de indexación representado na Figura 3.1.

Lucene aporta distintos tipos de analizadores que identifican e procesan cada termo normalizándoo e deciden se finalmente se indexa ou non. O máis usado é o *StandardAnalyzer* que converte todas as letras de cada vocabulo a minúsculas, recoñece URLs e mais elimina aqueles que non pasan polo filtro das *stopwords*. Existen analizadores máis exhaustivos en cada idioma, por exemplo o *EnglishAnalyzer*, que se especializa no procesado de textos en inglés contendo *stemming* e filtros de posesivos e de marcadores de palabras chave.

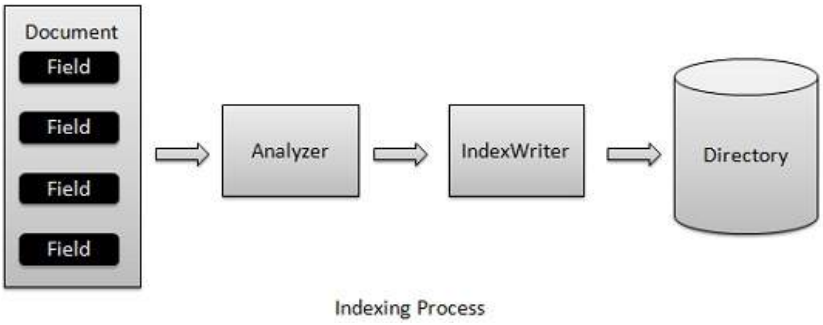

Figura 3.1: Proceso de indexación. Figura obtida de TutorialsPoint [1].

Para consultar a través do índice, Lucene aporta un soporte a distintos formatos de entrada a través do *QueryParser*. Pódese entón facer consultas con modificadores de términos (con caracteres comodín, procuras *fuzzy*,...), operadores booleanos, conxuntos de consultas e caracteres especiais entre outros.

A característica principal que destacou Lucene no seu lanzamento foi o seu método de indexación incremental, pois ata entón este tipo de implementacións só permitían a indexación por bloques. A indexación incremental facilita a adición, eliminación ou actualización de entradas individuais no índice.

3.3.2 Math3

Math3 é a librería de Apache Commons, a cal se utiliza para o proceso de agrupamento dos datos e de redución de dimensionalidade. Consta das clases KMeansPlusPlusCentered, encargada de realizar os conxuntos conforme as especificacións do algoritmo KMeans++ mencionadas anteriormente na Sección 2.3.3, e SingularValueDecomposition, implementación da descomposición de matrices explicada na Sección 2.4.1.

[1] https://www.tutorialspoint.com/lucene/lucene_indexing_process.htm

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 D3.js

Data-Driven Documents consiste nunha biblioteca de JavaScript para representación gráfica, dinámica e iterativa de datos en aplicacións web utilizando SVG, HTML5 e CSS. D3 fai especial énfase na súa capacidade de adaptación aos distintos navegadores para unha correcta visualización a través do uso dos estándares web.

Utilízase neste proxecto para a representación dos distintos grupos de termos ou documentos semellantes.

Capítulo 4

Metodoloxía e xestión do proxecto

Neste capítulo concretarase a metodoloxía usada e máis a xestión do proxecto seguida neste traballo.

4.1 Metodoloxía

Unha metodoloxía consiste nun conxunto de técnicas e procedementos para o desenvolvemento software que ten en conta factores como os custos, planificación, calidade e os riscos do produto. Por desgraza, non existe unha metodoloxía universal de modo que se deberán adaptar as distintas técnicas e procedementos a cada proxecto en particular.

As metodoloxías de desenvolvemento pódense dividir en dous tipos de grupo segundo as súas características e obxectivos, Táboa 4.1.

	Robustas	**Áxiles**
Centradas en	Documentación e produto final	Seguimento software e nas persoas
Evaluación de riscos	Complexa	Sinxela
Custo de cambios	Alto	Baixo
Preparadas para grupos	Grandes	Pequenos (<10 integrantes)
Contratos	Prefixados	Evolutivos

Táboa 4.1: Características das metodoloxías robustas e áxiles.

Segundo as características deste proxecto as metodoloxías áxiles son as que mellor se adaptan, están caracterizadas por seguir o ciclo de vida de forma iterativa e xerar produtos entregables a finais de cada iteración, como se pode ver na Figura 4.1. Isto beneficia ao cliente

[1] https://openwebinars.net/blog/que-es-scrum/

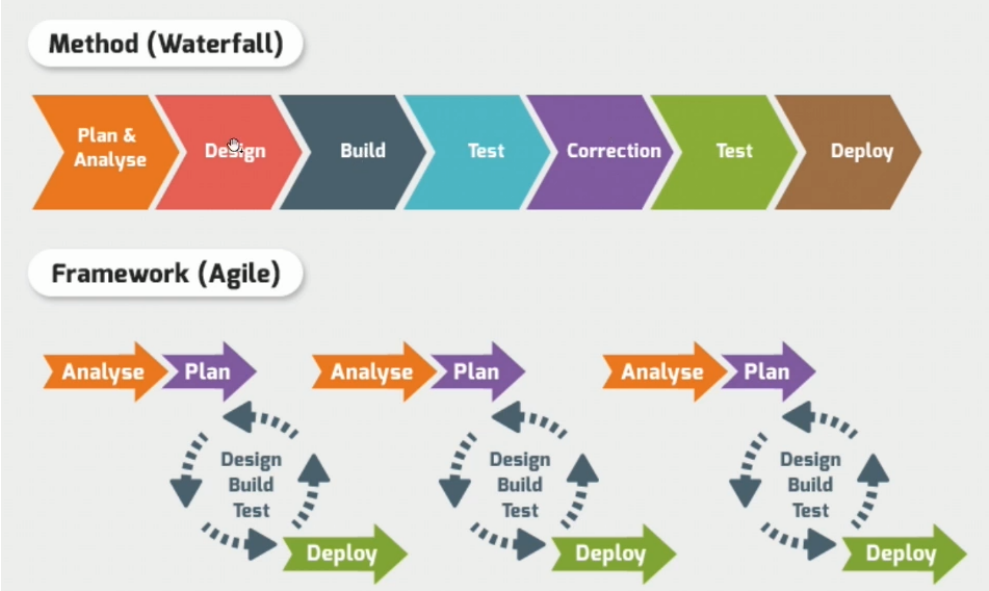

Figura 4.1: Diferencias entre metodoloxías tradicionais e áxiles obtidas de OpenWebinars [1].

de forma que na primeira iteración xa hai un produto parcial cos principais requisitos/obxectivos e posteriormente, nas seguintes iteracións, refinarase e completarase cos requisitos que falten ou surxan.

4.1.1 Scrum

Scrum é a metodoloxía áxil a seguir escollida neste proxecto. Scrum aparece oficialmente en 1995 gracias a Ikujiro Nonaka e Hirotaka Takeuchi como un conxunto de regras de boas prácticas para o desenvolvemento software.

O proceso seguido por Scrum, Figura 4.2, caracterízase por que os equipos serán auto-organizados e auto-dirixidos, o que quere dicir é que eles mesmos se van a asignar e dirixir as tarefas a realizar.

A maiores, Scrum caracterízase pola definición de *sprints*, roles e eventos.

***Sprints* ou ciclos de tempo:**

Un *sprint* é o intervalo de tempo (normalmente 2 semanas) no que se realiza cada iteración da metodoloxía áxil. O obxectivo de cada *sprint* é obter un produto de valor que sirva de base de desenvolvemento para os seguintes *sprints*.

A forma de realizar as iteracións de forma ordeada é organizando os tarefas segundo prioridades, así as primeiras iteracións satisfarán os obxectivos principais do proxecto e as últimas engadirán funcionalidades non preferentes.

[2] https://www.scrum.org/resources/blog/que-es-scrum

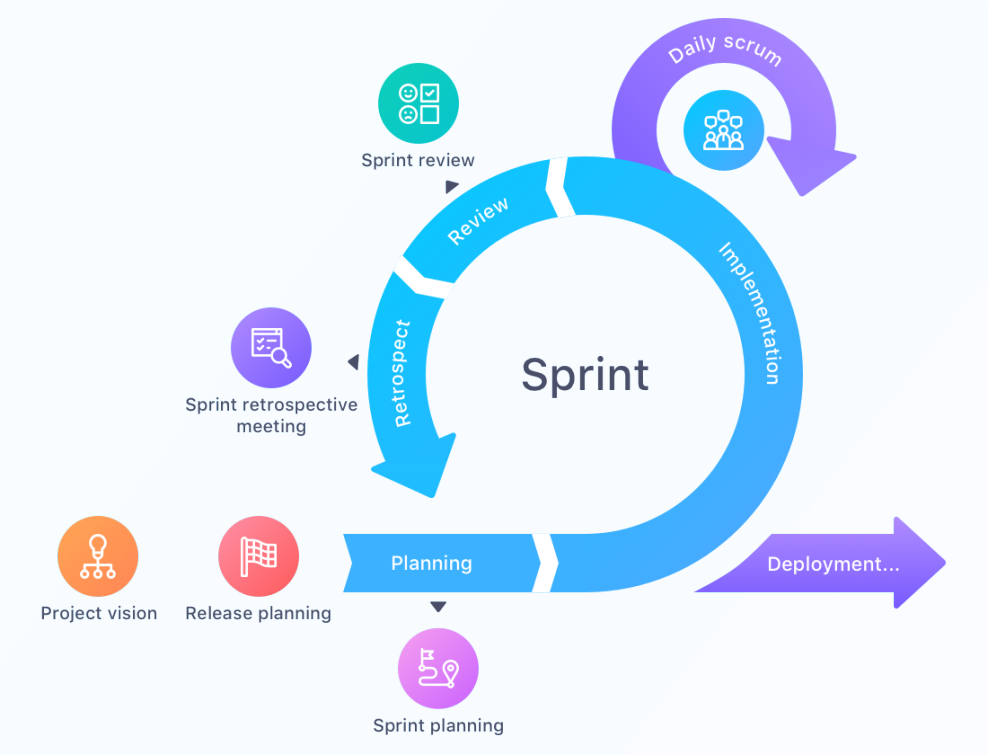

Figura 4.2: Proceso seguido por Scrum, imaxe obtida de Scrum.org [2].

Roles

Nos proxectos que utilizan Scrum diferencianse dous roles a maiores do equipo de desenvolvemento, o *Product Owner* e o *Scrum Master*, que van a encargarse de tarefas fundamentais para o seguimento do proxecto.

O *Product Owner* é o encargado de priorizar unha lista de requisitos/obxectivos recollidos no *Product Backlog* e de revisar se se cumpriron ao final de cada *sprint*.

O *Scrum Master* o que se encargue de que o equipo non se desvíe dos seus obxectivos durante o seguimento do *sprint* e elimine os obstáculos que o equipo non poida arranxar de por si.

Eventos

Antes de comezar cada *sprint*, convocarase unha reunión co cliente de aproximadamente 2 horas para especificar os requisitos ou obxectivos seleccionados do *Product Backlog* a satisfacer na iteración. Logo, ao comezar, realizarase unha reunión de tamén 2 horas para definir as pautas do propio *sprint*. Estas dúas reunións mencionadas son as que forman parte do *Sprint planning*.

Son necesarias en Scrum reunións diarias (*Daily scrum*) de 15 minutos para a comunicarse

co resto do equipo e saber en todo momento o estado do proxecto coa fin dunha monitorización continua.

Ao final de cada *sprint* haberá unha reunión de revisión (*Sprint review*) de hora e media onde o cliente analizará se se alcanzou o obxectivo final do propio *sprint*. En adición, tamén se realizará unha reunión de retrospectiva (*Sprint retrospective meeting)*, da mesma duración ca reunión de revisión, na que se valorarán os procesos seguidos na iteración e as posibles melloras para os seguintes.

Os tempos marcados neste apartado están recomendados para *sprints* de dúas semanas.

4.1.2 Adaptación da metodoloxía ao proxecto

Como este proxecto é un traballo individual e Scrum é unha metodoloxía aplicada a grupos de traballo, son necesarias certas modificacións para poder adaptala ao proxecto.

Entre as modificacións feitas encóntranse:

- Eliminación das reunións diarias, pois ao haber só un integrante, non hai necesidade de sincronizar o estado do proxecto con máis membros.
- Ao non haber un cliente real, é o alumno do proxecto o que toma o control por este de tarefas como priorizar os obxectivos/requisitos do *Product Backlog* e valorar se o obxectivo do *sprint* se cumpriu nos *Sprint reviews*.
- Os *Scrum Masters* son os directores do proxecto, Álvaro Barreiro e Javier Parapar, debido a que son eles quenes evitan as desviacións do alumno no proxecto.
- Debido ao carácter técnico do proxecto, o *Product Backlog* non segue a representación de historias de usuario senon que se especifican os requisitos ad hoc.

4.2 Xestión do proxecto

A xestión do proxecto consiste en seguir unha planificación para o inicio, seguimento e finalización deste tendo en conta os esforzos e custo que o propio proxecto ten e asumindo e controlando os problemas que poidan surxir durante o seu desenvolvemento.

Entre as restriccións que limitan un proxecto destacan as do "Triángulo de calidade": tempo, custo e alcance. Estas tres dependen en gran medida unhas das outras.

A norma ISO 21500 proporciona unha guía que define unha serie de conceptos e de boas prácticas para unha xestión de proxectos eficaz nas organizacións. Esta norma segue o esquema dun dos certificados estándar do *Project Management Institute* (PMI), o PMBoK.

Segundo PMBok, os procesos están descritos a través de entradas, ferramentas e técnicas para producir salidas (documentos, deseños, etc.). A continuación describiranse os 5 procesos polos que un proxecto debe pasar segundo as directrices de PMBoK.

4.2.1 Análise de viabilidade

Nesta fase decídese se un proxecto debe de seguir adiante ou non analizando os recursos, o tempo e custo do mesmo.

- Recursos
  - Humanos: En canto se refire aos recursos humanos contarase co equipo de desenvolvemento, a alumna Jennifer Dubra, e cos directores do proxecto, Álvaro Barreiro e Javier Parapar.
  - Materiais: Os recursos materiais que se empregarán serán propiedade da alumna ou proporcionados polos directores (GitLab e Taiga, por exemplo).
- Tempo

  Debido a que o único membro do equipo realizará o proxecto ao mesmo tempo que cursa as materias do último curso e traballando a media xornada, o tempo asignado ao desenvolvemento do proxecto non será constante pois depende da carga de traballo da alumna. Intentarase pois adicar 10 horas á semana ao proxecto na medida do posible.
- Custo

  Segundo o Infojobs (a bolsa de emprego online especializada no comercio español entre outras) na súa guía para traballar no sector IT[13], calculan o salario medio dun programador no 2017 nuns 35.870€/ano, e a través de Indeed[3], o salario medio ao ano en España dun *Scrum Master* é de 37.526€. Posto que no ano de realización deste proxecto foi no 2019, o cal conta con 250 días laborables, o custo por hora dos distintos roles sería o da Táboa 4.2.

Rol	**Custo**
Equipo de desenvolvemento	14,4€/h
Scrum Master	18,76€/h

Táboa 4.2: Custo por hora dos roles do proxecto.

  Supoñendo que cada *sprint* ten unha duración de aproximadamente 20 puntos de historia, espérase que os *Scrum Masters* adiquen 2,5h por cada iteración (o tempo dedicado aos *Daily Scrums*).

  O tempo estimado para a duración total do proxecto fixouse nos 198 puntos de historia (aproximadamente 20 semanas dedicándolle 10h á semana e correspondéndose un pun-

[3] https://www.indeed.es/salaries/scrum-master-Salaries

to de historia a unha hora), o custo dos recursos humanos sería 3.780€ como se indica na Táboa 4.3.

Rol	Tempo adicado	Custo total
Equipo de desenvolvemento	198h	2.851,2€
*Scrum Master*x2	24,75h x 2	464,38€ x 2 = 928,778€
		3.780€

Táboa 4.3: Estimación do custo total dos recursos humanos.

En canto aos recursos materiais só se dispón dun ordenador pois non é necesario investir en licencias de software xa que as que se usan son de código libre. Tendo en conta que o valor da computadora é de 1.000€ e a súa vida útil é de 3 anos, se se utiliza durante 20 semanas, o custo deste recurso será de 138,9€.

O custo total dos recursos deste proxecto será, como se representa na Táboa 4.4, de 3.919€.

Recursos	Custo total
Humanos	3.780€
Materiais	138,9€
	3.919€

Táboa 4.4: Estimación do custo total dos recursos.

4.2.2 Planificación detallada do traballo a realizar

Nesta fase procurase detallar as tarefas a realizar polo equipo de desenvolvemento. De non se definir con claridade pode repercutir de forma grave no proxecto pois o alcance do proxecto non queda claro.

As tarefas correspondentes a este apartado están definidas na sección de Obxectivos, Sección 1.2, seguindo unha liña de desenvolvemento similar ao estudo realizado por David Grefen, Jake Miller et al.[14] sobre a identificación de patróns nos rexistros médicos a través do análise semántico latente:

1. Indexación dos documentos nun índice invertido.
2. Implementación das funcionalidades de acceso e procura.
3. Obtención da similitude entre documentos e termos para logo realizar o *clustering* de ambos.

4. Deseño da interfaz gráfica coas funcionalidades implementadas do software.

5. Aplicación de técnicas de LSI ao corpus de documentos indexados.

4.2.3 Execución do proxecto

Durante este período ponse en práctica as distintas técnicas e formas de xestión de recursos e procesos, tamén coñecido como o *know how*. Isto especificarase máis adiante na sección de Desenvolvemento do proxecto cos detalles da estruturación e implementación.

4.2.4 Seguimento e control do traballo

Esta fase, en paralelo coa anterior, é das máis importantes na xestión de proxectos. Nesta etapa é onde se comproba que se está seguindo a planificación e a calidade do produto.

Taiga é a ferramenta que fixo posible o seguimento proporcionando unha plataforma web onde establecer os *sprints*, o *Product Backlog* e visualizar de forma gráfica o avance e desviacións do proxecto. Na Figura 4.3 pódese observar o seguimento e control levado no proxecto a través das iteracións, intentando seguir a planificación detallada na subsección 4.2.2. Na sección de Desenvolvemento, ao igual que a execución, tamén se especificarán as decisións de control e seguimento tomadas a través das iteracións.

4.2.5 Peche do proxecto

Nesta última fase avaliaranse e verificanse os resultados obtidos. Tamén se analizan os resultados finais cos resultados estimados para saber cales foron as fortalezas e fraquezas do proxecto e actuar en consecuencia en proxectos similares futuros.

Como nas últimas dúas fases, os resultados deste proxecto avaliaranse en seccións posteriores.

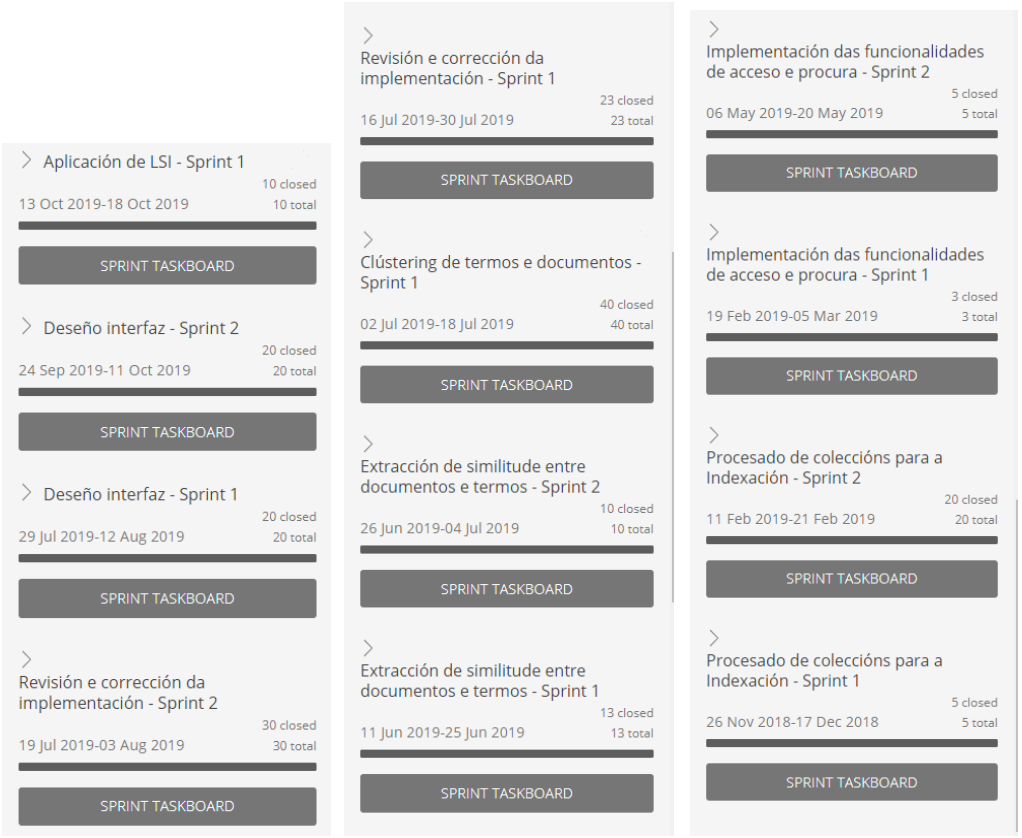

Figura 4.3: *Sprints* do proxecto.

Capítulo 5

Desenvolvemento

Este capítulo está dedicado a detallar o desenvolvemento do proxecto a través do tempo. Definiranse os requisitos funcionais e non funcionais a satisfacer así como a arquitectura escollida que mellor se adapta ao proxecto. Nesta sección entón, completaranse os detalles das fases 3 e 4 mencionadas no apartado de xestión do proxecto (Sección 4.2).

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.

Un requisito ben formulado debe de posuir as seguintes características:

- Descrito en linguaxe natural: Os requisitos deben estar descritos en linguaxe natural.
- Non ambiguo: Debe ser claro e preciso.
- Alcanzable: Un requisito ten que poder realizarse e estar adaptado ao custo do proxecto e o tempo que se estima dedicar.
- Verificable: Débese poder avaliar con certeza cando un requisito está finalmente satisfeito.
- Consistente co resto de requisitos: Un requisito non pode entrar en conflito cos outros.

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.

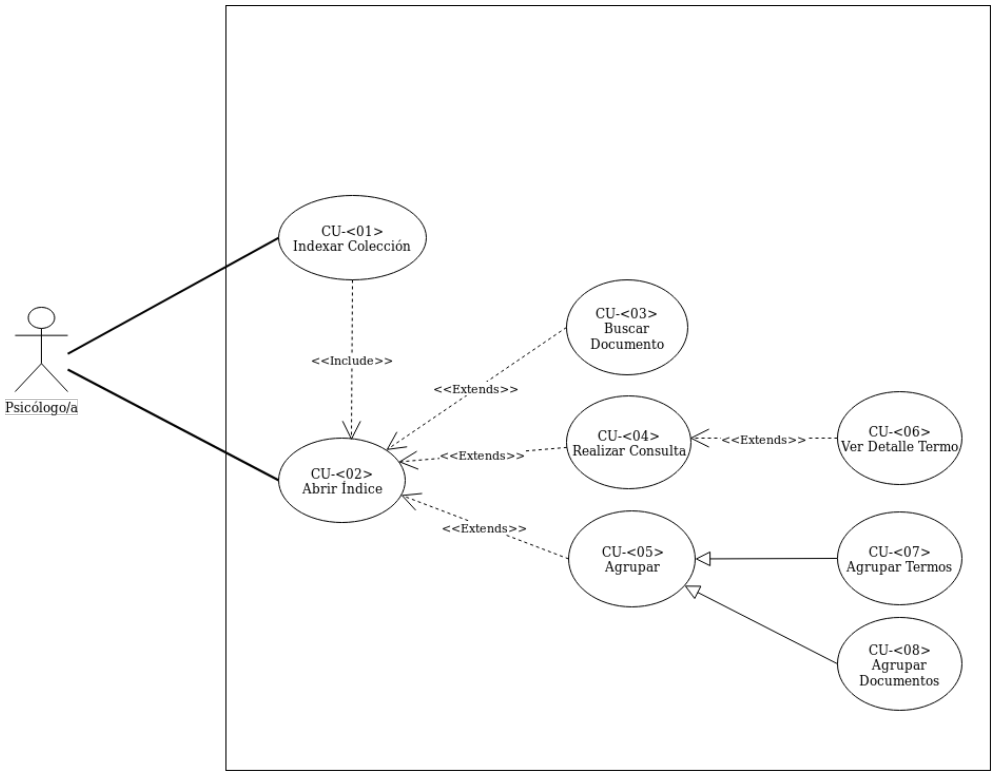

Figura 5.1: Casos de uso do sistema.

O único actor deste proxecto será o rol de psicólogo pois é el quen interactuará co software desenvolvido. Quedará representado nos UMLs como *Psicólogo/a.*

Casos de uso

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

CU-<01>	**Indexar Colección**
Descrición	A persoa usuaria debe ser capaz de indexar unha colección de documentos.
Precondición	Os documentos han de ter formatos compatibles coa aplicación.
Secuencia normal	1 O sistema ofrece a opción de indexar unha colección. 2 A persoa usuaria introduce a dirección da colección. 3 O sistema indexa e mostra as estadísticas dos documentos.
Postcondición	O índice debe quedar gardado na memoria.
Importancia	Alta.
Comentarios	Os documentos han de seguir o formato presentado pola Listaxe 5.1.

Táboa 5.1: Caso de uso CU-<01>.

CU-<02>	**Abrir Índice**
Descrición	A persoa usuaria debe poder abrir un índice que ten gardado en memoria.
Precondición	O índice ten que existir no directorio especificado.
Secuencia normal	1 O sistema ofrece a opción de abrir un índice. 2 A persoa usuaria introduce a dirección onde ten gardado un índice. 3 O sistema mostra as estadísticas do índice.
Postcondición	Non aplica.
Importancia	Alta.
Comentarios	Nas estadísticas do índice tamén se mostran os N termos máis frecuentes da colección.

Táboa 5.2: Caso de uso CU-<02>.

CU-<03>	**Buscar Documento**	
Descrición	A persoa usuaria poderá buscar un documento polo seu identificador ou poder visualizar a colección documento a documento.	
Precondición	Debe estar indexado, polo menos, un documento.	
Secuencia normal	1	A persoa usuaria indexa unha colección ou abre un índice.
	2	O sistema ofrece a posibilidade de acceder a un documento a través do seu identificador e campo.
	3	A persoa usuaria introduce o campo e o identificador do documento que desexa recuperar.
	4	O sistema devolve o contido do campo indicado do documento especificado.
Postcondición	Non aplica.	
Importancia	Media.	

Táboa 5.3: Caso de uso CU-<03>.

CU-<04>	**Realizar Consulta**	
Descrición	A persoa usuaria poderá lanzar unha consulta ao índice.	
Precondición	Non aplica.	
Secuencia normal	1	A persoa usuaria indexa unha colección ou abre un índice.
	2	O sistema ofrece a posibilidade de realizar unha consulta.
	3	A persoa usuaria introduce a consulta e especifica cantos documentos quere que sexan devoltos como máximo.
	4	O sistema devolve os documentos relevantes para a consulta.
Postcondición	Non aplica.	
Importancia	Alta.	
Comentarios	Pode non haber resultados.	

Táboa 5.4: Caso de uso CU-<04>.

CU-<06>	**Ver Detalle Termo**	
Descrición	A persoa usuaria debe poder visualizar o detalle dun termo.	
Precondición	O termo ten que estar presente nos documentos recuperados dunha consulta.	
Secuencia normal	1	A persoa usuaria indexa unha colección ou abre un índice, dispara o caso de uso CU-<04> e realiza unha consulta.
	2	O sistema ofrece a opción de ver os detalles de un termo que pertenza a algún dos documentos recuperados.
	3	A persoa usuaria escolle un termo.
	4	O sistema devolve o detalle correspondente ao termo escollido pola persoa usuaria.
Postcondición	Non aplica.	
Importancia	Media	

Táboa 5.5: Caso de uso CU-<06>.

CU-<07>	**Agrupar Termos**	
Descrición	A persoa usuaria debe poder visualizar a agrupación dos N termos máis semellantes a unha palabra escollida.	
Precondición	O termo polo que se realiza a agrupación debe aparecer nalgún documento da colección.	
Secuencia normal	1	A persoa usuaria indexa unha colección ou abre un índice.
	2	O sistema ofrece a posibilidade de agrupar por termos.
	3	A persoa usuaria indica: a representación dos termos, o termo polo que buscar, os N termos máis semellantes, o número de grupos a formar e se se realiza redución de dimensionalidade.
	4	O sistema mostra de forma gráfica os N termos máis semellantes agrupados.
Postcondición	Os resultados deben quedar representados graficamente.	
Importancia	Alta	
Comentarios	Este caso de uso é unha especificación do caso de uso CU-<05>.	

Táboa 5.6: Caso de uso CU-<07>.

CU-<08>	**Agrupar Documentos**	
Descrición	A persoa usuaria debe poder visualizar a agrupación dos M documentos máis semellantes a un documento escollido.	
Precondición	Deben de haber máis de un documentos indexados.	
Secuencia normal	1	A persoa usuaria indexa unha colección ou abre un índice.
	2	O sistema ofrece a posibilidade de agrupar por documentos.
	3	A persoa usuaria indica: a representación dos documentos, o documento polo que buscar, os M documentos máis semellantes, o número de grupos a formar e se se realiza redución de dimensionalidade.
	4	O sistema mostra de forma gráfica os M documentos máis semellantes agrupados.
Postcondición	Os resultados deben quedar representados graficamente.	
Importancia	Alta	
Comentarios	Este caso de uso é unha especificación do caso de uso CU-<05>.	

Táboa 5.7: Caso de uso CU-<08>.

5.1.2 Requisitos non funcionais

Os requisitos non funcionais son aqueles que se esperan de forma implícita das características que software. Este tipo de requisitos divídense en dous grupos: aqueles que definen as cualidades da execución (seguridade, usabilidade, etc.) e os que definen as cualidades da evolución (testeo, mantemento, escalabilidade, etc.). Dos requisitos non funcionais que estean definidos para a elaboración dun produto software vai depender a arquitectura do sistema.

Os requisitos non funcionais deste proxectos están documentados nas Táboas 5.8, 5.9, 5.10, 5.11 e 5.12.

RNF-<01>	**Flexibilidade**
Descrición	O sistema deberá poder dar soporte a distintos formatos de entrada sen que o rendemento se vexa afectado.
Importancia	Alta

Táboa 5.8: Requisito non funcional RNF-<01>.

RNF-<02>	**Eficacia**
Descrición	Os resultados do sistema deberán ser acordes aos que o usuario espera obter.
Importancia	Alta

Táboa 5.9: Requisito non funcional RNF-<02>.

RNF-<03>	**Eficiencia**
Descrición	O software deberá de funcionar de forma apropiada con grandes volumes de datos.
Importancia	Alta

Táboa 5.10: Requisito non funcional RNF-<03>.

RNF-<04>	**Mantemento**
Descrición	O sistema deberá ser fácil de conservar en bo estado para evitar a súa degradación.
Importancia	Alta

Táboa 5.11: Requisito non funcional RNF-<04>.

RNF-<05>	**Usabilidade**
Descrición	A aplicación deberá ser fácil de usar e intuitiva por parte dos usuarios finais.
Importancia	Alta

Táboa 5.12: Requisito non funcional RNF-<05>.

5.2 Arquitectura proposta

A arquitectura dun proxecto define a estrutura fundamental na que se desenvolve e caracteriza un produto software. É fundamental dedicarlle o tempo necesario a súa elección pois, unha vez comezado o desenvolvemento é difícil de cambiar.

Os requisitos non funcionais son os que definen, en gran medida, a escolla da arquitectura. Debido aos requisitos non funcionais especificados neste proxecto na Sección 5.1.2, decidiuse seguir a arquitectura MVC (Modelo-Vista-Controlador) cuxos compoñentes están definidos na Táboa 5.13.

	Función	**Acción sobre os outros compoñentes**
Modelo	Xestión da información. Encárgase de almacenar e traballar cos datos.	Actualiza a vista
Controlador	Medio de comunicación da vista co modelo.	Observa os eventos da vista e realiza as peticións correspondentes ao modelo.
Vista	Representación gráfica do modelo cos respectivos métodos de interacción.	

Táboa 5.13: Descrición dos compoñentes da arquitectura MVC.

O modelo de datos neste proxecto corresponde ao índice invertido, cuxos métodos proporciona Lucene (Sección 3.3.1), xunto coas funcionalidades implementadas para a satisfacción dos requisitos funcionais como a indexación, procura, acceso e agrupamento.

A arquitectura MVC favorece o mantemento por mor da separación de responsabilidades. Con isto, tamén permite múltiples implementacións de vistas para un mesmo modelo de datos. Ao haber separación de responsabilidades é doado para un grupo de desenvolvemento traballar de forma simultánea en varios módulos, neste proxecto non é o caso pero deberíase ter en conta.

Na Figura 5.2 pódese observar a estruturación do código en Eclipse conforme a arquitectura MVC correspondendo cada módulo a un dos compoñentes da mesma.

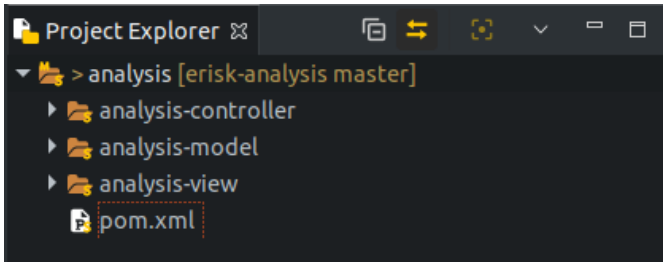

Figura 5.2: Estrutura do proxecto en Eclipse.

5.3 Desenvolvemento

A continuación detallarase o transcurso do proxecto no desenvolvemento das tarefas descritas na planificación detallada do traballo a realizar da Sección 4.2.2.

5.3.1 Indexación dos documentos nun índice invertido

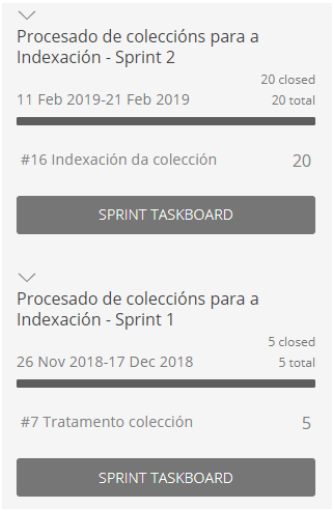

Figura 5.3: *Sprints* 1 e 2.

Nestas primeiras iteracións examináronse os documentos das coleccións de Reddit, que foron proporcionadas polos directores do proxecto, para poder procesalos axeitadamente. Estes documentos seguen o formato XML, como se mostra na Listaxe 5.1. Tras a súa análise concluíronse como campos relevantes o identificador da persoa usuaria do texto (IDs) máis os seus escritos (TEXTs).

```
<INDIVIDUAL>
    <ID>train_subject136</ID>
    <WRITING>
```

```
4       <TITLE> Titulo do post </TITLE>
5       <DATE> 2013-04-05 20:36:55 </DATE>
6       <INFO> reddit post </INFO>
7       <TEXT> Primero texto escrito polo usuario train_subject136
      </TEXT>
8     </WRITING>
9     <WRITING>
10      <TITLE>   </TITLE>
11      <DATE> 2013-04-05 20:10:09 </DATE>
12      <INFO> reddit post </INFO>
13      <TEXT> Segundo texto escrito polo usuario train_subject136
      </TEXT>
14    </WRITING>
15
16    ...
17
18    <WRITING>
19      <TITLE>   </TITLE>
20      <DATE> 2013-04-05 20:10:09 </DATE>
21      <INFO> reddit post </INFO>
22      <TEXT> N texto escrito polo usuario train_subject136 </TEXT>
23    </WRITING>
24  </INDIVIDUAL>
```

Listaxe 5.1: Formato dos documentos de entrada.

Pódese observar na Figura 5.4 o proceso polo que son sometidos os documentos dunha colección para que finalmente queden almacenados nun índice invertido.

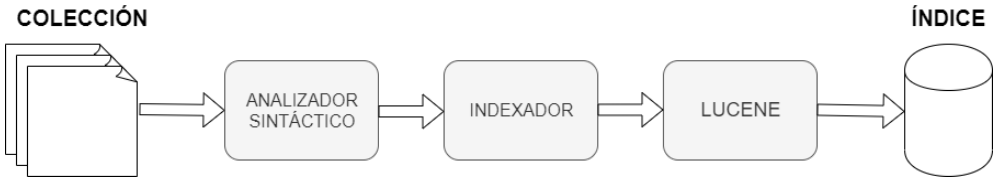

Figura 5.4: Procesado da colección para a súa indexación.

O analizador sintáctico (tecnicamente coñecido como *parser*), é o encargado de descompoñer, a través dunha serie de regras, os documentos para extraer a información dos campos que os conforman. É imprescindible que o *parser* estea separado do resto do código que se encarga da indexación para un un baixo acoplamento e maior cohesión que produce que sexa máis fácil adaptar o proxecto aos distintos tipos de entradas.

O indexador é o órgano que se encarga de recoller a información extraída polo analizador sintáctico e estruturala en entradas para a API de Lucene (*Document*s compostos por *Field*s) como se viu con anterioridade na Figura 3.1. É o indexador tamén quen establece a configu-

ración do índice (como as opcións de escritura ou as listas de *stopwords*) e quen decide a ruta onde se garda o mesmo.

Os dous sprints que se mostran na Figura 5.3, foron nos que se implementaron o analizador sintáctico da colección máis o indexador que conecta con Lucene, satisfacendo o caso de uso CU-<01> (Táboa 5.1).

5.3.2 Implementación das funcionalidades de acceso e procura

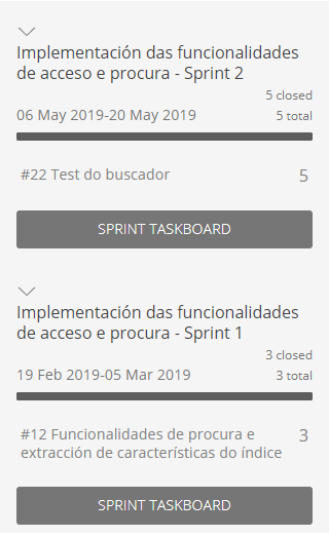

Figura 5.5: *Sprints* 3 e 4.

Nos sprints que se amosan na Figura 5.5, implementáronse as funcionalidades de acceso aos documentos a través dos seus identificadores e de consulta ao índice.

Nótese que houbo un salto temporal considerable entre os *sprint*s de indexación e de acceso e procura debido aos exames propios do primeiro cuadrimestre do cuarto curso de grao na cal a estudante non puido dedicarlle tempo ao proxecto. Tamén é amplo o espazo entre os dous *sprint*s de acceso e procura producido pola alta carga de entregas das materias do segundo cuadrimestre.

A maiores tamén se crearon os métodos para extraer as características propias dos termos e da colección como os *tf* s, *tf-idf* s, o número de documentos da colección, a data de creación do índice, etc.

Para comprobar o correcto funcionamento do buscador creáronse tests que o corroboren.

Na Figura 5.6 móstrase o diagrama do proceso de procura ou consulta no índice.

Rematadas estas dúas iteracións quedan satisfeitos os casos de uso CU-<02>, CU-<03> e CU-<04> (Táboas 5.2, 5.3 e 5.4 respectivamente).

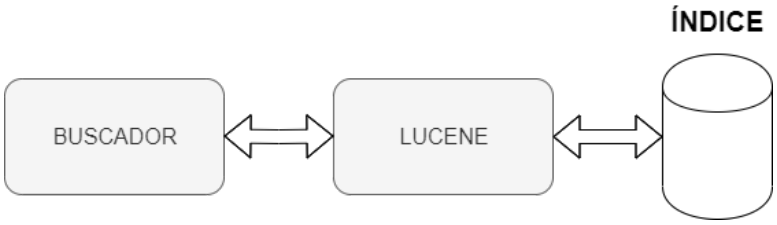

Figura 5.6: Proceso de procura no índice.

5.3.3 Obtención da similitude entre documentos e termos para logo realizar o *clustering* de ambos

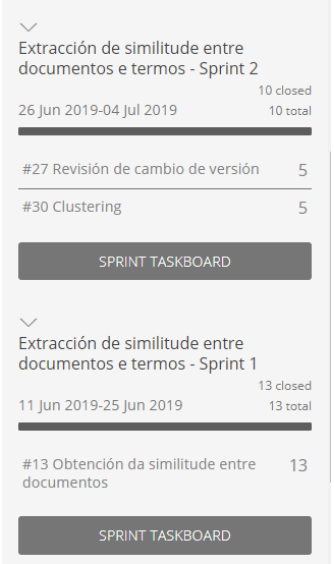

Figura 5.7: *Sprints* 5 e 6.

Nas dúas iteracións que se mostran na Figura 5.7 intentouse proceder coa funcionalidade de agrupamento de termos e documentos a través da librería de Apache Mahout. Apache Mahout é unha librería que proporciona implementacións de algoritmos de aprendizaxe automático enfocados principalmente en álxebra lineal.

Realizouse entón un estudo da propia librería e procedeuse a realización do agrupamento de todos os documentos da colección e termos a través da súa similitude, como se mostran nas subtarefas da tarefa 13 (Figura 5.8).

Presentouse o inconveniente de que foi necesario cambiar a versión de Lucene que se utilizaba no proxecto a unha anterior para que a librería de Mahout fose compatible. Isto produciu un gasto en tempo na revisión das funcionalidades xa implementadas debido a que foron necesarios realizar cambios nelas porque algunhas funcións non estaban dispoñibles na nova versión.

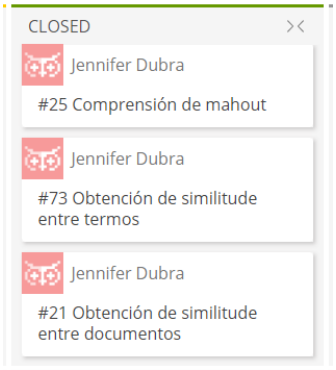

Figura 5.8: Subtarefas da tarefa 13.

Durante o proceso de desenvolvemento con Mahout, encontrouse a librería Math3 de Apache Commons. Debido que a complexidade de Mahout e a simpleza que presentaba esta última librería, decidiuse abandonar o proceso de *clústering* de Mahout para utilizar Math3 (Sección 3.3.2).

Figura 5.9: *Sprint* 7.

O proceso de comprensión do algoritmo KMeansPlusPlusCentered foi bastante máis áxil pois traballa con obxectos Java con distintos atributos, onde un deles debe de ser as coordenadas do termo ou documento no espazo vectorial. O algoritmo KMeans++ devolve os obxectos Java que se pasaron como entrada repartidos nos K grupos indicados.

Como é evidente, antes deste proceso de agrupamento de documentos e termos é necesario extraer os N termos ou documentos máis parecidos debido a que os requisitos do sistema son realizar o agrupamento dos N termos/documentos máis semellantes un termo/documento en particular.

Lucene xa aporta a funcionalidade para obter os N documentos máis similares a outro coa función *like* da clase Java *MoreLikeThis*.

En cambio para obter os termos máis similares a outro, implementouse unha función que devolvese a similitude coseno entre o resto de termos co termo que se procura nunha lista

ordeada descendentemente. Desta lista recolleranse entón os N primeiros termos.

Na Figura 5.9 móstrase o *sprint* onde se dá por finalizada a implementación dos proceso de agrupamentos de termos e documentos segundo a súa similitude cumprindo os casos de uso CU-<05>, CU-<07> (Táboa 5.6) e CU-<08> (Táboa 5.7).

Ata este momento estívose implementando sobre o compoñente modelo da arquitectura Modelo-Controlador-Vista, quedando estruturado o proxecto da forma na que se mostra na Figura 5.10.

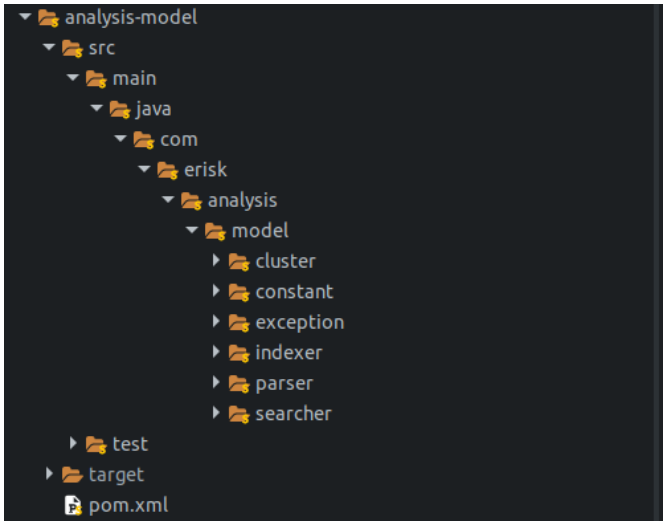

Figura 5.10: Compoñente modelo do proxecto.

5.3.4 Deseño da interfaz gráfica

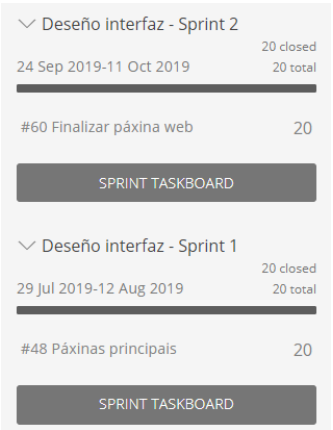

Figura 5.11: *Sprints* 8 e 9.

Tal e como se pode observar na Figura 5.11, estas iteraccións estiveron dedicadas á elabo-

ración da interfaz gráfica do proxecto, é dicir, ao compoñente Vista da arquitectura escollida.

No análise para a elaboración da interfaz gráfica web extraeuse a funcionalidade de ofrecer a posibilidade de configuración por parte do usuario das listas de *stopwords* usar na indexación dunha colección (a estándar ou a personalizada para a linguaxe inglesa).

No primeiro *sprint* centrouse en facer o deseño das páxinas principais e o segundo en crear o último compoñente da arquitectura, o controlador. O controlador é quen lle pasa á vista os parámetros necesarios e quen invoca os métodos do modelo, realiza a función de intermediario. As Figuras 5.12 e 5.13 corresponden aos módulos de vista e controlador do proxecto.

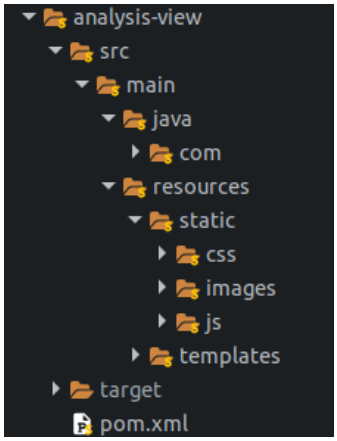

Figura 5.12: Compoñente vista do proxecto.

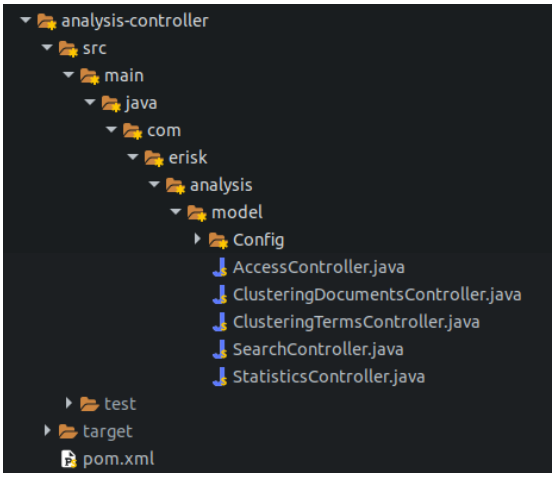

Figura 5.13: Compoñente controlador do proxecto.

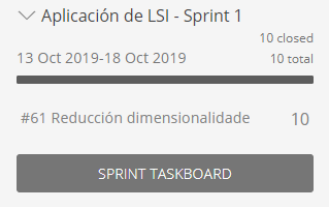

Figura 5.14: *Sprint* 10.

5.3.5 Aplicación de técnicas LSI.

Como xa se explicou anteriormente na Sección 2.4.1, SVD é unha aplicación de LSI que procura a redución de dimensionalidade.

Nesta última iteración, Figura 5.14, foi onde se engadiron as técnicas de *Latent Semantic Indexing*. A mesma librería que se usou para o agrupamento, Math3, consta da función SingularValueDecomposition que realiza a descomposición das matrices que se pasan como parámetros.

Á interfaz gráfica engadiuse a opción de utilizar este recurso ou non para extraer os termos máis semellantes ao vocábulo indicado para a realización do agrupamento de termos.

Finalizado este *sprint* e deuse por concluída a implementación do proxecto.

5.4 Balance do proxecto

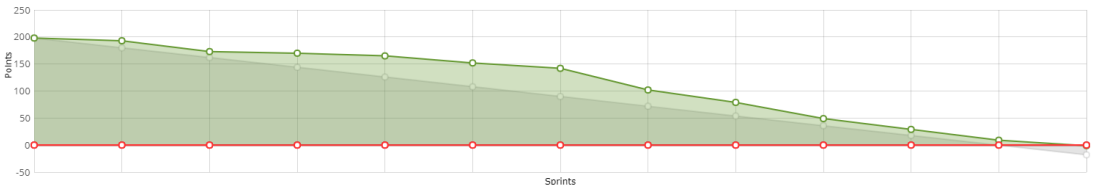

Figura 5.15: Seguimento do proxecto.

Na Figura 5.15 vese o seguimento do proxecto a través do tempo. A área máis escura sería o seguimento dun proxecto ideal e a verde o seguimento do proxecto real para os 198 puntos de historia indicados.

A primeira metade do proxecto transcorreu de forma lenta debido a alta carga de traballo extra da alumna debido a que se encontraba traballando e cursando o último ano de carreira. A segunda metade, en cambio, ao rematar o curso académico dedicóuselle maior tempo ao proxecto o que fixo que se acercasen mellor as horas invertidas ás horas estimadas nun desenvolvemento normal e que a desviación fose menor.

O proxecto foi subestimado pero, aínda así, a desviación coa que finalizou entra dentro duns rangos aceptables.

Capítulo 6

Resultados

Neste capítulo avaliarase a eficiencia do software. Tamén se mostrarán algúns exemplos gráficos dos resultados da aplicación que poderían ser de interese para os usuarios da mesma.

6.1 Avaliación

Nesta sección examinarase a eficiencia dos algoritmos de indexación, agrupamento e LSI implementados neste proxecto. Coñécese a eficacia dos tres algoritmos presentados pero considerouse relevante presentar a eficiencia dos mesmos para comprobar que o software non inflúe na propia dos algoritmos de indexación, agrupamento e LSI.

O software execútase en local nunha computadora coas características que se mencionan na Táboa 6.1.

Procesador	Intel Core i5-8250 CPU @ 1.60GHz 1.80GHz
RAM	8GB

Táboa 6.1: Características da máquina.

6.1.1 Eficiencia de indexación

Na Táboa 6.2 obsérvanse distintas mostras tempos na execución do algoritmo de indexación implementado neste proxecto para obter unha aproximación dos *megabytes* que se indexan por segundo. A eficiencia do algoritmo de indexación cambia en función do analizador que se use, xa que na aplicación están implementadas as opcións para utilizar o analizador estándar (*StandardAnalyzer*) ou o propio do idioma inglés (*EnglishAnalyzer*) examinaranse a eficiencia de ambos.

Pódese ver na Táboa 6.2 que o algoritmo de indexación implementado co analizador estándar tarda 918,7 milisegundos en indexar unha colección de 48,9 megabytes, o cal conclúe ca métrica de indexación de 53,27 MB/s. En cambio, a indexación co analizador para o idioma inglés custa aproximadamente 1011,66 milisegundos en indexar a mesma colección, o que quere decir que se indexan 48,34 MB/s.

Execución	***StandardAnalyzer*[ms]**	***EnglishAnalyzer*[ms]**
1	925	997
2	923	1024
3	908	1014
Media	918,7	1011,66

Táboa 6.2: Tempos de indexación do proxecto para unha colección de 48,9 megabytes cos analizadores *StandardAnalyzer* e *EnglishAnalyzer*

.

6.1.2 Eficiencia de agrupamento

Nesta subsección móstranse os tempos de agrupamento do algoritmo co cal se fixo o agrupamento de termos e documentos, o K-Means++. Na Táboa 6.3, pódense ver os tempos do proceso de *clustering* variando o número de grupos e mostras para ver como afectan cada un destes cambios. De haber máis de un algoritmo de agrupamento utilizado no proxecto, poderían ser útiles tales tempos para poder realizar comparacións e ver en qué casos cada un deles funcionan mellor.

Número de mostras	**Grupos**	**Tempo [ms]**
100	5	6
100	10	10
1000	5	92
1000	10	220

Táboa 6.3: Tempos de agrupamento en 1030 dimensións.

.

6.1.3 Eficiencia de LSI

A continuación, na Táboa 6.4 móstranse os tempos de execución do algoritmo de SVD implementado por Math3. Pódese comprobar que ten uns tempos de execución moi altos cando as dimensións da matriz que descompón son grandes, o cal se presenta como un gran inconvinte porque se pretende usar neste software coleccións de documentos grandes.

Dimensións da matriz	Tempo [ms]
30.731×1030	3.995.161
26.389×830	2.593.681
19.839×200	63.067

Táboa 6.4: Tempos de LSI.

.

6.2 Exemplos do software

O obxectivo principal deste proxecto, como se mencionou no primeiro capítulo, é proporcionar soporte aos especialistas da saúde mental para o etiquetado de cada suxeito. Para isto, as funcionalidades de agrupación de termos ou documentos semellantes xogan un papel moi importante.

A agrupación dos documentos dá unha idea sobre as relacións que hai entre os documentos máis semellantes a o que o usuario da aplicación proporcionou para a comparativa. Coa agrupación de termos os resultados son máis sinxelos de captar debido a que é máis fácil asimilar as relacións entre palabras que entre documentos.

A dirección deste proxecto proporcionou distintas coleccións de documentos de persoas diagnosticadas con diferentes trastornos mentais para poder probar o software. As coleccións que se proporcionaron foron: un conxunto de documentos do 2017 de pacientes que foron diagnosticados de anorexia e dúas coleccións (2017 e 2018) de textos escritos por persoas con depresión. Os textos están recollidos de Reddit como anteriormente se mencionou, así que hai que ter en conta os posibles erros ortográficos.

A continuación introdúcense algúns dos exemplos máis visuais dos distintos resultados do software nas diferentes coleccións de documentos proporcionadas polos directores do proxecto.

Os resultados serán obxectivos, deixando na man dos especialistas da saúde mental as interpretacións dos mesmos.

6.2.1 Colección de documentos de suxeitos con anorexia (2017)

Esta colección conta con 200 documentos e un total de 19.859 termos tras o filtrado de *stopword*s. Pese a ser un conxunto de datos reducido, pódense extraer resultados interesantes.

Pódese ver nas Figuras 6.1, 6.2 e 6.3 os distintos resultados de procura e agrupación dos 25 termos máis semellantes a *self* en 7 grupos coas distintas representación dos vocábulos. Nas diferentes imaxes obsérvanse as variacións das palabras máis semellantes e como iso afecta a organización dos grupos. Posto que o tf-idf indica que tan importante é un termo nunha colección, móstrase un exemplo na Figura 6.4 dos resultados do proceso de agrupación tras

terlle aplicado as técnicas de reducción de dimensionalidade.

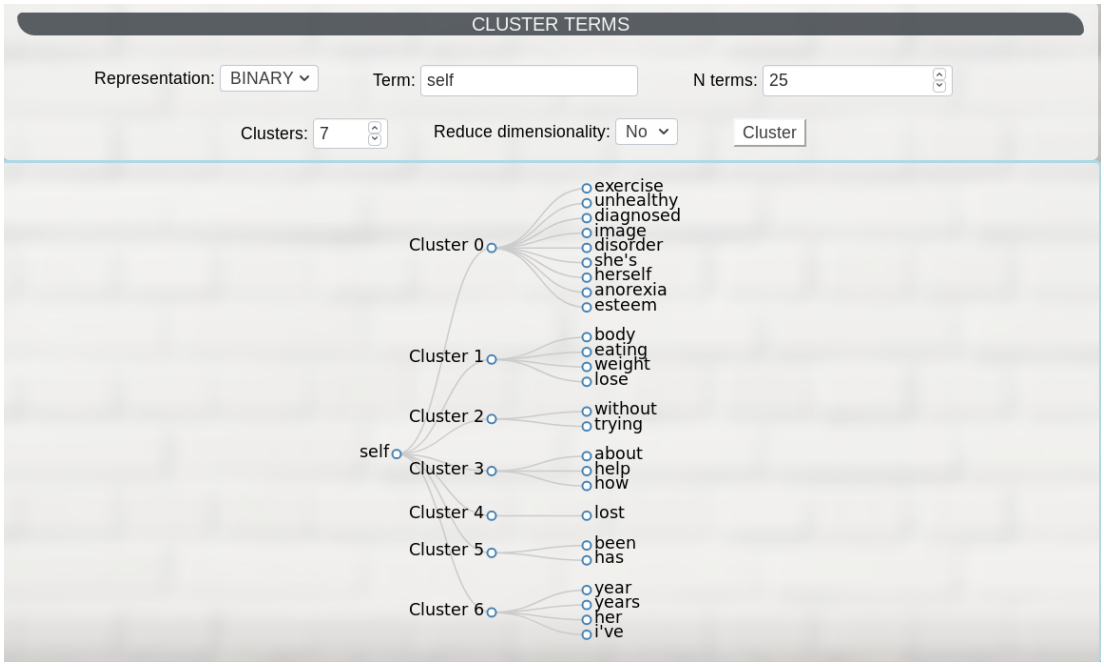

Figura 6.1: Agrupación dos termos máis semellantes a *self* con representación binaria.

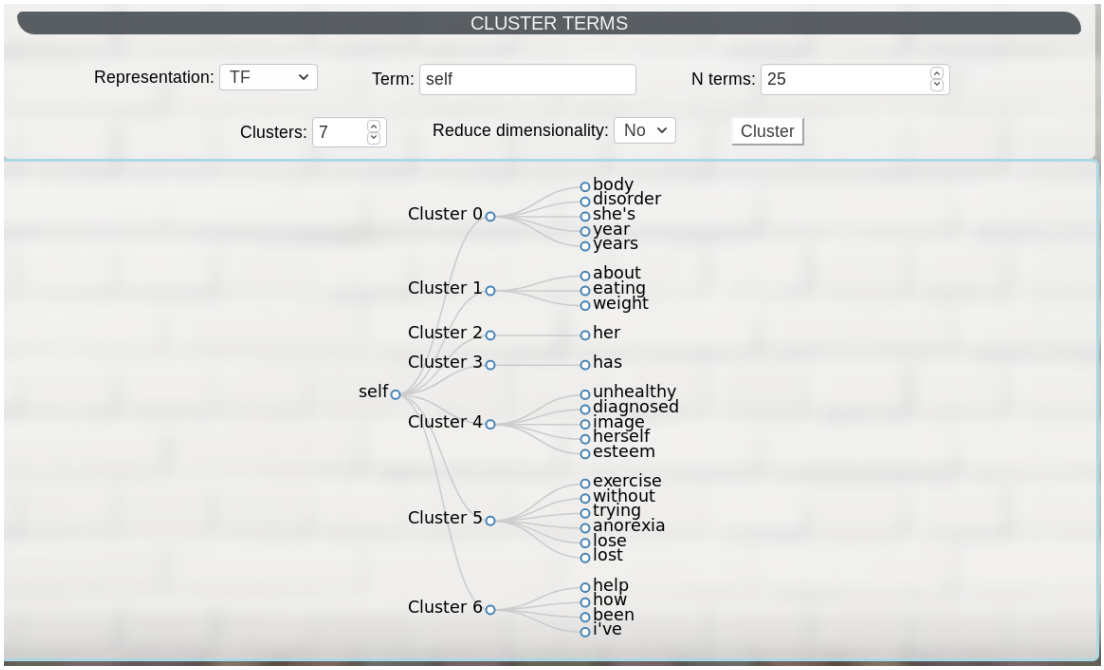

Figura 6.2: Agrupación dos termos máis semellantes a *self* con representación TF.

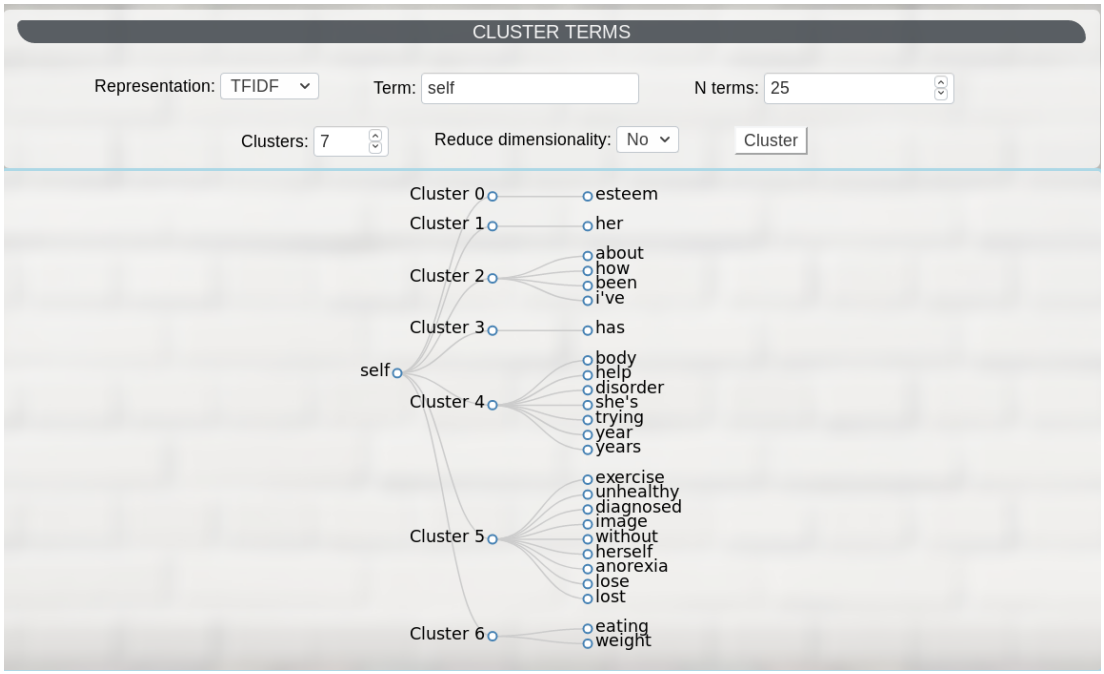

Figura 6.3: Agrupación dos termos máis semellantes a *self* con representación TFIDF.

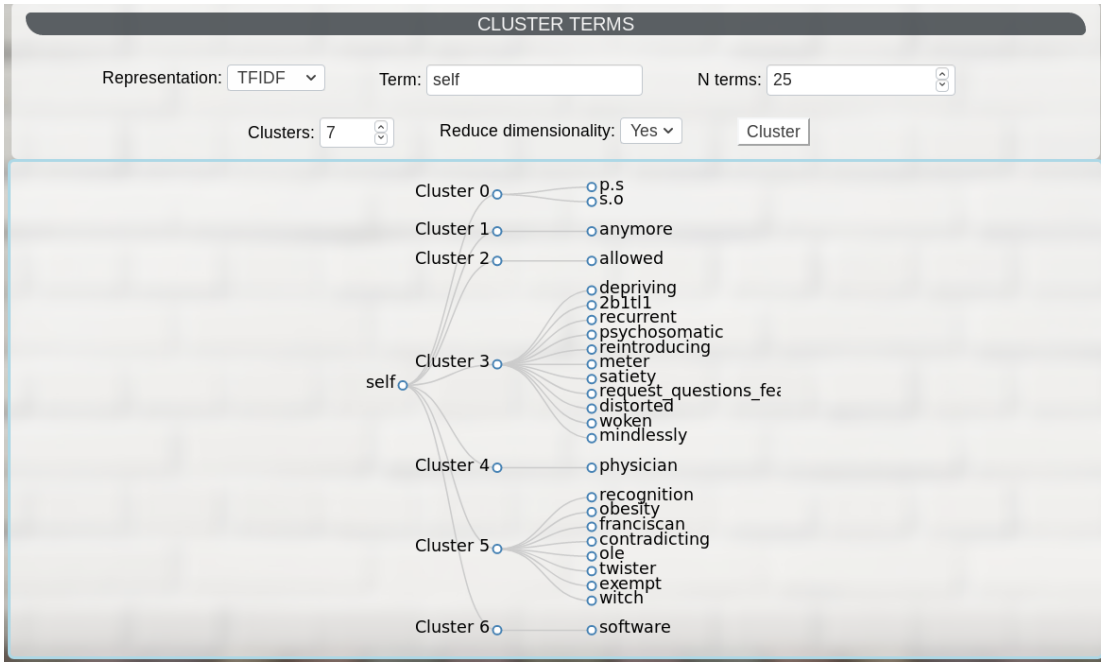

Figura 6.4: Agrupación dos termos máis semellantes a *self* con representación TFIDF e reducción de dimensionalidade.

6.2.2 Coleccións de documentos de suxeitos diagnosticados con depresión (2017 e 2018)

Nesta subsección decidiuse mostrar os resultados de procesar as dúas coleccións, as de 2017 e 2018, de forma conxunta para obter resultados máis representativos. O índice consta de, en conxunto, 1.030 documentos e un total de 30.834 termos tras o filtro de *stopword*s.

Posto que é unha colección de datos máis ampla ca de anorexia, pódense identificar mellor as relacións dos termos recuperados entre sí e coa palabra pola que se procura.

No exemplo que se mostra na Figura 6.5 móstranse as palabras máis relacionadas con *sleep* (durmir). Nun análise superficial dos resultados destacan os *cluster*s 1, 3 e 4: o grupo 1 é moi interesante debido a que o magnesio é un suplemento alimenticio natural que se utiliza cada vez máis para conciliar o sono, o único termo do grupo 3 trátase dun fármaco antidepresivo que combate o insomnio e o tamén único vocábulo do grupo 4, é a hormona de resposta ao estres.

Con estes resultados pódese deducir que os suxeitos con depresión cando falan sobre durmir é moi probable que falen de temas relacionados co insomnio.

Na Figura 6.6 é de interese os termos máis relacionados cando se falan de *they* (eles ou elas), son vocábulos como ”cazar”, ”burlar”, ”calumnias”, ”máis feliz/felices”.

Cando se procura pola palabra *useless* (inútil), Figura 6.7, as palabras recuperadas están case todas relacionadas cos videoxogos. Tamén ao procurar por *weirdo* (bicho raro), Figura 6.8, destacan termos relacionados coa hixiene persoal.

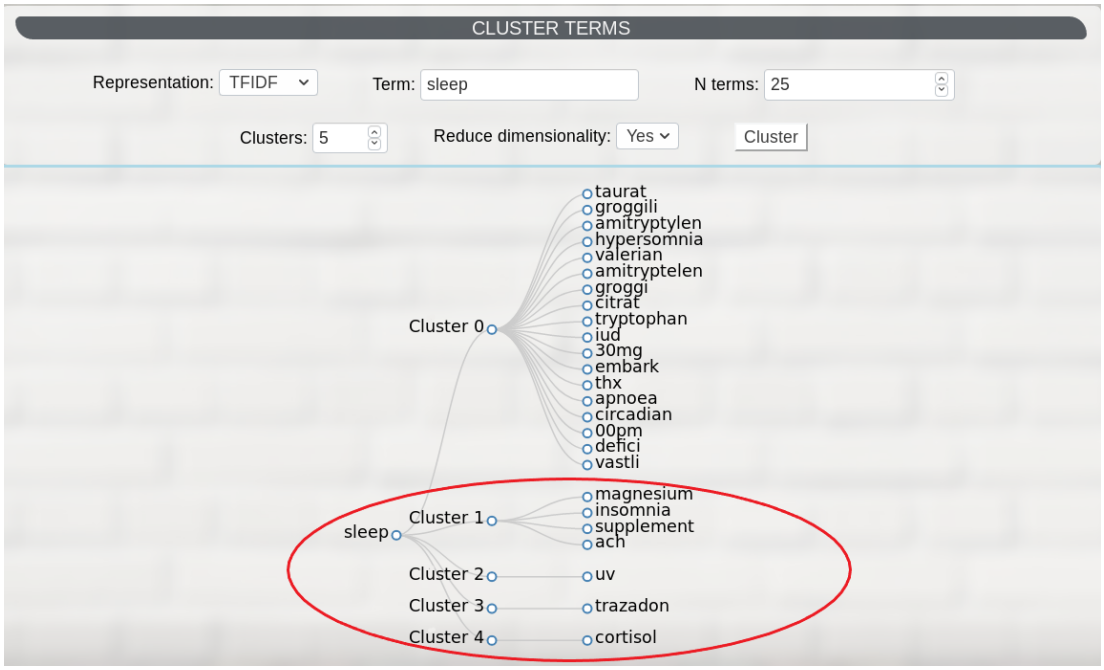

Figura 6.5: Agrupación dos termos máis semellantes a *sleep* con representación TFIDF e reducción de dimensionalidade.

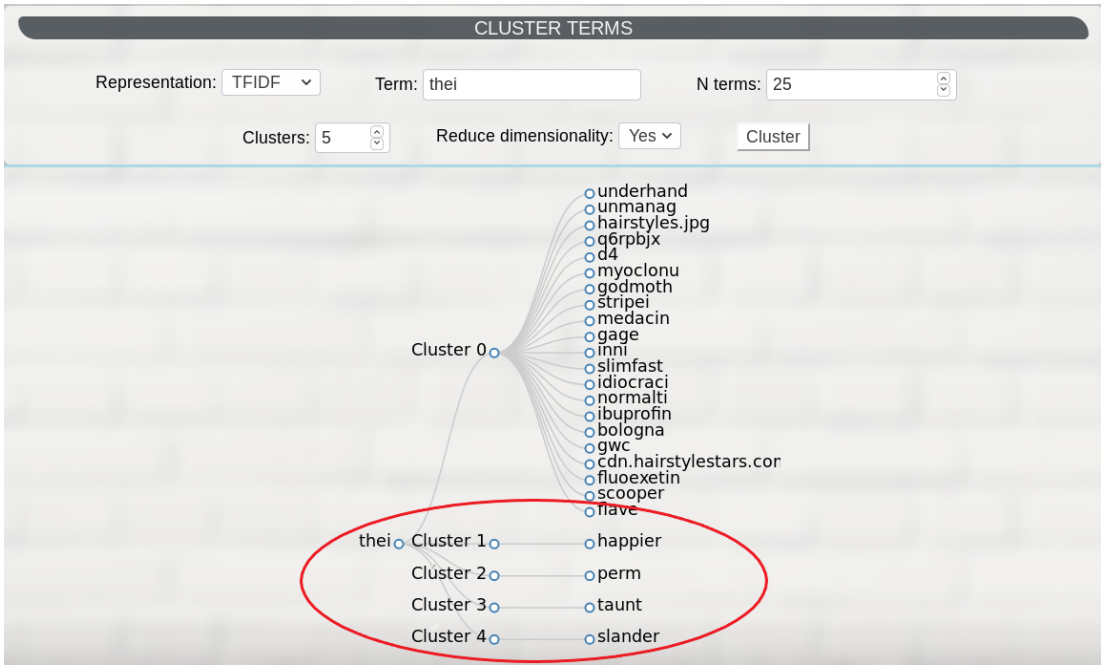

Figura 6.6: Agrupación dos termos máis semellantes a *they* con representación TFIDF e reducción de dimensionalidade.

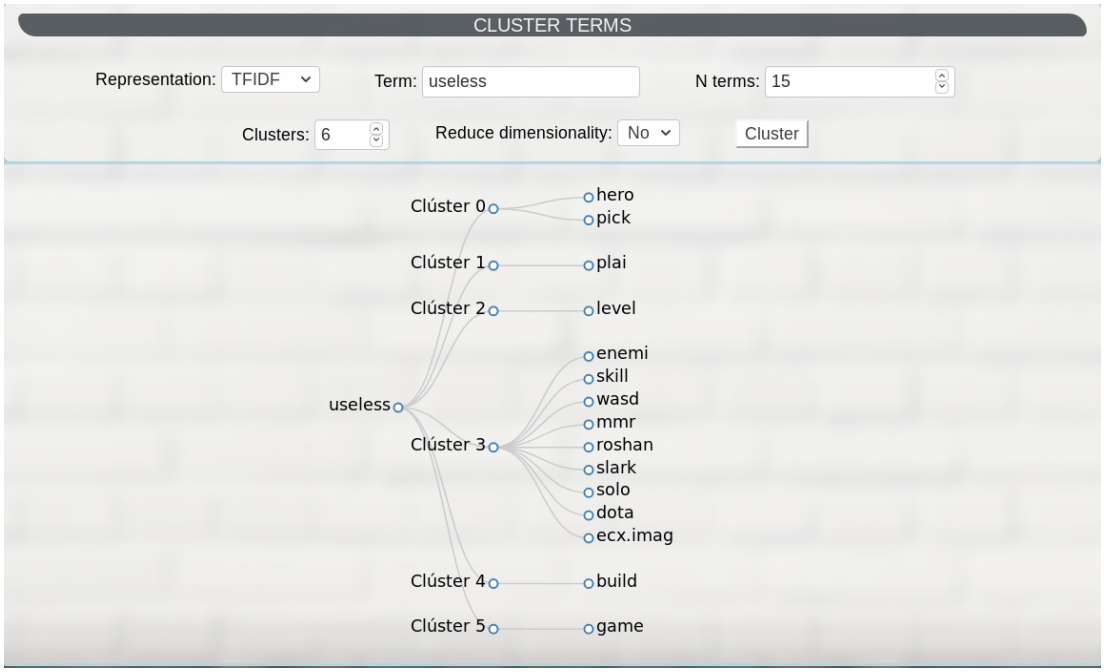

Figura 6.7: Agrupación dos termos máis semellantes a *useless*.

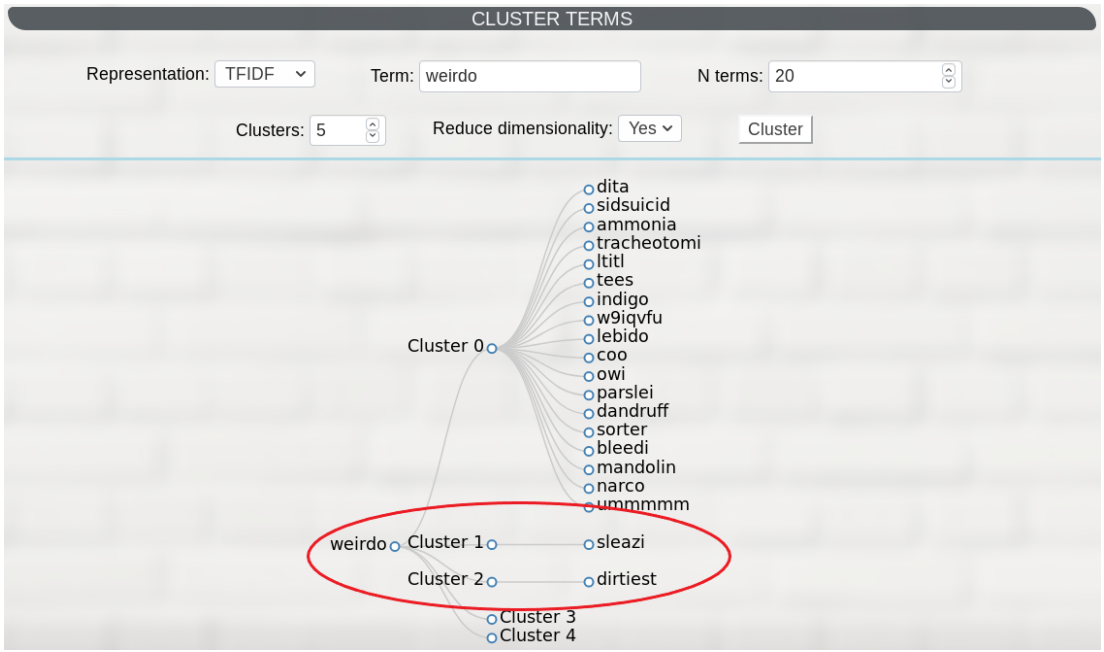

Figura 6.8: Agrupación dos termos máis semellantes a *weirdo* con representación TFIDF e reducción de dimensionalidade.

Capítulo 7

Conclusións e futuras liñas de traballo

Sendo este o capitulo derradeiro, presentaranse as conclusións finais deste proxecto así como as posibles liñas de traballo futuro.

7.1 Conclusións

É necesario poñer énfase nos métodos de prevención secundaria dos trastornos mentais xa que se sabe que unha detección a tempo pode significar unha maior posibilidade de éxito dos tratamentos e incluso unha restauración completa.

Cos resultados extraídos da aplicación, os especialistas da saúde mental poderán poñer en práctica os seus coñecementos da materia. Poderán buscar, consultar ou interpretar os conxuntos de datos que consideren relevantes da maneira máis oportuna.

É un feito que toda axuda en este ámbito da saúde é necesaria e o software implementado neste proxecto pode ofrecerlles aos especialistas relacións de suxeitos que padecen certas enfermidades con temas que poderían non ser evidentes a simple vista.

Este proxecto é o traballo a longo prazo por excelencia do grao nos cales se puido poñer en práctica os procesos de enxeñería e desenvolvemento software estudados ata o momento. A nivel teórico, indagouse en materias de recuperación de información que non se coñecían previamente como é o caso da indexación semántica latente. A nivel práctico, da mesma maneira, utilizáronse tecnoloxías que non foron ensinadas na mención de Computación tales como Springboot e Thymeleaf.

O proxecto tamén fomentou o traballo individual nas que se viron melloradas as capacidades de autocrítica e xestión do tempo. En poucas palabras, este traballo ensinou a *aprender a aprender*.

7.2 Traballo futuro

Durante o transcurso do proxecto consideráronse diferentes liñas de traballo a facer que non se realizaron por falta de tempo entre as que se encontran:

- *Multithreading*, creación de varios fíos de execución para aquelas operacións máis custosas como son a indexación, agrupamento ou técnicas de redución de dimensionalidade. Planteouse a posibilidade de realizalos de forma concorrente e paralela para estudar os diferentes resultados e tempos da aplicación.
- Ofrecer ao usuario varios métodos de agrupamento alternativos ao xa implementado, o K-Means++, para poder comparar os distintos resultados e tempos.
- Mellorar o *parser* para que poida permitir máis fontes de entrada.
- Mellorar o algoritmo de consulta para que procese *queries* máis elaboradas.
- Aplicar AJAX para unha mellor experencia de usuario.

Apéndices

Apéndice A

Manual de usuario

Neste apartado inclúese un manual de usuario no que se indica como usar a aplicación.

A.1 Páxina principal

Na páxina principal as dúas únicas opcións son as de indexar unha colección ou abrir un índice, como se mostra nas Figura A.1 e A.2. Tanto o índice como a colección indicaranse a través da ruta absoluta do sistema de ficheiros.

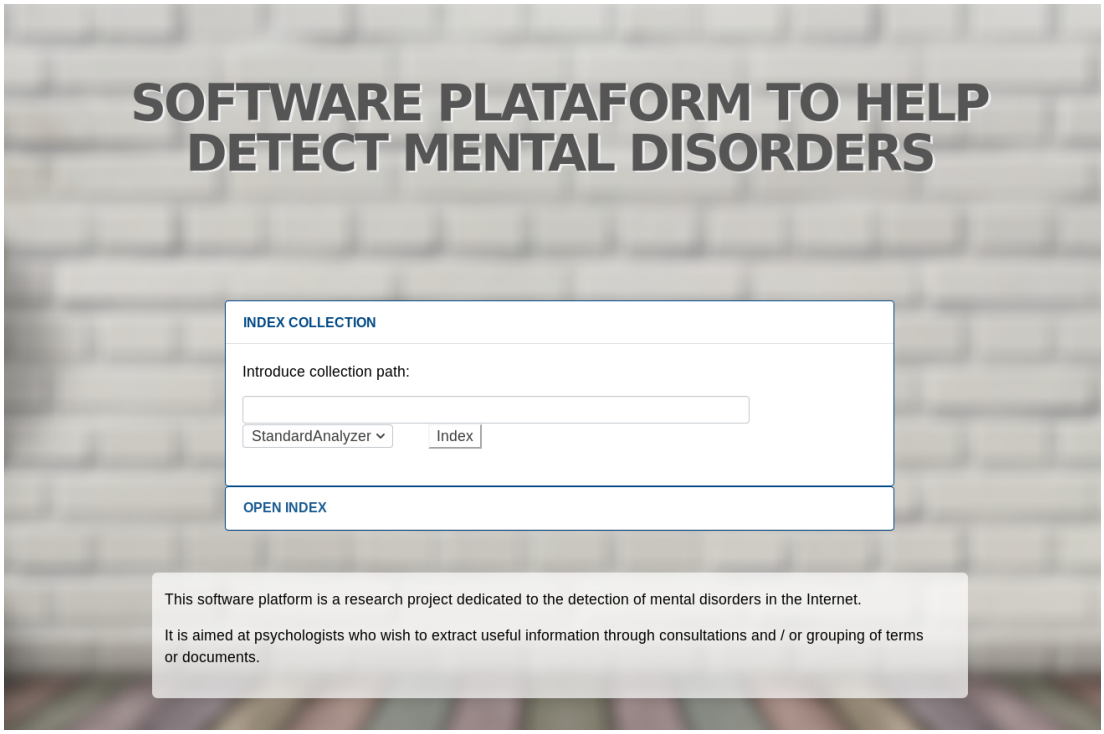

Figura A.1: Indexar colección.

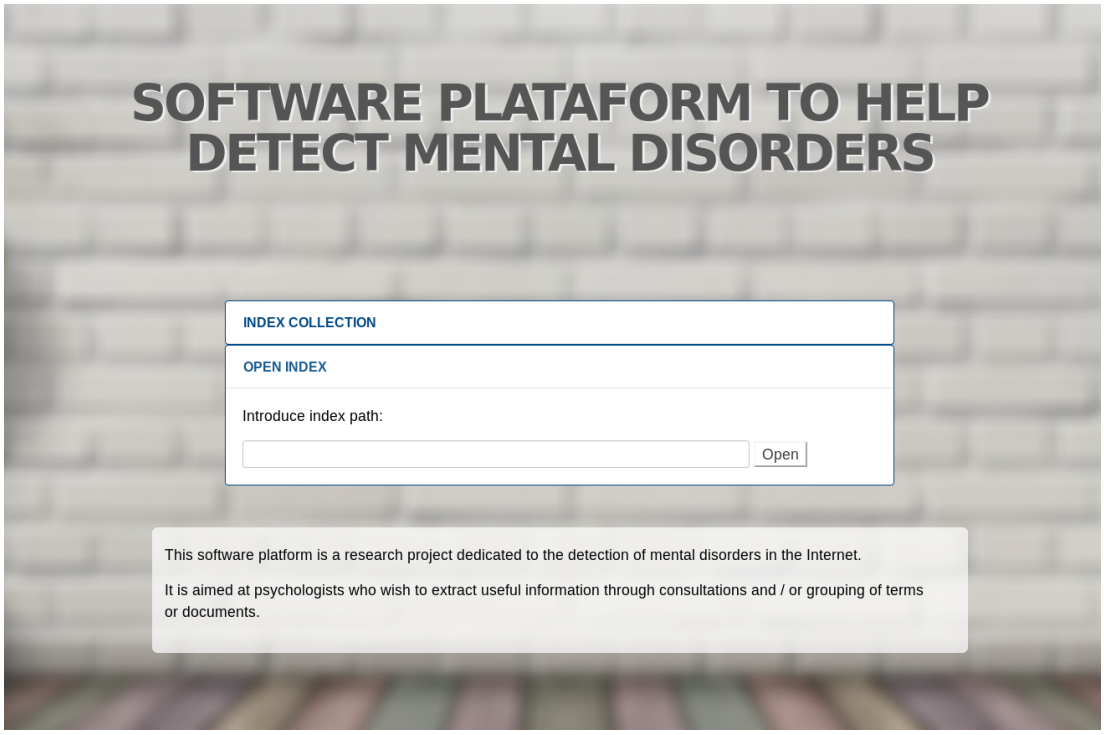

Figura A.2: Abrir colección.

Pódese observar na Figura A.3 que se ofrecen as varias opcións de analizador a utilizar.

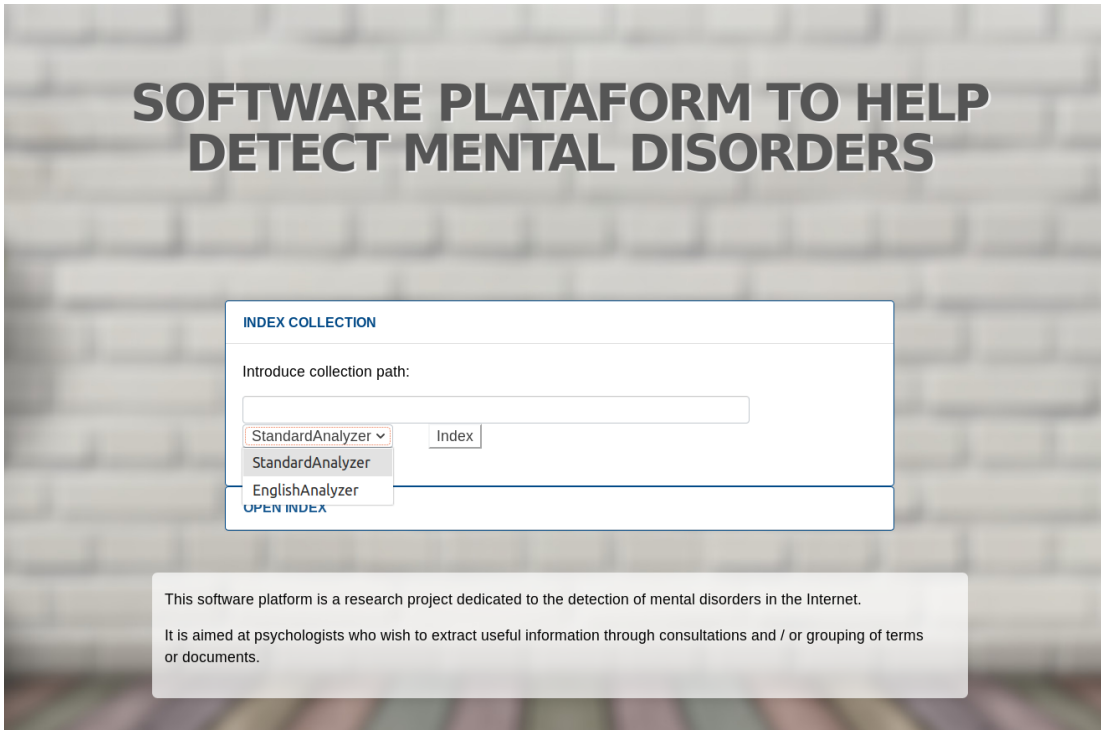

Figura A.3: Opcións de análise da colección.

A.2 Estadísticas

Na pantalla de estadísticas (Figura A.4) pódense observar os datos da colección indexada e un formulario no que se pode visualizar os Top-N dos termos que máis veces aparecen no campo escollido, como se mostra no exemplo da Figura A.5.

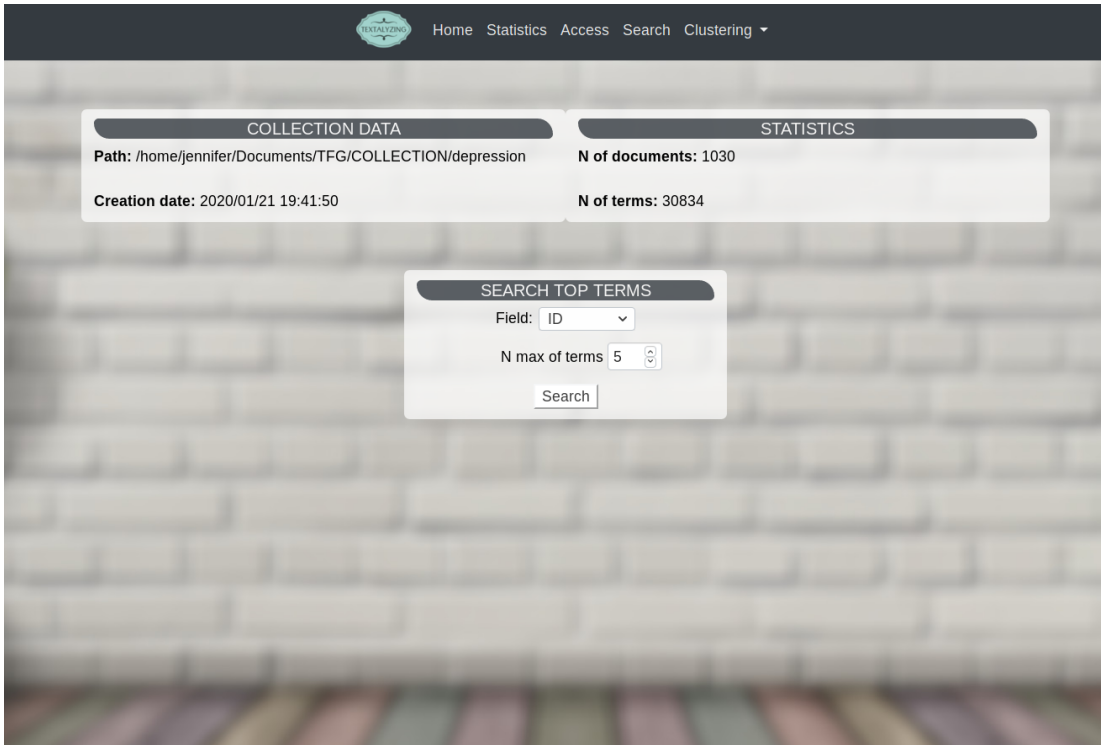

Figura A.4: Pantalla de estadísticas.

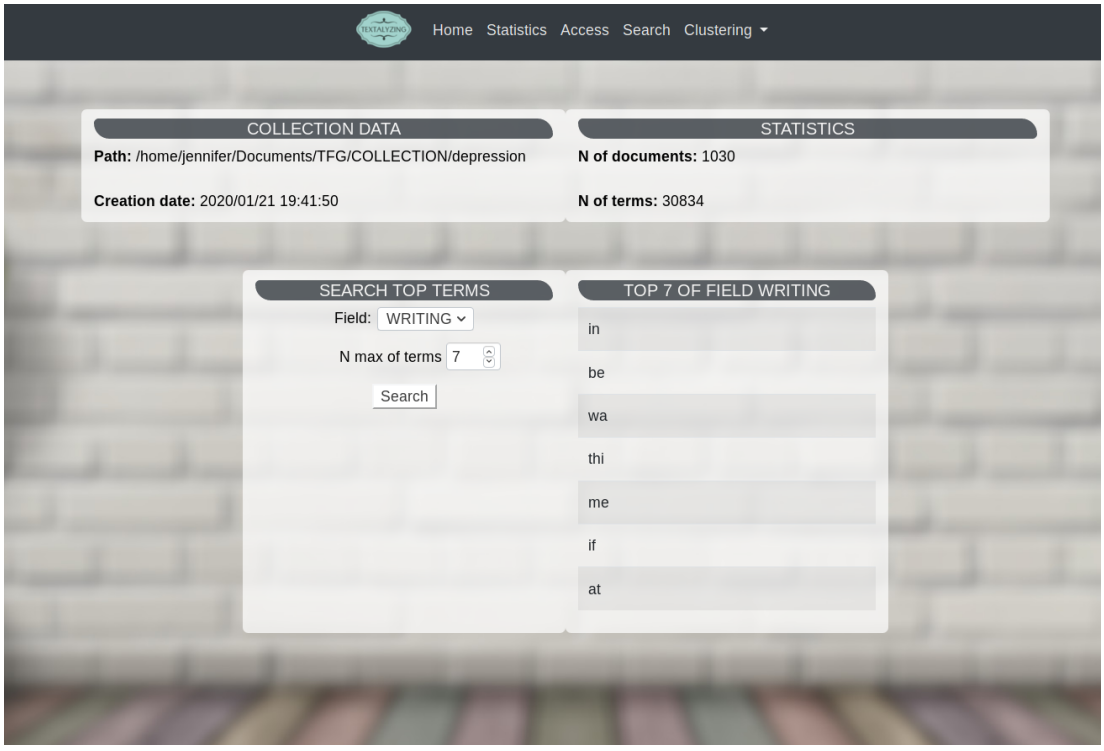

Figura A.5: Top 7 dos termos que máis aparecen no campo WRITING.

A.3 Acceso

A pantalla de acceso (Figura A.6) permite navegar a través da colección examinando o contido dos campos dos documentos, tal e como se pode observar na Figura A.7.

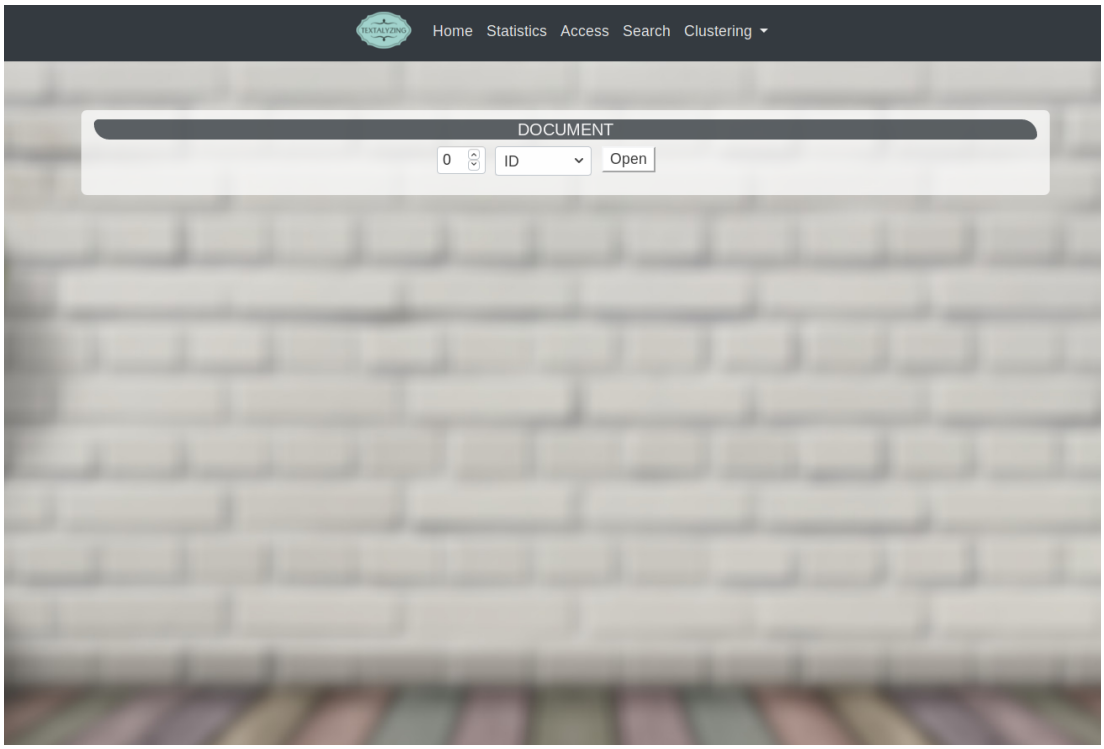

Figura A.6: Pantalla de acceso aos documentos.

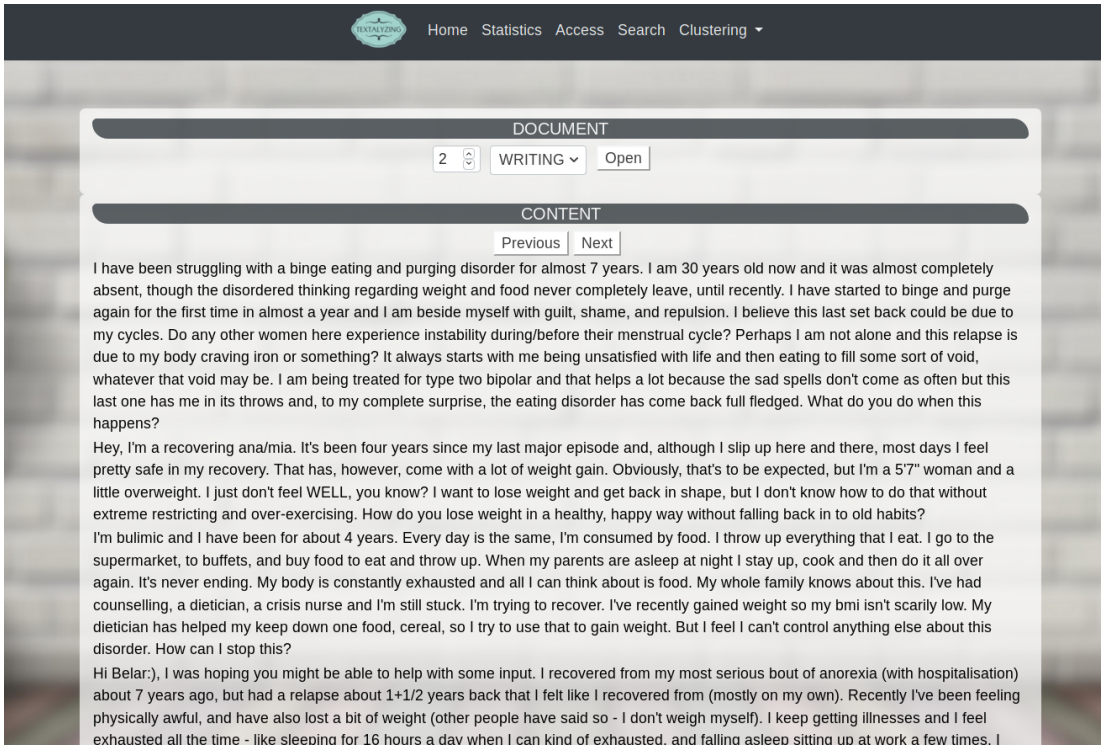

Figura A.7: Recuperación do campo WRITING do documento 2.

A.4 Procura

Na pantalla de procura (Figura A.8) pódese facer unha consulta permitindo moldear cantos resultados se queren recuperar como máximo, exemplo da Figura A.9. Os resultados serán o identificador do documento máis a súa puntuación con respecto á consulta.

Pódese observar na Figura A.10 que se poden escoller as palabras que forman parte dos documentos recuperados para ver os seus detalles. No documento en que se procura o detalle serve só para contar o número de veces que aparece o termo buscado no mesmo. Preséntase na Figura A.11 un exemplo desta funcionalidade.

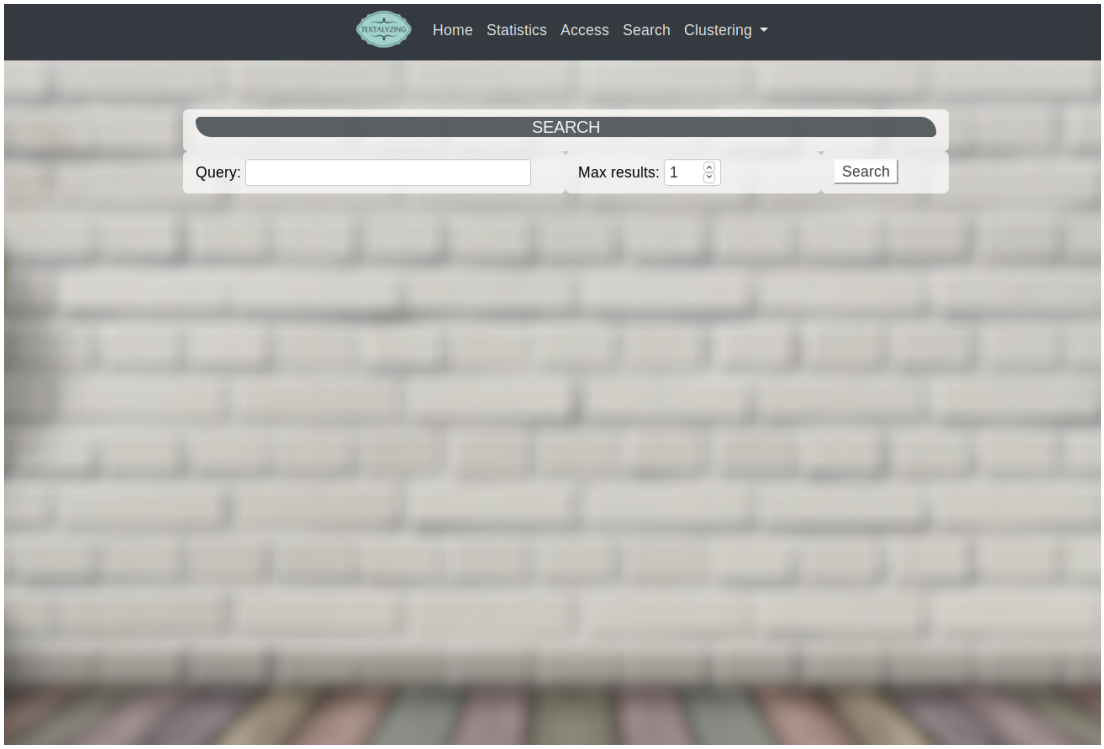

Figura A.8: Pantalla de procura no índice.

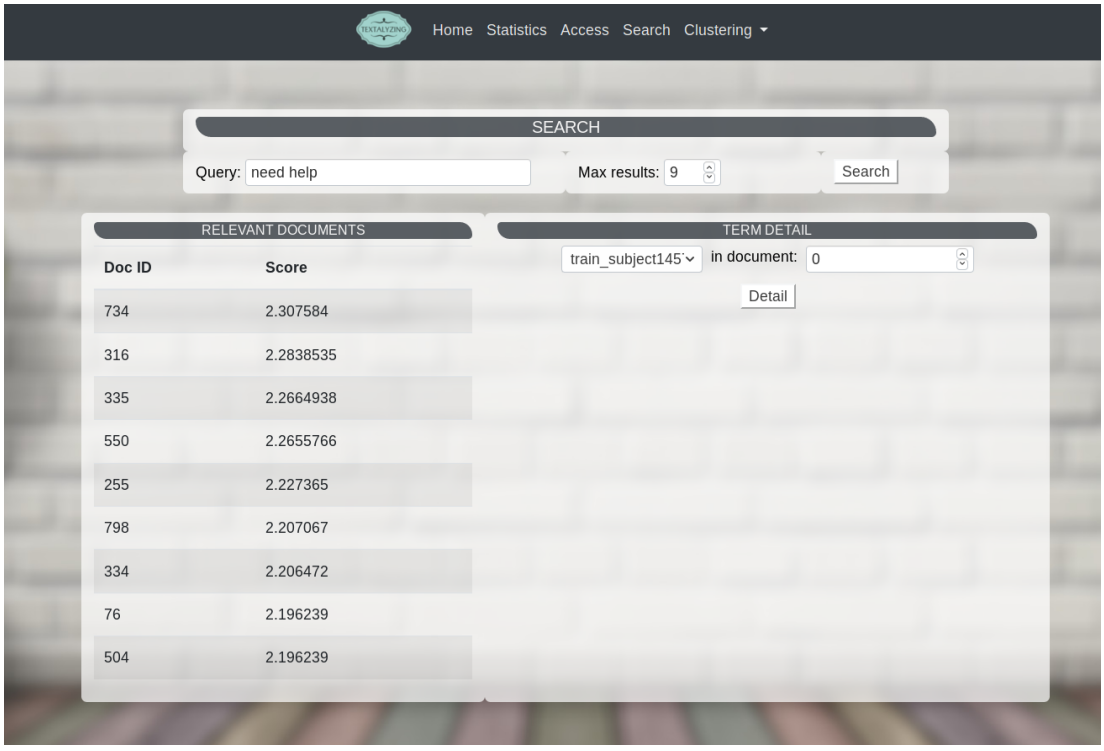

Figura A.9: Realización dunha consulta.

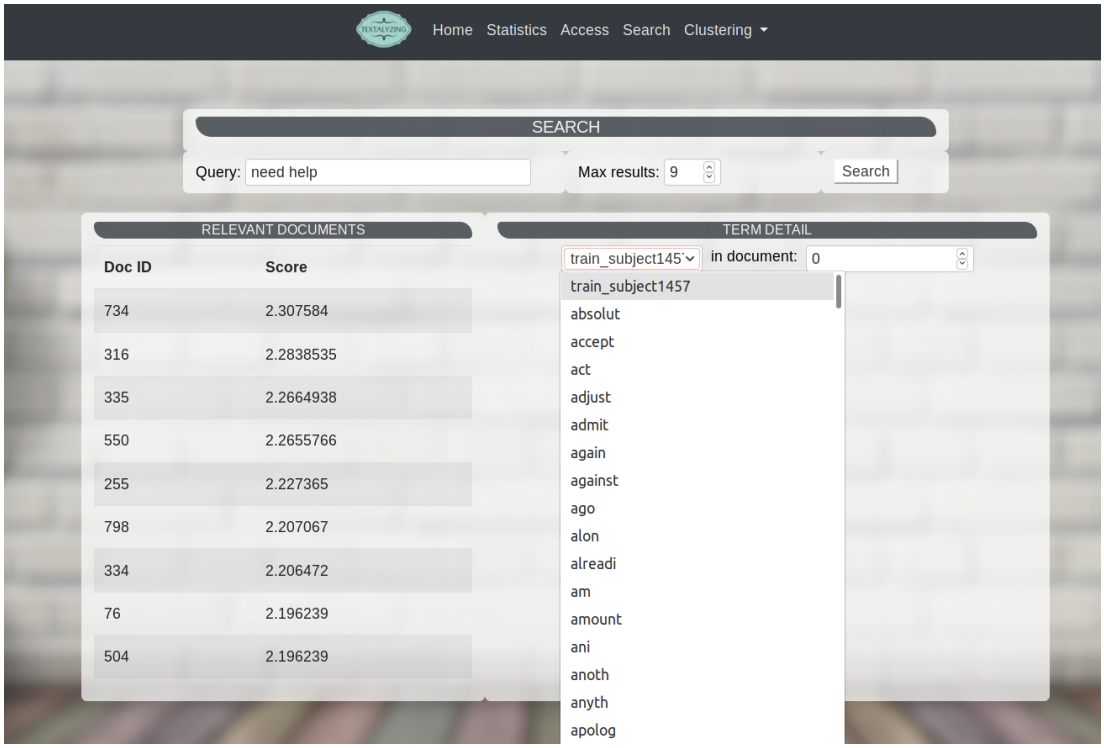

Figura A.10: Termos dispoñibles para ver o detalle.

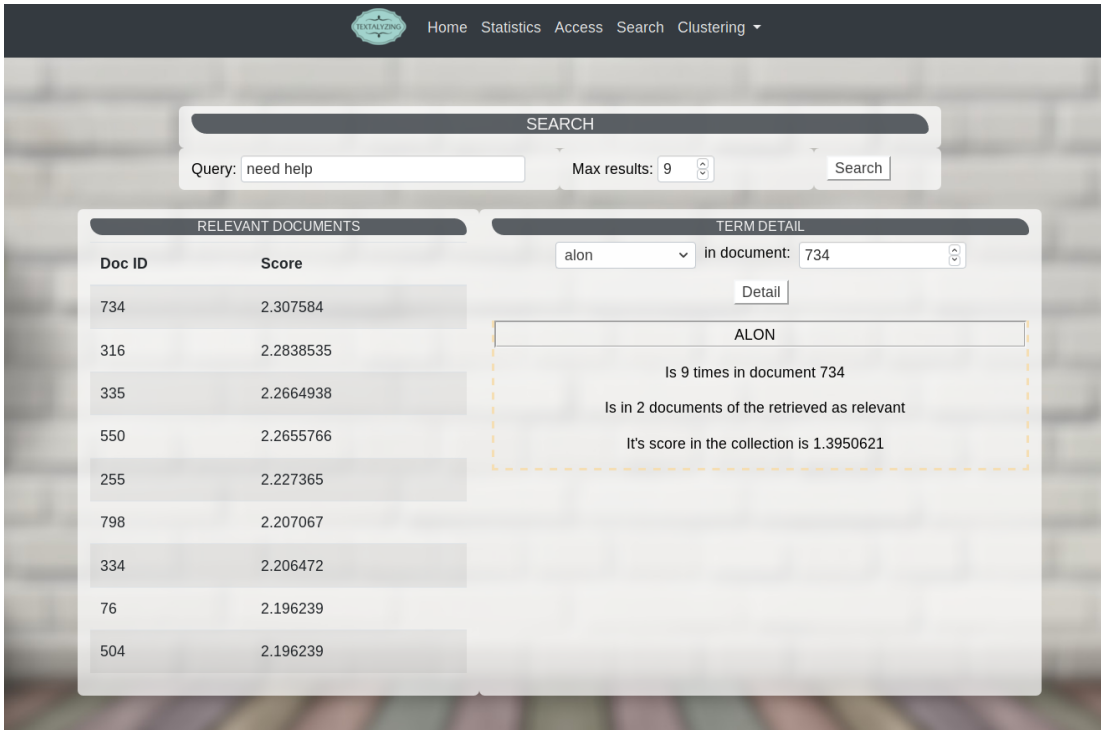

Figura A.11: Detalle dun termo.

A.5 Agrupamento

Dentro da sección ”Clustering” da barra superior de navegación encóntranse as opcións de agrupar por termos ou documentos.

A.5.1 Agrupamento de termos

Pódese ver na Figura A.12 as distintas opcións para o agrupamento de termos cuxas combinacións xa se mostraron na sección de resultados (Sección 6.2).

Especifícase a representación que van ter os termos máis similares ao introducido no campo *Term*, o número de termos que se queren agrupar e en cantos grupos. Da mesma maneira, pódese escoller se se quere ou non aplicar reducción de dimensionalidade para realizar o agrupamento. Pódese ver un exemplo de agrupamento de termos na Figura A.13.

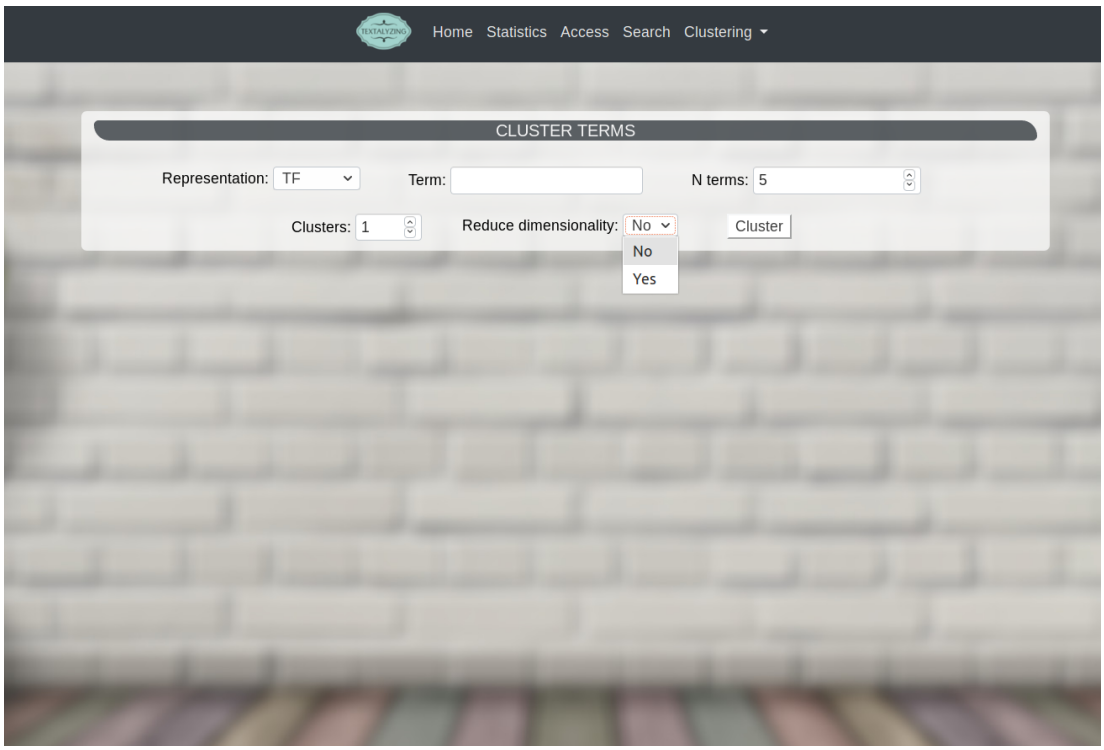

Figura A.12: Opcións para a agrupación de termos.

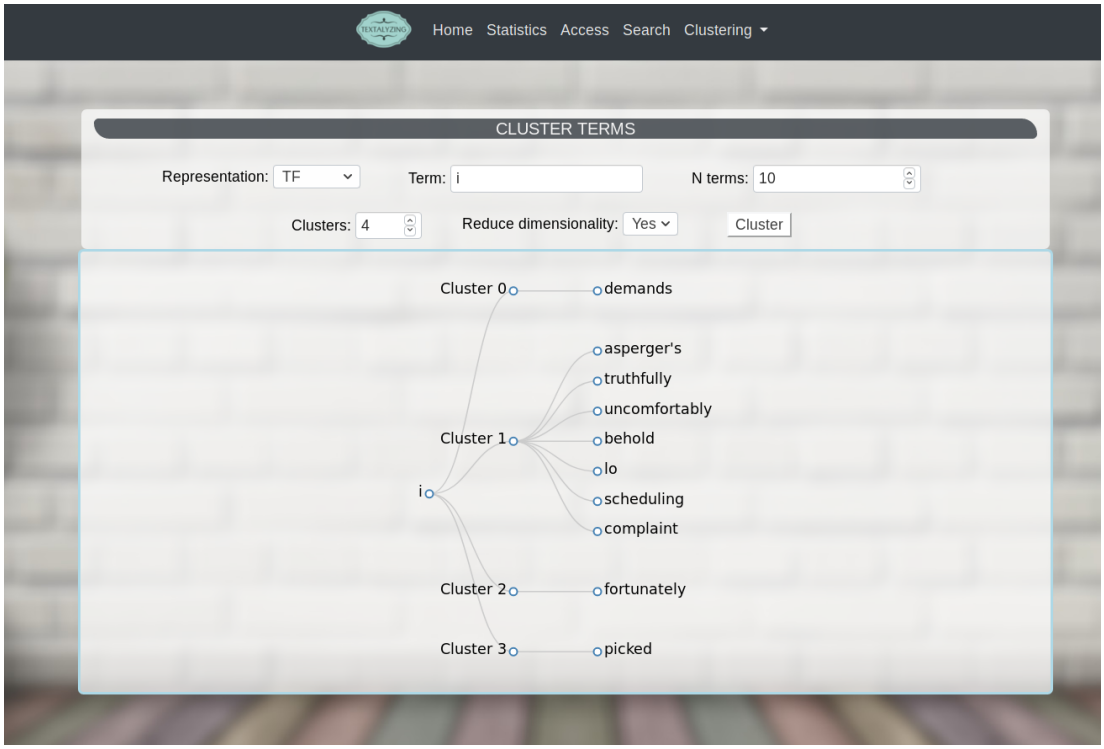

Figura A.13: Exemplo de agrupación dos 10 termos máis similares a *I*.

A.5.2 Agrupamento de documentos

Pódese ver na Figura A.14 as distintas opcións para o agrupamento de documentos. Especifícase a representación que van ter os documentos máis similares ao introducido no campo *Document*, o número de documentos que se queren agrupar e en cantos grupos. Pódese ver un exemplo de agrupamento de termos na Figura A.15.

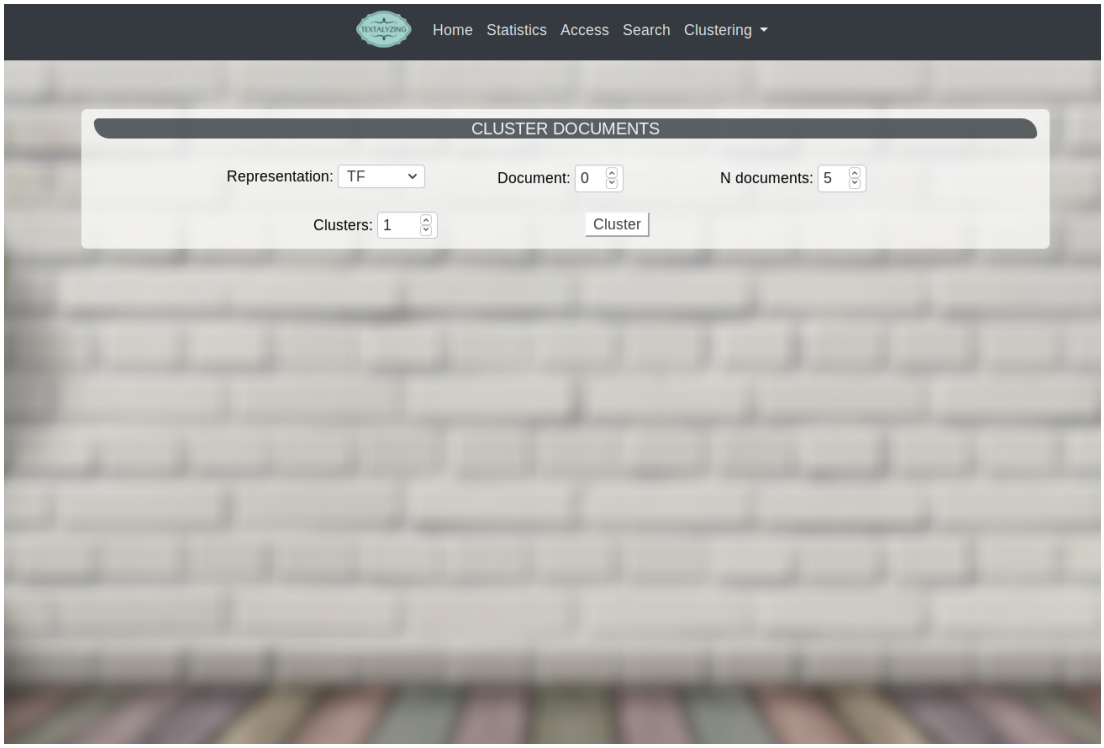

Figura A.14: Opcións para a agrupación de documentos.

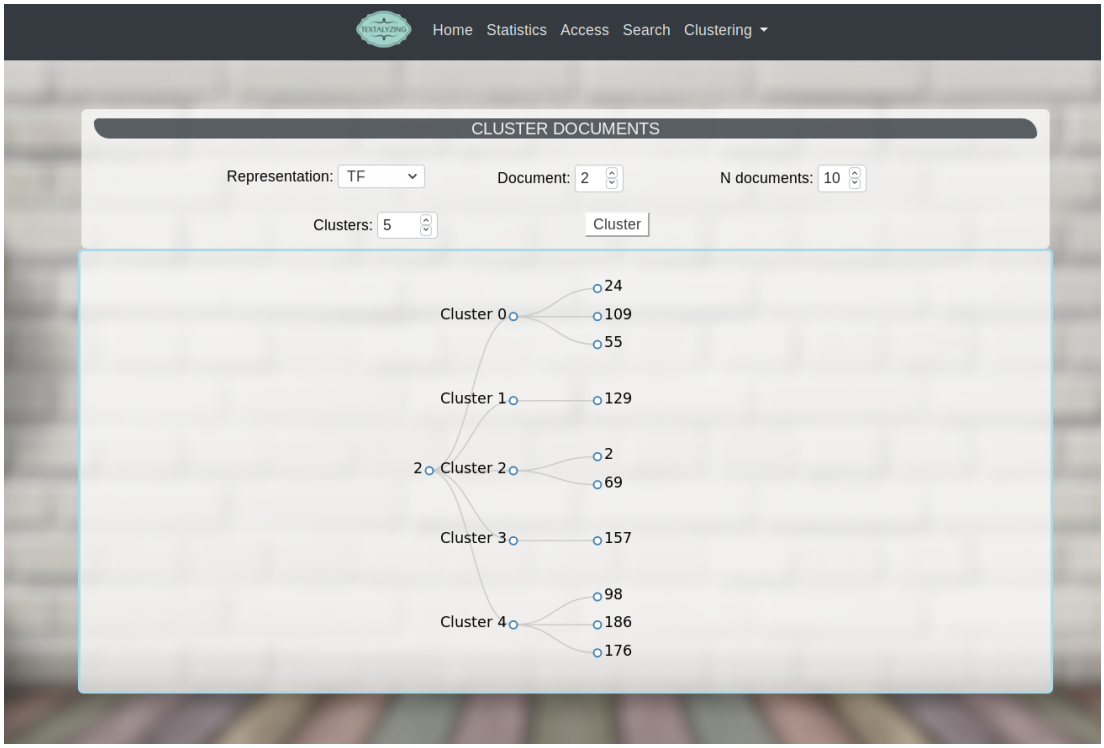

Figura A.15: Exemplo de agrupación dos 10 documentos máis similares ao documento con identificador 2.

Relación de Acrónimos

OMS *Organización Mundial da Saúde.*

CLEF *Conference and Labs of the Evaluation Forum.*

URL *Uniform Resource Locator.*

JVM *Java Virutal Machine.*

ERLANG/OTP *Erlang Open Telecom Platform.*

PMBoK *A Guide to the Project Management Body of Knowledge.*

IEEE *Institute of Electrical and Electronics Engineers.*

MVC *Modelo-Vista-Controlador.*

Glosario

Clústering Proceso no cal se divide en un conxunto de datos en grupos con características semellantes.

IEEE Asociación mundial de enxeñeiros dedicada á normalización e ao desenvolvemento en áreas técnicas.

Plos One Revista científica que abarca temas sobre a investigación en calquera materia relacionada coa ciencia e medicina.

OMS Organismo da Organización das Nacións Unidas encargado de xestionar políticas de prevención, pormoción e intervención da saúde a nivel mundial.

JVM Máquina virtual de Java que se sitúa no nivel superior do hardware sobre o que se executa a aplicación e actúa como ponte que entende tanto o *bytecode* como o sistema sobre o que se pretende executar.

Bytecode Código que xeran compiladores de determinadas linguaxes que é executado polo un intérprete.

Implementar Posta en marcha dunha idea programada.

Hipervínculo Enlace.

Array Conxunto de elementos ordeados en fila.

Query Palabra ou conxunto de palabras para realizar unha consulta.

Bibliografía

[1] WHO, "Adolescent mental health," 2019. [En liña]. Dispoñible en: https://www.who.int/news-room/fact-sheets/detail/adolescent-mental-health

[2] E. Ortiz-Ospina, "The rise of social media," 2019. [En liña]. Dispoñible en: https://ourworldindata.org/rise-of-social-media

[3] O. Parés-Badell, G. Barbaglia, P. Jerinic, A. Gustavsson, L. Salvador-Carulla, and J. Alonso, "Cost of disorders of the brain in spain," 2014. [En liña]. Dispoñible en: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0105471

[4] WHO, "Mental disorders," 2019. [En liña]. Dispoñible en: https://www.who.int/news-room/fact-sheets/detail/mental-disorders

[5] T. Catarci, P. Forner, D. Hiemstra, A. Peñas, and G. Santucci, Eds., *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics - Third International Conference of the CLEF Initiative, CLEF 2012, Rome, Italy, September 17-20, 2012. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7488. Springer, 2012. [En liña]. Dispoñible en: https://doi.org/10.1007/978-3-642-33247-0

[6] J. Parapar, D. E. Losada, and A. Barreiro, "Combining psycho-linguistic, content-based and chat-based features to detect predation in chatrooms," *J. UCS*, vol. 20, no. 2, pp. 213–239, 2014. [En liña]. Dispoñible en: https://doi.org/10.3217/jucs-020-02-0213

[7] D. E. Losada, F. Crestani, and J. Parapar, "erisk 2017: CLEF lab on early risk prediction on the internet: Experimental foundations," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings*, 2017, pp. 346–360. [En liña]. Dispoñible en: https://doi.org/10.1007/978-3-319-65813-1_30

[8] ——, "Overview of erisk: Early risk prediction on the internet," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 9th International*

Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings, 2018, pp. 343–361. [En liña]. Dispoñible en: https://doi.org/10.1007/978-3-319-98932-7_30

[9] ——, "Overview of erisk 2019 early risk prediction on the internet," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9-12, 2019, Proceedings*, 2019, pp. 340–357. [En liña]. Dispoñible en: https://doi.org/10.1007/978-3-030-28577-7_27

[10] H. Turtle and J. Flood, "Query evaluation: Strategies and optimizations," *Information Processing Management*, vol. 31, no. 6, pp. 831 – 850, 1995. [En liña]. Dispoñible en: http://www.sciencedirect.com/science/article/pii/030645739500020H

[11] W. Croft and J. Lafferty, *Language Modeling for Information Retrieval*, 01 2003.

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008.

[13] Infojobs, "Guía para trabajar en el sector it – informática y telecomunicaciones," 2019. [En liña]. Dispoñible en: https://orientacion-laboral.infojobs.net/guia-trabajar-sector-it

[14] D. Grefen, J. Miller, J. K. Armstrong, F. H. Cornelius, N. Robertson, A. Smith-Maclallen, and J. A. Taylor, "Identifying patterns in medical records through latent semantic analysis," 1993. [En liña]. Dispoñible en: https://cacm.acm.org/magazines/2018/6/228036-identifying-patterns-in-medical-records-through-latent-semantic-analysis/abstract