

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **Indexação de Documentos Clínicos**

**Estado da arte**

**João de Sá Balão Calisto Correia**



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Gabriel David (Prof.)

Fevereiro de 2016



# Resumo

Hoje em dia o acesso a informação estruturada e bem organizada é cada vez mais essencial no planeamento, desenvolvimento e desempenho de uma empresa. No meio clínico o acesso a este tipo de informação influencia os serviços prestados pelas entidades médicas, uma vez que a informação encontra-se demasiado dispersa por diversas fontes. Torna-se assim claro que inovar o acesso à informação nestes serviços é essencial para melhorar a eficiência e qualidade dos mesmos.

Na perspetiva de melhorar estes serviços, pretende-se criar uma solução inovadora na área da saúde em Portugal que possibilite extrair informação de diferentes fontes e com diferentes formatos, indexá-la e disponibiliza-la. As fontes podem ser *webservices*, sistemas de ficheiros, bases de dados relacionais e *document-oriented* enquanto os formatos podem ser PDF, Word, XML, JSON, etc. Relativamente ao conteúdo da informação temos por exemplo: resultados analíticos laboratoriais, relatórios clínicos, diagnósticos codificados em ICD-9, notas clínicas dos médicos, requisições de exames, prescrições de medicamentos e informação demográfica de pacientes. Para fins de indexação pretendem-se utilizar tecnologias na área, como *Apache Lucine*, *Elasticsearch*, *Apache Solr*, entre outros.

Este projeto destina-se a profissionais de saúde e tem como objetivo disponibilizar-lhes uma forma centralizada e pesquisável de obter todos os dados clínicos de um doente/ paciente. Pretende-se que a solução seja apresentada na forma de uma API para que possa ser reutilizada em projetos futuros.

A solução que será desenvolvida terá um forte impacto no dia-a-dia dos profissionais de saúde assim como nos seus pacientes, pois o acesso à informação tornar-se-á mais fácil e rápido, melhorando, com isto, a prestação de serviços de saúde em Portugal.

Este documento destina-se à apresentação da análise do estado da arte para o tema em causa.



# Abstract

Nowadays access to structured and well organized information is increasingly essential in the planning, development and performance of a company. In the middle clinician access to this type of information influences the services provided by medical entities, since the information is too dispersed by several sources. Thus it becomes clear that innovate access to information on these services is essential to improve efficiency and quality.

In the perspective of improving these services, we intend to create an innovative solution in the area of health in Portugal that allows extracting information from different sources and in different formats, index it and make it available. Fonts can be webservices, file systems, relational databases and document-oriented while the formats can be PDF, Word, XML, JSON, etc. Regarding the content of the information under eg laboratory analytical results, clinical reports, diagnoses coded in ICD9, clinical notes from doctors, requisitions, drug prescriptions and demographic information of patients. For indexing purposes intend to use technology in the area, such as *Apache Lucine*, *Elasticsearch*, *Apache Solr*, among others.

This project is aimed at health professionals and aims to provide them with a centralized, searchable way to get all the clinical data of a patient. The solution is intended to be presented in the form of an API that can be reused in future projects.

The solution that will be developed will have a strong impact of health professionals on a day-to-day as well as their patients, since access to information will become easier and faster, improving, thus, the provision of services health in Portugal.

This document is intended for the presentation of the study's analysis of the state of the art for the theme concerned.



# Conteúdo

<b>Introdução .....</b>	<b>1</b>
1.1 Contexto/Enquadramento .....	1
1.2 Motivação e Objetivos .....	2
1.3 Estrutura do Relatório.....	2
<b>Revisão Bibliográfica .....</b>	<b>3</b>
2.1 Introdução.....	3
2.2 Recuperação de Informação .....	4
2.2.1 Modelos Clássicos .....	5
2.2.2 Modelos Estruturados .....	6
2.3 Indexação .....	7
2.3.1 Apache Lucene .....	9
2.3.2 Apache Solr .....	11
2.3.3 ElasticSearch.....	11
2.3.4 Sphinx Search .....	12
2.3.5 Análise Tecnológica.....	13
2.4 Classificação.....	15
2.4.1 Relações Hierárquicas.....	16
2.4.2 UMLS .....	16
2.5 Resumo.....	19
<b>Conclusões e Trabalho Futuro .....</b>	<b>20</b>
3.1 Satisfação dos Objetivos.....	20
3.2 Trabalho Futuro .....	21
<b>Referências.....</b>	<b>22</b>





# Lista de Figuras

Figura 1 - Sistema RI	4
Figura 2 - Arquitetura do Apache Lucene, Hatcher and Gospodnetic [4]	10
Figura 3 - Arquitetura do <i>Sphinx Search</i> [2]	12
Figura 4 - Ranking de popularidade de ferramentas de pesquisa e bases de dados [3]	15
Figura 5 - Estrutura hierárquica e relações dos termos do SNOMED [5]	17
Figura 6 - Estrutura hierárquica do MeSH [1]	18
Figura 7 - Planeamento	21



# Lista de Tabelas

Tabela 1 - Comparação entre SOLR, ElasticSearch e Sphinx Search	14
Tabela 2 - Exemplo de ICD-9-CM [21]	18



# Abreviaturas e Símbolos

API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
ICD-9-CM	International Classification of Diseases, 9th Revision, Clinical Modification
JSON	JavaScript Object Notation
MeSH	Medical Subject Headings
PDF	Portable Document Format
REST	Representational State Transfer
RI	Recuperação de Informação
SNOMED CT	Systematized Nomenclature of Medicine – Clinical Terms
UMLS	Unified Medical Language System
XML	eXtensible Markup Language

# Capítulo 1

## Introdução

O presente relatório pretende descrever os resultados obtidos na análise do estado da arte para a criação de uma solução cujo objetivo é recolher e extrair informação de diferentes fontes e com diferentes formatos, indexá-la, relacioná-la e disponibilizar uma interface de pesquisa sobre a mesma.

Neste capítulo introdutório apresenta-se o contexto da dissertação, a motivação e os objetivos. É também apresentada uma visão global sobre a estrutura da mesma.

### 1.1 Contexto/Enquadramento

Com o passar dos anos, constata-se que a sociedade em que vivemos é baseada cada vez mais na informação. A quantidade de informação aumenta a um ritmo **voraz**, o que torna a sua gestão bastante difícil. Assim, para que a sociedade se desenvolva, torna-se progressivamente necessária a existência de soluções e mecanismos inovadores que possibilitem o acesso à informação de forma rápida e eficaz. Para fazer face a estes factos, surgem assim métodos de recuperação e de indexação que tornam possível o acesso à informação com as características referidas anteriormente.

Com isto, é possível assumir que esta dissertação se encontra nas áreas das ciências da informação e nas tecnologias de bases de dados, na medida em que aborda a importância do acesso e recuperação da informação em meios clínicos.

Esta dissertação será desenvolvida no âmbito empresarial da *Glintt - Healthcare Solutions*, a qual é focada no ramo da saúde e onde, a nível nacional, é líder destacada neste segmento do mercado.



## 1.2 Motivação e Objetivos

No meio clínico a quantidade de informação é elevada, o que torna difícil o acesso à mesma pelas entidades médicas. Além disto, outros fatores alarmantes são que a mesma quantidade de informação encontra-se dispersa por várias fontes e diferentes formatos, e por vezes, não se encontra estruturada o que faz com que as entidades médicas percam demasiado tempo na pesquisa/ procura dos dados clínicos de um doente.



Devido aos problemas referidos, há um atraso no acesso à informação dos médicos, há falta de informação e, por vezes, esta nem chega corretamente **as** mãos das entidades clínicas. Em outras palavras, os médicos têm um difícil acesso à informação.

Assim, surge o **propósito desta dissertação, proporcionar melhores condições de acesso e pesquisa de informação clínica, aglomerando a informação num só local, categorizando-a e relacionando-a, com** o objetivo de criar melhores condições de acesso à informação nos serviços de saúde em Portugal.

## 1.3 Estrutura do Relatório

O presente documento está dividido em 3 capítulos:

- **Capítulo 1 – Introdução:** é um capítulo onde é apresentado o contexto ou enquadramento do tema em causa;
- **Capítulo 2 – Revisão Bibliográfica:** é onde se descreve o estudo do estado da arte e está dividido em três subsecções: Recuperação de Informação, Indexação e Classificação;
- **Capítulo 3 – Conclusões e Trabalhos futuros:** é onde se apresenta conclusões relativas a este documento e se revela as etapas futuras do projeto em causa.

## Capítulo 2

# Revisão Bibliográfica

As secções seguintes descrevem o estado da arte relativa aos processos de RI (recuperação de informação) e indexação com uma breve descrição das tecnologias mais recentes e mais utilizadas na área. Na última secção é apresentada como é feita a classificação da informação.

### 2.1 Introdução

Desde há muito tempo atrás que a comunidade tem vindo a armazenar grandes quantidades de informação para que esta possa ser consultada por diversas pessoas de geração em geração. Face a esta acumulação de informação recolher/extrair informação com significado/utilidade começou a tornar-se bastante complicado. Entre 1948 e 1950, Calvin Mooers [6] cria o termo “Recuperação de Informação” **potenciado** pela necessidade de **gerenciar** a explosão da informação na literatura científica na segunda metade do século XX.

Nos anos seguintes foram feitas várias experiências na área e durante os anos 70 os sistemas de recuperação de informação começaram a tornar-se reais essencialmente devido ao aparecimento de processadores de texto [7].

Em 1980 com a diminuição constante do preço do espaço em discos a informação disponível em máquinas crescia rapidamente. Nesta década começaram a estar disponíveis textos completos e não apenas resumos e índices.

Com o passar dos anos **e, tendo em conta** o aparecimento de diversos dispositivos para armazenamento de informação digital, a informação continuou a crescer forçando os sistemas de recuperação de informação a desenvolverem-se da mesma forma com o objetivo de acompanhar o aumento de informação.

Hoje em dia, o processo de recuperação de informação está bastante **diluído** na sociedade, sendo que o desenvolvimento de ferramentas de pesquisa veio ajudar neste processo. O conceito




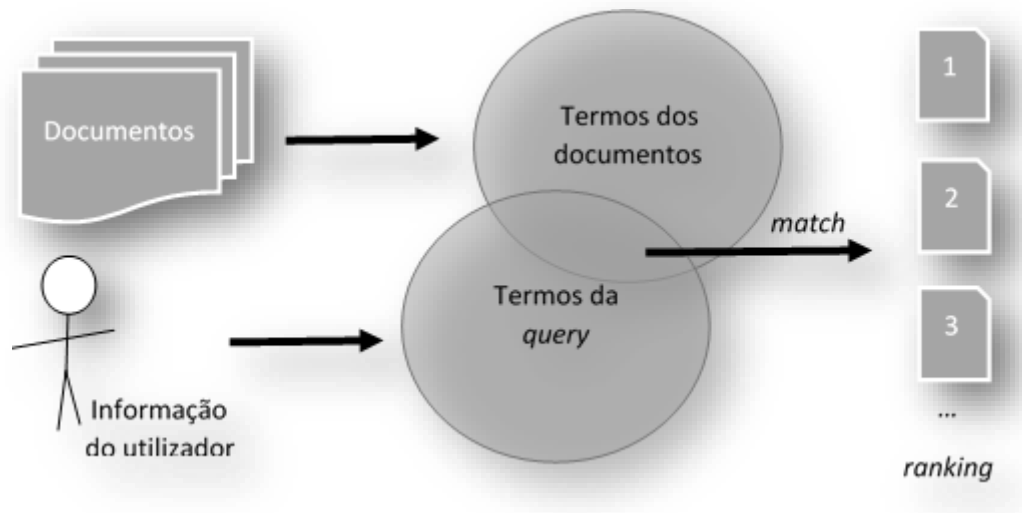
de recuperação de informação surgiu então da forte necessidade de extrair informação com valor das diversas fontes, organizando e facilitando o acesso aos conteúdos informacionais através dos processos de indexação e de descrição.

## 2.2 Recuperação de Informação

A RI pode ser definida como a atividade de obtenção de informação, devido a uma necessidade, a partir de uma coleção de recursos de informação. Segundo alguns autores a RI é descrita da seguinte forma:

*“Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).”*, Manning, et al. [8]

De uma forma geral, os sistemas de RI devem tentar corresponder da melhor forma às necessidades do utilizador recorrendo a *ranks* ordenados de acordo com o grau de proximidade à pergunta feita pelo utilizador. 



**Figura 1 - Sistema RI**

Na imagem anterior é ilustrado o funcionamento de um sistema RI. Todavia, a estratégia apresentada na figura 1 não é tão trivial como possa parecer, uma vez que um sistema RI enfrenta vários problemas, entre os quais se destaca o **fato** de não se saber quais os documentos relevantes para determinada pergunta do utilizador. Esta tarefa é normalmente realizada com base em

alguma heurística que decide quais os documentos com maior relevância a serem recuperados e ordena-os a partir de critérios estabelecidos [9].

Para a resolução deste problema surgem modelos para recuperação de informação. Estes estão divididos em modelos clássicos e modelos estruturados. Os modelos clássicos [6] são baseados em termos de indexação (palavras-chave), ou seja, cada documento consiste num conjunto desses termos. Já os modelos estruturados podem especificar alguma informação sobre a estrutura do texto, por exemplo, proximidade de palavras. A descrição de cada um dos modelos referidos encontra-se nos tópicos seguintes.

Em relação aos sistemas RI apenas resta dizer que têm um papel muito importante ou até fundamental na **concretização** das necessidades do utilizador e por isso, adaptar um sistema deste tipo num meio clínico **supõe-se que irá** ajudar, neste caso, as entidades médicas, de modo a que estas melhorem o seu desempenho no dia-a-dia.

## 2.2.1 Modelos Clássicos

### Modelo Booleano

É um modelo clássico baseado na lógica booleana, isto é, os termos **são** combinados com os operadores AND, OR, e NOT de forma a **fornecer ao utilizador a informação que este** precisa, por exemplo: “president AND lincoln”. Oferece algumas vantagens como o facto de os resultados serem previsíveis e fáceis de explicar e, como é um conceito intuitivo, os utilizadores sentem-se no controlo do sistema. No entanto são também visíveis algumas desvantagens dentro das quais se destaca o **fato** de não providenciar **um ranking** dos documentos encontrados em resposta à pergunta do utilizador [10].

### Modelo Vetorial

No modelo vetorial os documentos e as *queries* são representados por vetores de termos que têm um peso associado. Este peso não é binário de forma a não limitar os resultados da pesquisa (como é feito no modelo booleano). Cada peso, associado ao respetivo termo, é usado para calcular o grau de similaridade entre cada documento de uma coleção e a *query*. De uma forma mais detalhada, o peso referido especifica o tamanho e a direção do vetor e a similaridade é calculada com base no **ângulo** entre os dois vetores, da *query* e do documento [11].

O modelo vetorial foi criado com o objetivo de eliminar a maioria dos problemas associados ao léxico, em particular, uma vez que existem palavras com diferentes significados (dependentes do contexto), é difícil diferenciar dois documentos que partilham a mesma palavra. Além disto, existem diversas formas de representar um conceito com base em sinónimos. No entanto, técnicas



baseadas no léxico não apresentariam resultados com sinónimos da palavra pesquisada ou, no pior caso, nem apresentaria resultados.

Este modelo, ao efetuar o cálculo da similaridade entre as *queries* e os documentos, permite a existência de resultados ordenados de acordo com a medida de similaridade usada, ou seja, utiliza *rankings* fornecendo, desta forma, melhores resultados em termos contextuais para o utilizador.

### **Modelo Probabilístico**

O modelo probabilístico foi proposto por S.E. Robertson e K. Sparck Jones [12], e, de uma forma geral, consiste em estimar a probabilidade de um documento  $d_i$  ser relevante para uma *query*  $q$ . Este modelo assume que a probabilidade de relevância depende apenas da *query* e do documento. O resultado ideal de uma pesquisa seria um conjunto que contivesse todos e apenas os documentos relevantes para o utilizador. Este modelo ordena os documentos na ordem decrescente da probabilidade de relevância da informação.

Existem algumas limitações, entre as quais se verifica a ausência de atribuição de pesos a cada termo. Além disto, é necessário fazer uma estimativa inicial de forma a refinar o conjunto da resposta ideal sendo que, para cada termo, não é considerada a quantidade de vezes que este aparece num documento.

## **2.2.2 Modelos Estruturados**

### **Modelo da Proximidade de Nós**

O modelo da proximidade de nós visa a recuperação de documentos através de uma estrutura hierárquica de índices e, por isso, permite que várias estruturas possam ser definidas no mesmo texto, cada uma sendo uma hierarquia restrita mas permitindo a sobreposição entre áreas delimitadas por hierarquias diferentes. Uma *query* pode relacionar diferentes hierarquias mas retorna um subconjunto de nós de apenas uma delas. Cada nó tem associado um segmento que corresponde à área de texto a que este está associado. O segmento de um nó inclui os dos seus descendentes.

De uma forma geral, o modelo em causa possibilita a definição de estruturas de indexação hierárquicas e independentes sobre um mesmo documento [13].

## Modelo de Listas Não Sobrepostas

O modelo de listas não sobrepostas foi proposto por Burkowski [9] em 1992 e consiste na divisão de todo o texto de um documento numa região não sobreposta e juntá-las numa lista. Existem diversas formas de dividir o texto em regiões não sobrepostas que levam à geração de listas múltiplas:

- Lista para capítulos;
- Lista para secções;
- Lista para subsecções.

Regiões de texto de listas distintas podem sobrepor-se.

Relativamente à implementação, o modelo utiliza um ficheiro invertido que combina as palavras-chave e as regiões de texto. Para cada entrada neste ficheiro invertido é associada uma lista de regiões de texto. Listas de regiões de texto podem ser juntadas com listas de palavras-chave.

## 2.3 Indexação

Como referido no capítulo anterior, os sistemas RI organizam e viabilizam o acesso à informação através da descrição de documentos e da operação do tratamento documental, a indexação. Mas afinal, em **quê que** consiste a indexação?

A indexação tenta descrever e caracterizar um documento com o auxílio de representações dos conceitos contidos nesses documentos, ou seja, em transcrever para linguagens de indexação os conceitos depois de terem sido extraídos dos documentos por meio de uma análise dos mesmos. Este processo de indexar segue assim três etapas:

1. Compreensão do documento;
2. Seleção dos conceitos para a pesquisa;
3. Tradução destes conceitos nas linguagens de indexação.

Na compreensão do documento deve ser feita uma análise global do mesmo, isto é, ter em consideração as partes mais úteis como fonte de informação do tema/assunto. As partes referidas podem ser: título, resumo, índice, prefácio, introdução, entre outros. Nesta etapa não é, por isso, necessário ler o documento na íntegra.

Relativamente à segunda etapa, a seleção dos conceitos para a pesquisa, deve ser feita de forma exaustiva e específica, ou seja, todos os assuntos de que trata o documento devem ser registados e nunca um conceito deve ser traduzido.

No terceiro passo, é feita a tradução destes conceitos nas linguagens de indexação, isto é, transformar os conceitos selecionados em termos ou símbolos autorizados para representá-los no sistema. Após a execução destas etapas dá-se por terminado o processo de indexação.

Neste processo de indexar é ainda importante realçar dois conceitos: *stemming* e *stopwords*.

### ***Stemming***

No processo de indexar e, na maioria das vezes, existem palavras que, apesar de pertencerem à mesma família, são indexadas separadamente. Como é possível observar, nestes casos estão a ser criados vários índices que possivelmente representarão o mesmo conteúdo. Face a este e outros casos semelhantes surge o *stemming* [14] que consiste em reduzir as palavras à sua raiz morfológica. A raiz morfológica de uma palavra é o conjunto de caracteres que está presente em todas as suas derivações.

Pode parecer um processo simples, mas, se o fosse, apenas existiria uma única implementação dele, o que não é o caso. O *stemming* por vezes falha em reduzir as palavras com o mesmo significado para a mesma raiz (*understemming*) e por outras vezes falha **para** manter palavras com significados diferentes separadas (*overstemming*). Apesar disto, um algoritmo de *stemming* bem implementado é capaz de melhorar significativamente a qualidade da solução.

### ***Stopwords***

Em todos os textos/documentos existem palavras que, por aparecerem **muitas vezes**, não acrescentam valor **contextual** ao conteúdo de um documento como o “e”, “de”, etc. Além disso, estas palavras são raramente pesquisadas pelo utilizador uma vez que não têm valor para o documento. As palavras referidas são conhecidas como *stopwords*, pois devido à sua falta de valor contextual são eliminadas do processo de indexação. Este processo de eliminação referido permite um aumento de qualidade nos resultados.

Nos tópicos seguintes são apresentadas algumas das ferramentas mais usadas nos dias de hoje para indexar e pesquisar conteúdos de uma forma mais rápida e fácil. Essas ferramentas são: *Apache Lucene*, *Apache Solr*, *ElasticSearch* e *SphinxSearch*.

### 2.3.1 Apache Lucene

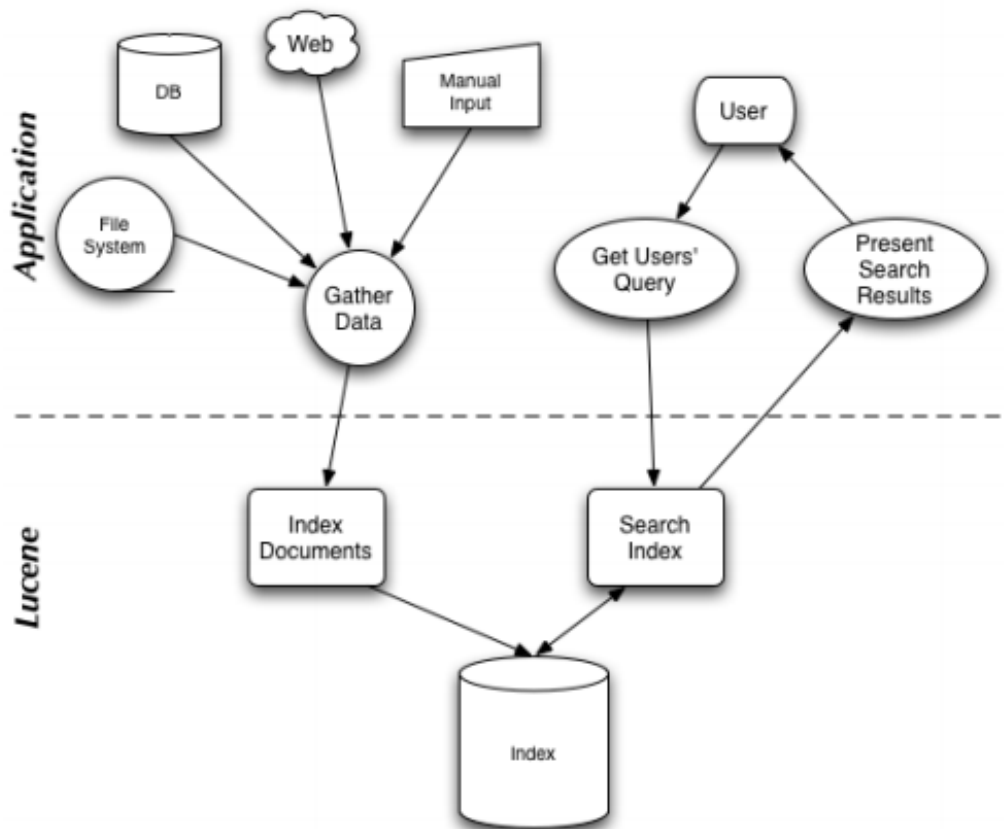
O *Apache Lucene* foi desenvolvido em 1997-8 por *Doug Cutting*. Inicialmente denominado apenas de *Lucene*, foi disponibilizado gratuitamente no *SourceForge* em 2000. Mais tarde, em 2001, juntou-se à *Apache Software Foundation's* onde passou a chamar-se *Apache Lucene*. Em 2005, já era um projeto de alto nível na *Apache Software Foundation's*. Segundo os autores Hatcher e McCandless o *Lucene* é descrito da seguinte forma:

*“Lucene is a high-performance, scalable information retrieval (IR) library. IR refers to the process of searching for documents, information within documents, or metadata about documents. Lucene lets you add searching capabilities to your applications. It’s a mature, free, open source project implemented in Java, and a project in the Apache Software Foundation, licensed under the liberal Apache Software License. As such, Lucene is currently, and has been for quite a few years, the most popular free IR library.”*, McCandless, et al. [15]

Descrito em outras palavras, *Lucene* é uma ferramenta que permite indexar e realizar pesquisa sobre os elementos indexados. Um índice é constituído por uma lista de documentos (ou outra fonte de informação), os quais são compostos por uma sequência de campos e em que cada um destes consiste numa sequência nomeada de termos, ou seja, por nome do campo e termo, por exemplo: (“título”, ”introduction to Lucene”). O termo pode conter uma ou várias palavras com valor no contexto do documento a indexar. O *Lucene* segue um sistema de indexação designado por *inverted index*, ou em português, índice invertido, pois em vez de, para cada documento, guardar os termos que nele existem, guarda, para cada termo, os documentos onde estes se encontram.

Para determinar a relevância de um documento perante a pergunta do utilizador, o *Apache Lucene* utiliza uma combinação entre o modelo booleano e o modelo vetorial.

A **figura 1** apresenta a arquitetura do *Apache Lucene* em 2004, apesar de continuar bastante atual.



**Figura 2 - Arquitetura do Apache Lucene, Hatcher and Gospodnetic [4]**

- **Gather Data:** nesta etapa são recolhidos os conteúdos dos diferentes documentos.
- **Index Documents:** é feita uma análise do documento e, após esta, inicia-se o processo de indexação.
- **Index:** base de dados de índices.
- **Get Users' Query:** uma vez feito o pedido de pesquisa do utilizador, a aplicação constrói a *query* com base no texto de pesquisa a fim de ser utilizada na interrogação às bases de dados de índices.
- **Search Index:** é a etapa onde se realiza a pesquisa por índices de acordo com a *query* construída a partir dos dados de pesquisa introduzidos pelo utilizador.
- **Present Search Results:** apresenta o resultado da pesquisa ao utilizador.

Com isto, termina a análise desta ferramenta afirmando que é uma ferramenta poderosa de indexação e pesquisa que permite rapidamente e eficazmente criar índices e realizar pesquisa sobre os mesmos.

### 2.3.2 Apache Solr

O *Apache Solr* surgiu em 2006 e consiste numa ferramenta construída a partir do *Apache Lucene*. Devido à sua origem a partir do *Apache Lucene*, implementa muitas das suas funcionalidades e acrescenta muitas outras. O *Apache Solr* apresenta assim as seguintes características [16]:

- Altamente escalável e tolerante a falhas;
- Otimizado para alto volume de tráfego;
- Fornece APIs baseadas em REST o que lhe permite ser integrado a praticamente qualquer linguagem de programação (XML, HTTP e JSON APIs);
- Fácil extensão através de *plugins*;
- Quase em tempo real, isto é, os documentos estão disponíveis para pesquisa quase imediatamente a seguir à sua indexação;
- Integra o *Apache Tika* [17], por isso, aceita para indexação documentos com diversas extensões como PDF, Word, XML, etc.

Além destas características pode ser ainda comentado o **fato** de este possuir uma boa documentação [18] que, juntamente com o excelente suporte que recebe, o torna uma ferramenta bastante utilizada hoje em dia.

### 2.3.3 Elasticsearch

Em semelhança ao *Apache Solr*, o *ElasticSearch* foi desenvolvido com base no *Apache Lucene* ao **que lhe foram introduzidas** algumas melhorias. Foi desenvolvido por Shay Banon e teve o primeiro lançamento em 2010.

O *ElasticSearch* permite facilmente aceder às funcionalidades do *Lucene* para indexação e pesquisa [19]. No entanto, proporciona um novo nível de abstração de análise em tempo real. Outro nível de abstração é a forma como podem ser organizados os documentos: múltiplos índices podem ser pesquisados em simultâneo ou separadamente, e é possível colocar diferentes tipos de documentos dentro de cada índice.



Por fim o *ElasticSearch* é como o próprio nome sugere, é elástico, ou seja, é possível acrescentar gradualmente servidores de forma a aumentar a capacidade e tolerância a falhas. De forma semelhante, é possível remover estes gradualmente de forma a reduzir os custos.

### 2.3.4 Sphinx Search

O *Sphinx Search* começou a ser desenvolvido em 2001 e consiste num servidor de pesquisa *full-text*, escrito em C++ que funciona com diversos sistemas operativos. Esta ferramenta permite indexar e pesquisar informação guardada em bases de dados SQL e NoSQL, ou apenas em ficheiros.

Na figura seguinte é apresentada a arquitetura da ferramenta.

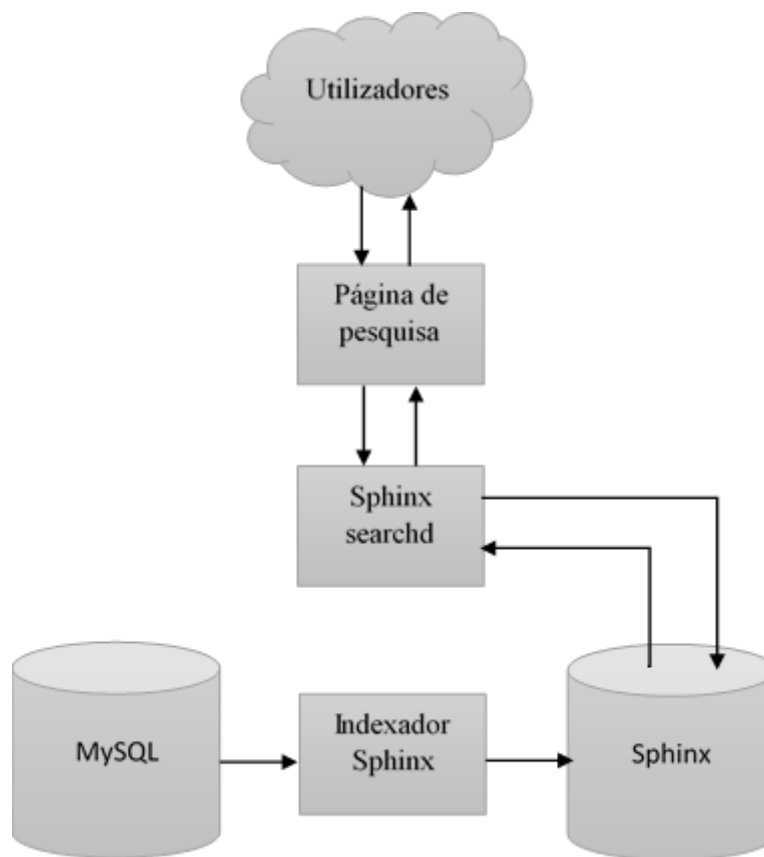


Figura 3 - Arquitetura do *Sphinx Search* [2]

Descrevendo com melhor pormenor o que se observa na imagem, o **searchd** é responsável por receber pedidos do cliente e executar pesquisa comparando com os índices no **Sphinx**. Já o

**Indexador Sphinx** é responsável por recolher a informação do **MySQL** (ou outra fonte suportada) e **indexa-la**.

Perante esta análise é ainda possível realçar algumas características [2] desta ferramenta:

- Indexação em tempo real;
- Indexação de base de dados SQL (MySQL, PostgreSQL, Oracle, SQLite, etc);
- Indexação de base de dados NoSQL;
- Vem com o *SphinxAPI* que é uma biblioteca nativa disponível para Java, PHP, Python, Perl, C entre outras linguagens;
- Suporta pesquisas sintaticamente complexas;
- Analisa proximidade entre palavras e ordena em rankings.

### 2.3.5 Análise Tecnológica


Nos tópicos anteriores foram descritas algumas das ferramentas usadas hoje em dia para efeitos de indexação e recuperação de informação. Não esquecendo que estamos perante uma indexação de documentos clínicos, é importante ter em conta quais das tecnologias melhor respondem às necessidades em causa. Para tal, e relembrando o que está em causa, é necessário recolher e extrair informação de diferentes fontes e com diferentes formatos. As fontes possíveis são: *webservices*, sistemas de ficheiros, bases de dados relacionais e *document-oriented*. Relativamente aos formatos podem variar desde ficheiros de texto simples a PDF, Word, JSON, XML entre outros. Face a estes dados é possível partir para uma análise comparativa entre as ferramentas em estudo.

Relativamente ao *Apache Lucene*, é possível afirmar que não trará grandes vantagens na sua utilização direta no âmbito em questão, pois os seus “sucessores”, *Solr* e *ElasticSearch* trazem uma abrangência muito maior relativamente, não só, às linguagens que disponibilizam, mas também aos diferentes formatos nos quais pode ser realizada a indexação e extraída a informação. Perante esta situação, na tabela seguinte é feita uma análise comparativa das ferramentas *Apache Solr*, *ElasticSearch* e *Sphinx Search*.

	<b>Solr</b>	<b>ElasticSearch</b>	<b>Sphinx Search</b>
API			
Bibliotecas	Ruby, Rails, PHP, Java, Python, Perl, C#, .Net, JavaScript	Java, Groovy, PHP, Ruby, Perl, Python, .NET, JavaScript	C++, Java, Perl, PHP, Python, Ruby
Indexação			
Importação de dados	JDBC, CSV, XML, Tika [17], URL, Flat File	Amazon SQS, CouchDB, Dropbox, DynamoDB, FileSystem, Git, GitHub, Hazelcast, JDBC, JMS, Kafka, LDAP, MongoDB, neo4j, OAI, RabbitMQ, Redis, RSS, Sofa, Solr, St9, Subversion, Twitter, Wikipedia	MySQL, PostgreSQL, MSSQL, ODBC source, XML pipe
Tempo Real	✓	✓	✓
Licença			
	Apache License 2.0	Apache License 2.0	GPLv2 e comercial

**Tabela 1 - Comparação entre SOLR, ElasticSearch e Sphinx Search**

Como é possível observar na tabela anterior, e começando por analisar a licença, tanto o *Solr* como o *Elasticsearch* têm a licença da Apache. Já o *Sphinx Search* tem a licença GPLv2 que, para fins comerciais, é necessário **compra**-la. Face **esta** informação temos aqui dois pontos de vista, ferramentas *opensource* e com licença comercial. Todavia ter licença comercial não é sinónimo de maior utilidade, isto é, no caso da importação de dados é possível observar que o *Sphinx Search* é muito mais orientado para as bases de dados e, por isso, carece de leitura de ficheiros PDF, Word entre outros que foram referidos anteriormente como uma necessidade para a solução em causa. Por outro lado, o *Solr* e o *Elasticsearch* implementam o *Apache Tika* [17], que consiste numa biblioteca que suporta a extração de texto de ficheiros binários, como por exemplo: PDF, Word, XML, formatos de áudio, etc. Mesmo relativamente às bibliotecas, o *Sphinx Search* apresenta menos variedade que as outras duas ferramentas. A figura seguinte apresenta um *ranking* de popularidade de diversas ferramentas de pesquisa e de bases de dados entre as quais se encontram as ferramentas estudadas neste projeto.

Rank			DBMS	Database Model	Score		
Feb 2016	Jan 2016	Feb 2015			Feb 2016	Jan 2016	Feb 2015
12.	12.	↑ 16.	Elasticsearch 	Search engine	77.84	+0.63	+25.01
13.	↑ 14.	13.	Teradata	Relational DBMS	73.38	-1.57	+3.93
14.	↓ 13.	↓ 12.	Solr	Search engine	72.27	-3.12	-9.21
15.	15.	↑ 17.	Hive	Relational DBMS	52.77	-0.81	+16.21
16.	16.	↓ 14.	HBase	Wide column store	52.02	-1.34	-5.12

**Figura 4 - Ranking de popularidade de ferramentas de pesquisa e bases de dados [3]**

Observando a figura anterior chegamos à conclusão que o *Solr* e *Elasticsearch* se encontram nas primeiras posições no que diz respeito a ferramentas de pesquisa e indexação. O *Sphinx Search* encontra-se na 35ª posição com uma popularidade de 9.08, bem inferior à do *Solr* e *Elasticsearch*. Perante esta informação e pelas características analisadas anteriormente segue-se a análise apenas com o *Solr* e *Elasticsearch*.

Decidir qual das duas ferramentas se adapta melhor ao problema em causa torna-se complicado, uma vez que as suas características são muito semelhantes. Segundo alguns autores tanto uma como outra merecem atenção:

*“Both search engines provide similar functionality, and features evolve quickly with each new version ... the functionality you need is covered by both, and, as is often the case with competitors, choosing between them becomes a matter of taste.”*, Gheorghe, et al. [19]

Tendo em consideração a opinião dos autores, apenas resta apelar à documentação. No *ElasticSearch*, vê-se que há uma tentativa de produzir boa documentação, mas como ainda é uma ferramenta relativamente recente esta carece de algum conteúdo. Já a do *Solr* é bastante boa e a comunidade é bastante ativa o que leva a tender para esta ferramenta.

## 2.4 Classificação

Com base no que foi dito até agora, é possível assumir que os resultados de uma pesquisa de elementos indexados seguem uma ordenação, ou não, por relevância, de acordo com os modelos clássicos e estruturados referidos anteriormente no tópico de recuperação de informação. Perante esta situação parece que o estudo levará a uma indexação simples de documentos sem uma perspectiva clínica associada, o que não é o caso. Pretende-se que a solução tenha teor clínico, isto é, que esteja ligada a uma classificação de termos, com um ponto de vista médico, de forma a apresentar ao utilizador resultados com mais significado e até mesmo acelerar o processo de

pesquisa. A classificação referida consiste na ordenação sistemática de todos os conceitos numa área científica, neste caso, na área da medicina.

### 2.4.1 Relações Hierárquicas

Segundo Rothwell, et al. [20], a relação mais importante usada para a classificação é uma relação hierárquica. Este tipo de classificação está dividida em:

- Classificação mono-hierárquica: um conceito só pode ter um pai na hierarquia;
- Classificação poly-hierárquica: um conceito pode ter vários pais na hierarquia.

Sendo assim, os autores referem ainda que uma mono-hierarquia não é suficiente para aplicações complexas, no entanto, as poly-hierarquias tendem a ser demasiado volumosas e redundantes não sendo, por isso, muitas vezes utilizadas. Para resolver tal problema os autores chegam à conclusão que a poly-hierarquia pode ser substituída por:

“Several disjoint monohierarchic classifications connected by a semantic model”, Rothwell, et al. [20]

Exemplo desta solução é o SNOMED, o qual se baseia num sistema de nomenclatura de termos médicos que providencia códigos, termos, sinónimos e definições usadas na documentação clínica.

### 2.4.2 UMLS

O SNOMED é uma nomenclatura que faz parte das ferramentas do UMLS que consiste num conjunto de ficheiros e *software* que contemplam diversos vocabulários de saúde e biomedicina. O UMLS é composto por três ferramentas (fontes de conhecimento):

- ***Metathesaurus*** que consiste em termos e códigos de diversos vocabulários, incluindo CPT, ICD-10-CM, LOINC, MeSH, RxNorm, e SNOMED CT;
- ***Semantic Network*** que consiste em relações semânticas entre uma vasta gama de categorias;
- ***SPECIALIST Lexicon and Lexical Tools*** que consiste no processamento de linguagem natural.

As duas últimas ferramentas são utilizadas para construir o *Metathesaurus*. A construção deste envolve o processamento de termos e códigos usando as ferramentas lexicais, agrupando termos sinónimos em conceitos, categorizando conceitos semanticamente e incorporando relações

e atributos providenciados pelos vocabulários. Nos tópicos seguintes são apresentadas as nomenclaturas SNOMED CT, MeSH e ICD-9.

## SNOMED CT

Como já foi referido o SNOMED consiste num conjunto de termos organizados hierarquicamente e que têm relações entre si. A estrutura do SNOMED encontra-se na figura seguinte.

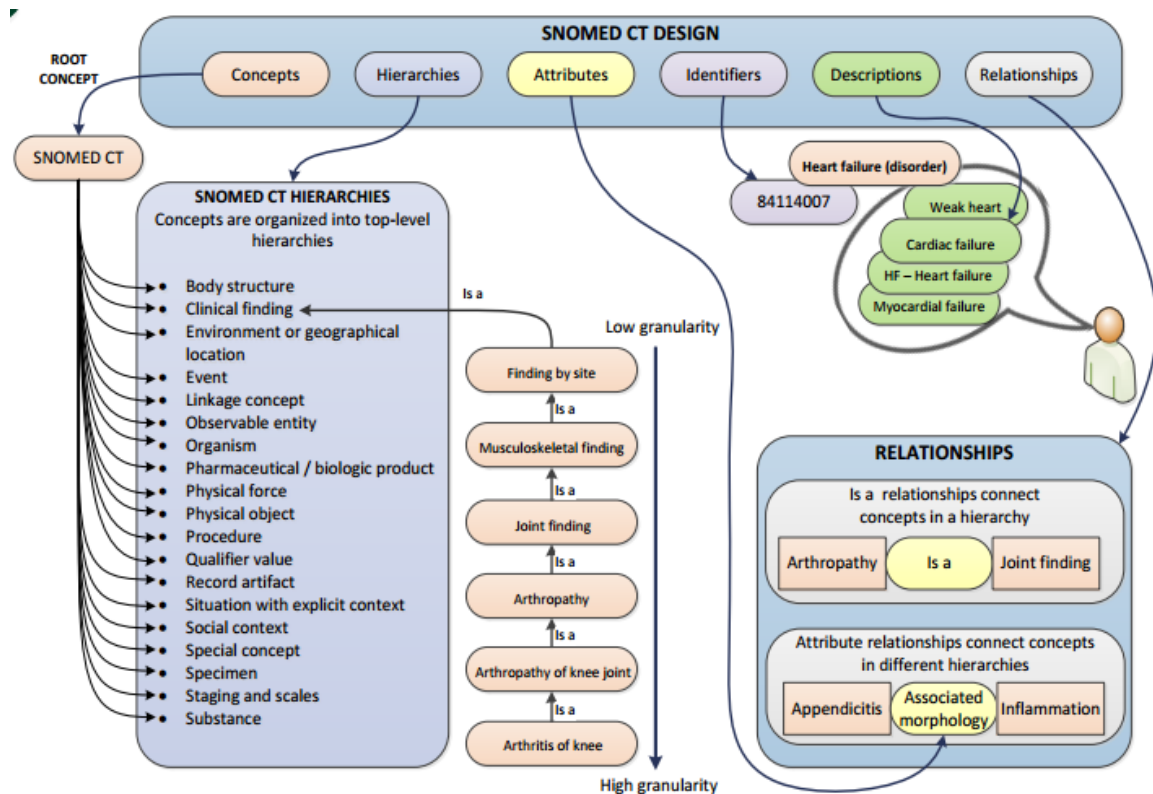


Figura 5 - Estrutura hierárquica e **relações** dos termos do SNOMED [5]

## ICD-9

O ICD-9 consiste num conjunto de códigos de diagnósticos e de procedimentos utilizados para classificação e codificação da informação de doenças e intervenções cirúrgicas, como é possível observar na tabela seguinte.



Código	Descrição
0010	Cholera due to vibrio cholerae
0011	Cholera due to vibrio cholerae el tor
0019	Cholera, unspecified
0020	Typhoid fever

**Tabela 2 - Exemplo de ICD-9-CM [21]**

## MeSH

MeSH é um vocabulário controlado de termos organizado hierarquicamente em 16 **árvores hierárquicas**. A cada árvore é atribuída uma letra de forma a identificar a mesma.

A	Anatomy
B	Organisms
C	Diseases
D	Chemicals and Drugs
E	Analytical, Diagnostic and Therapeutic Techniques and Equipment
F	Psychiatry and Psychology
G	Phenomena and Processes
H	Disciplines and Occupations
I	Anthropology, Education, Sociology and Social Phenomena
J	Technology and Food and Beverages
K	Humanities
L	Information Science
M	Persons
N	Health Care
V	Publication Characteristics
Z	Geographic Locations

**Figura 6 - Estrutura hierárquica do MeSH [1]**

Cada termo está localizado numa ou várias árvores simulando, desta forma, uma relação entre diferentes hierarquias, como o SNOMED. A estrutura da árvore tende de conceitos gerais para conceitos mais específicos.

Com a ajuda destas hierarquias pretende-se que o resultado de uma pesquisa por parte do utilizador venha associado a um maior valor no contexto clínico.

## 2.5 Resumo

Em suma, este capítulo permitiu realizar uma análise da recuperação, indexação e classificação de informação de um modo geral e direcionada para o meio clínico.

Relativamente à indexação, é possível afirmar que durante os últimos anos têm vindo a ser desenvolvidas diversas ferramentas que permitem a realização da mesma de forma eficaz. Exemplos disso são o *Apache Lucene*, *Apache Solr*, *Elasticsearch* e *Sphinx Search*. Com estas torna-se possível a indexação de documentos provenientes de diversas fontes e com diferentes formatos e, como são implementadas com a ajuda de alguns dos modelos referidos neste capítulo, os modelos clássicos e estruturados, permitem que a informação apresentada ao utilizador seja a mais relevante perante a pesquisa feita pelo mesmo. No entanto, os modelos referidos apenas dizem respeito a uma indexação geral, isto é, a uma indexação sem conteúdo clínico e, por isso, com este tipo de abordagem podem ser rejeitados resultados com bastante importância. Face a este problema alguns autores sugerem a utilização de métodos de classificação como forma de apresentar resultados com maior relevância para o utilizador. No âmbito dos métodos de classificação surgem assim nomenclaturas que, de uma forma hierárquica, permitem relacionar termos clínicos, como por exemplo: SNOMED, ICD-9 e MeSH.

Concluindo este capítulo, é possível afirmar que os métodos referidos anteriormente permitem que a pesquisa de termos clínicos seja apresentada ao utilizador de forma ordenada, de acordo com a relevância, apresentado, por isso, informação mais importante e com maior utilidade e valor para o utilizador.





## **Capítulo 5**

# **Conclusões e Trabalho Futuro**

Neste capítulo serão apresentadas algumas conclusões sobre os objetivos já concluídos e também dos não concluídos de forma a avaliar o trabalho que será desenvolvido num futuro próximo.

### **3.1 Satisfação dos Objetivos**

Como já foi referido em vários pontos deste documento, o objetivo desta dissertação é criar uma solução que disponibilize às entidades médicas uma nova forma de acesso à informação, mais rápida e eficaz do que a existente atualmente.

Para que os objetivos referidos sejam atingidos, nesta primeira parte do trabalho, foram estudadas diversas ferramentas, modelos e métodos que de alguma forma podem contribuir para a concretização dos mesmos.

Face ao estudo efetuado, só posso estar satisfeito com o trabalho desenvolvido, pois este permitiu-me conhecer melhor como funcionam os processos de recuperação, indexação e classificação de informação, assim como adaptar cada um deles ao contexto em causa.

### 3.2 Trabalho Futuro

Como ainda só foi realizada a primeira parte do trabalho, existem imensas tarefas por realizar e, por isso, na figura seguinte é apresentado o plano de trabalho que penso realizar na segunda parte deste projeto.

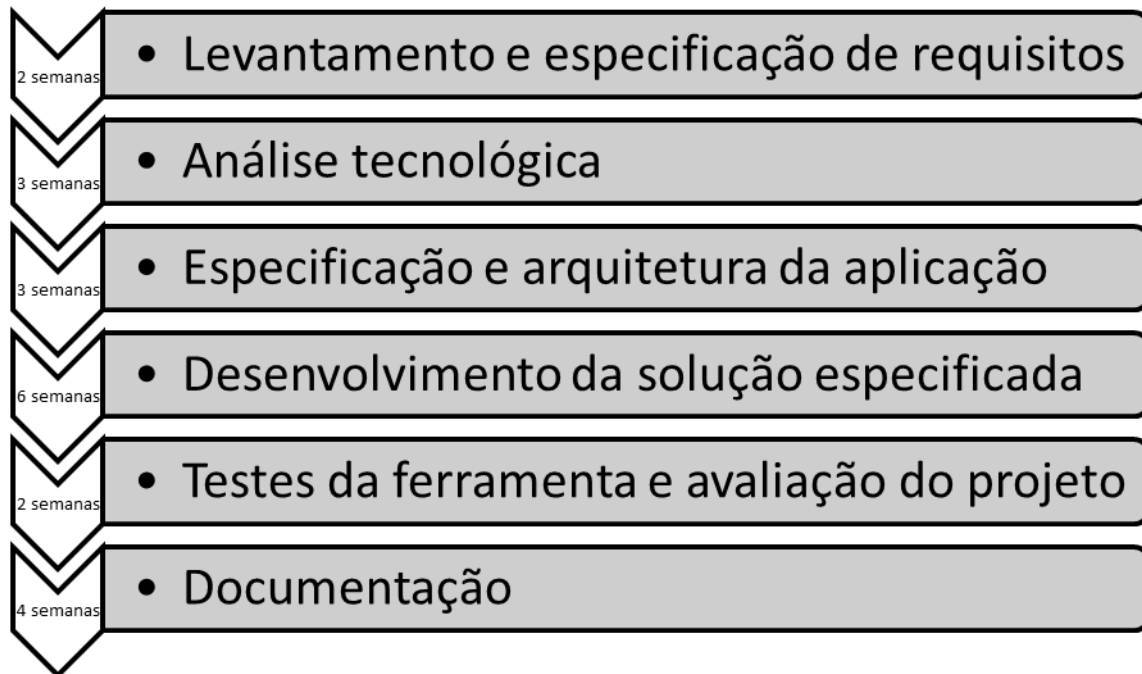


Figura 7 - Planeamento

Relativamente aos dois primeiros tópicos do planeamento, já foram iniciados ao longo deste estudo do estado da arte e, por isso, possivelmente serão cumpridos antes da data prevista. No entanto e, especialmente no que diz respeito à análise tecnológica, é ainda necessário realizar alguns testes práticos com as ferramentas de forma a concluir a **análise** iniciada e descrita neste documento. Após efetuar a análise tecnológica, procede-se à especificação e arquitetura da aplicação, seguida do desenvolvimento da mesma.

Uma vez concluída a etapa anterior, pretendem-se realizar testes ao *software* com o objetivo de garantir a estabilidade da aplicação. Nas mesmas duas semanas em que serão realizados os testes pretende-se ainda que seja feita uma avaliação ao projeto, ou seja, uma avaliação que permita perceber se a solução produzida realmente causa um impacto positivo no meio clínico. Para este ponto, é ainda necessário decidir como será feita essa avaliação.

Por fim, nas últimas quatro semanas, espera-se que haja a escrita da documentação referente à solução proposta assim como a continuação da escrita deste documento.

# Referências

- [1] (2016). *Home - MeSH - NCBI*. Available: <http://www.ncbi.nlm.nih.gov/pubmed/>
- [2] (2016). *Sphinx | Open Source Search Engine*. Available: <http://sphinxsearch.com/>
- [3] (2016). *DB-Engines Ranking*. Available: <http://db-engines.com/en/ranking/search+engine>
- [4] E. Hatcher and O. Gospodnetic, *Lucene in Action (In Action series)*: Manning Publications Co., 2004.
- [5] Ihtsdo. (2016). *SNOMED CT Document Library*. Available: [http://ihtsdo.org/fileadmin/user\\_upload/doc/](http://ihtsdo.org/fileadmin/user_upload/doc/)
- [6] C. R. L. Coelho, "Acesso e Recuperação de Informação em Catálogos Bibliográficos Online," Faculdade de Engenharia da Universidade do Porto, 2014.
- [7] M. Lesk, "The seven ages of information retrieval," ed, 1996.
- [8] C. D. Manning, P. Raghavan, H. Sch\, \#252, and tze, *Introduction to Information Retrieval*: Cambridge University Press, 2008.
- [9] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [10] D. Hiemstra, *Using language models for information retrieval*: Taaluitgeverij Neslia Paniculata, 2001.
- [11] D. L. Lee, H. Chuang, and K. Seamons, "Document ranking and the vector-space model," *Software, IEEE*, vol. 14, pp. 67-75, 1997.
- [12] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *Journal of the American Society for Information science*, vol. 27, pp. 129-146, 1976.
- [13] R. Baeza-Yates and G. Navarro, "XQL and proximal nodes," *Journal of the American Society for Information Science and Technology*, vol. 53, pp. 504-514, 2002.
- [14] D. A. Hull, "Stemming algorithms: A case study for detailed evaluation," *JASIS*, vol. 47, pp. 70-84, 1996.
- [15] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*: Manning Publications Co., 2010.
- [16] T. Grainger and T. Potter, *Solr in action*: Manning Publications Co., 2014.
- [17] C. Mattmann and J. Zitting, *Tika in Action*: Manning Publications Co., 2011.
- [18] (2016). *Apache Solr*. Available: <http://lucene.apache.org/solr/>
- [19] R. Gheorghe, M. L. Hinman, and R. Russo, *Elasticsearch in Action*: Manning Publications Co., 2015.
- [20] D. Rothwell, F. Wingert, R. Cote, R. Beckett, and J. Palotay, "Indexing Medical Information: The Role of SNOMED," in *Proceedings/the... Annual Symposium on*

- Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, 1989, pp. 534-539.
- [21] (2016). *ICD - ICD-9-CM - International Classification of Diseases, Ninth Revision, Clinical Modification*. Available: <http://www.cdc.gov/nchs/icd/icd9cm.htm>