UNIVERSIDADE DO SUL DE SANTA CATARINA

EZIRIO BENTO CARLESSO BORGES

RECUPERANDO INFORMAÇÕES TEXTUAIS, UTILIZANDO RECURSOS SEMÂNTICOS

Palhoça

2014

EZIRIO BENTO CARLESSO BORGES

RECUPERANDO INFORMAÇÕES TEXTUAIS UTILIZANDO RECURSOS SEMÂNTICOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof.Flávio Ceci, M.Eng.

Florianópolis

2014

EZIRIO BENTO CARLESSO BORGES

RECUPERANDO INFORMAÇÕES TEXTUAIS, UTILIZANDO RECURSOS SEMÂNTICOS

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Graduação em Sistemas de Informação da Universidade do Sul de Santa Catarina.

Palhoça, 09 de Maio de 2014.

Professor e orientador Flávio Ceci, M.Eng.
Universidade do Sul de Santa Catarina

Prof. Aran Bey Tcholakian Morales, Dr. Eng.
Universidade do Sul de Santa Catarina

Prof. Julio Gonçalves Reinaldo, abreviatura da titulação.
Universidade do Sul de Santa Catarina

Dedico este trabalho aos meus pais, Onice Maria Carlesso Borges e Manoel Rodrigues Borges Neto, à meus irmãos João Vitor Carlesso Borges e Manoel Vergilio Carlesso Borges, e à minha namorada Fernanda de Ávila Moukarzel.

AGRADECIMENTOS

Agradeço aos meus pais, Onice Maria Carlesso Borges e Manoel Rodrigues Borges neto, pelo exemplo de vida, por me apoiar em toda essa jornada da faculdade e pelo esforço para que eu concluísse minha graduação.

Meus agradecimentos à minha namorada Fernanda de Ávila Moukarzel, por estar ao meu lado durante essa caminhada e ser minha grande inspiração para alcançar meus objetivos.

Agradeço também meus irmãos João Vitor Carlesso Borges e Manoel Vergilio Carlesso Borges, pela cumplicidade.

Meu muito obrigado ao colega de trabalho e meu orientador, professor Prof. M.Eng. Flávio Ceci, que se colocou a minha disposição para me ajudar e me coordenar em tudo que foi preciso, assim possibilitando o desenvolvimento deste trabalho. Uma pessoa ímpar, e um exemplo a ser seguido.

E, também, agradeço todos os professores do curso de Sistemas de Informação, da Universidade do Sul de Santa Catarina, pelo profissionalismo e por transmitirem seu conhecimento para os alunos, e, em especial ao Prof. Dr. Engº. Aran Bey Tcholakian e ao Prof. Julio Gonçalves Reinaldo, por terem aceitado o meu convite para a participação da banca examinadora.

“O insucesso é apenas uma oportunidade para recomeçar de novo com mais inteligência.” (Henry Ford).

RESUMO

Retornar documentos relevantes para uma pesquisa é um problema conhecido para as empresas. Principalmente com bases de arquivos muito grandes. A busca semântica veio para ajudar os sistemas de recuperação de informação a entender o que o usuário precisa. Tendo este cenário como base, este trabalho tem como objetivo desenvolver um sistema de recuperação de informação que utiliza uma base de conhecimento para realizar a anotação semântica e separe os documentos por conceitos. A base de conhecimento utilizada no trabalho foi retirada do Vocabulário Controlado do Governo Eletrônico e os documentos inseridos no sistema são trabalhos de conclusão de curso da Universidade de Sul de Santa Catarina, dos cursos de Ciências da Computação e Sistemas de Informação. O protótipo proposto é baseado em dois perfis, o Administrador, que vai inserir documentos no sistema, e o usuário comum (não logado), chamado de Usuário, que vai consumir o conteúdo que o Administrador inserir. O resultado atingido com o trabalho foi satisfatório, o sistema indexa os arquivos inseridos no sistema e anota semanticamente, para posteriormente o Usuário (perfil não logado) realizar pesquisas, o resultado de uma pesquisa é separado por conceitos, relacionados ao termo pesquisado. Para a avaliação do protótipo, foi realizado um questionário com pessoas de dois perfis, pessoas comum e pessoas que trabalham com tecnologia.

Palavras-chave: Busca Semântica. Recuperação de Informação.

LISTA DE ILUSTRAÇÕES.

SUMÁRIO

1 - INTRODUÇÃO

1.1 - DEFINIÇÕES DO PROBLEMA

Recuperar documentos irrelevantes é um problema que vem afetando cada vez mais o desempenho em buscas de informações. Com a grande quantidade de documentos digitais nas empresas, procurar informações, em tempo hábil, é uma necessidade para que não seja despendido tempo com informações irrelevantes. (CECI, 2010)

Cardoso (2000) entende que um dos problemas em recuperação de informação é que os autores nem sempre usam as mesmas palavras que os usuários para descrever o mesmo conceito.

Um dos grandes problemas de uma busca é a perda de tempo gasta em procurar documentos relevantes. Junto com esse problema, vem outro problema, o da busca pelo contexto da palavra que foi procurado. Uma palavra tem vários contextos, por exemplo, para palavra "java" podem ser retornados documentos falando sobre a ilha java, ou sobre o café java, ou até mesmo sobre a linguagem de programação java. (CECI, 2010)

> A crescente complexidade dos objetos armazenados e o grande volume de dados exigem processos de recuperação cada vez mais sofisticados. Diante deste quadro, recuperação de informação apresenta a cada dia, novos desafios e se configura como uma área de significância maior. (CARDOSO, 2000, p. 1).

Junto a necessidade de que uma busca seja rápida, a criação de índices é uma parte fundamental para um processo de recuperação de informação. Com essa indexação, a busca não tem a necessidade de percorrer o documento inteiro, comparando textos. (FERNEDA, 2003).

Com o crescimento de documentos digitais, surgiu a dificuldade no gerenciamento e usufruto de informações realmente relevantes aos interesses de quem as procura. Assim;, se faz o seguinte questionamento:

Como criar uma busca textual eletrônica com índices, por meio de uma estrutura de dados que suporte a relação entre documentos e as classes da base de conhecimento, para evitar desperdício de tempo com os resultados irrelevantes?

1.2 - OBJETIVOS DO TRABALHO

Os objetivos encontram-se divididos em objetivo geral e objetivos específicos.

1.2.1 - Objetivo geral

O presente trabalho tem por objetivo:

Desenvolver um sistema que recupere informações textuais a partir de recursos semânticos para desambiguação de textos, e traga resultados pertinentes de acordo com a pesquisa do usuário. Entende-se por desambiguação, palavras que contém a mesma escrita mas tem significados diferentes. Por exemplo, a palavra “jaguar”, pode ser a marca de carro, o animal felino, ou o cartunista.

1.2.2 - Objetivos específicos

Formularam-se os seguintes objetivos específicos:

- modelar um índice textual que facilite a recuperação e navegação dos resultados;
- propor uma estrutura de dados que suporte a relação entre os documentos e as classes da base de conhecimento;
- desenvolver um módulo de indexação textual e anotação de documentos baseados em conceitos da base de conhecimento;
- construir um protótipo de busca para web;
- validar o protótipo a partir de um estudo de caso.

1.3 - JUSTIFICATIVA E RELEVÂNCIA DO TEMA

De acordo com Popov et al. (2003, apud CECI; Woszezenki; Gonçalves, 2014 p.4) “a anotação semântica é um processo de geração de metadados específicos para possibilitar novos métodos de acesso à informação e mesmo estender os métodos existente”.

A busca semântica tem atraído grande interesse das indústrias e dos pesquisadores, no que resulta uma grande variedade de soluções para diversas tarefas diferentes. (BLANCO et al. 2013).

A busca de informações tradicional nada mais é que a comparação entre palavras chave, não tendo suporte a comparação semântica. Nesse cenário, é que surge a recuperação de informação semântica que se baseia em ontologias de domínio, onde nelas, vão conter mapas de conhecimento (*KnowledgeMap*), que vão garantir uma maior precisão na busca de resultados. (TANG, CHEN. 2011).

Segundo ZHANG et al (2012), a ontologia é uma abordagem para representar o mundo real, com a qual se consegue modelar o conhecimento, fornecendo uma compreensão não ambígua para o usuário e o sistema se comunicarem.

Com a necessidade do mercado de que os sistemas de recuperação de informação sejam cada vez mais eficazes, para que não haja desperdício de tempo com informações que não são necessárias, tem se buscado soluções de pesquisas semânticas. O fato de existirem inúmeros jeitos de se formar uma pesquisa que tem o mesmo significado, tem grande impacto em uma pesquisa, tentar compreender o que o usuário quer receber de uma pesquisa é uma tarefa difícil para os motores de busca.

A utilização de recursos semânticos traz muitos benefícios para um sistema de recuperação de informação. Segundo Ceci; Woszezenki; Gonçalves a desambiguação de temas é um desses benefícios, onde é possível identificar em qual foco o documento se encontra (exemplo: duas palavras iguais, mas com significados diferentes). Os autores também citam outro beneficio, a organização de documentos utilizando uma base de conhecimento, facilitando a organização de uma base de documentos utilizada por alguma instituição.

1.4 - ESTRUTURA DA MONOGRAFIA

Este trabalho está dividido em seis capítulos e da seguinte forma: introdução, referencial bibliográfico, método de pesquisa, modelagem da proposta de solução, sistema proposto, estudo de caso e validação e trabalhos futuros junto com a conclusão.

Capítulo 1 – Introdução: Apresentação da visão geral do tema pesquisado neste trabalho. Apresenta também a problemática junto com a sua justificativa, os objetivos específicos e o objetivo geral.

Capítulo 2 – Referencial Bibliográfico: Fundamentação teórica que serviu de base para a pesquisa.

Capítulo 3 – Método de pesquisa: Estrutura para o desenvolvimento do projeto.

Capítulo 4 – Modelagem da proposta de solução:

Capítulo 5 – Sistema proposto, estudo de caso e validação: Neste, será encontrado a validação de um caso de uso de acordo com o problema definido.

Capítulo 6 – Conclusão e trabalhos futuros: Capítulo final, onde vai ser apresentada a conclusão do autor da pesquisa e recomendações para futuros trabalhos.

2 - REFERENCIAL BIBLIOGRÁFICO

Neste capítulo são abordados os temas referentes à recuperação de informação, tais como: extração de informação, modelo booleano, modelo de fuzzy, modelo vetorial, indexação, desambiguação de palavras e tesauros. Na sequência, serão levantados conceitos sobre anotações de documentos, extração de informação e construção de ontologias.

2.1 - RECUPERAÇÃO DE INFORMAÇÃO

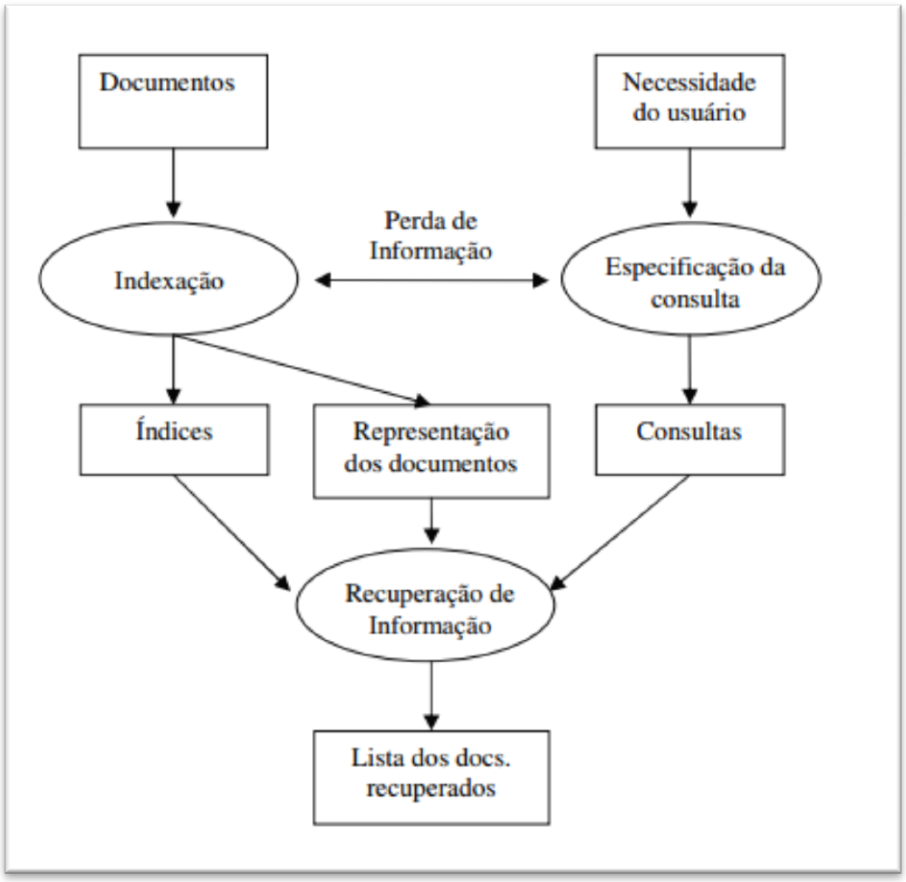
Segundo Cardoso (2002), a recuperação de informação faz parte de uma subárea da ciência da computação, onde se estuda o armazenamento e a recuperação de documentos, que geralmente são textos. O processo de recuperação consiste em gerar uma lista de documentos recuperados para atender a uma consulta formulada pelo usuário. Esta lista de documentos é classificada em ordem decrescente de um grau de similaridade entre o documento e a consulta.

Para Baeza (1999, p.73), "o objetivo preliminar de um sistema de recuperação de informação é recuperar todos os documentos que são relevantes a uma solicitação do usuário com uma quantidade mínima de documentos não relevantes".

> O objetivo principal da RI é tornar o acesso mais fácil aos documentos de maior relevância conforme a necessidade de informação do usuário. Essa necessidade normalmente é simbolizada por meio de uma busca por palavra-chave. A recuperação de informação, nesse contexto, consiste basicamente na determinação de quais documentos de uma coleção contêm as palavras-chaves da consulta realizada pelo usuário. A dificuldade está não somente em extrair a informação, mas sim também em decidir sua relevância.(Ceci, 2010, p.35).

Um sistema de recuperação de informação pode ser representado conforme a Figura 1. Com a necessidade do usuário, uma pergunta é formulada, iniciando-se o processo de recuperação. A partir da consulta recupera-se uma lista de documentos.

Figura 1 – Componente de um sistema de recuperação de informação



Fonte: Gey (1992).

De acordo com Schreiber e outros (2008), a elaboração de pesquisa é difícil e geralmente tem uma grande diferença entre o que o usuário está procurando e o que é mostrado na consulta formulada. Essa diferença é gerada pelo limitado conhecimento do usuário, o universo de pesquisa e pelo formalismo da linguagem de consulta.

2.1.1 – Modelos de recuperação de informação

Nesta seção do trabalho, são abordados os modelos de recuperação de informação, sendo eles os modelos clássicos, modelo booleano, modelo vetorial e o modelo probabilístico, e o modelo de fuzzy.

2.1.1.1 – Modelo booleano

Modelo de recuperação de dados mais comum entre os sistemas de recuperação de informação, o modelo booleano é fácil de utilizar. É baseado em conceitos da álgebra booleana, em que se utiliza os operadores lógicos E, OU e NÃO para aprimorar suas consultas (KORFHAGE, 1997).

Souza (2006) diz que o modelo booleano:

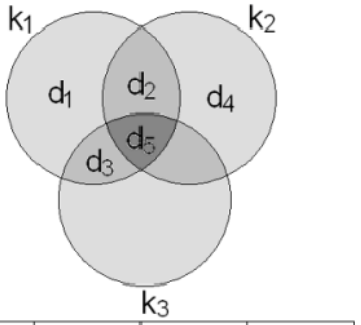
> É baseado na teoria dos conjuntos, é simples e elegante, embora não seja dos mais eficazes. Para cada query, são recuperados todos os documentos que possuem os termos nas condições especificadas pelo usuário, que ainda pode utilizar os operadores booleanos E, OU e NÃO para estabelecer relações específicas de ocorrência com as palavras-chave, de forma a especificar os documentos a serem recuperados.(SOUZA, 2006, p.6).

No entanto, o modelo booleano apresenta alguns problemas. Nesse sentido, Ceci (2010) cita alguns problemas com o modelo booleano:

- Não poder controlar o tamanho do resultado, fazendo com que traga muitos itens para a consulta.
- Não conseguir utilizar pesos para uma consulta mais eficiente e os resultados não são ordenados de acordo com a relevância.
- A seleção dos termos que irá compor a consulta, quando não feita por um especialista, pode ser bastante complicada.

Cardoso (2000), completando o que foi dito antes, diz que alguns dos principais problemas do modelo booleano "são a ausência de ordem na resposta, e as respostas podem ser nulas ou muito grandes" e suas vantagens são a fácil implementação e a expressividade completa das expressões. A figura 2 demonstra o funcionamento do modelo booleano.

Figura 2 – Representação do modelo booleano.



	k_1	k_2	k_3
d_1	1	0	0
d_2	1	1	0
d_3	1	0	1
d_4	0	1	0
d_5	1	1	1

$q = k_1$ and (k_2 or (not k_3))

Fonte: Ceci, 2010

Para suprir as deficiências do modelo booleano comum, Salton, Fox e Wu (1983) propuseram o modelo booleano estendido, que tenta reunir a potencialidade da expressão booleana com a precisão do modelo vetorial. Por um lado busca a flexibilidade do modelo booleano com uma introdução do conceito de relevância e, por outro lado, procura dar maior potencial para as buscas vetoriais com o uso dos operadores booleanos. (FERNEDA, 2003).

2.1.1.2 – Modelo vetorial

Segundo Cardoso (2000), o modelo espaço vetorial representa suas consultas e seus documentos como vetores de termos, os quais são ocorrências únicas nos documentos. O vetor de retorno de uma consulta é representado por um cálculo de similaridade.

Para calcular o grau de similaridade entre dois vetores em um espaço vetorial de "n" dimensões, é preciso achar o co-seno do ângulo formado por estes vetores, utilizando-se a fórmula representada na figura 3. (FERNEDA, 2003).

Ferneda (2003) também explica que o grau de similaridade é calculado através da associação dos pesos da indexação e dos termos da expressão de busca, em que é possível obter documentos que respondem parcialmente a uma expressão de busca. A baixo temos a fórmula1 utilizada para efetuar o cálculo, onde "w" é o peso do elemento "i" nos vetores "x" e "y"

Fórmula 1 - Cálculo de similaridade

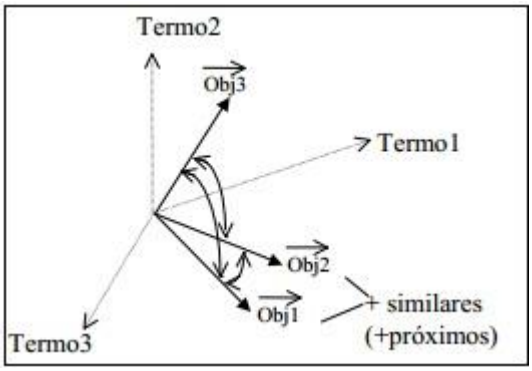
$$sim(x,y) = \frac{\sum_{i=1}^{t}(W_{i,x}\, x\, W_{i,y}\,)}{\sqrt{\sum_{i=1}^{t}\,(W_{i,x})^2}\; x\; \sum_{i=1}^{t}\,(W_{i,y})^2}$$

Fonte: Ceci, 2010

Wives (2002, p.39) afirma que :

> Cada elemento do vetor é considerado uma coordenada dimensional. Assim, os documentos podem ser colocados em um espaço euclidiano de n dimensões, (onde n é o número de termos) e a posição do documento em cada dimensão é dada pelo seu peso.

Figura 3 - Modelo espaço-vetor.



Fonte: Wives, 2002

A distância entre documentos indicam o seu grau de similaridade, documentos que contêm os mesmos termos acabam sendo colocados em uma mesma região do espaço, que, em teoria, tratam de um assunto parecido. (WIVES, 2002).

Para Manning, Raghavan e Schutze (2008), os termos relevantes de um documento também podem ser calculados pelo *TF-IDF*. Com a frequência dos documentos que contêm o termo a ser pesquisado (*DF – Document Frequency*), é calculado o IDF, em que é encontrada a quantidade de vezes que o termo aparece no conjunto de documentos. A fórmula 2, abaixo, ilustra o cálculo.

Fórmula 2 – Frequência dos documentos

$$IDF = \log\left(\frac{|D|}{DF}\right)$$

Onde "DF" é a frequência dos documentos, "D" a cardinalidade dos documentos armazenados. Depois de descobrir o IDF, teremos todos os termos para calcular o TF-IDF(d,v) onde "d" é a dimensão e "v" é o vetor. A fórmula 3 mostra como deve-se aplicar os valores:

Fórmula 3 – Cálculo dos termos mais relevantes de um documento

$$TF - IDF(d, v) = TF(d, v) \times IDF(d)$$

Na opinião de Korfhage (1997), o processo de atribuir pesos aos termos de um documento é uma tarefa muito complexa, entretanto os pesos podem ser atribuídos de uma

forma automática, com base na contagem de frequência do termo em um documento. É possível afirmar que quanto mais frequente um termo em um documento, mais importante esse documento é.

Ferneda (2003) afirma que a diferença entre o modelo booleano e o modelo vetorial é que o modelo vetorial permite calcular pesos tanto para os termos indexados quanto para as expressões de busca. Com essa diferença, é possível ter um indicador que representa a relevância de cada documento em relação à busca.

2.1.1.3 –Modelo *fuzzy*

Segundo Leite (2009), o modelo *fuzzy* surgiu para lidar com a limitação de pertinência binária do modelo booleano. A estrutura para representação formal dos relacionamentos deste modelo é baseada no conceito da matemática de relação. Enquanto as relações matemáticas clássicas descrevem apenas presença ou ausência entre elementos de dois conjuntos, as relações do modelo *fuzzy* permitem calcular o grau da relação entre os elementos (0.0 até 1.0). Este modelo é uma extensão do modelo booleano, mas com um método de ordenação, no qual é possível extrair a aproximação do retorno da consulta com os documentos. (BAEZA-YATES, RIBEIRO NETO, 1999).

Souza (2006, p. 6) afirma que o modelo de *fuzzy*:

> Busca-se estender o conceito da representação dos documentos por palavras-chave, assumindo que cada query determina um conjunto difuso e que cada documento possui um grau de pertencimento a esse conjunto, usualmente menor do que 1. O grau de pertencimento pode ser determinado pela ocorrência de palavras expressas na query, tal como no modelo booleano, mas pode também utilizar um instrumento – como um tesauro – para determinar que termos relacionados semanticamente aos termos índice também confiram algum grau de pertencimento ao conjunto difuso determinado pela query.

Um documento pode ser visto como um conjunto *fuzzy* de termos, em que os pesos dependem do documento e do termo em questão. Dessa forma, uma representação fuzzy de um documento é baseada na função *F(d, t,)* em que o termo em questão é representado por *t* e o documento por *d*. (FERNEDA, 2003).

Wives (2002) explica que, dependendo da função adotada para cada operador, pode haver termos prejudicados na consulta, a exemplo da função mínimo, que desconsidera o termo mais importante. A escolha de um operador deve ser vista com uma tarefa muito importante, porque cada uma obtém resultados diferentes, e a análise da consulta tem como objetivo identificar e proporcionar o melhor resultado. A lógica fuzzy ou lógica difusa tem como objetivo trabalhar com a incerteza de uma forma rigorosa e sistemática. (FERNEDA, 2003).

Wives (2002) cita que as funções correspondentes mais comuns são a função máxima, em que há o retorno do maior entre os dois valores (substitui o operador *and*), função mínimo, retorno do menor valor (substitui o operador *or*) e a função complemento de um que retorna o complemento de um para o termo seguinte ao operador (substitui o operador *not*).

2.1.1.4 - Modelo Probabilístico

O modelo probabilístico surgiu da teoria das probabilidades, estudada na matemática, e tem como objetivo realizar e analisar experimentos aleatórios, que, repetidos em condições idênticas podem apresentar uma gama de resultados diferentes e imprevisíveis. (FERNEDA, 2003).

> No modelo probabilístico a função de similaridade pode aproveitar-se das informações estatísticas de distribuição dos termos contidos no índice. Com isso, determinados parâmetros podem ser ajustados de acordo com a coleção em questão, obtendo assim resultados mais relevantes. (WIVES, 2002 p.40).

Para calcular a chance de um evento ocorrer, deve-se calcular a razão entre o número de elementos E, representados por *n(e)* e o número de elementos de *S*, representado por *n(S)*. (FERNEDA, 2003).

Fórmula 4 – Probabilidade de um evento ocorrer

$$p(E) = \frac{n(E)}{n(S)}$$

No lançamento de um dado de seis lados, o espaço amostral é S={1,2,3,4,5,6} e a probabilidade de sair o número 3 (E={3}) é:

$$p(3) = \frac{n(1)}{n(6)} = \frac{1}{6}$$

Cardoso (2003) afirma que o modelo probabilístico foi baseado no principio probabilístico de ordenação. Esse princípio tem como base a hipótese de que a relevância de um determinado documento para uma consulta seja independente de outros documentos. Para auxiliar nesse modelo, a principal ferramenta matemática é o teorema de Bayes.

Deve ser calculada a fórmula *P(+Rq|d)* para achar a probabilidade de um documento *d* seja relevante para uma consulta *q*, e *P(-Rq|d)* para a probabilidade do documento não ser relevante para uma consulta. O documento é considerado relevante para uma consulta se a probabilidade de o documento ser relevante for maior do que a probabilidade deste mesmo documento não o ser. O resultado é decidido em um fator *Wd|q*, mostrado na fórmula 4. (CARDOSO, 2003).

Fórmula 4 – Probabilidade de um evento ocorrer

$$W_{d|q} = \frac{P(+R_q|d)}{P(-R_q|d)}$$

Onde **+Rq** é o conjunto de documentos relevantes e **–Rq** o conjunto de documentos não relevantes e **d** é o documento.

2.1.2 – Stopwords

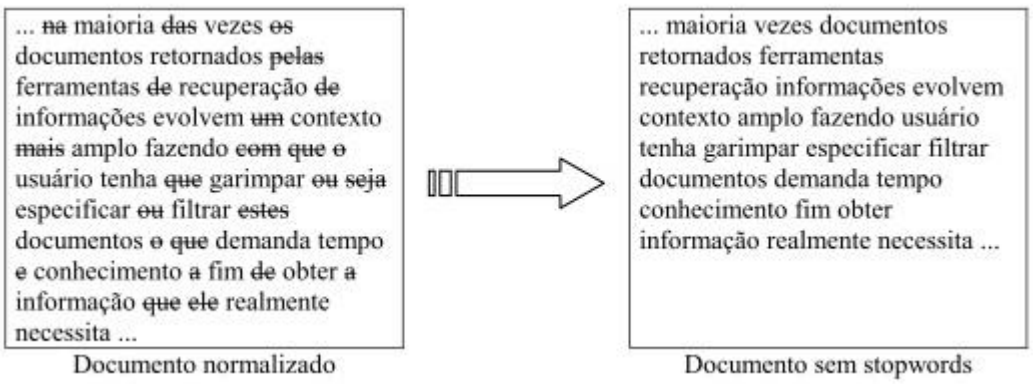
Stopwords são palavras que podem ser considerados irrelevantes quando está analisando um documento. Para entender melhor o conceito sobre stopwords, segue a definição proposta por Galho e Moraes (2003, p. 18):

> As palavras que não são passíveis de serem representantes de alguma categoria são conhecidas como stopwords ou palavras negativas e podem ser representadas por artigos, pronomes, preposições, advérbios e outras palavras que se apresentem com elevada ou baixa frequência nos textos.

Korfhage (1997) leciona que, antes de sujeitar um documento a um processo de indexação, deve-se limpar todas as ocorrências de *stopwords*.

A figura 4 mostra um exemplo de remoção de palavras que não são necessárias para uma consulta. Essas palavras são dificilmente utilizadas na recuperação de informação, pois no processo de indexação só tornaria os índices maiores do que o necessário para a consulta. (WIVES, 2002).

Figura 4 – Identificação de *Stopwords*.



Fonte: Wives, 2002

Wives (2002) também afirma que existem estudos que oferecem listas de *stopwords* (palavras que não são relevantes para a pesquisa) que podem ser utilizadas para auxiliar processo de remoção de *stopwords*.

2.1.3 – Stemming

Wives (2002) explica que o método *stemming*, tem o objetivo eliminar as variações morfológicas de uma palavra. Essas variações são eliminadas por meio da identificação do radical da palavra. Para chegar ao radical são extraídos o prefixo e o sufixo de uma palavra, os radicais resultantes da extração são adicionado a uma estrutura de índices.

> As características de gênero, número e grau das palavras são eliminadas. Isso significa que várias palavras acabam sendo mapeadas para um único termo, o que aumenta a abrangência das consultas. Com essa técnica o usuário não necessita preocupar-se com a forma ortográfica com a qual uma palavra foi escrita no texto. Assim, uma ideia, independente de ter tendo sido escrita através de seu substantivo, adjetivo ou verbo, é identificada por um mesmo (e único) radical. Essa aparente vantagem ocasiona uma diminuição na precisão, já que o usuário não consegue mais procurar por uma palavra específica. (WIVES, 2002).

Ebecken, Lopes e Costa (2005) explicam que os algoritmos de *stemmimg* não usam informações do contexto para determinar o sentido correto de cada palavra, e casos em que o contexto ajuda no processo *stemming* não são frequentes, sendo que a maioria das palavras pode ser considerada como apresentando um significado único. Os erros que foram resultados de uma análise de sentido incerta das palavras não compensam os ganhos que possam ser obtidos pelo aumento da precisão do *stemming*.

2.1.4 – Tesauro

Para Cavalcanti (1998) tesauro é:

> Uma lista estruturada de termos associada empregada por analistas de informação e indexadores, para descrever um documento com a desejada especificidade, em nível de entrada, e para permitir aos pesquisadores a recuperação da informação que procura.

Para Gonzales e Lima (2003, apud CECI 2010), os tesauros podem desempenhar as funções de auxiliar na classificação de documentos e seus conceitos, auxiliar na produção e tradução de textos, bem como no processo decisório de classificação de assuntos e apoio a recuperação de informação.

Jesus (2002) explica que, para a construção de um tesauro, é primordial examinar seus elementos e selecionar aquele que produzirá um bom desempenho para um determinado sistema. Para garantir a recuperação de um número apetente de documentos relevantes e assegurar uma seleção precisa, deve-se fazer o controle da terminologia.

O principal problema a ser considerado é o uso de termos similares ou relacionados, esses termos não podem ser manipulados por um algoritmo simples. Um bom tesauro pode contar sinônimos e antônimos para cada palavra em questão, e esse tesauro pode ser utilizados durante o processo de armazenamento de um documento, fazendo o controle da terminologia. (KORFHAGE, 1997).

2.1.5 – Indexação

Segundo Ebecken, Lopes e Costa (2003, apud Ceci 2010), "a indexação tem como função permitir que se efetue uma busca em texto sem a necessidade de varrer o documento inteiro".

Baeza-Yates e Ribeiro-Neto (1999) citam as três principais técnicas para a construção de arquivos de indexação, sendo eles:

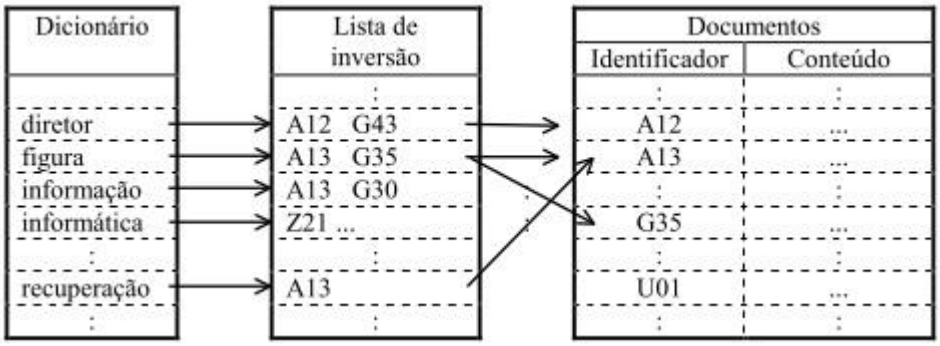
- arquivos invertidos;
- arquivos de assinatura;
- árvore e vetores de sufixos.

Wives (2002) explica que a técnica de arquivos invertidos nada mais é que uma lista ordenada de palavras, em que cada palavra possui uma referência para os documentos onde ela aparece. Essa estrutura geralmente é composta por três arquivos, o dicionário ou lista de palavras, a lista de inversão e os documentos. Ao ser encontrada uma palavra no dicionário, identifica-se sua lista invertida de documentos correspondentes.

> Um índice invertido é uma estrutura de dados em que se relaciona cada palavra com todos os documentos que a contêm, e também armazena a frequência com que a palavra ocorre no documento. A utilização do índice invertido torna mais fácil a busca de informação em documentos. O autor afirma que as versões mais sofisticadas de índice invertido também armazenam as posições do documento.(MANNING, SCHUTZE, 1999 apud CECI, 2010, p. 46).

Na figura 5, podemos visualizar como é a estrutura de uma lista invertida.

Figura 5 – Estrutura de uma lista invertida.



Fonte: Wives, 2002

O principal objetivo do método de assinatura é prover em um teste em que vai indicar quais são os registros mais relevantes para a consulta do usuário, eliminando a maioria dos resultados irrelevantes. (WIVES, 2002).

Wives (2002) explica que, em uma estrutura de arquivos de assinatura os documentos costumam ser divididos em blocos, com o objetivo de evitar que as assinaturas sejam muito densas, o que ocasionaria palavras com a assinatura similares (colisões). Quanto maior for a assinatura, menor a chance de colisões.

Cada palavra dentro de uma assinatura é mapeada com um número fixo de bits denominado de assinatura. Depois de determinar os códigos para todas as palavras de um bloco, eles são combinados geralmente através estilo OR, definindo uma assinatura de um bloco. (WIVES, 2002). Na figura 6 temos um exemplo de como a assinatura é mapeada em números de bits.

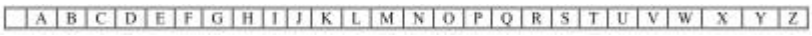
Figura 6 – Exemplo de uma assinatura.

Computer	0001	0110	0000	0110
Science	1001	0000	1110	0000
Graduate	1000	0101	0100	0010
Students	0000	0111	1000	0100
Study	0000	0110	0110	0100
Assinatura do bloco:	1001	0111	1110	0110

Fonte: Wives, 2002

O método árvore é uma estrutura criada para indexar palavras. Seu objetivo principal é armazenar palavras. Cada nodo desta estrutura é contido por um vetor de 27 componentes, que correspondem às letras do alfabeto, mais um componente em branco. (WIVES, 2002). Na figura 7, temos um exemplo de um nodo, onde contém um vetor de 27 componentes, correspondente às letras do alfabeto.

Figura 7 – Nodo de uma árvore.



Fonte: Wives, 2002.

Segundo Wives (2002), o conteúdo de cada nodo pode ser um conjunto de caracteres ou um número que aponta para o nodo seguinte.

Wives (2002) explica a estrutura de uma árvore da seguinte maneira:

> A estrutura é construída um nível por vez e cada nodo contém uma letra do termo sendo armazenado. O primeiro nodo, o nodo raiz, contém a primeira letra de todas as palavras indexadas pela estrutura. Essa técnica facilita a identificação de palavras inexistentes, já que com um único acesso é possível descobrir se determinada letra possui ou não conteúdo. Se determinada letra não possuir conteúdo é porque não existem palavras indexadas que iniciem com aquela letra. (WIVES, 2002 p. 60).

A figura 8 ilustra como é uma estrutura de uma árvore:

Figura 8 – Estrutura de uma árvore.

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		2	6				FUGA																				
2			ABACATE																		3						
3						4																					
4	ATÉ														ATENÇÃO												
5																											
6		BALEIA														BOLO											

Fonte: Wives, 2002.

Os nodos são adicionados quando existirem palavras com letras iguais. não é necessário que existam nodos para todas as letras de uma palavra. (WIVES, 2002).

2.2 - EXTRAÇÃO DE INFORMAÇÃO

Segundo Ceci (2010) a extração de informação faz parte da área de Processamento de Linguagem Natural (PLN), e tem como objetivo identificar informações importantes em base textuais. Por possuir muitos componentes de um sistema PLN e muitos desses componentes também são utilizados por sistemas de recuperação de informação para indexar documentos, a extração de informação torna-se um processo muito parecido com o processo de indexação de informações, mas há diferenças entre eles. (WIVES, 2002).

Wilks e Catizone (1999, apud CECI 2010, p. 48) comentam que:

> A extração e gerenciamento de informação sempre tiveram uma grande importância para as agências de inteligência, mas está claro que atualmente e nas próximas décadas essa é uma área crucial para a educação, a medicina e o comércio. É estimado que 80% das informações estão no formato textual, e por esse motivo essa é uma área tão importante.

A extração de informação também é classificada como modelo de um banco de dados para ser preenchido pelas instâncias extraídas. Sistemas de extração de informação podem ser usados para ajudar a popular ontologia. Os termos extraídos de documentos podem ser possíveis instâncias e classes das ontologias. (NÉDELLEC, NAZARENKO, 2005)

Uma informação extraída precisa ser anotada, e esta anotação possui intervenção humana, tornando-se um processo custoso. Uma abordagem mais simples é a anotação de

pequenos números de textos com a informação a ser extraída, e através de métodos de aprendizado, a coleção de documento rotulada é praticado sobre grandes coleções de documentos. (MOONEY; NAHM, 2005 apud GONÇALVES 2006)

2.2.1 – Reconhecimento de entidades nomeadas

Reconhecimento de entidades nomeadas é uma técnica da área de extração de informação, que tem a função de reconhecer entidades em textos de diferentes tipos e domínios. (CECI, 2010).

Segundo Kozareva (2006), uma vez que todas as entidades de um texto são detectadas, elas são passadas para a classificação de um conjunto pré-definido de categorias (pessoas, organizações, localização).

> Há muitas técnicas automáticas que podem ser utilizadas para realizar tal função, tais como: a aplicação de expressões regulares (muitas usadas para identificar datas, e-mails, URI, nomes seguidos de abreviações, etc.); o uso de dicionários (thesaurus); os modelos estatísticos; as heurísticas, regras conforme o padrão léxico e sintático do idioma; e também o uso de ontologias. (CECI et al, 2010, p. 05).

Kozareva (2006) explica que a tarefa de identificar as entidades também consiste em determinar o início e fim de uma entidade, isto é muito importante para ocasiões em que as entidades são compostas como, por exemplo, “Serviço orientado a arquitetura”, onde o inicio é a palavra serviço e o final é arquitetura.

2.2.1.2 – Resolução de ambiguidade

A resolução de ambuiguidade de palavras é uma tarefa do nível semântico, pois as ambiguidades muitas, vezes só podem ser resolvidas no contexto de um texto maior, como uma

frase ou um parágrafo onde a palavra está posicionada. Algumas vezes a ambiguidade só pode ser solucionada através de um conhecimento do mundo real. (FERNEDA, 2003).

Ferneda (2003) cita um exemplo para ambiguidade semântica, o verbo "passar" pode apresentar mais de um significado, como "passar a ferro", "passar no exame", "passar em casa" As causas da ambiguidade podem ser do tipo lexical, que ocorre quando uma mesma palavra pode possuir diversos significados, ou do tipo estrutural, quando é possível mais de uma estrutura sintática para a sentença, podendo ser: local quando a ambiguidade pode ser tratada dispensando o conhecimento do contexto onde ela ocorre; ou global quando é necessário o conhecimento e análise do contexto para a sua resolução. (BERDON, LUMSDEN, HOLMES 1991, apud FERNEDA 2003)

Ceci et al. (2010) comentam três problemas de ambiguidade:

1. erro de ambiguidade entre substantivos e entidades;
2. erro de detecção de limite de entidades;
3. erro de ambiguidade entre entidades.

O erro de ambiguidade entre substantivos e entidades ocorre quando ambos são homógrafos. Assim, o termo "Jobs", no idioma inglês pode representar o substantivo trabalho, ou até mesmo representar o sobrenome de uma pessoa em alguma entidade. Em um documento, esse problema pode ser resolvido, identificando quando a primeira letra for maiúscula (capitalizado), exceto quando há ocorrências no documento sem capitalização, ou somente aparece no inicio de uma frase ou entre aspas, ou quando aparecer dentro de sentenças nas quais todas as palavras com mais de três letras iniciam com maiúsculo. (CECI et al, 2010).

Quanto ao erro de detecção de limite de entidades, Ceci et al. (2010, p. 09) explicam que se refere "à estratégia de reconhecer onde uma entidade inicia e onde ela termina o texto. Isso acontece sempre com entidades que são formadas por duas ou mais palavras".

O terceiro problema está relacionado à ambiguidade entre entidades. Esse caso ocorre quando as mesmas entidades pertencem a mais de um tipo de classe. Para resolver o problema de ambiguidade entre entidades, foi criado um algoritmo específico. Quando for encontrada a ambiguidade, seus termos podem ser utilizados de duas formas. A primeira é verificar se a palavra que define uma entidade não é ambígua, podendo auxiliar na desambiguação. Por exemplo, "Oceano Atlântico" pode significar um lugar ou uma localização, porém se isolarmos "Atlântico" pode ter sentido de localização ou de uma organização. Se as palavras pertencem a um conjunto de termos que formam a entidade, então é possível afirmar que o conjunto inteiro é do tipo localização. E a segunda forma seria incluir

palavras próximas aos termos da entidade, com o objetivo de utilizá-las em contextos maiores para conseguir classificá-las. (NEDEAU, 2005, apud CECI et al. 2010).

2.2.1.3 – Utilização de sistemas *Named Entity Recognition* (NER)

Ceci (2010) mostra que a utilização de sistemas NER pode trazer uma série de benefícios para outros sistemas e áreas, como, por exemplo:

- Auxiliar no processo de recuperação de informação: o sistema NER identifica as entidades do texto antes do processo de indexação, fazendo com que seja indexada a entidade (que pode ser composta de vários termos) em vez de apenas os termos;
- Detecção de eventos: por meio das datas encontradas nos textos, pode-se fazer uma relação com os termos próximos e verificar a evolução;
- Manutenção em ontologias: através das entidades levantadas pelo sistema NER, pode-se verificar qual delas é uma possível classe da ontologia em questão e quais termos estão relacionados com a classe a fim de atualizar essa ontologia (GIULIANO, 2009, apud CECI 2010).

O autor ainda cita que uma grande vantagem de sistemas NER é que:

> Quando uma entidade é reconhecida, o sistema identifica uma possível classe para ela. O problema encontrado em utilizar listas de termos (ou tabelas léxicas) para classificação e reconhecimento das entidades é que isso torna o sistema sensível à linguagem das entidades levantadas, já que cada classe do sistema terá uma lista de termos relacionados nos idiomas previamente selecionados.

2.2.2 – Clusterização

Clusterização ou agrupamento tem sua principal atividade particionar ou dividir um conjunto de itens em grupos de itens com características semelhantes. (WIVES, 2003)

Segundo Wives (2003), o *clustering* é um "método de descoberta utilizado para identificar co-relacionamentos e associações entre objetos, facilitando assim a identificação de classes".

Usar a clusterização para auxiliar na área de recuperação de informação é muito importante, pois com esse processo é possível a visualização dos resultados das buscas em forma de conjunto relacionado pelo seu conteúdo. A clusterização em documentos pode proporcionar a descoberta não supervisionada de temas e principais tópicos da coleção de documentos. (ALJABER et al., 2009 apud CECI, 2010).

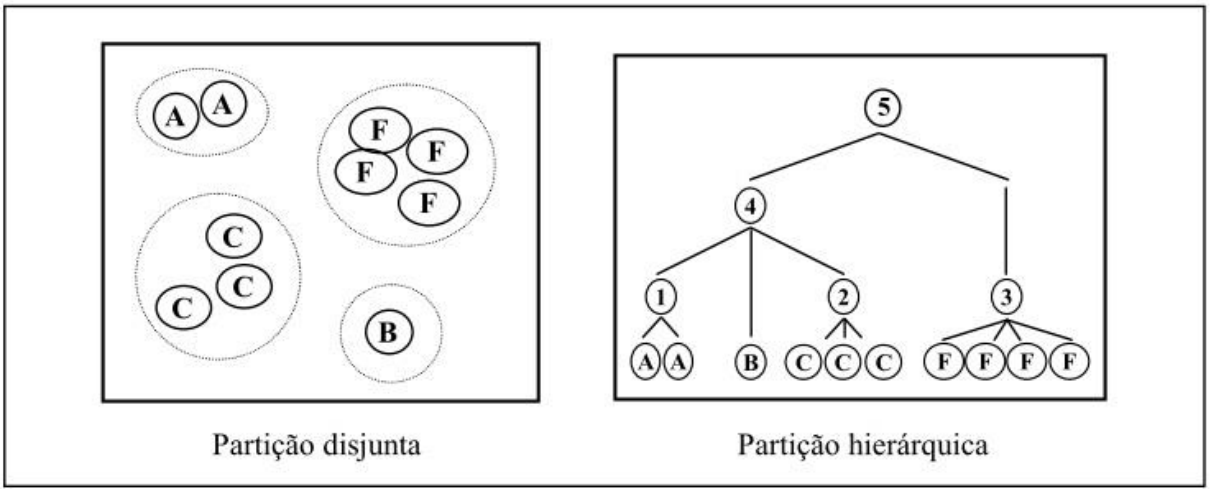
Wives (2003, p. 93) explica que:

> Essa técnica é geralmente utilizada antes de um processo de classificação, facilitando a definição de classes, pois o especialista pode analisar os co-relacionamentos entre os elementos de uma coleção de documentos e identificar a melhor distribuição de classes para os objetos em questão. Isso significa que não há a necessidade de se ter conhecimento prévio sobre os assuntos dos documentos ou do contexto dos documentos. Os assuntos e as classes dos documentos são descobertos automaticamente pelo processo de agrupamento.

O *clustering* pode ser representado por uma partição disjunta ou uma partição hierárquica. Na partição disjunta, um algoritmo de partição é aplicado à coleção de documentos e esses documentos são separados em grupos distintos, não havendo nenhum relacionamento entre os grupos identificados. Na partição hierárquica, o processo de identificação de grupos é aplicado recursivamente e acaba gerando uma espécie de árvore, em que as folhas representam os grupos mais específicos e os nodos principais representam grupos mais abrangentes. (WIVES, 2003).

A figura 9 mostra como é a estrutura da partição disjunta e hierárquica:

Figura 9 – Tipos de agrupamentos



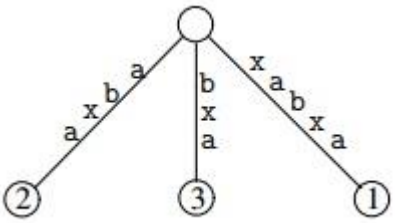
Fonte: Wives, 2003

Nesta seção foi apresentado o que é *clusterização* e também suas partições, disjunto ou hierárquico.

2.2.2.1 - Algoritmo STC

Elssen, Stein e Potthast (2005) explicam que o modelo, baseado em *Suffix tree Clustering* (STC), é um modelo completo em que se preserva a informação completa e não se considera um documento somente como um conjunto de palavras, mas, sim, uma sequência de palavras relacionadas entre si. A sequência das palavras é o modelo de documento para uma árvore de sufixo. Presenta na figura 10, um exemplo que mostra um árvore de sufixos de sequências s = xabxa.

Figura 10 – Representação de uma árvore de sufixos de sequência s = xabxa



Fonte: ALMEIDA, MARTINEZ, TELESS, 2005

Segundo Almeida, Martinez, Teless (2005), a busca de uma sequência em um texto, usando algoritmo STC, é a busca de todas as ocorrências de uma sequência *p* de tamanho *m* em um texto *s* de tamanho *n*. Com a ideia de que qualquer sufixo *s[i..m]* de *s* deve estar representado na árvore de sufixos através de um único caminho até a raiz T_s até a folha rotulada com *i*.

> Uma vez determinado que *p* ocorre em *s*, nas posições determinadas pelos rótulos das folhas abaixo do (ou exatamente no) vértice onde a busca terminou, basta executarmos um algoritmo qualquer de percurso na sub-árvore enraizada por esse vértice, coletando os rótulos das folhas. Assim, pode-se determinar as ocorrências de p em s em tempo $O(M + K)$ onde *k* é o número de ocorrências. (ALMEIDA, MARTINEZ, TELESS, 2005, p. 20).

Ceci (2010) leciona que o algoritmo de STC é basicamente divido em 4 partes:

- limpeza dos documentos, onde é feita a retirada os espaços entre palavras mais que um e aplicar o *stemming;*
- identificar frases comuns;
- calcular um peso para cada frase levando em consideração a sua importância para o documento.

Nesta ultima etapa é feito, o *merging* entre os clusters.

O algoritmo de *clustering*, citado por Zamir e Etzioni, é visto como uma heurística para avaliar com maior nível de precisão (baseado em gráficos) em coleções de documentos grandes. (ELSSEN, STEIN, POTTHAST, 2005) .

2.3 – ANOTAÇÃO DE DOCUMENTOS

Segundo Araújo (2003), o processo de anotação de documentos serve para que esses documentos possam ser compreendidos pelas máquinas, de modo que a recuperação de informação possa ser incrementada, tendo como base um modelo ontológico. Para isso, é necessário que a anotação esteja de acordo com os metadados que descrevem os matériais de aprendizagem e que estão definidos na ontologia de domínio. Araújo (2003) também afirma que este processo pode ser demorado e não produtivo.

Eller (2008) cita em seu trabalho que as anotações podem ser representadas de duas maneiras, intrusivas e não intrusivas. Serão classificadas intrusivas quando as anotações forem guardadas dentro do próprio documento, e não intrusivas quando as informações forem guardadas em repositórios de anotações que apontam para os documentos que passaram pelo processo de anotação.

Eller (2008) demonstra em seu trabalho que a anotação de documento pode ser automática onde é recomendado quando há um grande volume de documentos (esta sujeita a falhas), semiautomática quando não é possível que o processo de anotação seja totalmente automática (requer iteração com o usuário no processo de anotação), anotação manual, que necessita total iteração com o usuário durante o processo de anotação, este método de anotação não é uma boa alternativa quando se tem grandes volumes de documentos, pois sempre surgem novos documentos e novos termos, outro ponto negativo para o processo manual é que exige grande nível de conhecimento do domínio e da ontologia utilizada para efetuar as anotações.

Por último, o processo de anotação semântica, que será abordado na seção seguinte.

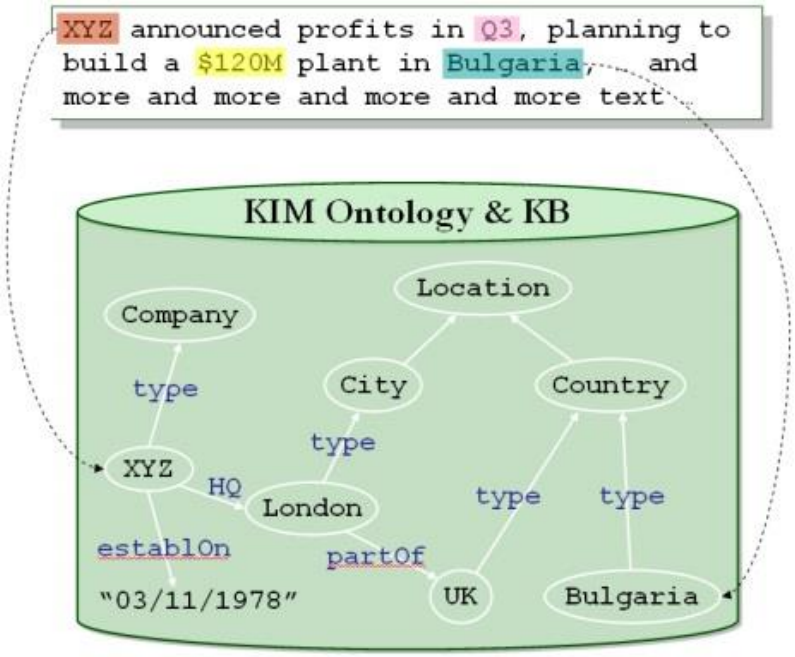
2.3.4 – Anotação semântica

Segundo Popov e outros (2003), a anotação semântica tem a ideia de atribuir as entidades links de textos para sua descrição semântica.

> A anotação semântica de documentos possui como objetivo facilitar a busca dos documentos no repositório digital. Com ela, é possível correlacionar termos (conceitos, instâncias ou propriedades) da ontologia a palavras, simples ou compostos, do texto que passou pelo processo de anotação semântica. Ela atribui às palavras que aparecem no documento ligações com suas descrições semânticas na ontologia. (ELLER, 2008 p. 83).

Na figura 11, temos o exemplo de um texto que foi ligado a uma base de conhecimento para poder fazer a anotação semântica.

Figura 11 – Anotação Semântica



Fonte: Popov et al. 2003

Para Eller (2008), anotação semântica de documentos pode ser manual ou automática. Por um lado, a anotação manual busca sobre a representação da anotação, compartilhamento e armazenamento. Por outro lado, a investigação sobre ferramentas de anotação foca em formar de criar anotações, de acordo com determinados domínios de ontologias.

Popov e outros (2008) explicam que a estrutura de uma anotação semântica deve possuir uma ontologia (ou taxonomia) que carrega as classes de entidades, identificadores únicos que serão capazes de ligar com sua descrição semântica e uma base de conhecimento.

Um problema citado por Reeve e Han (2005), com a anotação semântica que não foi totalmente resolvido, é a sua dificuldade de automatizar esse processo. Pelo fato de se encontrar essa barreira na anotação semântica automática, os sistemas atuais se concentram no modelo semiautomático, que não tem seu processo totalmente automatizado, necessitando interversão humana.

2.4 – BUSCA SEMÂNTICA

Segundo Guha, McCool e Miller (2003), a busca semântica tenta aumentar e melhorar os resultados de buscas tradicionais. Os autores também explicam que a adição de semântica a pesquisa pode melhorar significativamente o resultado de uma recuperação de informação.

Guha, McCool e miller (2003) mostram em seu trabalho que o objetivo da busca semântica é permitir que o motor de busca compreenda que diferentes ocorrências com as mesmas palavras podem ter significados distintos. Além de compreender, o motor de busca será capaz de filtrar e classificar os resultados e retornar documentos referentes à denotação correta.

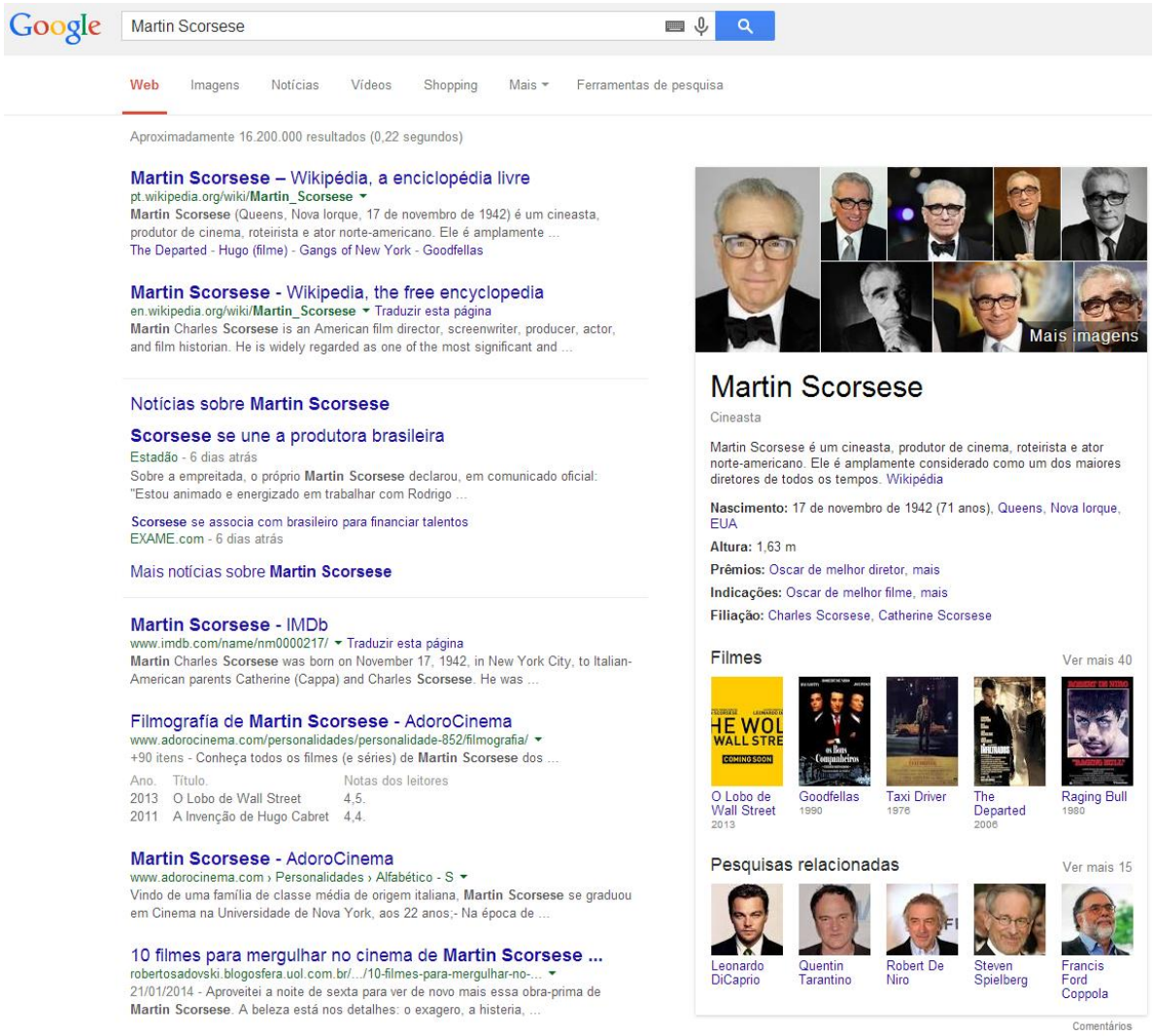
Para Ceci, Woszezenki, Gonçalves (2014, p.5) "o uso de ontologias possibilita definir conceitos e suas relações representando o conhecimento sobre o documento em termos específicos de um domínio." Jesus (2009) define ontologia como um vocabulário estruturado que contém as relações entre termos, que visa à melhoria da descrição feita por usuários não especialistas.

Um dos benefícios da anotação semântica é a desambiguação de palavras, afirmam Ceci, Woszezenki, Gonçalves (2014). Bräscher (2002) explica que a ambiguidade é uma palavra ou forma que pode possuir vários significados distintos, podendo ser compreendido de maneiras diferentes pelo sistema de recuperação de informação.

Segundo Bräscher (2002) a ambiguidade de palavras pode confundir o sistema de recuperação de informação, pois sob um termo, o usuário que realizou a pesquisa pode encontrar informações relevantes e irrelevantes. A palavra "Jaguar" é um exemplo de ambiguidade, podendo confundir o sistema de recuperação de informação. O termo pode significar animal felino, a fábrica e marca de automóveis, ou o cartunista Jaguar.

Para aprimorar o resultado das consultas, o Google (2014), aperfeiçoou seu sistema de busca para tentar compreender o significado das palavras, oferecendo para o usuário o conteúdo que ele realmente precisa, trazendo a ferramenta *knowledge map*. O mapa do conhecimento (*knowledge map*) envolve uma coleta de informações e faz ligações entre eles. Podendo ligar atores a filmes, diretores, data lançamento e até outros filmes. A figura 47 exemplifica como o mapa do conhecimento funciona.

Figura 12 – Pesquisa no Google com o termo “Martin Scorsese”.



Fonte: Autor, 2014

Pesquisando pelo nome do cineasta Martin Scorsese, podemos visualizar na figura 47, que na lateral da imagem possui um espaço onde vai ser ocupado por dados que foram localizados com as ligações do mapa do conhecimento. O espaço foi preenchido com uma breve explicação sobre o cineasta, imagens, nascimento, filmes, e atores relacionados ao cineasta.

2.5 – CONSIDERAÇÕES FINAIS

Neste capitulo foi abordado conceitos com base referencial para o desenvolvimento do trabalho. Destacando o conceito de recuperação de informação e os modelos usados no processo de RI, a importância do processo de indexação, a forma de como as anotações facilitam o acesso à recuperação de documentos em buscas e a conceitualização do processo de extração de informação.

3 – MÉTODO

Neste capitulo, será abordado o tipo de pesquisa proposto neste trabalho, a definição da lista de etapas que servirá como base para a conclusão da pesquisa. Apresenta-se também a arquitetura da solução proposta, com o esquema da solução e também as delimitações da pesquisa.

3.1 – CARACTERIZAÇÃO DO TIPO DE PESQUISA

Gil (1996) leciona que uma pesquisa bibliográfica é desenvolvida através de materiais já elaborados, podendo ser livros, e artigos científicos.

De acordo com Jung (2003), o que se chama de método cientifico consiste em um "conjunto de técnicas e processos utilizados pela ciência para formular e resolver problemas de aquisição objetiva do conhecimento de maneira sistemática".

Este trabalho se enquadra em uma pesquisa aplicada e qualitativa. Segundo Cervo (2002, p.65) "na pesquisa aplicada, o investigador é movido pela necessidade de contribuir para fins práticos mais ou menos imediatos, buscando soluções para problemas concretos".

O trabalho também foi classificado como uma pesquisa qualitativa, que, segundo Silva e Menezes (2005 p. 20), a pesquisa aplicada tem uma relação entre o mundo real e o sujeito, ou seja, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito

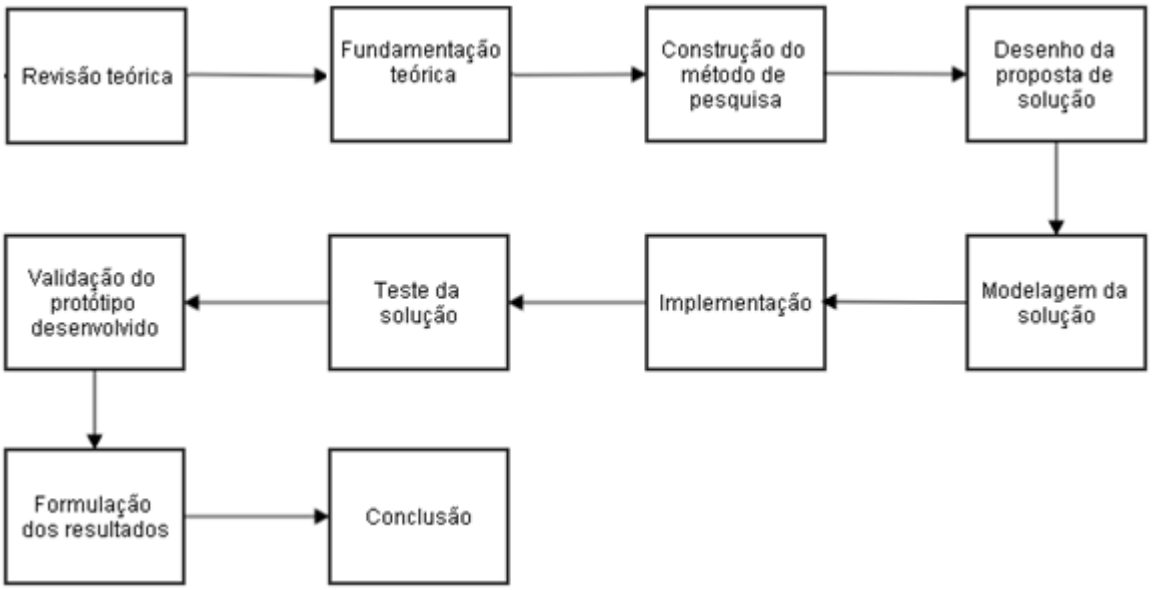
que não pode ser traduzidos em números. O mesmo afirma que a pesquisa qualitativa não requer o uso de métodos e técnicas estatísticas, o ambiente em si é a fonte para coletas de dados e o pesquisador é o instrumento chave.

Por fim, o modelo proposto vai ser avaliado através de um estudo de caso. Gil (1996) explica que o estudo de caso consiste em um "estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento".

3.2 – ETAPAS

Na figura 13, esta presente as etapas para concluir os objetivos propostos nesse trabalho.

Figura 13 – Etapas do trabalho.



Fonte: Autor, 2013

Este trabalho possui 10 etapas para concluir os objetivos propostos.

- Fundamentação teórica: Etapa que compreende a pesquisa bibliográfica realizada com a finalidade de criar o conhecimento necessário para entender o problema.

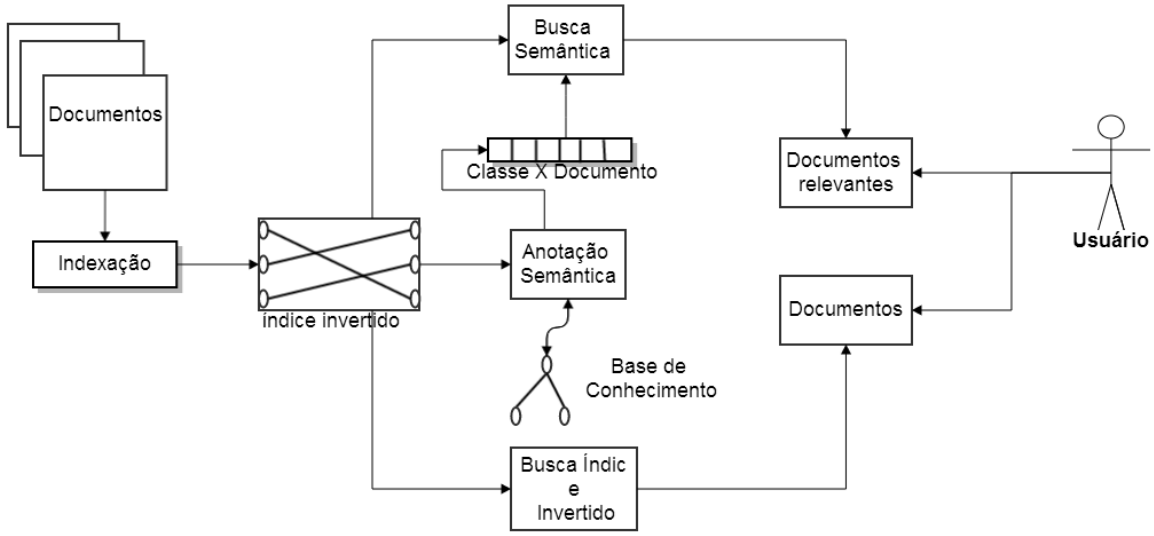
- Revisão teórica: Nessa etapa, pesquisaram-se os principais autores sobre o assunto, que serviu de base para escrever a fundamentação teórica.
- Construção do método de pesquisa: Etapa em que vão ser descritos as características e quais métodos de pesquisa vão ser usados.
- Desenho da proposta da solução: Nessa etapa vai ser desenvolvido um esboço da proposta da solução e de como vai funcionar.
- Modelagem: Nessa etapa, serão desenvolvidos a modelagem baseado nos diagramas UML (Casos de Uso, Diagrama de Sequência) da solução proposta, que serão utilizados para o entendimento e a implementação da proposta.
- Implementação: Etapa que consiste em implementar a solução, seguindo a modelagem e o conhecimento obtido.
- Teste da solução: Etapa em que vai ser validada a modelagem e implementação feita e analisar se os objetivos foram alcançados.
- Validação do protótipo desenvolvido: Essa etapa vai validar, através de um estudo de caso, o protótipo desenvolvido para a solução.
- Formulação e resultados: Etapa em que vão ser analisados e apresentados os resultados obtidos pelo trabalho.
- Conclusão: Na conclusão, serão comparados os objetivos propostos neste trabalho e se esses objetivos foram alcançados com a solução proposta.

Na próxima seção, é proposto um esquema de arquitetura, para solucionar a proposta e seus objetivos.

3.3 – ARQUITETURA DA SOLUÇÃO PROPOSTA

Na figura 14 é apresentado a arquitetura da solução para atingir os objetivos da solução proposta.

Figura 14 – Arquitetura da solução proposta



Fonte: Autor

Na arquitetura da solução, foi proposto o seguinte:

Documentos passarão pelo processo de indexação, utilizando índice invertido. Depois de ser indexado, o documento vai passar pela anotação semântica, utilizando a base de conhecimento extraindo informação do documento para conseguir gerar um índice conceito/documento. A busca semântica buscará documentos relevantes dentro do índice classe x documento, trazendo os resultados mais relevantes para o usuário.

3.4 – DELIMITAÇÕES

Este trabalho não foca em nenhuma área especifica do processo de recuperação de informação.

Não tem como objetivo criar um método novo para indexação de documentos e nem para anotação de documentos. Também não se tem como foco construir um mecanismo que permita inferência nos relacionamentos ontológicos.

A solução proposta não caberá para qualquer assunto, é necessário que a base de conhecimento dê suporte a área de aplicação em questão.

4 – PROJETO DE SOLUÇÃO

Nesta seção, serão apresentadas definições de técnicas e metodologia, conceitos de UML (*Unified Modelling Language*) e Orientação a Objeto (OO), o modelo Iconix e seu processo usado para o desenvolvimento, e o estudo de caso para validar a proposta da solução.

4.1 – DEFINIÇÃO DE TÉCNICA E METODOLOGIA

Fachin (2001, p.29) afirma em seu livro que "métodos e técnicas se relacionam, mas são distintos".

> O método é um plano de ação, formado por um conjunto de etapas ordenadamente dispostas, destinadas a realizar e antecipar uma atividade na busca de uma realidade. (FACHIN, 2001 p.29).

Já a técnica é o "modo de fazer de forma mais hábil, mais seguro, mais perfeito, algum tipo de atividade, arte ou ofício". (GALLIANO, 1986, p.6).

4.1.1 – Iconix

Segundo Guimarães e outros (2007), o ICONIX é um processo de desenvolvimento em que se aplica uma metodologia prática e simples. Não é um processo tão burocrático comparando ao RUP (*Rational Unified Process*), em que há uma grande quantidade de documentação, mas não é tão simples como o XP (*eXtreme Programing*).

Maia (2005) define que um processo de desenvolvimento de *software* "são etapas cuidadosamente planejadas em que o sucesso de cada etapa é primordial para produtos com qualidade, baixo custo e rapidez na construção, ou seja, pode possibilitar bom resultado final no produto de *software*".

Maia (2005) explica que o ICONIX utiliza também uma linguagem robusta para modelagem, chamada UML (*Unified Modeling Language*).

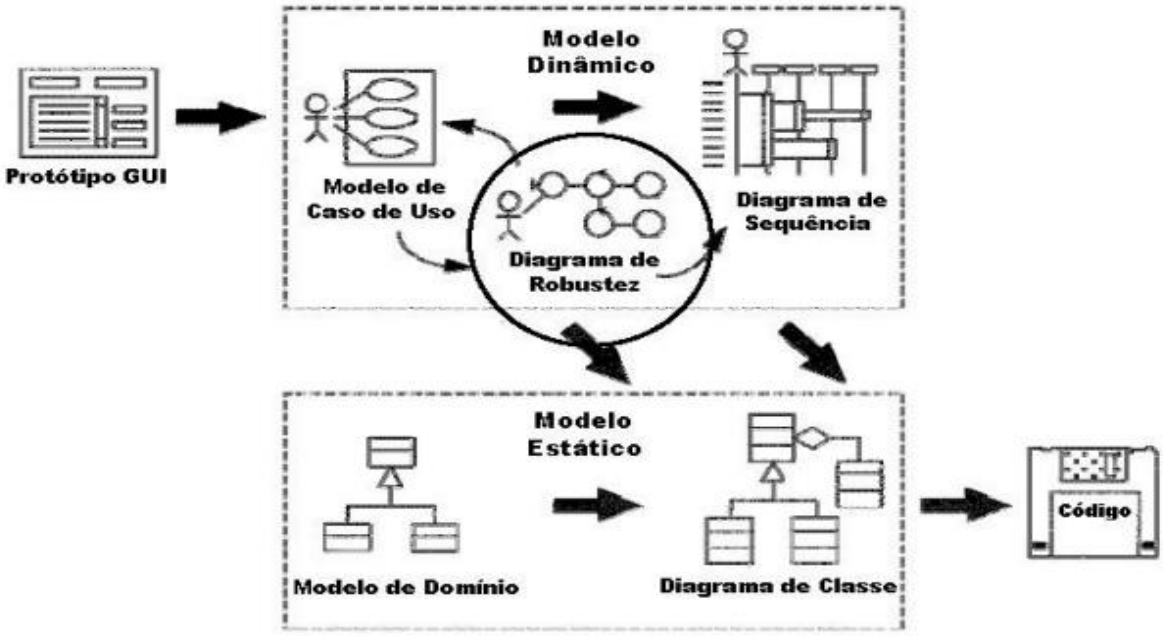
O processo ICONIX também possui uma característica exclusiva, chamada "Rastreabilidade de Requisitos", fazendo com que, através dos seus mecanismos, confirme em todas as fases do processo, se os requisitos estão sendo atendidos. (MAIA, 2005).

> A proposta do ICONIX permite um alto grau de Rastreabilidade. Em cada fase ao longo do caminho é necessário rever os requisitos em algum momento. Não existe um ponto em que o processo permite distanciar dos requisitos do usuário. Rastreabilidade também significa encontrar novos objetos em cada fase durante o projeto. (MAIA, 2005 p.3)

O ICONIX é composto por cinco principais fases (modelo de domínio, modelo de caso de uso, análise robusta, diagrama de sequência, diagrama de classe) e por dois grandes setores, que podem ser desenvolvidos em paralelo e de modo recursivo (ver Figura 14), sendo eles o modelo estático e o modelo dinâmico. (MAIA, 2005).

A figura 15 demostra as cinco principais fases e os dois grades setores, modelo estático e o modelo dinâmico.

Figura 15 – Visão geral do ICONIX



Fonte: Adaptação de Rosenberg (2005)

A seguir, as cinco principais fases para seguir o processo de desenvolvimento de software do modelo ICONIX :

4.1.1.1 – Modelo de domínio

Para Maia (2005), o modelo de domínio é a parte essencial do processo ICONIX, construindo uma parte estática inicial de um modelo que será usado para dirigir a fase de design a partir dos casos de uso. O modelo de domínio consiste basicamente em descobrir objetos de um problema do mundo real.

4.1.1.2 – Modelo de caso de uso

O modelo de caso de uso é usado para representar as funcionalidades que o usuário exija. Na descrição do caso de uso, deve-se detalhar, de forma clara e legível, todos os cenários que os usuários executarão para realizar a tarefa. (MAIA, 2005).

4.1.1.3 – Diagrama de robustez

Já, o modelo de análise robusta tem o objetivo de conectar a parte de análise com a de projeto, garantindo que as descrições dos casos de uso estão corretas, também poderá descobrir novos objetos através do fluxo de ação. (MAIA, 2005).

4.1.1.4 – Diagrama de sequência

O objetivo do diagrama de sequência é construir um modelo dinâmico entre o usuário e o sistema. Para fazer esse modelo dinâmico, devemos utilizar os objetos e suas interações identificadas na análise robusta, também será necessário descrever o detalhamento de cada fluxo de ação. (MAIA, 2005).

4.1.1.5 – Diagrama de classe

Por último, o diagrama de classe que é o amadurecimento da ideia do modelo de domínio, que foi adquirido ao longo das fases do ICONIX, representando as funcionalidades do sistemas de modo estático, sem a interação do usuário com o sistema. (MAIA, 2005).

4.1.2 – *Unified modeling language* (UML)

"O UML (*Unified Modelling Language*) é uma linguagem diagramática, utilizável para especificação, visualização e documentação de sistemas de software." (SILVA; VIDEIRA, 2001).

Folwer (2005) complementa que o UML ajuda no projeto de sistemas de software construídos, utilizando o paradigma orientado a objetos (OO).

Bell (2003) mostra que os diagramas mais utilizados do UML são: diagrama de casos de uso, diagrama de classe, diagrama de sequência, diagrama de estados, diagrama de atividade, diagrama de componentes e diagrama de implantação.

- **Diagrama de casos de uso:** O principal objetivo é descrever requerimentos funcionais do sistema. (FURLAN, 1998).
- **Diagrama de classe:** É utilizado para visualizar os aspectos estáticos, os seus relacionamentos e detalhes da construção. (BOOCH et al, 2005).
- **Diagrama de sequência**: Segundo Booch (2005) serve para descrever o fluxo principal e os fluxos excepcionais de um caso de uso.
- **Diagrama de estados:** São diagramas que representam o comportamento interno de uma classe. Isso não significa que toda classe tem que ter um diagrama de estado, apenas as classes que possuem três ou mais estados potenciais durante a atividade do sistema. (BELL, 2003).
- **Diagrama de atividades:** São diagramas que mostram o fluxo processual de controle entre dois ou mais objetos da classe durante o processamento de uma atividade.
- **Diagrama de componentes:** Tem a função de mostrar as partes internas, os conectores e as portas que implementam um componente. (BOOCH et al, 2005).
- **Diagrama de implantação:** Mostra como o sistema será implantado fisicamente no ambiente de hardware. (BELL, 2003).

4.1.3 – Orientação a objeto (OO)

Segundo Kamienski (1996), a orientação a objetos é uma metodologia de programação que promove a modularidade do sistema e o reaproveitamento do código fonte. A visualização do programa em execução é dada como uma coleção de objetos se comunicando e onde cada um dos objetos é uma instância de uma classe se unindo, utilizando herança.

Fedeli (2002) afirma que: "a condição de orientação a objetos desencoraja o desenvolvedor a pensar em uma aplicação da forma hierárquica (ou seja, de cima para baixo, funcionalmente decomposta), mas incentiva a pensar em componentes de forma plana e reutilizável".

Kamienski (1996) e Meyer (1996) citam as principais vantagens de utilizar o paradigma orientado a objetos:

- reaproveitamento do código;
- escalibidade de aplicações;
- manutenção;
- apropriação;
- abstração de dados;
- compatibilidade.

Apesar das diversas vantagens, a orientação a objetos também possui algumas desvantagens. Kamienski (1996) e Meyer (1996) explicam que o trabalho em equipe, no momento do desenvolvimento pode gerar problemas e a dificuldade para dominar o paradigma.

4.2 – MODELAGEM DO SISTEMA PROPOSTO

Nesta seção do trabalho, é apresentada a modelagem do sistema proposto, onde inicialmente vão ser apresentados os atores envolvidos na utilização do sistema. Após a ilustração dos atores, vão ser listados os requisitos funcionais e não funcionais do sistema.

Com base na metodologia de desenvolvimento ICONIX, após a identificação dos atores e da listagem de requisitos do sistema, serão desenvolvidos todos os modelos e diagramas descritos neste processo, sendo eles: modelo de casos de uso, prototipação de telas, diagrama de sequência, diagrama de robustez e o modelo de domínio que será evoluído durante o processo para virar um diagrama de classe.

Para cada modelo de caso de uso descrito, é criado respectivamente seu diagrama de sequência e robustez, que servirão para a construção do diagrama de classe.

4.2.1 – Atores

1. Administrador – Este perfil tem a permissão de inserir e excluir documentos da base de dados e atribuir o perfil administrador a outros usuários.

2. Usuário não logado – Este perfil tem somente a permissão de pesquisar por documentos no sistema. O usuário não precisa estar logado no sistema.

4.2.2 – Requisitos

Para Sommerville (2007, p. 79), “os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema”.

4.2.2.1 – Requisitos Funcionais

Os requisitos funcionais “são as declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações”. (SOMMERVILLE, 2007, p. 80).

- RF001 – Somente o usuário administrador poderá inserir um novo documento no sistema.
- RF002 – O sistema deve permitir que o usuário logado com o perfil Administrador cadastre novos usuários com o perfil Administrador.
- RF003 – O sistema deve permitir que o usuário administrador exclua administradores.
- RF004 – O sistema deve permitir que outro usuário administrador visualize os documentos inseridos por um administrador.
- RF005 – O sistema deve permitir que todos os usuários que acessarem o sistema consiga realizar buscas por conteúdos cadastrados por administradores.
- RF006 – As buscas por conteúdo devem apresentar como resultado arquivos no formado pdf.
- RF007 – As buscas por conteúdo devem apresentar como resultado para o usuário resultados textos referente à consulta efetuada.
- RF008 – Não é preciso estar logado no sistema para realizar buscas.
- RF009 – O sistema deve ter controle de quantos downloads foram feitos de um determinado arquivo.

Nesta seção do trabalho foram apresentados os requisitos funcionais que o sistema proposto deve atender.

4.2.2.2 – Requisitos não-funcionais

Requisitos não funcionais “são restrições sobre os serviços ou as funções oferecidas pelo sistema. Eles incluem restrições de timing, restrições sobre o processo de desenvolvimento e padrões”. (SOMMERVILLE, 2007, p. 80).

- RNF001 – Para qualquer consulta que o usuário realizar, o tempo de resposta não poderá passar de 2 segundos.
- RNF002 – O sistema deve garantir que o retorno de registro de uma busca, deverá apresentar somente documentos relevantes à aquela consulta.
- RNF003 – Em qualquer busca feita pelo usuário, se não retornar nenhum registro, deverá aparecer uma mensagem informando.
- RNF004 – Obrigatoriedade do uso do banco de dados PostgreSQL.
- RNF005 – Banco de dados deve suportar mais de dez mil registros em uma tabela.
- RNF006 – Antes de um usuário administrador excluir qualquer registro do sistema, deverá ser apresentada uma mensagem de confirmação.
- RNF007 – O sistema deve permitir que os usuários selecionem qualquer arquivo retornado de uma pesquisa e possam visualizar e executá-los.
- RNF008 – O sistema deverá permitir documentos do tipo pdf, doc e docx.
- RNF009 – A listagem de arquivos deve ser feita por ordem de relevância aos termos pesquisados pelo usuário.

Nesta seção do trabalho foram apresentados os requisitos não-funcionais que o sistema proposto deve atender.

4.2.2.3 – Regras de negócio

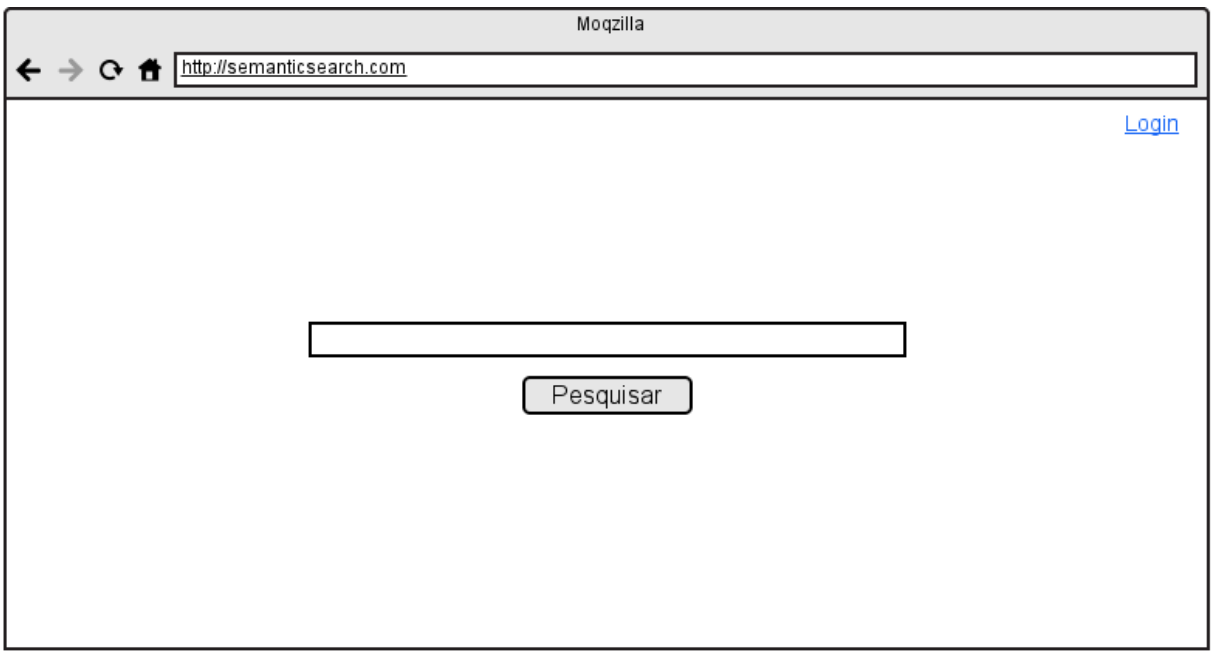
Nesta seção, encontram-se as regras que devem ser implementadas na solução proposta.

- RN001 – Poderá existir apenas uma única conta de administrador por CPF.
- RN002 – Somente usuários logados com o perfil administrador pode alterar ou excluir qualquer informação do sistema.
- RN003 – Somente usuários logados com o perfil administrador pode inserir documentos no sistema.
- RN003 – Somente usuários logados com o perfil administrador pode cadastrar novos usuários.
- RN004 – Login para acessar o sistema deverá ser feito através do CPF.

4.2.3 – Protótipos de tela

Nesta seção, são apresentados todos os protótipos de tela do sistema, e, com essas imagens, é possível visualizar onde cada funcionalidade do sistema ocorre.

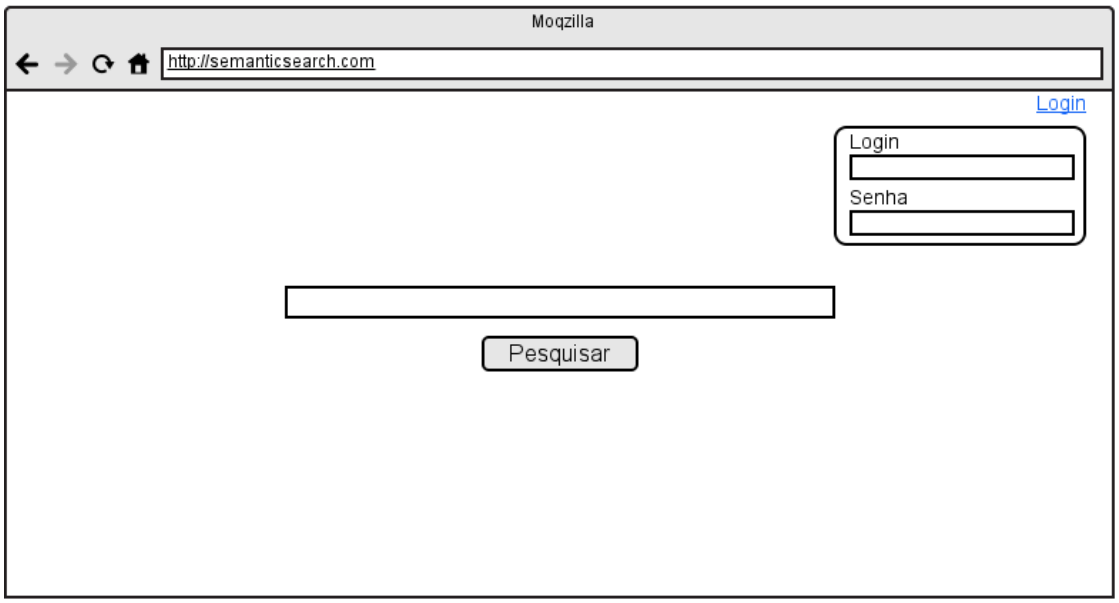
Figura 16 – Tela inicial pública



Fonte: Autor, 2013

Na tela inicial pública (figura 16), o usuário irá informar o termo para realizar a busca. Esta tela será apresentada tanto para o usuário comum que entra no sistema sem estar logado, quanto para o usuário administrador logado. No canto superior esquerdo, tem a opção para logar no sistema, a figura 17 mostra esta opção.

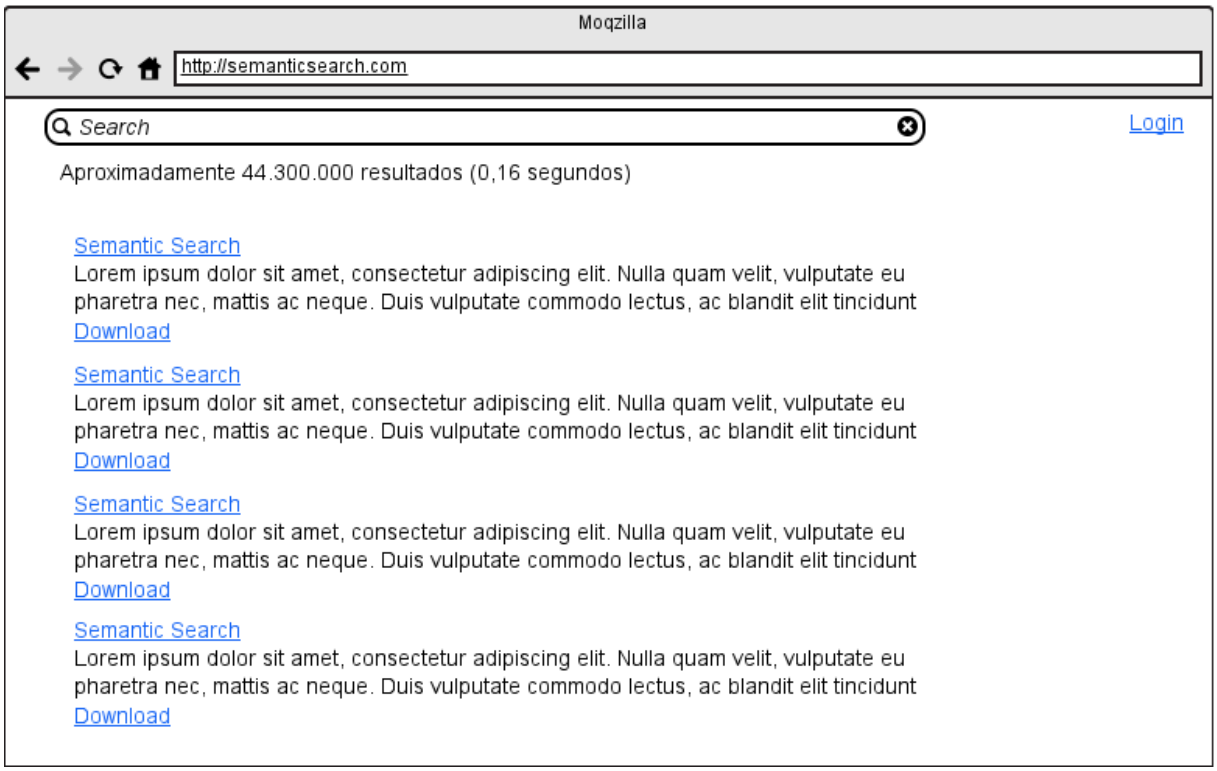
Figura 17 – Login de acesso ao sistema



Fonte: Autor, 2013

A página de listagem do conteúdo pesquisado (figura 18) ilustra como as informações devem ser apresentadas.

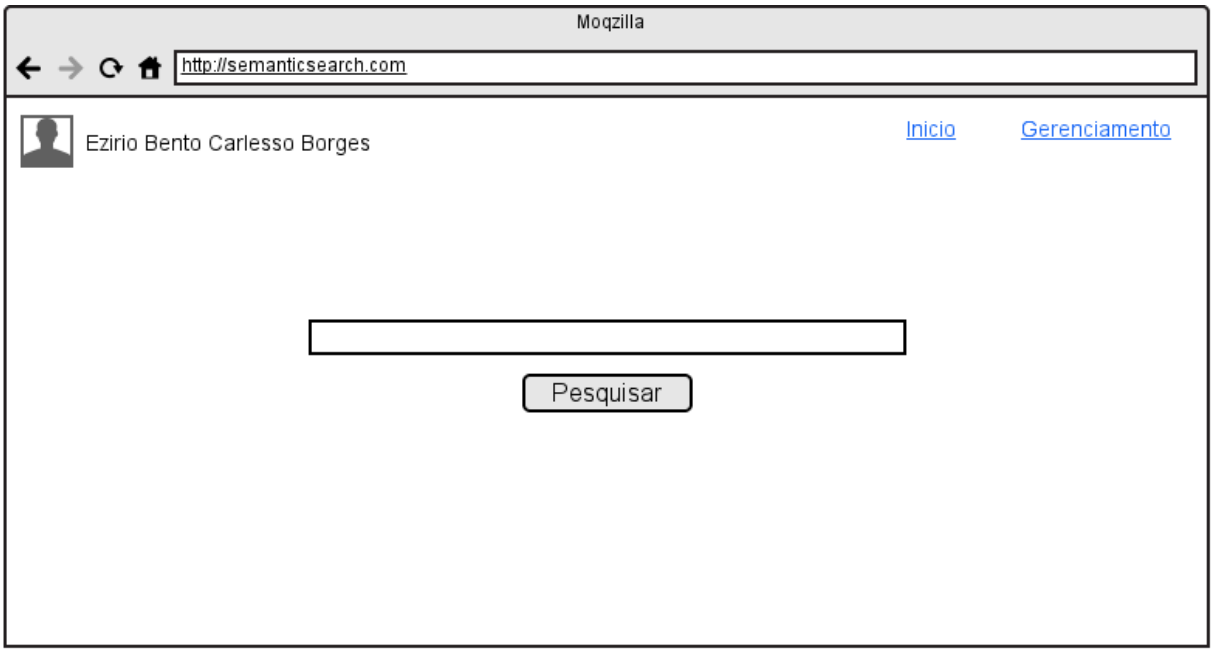
Figura 18 – Pagina de listagem



Fonte: Autor, 2013

A página inicial privada é apresentada somente depois do usuário fazer login no sistema, e enquanto o usuário estiver logado, como administrador será apresentada a página inicial privada. A figura 19 ilustra essa situação.

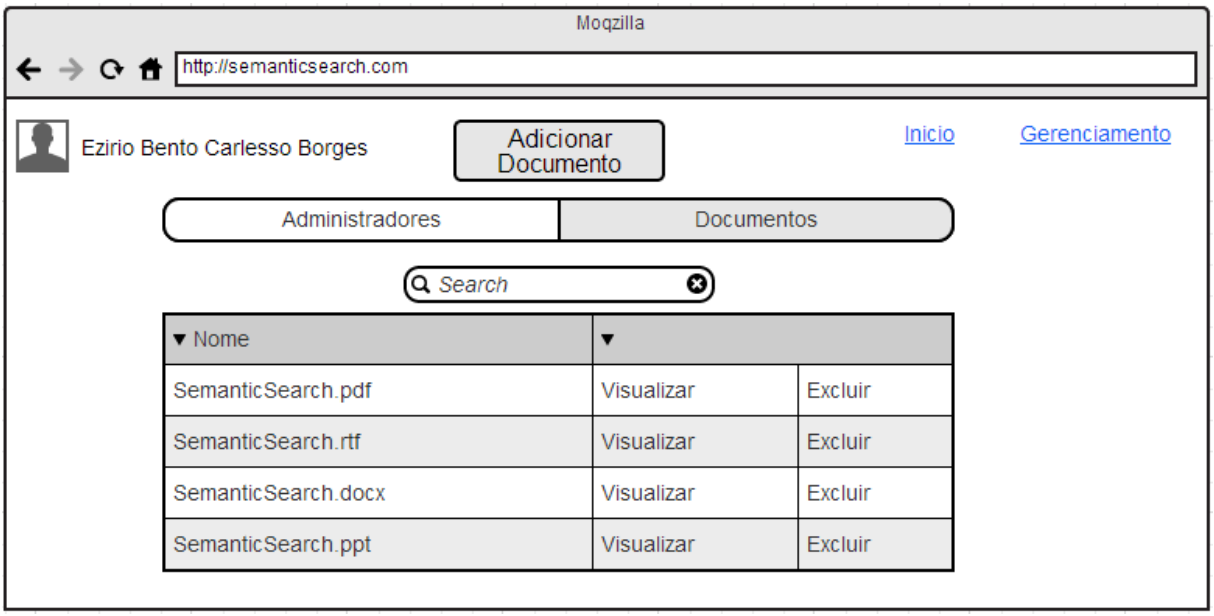
Figura 19 – Página inicial privada



Fonte: Autor, 2013.

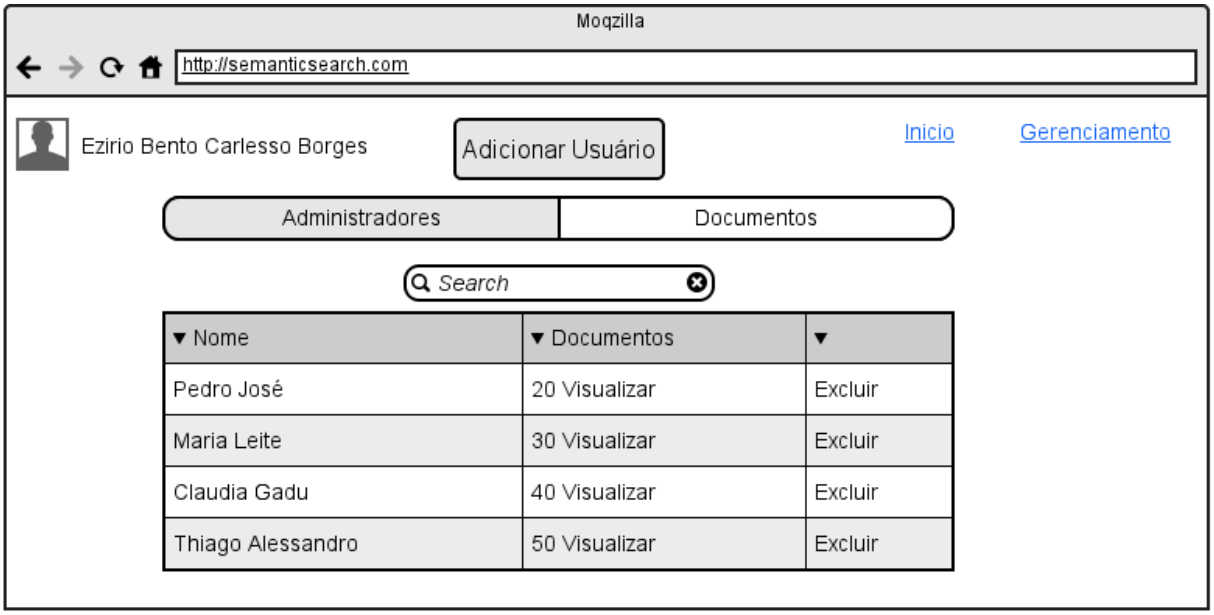
A figura 20 e 21 demonstra as funcionalidades no sistema relativo aos usuários administradores e aos documentos inseridos na plataforma. As duas situações são parecidas, tendo as funcionalidades de visualizar, excluir e editar seus dados.

Figura 20 – Página privada gerenciamento (documentos)



Fonte: Autor, 2013.

Figura 21 – Página privada gerenciamento (administradores)



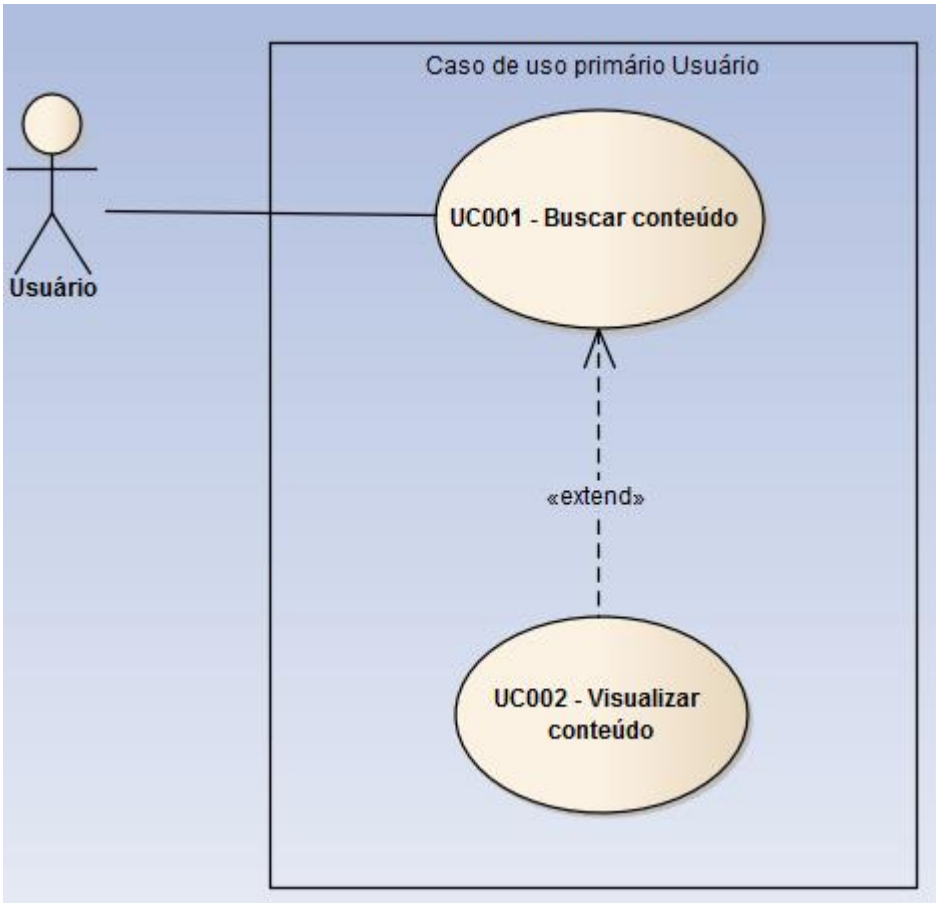
Fonte: Autor, 2013

Na seção a seguir, são detalhados os casos de uso identificados na modelagem da solução proposta. Os casos de uso apresentados são separados por ator.

4.2.4 – Casos de Uso

Nesta seção do trabalho, são apresentados os casos de uso primário do sistema, mostrando seus possíveis fluxos principais (fluxo base) e alternativos para cada ator identificado. Na figura 22, são apresentados os casos de uso para o perfil Usuário.

Figura 22 – Casos de uso para o perfil usuário.



Fonte: Autor, 2013

UC001 – Buscar conteúdo (Fluxo principal)

1. Usuário informa o termo a ser consultado na tela de busca (TL001).
2. O sistema retorna o conteúdo e mostra na tela de listagem (TL002).
3. Usuário escolhe qual conteúdo ele quer visualizar.

UC001 – Buscar conteúdo (Fluxo alternativo)

1. Usuário informa o termo a ser consultado na tela de busca (TL001).
2. O sistema não encontrou nenhum resultado com o termo informado.
3. Mostrar no lugar da listagem de conteúdo a seguinte mensagem de validação: “O sistema não encontrou nenhum resultado para esse termo.”.

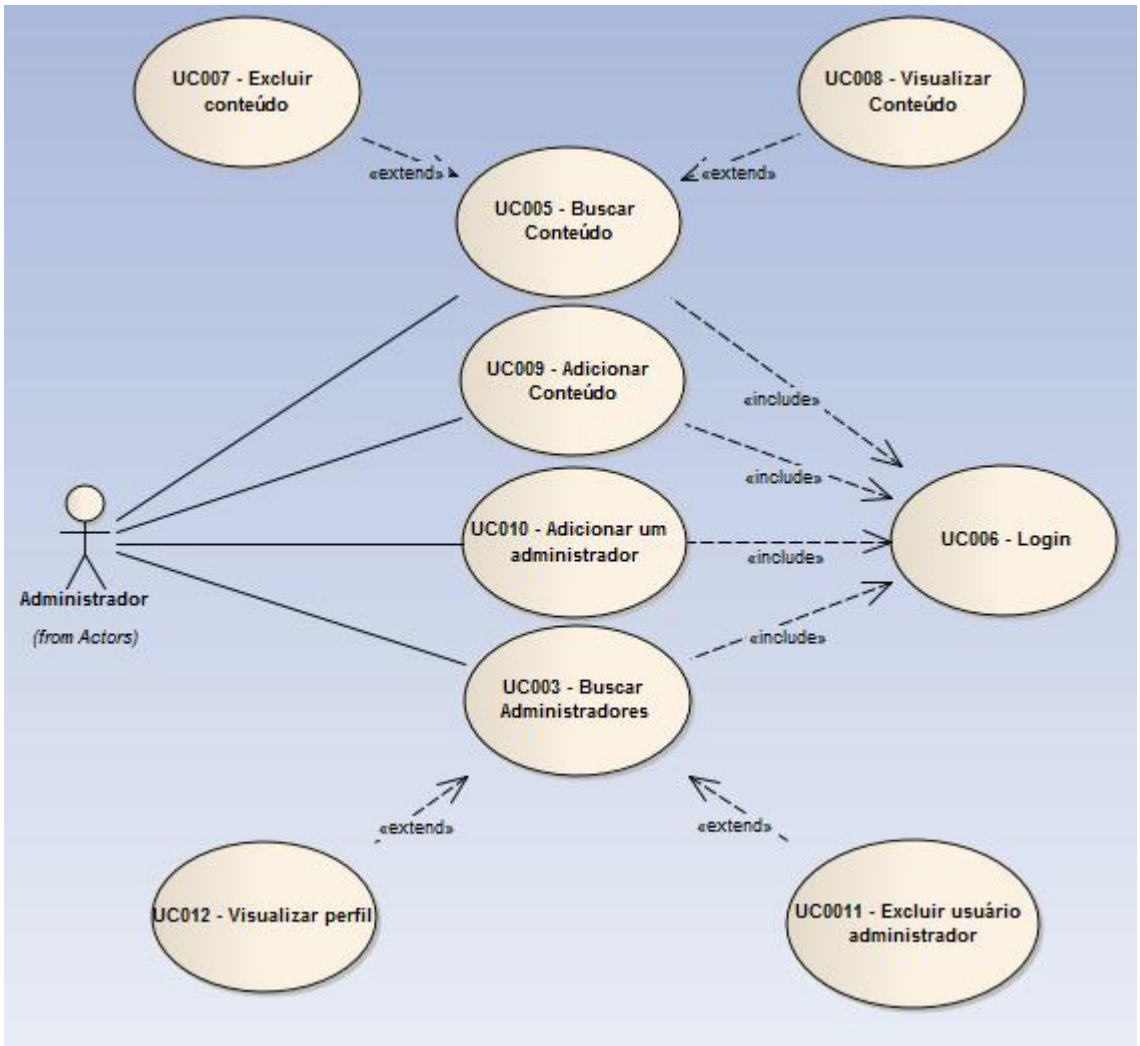
UC002 – Visualizar conteúdo (Fluxo principal)

1. Usuário informa o termo a ser consultado na tela de busca (TL001).

2. O sistema retorna o conteúdo e mostra na tela de listagem (TL002).
3. Usuário escolhe qual conteúdo ele quer visualizar.
4. Usuário escolhe em qual formato ele vai querer visualizar, de acordo com o que tem disponível no sistema.

Para o perfil Adminitrador foi identificado nove casos de uso (figura 23), na sequencia dessa seção, são apresentados e detalhados todos os casos de uso que pertencem ao perfil Administrador.

Figura 23 – Casos de uso para o perfil administrador.



Fonte: Autor, 2013

UC005 – Buscar conteúdo (Fluxo principal)

1. Administrador informa o termo a ser consultado na tela de busca.
2. O sistema retorna o conteúdo e mostra na tela de listagem.

3. Usuário escolhe qual conteúdo ele quer visualizar.

UC005 – Buscar conteúdo (Fluxo alternativo)

1. Administrador informa o termo a ser consultado na tela de busca.
2. O sistema não encontrou nenhum resultado com o termo informado.
3. Mostrar no lugar da listagem de conteúdo a seguinte mensagem de erro: “O sistema não encontrou nenhum resultado para esse termo.”.

UC007 – Excluir conteúdo (Fluxo principal)

1. Administrador informa o termo a ser consultado na tela de busca.
2. O sistema retorna o conteúdo e mostra na tela de listagem.
3. Administrador seleciona a opção de “excluir’.
4. O sistema apresenta uma mensagem de confirmação de exclusão, “Você deseja apagar o conteúdo selecionado?”
5. Sistema volta para a listagem de documentos.

UC008 – Visualizar conteúdo (Fluxo principal)

1. O Administrador informa o termo a ser consultado na tela de busca.
2. O sistema retorna o conteúdo e mostra na tela de listagem.
3. O Administrador escolhe a opção de visualizar o documento selecionado.
4. Abre o documento e seleciona em outra janela.

UC009 – Adicionar conteúdo (Fluxo principal)

1. O Administrador seleciona a opção “Adicionar documento”.
2. Seleciona o documento a ser inserido na base de dados.
3. O documento passa pelo processo de anotação semântica.
4. Volta para a tela de listagem de documentos.

UC010– Adicionar administrador (Fluxo principal)

1. O Administrador seleciona a opção “Adicionar administrador”.
2. Digita o CPF do usuário a ser inserido.
3. Valida se o usuário já existe.

4. Preenche o resto do conteúdo obrigatório para registro.
5. Sistema mostra mensagem de confirmação do cadastro.
6. Volta para a tela de listagem de usuários administradores.
7. O documento passa pelo processo de anotação semântica.
8. Volta para a tela de listagem de documentos.

UC010– Adicionar administrador (Fluxo alternativo)

1. O Administrador seleciona a opção “Adicionar administrador”.
2. Digita o CPF do usuário a ser inserido.
3. Valida, se o usuário já existe.
4. CPF informado já foi cadastrado.
5. Mostra a mensagem, “CPF digitado já possui cadastro”.
6. Volta para a tela de listagem de usuários administradores.

UC003– Buscar administrador (Fluxo principal)

1. Administrador informa o nome a ser consultado na tela de busca
2. O sistema retorna o conteúdo e mostra na tela de listagem

UC003– Buscar administrador (Fluxo alternativo)

1. Administrador informa o nome a ser consultado na tela de busca
2. O sistema não encontrou nenhum resultado com o n informado.
3. Mostrar no lugar da listagem de conteúdo a seguinte mensagem de erro: “O sistema não encontrou nenhum resultado para esse termo.”.

UC012– Visualizar perfil administrador (Fluxo principal)

1. Administrador informa o nome a ser consultado na tela de busca.
2. O sistema retorna o conteúdo e mostra na tela de listagem.
3. Administrador seleciona opção “Visualizar”.
4. Abre uma tela com todas as informações referentes ao usuário selecionado.

UC007 – Excluir usuário administrador (Fluxo principal)

1. Administrador informa o nome a ser consultado na tela de busca.
2. O sistema retorna o conteúdo e mostra na tela de listagem.
3. Administrador seleciona a opção de “excluir”.
4. O sistema apresenta uma mensagem de confirmação de exclusão, “Você deseja apagar o usuário selecionado?”.
5. Sistema volta para a listagem de usuários administrador.

UC006 – Login (Fluxo principal)

1. Usuário não logado seleciona a opção “Login”
2. O usuário informa o CPF e a senha.
3. Sistema valida, se o formato do CPF esta correto.
4. Verificar se o CPF e a senha estão corretos.
5. Mostra a tela privada do Administrador.

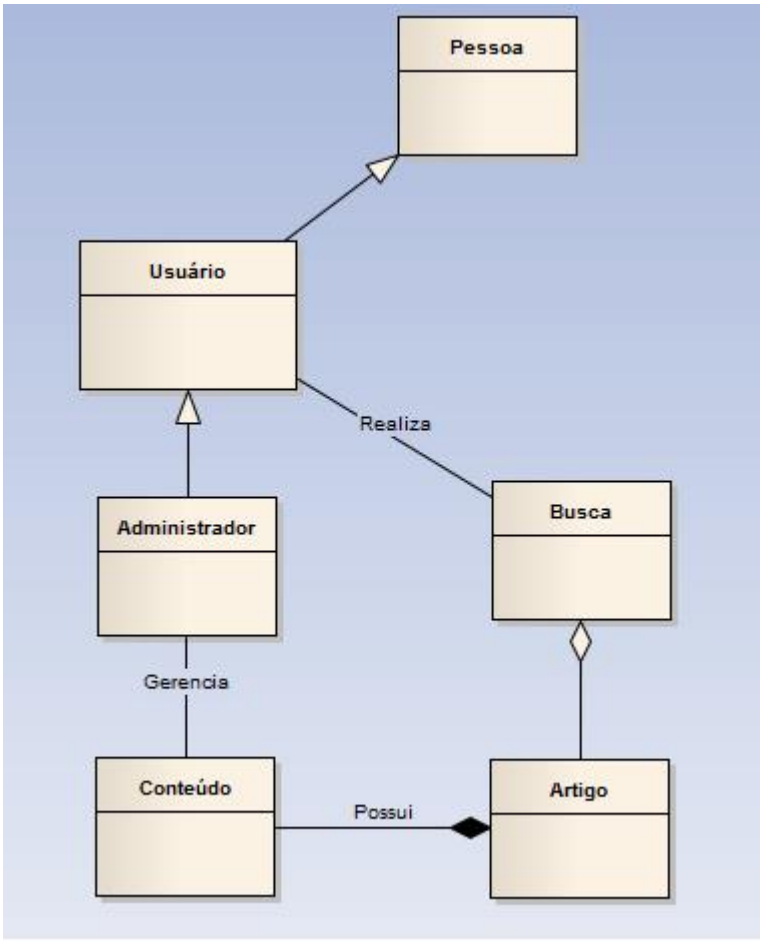
UC006 – Login (Fluxo de exceção)

1. Usuário não logado seleciona a opção “Login”
2. O usuário informa o CPF e a senha.
3. Sistema valida, se o formato do CPF esta correto.
4. Verificar se o CPF e a senha estão corretos.
5. Mostra mensagem de erro, “CPF ou senha inválidos”.
6. Retorna para a página principal pública.

4.2.5 – Modelo de domínio

Nesta seção encontra-se o modelo de domínio (figura 24) em que é possível visualizar os dados de uma maneira geral.

Figura 24 – Diagrama de domínio.



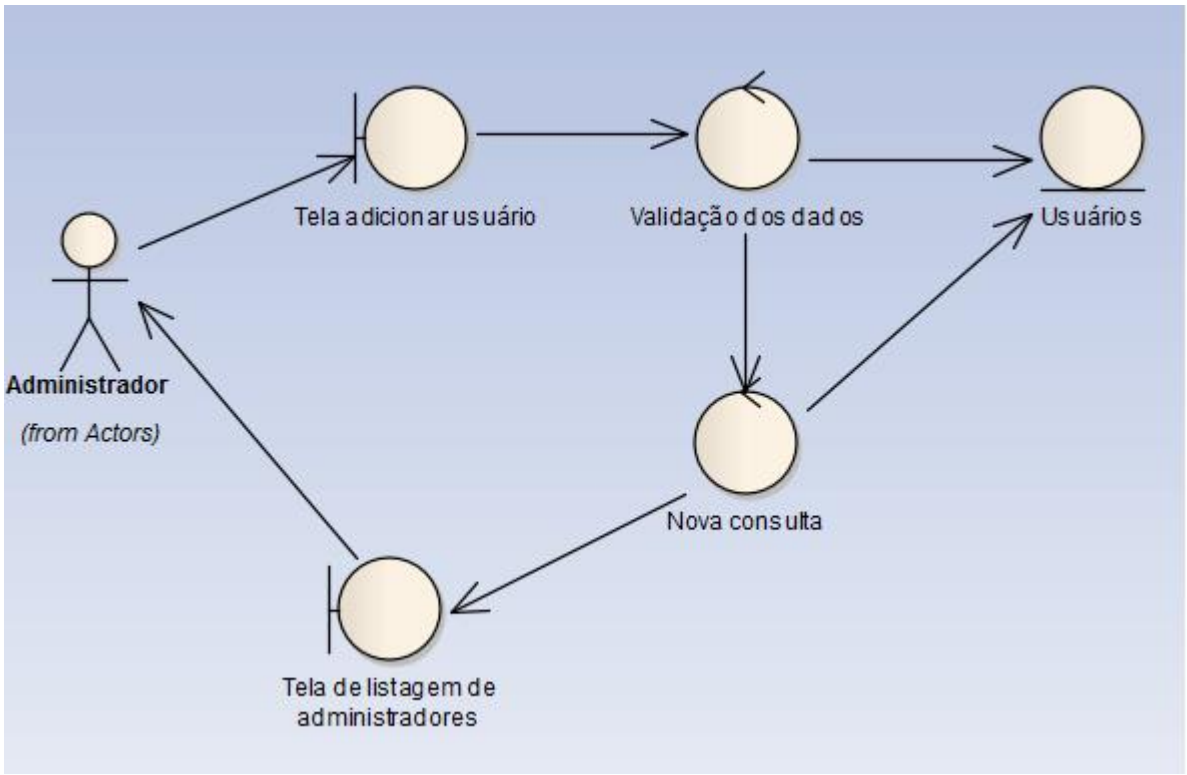
Fonte: Autor, 2013

4.2.6 – Diagrama de robustez

Estão presentes, nesta seção, os diagramas de robustez identificados para Administrador e Usuário.

Na operação de cadastro de um usuário administrador (figura 25), o Administrador visualiza a tela onde são adicionadas as informações do novo usuário administrador. Após preencher todos os dados obrigatórios o sistema faz uma validação de todas as informações preenchidas na tela. Somente depois da validação, o usuário é inserido no banco de dados. Depois do novo usuário for inserido no banco, é feito uma nova consulta para visualizar todos os usuários do sistema.

Figura 25– Diagrama de robustez: Cadastro de um administrador.

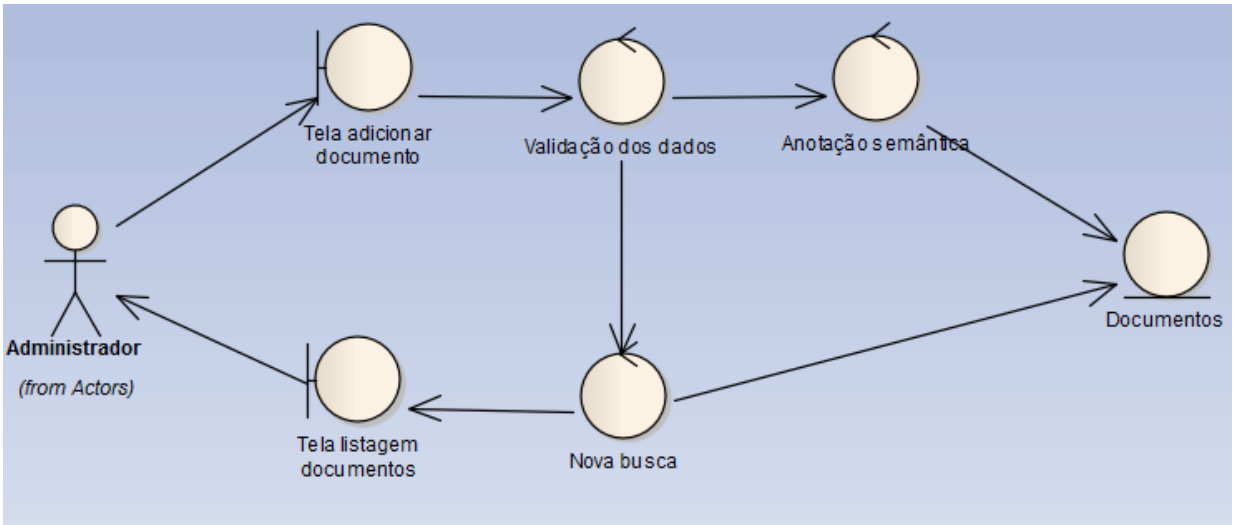


Fonte: Autor, 2013

A figura 26 mostra como é feita a inserção de documentos. O administrador abre a tela de adição de documentos e seleciona o documento a ser inserido e é feita uma validação dos dados inseridos e, após a validação, o documento passa pelo processo de anotação

semântica para ser inserido no banco de dados. Depois que o documento foi inserido na base de dados, realiza-se uma nova consulta para mostrar uma nova listagem de documentos para apresentar ao administrador.

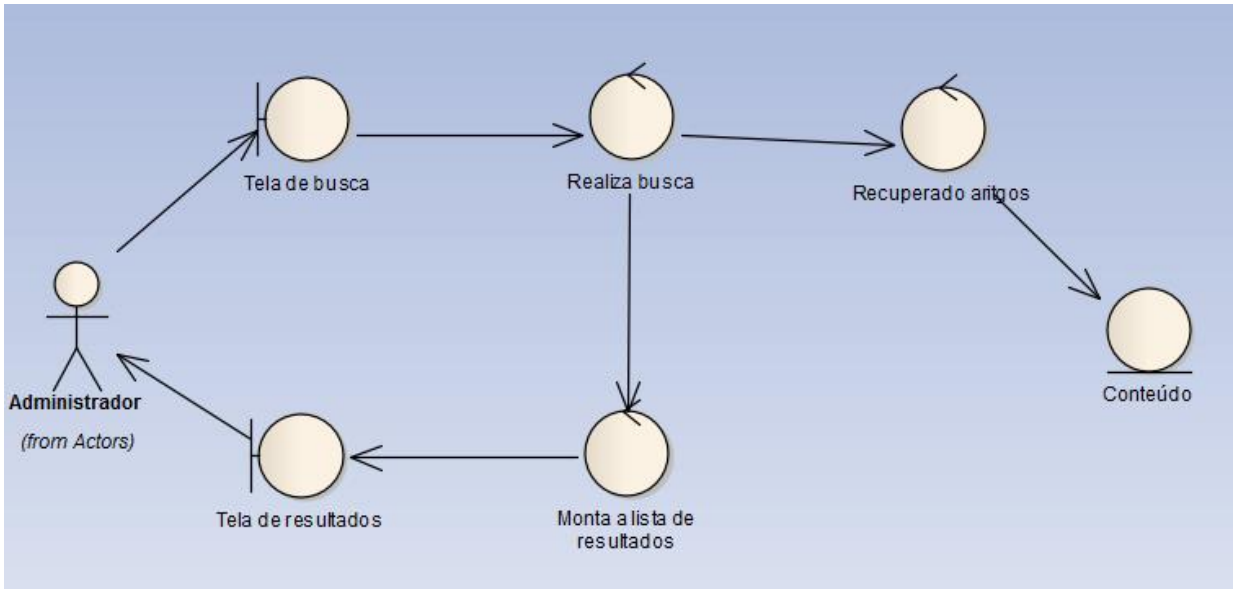
Figura 26 – Diagrama de robustez: Cadastro de um documento.



Fonte: Autor, 2013

A figura 27 demonstra a busca de um documento, que começa com a tela de busca onde é informado o termo a ser pesquisado. Com o termo informado, é realizada a busca nos artigos relevantes para o tema. Após a busca, monta-se a lista com os documentos retornados e mostra-se uma lista de documentos.

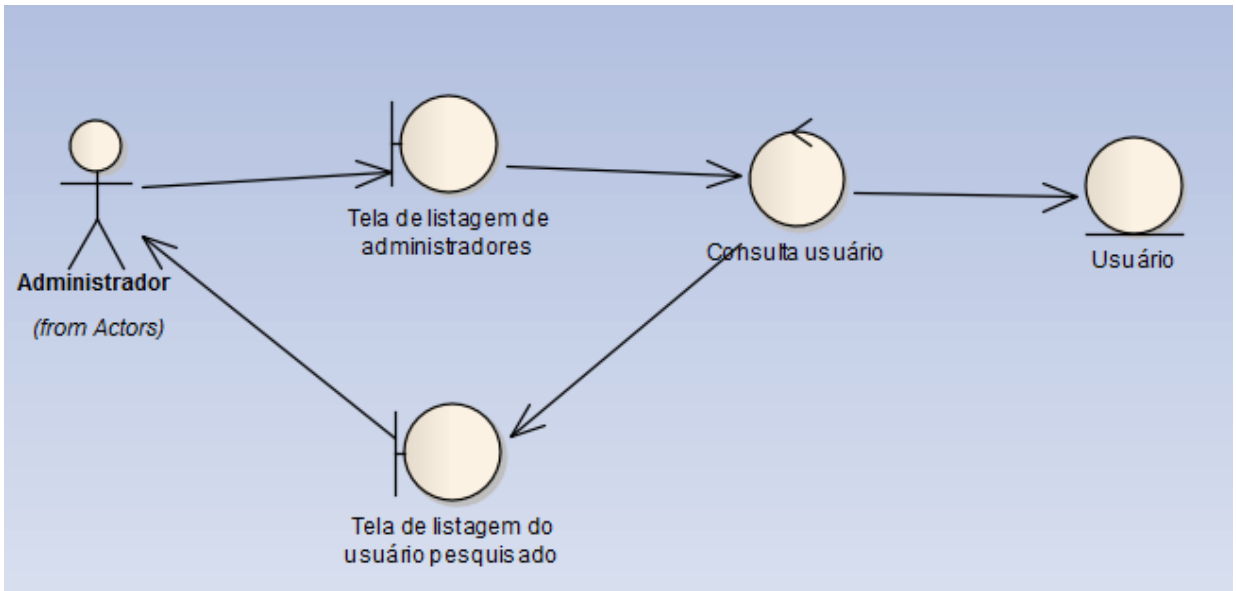
Figura 27 – Diagrama de robustez: Busca de um documento.



Fonte: Autor, 2013.

Para a busca de um administrador, o usuário informa o nome do administrador a ser procurado. É realizada uma busca no banco de dados e retorna para o usuário uma nova lista de usuários com o termo informado. A figura 28 representa este fluxo.

Figura 28 – Diagrama de robustez: Busca de um administrador.

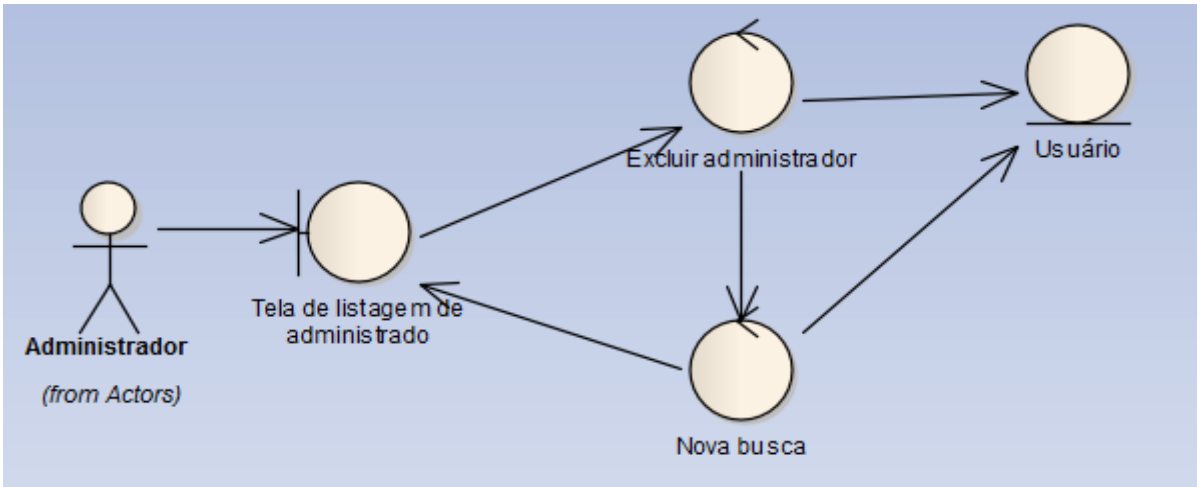


Fonte: Autor, 2013.

Para excluir um administrador, o usuário vai acessar à tela de listagem, seleciona o administrador a ser excluído e, após a confirmação da exclusão, o administrador selecionado é

excluído do banco de dados, em seguida, é feito uma nova consulta que atualiza a listagem de usuários administradores. O ciclo deste fluxo é representado na figura 29.

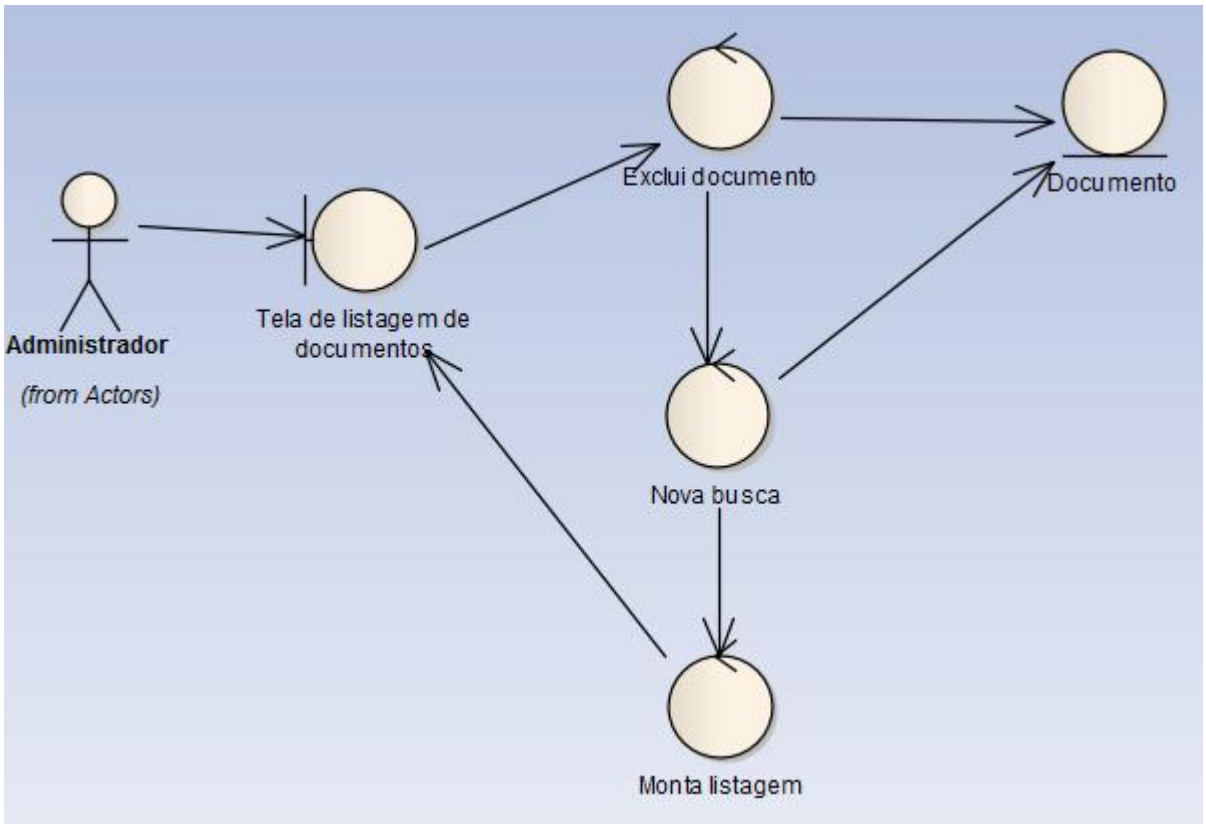
Figura 29– Diagrama de robustez: Exclusão de um administrador.



Fonte: Autor, 2013

Semelhante ao diagrama de exclusão de administrador, a exclusão de um documento passa pelo mesmo processo: o usuário vai acessar à tela de listagem de documentos, vai selecionar o documento a ser excluído e confirmar a sua exclusão, logo após a confirmação o documento é excluído do banco de dados. Feito a exclusão no banco, é realizado uma nova consulta, e a listagem é mostrada para o administrador. A figura 30, a seguir, mostra como é feito o fluxo.

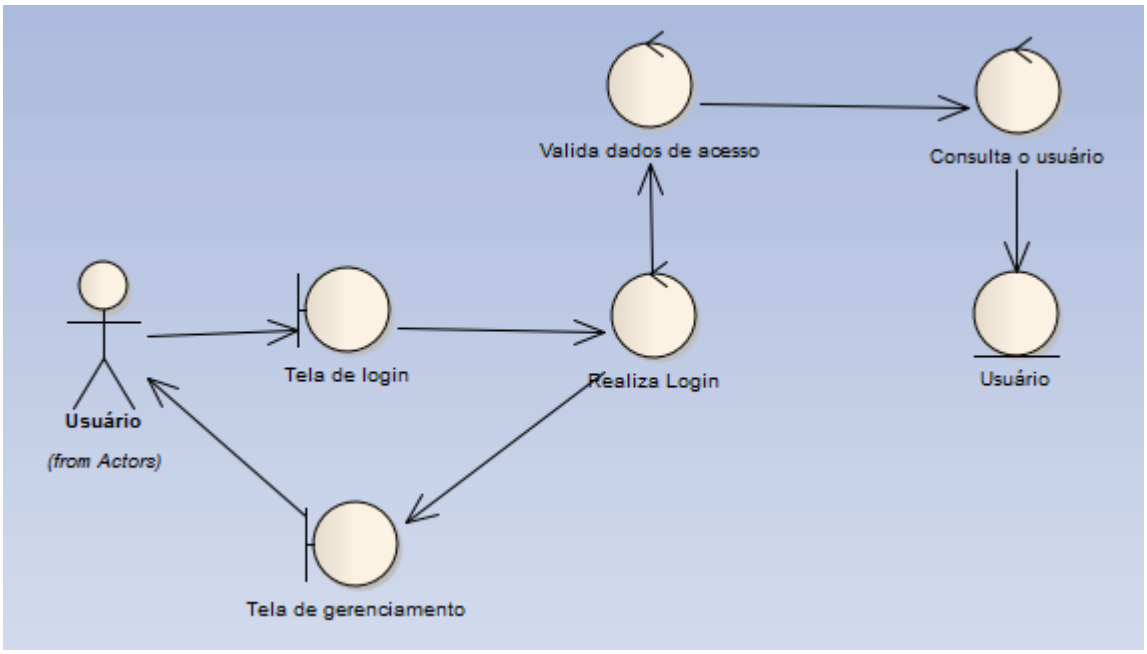
Figura 30– Diagrama de robustez: Exclusão de um documento.



Fonte: Autor, 2013

Na operação de login (Figura 31), o usuário não logado informa o login e a senha. O sistema faz uma validação dos dados inseridos. O sistema busca o usuário no banco de dados e faz a verificação para, posteriormente, realizar a autenticação do usuário e o redirecionamento para a tela de gerenciamento.

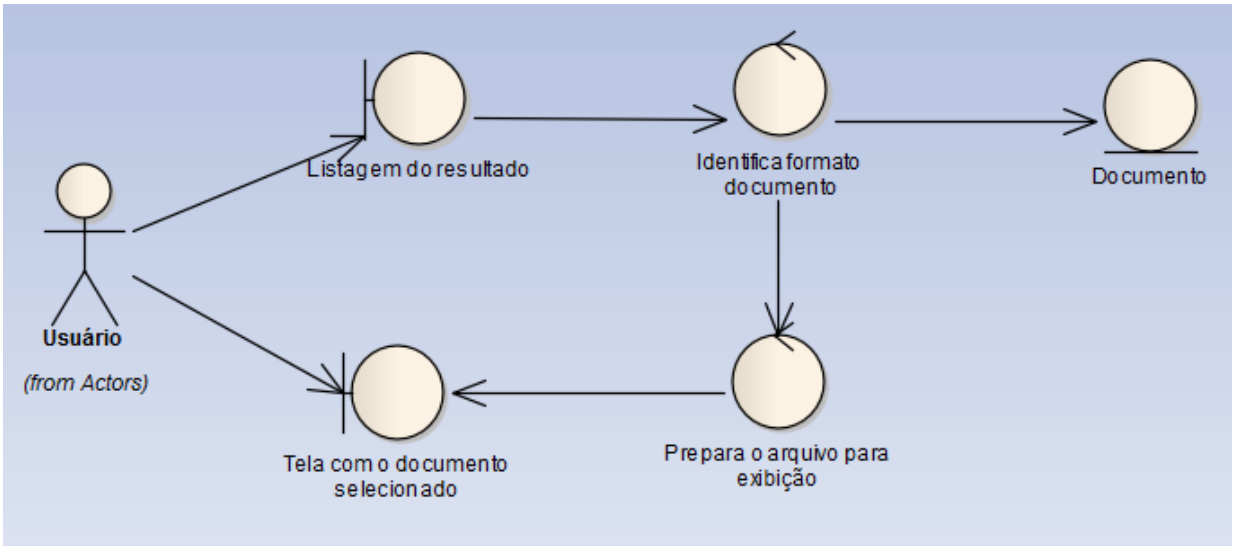
Figura 31 – Diagrama de robustez: Login.



Fonte: Autor, 2013.

O usuário seleciona o documento a ser visualizado, o sistema identifica qual é formato do arquivo. Prepara a visualização, de acordo com o formato, e mostra em uma tela separada da listagem para o usuário. A figura 32 ilustra o fluxo.

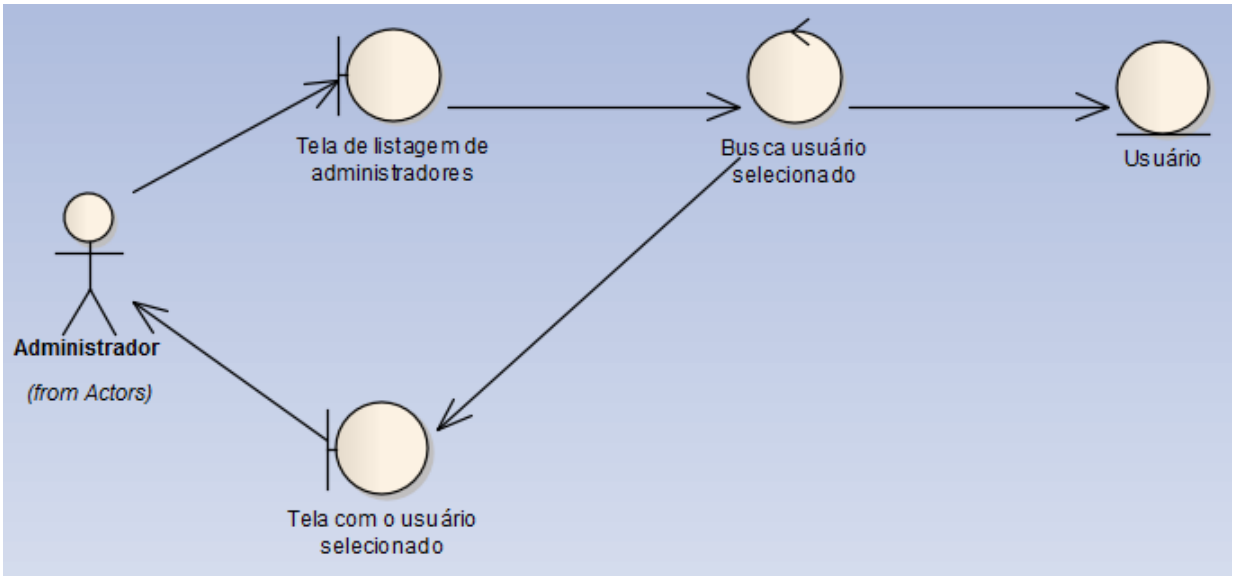
Figura 32 – Diagrama de robustez: Visualização do documento.



Fonte: Autor, 2013.

Para a operação de visualizar o perfil de um administrador (figura 33), o sistema pesquisa na base de dados as informações referentes ao usuário selecionado e mostra em uma tela separada da listagem.

Figura 33 – Diagrama de robustez: Visualização perfil.



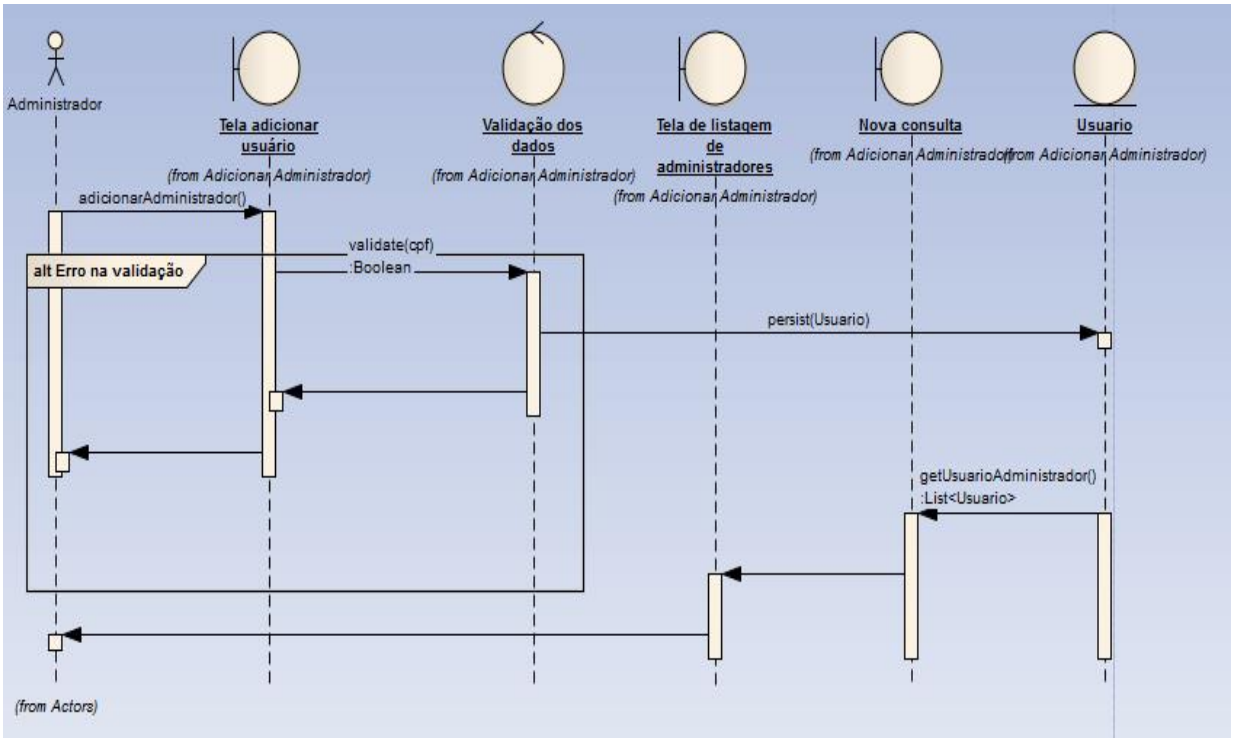
Fonte: Autor, 2013

Nesta seção foi apresentado e detalhados todos os diagramas de sequencia modelados para o sistema proposto.

4.2.7 – Diagrama de sequência

No diagrama de sequência para adicionar um administrador, podemos ver que são informados os dados do novo usuário na tela adicionar usuário. Após a inserção dos dados, é feita uma validação com os dados. Se acontecer algum erro na validação, volta para a tela de adicionar usuário. Caso não tenha erro, o usuário é inserido no banco de dados e volta na tela de listagem de administradores. A seguir, a figura 34 ilustra este diagrama.

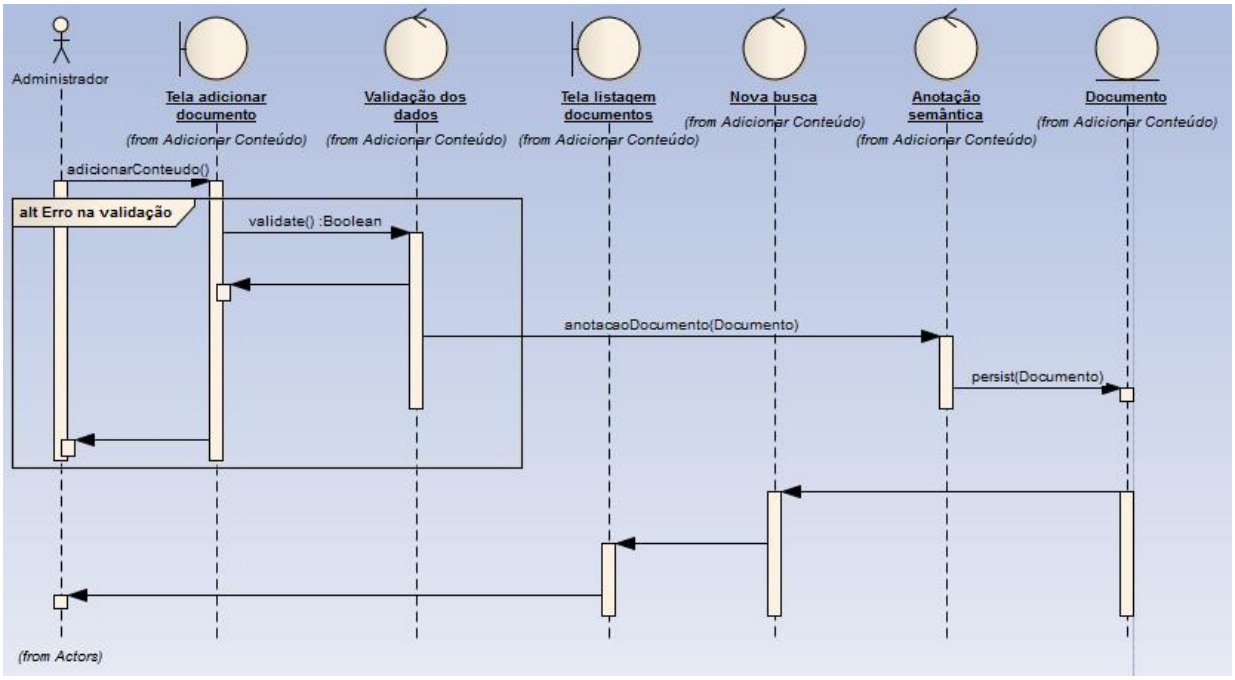
Figura 34 – Diagrama de sequência: Adicionar administrador.



Fonte: Autor, 2013

Para adicionar um documento (Figura 35), o administrador informa qual arquivo ele quer inserir. É feita uma validação dos dados, e se ocorrer algum erro na validação, volta para a tela de inserção de documento. Se não tiver erro, o documento passa pela anotação semântica e é inserido no banco de dados. Depois de inserir no banco, uma nova busca é realizada e volta para a tela de listagem de documentos.

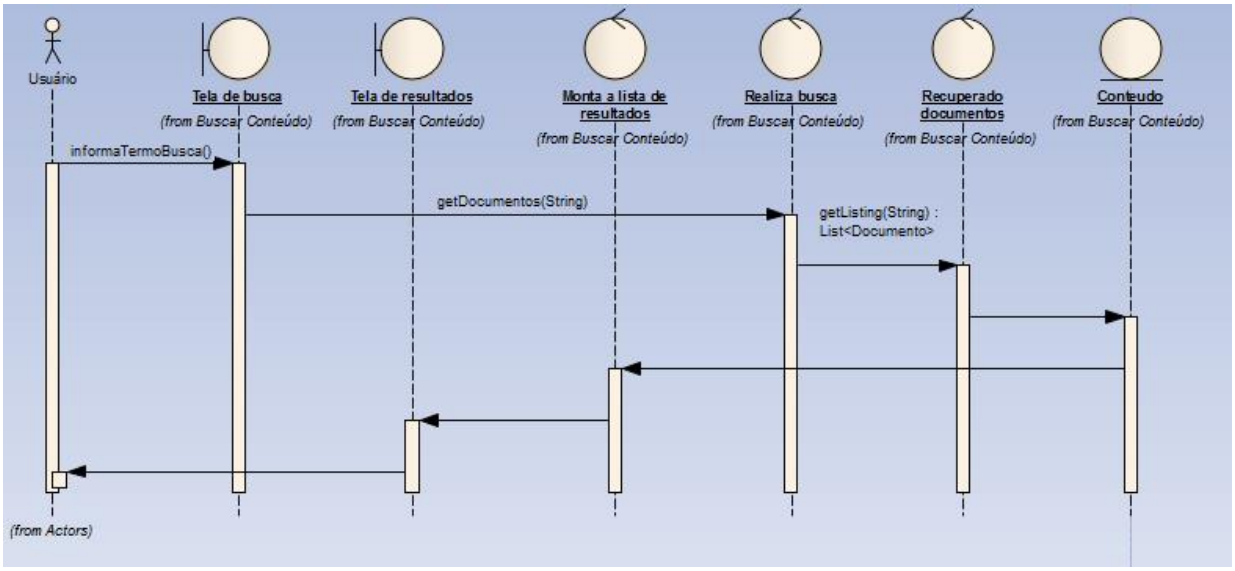
Figura 35 – Diagrama de sequência: Adicionar documento.



Fonte: Autor, 2013.

No diagrama para buscar conteúdo (Figura 36), o usuário informa o termo para busca, a aplicação busca, no banco de dados, os documentos referente ao termo inserido pelo usuário, monta a lista de documentos relevantes e mostra para o usuário de forma que o primeiro documento é o mais relevante.

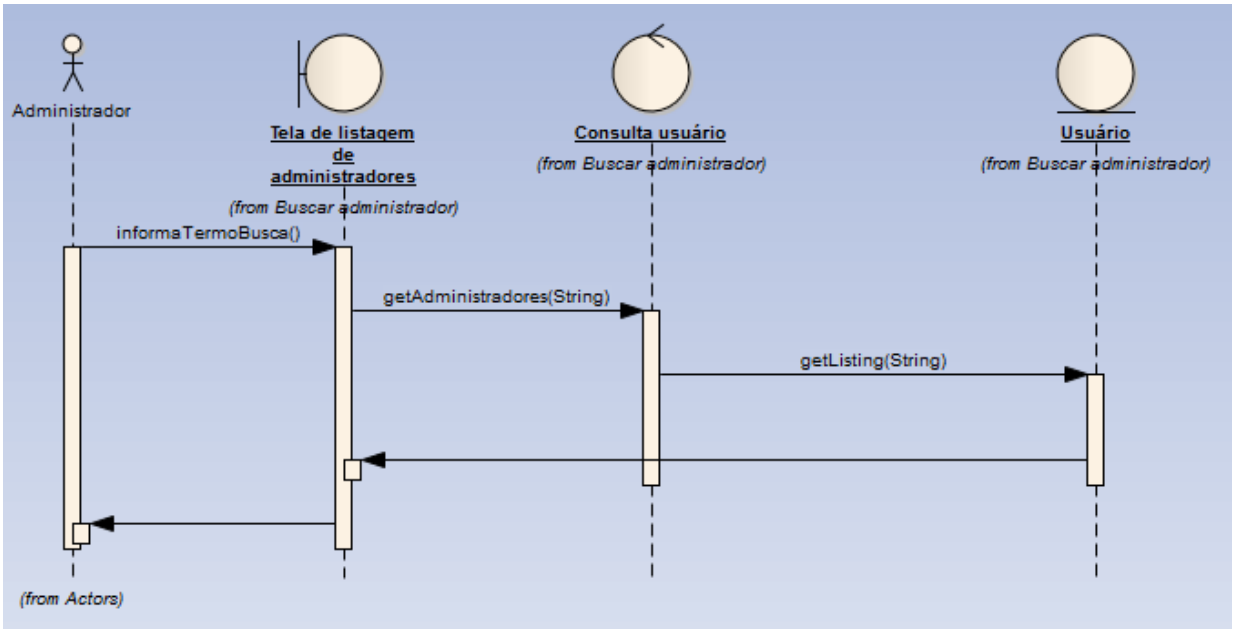
Figura 36– Diagrama de sequência: Buscar conteúdo.



Fonte: Autor, 2013.

Para buscar um usuário administrador (Figura 37), é informado na tela de listagem o nome do administrador a ser procurado, o sistema procura na base de dados usuários com o termo informado e retorna uma lista com o resultado da pesquisa.

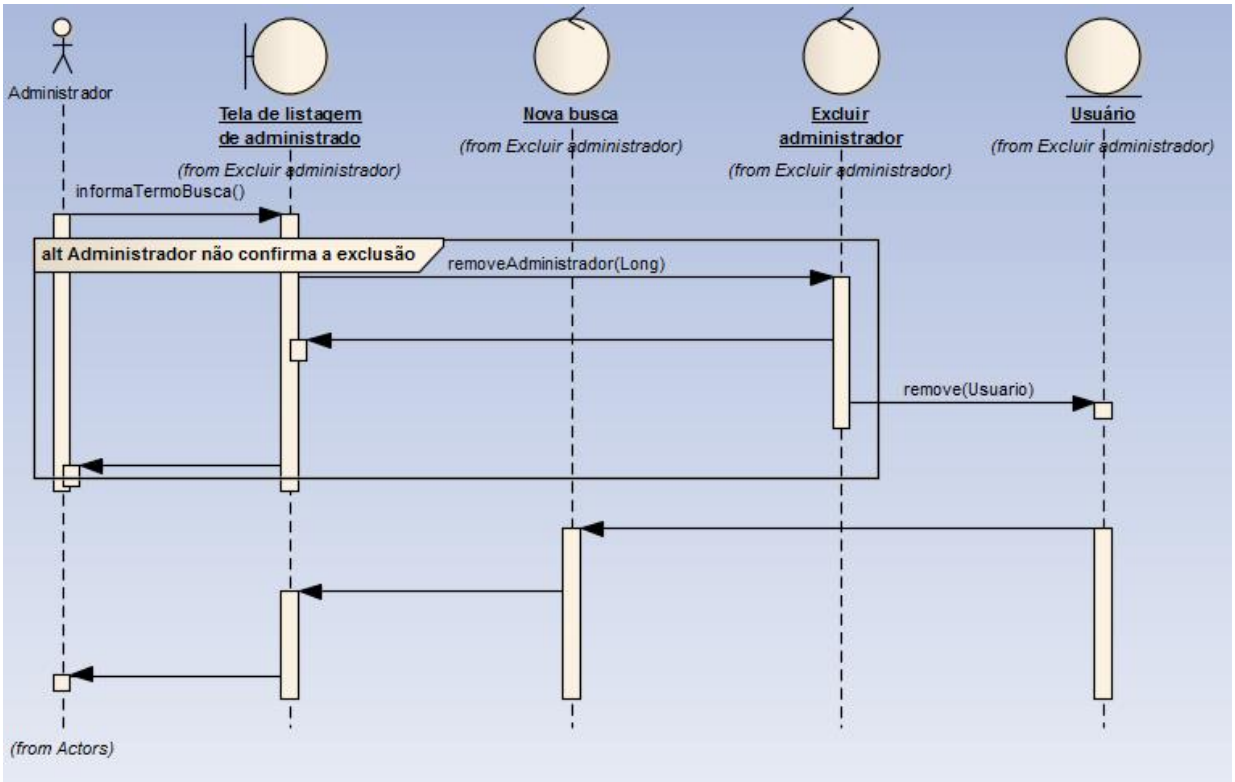
Figura 37 – Diagrama de sequência: Buscar administrador.



Fonte: Autor, 2013.

A exclusão de um administrador (figura 39) é demonstrada pelo seguinte fluxo: O usuário logado e autenticado como administrador informa na tela de listagem o nome da pessoa para ser excluída. Após selecionar o usuário (administrador) a ser excluído, é mostrada uma mensagem de confirmação de exclusão. Se confirmar, o usuário é removido do banco de dados e é feito uma nova consulta para atualizar a tela de listagem. Senão volta para a tela de listagem de administradores.

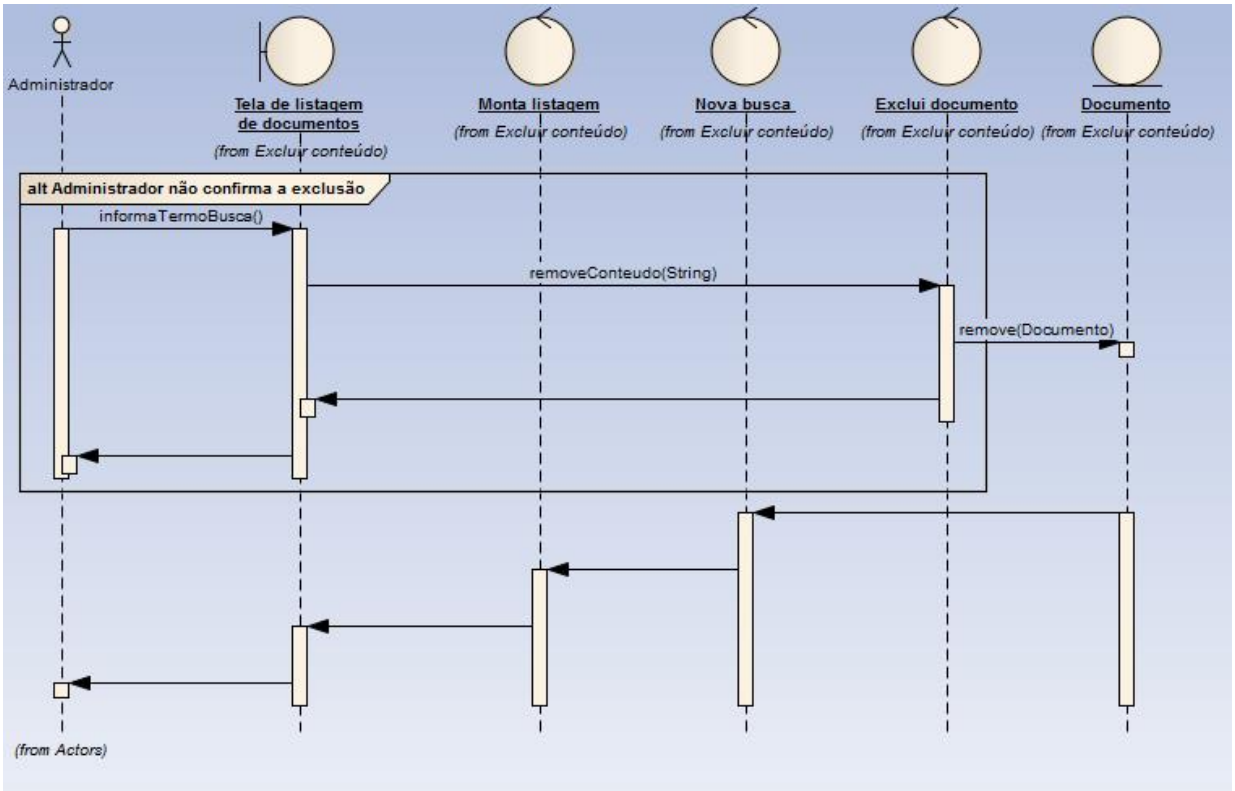
Figura 39 – Diagrama de sequência: Excluir administrador.



Fonte: Autor, 2013.

Para excluir um documento, o administrador deve informar na tela de listagem de documentos o nome do documento que será excluído. Com a lista de possíveis documentos o administrador seleciona qual ele deseja excluir. Para prosseguir com a exclusão, é feito um questionamento se o administrador realmente deseja excluir, se sim, é feito a remoção do banco de dados. Após a exclusão, é realizado uma nova busca e com o resultado é montada uma lista com todos os documentos. A figura 40 ilustra este fluxo.

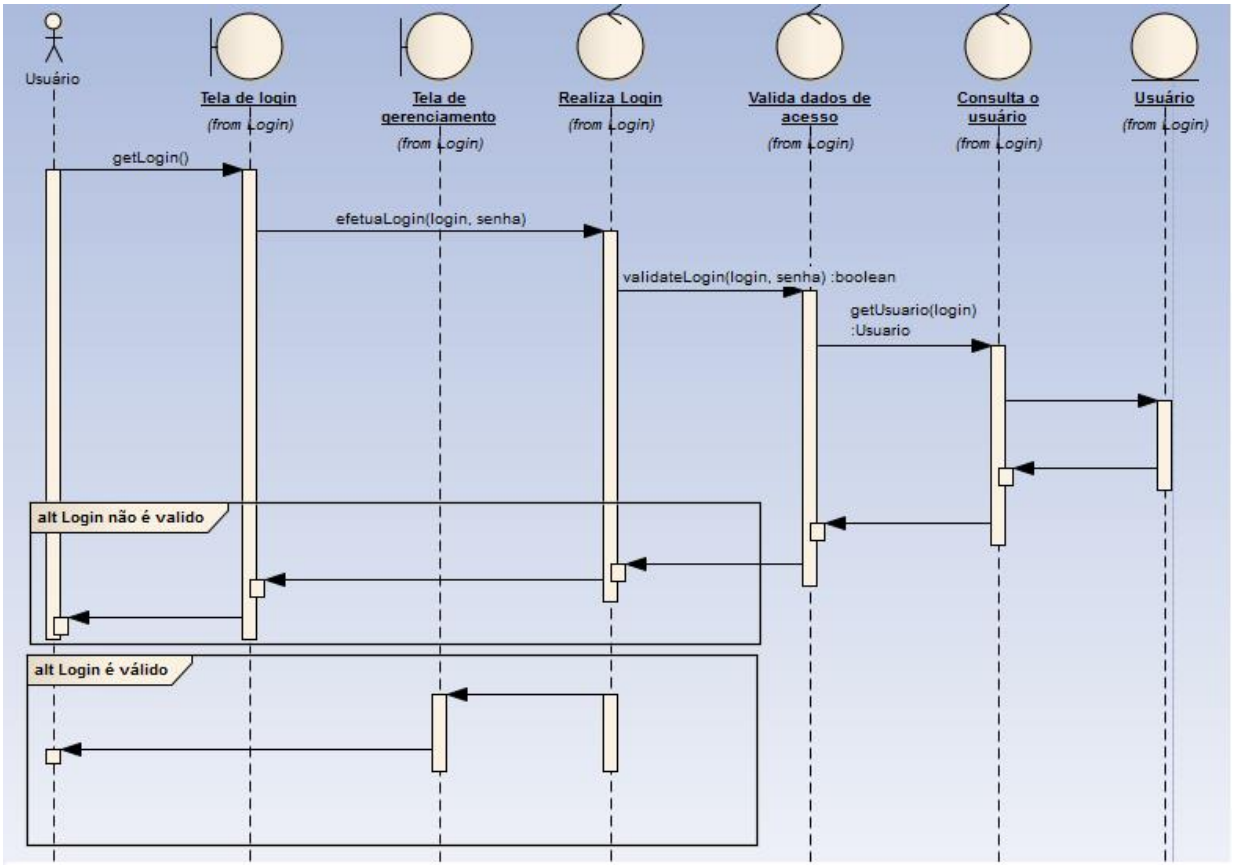
Figura 40 – Diagrama de sequência: Excluir documento.



Fonte: Autor, 2013.

Para fazer login (figura 41), o usuário informa o CPF e a senha. Antes de garantir a permissão de administrador é realizado uma verificação dos dados informados. Se o CPF e a senha foram inseridos corretamente, o usuário ganhará a permissão de administrador e é encaminhado para a tela de gerenciamento. Se os dados de acesso não forem validados, o usuário ficará na tela de login, e uma mensagem de erro é exibida.

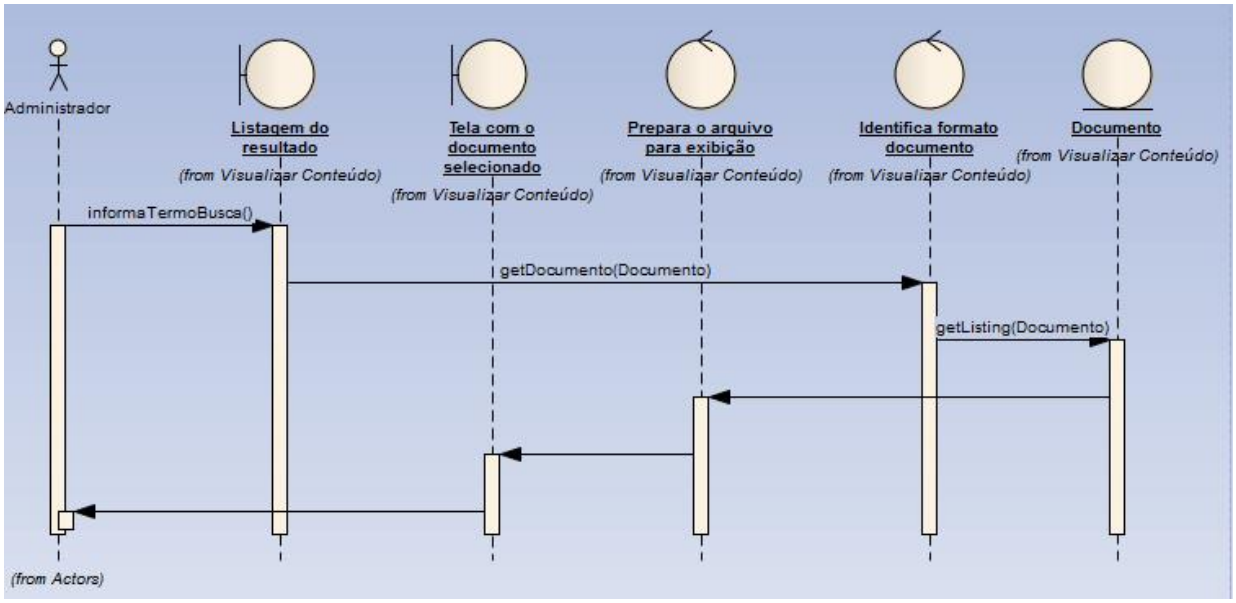
Figura 41 – Diagrama de sequência: Login.



Fonte: Autor, 2013.

No diagrama de sequência do fluxo, para visualizar conteúdo (Figura 42), verifica-se que o usuário poderá, somente, visualizar o conteúdo que foi selecionado na tela de listagem de documentos. A aplicação será responsável por identificar qual o formato do conteúdo e formatar o conteúdo para a visualização.

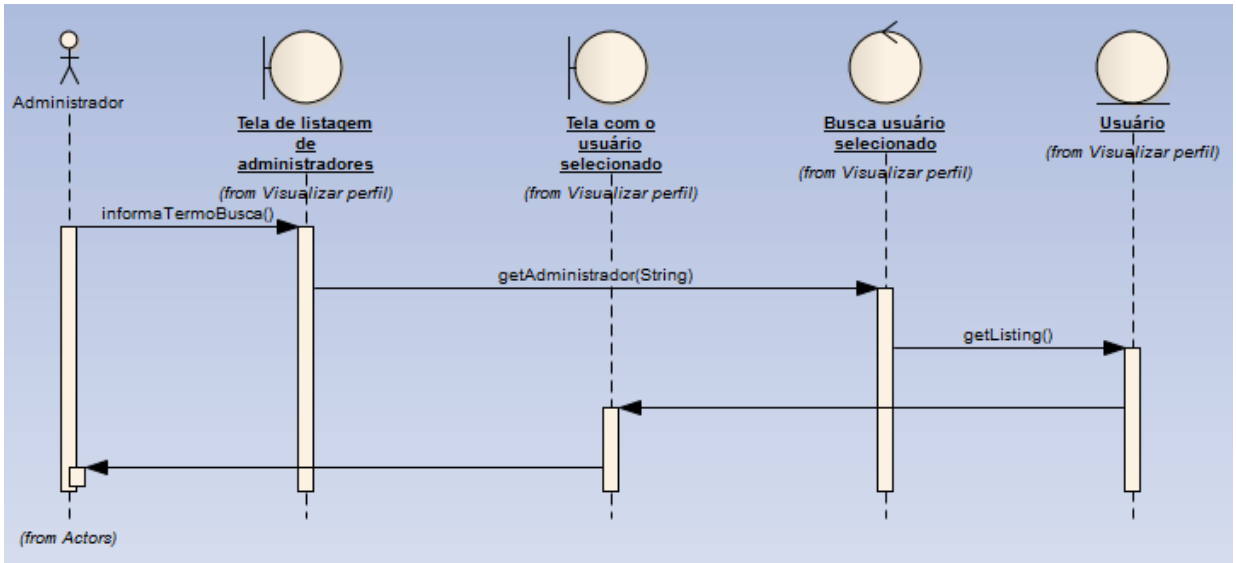
Figura 42 – Diagrama de sequência: Visualizar conteúdo.



Fonte: Autor, 2013.

No diagrama de sequência do fluxo, para visualizar perfil (Figura 43), verifica-se que o usuário poderá, somente, visualizar o perfil que foi selecionado na tela de listagem de usuários.

Figura 43 – Diagrama de sequência: Visualizar perfil.

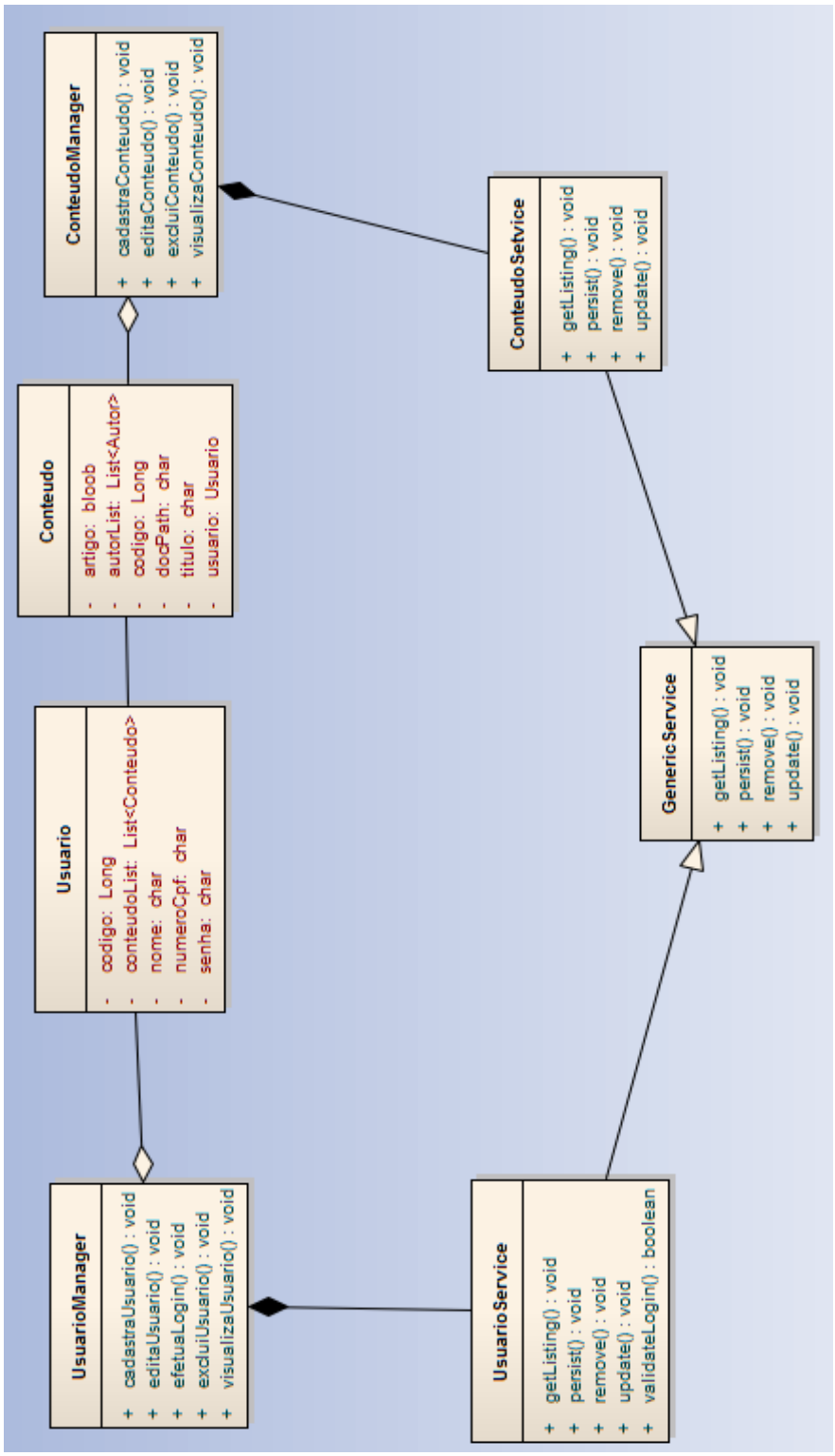


Fonte: Autor, 2013.

4.2.8 – Diagrama de classe

Na figura 44, é possível visualizar o diagrama de classe modelado para atender o protótipo proposto nesse trabalho.

Figura 44 – Diagrama de Classe.

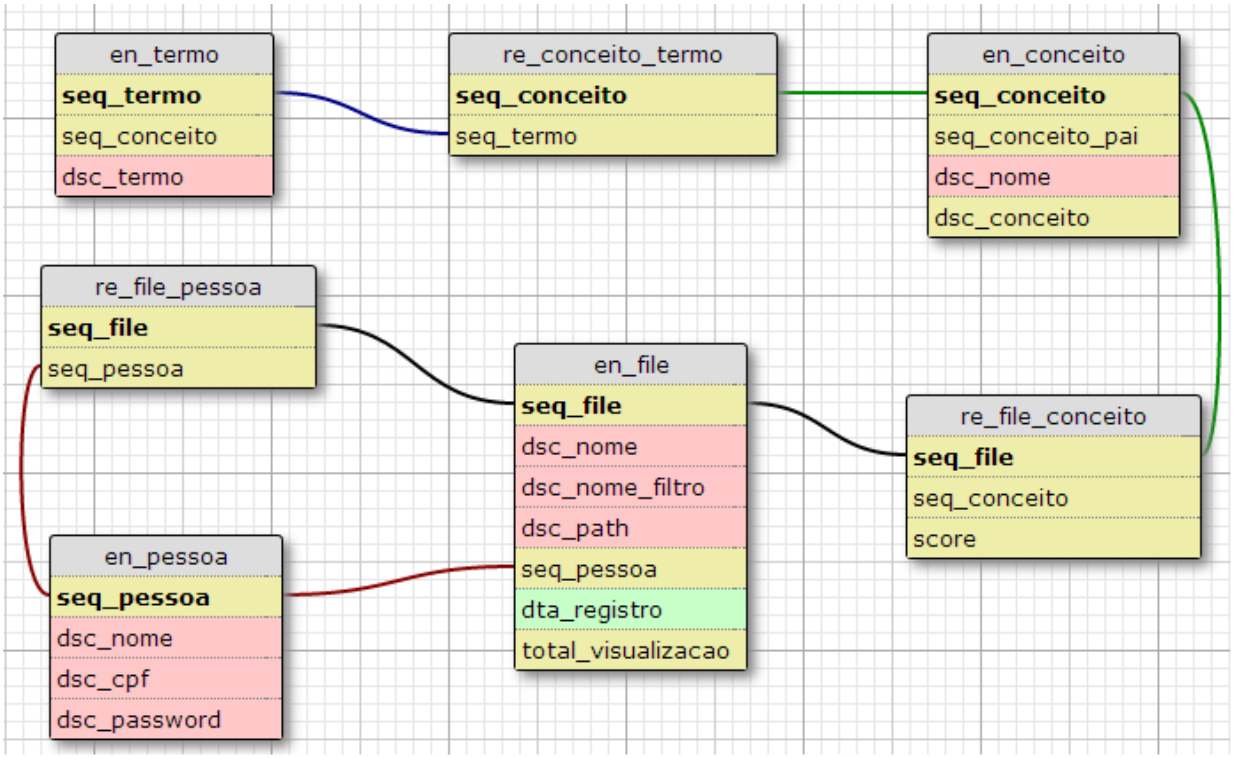


Fonte: Autor, 2013

4.2.9 – Modelo de dados

Na figura 61, é representado o modelo de dados para suportar os requisitos da aplicação.

Figura 61 – Modelo de dados.



Fonte: Autor, 2013

5 – DESENVOLVIMENTO DA SOLUÇÃO PROPOSTA

O seguinte capitulo mostra quais tecnologias foram utilizadas para o desenvolvimento da aplicação, e também contém a explicação de como a busca semântica esta atuando no sistema. Esta seção também explica como foi feita a validação do protótipo, descrevendo o método abordado, seu cenário e quais foram os resultados obtidos.

E por final, é apresentado um caso de teste, mostrando o fluxo de busca do sistema, e o fluxo de adição de documentos.

5.1 – FERRAMENTAS E TECNOLOGIAS

Nesta seção do trabalho, são apresentado as ferramentas e tecnologias utilizadas para a o desenvolvimento da solução proposta e os motivos que influenciaram na escolha destas tecnologias.

5.1.1 – Plataforma Java

Java (2014) explica que:

> Java é uma linguagem de programação e plataforma computacional lançada pela primeira vez pela Sun Microsystems em 1995. Existem muitas aplicações e sites que não funcionarão, a menos que você tenha o Java instalado, e mais desses são criados todos os dias. O Java é rápido, seguro e confiável.

Para Java (2014), java é a linguagem base para quase todos os tipos de aplicação em rede, também é padrão global para desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos conteúdo baseado na Web e softwares coorporativos. Java pode estar em datacenter, computadores, consoles de games, supercomputadores científicos , telefones celulares.

Alguns dados quantitativos segundo Java (2014) :

- 97% dos Desktops Corporativos executam o Java
- 89% dos Desktops (ou Computadores) nos EUA Executam Java

- 9 Milhões de Desenvolvedores de Java em Todo o Mundo
- A Escolha Nº 1 para os Desenvolvedores
- Plataforma de Desenvolvimento Nº 1
- 3 Bilhões de Telefones Celulares Executam o Java
- 100% dos Blu-ray Disc Players Vêm Equipados com o Java
- 5 bilhões de Placas Java em uso
- 125 milhões de aparelhos de TV executam o Java
- 5 dos 5 Principais Fabricantes de Equipamento Original Utilizam o Java ME

Além dos motivos já citados, Java é uma é totalmente estável pois foi testado, refinado, estendido e comprovado por uma comunidade dedicada de desenvolvedores, arquitetos e entusiastas do Java. Outro motivo que também influenciou na escolha da linguagem Java, foi o fato possuir experiência, tanto profissional como acadêmica.

5.1.2 – Apache Lucene

Segundo Lucene Project (2014), o Lucene é um motor de busca de texto de alto desempenho e escrito inteiramente em Java. É uma tecnologia que se adequada para quase qualquer aplicação que necessita de um pesquisador de texto, e é mantido pela comunidade Apache, possuindo o s código fonte aberto (*open source*).

O maior dos motivos da escolha do Lucene é o fato de ele ser escrito na linguagem Java, possibilitando maior facilidade na implementação e flexibilidade no desenvolvimento do protótipo proposto.

5.1.3 – Servlet 3.0

Segundo Mordani (2009), servlet é um componente *web* baseado na tecnologia Java, que é gerido por um container que gera conteúdo dinâmico. Assim como outros

componentes baseados em Java, servlets são classes Java, ou seja, de plataforma independente, não se limitando a uma plataforma para rodar.

Mordani (2009) também explica que os servlets interagem com os clientes web por meio de um paradigma de pedido e resposta (*requeste, response*) implementada por um container que permita a execução de servlets.

O que influenciou na escolha para utilizar o Servlet, foi querer a independência de utilizar algum *framework* que abstrai essa tecnologia, por ser nativo da plataforma Java, possui uma boa documentação e também por possuir experiências acadêmicas com a tecnologia.

5.1.4 – JavaServer Page

Calegari (2004) explica que:

> As páginas JSP, ou Java Server Pages, foram criadas para contornar algumas das limitações no desenvolvimento com Servlets: se em um Servlet a formatação da página HTML resultante do processamento de uma requisição se mistura com a lógica da aplicação em si, dificultando a alteração dessa formatação, em uma página JSP essa formatação se encontra separada da programação, podendo ser modificada sem afetar o restante da aplicação.

Sendo assim, uma página JSP nada mais é do que uma página HTML com alguns elementos especiais, que conferem o carácter dinâmico da página. Esses elementos podem realizar um processo em si, ou também pode recuperar uma informação de um resultado do processamento realizado em um servlet. (CALEGARI, 2004)

Para chegar ao resultado para utilizar JSP, foi feito uma pesquisa para encontrar *frameworks* capaz de suportar o protótipo proposto, e a maioria dos *frameworks* possui alto nível de complexidade para aprender a utilizar e também são feitos para suportar aplicação de grande porte. Como a solução proposta não necessita de um *framework* complexo, o JSP atendeu todos os requisitos necessários para ser utilizado na aplicação, além de poder utilizar HTML bruto na parte de visão.

5.1.5 – PostgresSQL

Postgresql (2014) explica:

> O PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional de código aberto, com mais de 15 anos de desenvolvimento. É extremamente robusto e confiável, além de ser extremamente flexível e rico em recursos. Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

A escolha desta ferramenta foi definida por ela ser *open source*, e também por ser um sistema de gerenciamento de banco de dados onde tenho experiência profissional atende todos os requisitos do protótipo proposto no projeto.

5.1.6 – Enterprise Architect

Enterprise Architect, segundo Sparx Systems (2014), é um sistema de modelagem que fornece o ciclo de vida completo para sistemas de negócios de TI e engenharia de softwares.

Enterprise Architect fornece recursos de gerenciamento de requisitos, modelos de projetos, implementação, teste e manutenção, utilizando modelos como UML, SysML, BPMN e outras especificações. O EA ajuda na construção de sistemas pequenos até sistemas robusto e facilita a manutenção do software fornecendo uma documentação completa para o usuário. O Enterprise Architect também possui módulos de integração com ferramentas de desenvolvimento, como o Eclipse e o Visual Basic.(SPARX SYSTEMS, 2014).

A escolha do Enterprise Architect foi definida pois é uma ferramenta de modelagem onde possuo maior experiência, e foi a ferramenta com o qual trabalhei durante a graduação.

5.2 – HISTÓRICO DO DESENVOLVIMENTO

A primeira etapa para o desenvolvimento da solução proposta, foi procurar uma lista de conceito e termos (thesauro) na internet. Após uma busca a fundo na internet foi encontrado o repositório de vocabulários e ontologias do governo eletrônico (http://vocab.e.gov.br/).

A segunda etapa foi definir quais tecnologias seriam utilizadas, e três delas estavam definidas, que era a linguagem Java, o banco de dados Postgres e o Lucene. Após algumas pesquisas foi integrado o framework de desenvolvimento PlayFramework, uma ferramenta MVC, mas com essa primeira ferramenta foi encontrado o primeira problema, não saber programar na linguagem Scala, o PlayFramework utiliza essa linguagem para a camada de visão onde é renderizado os componentes visuais, então foi decidido utilizar páginas JSP, onde atende os requisitos e é uma ferramenta mais simples de integrar e desenvolver.

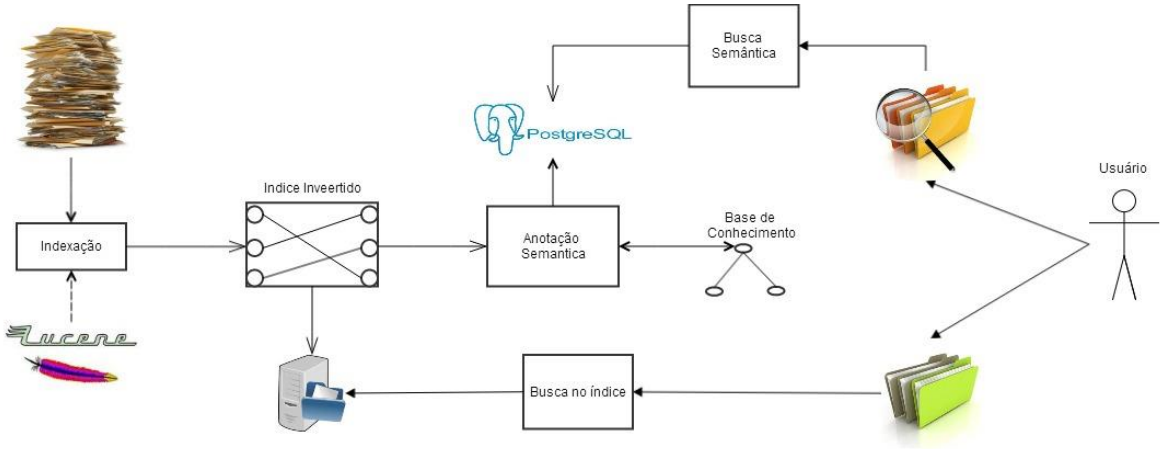
Também na segunda etapa, foi pesquisado e integrado o framework para persistência Hibernate, mas também foi encontrado barreiras na utilização da ferramenta. Então se optou utilizar o JDBC nativo do java, onde foi muito mais rápido e ágil para desenvolver a aplicação.

Apesar dos erros na escolha de tecnologia, foi possível implementar a solução proposta e chegar a um resultado positivo, conseguindo anotar semanticamente os documentos inseridos na base de dados.

5.3 – ESQUEMA FISICO DO SISTEMA

Na figura 45 é apresentado o esquema físico do sistema proposto junto com as tecnologias utilizadas.

Figura 45 – Exemplo da estrutura da árvore de conceito e termos



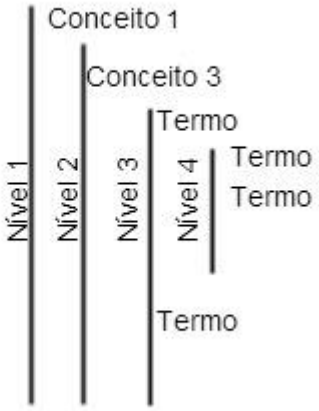
Fonte: Autor, 2013

Documentos são indexados pelo Apache Lucene gerando o índice invertido, esse índice é gravado no servidor. Com o documento já indexado é feito a anotação semântica, utilizando a base de conhecimento, sobre o documento e seu índice. Essa relação entre o documento, índice e a base de conhecimento é gravado em uma tabela no banco de dados para realizar a busca semântica. Para buscar utilizando apenas o índice do documento, a consulta é feita no índice gravado no servidor.

O sistema desenvolvido tem como foco principal a anotação semântica de um documento. A seguir é descrito as duas etapas que são necessárias para acontecer a anotação semântica, a primeira é a parte onde é gerado um índice a partir do lucene, e a outra é a anotação semântica onde o documento vai ser separado por conceito.

Foi definido que é considero um conceito somente o nível 1 e o nível 2, a seguir na figura 46, é ilustrado a árvore definida.

Figura 46 – Exemplo da estrutura da árvore de conceito e termos



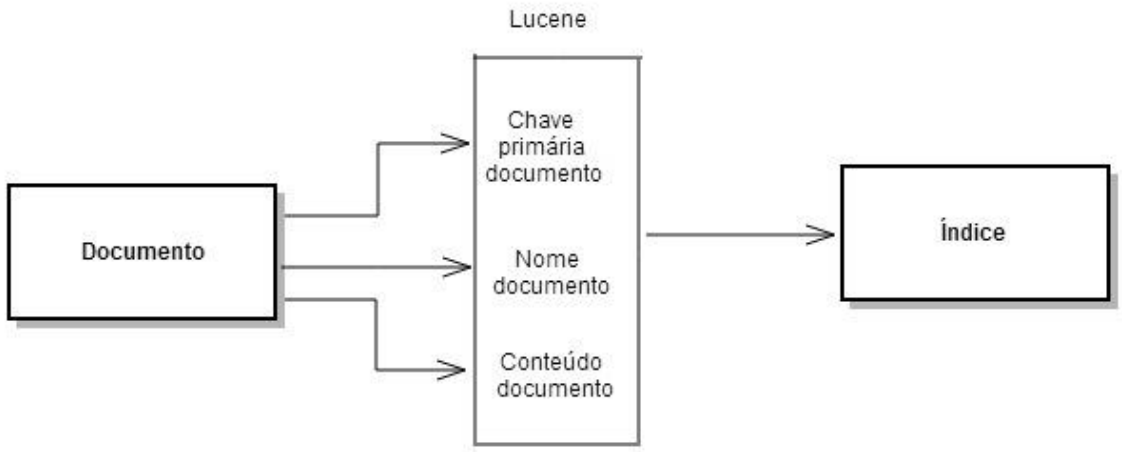
Fonte: Autor, 2014

Onde todos os termos tem como conceito pai o conceito 3 e por sua vez, o conceito 3 tem como pai o conceito 1. Conceitos no primeiro nível nunca terá um conceito pai.

5.3.1 – Indexação

A parte de indexação é onde começa o fluxo, nesta parte é gerado um índice utilizando os recursos do lucene. A figura 47 demonstra o fluxo a parte de indexação feita pelo lucene.

Figura 47 – Esquema de indexação.



Fonte: Autor, 2013

O índice de um documento é gerado a partir de sua chave primária (chave do banco de dados), o nome do conceito que esta sendo indexado, e o conteúdo extraído de dentro do documento. Essas informações que fazem parte na hora da indexação, também são utilizadas na hora da busca, pelo fato que o lucene realiza a busca sobre essas informações (também chamado de *field*)

5.3.2 – Anotação semântica

O documento passa pelo seguinte processo para realizar a anotação semântica:

Sobre o documento indexado (documento que vai ser anotado) é realizado uma pesquisa, utilizando os recursos do Lucene. Para realizar a pesquisa é feito uma iteração sobre os conceitos e termos extraídos da base de conhecimento, após realizar a pesquisa sobre o termo/conceito em questão o Lucene gera um *score* automático, indicando a relevância da pesquisa dentro do documento. Com o *score* que o Lucene define para o conceito (ou termo) dentro do documento, conseguimos identificar conceitos chaves. Após a identificação dos

conceitos chaves, a aplicação salva no banco de dados a relação entre o conceito e o documento.

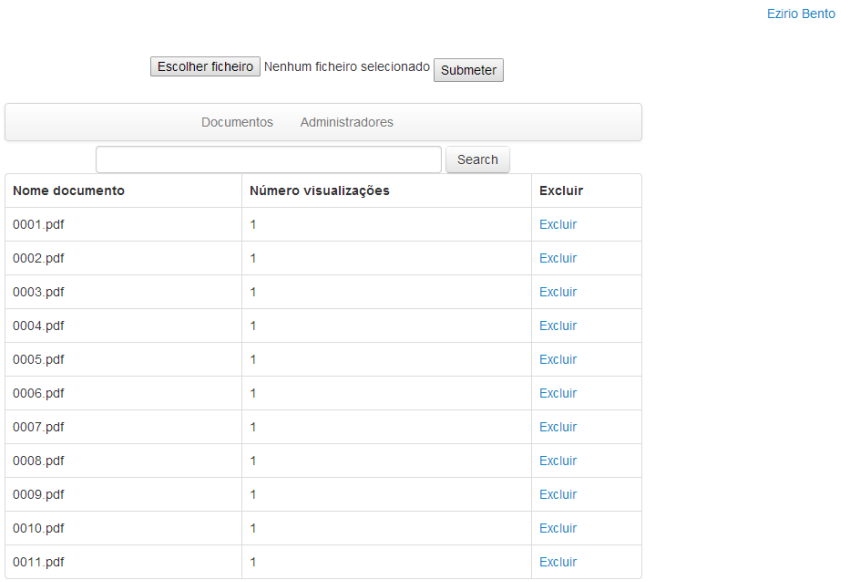
Para realizar uma busca, o protótipo se comporta da seguinte maneira: é informado o termo de busca e o primeiro fluxo é buscar esse termo de busca no índice do lucene. Com a lista de documentos retornado da busca indexada, é feito a separação do documento com os termos que são relacionados, para fazer essa separação busca-se na base de dados as informações salvas na parte da anotação (relação documento x conceito), e monta o resultado separando os conceitos relacionados aos documentos retornados do índice.

Supondo que a base de dados possui documentos relacionados com a palavra java e a os conceitos relacionado a todos os documentos que possuem a palavra java são; Ilha java, café java e linguagem java. Quando um usuário for fazer uma consulta, por exemplo, com a palavra java, o lucene traz todos os documentos que contém a palavra java, mas a aplicação filtra esses documentos e separar por conceitos, Ilha java, café java e linguagem java. A cada conceito que o usuário clicar, é mostrado somente os documentos relacionados ao mesmo.

5.4 – SISTEMA DESENVOLVIDO

A figura 48 ilustra a tela inicial do gerenciamento do sistema, para acessar essa parte do sistema precisa ser um usuário autenticado. Nesta tela é possível inserir novos documentos clicando no botão escolher ficheiro e submeter. Também é possível excluir documentos do sistema, clicando no botão excluir na linha referente ao documento a ser excluído da aplicação.

Figura 48 – Página inicial do gerenciamento.



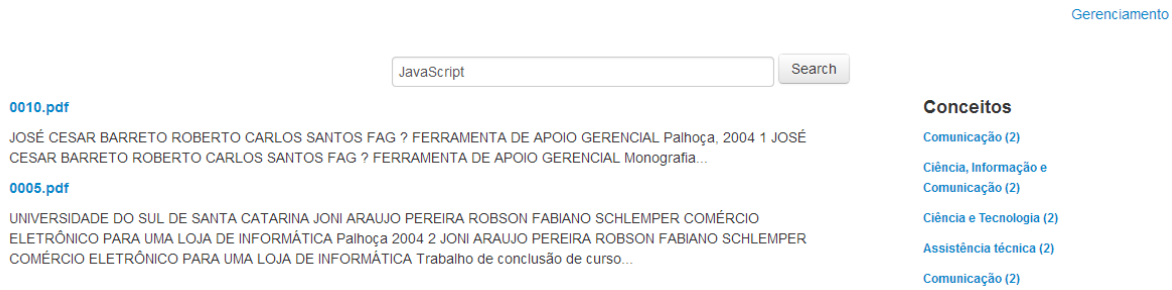
Fonte: Autor, 2013

Na lista de documentos na página inicial do gerenciamento, são documentos recuperados do banco de dados e não do indice. No campo pesquisa, é inserido o o nome do documento a ser procurado, que também vai ser buscado na base de dados.

Nessa mesma tela tem dois botões, “Documentos” onde são mostrados todos os documentos da aplicação recuperados da base de dados, e botão “Administradores” que mostra uma lista com todos os administradores do sistema.

Na página principal do sistema, onde é feito consultas utilizando os recursos semânticos. Nesta parte do sistema o usuário não precisa estar com permissão de acesso, ou seja, qualquer usuário pode utilizar ferramenta de busca. A seguir a figura 49 ilustra uma pesquisa.

Figura 49 – Tela de busca.



Fonte: Autor, 2013

É nessa parte do sistema que acontece a ação mais importante do protótipo, a separação dos documentos por conceito. A busca é feita sobre o índice gerado pelo lucene, e junto com esse índice é retornado a chave primaria do arquivo que vai servir para pesquisar na base de dados os conceitos relacionados aos documentos retornados da busca.

Quando o usuário clica em um conceito, o sistema busca somente os documentos relacionados a aquele conceito, atualizando os documentos da tela. E todo documento esta disponível para *download* no formato em que ele foi inserido, a cada download feito é contabilizado uma visualização para o documento.

5.5 – AVALIAÇÃO DO SISTEMA

O protótipo do sistema proposto foi validado através de uma pesquisa realizada com usuários que não possuem conhecimento do sistema. Para a entrevista foi aplicado um formulário com perguntas referente ao sistema e a sua usabilidade e funcionalidades. Com a aplicação do formulário buscaram-se resultados qualitativos para a proposta de resolução do protótipo apresentado.

5.5.1 – Estudo de caso

Para realizar este estudo de caso foi utilizada uma base com 89 documentos. Todos os documentos são monografias do curso de Sistemas de Informação e Ciência da Computação da Universidade do Sul de Santa Catarina.

As cargas dos documentos foram feitas através da aplicação. No ambiente privado da aplicação possui um botão para escolher o documento a ser inserido no sistema. Selecionando o documento e confirmando a ação, inicia-se o processo de indexação e anotação do documento. Para a indexação, o Lucene conta com três atributos do documento, seu nome, o conteúdo do documento e a chave primária da tabela de documentos do banco de dados.

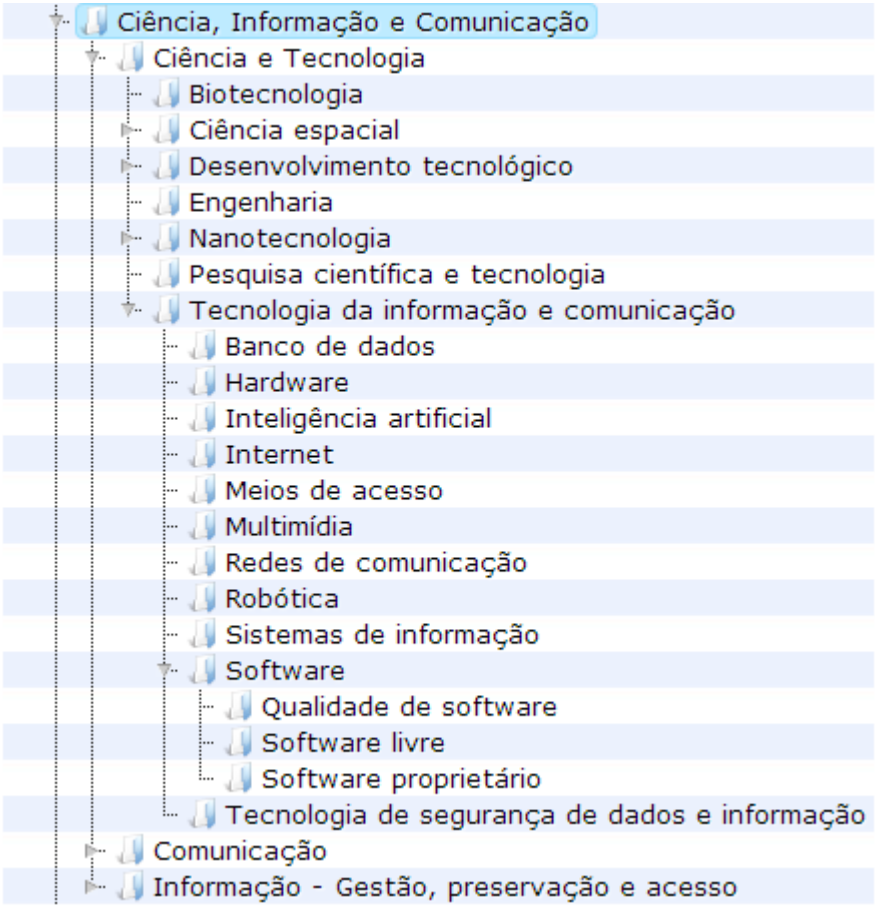
Feito a indexação com os atributos, o documento passa por outro processo, o de anotação semântica. Para realizar a anotação semântica o documento precisa estar indexado e as tabelas do banco de cacos que contém os conceitos e termos precisam estar populadas com as informações extraídas da base de conhecimento. Neste processo é carregado em memória todos os conceitos e termos da base, e buscado um a um dos conceitos e termos no documento indexado utilizando os recursos do Lucene, O resultado da busca no documento é gravado na tabela de relação documento x conceito e o score gerado pelo Lucene.

Para realizar a carga da base de conhecimentos para o banco de dados, foi desenvolvida uma aplicação que lê um arquivo texto que foi estruturado de acordo com a base de conhecimento, então se no futuro essa base de conhecimento for modificada ou se precisar inserir um novo conceito, basta preencher esse arquivo texto na estrutura correta e rodar a aplicação de novo.

A seguinte estrutura foi definida para o arquivo da base de conhecimento, primeiro é inserido de qual nível é conceito (ou termo), seguido com o nome do conceito (ou termo) e no final da linha do arquivo deve-se apresentar qual o nome do conceito pai do conceito em questão. Para finalizar a carga e inserir nas tabelas no banco de dados, os conceitos são inserido na tabela de conceitos, os termos são inseridos na tabela de termos e se um termo pertencer a mais de um conceito é gravado na tabela de relação entre conceito x termo.

Foi definido que só são considerados conceitos, palavras que estão até o segundo nível da base de conhecimento o restante são termos. A figura 50 demonstra a tabela de conhecimento e seus níveis.

Figura 50 – Base de Conhecimento.



Fonte: Vocabulário Controlado do Governo Eletrônico, 2011.

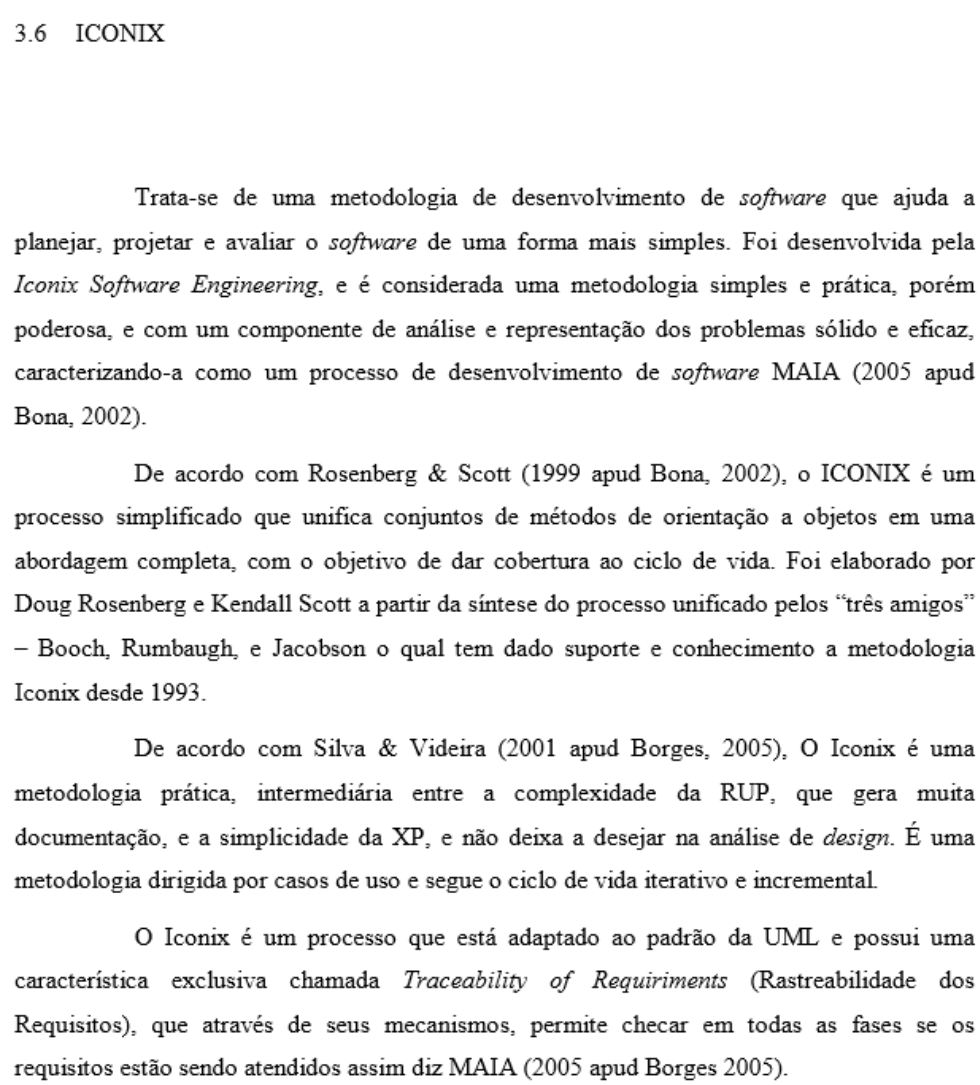
Feito o estudo de caso e dado o cenário as próximas etapas são, a entrevista com o usuário, e o caso de teste onde é avaliado o protótipo proposto nesse trabalho.

5.5.3 – Caso de teste

Nesta seção do trabalho é apresentado um caso de teste, fazendo todo o ciclo da aplicação, iniciando com a inserção do documento da aplicação até a sua recuperação. A ideia do caso de teste, é verificar se documento inserido no inicio do teste, vai fazer parte do resultado da busca feita pelo termo. O termo escolhido para realizar o teste é ICONIX.

Para realizar o teste, foi escolhido um documento com o nome “Xp versus ICONIX comparação de métodos ágeis”, e que contém o termo “ICONIX”, como conseguimos ver na figura 60.

Figura 60 – Documento com o termo ICONIX.



3.6 ICONIX

Trata-se de uma metodologia de desenvolvimento de *software* que ajuda a planejar, projetar e avaliar o *software* de uma forma mais simples. Foi desenvolvida pela *Iconix Software Engineering*, e é considerada uma metodologia simples e prática, porém poderosa, e com um componente de análise e representação dos problemas sólido e eficaz, caracterizando-a como um processo de desenvolvimento de *software* MAIA (2005 apud Bona, 2002).

De acordo com Rosenberg & Scott (1999 apud Bona, 2002), o ICONIX é um processo simplificado que unifica conjuntos de métodos de orientação a objetos em uma abordagem completa, com o objetivo de dar cobertura ao ciclo de vida. Foi elaborado por Doug Rosenberg e Kendall Scott a partir da síntese do processo unificado pelos “três amigos” – Booch, Rumbaugh, e Jacobson o qual tem dado suporte e conhecimento a metodologia Iconix desde 1993.

De acordo com Silva & Videira (2001 apud Borges, 2005), O Iconix é uma metodologia prática, intermediária entre a complexidade da RUP, que gera muita documentação, e a simplicidade da XP, e não deixa a desejar na análise de *design*. É uma metodologia dirigida por casos de uso e segue o ciclo de vida iterativo e incremental.

O Iconix é um processo que está adaptado ao padrão da UML e possui uma característica exclusiva chamada *Traceability of Requiriments* (Rastreabilidade dos Requisitos), que através de seus mecanismos, permite checar em todas as fases se os requisitos estão sendo atendidos assim diz MAIA (2005 apud Borges 2005).

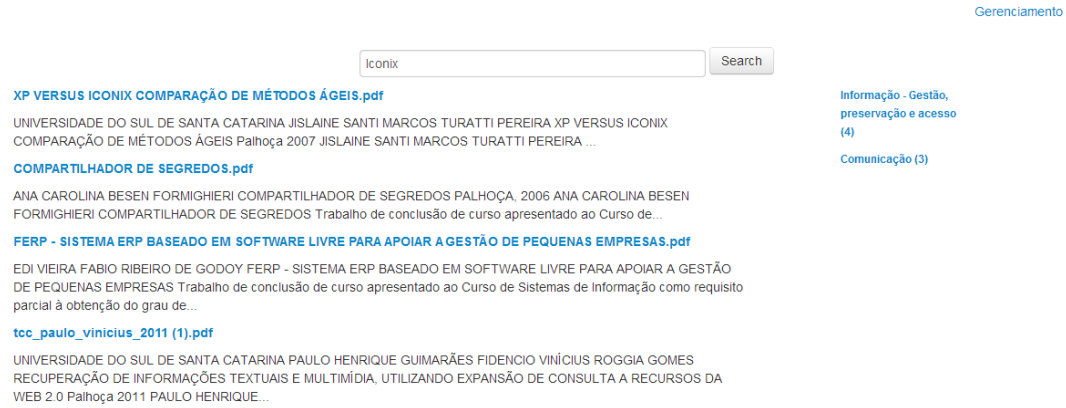
Fonte: Autor, 2013

O documento inserido na aplicação vai passar pelos processos de indexação e anotação semântica, para conseguirmos buscar com o termo que foi definido.

Após o documento passar pelo processo de inserção, foi feito uma pesquisa com o termo “ICONIX”. Como podemos ver na figura 56, pesquisando por “ICONIX”, temos como

retorno o documento que foi inserido no começo do teste, e também outros documentos relacionados a pesquisa.

Figura 61 – Resultado pesquisa com o termo “Iconix”.



Fonte: Autor, 2013

Na lateral estão listados todos os conceitos relacionados ao termo “ICONIX” junto com o total de documentos por conceito. Para essa consulta foi encontrado dois conceitos, “Informação – Gestão, preservação e acesso” e “Comunicação”. Quando for clicado em qualquer um dos conceitos, é mostrado os documentos relacionado ao mesmo. Os documentos foram classificados por conceitos (desambiguação), a partir do seu conteúdo utilizando como base uma base de conhecimento.

Nesta parte do trabalho foi apresentado os dois principais fluxos do protótipo proposto, a inserção de um documento na aplicação e a recuperação de informação. Baseado no caso de teste, podemos afirmar a aplicação recuperou o documento certo e conseguiu separar os documentos por conceito.

5.5.2 – Entrevistas com usuário

A validação foi feita através de um questionário de nove questões, onde tem perguntas relacionadas ao sistema e à sua proposta de solução. Para iniciar a entrevista, é realizado antes de apresentar o questionário uma breve apresentação do sistema e dos seus objetivos e funcionalidades. Feito a apresentação, é liberado para o usuário navegar no sistema durante quinze minutos, após esses quinze minutos, o usuário responde o questionário de acordo com que ele viu na navegação do sistema.

As nove questões apresentado ao usuário (entrevistado), terão como respostas 3 alternativas;

1. Atende o esperado.
2. Atende em partes.
3. Não atende.

As seguintes questões foram apresentadas no questionário:

- O sistema efetua o registro de novos documentos?
- É feito a anotação semântica?
- O sistema é de fácil navegação?
- O sistema recupera documentos relevantes à pesquisa?
- O sistema separa a pesquisa por conceitos?
- Em uma pesquisa considera simples, o sistema recupera documentos relevantes?
- O sistema tem um bom desempenho na hora de inserir um novo documentos na aplicação?
- Na hora de uma consulta, o sistema tem um bom desempenho?
- A interface é amigável e intuitiva?

5.5.2.1 – Cenário de avaliação

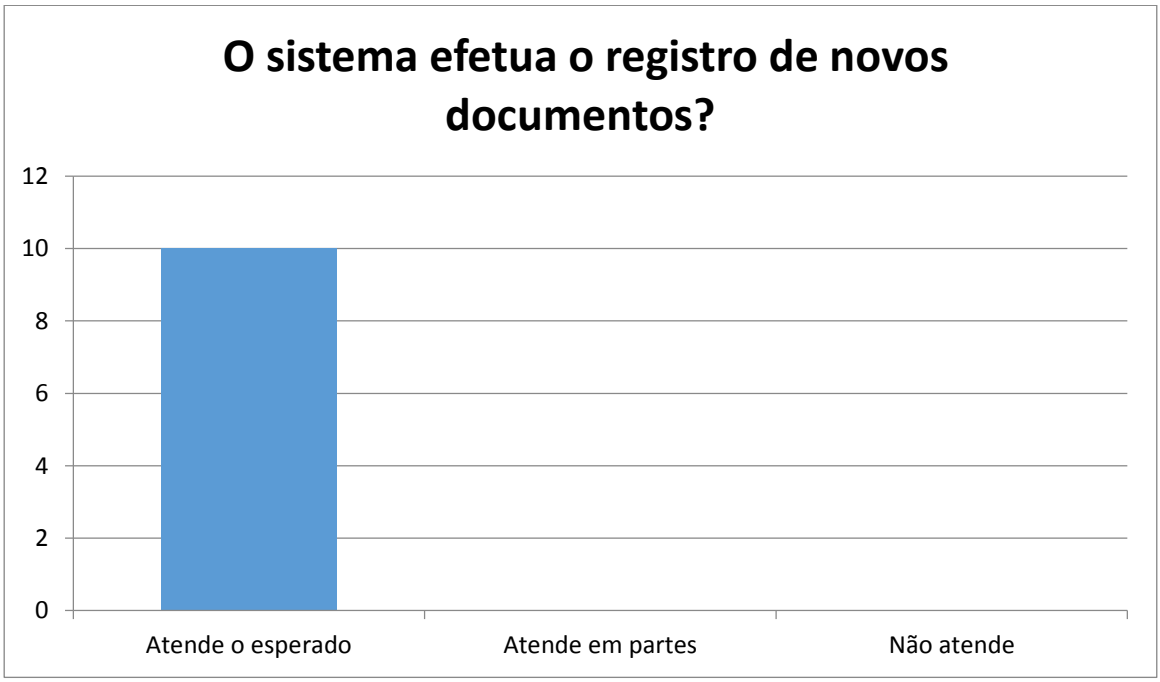
A amostra é composta por 10 entrevistados, de ambos os sexos e faixa de idade entre 20 e 45 anos. A amostra também foi dividida em dois grupos, um grupo com profissionais que atuam na área de tecnologia e informação e o outro grupo não focal, formado por profissionais de diversas áreas.

O cenário apresentado aos entrevistados, foi uma plataforma de recuperação de informação que tem como objetivo retornar documentos relevantes ao termo buscado, e separar esses documentos por conceito, a onde o usuário clica em cada conceito e é mostrado os documentos referentes ao conceito clicado.

5.5.2.2 – Resultados da avaliação

Nessa seção encontra-se o resultado da avaliação baseando-se no questionário feito com um grupo de 10 pessoas. Os gráficos a seguir exibem o resultado da avaliação.

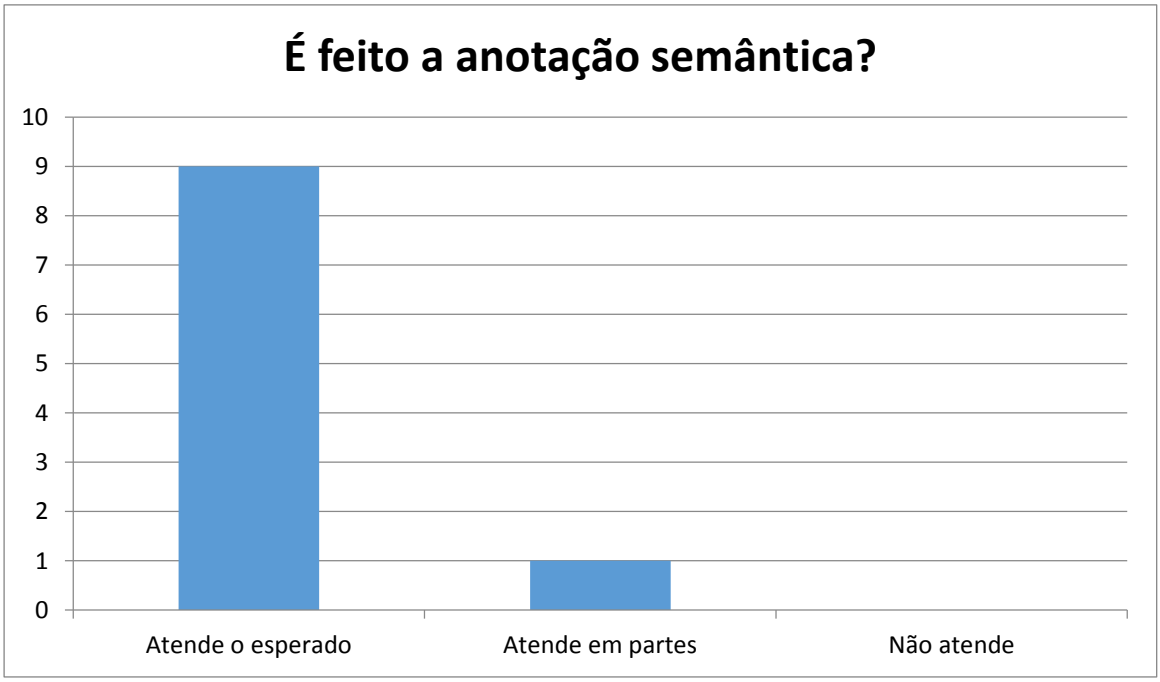
Figura 51 – Questão 1.



Fonte: Autor, 2014.

A figura 51 – Questão 1, tinha como proposito perguntar para o entrevistado a eficácia do sistema na hora de realizar novos registros de documentos na aplicação. E para 100% dos entrevistados , o sistema atende o esperado.

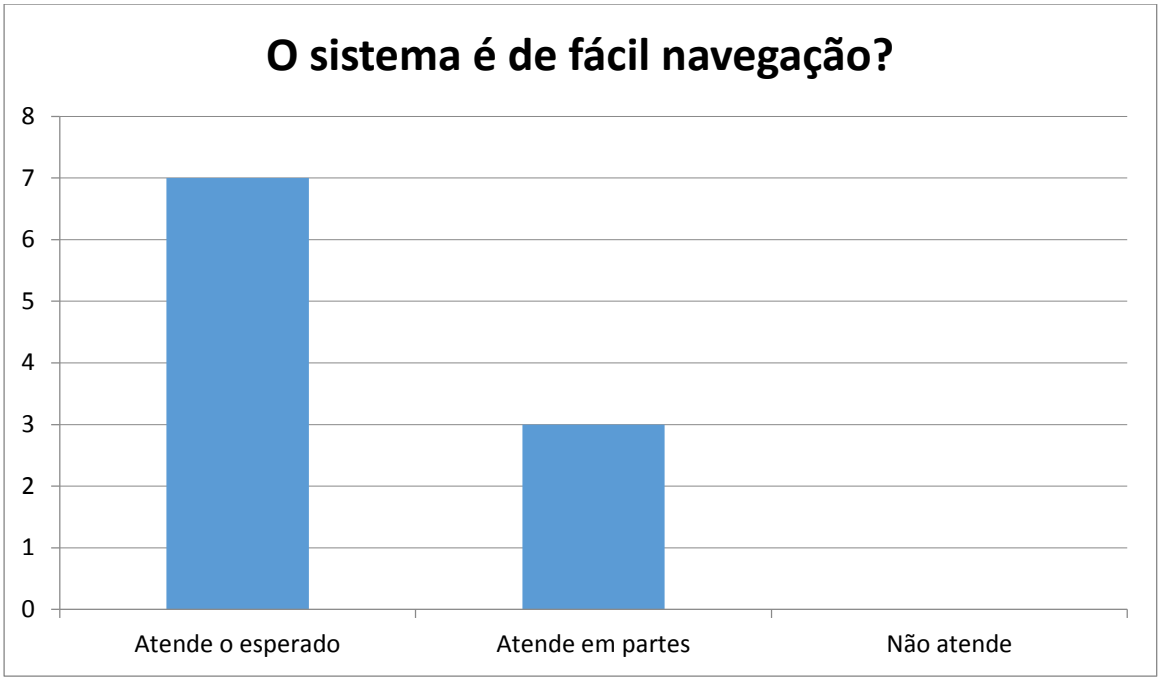
Figura 52– Questão 2.



Fonte: Autor, 2014.

A figura 52 – Questão 2, comprova que, para 90% dos entrevistados, o sistema atende o esperado no que se diz na anotação semântica. E para os 10% restantes, atende em partes.

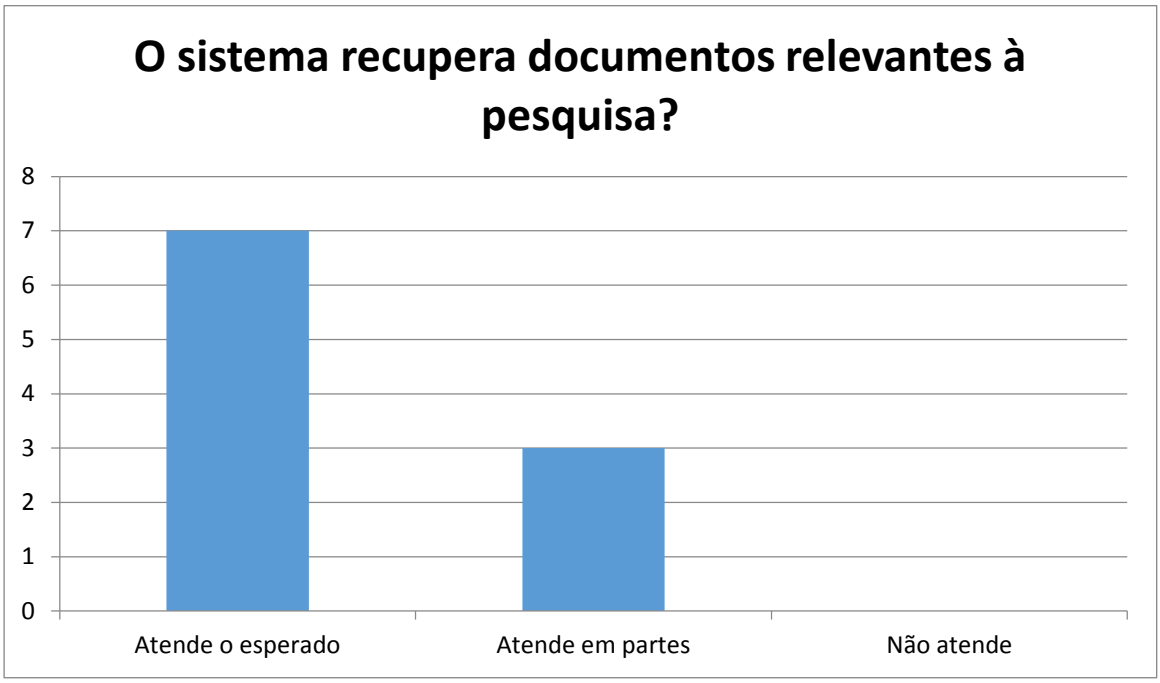
Figura 53 – Questão 3.



Fonte: Autor, 2014.

O gráfico apresentado na Figura 53 – Questão 3, tem como objetivo comprovar que o sistema é de fácil navegação. Observa-se que, em 70% dos casos, atende completamente, e 30% atende em partes.

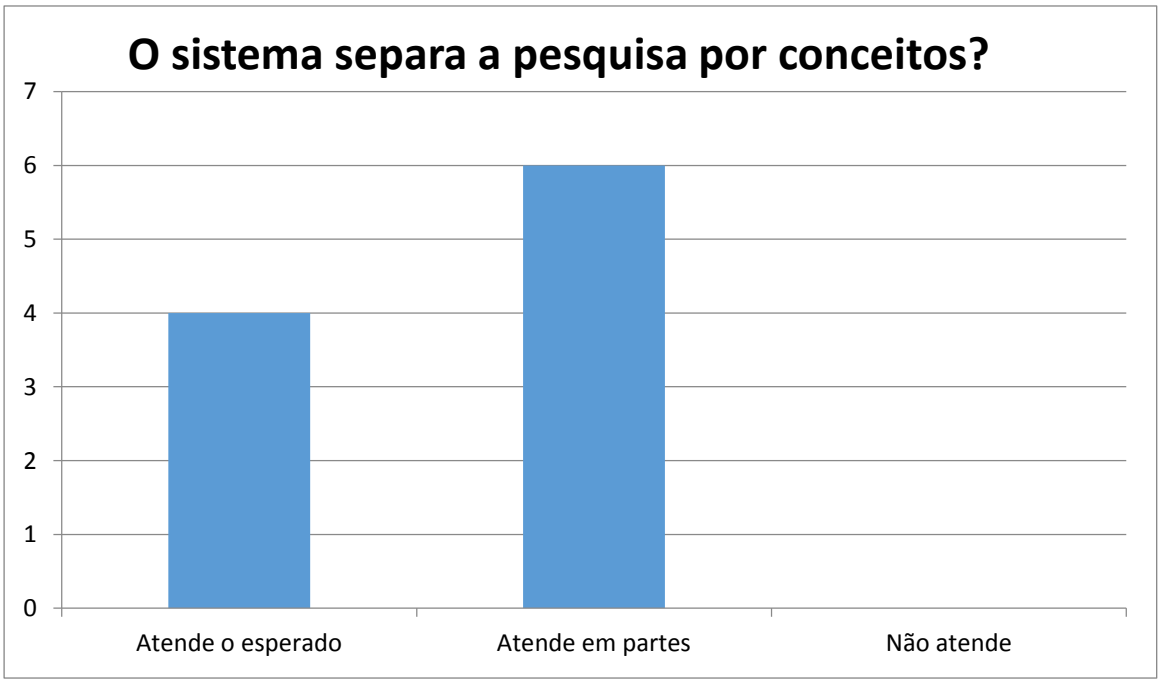
Figura 54 – Questão 4



Fonte: Autor, 2014.

Um dos fatores mais importantes para a recuperação de informação e para um sistema de busca, é se os resultados retornados da pesquisa são relevantes, e o gráfico (figura 54) mostra que para 70% dos entrevistados o resultado foi considerado relevantes, e para 30% dos entrevistados os resultados atende em partes.

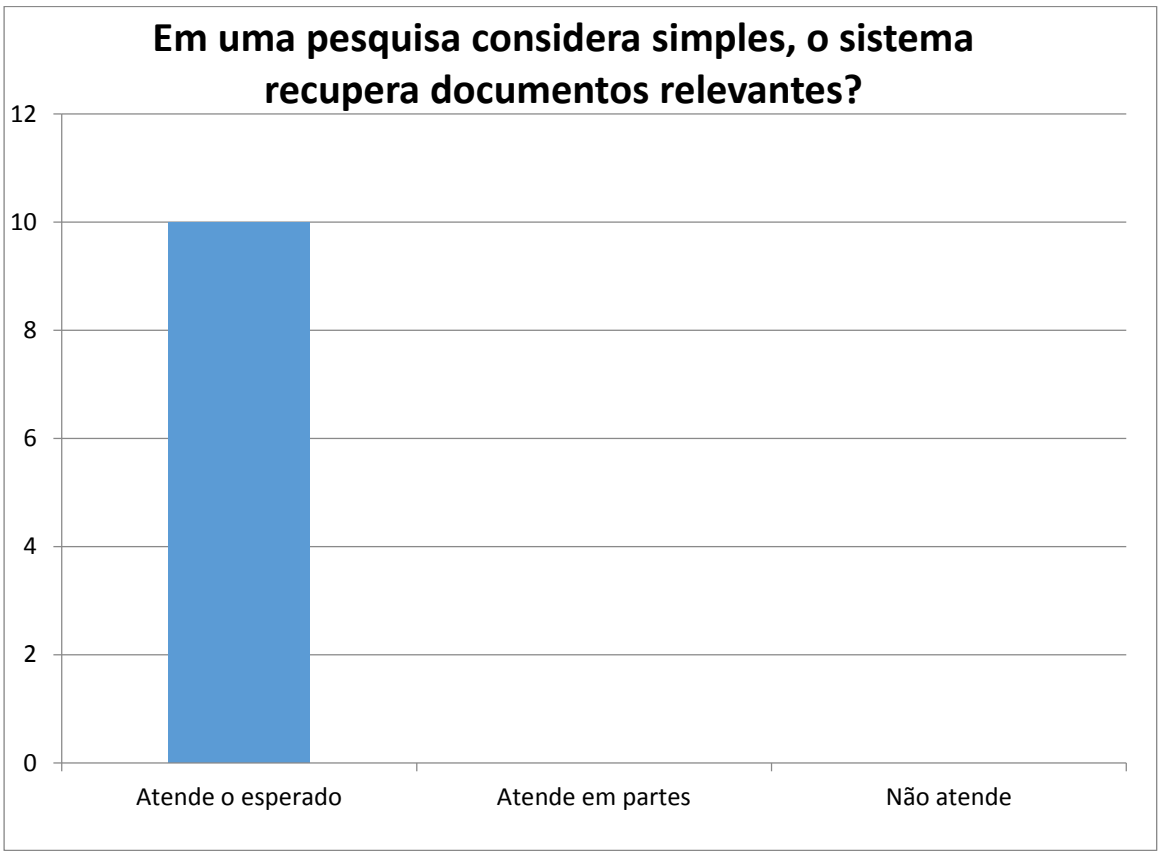
Figura 55 – Questão 5



Fonte: Autor, 2014.

A figura 55 – Questão 5, tinha como proposito perguntar para o entrevistado se o sistema separa o resultado da pesquisa por conceitos. Para 40% dos entrevistados o sistema atende o esperado, já para 60% o sistema atende em partes.

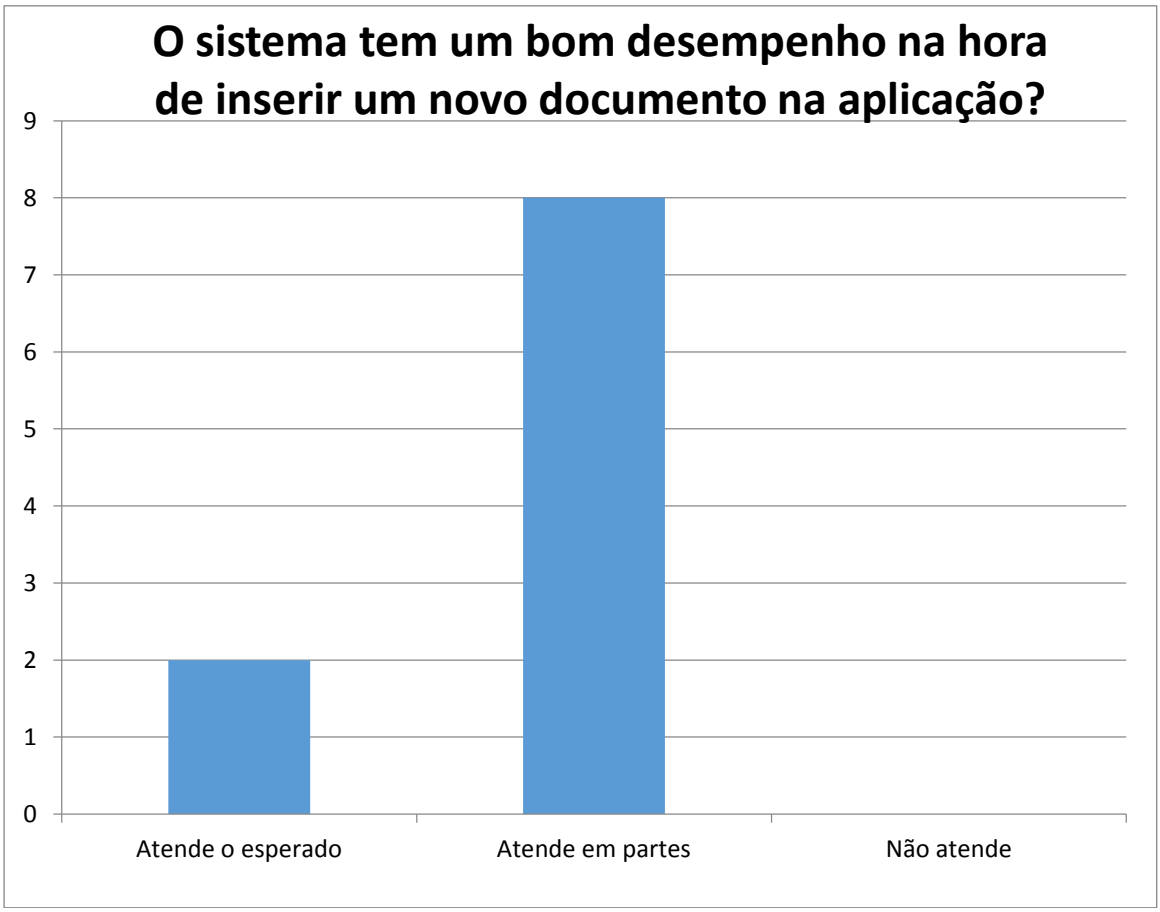
Figura 56 – Questão 6.



Fonte: Autor, 2014.

O gráfico apresentado na figura 56, é referente a questão 6 do questionário, onde foi perguntado ao entrevistado se em uma pesquisa considerada simples, o sistema recupera documentos relevantes. Com unanimidade, 100% dos entrevistados responderam que o sistema recupera documentos relevantes.

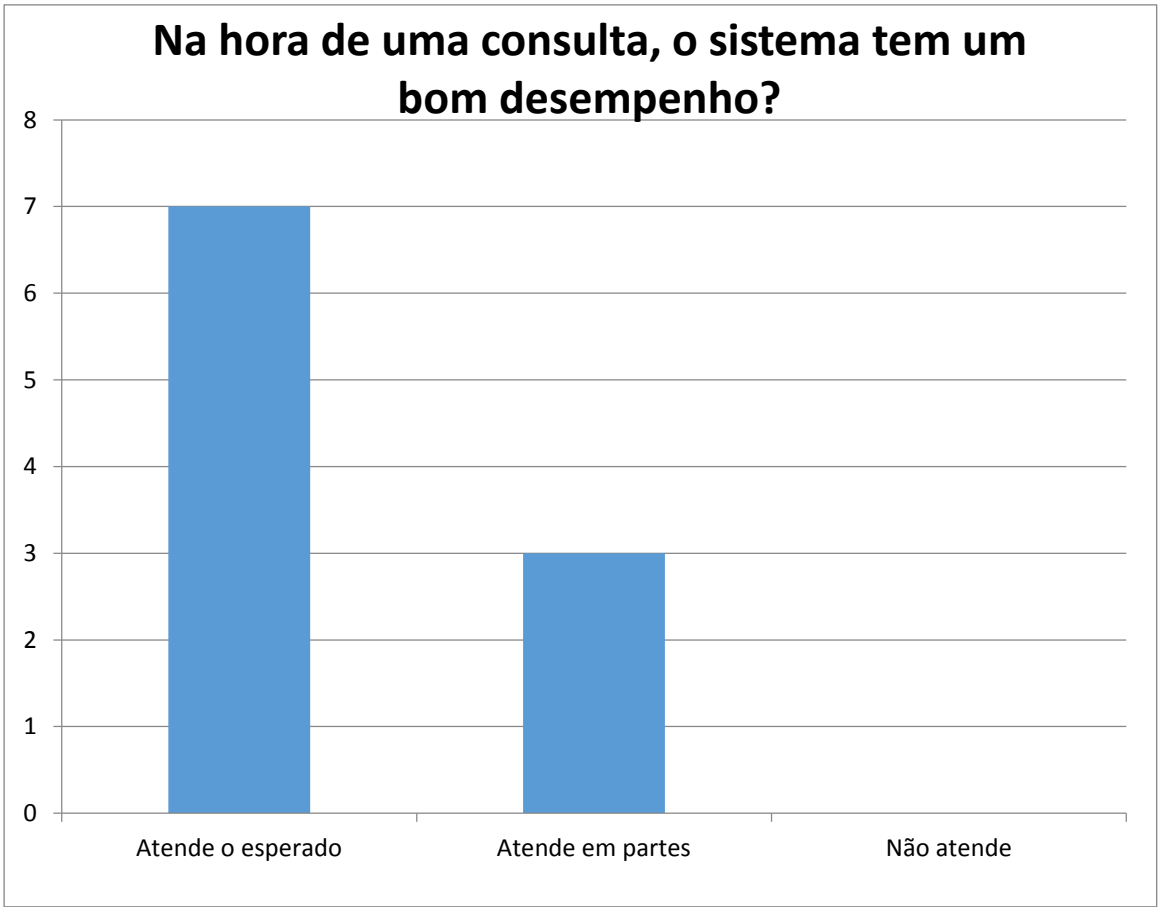
Figura 57 – Questão 7.



Fonte: Autor, 2014.

Referente ao desempenho da aplicação, a questão 7 da entrevista, foi perguntado para o usuário se o sistema tem um bom desempenho na hora de inserir um novo documento. Levando em consideração que o teste feito pelo usuário foi realizado em uma máquina comum, e não em um servidor. De acordo com a figura 57, para 80% dos entrevistados o sistema atende em partes, e 20% acharam que o sistema atende o esperado em relação ao desempenho na inserção de um documento.

Figura 58 – Questão 8.



Fonte: Autor, 2014.

Referente também ao desempenho da aplicação, a questão 8 da entrevista, foi perguntado para o usuário se o sistema tem um bom desempenho quando é feito uma consulta. Levando em consideração que o teste feito pelo usuário foi realizado em uma máquina comum, e não em um servidor. De acordo com a figura 58, para 70% dos entrevistados o sistema atende o esperado, e 30% acharam que o sistema atende em partes.

Figura 59 – Questão 9.

Fonte: Autor, 2014.

A Figura 59 – questão 9, comprova que, para 90% dos entrevistados, o sistema atende o esperado em relação a interface do sistema, e 10% atende em partes.

5.6 – CONSIDERAÇÕES FINAIS

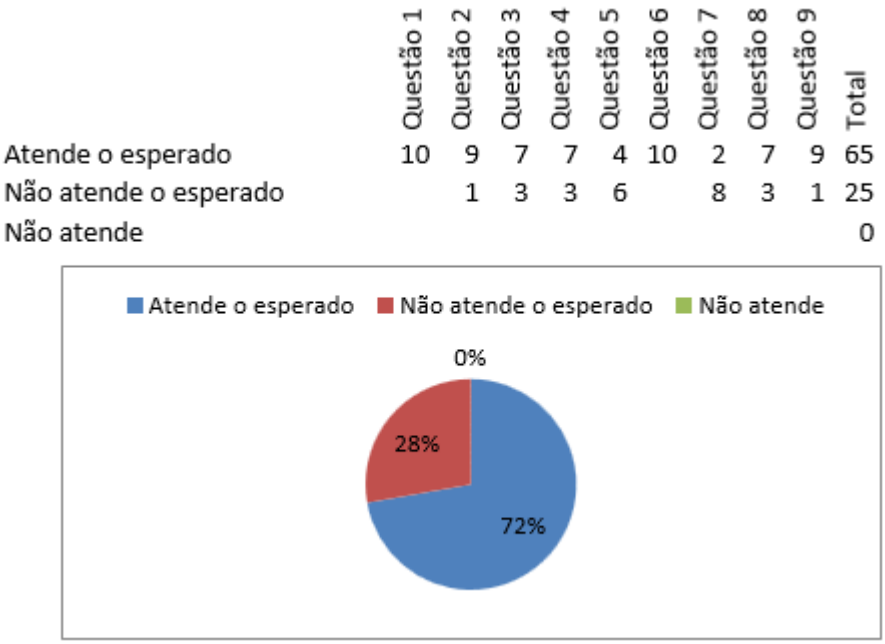
Esse capítulo teve como objetivo apresentar o protótipo do sistema parar solução do problema proposto, citando as tecnologias usadas e o esquema abordado para chegar a um resultado final. Também foi avaliado, através de um questionário, a solução proposta.

Realizando a classificação dos documentos é possível separar os documentos por conceitos, tendo como base uma base de conhecimento. Para o usuário que esta fazendo a pesquisa, a desambiguação do termo a ser pesquisado é uma grande vantagem, pois, ele vai

selecionar a categoria que se relaciona ao tipo de documento que ele esta procurando. Trazendo somente resultados relevantes a pesquisa.

Na figura 60, é apresentado o total de cada resposta para cada questão.

Figura 60 – Gráfico com o total de respostas por questão.



Fonte: Autor, 2014.

Observando o gráfico com o total das respostas por questão, é possível afirmar que o protótipo proposto teve um nível satisfatório de aceitação. Atendo o esperado na maioria das repostas.

6 – CONCLUSÕES E TRABALHOS FUTUROS

Nesse capitulo serão abordados os resultados obtidos com o desenvolvimento do protótipo e sugestões para a melhoria do sistema de recuperação de informação para trabalhos futuros. Também vão ser abordados os problemas encontrados na solução propostas e a satisfação com os resultados alcançados.

6.1 - CONCLUSÃO

Foi apresentado neste trabalho, conceitos sobre sistemas de recuperação de informação, tendo como principal objetivo um sistema de RI, que indexa documentos e realiza a anotação semântica baseado em uma base de conhecimentos.

Para avaliar o protótipo proposto, foi realizado um questionário com usuários de dois perfis diferentes, usuários comuns e usuários que trabalham com tecnologia. As opiniões dos usuários foram analisadas levando em conta os resultados obtidos com a busca, a facilidade de utilizar o sistema, o desempenho, e a separação do resultado por conceitos.

O maior problema encontrado foi a limitação da base de conhecimento utilizada no desenvolvimento, mas foi delimitado no trabalho que não era a intenção criar uma ontologia ou uma base de conhecimentos, fica como sugestão para trabalhos futuros.

Com relação aos problemas apresentados no inicio deste trabalho, a proposta da solução, e os resultados obtidos através da pesquisa, pode-se concluir que a proposta é válida, podendo ser aplicada em outros cenários, basta utilizar uma base de conhecimento que atenda as necessidades.

Com base nos resultados da pesquisa, pode-se concluir que o sistema desenvolvido atende, completamente, à maior parte dos objetivos apresentados. O sistema recupera corretamente um documento inserido na aplicação e separa o resultado por conceitos ligado ao documento. Contudo, o algoritmo para realizar a indexação e a anotação semântica pode ser otimizado para melhor o desempenho da aplicação.

6.2 – TRABALHOS FUTUROS

Com relação aos trabalhos futuros, pode-se dizer que a anotação semântica pode ser melhorada, trazendo resultados mais relacionados aos documentos indexados. Essa melhoria pode ser feita através da otimização do algoritmo que realiza indexação e a anotação semântica e também da criação de uma base de conhecimento que consiga mapear todos os conceitos e termos relacionados aos documentos a serem inseridos na aplicação.

Uma boa solução para melhorar o sistema de recuperação de informação, é a criação de uma ontologia. Essa ontologia representará os conceitos e termos relacionados aos termos de um documento.

A ontologia deve ser feita, preferencialmente, por especialistas do assunto do contexto da base de documentos a serem anotadas semanticamente. Assim, garantindo a relação correta dos documentos com os conceitos.

REFERÊNCIAS

ALMEIDA, F. N. MARTINEZ, V. H. F. TELLES, P. G. **Algoritmos e heurísticas para comparações exata e aproximidade de sequências**. XVV Congresso da sociedade brasileira de computação. 2005.

ARAUJO, M. FERREIRA, M. A. G. V. **Educação a Distância e a Web Semântica**: Modelagem Ontológica de Materiais e Objetos de Aprendizagem para a Plataforma CoL. 2003. 178 f. Dissertação (Doutorado em Engenharia de Computação e Sistemas Digitais) Universidade de São Paulo, São Paulo.

ARANTES, L. O. **Documentação semântica no apoio à integração de dados e rastreabilidade.** 2010. 284 f. Dissertação (Mestrado em informática) Universidade Federal do Espirito Santo.

BAEZA-YATES, Ricardo A.; RIBEIRO-NETO, Berthier. **Modern information retrieval**. New York: ACM Press, 1999.

BELL, Donald. **UML basics**: An introduction to the Unified Modeling Language, [Acesso em 2013 Nov 13] Disponível em: < http://www.ibm.com/developerworks/rational/library/769.html >.

BOOCH, G; RUMBAUGH, J; JACOBSON, I. **UML: guia do usuário**. 2 ed. São Paulo; Elsevier, 2005,.479 p.

BLANCO, Roi; et al.**Repeatable and reliable semantic search evaluation.** ELSEVIER. 2013.

BRÄSCER, M. A ambiguidade na recuperação da informação. Data Gama Zero, Rio de Janeiro: Revista de Ciência da Informação, v.3, n. 1, Fev. 2002. [Acesso em 2014 Maio 28] Disponível em: < http://www.dgz.org.br/fev02/Art_05.htm#R8>.

CALEGARI, D. T et al. **Programação Web com Jsp, Servlets e J2EE**, 2004. [Acesso em 2014 Maio 12] Disponível em < http://www.icmc.usp.br/pessoas/mello/livro-j2ee.pdf>.

CARDOSO, O. N. P.. **Recuperação de Informação.** INFOCOMP Journal of Computer Science.v. 2, n. 1, p. 33-38, 2002

CECI, F. **Um modelo semiautomático para a construção e manutenção de ontologias a partir de bases de documentos não estruturados** . 2010. 129 f. Dissertação (Mestrado em Engenharia e Gestão do Conhecimento) - Universidade Federal de Santa Catarina, Florianópolis, 2010.

CECI, F. et al. **Towards a semi-automatic approach for ontology maintenance**. In: Contecsi International Conference On Information Systems and Technology Management. 7., 2010, São Paulo. Anais… São Paulo: USP, 2010.

CECI, F. Woszezenki, C. R. Gonçalves, A. L. **O uso de anotações semânticas e ontologias para a classificação de documentos.** In: International Journal of Knowlodge Engineering and Management (IJKEM). 14p, 2014, Florianópolis, Santa Catarina.

CERVO, A. L.; BERVIAN, P. A. **Metodologia Científica**. São Paulo: Prentice Hall, 2002. P.65.

CAVALCANTI, C. R. **Indexação e tesauro: metodologia e técnica**. Brasília, ABDF, 1978.

DAVIES, John. FENSEL, Dieter.VAN HARMELEN, Frank.**Towords the semantic web**: **Ontology-driven knowledge management.** Sannon, Irland: 2003.

EBECKEN, Nelson F. F.; LOPES, Maria Celia S.; COSTA, Myrian C. A. Mineração de texto. In: REZENDE, Solange O. (Coord.). **Sistemas inteligentes** São Paulo: Manole, 2005.

ELSSEN, S. M. Z, STEIN, B. POTTHAST, M. **The Suffix Tree Document model Revisited**. Graz, Austria, 2005.

ELLER, Markus Pereira. **Anotações Semânticas de Fontes de Dados Heterogêneas: Um Estudo de Caso com a Ferramenta Smore**. 2008. 89 f. Trabalho de Conclusão de Curso (Graduação em Sistemas da Informação) – Universidade Federal de Santa Catarina, Florianópolis, 2008.

FACHIN, Odília. **Fundamentos de metodologia** . 3. ed. São Paulo: Saraiva, 2001.

FERNEDA, Edbert. **Recuperação de informação**: Análise sobre a contribuição da Ciência da Computação para a Ciência da Informação. 2003. Tese (Doutorado em Ciências da Comunicação) – Universidade de São Paulo, São Paulo, 2003.

FOWLER, Martin. **UML essencial** : um breve guia para a linguagem-padrão de modelagem de objetos. Trad. João Tortello, 3 ed. Porto Alegre: Bookman, 2005.

FURLAN, José David. **Modelagem de objetos através da UML** . São Paulo: Makron Books, 1998.

GALLIANO, A. G. **O método científico: teoria e prática**. São Paulo: Harbra, 1986.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. São Paulo: Atlas, 1996.

GALHO, Thaís Silva; MORAES, Silva Maria Wanderley. **Categorização Automática de Documentos de Texto Utilizando Lógica Difusa** . 2003. 75 f. Monografia (Bacharelado em Ciência da Computação) – Universidade Luterana do Brasil, Gravataí, 2003.

GONÇALVES, L. A. **Um modelo de descoberta de conhecimento baseado na correlação de elementos textuais e expansão vetorial aplicado à engenharia e gestão do conhecimento**. 2006. 196f. Dissertação (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

GOOGLE – **Knowledge Graphs.** [Acesso em 2014 Maio 11], Disponivel em < http://www.google.com/insidesearch/features/search/knowledge.html>.

GUHA, R. MCCOOL, R. MILLER, E. **Semantic Search**. Proceedings of the 12th international conference on World Wide Web. ACM, 2003. p. 700-709.

GUIMARÃES, G; ROSSINI, T; MEDEIROS, R; SILVA, G; SILVA, George. **Utilizando ICONIX no desenvolvimento de aplicações delphi**. In: CONGRESSO DE PESQUISA E INOVAÇÃO DA REDE NORTE NORDESTE DE EDUCAÇÃO TECNOLÓGICA, 2., 2007, João Pessoa. Anais eletrônicos

Internet World Stats - **Internet Usage Statistics.** [Acesso em 2013 Agosto 23] Disponível em http://www.internetworldstats.com/stats.htm.

Java (2014): Página oficial da plataforma Java, [Acesso em 2014 Maio 11] Disponível em <http://www.java.com/pt_BR/about/>.

JESUS, B. M J. : **Um instrumento de representação de conhecimento em sistemas de recuperação de informação**. XII Seminário Nacional de Bibliotecas Universitárias, Recife - 2002.

JUNG, C. F. Metodologia Científica: Ênfase em pesquisa e tecnologia, 4ed., 395 p., [Acesso em 2013 Agosto 15] Disponível em < http://www.unisc.br/portal/upload/com_arquivo/metodologia_cientifica....pdf >

KAMIENSKI, C. A. **Orientação a objetos**. Centro federal de educação tecnológica da paraíba diretoria de ensino. João Pessoa, 1996.

KOZAREVA, Zornitsa. **Bootstrapping named entity recognition with automatically generated gazetteer lists**. In: CONFERENCE OF THE EUROPEAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (EACL), 11., 2006, Trento, Italy. Proceedings… Trento, Italy, Abril 2006.

KORFHAGE, Robert. R. Information Storage and Retrieval. New York: John Wiley & Sons, Inc., 1997.

LEITE, Maria Angelica de Andrade. **Modelo Fuzzy para Recuperação de Informação Utilizando Múltiplas Ontologias Relacionadas**. 2009. 164 f. Tese (Doutorado) – Universidade Estadual de Campinas, Campinas, 2009.

LUCENE PROJECT (2014). Página do projeto Lucene. [Acesso em 2014 Maio 11] Disponível em < http://lucene.apache.org/core/>.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHUTZE, Hinrich. Introduction to information retrieval. New York: Cambridge University Press, 2008.

MAIA, J. A. **Construindo softwares com qualidade e rapidez usando ICONIX**, 2005.

MARTIN, F. **UML Essencial**: Um Breve Guia para Linguagem Padrão de Modelagem de objetos. São Paulo: Bookman, 2005. 160 p.

MEYER, Bertrand; COLEMAN, Derek; ARNOLD, Patrick; BODOFF, Stephanie; DOLLIN, Chris; GILCHRIST, Helena; HAYES, Fiona; JEREMAES, Paul. **Desenvolvimento Orientado a Objetos**: o método fusion. Rio de Janeiro: Campus, 1996.

MONTEIRO, Luís. **A internet como meio de comunicação**: Possibilidades e limitações. XXIV Congresso Brasileiro da comunicação – Campo Grande, Mato Grosso do Sul, 2011

MORDANI, R. Java Servlet Specification. Version 3.0. Sun Microsystems, Santa Clara, California. Dezembro 2009 [Acesso em 2014 Maio 02] Disponivel em: < http://download.oracle.com/otn-pub/jcp/servlet-3.0-fr-eval-oth-JSpec/servlet-3_0-final-spec.pdf?AuthParam=1400726923_37137b6366ac528bd5e41c180a2691af>

NÉDELLEC, C.; NAZARENKO, A**. Ontologies and information extraction**. LIPN Internal Report, 2005.

POPOV, B. et al. **Towards Semantic Web Information Extraction**. The Second International Semantic Web Conference (ISWC2003), Florida, USA, 2003. [Acesso em 2013 Setembro 15] Disponível em: <http://gate.ac.uk/conferences/iswc2003/proceedings/popov.pdf>.

POSTEGRESQL. Site Oficial, 2014 . [Acesso em 2014 Maio 12]. Disponível em: <https://wiki.postgresql.org/wiki/Introdu%C3%A7%C3%A3o_e_Hist%C3%B3rico>

REEVE, L., HAN, H.. Survey of semantic annotation platforms. ACMSymposium on Applied Computing (SAC), 2005.

ROSENBERG, Doug; STEPHENS, Matt; COPE, Mark Collins-. Agile Development with ICONIX Process : People, Process, and Pragmatism. New York: Apress , 2005.

SALTON, G. FOX, E. WU, H. **Extended Boolean Information Retrieval.** Comuns. ACM 26, 11 (Novembro, 1983)

SCHEREIBER, J. N. C; et al. GIRS – Genetic Information Retrieval System, XXVIII Encontro Nacional de Engenharia de Produção (ENEGEP), 2008.

SHAW, I.S e SIMÕES, M.G (1999), **Controle e Modelagem fuzzy**. São Paulo: Edgard Blucher.

SILVA, Alberto; VIDEIRA, Carlos; **UML Metodologias e Ferramentas CASE**, Centro Atlântico, 2001

SILVA, E. L; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4ª edição. Santa Catarina: Florianópolis, 2005: 139p.

SOUZA, Renato R. **Sistemas de Recuperação de Informação e Mecanismos de Busca na web: panorama atual e tendências**. Perspect. Ciênc. Inf., Belo Horizonte, v.11 n.2, p. 161-173, maio./ago. 2006.

SOMMERVILLE, Ian. **Engenharia de Software .** 8ª ed. São Paulo: Pearson Addison-Wesley, 2007.

SPARX SYSTEMS (2014): Site oficial da ferramenta, 2014. [Acesso em 2014 Maio 12] Disponível em <http://www.sparxsystems.com/products/index.html>.

___: Vocabulário Controlado do Governo Eletrônico, 2011. [Acesso em 2013 Outubro 14]. Disponível em < http://vocab.e.gov.br/2011/03/vcge#ciencia-informacao-comunicacao>

KUMA, T. H. et al**. Recuperação Semântica de Objetos de Aprendizagem**: Uma Abordagem Baseada em Tesauros de Propósito Genérico. Anais Simpósio Brasileiro de Informática na Educação (SBIE). 2008.

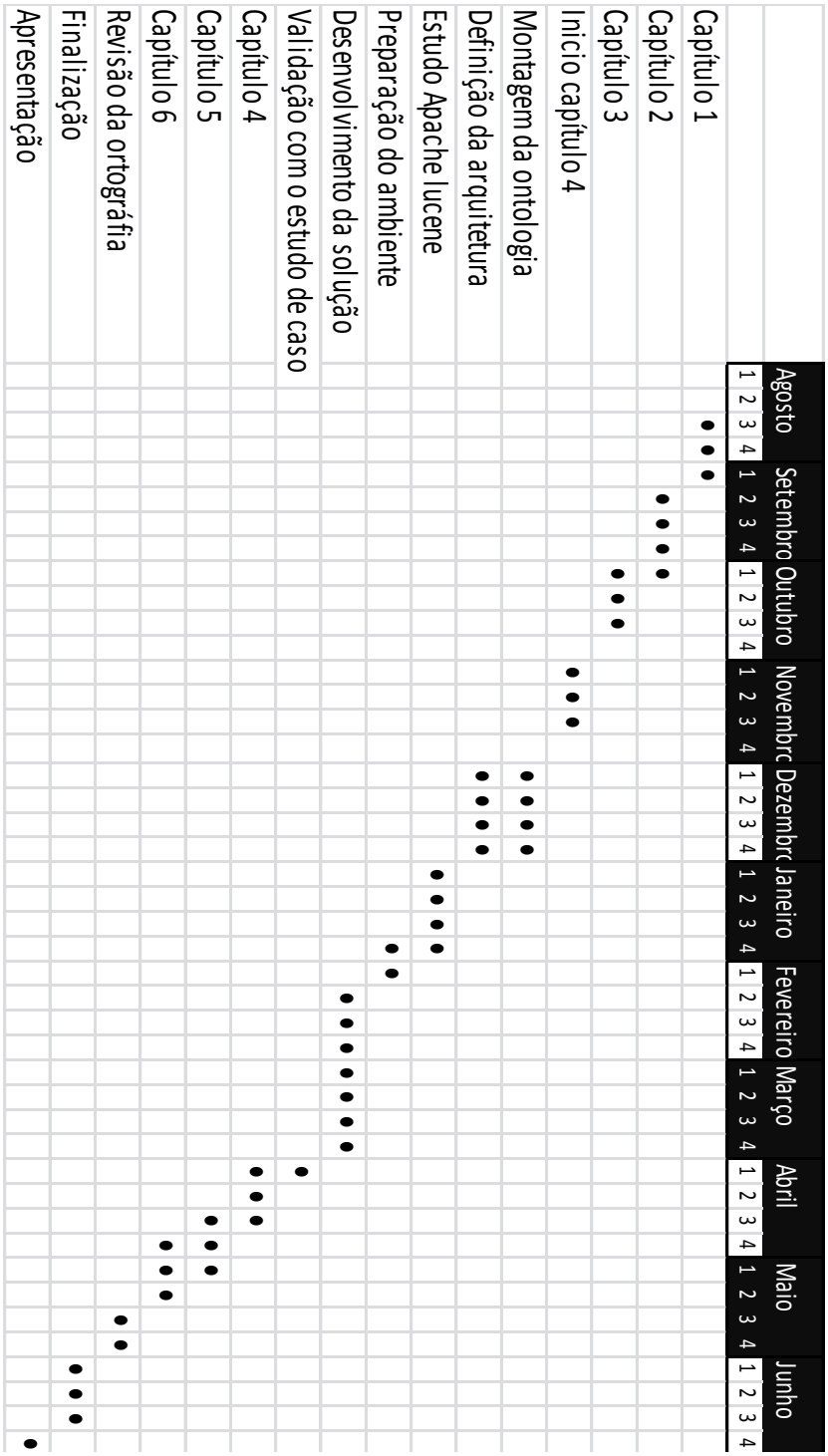
TANG, Lijun. CHEN, Xu. **The Study of Semantic Retrieval Base on the Ontology of Teaching Management**. ELSEVIER. 2011.

XU, J. & CROFT, B. W. **Query Expansion Using Local and Global Document analysis**, Procceding of the 19th ACM Conference on Research and Development in Information Retrieval (SIGIR), 1996.

WIVES, Leandro Krug. **Tecnologias de Descoberta de Conhecimento em Textos Aplicadas à Inteligência Competitiva**. 2002. 116 f. Dissertação (Pós Graduação em Computação)-Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

APÊNDICES

APÊNDICE A – CRONOGRAMA



APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO PARA RECUPERAÇÃO DE INFORMAÇÃO

Por favor, leia com atenção e responder de acordo com o que você visualizou na navegação da ferramenta de busca e escolher qual opção se adequa melhor a sua opinião. Cada pergunta contém 3 alternativas, atende o esperado, atende em partes e não antedê. Não há alternativa certa ou errada, sinta-se tranquilo para responder as questões de forma mais honesta possível.

1) **O sistema efetua o registro de novos documentos?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

2) **É feito a anotação semântica?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

3) **O sistema é de fácil navegação?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

4) **O sistema recupera documentos relevantes à pesquisa?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

5) **O sistema separa a pesquisa por conceitos?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

6) **Em uma pesquisa considera simples, o sistema recupera documentos relevantes?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

7) **O sistema tem um bom desempenho na hora de inserir um novo documento na aplicação?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

8) **Na hora de uma consulta, o sistema tem um bom desempenho?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.

9) **A interface é amigável e intuitiva?**
   A. Atende o esperado.
   B. Atende em partes.
   C. Não atende.