



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**WESLEY GONÇALVES ANIBAL**

**SISTEMA PARA ITENS ACHADOS E PERDIDOS USANDO  
FERRAMENTA DE COMPARAÇÃO DE TEXTO – *Me Acher***

**CAMPINA GRANDE - PB**

**2021**

**WESLEY GONÇALVES ANIBAL**

**SISTEMA PARA ITENS ACHADOS E PERDIDOS USANDO  
FERRAMENTA DE COMPARAÇÃO DE TEXTO – *Me Acher***

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em Ciência  
da Computação.**

**Orientador: Professor Dr. Matheus Gaudencio do Rêgo.**

**CAMPINA GRANDE - PB**

**2021**



A597s Anibal, Wesley Gonçalves.  
Sistema para itens achados e perdidos usando  
ferramenta de comparação de texto - Me Acher. / Wesley  
Gonçalves Anibal. - 2021.

10 f.

Orientador: Prof. Dr. Matheus Gaudencio do Rêgo  
Trabalho de Conclusão de Curso - Artigo (Curso de  
Bacharelado em Ciência da Computação) - Universidade  
Federal de Campina Grande; Centro de Engenharia Elétrica  
e Informática.

1. Sistema de achados e perdidos. 2. Comparação de  
texto - ferramenta. 3. Me Acher - ferramenta de  
comparação de texto. 4. Sistema para objetos perdidos.  
6. Biblioteca Apache Lucene. 7. Gerenciamento de objetos  
perdidos - UFPA I. Rêgo, Matheus Gaudencio do. II.  
Título.

CDU:004(045)

**Elaboração da Ficha Catalográfica:**

Johnny Rodrigues Barbosa  
Bibliotecário-Documentalista  
CRB-15/626

**WESLEY GONÇALVES ANIBAL**

**SISTEMA PARA ITENS ACHADOS E PERDIDOS USANDO  
FERRAMENTA DE COMPARAÇÃO DE TEXTO – *Me Acher***

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em Ciência  
da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Matheus Gaudencio do Rêgo  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Marcus Salerno de Aquino  
Examinador – UASC/CEEI/UFCG**

**Professor Dr. Tiago Lima Massoni  
Examinador – UASC/CEEI/UFCG**

**Trabalho aprovado em: 24 de maio de 2021.**

**CAMPINA GRANDE - PB**

# Sistema para itens achados e perdidos usando ferramenta de comparação de texto - Me Acher

Trabalho de Conclusão de  
Curso  
Wesley Gonçalves Anibal  
wesley.anibal@ccc.ufcg.edu.br  
Universidade Federal de  
Campina Grande  
Campina Grande, Paraíba,  
Brasil

## RESUMO

Atualmente um dos problemas, não só da UFCG, como também de diversos ambientes públicos, é o controle de itens perdidos. A ausência de um controle, gera mais trabalho aos funcionários da universidade, onde na maioria dos casos deixam o item guardado e esperam um parecer do dono confiando naqueles que se apresentam como donos. Este trabalho compreende o desenvolvimento de um sistema de achados e perdidos, denominado Me Acher, onde o usuário poderá cadastrar itens que foram encontrados e pesquisar por itens perdidos, onde o sistema realizará pareamento automático de ambas descrições, e retornará o item com maior probabilidade de ser o que está sendo procurado.

## 1. INTRODUÇÃO

Atualmente, na UFCG, não existe um controle de itens achados e perdidos, onde a dificuldade para encontrar um item se torna alta, pois além de depender de que o item seja encontrado, é preciso identificar o local onde o item será mantido, quem é o responsável que encontrou o item. Além disso, o responsável pelo item encontrado precisa fazer ainda alguma verificação de posse por parte da pessoa que se assume como dona do item perdido..

Não se tem exatamente em dados, os número de itens perdidos dentro dos domínios da UFCG, na qual não se sabe mensurar o tamanho do problema, porém é comum encontrar alunos a procura dos seus pertences perdidos nos principais pontos de aglomeração da universidade, como na praça de alimentação, em lanchonetes, na entrada principal, nos laboratórios e nos principais blocos da UFCG como o CAA. Alguns centros tentam remediar de alguma forma e auxiliar os alunos que perderam algo, a forma mais comum é divulgando itens encontrados em seus laboratórios ou nos arredores nas listas de email do curso, como por exemplo, no curso de Ciência da Computação, onde ao um item ser encontrado, o item é levado a secretaria, e as funcionárias da coordenação do curso, divulga o item na lista de email para todos os alunos de domínio CCC, onde possivelmente tenha a maior probabilidade de o dono ser encontrado.

Alguns dos principais problemas, além da falta de controle dos itens perdidos, se trata da índole de algumas pessoas, que infelizmente acabam se identificando como donos do material encontrado e furtando este item.

Levando em conta tudo que foi falado até então, utilizar de um software para controlar os itens perdidos e mais ações relacionadas, garante e oferece mais eficiência e segurança, além do fácil acesso a um site onde qualquer pessoa pode procurar itens perdidos, tendo o controle inclusive da identificação automática dos verdadeiros donos dos bens em questão.

Tendo em mente todo o problema, o presente trabalho tem como objetivo a criação do sistema Me Acher, onde o usuário poderá ter um experiência melhor ao procurar seu objeto perdido, e encontrar o real dono do bem. Para isso, este documento apresentará todas decisões tomadas e geridas, além de todos os problemas encontrados durante o desenvolvimento do sistema.

Este documento contém as seguintes seções: Arquitetura do sistema, dividido em Back-end, Front-end e Bibliotecas, Resultados e Experiência.

## 2. ARQUITETURA DO SISTEMA

Dentre vários tipos de arquitetura, a solução escolhida e adotada é a de cliente-servidor, uma das arquiteturas mais utilizadas atualmente. Esta arquitetura persiste em uma divisão onde o servidor tem como responsabilidade gerir toda lógica de negócio, todas escolhas, manipulação e persistência dos dados. O cliente por sua vez, se trata de uma interface, apresentável ao usuário, que vai se comunicar com o servidor, e onde o usuário poderá manipular, obter informações, criar informações. O cliente pode se tratar de qualquer interface onde o usuário poderá ter acesso ao servidor, como um aplicativo no celular, página web, aplicativo no próprio PC.

## 2.1 Back-end

O back-end do Me Acher, tido com servidor da arquitetura cliente-servidor, devido ao mapeamento de entidades que é preciso ter em relação ao sistema, se baseia no paradigma de orientação a objetos, atualmente o paradigma mais utilizado no mundo devido sua aplicabilidade ao mundo real, onde se torna relativamente fácil transformar entidades em itens do sistemas, abstraídos entre seus diferentes tipos.

Levando em conta que o paradigma orientado a objetos, a tecnologia escolhida para o desenvolvimento do back-end foi o Spring boot. O framework Spring Framework<sup>1</sup>, baseado em Java, conta com diversos tipos de configurações, e possui serviços, implantação e compatibilidade com as tecnologias mais avançadas atualmente de cache, contêineres, micros serviços, orquestradores, etc. Por esse motivo, ainda hoje, é um dos frameworks mais utilizados no mundo, por diversas empresas de ponta.

O Spring Framework utiliza-se do modelo de componentização, dividido em: Services, Controllers e Repositories, além das Entities e Configurations, oferecendo assim o necessário para representação arquitetural do sistema.

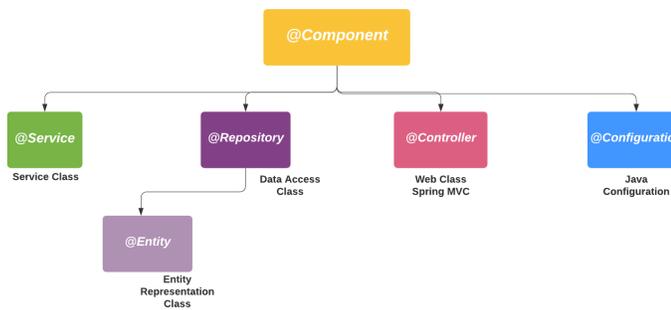


Figura 1 - Representação de componentes do Spring boot

A Figura 1 é uma representação dos componentes existentes no Spring Framework, com toda sua hierarquia de componentes, estabelecendo uma arquitetura muito funcional.

Para persistência de dados foi utilizado o MySQL<sup>2</sup>. O MySQL é possivelmente o serviço de banco-de-dados mais utilizado do mundo, utilizado por diversas empresas gigantes, como o YouTube, PayPal, Netflix, Twitter, Uber, entre outras, devido a sua comunidade e sua segurança na persistência e manutenção de dados. O MySQL, como o próprio nome já diz, se utiliza da linguagem SQL e uma arquitetura de dados relacional.

Foi criado a entidade Item, com os atributos: Descrição, Descrição abstrata, Data e Score, onde Descrição é a descrição do objeto perdido, de forma detalhada, Descrição abstrata é uma descrição feita de forma superficial, data é o dia no qual o item foi encontrado, podendo ficar vazia caso o usuário não lembre do dia exato, e score, que se trata do valor do cálculo de score de semelhança, que será explicada na Subseção 2.3 mais detalhadamente.

<sup>1</sup> <https://spring.io/>

<sup>2</sup> <https://www.mysql.com/>

Além disso, foi criado, respectivamente, o controller de item, contendo um endpoint para cada operação de item (create, update, get e delete). Além da criação do service de item, seguindo a arquitetura proposta pelo Spring. De forma a abstrair abstrair tipos entre o usuário e a API, foi utilizado o padrão de projeto conhecido como Data Transfer Object (DTO) onde é criado um objeto com valores simples, para servir de comunicação entre o usuário e a api, e dentro da API esse objeto é convertido num objeto do tipo Item, onde poderá ser realizado as operações, e persistência no banco de dados.

## 2.2 Front-end

O front-end foi construído utilizando o framework Angular 2+<sup>3</sup>. O Angular, vem com um conjunto de bibliotecas, formando um framework completo e que permite a construção de aplicações avançadas.

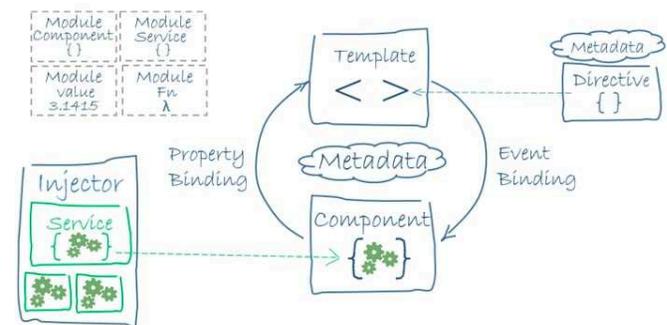


Figura 2 - Representação de componentes e interação do Angular<sup>4</sup>

O Angular se utiliza também das arquiteturas de componentes, como descrita na Figura 2 sendo dividida em: Service, Component, Template, Directive. Além de diversas bibliotecas que servirão para qualquer situação, como por exemplo a Angular Route, que serve para a navegação de componentes, permitindo o envio de informações entre essas entidades.

## 2.3 Bibliotecas

Também foram usadas algumas bibliotecas, nas quais formam e auxiliam na execução do sistema. A principal biblioteca é a Apache Lucene<sup>5</sup> de código aberto, utilizada para indexação e pesquisa, onde torna-se possível fazer buscas de qualquer dado relativo a textos, inclusive em diferentes tipos de arquivos como PDF. Bastante robusta, o Lucene é usado em diversos aplicativos, que se utilizam de pesquisas, como: Eclipse, Jira, Wikipédia, SourceForge, dentre outros.

O Lucene se utiliza de uma análise de documentos, os convertendo em palavras com pesos toda extração é feita por um analyser, que pode operar em diferentes linguagens, como o português ou japonês por exemplo. No MeAcher foi utilizado o **BrazilianAnalyser**, onde são analisados todas as palavras da língua portuguesa brasileira (PT/BR), inclusive incluindo palavras nas quais não possui valor algum para indexação, conhecida como "Stopwords", como por exemplo conectores: a, o, e, com, entre outras palavras, nas quais

<sup>3</sup> <https://angular.io/>

<sup>4</sup> Retirada do site <https://www.geekhunter.com.br/>

<sup>5</sup> <https://lucene.apache.org/>

podem ser ignoradas, ao gerar uma indexação de comparação a outros textos.

No nosso caso, indexamos o atributo chamado Descrição, da classe Item, onde será através dela que o sistema baseia sua busca e consultas de pertencimento do objeto encontrado.

O Lucene se baseia na teoria de recuperação de informação, na qual se utiliza técnicas probabilísticas para se chegar a uma pontuação em cada documento. Toda lógica do sistema MeAcher é concentrada no poder da ferramenta de indexação, onde se utiliza da probabilidade para fazer a busca do item perdido. Quando algum usuário cadastrar um item, ele o descreve, gerando termos com alta ou baixa relevância, e sendo salvo no banco de dados. Logo após isso, algum usuário que teve seu item perdido, faz uma busca ao seu item, descrevendo o item, gerando novamente termos com suas respectivas relevâncias, a sua descrição é comparada aos itens cadastrados no sistema, comparando com as descrições dos usuários que encontraram itens, sendo retornado aqueles itens que possuem os maiores scores.

Documento, segundo o Lucene, trata-se de uma coleção de campos, que serão indexados, transformados em tokens. No Me Acher é somente o campo Descrição.

O cálculo do score de Lucene se utiliza da combinação de alguns modelos probabilísticos, o Modelo de Espaço Vetorial(VSM) e o Modelo Booleano. O VSM se trata de um modelo em que quanto mais vezes um termo da consulta aparece em um documento em relação ao número de vezes que o termo aparece em todos os documentos da coleção, o documento se torna assim mais relevante para a consulta. No caso do modelo booleano, ele é usado para primeiro restringir os documentos que precisam ser pontuados, com base no uso da lógica booleana, na especificação da consulta. O Lucene também adiciona alguns recursos e refinamentos a este modelo para oferecer suporte à pesquisa booleana e difusa, mas basicamente esta continua sendo um sistema baseado em VSM.

O Score da comparação também se baseia em outra técnica probabilística, buscando dar relevância a cada termo encontrado conforme sua frequência em cada documento, essa estratégia conhecida como TF-IDF (term frequency-inverse document frequency), o valor gerado pelo TF-IDF sobre uma palavra, aumenta conforme o número de ocorrências da palavra no documento também aumenta, porém esse valor também é afetado devido a certas palavras serem mais comuns do que outras, verificando sua ocorrência em outros documentos. O valor da frequência em um documento chamado TF (Term Frequency), verifica quantas vezes aquele termo aparece no documento, por exemplo, um texto, qualquer, caso o usuário pesquise uma palavra aleatória, “vaca” por exemplo, TF se baseia em quantas vezes essa palavra aparece no texto. Porém, existe um problema, certas palavras se tornam mais comuns que outras, por exemplo a palavra “uma” ou qualquer outro conector, porém sua relevância é baixa, para isso existe o IDF (Inverse Document Frequency), ela aumenta o peso dos termos que ocorrem, mais raramente, conforme diminui o peso daqueles termos que ocorrem mais comumente.

```
{
  "id": 9,
  "descricao": "chaveiro de violão marrom com 3 chaves, sendo duas delas da marca gold",
  "data": "2020-01-19",
  "score": 0.74536043,
  "percentage": 82.28571428571429,
  "descricaoGenerica": "chaveiro de violão marrom",
  "possivelItem": true
}
```

**Figura 3** - Representação dos valores retornados pela api.

Na Figura 3, foi pesquisado a palavra “chaveiro”, sendo retornado o único item com valor chaveiro, e o score de valor 0.74536043. Ao ser pesquisado, a API ignora palavras que não adicionam nenhum valor à pesquisa, como as “StopWords” como mencionado mais acima.

### 3. RESULTADOS

Neste módulo será demonstrado todo o fluxo do sistema, além dos resultados obtidos com o teste do algoritmo de comparação.

Inicialmente iremos falar do fluxo de cadastro de item encontrado, de forma bem simples, o usuário no qual encontrou o item, irá acessar a tela de cadastro, onde informações como: Item, Descrição resumida, descrição detalhada e a data, será solicitada, logo após o preenchimento de todos os campos, o usuário cadastra o item, como segue a Figura 4.

**Figura 4** - Tela de cadastro do item encontrado.

Logo após o cadastro o item é inserido no banco de dados, podendo agora ser pesquisado por outros usuários.

Seguindo, na Figura 5, o próximo fluxo será o de pesquisa, onde é o nosso ponto focal do sistema. Um usuário no qual teve seu item perdido, irá acessar a página de pesquisa, onde informações como: Descrição e a possível data que perdeu o item (o campo não é obrigatório, devido o usuário não saber o dia que perdeu o item), deverão ser preenchidas.

**Figura 5** - Tela de pesquisa do item perdido.

Após efetuar a pesquisa, será retornado ao usuário a listagens de possíveis itens que podem ser o seu além de informações sobre a porcentagem daquele item ser o seu, no qual foi perdido, filtrando também pelo dia que foi passado. A sua descrição foi comparada a descrição básica, genérica, onde posteriormente, o mesmo usuário, deverá descrever de forma ainda mais detalhada o seu item perdido, onde assim devemos diminuir o uso indevido do sistema, passando por dois níveis, e possivelmente mais a frente em novas features, por mais níveis de avaliação. Na Figura 6 foi pesquisada a palavra “chave”, sendo retornado aqueles itens que tem algo relacionado a palavra, além de mostrar seu score, mostrando uma barra, apresentando a possibilidade do item ser o pesquisado pelo usuário, como apresentado na Figura 6, logo abaixo.

Score	Descrição	%
0.74536043	chaveiro de violão marrom com 3 chaves, sendo duas delas da marca gold	<div style="width: 100%; height: 10px; background-color: #ff0000;"></div>

**Figura 6** - Lista de itens retornada ao usuário.

Essas são todas as telas do sistema atualmente, todas informações são cruciais para o entendimento e o funcionamento do algoritmo, que será o próximo ponto no qual mostraremos os resultados.

#### 4. ESTUDO DE CASO SOBRE O SISTEMA

Para fim de testes, coletamos informações a usuários sobre 3 itens, mostrando uma imagem e pedindo para que descrevessem o objeto como se o estivessem tentando encontrá-lo. Os itens são, um chaveiro, uma pulseira e um documento de identidade.

**Tabela 1** - Itens cadastrados e suas respectivas descrições.

Item	Descrição
Chaveiro	Chaveiro de violão marrom com 3 chaves, sendo duas delas da marca gold
Pulseira	Pulseira de aço, pequeno, com alguns arranhões e marcas de uso
Documento de Identidade	Identidade no nome de Maria Eduarda da Silva, natural de Vitória-ES

Inicialmente, para o item Chaveiro, ele foi cadastrado com a seguinte descrição: chaveiro de violão marrom com 3 chaves, sendo duas delas da marca gold. Uma das pesquisas se trata de um teste básico, onde, ao pesquisar algo relacionado ao chaveiro, uma simples palavra, o sistema só retorne o item relacionado ao chaveiro, não retornando a pulseira ou documento de identidade.

**Figura 7** - Pesquisa pela palavra “Chave”.

Como apresentado na Figura 7, foi pesquisada a palavra “Chave”, onde foi somente retornado o item relacionado à palavra, além de seu score de 0.74536043. Ao inserirmos mais informações, por

exemplo, a cor do chaveiro, vemos que existe um crescimento no valor do score, como vemos na Figura 8.

**Figura 8** - Pesquisa por “Chaveiro Marrom”.

O valor do score subiu para 1.054099, valor considerável para a pesquisa feita pelo usuário, relacionando a pesquisa ao item no qual foi retornado. Por sua vez, ao inserir outra cor, como por exemplo “chaveiro amarelo”, o valor cai consideravelmente, como podemos ver na Figura 9.

**Figura 9** - Pesquisa por “chaveiro amarelo”.

O valor do score caiu para 0.2073781 aproximadamente, mostrando que o item possui pouca relação ao que foi pesquisado e não é o procurado pelo usuário.

Ao mudarmos de item, agora pesquisando pela identidade, inserimos na pesquisa o valor exatamente igual à descrição do item, “identidade no nome de Maria Eduarda da Silva, natural de Vitória-ES”, o valor do score cresce validade que aquele item é o que o usuário está procurando, como mostrado na Figura 10.

**Figura 10** - Pesquisa pelo valor literal do objeto.

O valor do score sobe para 1.844 aproximadamente, pelo fato dos fatores únicos, relacionado ao item e também ao nome da pessoa e o local, que foram inseridos na pesquisa e na descrição do item, tendo um alto valor para a pesquisa.

#### 4.1 Resultado do caso de uso

Os resultados obtidos pelo experimento acima, foram resultados pontuais, para itens totalmente diferentes, onde se trata de um experimento do uso do sistema em si, sua usabilidade, não detalhando o cálculo do score.

O score é o ponto focal do nosso sistema, o score ideal é determinado pela descrição oferecida pelo usuário e também pelo número de itens cadastrados no sistema, aplicando a teoria relacionada ao score ao experimento realizado, o score terá uma boa

influência relativa ao TF, parte do cálculo que determina o peso do termo em sua frequência na descrição do próprio item, porém, não terá uma influência considerável relativa ao IDF, segunda parte do cálculo, IDF, que determina o peso do termo relativo ao número de itens cadastrados no sistema, afinal para o experimento realizado só existiam 3 itens cadastrados.

O experimento não permitiu determinar o score com mais precisão, sendo necessário um preenchimento maior da base, com diferentes itens e descrições para poder dar maior ênfase ao cálculo do score, aumentando sua qualidade e exatidão. Devido ao prazo curto para obtenção de mais itens, não foi possível a realização de um experimento mais abrangente e com maior número de itens.

## 5. EXPERIÊNCIA

### 5.1 Processo de desenvolvimento

No início do processo, foi definido o escopo e a linha na qual o sistema iria seguir, uma análise de requisitos, definindo assim quais funcionalidades o sistema iria possuir, além do seu algoritmo de comparação. Logo após, foi definido quais tecnologias seriam usadas, além de estudos sobre tecnologias de indexação que foram usados no algoritmo de comparação, pois, até então existe um desafio quanto a indexação de elementos textuais, inclusive existem bons estudos e artigos na academia quanto ao assunto, no qual me ajudaram a entender mais sobre o assunto, afinal o sistema iria se basear totalmente na qualidade do seu algoritmo, que por sua vez, se utiliza de assim como estudos da ciência da computação, utiliza de elementos probabilísticos complexos.

Inicialmente, foi criado o projeto base, somente com o trecho de indexação e comparação de textos, onde foram realizados pequenos testes, comparando trechos de textos aleatórios, independente da língua escolhida, amadurecendo aos poucos a API, para a cada passo fosse implementado mais regras de negócio. Semanalmente eram realizadas reuniões em busca de amadurecer ainda mais a API, e gradualmente o, até então, protótipo foi evoluindo com mais funcionalidades e regras, em paralelo a implementação de testes e correção de bugs.

### 5.2 Desafios

Durante todo desenvolvimento foram encontrados desafios, a cada implementação de uma nova feature era encontrado um problema, que deveria ser resolvido. Inicialmente o principal desafio era como iríamos indexar os elementos que seriam comparados, até então não se possuía nenhum conhecimento quanto a bibliotecas que fariam isso, e nenhum conhecimento probabilístico que ajudasse a criar um dado que seria o ponto focal do algoritmo de comparação, esse problema foi resolvido com a biblioteca do Apache Lucene, na qual além de indexar oferece dados em relação a comparação, além de como esses dados são gerados, como vemos na Figura 10.

```
2021-05-12 22:43:45.433 INFO 5810 --- [nio-80]
1.0162666 = fieldWeight in 0, product of:
1.4142135 = tf(freq=2.0), with freq of:
2.0 = termFreq=2.0
1.9162908 = idf(docFreq=1, maxDocs=5)
0.375 = fieldNorm(doc=0)
```

Figura 10 - Log da explicação do Score oferecida pelo Lucene.

Dados como TF (Term Frequency), IDF (Inverse Document Frequency), são exibidos, mostrando o todo cálculo até chegar no valor final do score.

O segundo problema foi relativo a nossa língua, a língua portuguesa é uma linguagem rica, cheia de regras, conectores, onde para nossa aplicação, elas não teriam nenhum valor, devendo ser ignorado. Os valores dessas palavras, como explicada acima chamadas de StopWords, apesar de ao final do cálculo eles possuírem um pequeno valor, devido ao cálculo de IDF, afetava significamente o valor final do algoritmo, sendo retornado itens que foram cadastrados com conectores, como por exemplo: da, o, a, de, com, entre outras. Após algumas buscas, foi encontrada uma única biblioteca, que era própria para a língua portuguesa, solucionando assim este problema.

O último problema, e possivelmente o mais importante, se trata do uso indevido do sistema, se beneficiando de alguma forma, furtando itens, se passando pelo real dono do item. Algumas estratégias foram pensadas, afinal, não era um problema simples a se resolver, após estudos, pesquisas e discussões, a estratégia escolhida, foi a de separar a pesquisa em duas etapas, a etapa inicial, onde o usuário que encontrou o item, iria cadastrar duas descrições do sistema, uma básica e outra mais detalhada, o segundo usuário ao pesquisar o item, a comparação inicial seria com a descrição básica do item, o item com maior valor de score, ou com match completo, iria para a segunda fase, onde o segundo usuário faria uma segunda pesquisa, oferecendo mais detalhes do item. Até então, essa foi a estratégia escolhida, podendo ainda ser burlada, porém, futuramente a evolução do sistema, novas estratégias serão tomadas.

### 5.3 Limitações e Trabalhos Futuros

O Me Acher se trata ainda de um sistema em sua fase inicial, sendo ainda baseado em seu algoritmo de comparação, é claro que ainda precisa de evolução em vários pontos. A aplicação ainda não possui a funcionalidade de cadastro e login, impossibilitando a identificação do usuário, essa se trata de uma feature que será implementada, onde o usuário deverá se identificar, oferecendo informações, além de uma imagem, que ajudará a sua identificação.

Além da funcionalidade citada acima, outra limitação do sistema, apesar de existir uma estratégia, se trata do furto de itens, com pessoas mal intencionadas. A estratégia por mais que em alguns casos funcione, ainda poderá ser burlada, fazendo com que se tenha um prejuízo ao real dono do item, além do sistema, onde na legislação do país, a empresa, dona do Me Acher, poderia responder por isso, como explicado abaixo, no trecho retirado do artigo no site <https://jus.com.br/>.

“Previsto no art. 186, art. 187 e art. 927 do Código Civil, a responsabilidade civil consiste no dever de reparar o dano por todo aquele que violar um dever jurídico, através de um ato ilícito, sendo atualmente aplicável à responsabilidade das redes sociais, como a responsabilidade de seus usuários.”

Para isso novas estratégias foram pensadas, uma delas será a opção de o usuário que encontrou o item, possa fazer perguntas sobre ele ao usuário que está tentando encontrar o item, perguntas detalhadas, onde somente o real dono saberia responder, a cada pergunta respondida, o valor final do score aumentaria ou diminuiria, conforme fosse respondida corretamente ou incorretamente. Além

disso, a organização produzirá um termo de uso e responsabilidade, blindando assim o sistema Me Acher de possíveis usos indevidos, implementado paralelamente a opção de banir tais usuários do sistema e oferecendo a opção de denúncia desses usuários.

## 6. AGRADECIMENTOS

Ao professor doutor Matheus Gaudencio do Rêgo, um profissional pelo qual tenho a mais sincera reverência e admiração, onde sempre foi e será sempre espelho para mim, não só no âmbito profissional e educacional, mas também no humanitário, sempre aberto a ouvir, ensinar, dar conselhos e ajudar, pessoa de uma inteligência absurda, sempre terá meu profundo respeito e sempre será exemplo para mim.

Aos meus colegas de curso, Yovany, Ariann, Matheus Melo, Tiago, Luan, Agnaldo, Damião, Dayvson, João Pedro, Rubens, Victor, Nilton, Itamar, Samuel, Hector e Eri, pessoas que sempre me ajudaram a enfrentar esse caminho que é o da graduação, serão amizades que levarei pelo resto da vida. Aos meus amigos de Olivedos que estudaram comigo na UFCG, alguns ainda batalhando e outros já formados, Roberto Junior, Bianca Costa, Wuallison Firmino, entre outros.

Aos motoristas, Zezinho, Maelson, Romário, Aluisio, Jotinha, que sempre estiveram abertos ajudar, em todos esses anos de idas e vindas a Campina Grande.

A minha família de EJC, meus pais, Ademária, Arquimedes, meus irmão, Vitória, Mateus, Amanda, Bruna, Ellen, Guilherme, Neném, Alcilânia, Nubia e Ruty, pessoas que aprendi a amar e gostar, onde sempre estiveram dispostos a ajudar, terão sempre meu carinho e respeito.

E finalmente, e mais importante agradecimento, aos meus pais, José Pascoal e Lucineide Anibal, as principais pessoas que sempre acreditaram em mim, que sempre me deram força para enfrentar todos os problemas, onde sempre deixaram de possuir algo, para me oferecerem, investindo e confiando em mim sempre. Também ao meu irmão, Christyan Gonçalves, onde sempre foi e sempre será espelho para mim, pessoa que sempre esteve aberto a conversar e dialogar, onde foi a pessoa que mais acreditou em mim todo esse tempo, companheiro em todos os momentos, bons e ruins.

## 7. CONCLUSÃO

Concluimos que a solução proposta pelo sistema é factível, ao ponto de comparação, apesar do pouco número de itens cadastrados no sistema, os resultados obtidos em todas as pesquisas foram positivos, demonstrando que o sistema pode oferecer qualidade na pesquisa de itens perdidos, se tratando ainda de está em sua fase inicial. Os resultados demonstraram que a pesquisa pontual a algum item teve uma alta qualidade, pois em nenhum momento algum item aleatório, ou então mesmos itens com descrições diferentes, teve um alto valor no score, conforme o usuário detalha ainda mais seu item, o score aumenta. Nesta fase inicial se teve um número de itens cadastrados, o volume destes itens tende a aumentar conforme o uso do sistema, fazendo que o score se torne o “ideal” ao passar do tempo.

Os resultados dos primeiros usuários a utilizarem o Me Acher terá algum efeito no seu score, fazendo com que esse valor tenha sua

variância alta, então será recomendado a pesquisa após o cadastro de um certo número itens, fazendo com que se aumente a qualidade do score.

## 8. REFERÊNCIAS

- [1] *VMware, Inc. or its affiliates.* <https://spring.io/>, acessado em 28 de fevereiro de 2021.
- [2] *Oracle Corporation and/or its affiliates* <https://www.mysql.com/>, acessado em 28 de fevereiro de 2021.
- [3] *Super-powered by Google* <https://angular.io/> acessado em 5 de janeiro de 2021.
- [4] *The Apache Software Foundation* <https://lucene.apache.org/>, acessado em 6 de janeiro de 2020.
- [5] *Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.*
- [6] *Juliana Morigi , Responsabilidade civil das redes sociais,* <https://jus.com.br/artigos/57644/responsabilidade-civil-das-redes-sociais>, acessado em 12 de maio de 2021.