UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE FORMAÇÃO DE OFICIAIS BM

FELIPE DO CARMO SOUSA

DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA O CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO DO CURSO DE FORMAÇÃO DE OFICIAIS BM

São Luís
2020

FELIPE DO CARMO SOUSA

DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA O CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO DO CURSO DE FORMAÇÃO DE OFICIAIS BM

Monografia apresentada ao Curso de Formação de Oficiais – Bombeiro Militar da Universidade Estadual do Maranhão para o grau de Bacharel em Segurança Pública e do Trabalho.

Orientador: Prof. Dr. Mauro Sérgio Silva Pinto

São Luís
2020

Sousa, Felipe do Carmo.
Desenvolvimento de uma ferramenta computacional para o cálculo do coeficiente classificatório do curso de Formação de Oficiais BM / Felipe do Carmo Sousa. – São Luís, 2020.

102 f.

Monografia (Graduação) – Curso de Formação de Oficiais BM-MA, Universidade Estadual do Maranhão, 2020.

Orientador: Prof. Dr. Mauro Sérgio Silva Pinto.

1.Automatização. 2.Classificação dos cadetes. 3.Ferramenta computacional. I.Título.

CDU: 355.235:004.9

Elaborado por Giselle Frazão Tavares – CRB 13/665

FELIPE DO CARMO SOUSA

DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA O CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO DO CURSO DE FORMAÇÃO DE OFICIAIS BM

Monografia apresentada ao Curso de Formação de Oficiais BM da Universidade Estadual do Maranhão para o grau de Bacharel em Segurança Pública e do Trabalho.

Aprovado em: 27/ 07/ 2020

BANCA EXAMINADORA

Prof. Dr. Mauro Sérgio Silva Pinto (Orientador)
Doutor em Engenharia de Eletricidade
Universidade Estadual do Maranhão

Ana Paula Ferreira Costa
Mestra em Engenharia da Computação e Sistemas

Prof. Me. Pedro Brandão Neto
Mestre em Ciência da Computação
Universidade Estadual do Maranhão

Laurinalva Nívea F. de Melo Silva
MAJOR QOCBM
MF. 1690569

Laurinalva Nívea Ferreira de Melo Silva – Major QOCBM
Especialista em Gestão de Segurança Pública
Corpo de Bombeiros Militar do Maranhão

Dedico este trabalho a Deus todo poderoso e a toda a minha família, pelo apoio, incentivo e compreensão.

AGRADECIMENTOS

A Deus, por todas as bençãos e condições concedidas, por me guiar nessa jornada e colocar pessoas especiais em minha vida para que eu pudesse chegar até aqui.

A minha família, Rubem Luís, Maria Das Neves, Vitor Samuel. A meus irmãos em Cristo, Márcio e Glaucia, e demais amigos. Por todo apoio, incentivo, investimento, cuidado e ajuda em todos os momentos. Por todo esforço empregado e dedicação para que pudesse atingir os objetivos e vencer os desafios da vida.

A UEMA, pela formação de excelente qualidade e por conceder um ambiente totalmente favorável e completo para que os objetivos da formação fossem alcançados.

A Academia de Bombeiros Militar, por, assim como a Universidade, também conceder um ambiente favorável para o desenvolvimento da pesquisa e também por possibilitar e disponibilizar horários para que a pesquisa fosse realizada, além de conceder informações necessárias para a pesquisa.

Ao meu orientador Professor e Coordenador do CFO BM na UEMA, Mauro Sérgio, pelo convite para realizar tal projeto, pelo apoio e por disponibilizar os recursos necessários para que o projeto fosse desenvolvido com êxito.

Ao Prof. Me. Alfredo Oliveira, arquiteto do Núcleo de Tecnologia da Informação (NTI) da UEMA, e ao Patrick, analista de sistema do NTI, por contribuírem com grande disponibilidade de informações e recursos. Agradeço por disponibilizarem o acesso ao sistema com o banco de dados do SIGUEMA (SIGAA), acesso extremamente necessário para que a pesquisa pudesse ser realizada.

"Uma máquina consegue fazer o trabalho de 50 homens ordinários. Nenhuma máquina consegue fazer o trabalho de um homem extraordinário".

Elbert Hubbard

RESUMO

O presente trabalho objetivou desenvolver um software para gerar a classificação dos cadetes do Curso de Formação de Oficiais a partir do cálculo da média das disciplinas do curso. Neste contexto, cabe ressaltar a importância da automatização desse processo, trazendo maior segurança, precisão e eficiência para a Academia de Bombeiros Militar "Josué Montello" e para os alunos. Assim, os passos galgados para chegar lá foram analisar a importância da automatização de processos manuais, bem como demonstrar a relevância da classificação dos cadetes, explicando suas dificuldades e problemas encontrados, e explicar método de cálculo para classificação dos cadetes, com o fim de desenvolver a ferramenta computacional que gera a classificação dos cadetes. Desse modo, foi utilizada a pesquisa exploratória e aplicada, além disso, foi usado como método para coleta de dados, o questionário e a entrevista, no que se refere a dados primários e pesquisa bibliográfica, no que se refere aos dados secundários, através do estudo levantado no referencial teórico sobre automatização de processos, cálculo da classificação dos cadetes e importância da classificação. Considerando que a classificação atualmente é feita de forma manual, surge a necessidade de automatizar um processo tão sério e importante. A partir dos estudos foi possível desenvolver uma ferramenta computacional que gera a classificação de forma automatizada, destacando a importância da ferramenta, pois é altamente eficiente e segura. Destaca-se a não intervenção humana no cálculo, sendo essa a parte crítica do processo. Entretanto, há limitações quanto a dependência da rede de internet da UEMA. Portanto, por meio de todo estudo realizado ficou demonstrado a importância de tal ferramenta para o os alunos do CFO BM e para a ABMJM. E é possível confirmar que o desenvolvimento da ferramenta computacional gera a classificação dos cadetes com precisão, eficiência e segurança.

Palavra-chave: Automatização. Classificação dos cadetes. Ferramenta computacional.

ABSTRACT

The present work aimed to develop a software to generate the classification of cadets in the Officer Training Course from the calculation of the average out of course subjects. In this context, it is worth emphasizing the importance of automating this process, bringing greater safety, precision and efficiency to the Military Firefighters Academy "Josué Montello" and to the students. Thus, the steps taken to get there were to analyze the importance of automating manual processes, as well as to demonstrate the relevance of cadets 'classification, explaining their difficulties and problems encountered by them, and also to explain the calculation method for cadets' classification, in order to develop the computational tool that generates those classifications. In this way, exploratory and applied research was used, in addition, it was used as a method for data collection, the questionnaire and the interview, with regard to primary data and bibliographic research, with regard to secondary data, through the study raised in the theoretical framework on process automation, calculation of cadet classification and importance of classification.Considering that the classification is currently done manually, there is a need to automate such a serious and important process. From the studies it was possible to develop a computational tool that generates the classification in an automated way, highlighting the importance of the tool, as it is highly efficient and safe. Noteworthy is the lack of human intervention in the calculation, which is the critical part of the process. However, there are limitations regarding the dependence on UEMA's internet network. Therefore, through all the study carried out, the importance of such a tool was demonstrated for the students of the CFO BM and for the ABMJM. And it is possible to confirm that the development of the computational tool generates the classification of cadets with precision, efficiency and safety.

Keyword: Automation. Classification of cadets. Computational tool.

LISTA DE ILUSTRAÇÕES

Figura 1- Fluxograma da ferramenta computacional 33
Figura 2 - JSON com dados dos alunos 35
Figura 3- Classe de conexão com o Banco do SIGUEMA 35
Figura 4 - Método para preenchimento de aluno 36
Figura 5- Método para preencher disciplinas 37
Figura 6 - Método que calcula a média do semestre 38
Figura 7 - Método que calcula a média do ano letivo 38
Figura 8 - Método que calcula a média final 39
Figura 9- Método que gera o ".pdf" com a média do ano 40
Figura 10 - Banco de dados com os alunos 41
Figura 11 - Banco de dados com as disciplinas 41
Figura 12 - Banco de dados com média calculada do 1º Ano 42
Figura 13 - Banco de dados com a Média final média calculada 42
Figura 14 - Classificação 1º ANO 43
Figura 15 - Classificação Final 43
Gráfico 1 - A qual círculo hierárquico você pertence? 44
Gráfico 2 – Na sua opinião, o processo de cálculo da antiguidade seria mais eficiente e seguro se fosse realizado sem a intervenção humana? 44
Gráfico 3 – Sobre processos judiciais, acerca deste tema, em sua opinião, essa demanda diminuiria? 45

LISTA DE SIGLAS

ABMJM – Academia de Bombeiros Militar “Josué Montello”

BM – Bombeiro Militar

CBMMA – Corpo de Bombeiros Militar do Maranhão

CFO – Curso de Formação de Oficiais

DETOP – Departamento Técnico Operacional

NTI – Núcleo de Tecnologia da Informação

POO – Programação Orientada a Objetos

QOCBM – Quadro de Oficiais Combatentes Bombeiro Militar

SIGAA – Sistema Integrado de Gestão de Atividades Acadêmicas

SIGUEMA – Sistema Integrado de Gestão de Atividades Acadêmicas da UEMA

UEMA – Universidade Estadual do Maranhão

URL – Local Uniforme de Recursos (Uniform Resource Locator)

SUMÁRIO

1 INTRODUÇÃO 12
1.1 Objetivos 15
1.1.1 Objetivo Geral 15
1.1.2 Objetivos Específicos 15
2 REFERENCIAL TÉORICO 16
2.1 Tecnologia da Informação na atualidade 16
2.2 Visão sobre a automatização de processos 17
2.3 O uso de linguagem de programação 19
2.3.1 A linguagem Java 20
2.4 Importância do banco de dados 23
3 CLASSIFICAÇÃO DOS CADETES 25
3.1 Importância da classificação dos cadetes 25
3.2 Método de cálculo 26
4 METODOLOGIA 29
4.1 Requisitos do software 32
4.1.1 Levantamento de requisitos 32
4.1.1.1 Requisitos funcionais 32
4.1.1.2 Requisitos não funcionais 33
4.2 Questionário sobre a importância da ferramenta computacional 34
4.3 Entrevista com o Oficial responsável pelo cálculo 35
5 ANÁLISE DOS RESULTADOS 36
5.1 Construção da ferramenta computacional 36
5.1.1 Projeto 36
5.1.2 Implementação 38
5.1.3 Teste 43
5.2 Análise e Discussão dos Resultados da Pesquisa de Levantamento de Demanda 46
5.3 Análise e Discussão da entrevista 49

6 CONSIDERAÇÕES FINAIS 51
7 REFERÊNCIAS 53
APÊNDICE A – CÓDIGO JAVA DA FERRAMENTA COMPUTACIONAL ... 57
APÊNDICE B – PESQUISA DE LEVANTAMENTO DE DEMANDA PARA VERIFICAR A OPINIÃO DAS PRAÇAS ESPECIAIS E OFICIAS SOBRE A FERRAMENTA COMPUTACIONAL PARA CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO 94
APÊNDICE C – TABULAÇÃO DE DADOS DA PESQUISA 95
ANEXO A - PORTARIA Nº 06/2017/ABMJM/CBMMA – PESOS DAS DISCIPLINAS DO CFO BM 97
ANEXO B - DECLARAÇÃO DE ORIGINALIDADE 102

1 INTRODUÇÃO

Tem-se vivido uma época onde há o crescimento da Tecnologia da Informação, que é um termo amplamente utilizado na computação para mencionar um conjunto de técnicas e recursos tecnológicos, nos quais possibilitam gerir e armazenar informações. Desse modo, o avanço tecnológico tem proporcionado a automatização de diversos processos por meio do uso da computação, o homem deixa de intervir diretamente e passa a ocupar uma posição de gerenciador, os processos se tornam mais rápidos e eficientes, em uma tentativa de minimizar os erros.

Toda essa automatização de processos se deve ao avanço da Ciência da Computação, a qual possibilitou que um profissional especializado nessa área seja capaz elaborar programas de informática. Os programas e aplicativos de computador, por mais simples que pareçam, são constituídos por um aglomerado de códigos, os quais não são tão simples sob o ponto de vista da interpretação de um leigo, por exemplo, já que muitas vezes a lógica não está tão evidente para o senso comum. Para entender a lógica de programação é necessário ter a capacidade de pensar de maneira congruente. Essa necessidade é usada para compreender um macroproblema de computação, dividi-lo em problemas menores, resolvê-los e consequentemente solucionar o problema maior. Desse modo, desenvolve-se algoritmos, que são ditos como uma sequência lógica de ações capaz de resolver um problema, ou seja, sistematizar o problema.

Embora uma grande parte das pessoas não perceba, utiliza-se algoritmo de maneira intuitiva e automática diariamente quando nós executamos tarefas comuns. Como exemplo tem-se uma receita de bolo, é preciso seguir corretamente os passos da receita a fim de que tudo dê certo. Os programas e aplicativos funcionam da mesma forma. Para existir comunicação entre o homem e a máquina foram desenvolvidas as linguagens de programação, as mais comuns e populares são: C, C#, Java, Python, Ruby, dentre outras.

Assim, uma das tecnologias a serem usadas é a linguagem de programação Java, na qual será abordada nesse trabalho. A mesma é uma linguagem orientada a objetos, que é um tipo de paradigma de programação, suportado por várias linguagens como Python, C#, Ruby, dentre outros. Esse paradigma se baseia fundamentalmente no conceito de objetos. Um objeto representa entidades do mundo real, como pessoas, conta correntes, caixa eletrônico, entre outros. Abstração é a palavra-chave para entender orientação a objetos, basicamente significa selecionar aspectos específicos de um problema a ser analisado, deixando de lado outros

aspectos. Abstrair é a representação de uma entidade do mundo real transformada no chamado objeto em Java.

Este trabalho busca melhorar os resultados do cálculo da classificação do curso de formação de oficiais BM, o seguinte problema de pesquisa foi levantado: o processo de classificação dos cadetes da ABMJM, conhecido como nota de antiguidade é feita, em parte de forma manual com o uso de uma tabela de Excel, propiciaria legitimidade na composição dos dados? Já que há uma possibilidade a erros de digitação, erros de atenção e mesmo erros do ponto de vista da vulnerabilidade dos dados.

Neste trabalho usa-se Java, pois é uma linguagem universal e com elementos que a tornam fácil de utilizar e foi usada para manipular dados que serão armazenados em um banco de dados. Os bancos de dados estão presentes em nossas vidas há muito tempo, a lista telefônica pode ser um exemplo de banco de dados. Em épocas passadas, as grandes empresas e instituições mantiveram seus dados em arquivos físicos, contudo com a ascensão dos computadores possibilitou-se o armazenamento digital. O gerenciador de banco de dados utilizado neste projeto trata-se do PostgreSQL, que conta com o padrão SQL, usado para criar bancos de dados relacionais.

Desse modo, o trabalho visa automatizar o processo de cálculo para obtenção do coeficiente classificatório por meio de uma aplicação ligada ao SIGUEMA (Sistema Integrado de Gestão de Atividades Acadêmicas da UEMA), uma vez que o processo é feito manualmente. A classificação é importante pois é com base nesta que se define a “antiguidade” entre os militares durante o curso e após a formação. Logo, com base na antiguidade que é efetuada as promoções dos oficiais para os postos superiores. Devido a complicação para gerar a classificação do Curso de Formação de Oficiais essa pesquisa se justifica através do desenvolvimento do software para calcular as notas dos cadetes em contribuição para o seu público alvo o benefício de oferecer segurança, precisão e eficiência no cálculo da classificação.

Assim, no âmbito metodológico esta pesquisa será de caráter exploratório e aplicado, fazendo uso de levantamento bibliográfico, onde serão analisados livros, periódicos especializados bem como informações técnicas adquiridas a partir da análise de outros estudos. Uma vez que, de acordo com Gil (2008), “a pesquisa exploratória proporciona maior familiaridade com o problema. Pode envolver levantamento bibliográfico, entrevistas com pessoas experientes no problema pesquisado”. A coleta de dados será realizada na Academia de Bombeiros Militar “Josué Montello” e Universidade Estadual do Maranhão, onde serão

feitas as observações como técnica de coleta de dados, obedecendo aos critérios de sistematização.

1.1 Objetivos

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo geral o desenvolvimento de um software capaz de calcular as notas finais dos cadetes com intuito de automatizar todo o processo de obtenção do coeficiente de antiguidade dos cadetes em forma de lista classificatória do Curso de Formação de Oficiais como forma de melhorar o processo sob o ponto de vista da segurança, precisão, eficiência e transparência no cálculo da classificação para a Academia de Bombeiros Militar "Josué Montello".

1.1.2 Objetivos Específicos

- Analisar a importância da automatização de processos manuais, sob o ponto de vista da produtividade e eficientização de processos;
- Demonstrar a relevância da classificação dos cadetes, explicando suas dificuldades e problemas encontrados;
- Explicar método de cálculo para classificação dos cadetes.
- Desenvolver a ferramenta computacional que gera a classificação dos cadetes.

2 REFERENCIAL TÉORICO

Nesta etapa do projeto serão apresentadas considerações acerca do tema proposto, com base nas teorias existentes. O referencial servirá de base para análise dos dados que serão coletados e estabelecerá as relações entre o problema de pesquisa os objetivos propostos e relatório final, com todos os resultados obtidos.

2.1 Tecnologia da Informação na atualidade

Segundo Freire de Sá (2015), Tecnologia da Informação é uso de recursos computacionais como forma de obter soluções para diversos problemas, com objetivo de manipular a informação, seja para produzi-la, armazená-la, transmiti-la, acessá-la e usá-la. Como bem nos assegura Furtado (2002), Tecnologia da Informação é a utilização de tecnologia, computacional ou não, com objetivo de trabalhar a informação dentro de uma determinada organização, ou seja, é desenvolvimento de sistemas de informação com uso de recursos tecnológicos.

Para Vicente Paludo (2010, p. 153), Tecnologia da Informação facilita determinar um conjunto de tecnologias, computacionais ou não, podem ser *hardware*, software, dados ou telecomunicações, para gerar, armazenar, processar ou difundir informação em diversas formas, por exemplo, imagens, voz etc.:

> Tecnologia da informação permite alavancar a modernização, aumentar a competitividade e apoiar o desenvolvimento econômico e social de todos os setores da economia: local, nacional e mundial - públicos e privados. As tecnologias abrangem todos os aspectos da vida humana. Elas provocam uma gama de mudanças que envolvem toda a estrutura da empresa, incluindo os recursos humanos, e não somente o ambiente tecnológico.

Como se pode verificar nessa citação, Tecnologia da Informação é aplicado na área de sistemas de informações. Evidentemente a aplicação pode ser utilizada para acessar, processar, armazenar e garantir segurança de informações com uso da tecnologia.

A tecnologia da Informação é utilizada através de processamento de informações, aonde há processamento de informação por meio de tecnologia, computacional ou não, dessa forma ela funciona como uma grande e complexa ferramenta por meio da manipulação de informação. Pode-se dizer que a Tecnologia da Informação se refere a área da informática que trata a informação, organização e a classificação de forma a permitir a tomada de decisão em prol de algum objetivo. Cita-se, como exemplo, a utilização de banco de dados online para armazenar informações de funcionários de uma empresa.

Ainda para Vicente Paludo (2010, p. 154):

> A tecnologia da informação funciona como um conjunto de todas as atividades e soluções providas por recursos de computação; designa o conjunto de recursos tecnológicos e computacionais para geração e uso da informação. Na verdade, as aplicações de TI são tantas, e estão ligadas às mais diversas áreas, que nenhuma consegue designá-la por completo. Nesse sentido, Tecnologia da informação permite a difusão da informação por meios das tecnologias, computacionais ou não.

Logo, é importante compreender que na Tecnologia da Informação é essencial a utilização dos recursos tecnológicos para trabalhar a informação, visto que o processamento da informação é parte integral de toda atividade humana e grande parte dos processos humanos são moldados pela tecnologia. Nesse sentido, vamos exemplificar Tecnologia da Informação como sendo um processo essencial para manipulação, transformação e difusão da informação para o ser humano, já que com o desenvolvimento tecnológico surge a necessidade de desenvolver um meio para facilitar a produção, acesso, armazenamento e segurança de informações.

2.2 Visão sobre a automatização de processos

Com o crescimento tecnológico exponencial e o desenvolvimento de máquinas, dentre elas, os computadores, se inicia a substituição do homem em diversos processos manuais que demandavam muito ou pouco esforço humano. Portanto, as máquinas eletrônicas proporcionam a automatização de processos que antes eram feitos de forma manual e como consequência exigiam maior tempo e maior quantidade de recursos (BUENO, 2014).

> Representado o tratamento automático da informação atualmente a informática está presente em todos os ramos de negócio fornecendo as empresas além da automatização de processos meios para os quais os administradores possam tomar melhor decisões empregando da melhor forma seus recursos e desta maneira obtendo sempre melhores resultados (BUENO, 2014, p. 7).

A automatização começa a alcançar diversos processos, o homem começa a desenvolver atividades de monitoramento e passa utilizar mais a mente, os processos se tornam mais eficientes, uma vez que os erros diminuem e a precisão aumenta, visto que as máquinas são programáveis e fazem, de forma ideal, aquilo que lhes é determinado, como diz Castells (1999, p. 51), “computadores, sistemas de comunicação, decodificação e programação genética são todos amplificadores e extensões da mente humana”.

Por conseguinte, as vantagens dessa automatização são visíveis, para Srour (1998), a Revolução Digital trouxe mudanças tanto em aspectos do trabalho manual profissional e qualificado, como no trabalho manual e repetitivo, no trabalho intelectual de execução de rotinas padronizadas e até no trabalho intelectual de concepção criativa. Ou seja, pode-se

verificar diminuição de tarefas repetitivas, integração de setores, padronização dos serviços e produtos e além disso, é possível analisar o desempenho dos processos, uma vez os mesmos são digitais.

Para Cavalcanti (2017), automatização de processos facilita modernizar processos manuais e dar a eles maior confiabilidade e eficiência, diminuindo erros de procedimento, erro de execução e garantindo a conclusão do processo dentro dos prazos estabelecidos:

> Ao automatizar, garantimos a execução do processo em sua forma, cumprindo-o fielmente da maneira como foi idealizado, mitigando e até eliminando as possibilidades de infração das regras de integridade do processo, seja por fraude, desconhecimento ou negligência. Um processo automatizado não está limitado a repassar atividades, ele também controla os prazos de execução de tarefas e excuta as ações predefinidas quando os prazos não são cumpridos, sem a necessidade de intervenção ou monitoramento (CAVALCANTI, 2017, p. 143).

Como se pode verificar nessa citação, a automatização de processos pode ser aplicada em diversas áreas onde há a presença de processos manuais repetitivos, desse modo as mais comuns são industrias e empresas. Evidentemente a aplicação pode ser utilizada para modernizar diversos processos manuais repetitivos de larga em escala, por meio de computadores ou máquinas.

Automatização funciona como um otimizador de processos na qual os processos deixam de ser feitos diretamente pelo homem e passam a ser realizados por máquinas ou computadores. Desse modo, os processos passam a ser gerenciados ou executados por meio da tecnologia e o homem passa a monitorar todo esse sistema. Logo, há diminuição de custos operacionais, aumento da qualidade, diminuição de erros diversos e eficiência na completude do ciclo processual. Cita-se, como exemplo, o uso de computadores para armazenar e processar informações de funcionários de uma empresa.

Ainda para Cavalcanti (2017, p. 143):

> A automação de um processo permite obter uma enorme quantidade e variedade de indicadores extremamente úteis para a gestão. São meios que permitem saber, por exemplo, quanto tempo o processo está levando, se parou, quanto tempo parou e por que parou. [...] Com isso, é possível consultar o estado atual e o histórico de cada processo ou atividade, identificando participantes, dados inseridos e resultados obtidos em cada uma das instâncias. A consolidação e análise dessas informações permitem a melhoria contínua desses processos, dando ao gestor as informações de que este necessita.

Portanto, automatização de processos permite alcançar resultados por meio do uso de tecnologia nos processos, dando a eles maior qualidade e eficiência. Logo, é importante compreender que todo processo pode ser automatizado, sejam processos administrativos como também processos industriais, o processo precisa ser analisado e após isso definir o tipo de tecnologia que irá realizá-lo. Nesse sentido, vamos exemplificar automatização de processos

como uma ferramenta que possibilita a utilização da tecnologia para transformar processos lentos e de baixa qualidade em processos rápidos e eficientes.

2.3 O uso de linguagem de programação

Em meio ao desenvolvimento tecnológico atual e como consequência a criação de máquinas com capacidade de processamento de informações, por exemplo os computadores, fez-se necessário criar um meio para que o homem se comunique com as máquinas e assim possa determinar suas funções e forma de operação. Diante disso foram desenvolvidas as linguagens de programação, com elas é possível se comunicar com os computadores e programá-los para que façam o que se deseja e se atinja um determinado fim.

Segundo Lorenzi e Lopes (2000), linguagem de programação é um conjunto de instruções que tem como objetivo utilizar recursos computacionais. Como bem nos assegura Reisswitz (2009), linguagem de programação é um conjunto de padrões textuais usados para designar instruções para um computador.

De acordo Pinto (2009), linguagem de programação facilita estabelecer comunicação entre o ser humano e uma máquina, para que se possa passar comandos e por fim serem executados pelo computador:

> [...] sabemos que uma linguagem de programação é um método totalmente padronizado que expressa algumas instruções (tarefas) a serem executadas pelo computador, tais como tipo de dados deve-se manipular, armazenar e quais as ações que devem ser executadas ao se atingir determinado objeto (PINTO, 2009, p. 26).

Como se pode verificar nessa citação, linguagem de programação é aplicado na área de informática, principalmente no desenvolvimento de programas computacionais. Evidentemente a aplicação pode ser utilizada para passar um conjunto de comandos para uma máquina a fim de que ela interprete e execute as instruções comunicadas.

A linguagem de programação funciona como um conjunto de elementos textuais composto de sentido e com uma sequência de comandos lógicos que serão interpretados por um computador. As linguagens são usadas com sistemas de escrita e de interpretação, chamados de compiladores. Cita-se, como exemplo, uso de uma linguagem de programação para registrar e gerenciar notas de alunos de uma universidade. Nesse sentido, linguagem de programação permite o desenvolvimento de programas capazes de direcionar recursos computacionais para atingir um determinado fim.

Logo, é importante compreender que existem diversos tipos de linguagem, mas todas elas são feitas com base em instruções lógicas capazes de definir uma série de

procedimentos para serem executados por computadores. Nesse sentido, vamos exemplificar linguagem de programação como o meio pelo qual o homem consegue estabelecer comunicação instrutiva com computadores, fazendo com que eles realizem atividades conforme a vontade humana.

2.3.1 A linguagem Java

Nesse contexto, destaca-se o Java, linguagem a qual utilizaremos nesta pesquisa para desenvolver a ferramenta computacional. Escolheu-se essa linguagem por conta das características que a aproximam do mundo real, como a orientação a objetos que será abordada dentro deste capítulo. O JAVA foi criado pela empresa SUN Microsystems (CAMPOS E CAMPOS, 2018, p. 8).

> Muitas das interessantes criações da SUN nasceram do SUN Microsystems Laboratories, incluindo a linguagem Java. A linguagem foi criada com o objetivo de ser compatível com diversas plataformas que já rodavam em equipamentos eletrônicos na época, ou seja, em 1991. O objeto seria algo como: "escrever uma vez e rode em qualquer lugar". Isso era bem melhor do que ter que escrever um programa diferente para cada uma das plataformas existentes ou que poderiam vir a existir. Isso mesmo, o Java não foi originalmente criado pensando em computadores, mas sim em dispositivos eletrônicos. O projeto seria o de uma TV interativa. Mas não deu muito certo. Muito avançado para a época. O projeto foi engavetado. [...] Em 1995 a empresa dona do maior navegador de Internet da época, a Netscape, fez uma parceria com a SUN para rodar applets Java dentro de seu navegador. Daí para a popularização da linguagem Java foi uma questão de tempo (pouco tempo) (CAMPOS E CAMPOS, 2018, p. 8).

Java é uma das linguagens mais utilizadas no mundo por não ser apenas uma linguagem de programação, mas sim uma plataforma de desenvolvimento (FURGERI, 2018). Ela se popularizou rapidamente, pois permite que o programador a escreva apenas uma vez e consiga reutilizar o código em diversos sistemas presentes em equipamentos, daí o jargão "escrever uma vez e rode em qualquer lugar".

Desta forma, a linguagem Java passa a estar presente em toda esfera tecnológica, principalmente em equipamentos eletrônicos, mas também em carros, eletrodomésticos e servidores corporativos (CAMPOS E CAMPOS, 2018). Dentre as diversas vantagens, de acordo com Mendes (2009), ainda se tem que a linguagem Java, uma vez que permite que o programador a trabalhe em diferentes sistemas computacionais, é considerada simples, além do fato que a preocupação com detalhes de infraestrutura não existe mais, com isso a produtividade e a eficiência dos programadores aumenta.

Por essas diversas características, dentre elas a orientação a objetos, é que esta linguagem é usada em diversos projetos, inclusive no sistema da SIGUEMA utilizado na UEMA e como consequência no desenvolvimento da ferramenta em questão.

> A programação orientada a objetos traz uma ideia muito interessante: a representação de cada elemento em termos de um objeto, ou classe. Esse tipo de representação procura aproximar o sistema que está sendo criado ao que é observado no mundo real, e um objeto contém características e ações, assim como vemos na realidade. Esse tipo de representação traz algumas vantagens muito interessantes para os desenvolvedores e também para o usuário da aplicação. A orientação a objetos permite que haja uma reutilização do código criado, diminuindo o tempo de desenvolvimento, bem como o número de linhas de código. Isso é possível devido ao fato de que as linguagens de programação orientada a objetos trazem representações muito claras de cada um dos elementos, e esses elementos normalmente não são interdependentes. Essa independência entre as partes do software é o que permite que esse código seja reutilizado em outros sistemas no futuro (GASPAROTTO, 2014).

Desta forma, a Programação Orientada a Objetos (também conhecida pela sua sigla POO) é um modelo de análise, projeto e programação de software baseado na composição e interação entre diversas unidades chamadas de objetos. Além, dos objetos há as classes, estas são o tipo de abordagem mais comum da POO, assim a classe determina o comportamento, os estados dos objetos e a relação entre diversos objetos. Ademais, há ainda outros atributos, elementos que compõem a orientação a objetos, os principais são a herança, encapsulamento e polimorfismo.

Assim, a herança permite legar características grupais de variadas classes, seja ela de base ou superclasse. Fundamentado, assim, as demais classes podem ser especificadas (RICARTE, 2001). Segundo Saraiva Junior (2017, p. 104), “a relação de herança se baseia no princípio de que toda a codificação mais genérica pode ser transmitida para as classes mais específicas. Esta relação é conhecida por meio do relacionamento é-um.”. Assim, não é necessário reimplementar características já descritas em uma classe mais ampla, por exemplo, mamíferos, desse modo é possível utilizar essa classe para criar objetos como cachorro e gato, ambos tem caraterísticas comuns, logo seus atributos podem ser herdados para criação de outros objetos semelhantes quanto uma determinada característica, nesse caso, serem mamíferos.

Também se tem o encapsulamento, que segundo Schildt (2015) encapsulamento em Java é um mecanismo de segurança que vincula o código e os dados por ele usado, dessa forma ambos ficam protegidos contra interferências externas. Como bem nos assegura Finegan e Liguori (2018), encapsulamento é a união, por meio do armazenamento, de métodos juntamente com os dados a ele relacionados em uma única classe. Encapsulamento em Java facilita proteger o código de uma interferência externa ou de um acesso indevido de outros objetos.

> O encapsulamento é uma das principais técnicas que define a programação orientada a objetos. Se trata de um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos pelo fato de esconder as propriedades, criando uma espécie de caixa preta. Para fazermos um paralelo com o que vemos no mundo real, temos o encapsulamento em outros elementos. Por exemplo, quando clicamos no botão ligar da televisão, não sabemos o que está acontecendo internamente. Podemos então dizer que os métodos que ligam a televisão estão encapsulados (GASPAROTTO, 2014).

Como se pode verificar nessa citação, encapsulamento é aplicado dentro dos códigos escritos em Java. Evidentemente que esse mecanismo pode ser utilizado para proteção de objetos que são criados com uso do Java.

Encapsulamento em Java funciona como uma caixa que abriga o código e os dados que são usados por ele, desse modo, é possível proteger o código e garantir que eles passem para a máquina as instruções desejadas. Cita-se, como exemplo, ao criarmos uma classe "conta" e atribuirmos a ela número, saldo, juros e vencimento e então ao criarmos um objeto "conta1" com essas características, esse objeto será encapsulado com as informações atribuídas a ele e outro objeto não poderá alterar as informações contida no objeto "conta1".

Logo, é importante compreender que o encapsulamento é um elemento essencial, uma vez que fornece segurança imprescindível como é visto no exemplo citado. Nesse sentido, vamos exemplificar o encapsulamento como um mecanismo de proteção de código para garantir maior segurança na execução das instruções passadas para a máquina.

E por fim, o polimorfismo, que segundo Moreira Neto (2009) polimorfismo é o mecanismo que permite a criação de objetos a partir de uma classe mais específica, subclasse, utilizando métodos de uma classe genérica, superclasse. E também, para Manzano e Junior (2014), polimorfismo é a capacidade de um método ser utilizado por objetos de tipos diferentes. E para Finegan e Liguori (2018) polimorfismo facilita que um objeto, que se deseja criar, possa adquirir a forma de um objeto já existente de tipo diferente.

> Outro ponto essencial na programação orientada a objetos é o chamado polimorfismo. Na natureza, vemos animais que são capazes de alterar sua forma conforme a necessidade, e é dessa ideia que vem o polimorfismo na orientação a objetos. Como sabemos, os objetos filhos herdam as características e ações de seus “ancestrais”. Entretanto, em alguns casos, é necessário que as ações para um mesmo método sejam diferentes. Em outras palavras, o polimorfismo consiste na alteração do funcionamento interno de um método herdado de um objeto pai. Como por exemplo, temos um objeto genérico “Eletrodoméstico”. Esse objeto possui um método, ou ação, “Ligar()”. Temos dois objetos, “Televisão” e “Geladeira”, que não irão ser ligados da mesma forma. Assim, precisamos, para cada uma das classes filhas, reescrever o método “Ligar()” (GASPAROTTO, 2014).

Como se pode ver nessa citação, polimorfismo é aplicado em linguagem orientada a objetos, neste caso, em códigos escritos e Java. Evidentemente a aplicação pode ser utilizada para permitir uso de um método em diferentes tipos de objetos.

O polimorfismo funciona como um mecanismo que permite que o método assuma várias formas e possa ser utilizado por diversos objetos, dessa forma o programador pode escrever códigos mais genéricos que podem ser reutilizados com facilidade. Cita-se, como exemplo, a possibilidade de criar um método que precise do objeto chamado "humano" e também pode-se criar outros dois objetos chamados "criança" e "adulto" a partir da classe "humano", desta forma "criança" e "adulto" vão herdar todas as funcionalidades de "humano".

Logo, é importante compreender que este recurso é altamente benéfico, uma vez que possibilita que o tempo de desenvolvimento seja menor, pois os elementos desenvolvidos no código podem ser reaproveitados. Nesse sentido, vamos exemplificar polimorfismo como um recurso de que permite a reutilização dos objetos criados para criação de novos objetos diferentes com as caraterísticas do anterior e com possibilidade de serem inseridas novas características.

2.4 Importância do banco de dados

Para que os computadores, sistemas e diversos equipamentos possam desenvolver suas funções é necessário guardar as informações que norteiam seu funcionamento e as informações para que os mesmos efetuem as atividades para as quais foram programados.

Existe uma extrema necessidade, por parte de sistemas computacionais, de armazenar informações, onde essas informações podem ser usadas para consulta, para geração de relatórios e também para possíveis acréscimos ou alterações. Desse modo, essa informação é guardada em banco de dados, onde são organizadas e mantidas para consulta ou manipulação (CAELUM, 2019).

Assim, com a necessidade de armazenar dados surgiu os bancos de dados e, anteriormente, os dados eram armazenados em fichas de papel organizadas em arquivos físicos por meio de pastas. Em meio a diversos documentos, o acesso às informações se tornava extremamente complexo e demorado, pois o processo manual é lento e exige uma verificação minuciosa. Como resultado, o homem começou a desenvolver bancos de dados virtuais, ou seja, os dados passaram a ser armazenados nos computadores, facilitando o acesso aos mesmos.

Deste modo, segundo Ramarkrishnan e Gehrke (2011) banco de dados pode ser conceituado como um agrupamento de dados descritivos sobre uma determinada organização. Além disso, para Date (2003), banco de dados também pode ser conceituado como um conjunto de dados permanentes que são usados por sistemas de uma determinada organização.

Para Elmasri e Navathe (2011, p. 3) banco de dados facilita a otimização de processos de uma organização, uma vez que propicia o registro de informações para que se possa manipular ou alterar esses dados.

> [...] banco de dados tem alguma fonte da qual o dado é derivado, algum grau de interação com eventos no mundo real e um público que está ativamente interessado em seu conteúdo. [...] Para que um banco de dados seja preciso e confiável o tempo todo, ele precisa ser um reflexo verdadeiro do minimundo que representa; [...] (ELMASRI e NAVATHE, 2011, p. 3).

Como se pode verificar nessa citação, banco de dados é aplicado é aplicado em todo sistema em que se faça necessário armazenamento de informação. Evidentemente a aplicação pode ser utilizada para guardar informações para que se possa acessá-las ou manipulá-las.

Um banco de dados é como uma caixa que permite que se guarde dados de forma permanente, além de guardá-los, também é possível ler, alterar ou gerar relatórios a partir dos dados, as possibilidades são diversas. Dessa forma, temos um local virtual, que ao se trabalhar qualquer informação é possível guardar o que foi feito. Cita-se, como exemplo, uso de sistemas de gerenciamento de bancos de dados para criar um banco de dados com nomes e notas de alunos de um curso superior em um sistema online.

Logo, é importante compreender que banco de dados como uma ferramenta essencial para tratamento de dados, uma vez que ao se manipular qualquer informação é possível armazená-la e, caso seja necessário, os dados podem ser utilizados novamente de diversas formas como já foi citado.

3 CLASSIFICAÇÃO DOS CADETES

3.1 Importância da classificação dos cadetes

A classificação dos cadetes da Academia de Bombeiros Militar "Josué Montello" é regulamentada pelas Normas Gerais de Ação do Corpo de Aluno, onde se lê o procedimento para o cálculo, como visto no capítulo anterior, e também diz que a classificação dos militares incorrerá na antiguidade (ABMJM, 2010).

Art. 166. II – A Classificação de que trata o presente artigo, a Classificação Intelectual Final incorrerá na definição da antiguidade do militar, no Estágio de Adaptação para Oficiais de Saúde, no Curso de Formação de Oficiais BM, no Curso de Formação de Sargentos BM e no Curso de Formação de Soldados BM. (ACADEMIA DE BOMBEIROS MILITAR JOSUÉ MONTELLO, 2010).

Como visto acima, a classificação realizada durante o período de curso, no presente caso o Curso de Formação de Oficias, é utilizada para definir a antiguidade dos cadetes, que é um elemento essencial na vida do militar.

Ademais, conforme a lei estadual 6513 de 30 de novembro de 1995, que dispõe sobre o Estatuto dos Policiais Militares da Polícia Militar do Maranhão, diz que:

Art. 21º – A antiguidade no posto ou graduação será regulada: [...] § 2º - Os Aspirantes-a-Oficial formados na Academia de Polícia Militar do Maranhão e por outras Academias, para efeito de antiguidade, considerar-se-ão as médias obtidas nos respectivos cursos e a mesma data de declaração de Aspirante-a-Oficial (MARANHÃO, 1995).

Como se pode ver na citação acima, as notas obtidas durante o curso de formação de oficias, que definem a classificação dos alunos, são usadas para os efeitos da antiguidade. Assim, aqueles que obtiverem notas maiores obterão uma antiguidade maior que os demais. A legislação, mesmo citando apenas a Academia de Polícia de Militar, é válida para Academia Militar do Corpo de Bombeiro, uma vez que em seu artigo 167º diz que essa legislação é aplicada aos militares integrantes do Corpo de Bombeiros Militar (MARANHÃO, 1995).

Desse modo, tem-se que a Antiguidade é meio pelo qual os militares definem a precedência hierárquica entre si. De acordo com a lei federal 6880 de 9 de dezembro de 1980, que dispõe sobre estatuto dos miliares, lê-se, no artigo 17º: "A precedência entre militares da ativa do mesmo grau hierárquico, ou correspondente, é assegurada pela antiguidade no posto ou graduação, salvo nos casos de precedência funcional estabelecida em lei" (BRASIL, 1980).

Além dessa legislação federal, há também a legislação estadual, na qual as normas da ABMJM estão diretamente subordinadas. De acordo com a lei estadual 6513, em seu artigo vigésimo diz que a precedência hierárquica é regulamentada por três quesitos, dentre eles está a antiguidade (MARANHÃO, 1995).

Além disso, a antiguidade é um critério para que o militar possa atingir postos mais altos, por meio da promoção. Como nos assegura a lei federal 6880, no seu artigo 60º, que "As promoções serão efetuadas pelos critérios de antiguidade, merecimento ou escolha, ou, ainda, por bravura e post mortem." (BRASIL, 1980). E também, a lei estadual 6513, no seu artigo 78º, diz que "As promoções serão efetuadas pelos critérios de antiguidade, merecimento, tempo de serviço, por bravura e 'post mortem', mediante ato do Governador do Estado para Oficiais e do Secretário de Estado da Segurança Pública para praças." (MARANHÃO, 1995).

Desse modo, devido a importância desse processo, em casos de erros diversos, erros de digitação por conta do processo repetitivo ou erros nas fórmulas de cálculo no Excel etc., na classificação final, acabam gerando demandas judiciais, solicitando correções e revisão do cálculo, o que acaba gerando transtornos.

Diante disso, pode se verificar a grande relevância da antiguidade para a vida do militar durante o Curso de Formação de Oficiais e, principalmente, para sua carreira após a formação. Dessa forma, a classificação dos cadetes define a antiguidade, que por sua vez definirá a precedência hierárquica durante o curso e após a formação, e também, influenciará para a promoção para postos superiores.

3.2 Método de cálculo

O método de cálculo da média final e classificação dos alunos do Curso de Formação de Oficiais Bombeiro Militar é regulamentado pelas Normas Gerais de Ação do Corpo de Alunos (NGA). Lê-se no Capítulo XV – Da Classificação, no art. 166 e 167:

> Art. 166 – Ao final do curso, estágio ou treinamento, os Alunos BM serão classificados por ordem decrescente de merecimento intelectual, que será calculado através da média ponderada das notas finais de aprovação em cada matéria, considerando-se os respectivos pesos, que constituirá uma nota final de curso com aproximação até milésimos. [...] IV – Ao final de cada etapa ou ano escolar, os militares serão classificados por ordem decrescente de merecimento intelectual, que será calculada através da média aritmética das notas finais de aprovação em cada etapa ou ano de curso, que constituirá uma nota final de curso com aproximação até milésimos. Em caso de empate, a precedência na classificação seguirá o previsto na Lei 6513, de 30 de novembro de 1995. Art. 167 - No Curso de Formação de Oficiais BM a classificação será estabelecida observando as seguintes situações: [...]Parágrafo único - Ao término de cada ano escolar, os militares permanecerão com a numeração inicial e terão sua precedência alterada pela ordem decrescente de merecimento intelectual, que será calculada através da média aritmética das notas finais de

> aprovação em cada etapa ou ano de curso, que constituirá uma nota final de curso com aproximação até milésimos. Em caso de empate, a precedência na classificação seguirá o previsto na Lei 6513, de 30 de novembro de 1995. (ACADEMIA DE BOMBEIROS MILITAR JOSUÉ MONTELLO, 2010).

Desse modo, durante o processo de cálculo da média de cada ano são calculadas duas médias parciais ponderadas, com as disciplinas e seus respectivos pesos, uma para cada semestre daquele ano. As médias ponderadas são calculadas pela razão entre o somatório do produto entre as médias das disciplinas e seus pesos e dividido pelo somatório dos pesos. Como o tempo total do curso é três anos, tem-se seis médias parciais ponderadas, onde a média aritmética das médias ponderadas do primeiro e segundo semestre compõe a primeira média do curso e assim sucessivamente com as demais.

Ao término de cada ano letivo, dois semestres letivos, a classificação é gerada e disponibilizada para o corpo de alunos, e assim define-se a precedência hierárquica entre os militares do curso. Portanto, ao longo dos três anos são geradas três médias, como consequência, três classificações, que também são chamadas de "Antiguidade". A partir dessas três médias se calcula uma média aritmética simples que é a média final do curso. Desse modo, temos três processos: 1º) Cálculo da média ponderada dos semestres, 2º) Cálculo da média dos anos e 3º) Cálculo da média final.

Na prática tem-se o seguinte cálculo, primeiro o cálculo de uma média ponderada:

$$MP1=\frac{(n1 \cdot p1) + (n2 \cdot p2) + \ldots + (ni \cdot pi)}{p1 + p2 + \ldots + pi} \quad (1)$$

Onde, *n1* e *n2* são as notas finais de cada disciplina, onde *p1* e *p2* são os pesos referentes as notas. Onde o índice *i* é a quantidade total de notas, que é igual a quantidade de pesos do semestre em questão, no caso o primeiro semestre.

Tem-se as médias anuais:

$$M_1=\frac{MP1+MP2}{2} \quad (2)$$

$$M_2=\frac{MP3+MP4}{2} \quad (3)$$

$$M_3=\frac{MP5+MP6}{2} \quad (4)$$

$$M_F=\frac{M_1+M_2+M_3}{3} \quad (5)$$

Onde *MP1, MP2, MP3, MP4, MP5* e *MP6* são as médias ponderadas dos respectivos seis semestres do curso. Onde M_1, M_2 e M_3 são as médias dos respectivos três anos do curso. Onde M_F é a média final do curso. As disciplinas estão registradas no Projeto Político

Pedagógico do Curso de Formação de Oficias Bombeiro Militar e os pesos conforme a portaria Nº 06/2017/ABMJM/CBMMA.

Até o presente ano, os cálculos e as classificações são feitos manualmente, por meio dos dados do SIGUEMA e também das atas de notas da ABMJM, daí as notas são inseridas no aplicativo Excel e são realizadas as manipulações dos dados para gerar as informações necessárias e o produto final, a classificação por ordem decrescente de nota.

4 METODOLOGIA

De acordo com Rampazzo (2009), pesquisa é um procedimento no qual se objetiva resolver um determinado problema por meio de métodos científicos. Por meio da pesquisa é possível descobrir novas leis, fatos ou dados em todas as áreas do conhecimento.

De acordo com Prodanov e Freitas (2013, p. 51), "a pesquisa aplicada objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos. Envolve verdades e interesses locais". A pesquisa aplicada é adequada a essa pesquisa devido ao objetivo de desenvolver um software para solucionar o problema apresentado.

Segundo Brocke e Roseman (2013), uma pesquisa exploratória é o estudo que busca examinar um problema no qual existem poucos estudos ou que nunca foi estudado. Além disso, a pesquisa descritiva, ainda para Brocke e Roseman (2013), é o estudo utilizado quando o pesquisador deseja descrever um determinado fato, para tanto detalhando o que são e a forma como ocorrem.

Considerando o tipo de pesquisa, observou-se que se trata de uma pesquisa exploratória devido a busca de fontes bibliográficas para que se possa obter maior familiaridade com o problema, a fim de entendê-lo mais profundamente. Além disso, pode-se classificá-la como descritiva também, uma vez que há descrição de fenômenos por meio de confrontos e correlacionamentos de hipóteses de diversos autores.

Prodanov e Freitas (2013, p. 70), destacam que: "Na pesquisa qualitativa, a pesquisa tem o ambiente como fonte direta dos dados. O pesquisador mantém contato direto com o ambiente como fonte direta e o objeto de estudo em questão, necessitando de um trabalho mais intensivo de campo[...]".

Conforme citado acima, quanto à abordagem do problema, a pesquisa é de caráter qualitativo, onde serão interpretados os fenômenos com base na pesquisa bibliográfica realizada. Desse modo, diante do problema levantado, tem-se o tipo de análise hipotético-dedutivo, uma vez que por meio das experimentações desenvolvidas, na ferramenta computacional, busca-se resolver o problema.

Prodanov e Fretias (2013, p. 80), por sua vez, afirmam que: "O levantamento bibliográfico é um apanhado geral sobre os principais documentos e trabalhos realizados a respeito do tema escolhido, abordados anteriormente por outros pesquisadores para obtenção de dados para a pesquisa."

Com base no uso de diversos livros, artigos científicos e artigos na internet definiu-se tal pesquisa como bibliográfica, ou seja, usa-se diversos textos de outros pesquisadores e

estudiosos para que se possa fundamentar e também entender o problema, e assim então identificar as possíveis soluções e neste caso, desenvolver a solução, a ferramenta computacional.

Além disso, ainda para Prodanov e Fretias (2013, p. 59):

> Pesquisa de campo é aquela utilizada com o objetivo de conseguir informações e/ou conhecimentos acerca de um problema para o qual procuramos uma resposta, ou de uma hipótese, que queiramos comprovar, ou, ainda, descobrir novos fenômenos ou as relações entre eles.

Em virtude do grande acesso ao público do processo de cálculo da classificação dos cadetes, além da grande importância do processo para os envolvidos, foi escolhido a técnica de pesquisa de campo e de entrevista para que se possa comprovar os argumentos já desenvolvidos no referencial teórico.

O instrumento para coleta de dados utilizado foi registros institucionais, uma vez que se realizou uma análise documental e em legislações internas do Corpo de Bombeiros Militar do Maranhão. E, utilizando livros, por meio de revisão literária, onde buscou-se aqueles que possuem maior relevância nos assuntos abordados e uma grande variedade de autores. Além disso, se usou um questionário e uma entrevista. No que se refere ao questionário, foi usado a fim de se obter uma melhor apreciação do conteúdo apresentado no trabalho, os dados serão captados através de questionário padronizado para sua tabulação. E, quanto a entrevista, esta foi realizada com a 1º Tenente QOCBM Gilzimary de Jesus Sobrinho Privado de Jesus Sobrinho Privado, oficial auxiliar da DETOP da ABMJM, responsável pelo cálculo do coeficiente e da produção da classificação dos cadetes por nota.

Prodanov e Freitas (2013, p. 102), assim conceituam:

> Os "dados", em uma pesquisa, referem-se a todas as informações das quais o pesquisador pode se servir nas diferentes etapas do trabalho. Existem aqueles já disponíveis, acessíveis mediante pesquisa bibliográfica e/ou documental. São chamados dados secundários por se tratarem de "dados de segunda-mão". Cumpre ressaltar que essa expressão não tem caráter pejorativo, apenas indica que são dados disponíveis e que não foram coletados especificamente para nosso trabalho em particular.

Como visto acima, a busca em dados secundários foi o alvo primordial desta pesquisa, o uso de diversos livros, artigos e outros documentos científicos ou não, foram a fonte de coleta de dados para que se pudesse realizar tal pesquisa, uma vez que se fez necessário especificar o assunto do tema e definir o problema de pesquisa. Além disso, fez-se uso de dados primários também, através de um questionário e de uma entrevista, considerando a importância a da opinião dos indivíduos diretamente ou indiretamente afetados por esse processo.

Esta pesquisa trabalhou com análise de um problema existente no Corpo de Bombeiros Militar do Maranhão, a partir de então buscou-se entender tal questão e então, por

meio de pesquisa bibliográfica para entender melhor o problema e se familiarizar com ele. Em seguida, buscou-se desenvolver uma ferramenta computacional que serve como solução para o problema, assim desenvolveu-se protótipos e fez-se diversos testes para que tal ferramenta possa ser aplicada.

Após tal procedimento descrito acima, utilizou-se a pesquisa de campo por meio de um questionário online, feito na plataforma do Google Formulários. O formulário foi disponibilizado para os oficiais e praças especiais do CBMMA. Por fim, foi realizada uma entrevista com a Oficial responsável, a 1º Tenente QOCBM Gilzimary de Jesus Sobrinho Privado, pelo cálculo da nota e por produzir a classificação dos cadetes em ordem decrescente de média.

O universo da pesquisa foi composto por toda a literatura relacionada ao tema da pesquisa, e encontrada nos bancos de dados Scielo (*Electronic Science Library Online*) e *Google Books*. Para a amostra, os livros e artigos foram selecionados dentre as variáveis de interesse, num total de 26 livros e artigos. A seleção foi feita a partir de livros, artigos e literatura e então, de acordo com o tema abordado foi selecionada a literatura que mais se adequava as questões levantadas. Foram incluídas apenas livros e artigos, pois apresentam maior confiabilidade, publicadas no período de 1995 até 2019, uma vez que se usou legislações vigentes, porém antigas e livros relevantes, que mesmo antigos, expressão conceitos que ainda são usados.

A pesquisa possui um grau de confiabilidade baseado na autoridade e relevância dos autores escolhidos. Uma vez que a apresentação dos resultados tem como base em livros considerados importantíssimos para o tema do estudo referido. É importante destacar como autores principais Vicente Paludo, Rubens Cavalcanti, André Campos e Arthur Campos e artigos do site DevMedia. Além disso, muitos outros autores estão presente para reforçar o pensamento dos autores principais. No capítulo I é possível perceber como o autor Vicente Paludo expressa seu conceito de forma clara, onde foi possível realizar um bom embasamento e credibilizar ainda mais o estudo, também Rubens Cavalcanti perpetua conceitos no capítulo II do estudo, bem como André Campos, Arthur Campos e DevMedia apresentaram embasamentos que possibilitaram argumentos claros e fortes para o tema da pesquisa.

Portanto, foram desenvolvidos protótipos da ferramenta computacional para solucionar o problema da geração da classificação dos cadetes do Curso de Formação de Oficias, com base em pesquisas bibliográficas pode-se entender melhor o problema e também a necessidade de o solucionar. Entretanto, ainda há questões judiciais sendo levantadas por conta de erros no processo, assim, deseja-se mitigar e até eliminar a maioria dos erros para que

as demandas judiciais diminuam. Além disso, por meio da pesquisa exploratória, buscou-se aprofundar o conhecimento sobre o referido problema, uma vez que este nunca havia sido abordado em outra literatura. Assim, foi possível descrever e analisar qualitativamente os dados obtidos, para daí solucionar a questão levantada por meio do desenvolvimento do *software.*

4.1 Requisitos do *software*

Para que se possa atender às necessidades do usuário, os requisitos expressam as características e limitações do *software* e são a parte mais complexa e propensa a erros do desenvolvimento de *software*. Requisitos são metas ou restrições estabelecidas pelos clientes e usuários do sistema e definem vários atributos do sistema. Desse modo, requisitos de software são aqueles relacionados aos atributos do sistema.

Como bem afirma Engholm Júnior (2010, p. 151):

> Podemos definir requisito como uma condição ou capacidade de um *software* que deve ser implementada por um sistema ou componente de sistema para se alcançar determinado fim. Todo projeto de software tem um conjunto de requisitos, definidos pelas necessidades e expectativas dos usuários que efetivamente utilizarão o mesmo, relacionado ao atendimento dos objetivos de negócio da empresa onde trabalham.

Como observado na citação acima, os requisitos descrevem o funcionamento do sistema e aquilo que se espera como resultado final. Ademais, para Martins (2007, p.208): "Os requisitos são características funcionais e não funcionais que o sistema precisa apresentar". Dessa forma, os requisitos foram divididos entre funcionais e não funcionais e serão apresentadas na seção adiante.

4.1.1 Levantamento de requisitos

4.1.1.1 Requisitos funcionais

Como já visto, os requisitos descrevem o funcionamento do sistema, desse modo eles mostram as funções e atividades do *software*, para tanto devem ser definidos de forma clara. Logo foi definido como requisitos funcionais os itens listados abaixo:

1. Selecionar ano de ingresso;
2. Selecionar ano letivo ou média final;
3. Efetuar cálculo;
4. Gerar arquivo ".pdf".

O primeiro requisito consiste na seleção do ano de ingresso da turma no Curso de Formação de Oficiais. Tal requisito foi definido, pois uma das principais formas de identificar o aluno dentro do sistema de notas da UEMA é pelo ano de ingresso.

Já o segundo requisito consiste na seleção do ano letivo para o qual se deseja calcular a classificação ou a média final do curso, ou seja, a antiguidade. Esse requisito foi definido, pois a classificação dos cadetes é gerada ao fim de cada ano letivo, ou seja, a cada dois semestres concluídos, a contar do ano de ingresso. E também, para gerar a média final do curso, que considera todos os anos cursados no cálculo.

O terceiro requisito foi definido para efetivação cálculo da média referente ao ano selecionado pelo usuário. Este requisito representa a ação principal do sistema, ao ser executado o software irá buscar os alunos do ano selecionado e irá realizar o cálculo.

O quarto requisito foi definido para produzir um documento que terá as informações desejadas, com extensão ".pdf". Ao ser executado, será gerado o arquivo e este conterá a classificação dos alunos por média em ordem decrescente.

Estes requisitos representam as ações da ferramenta e o que ela fará. Portanto, é fácil explicar às partes interessadas, de modo em que elas possam ter uma visão do funcionamento do *software*, e o desenvolvedor pode ter uma noção do que precisa ser feito durante o desenvolvimento do *software* para atender esses requisitos definidos.

4.1.1.2 Requisitos não funcionais

Aqui se especifica os requisitos não funcionais relacionados ao desenvolvimento do *software*. Desse modo, temos como requisitos:

1. A página será compatível com os sistemas operacionais do *Windows*;
2. A tecnologia utilizada é a linguagem JAVA;
3. O sistema será intuitivo para o usuário, fácil de aprender e fácil de usar;
4. O sistema será capaz de tratar exceções e será tolerante a falhas;
5. O sistema terá como tempo de espera no máximo 5 segundos;
6. Será utilizado o banco de dados, *PostgreSQL*;
7. A ferramenta estará disponível para acesso conforme disponibilidade de rede da UEMA.

O primeiro requisito trata da portabilidade, uma vez que representa o sistema no qual será possível executar a aplicação. O *software* foi desenvolvido para ser executado em desktop com o sistema operacional *Windows*.

O segundo se trata da linguagem de programação utilizada. A linguagem Java será utilizada devido a sua presença universal e pelos recursos disponibilizados pela linguagem.

O terceiro requisito trata da usabilidade, uma vez que representa a interação entre o sistema e o usuário. O sistema permitirá fácil entendimento de seu uso e facilidade no manuseio, uma vez que são poucos os processos de interação entre o usuário e a ferramenta, além disso as ações dos usuários serão bem intuitivas e descritas pela aplicação.

O quarto requisito trata da confiabilidade, uma vez que representa os possíveis erros que o sistema possa apresenta, para tal, ao ocorrer erros, estes serão tratados e serão mostrados ao usuários para que os mesmos possam entender que tipo de falha o sistema apresenta, além disso a informação será disponibilizada para que os erros sejam corrigidos.

Já o quinto requisito trata do desempenho, uma vez que representa o tempo estimado para que a informação desejada seja processada e mostrada ao usuário na sua forma final.

O sexto trata do gerenciador de banco de dados que será o PostgreSQL. Por sua capacidade de processamento de informação. O banco será o local de armazenamento de todos os dados tratados.

O sétimo requisito trata da disponibilidade, uma vez que representa o tempo em que a ferramenta estará disponível para uso. A ferramenta estará disponível a todo momento, considerando que a rede da UEMA esteja permanentemente ativa. Em caso de falha na rede da UEMA, a ferramenta não realizará sua atividade fim.

4.2 Questionário sobre a importância da ferramenta computacional

Em virtude da necessidade de coletar a opinião dos indivíduos afetados diretamente por este processo, fez se um questionário para analisar a visão destes indivíduos quanto a implementação da ferramenta e suas consequências.

Desse modo temos que o questionário é "[...] a técnica de investigação composta por um conjunto de questões que são submetidas a pessoas com o propósito de obter informações sobre conhecimentos, crenças, sentimentos, valores, interesses, expectativas, aspirações, temores, comportamento presente ou passado etc." (GIL, 2008, p. 121). Abaixo, estão relacionadas as perguntas, em forma de consulta aos oficiais e praças especiais, sobre o desenvolvimento da ferramenta computacional.

1. A qual círculo hierárquico você pertence:

( ) Oficial

( ) Praça especial

2. Na sua opinião, o processo de cálculo da antiguidade seria mais eficiente e seguro se fosse realizado sem a intervenção humana?

( ) Sim

( ) Não

3. Sobre processos judiciais, acerca deste tema, em sua opinião, essa demanda diminuiria?

( ) Sim

( ) Não

4.3 Entrevista com o Oficial responsável pelo cálculo

Além do questionário, também se optou pelo uso da técnica de entrevista. A entrevista foi realizada com base em três perguntas.

> As entrevistas têm sido realizadas face a face. Essa tem sido a característica mais considerada para distingui-la do questionário, cujos itens são apresentados por escrito aos respondentes. Boa parte das considerações feitas nos manuais de pesquisa acerca da elaboração da entrevista referem-se à situação face a face. No entanto, nas últimas décadas vem sendo desenvolvida outra modalidade: a entrevista por telefone. (GIL, 2008, p.113)

Desse modo, a entrevista realizada foi por meio de telefone celular. E as perguntas utilizadas na entrevista foram:

1. De que forma é feito o cálculo atualmente?
2. Quanto tempo, em média, leva o processo de cálculo?
3. Em vista disso, na sua opinião, os serviços seriam reduzidos caso a ferramenta fosse implementada?

5 ANÁLISE DOS RESULTADOS

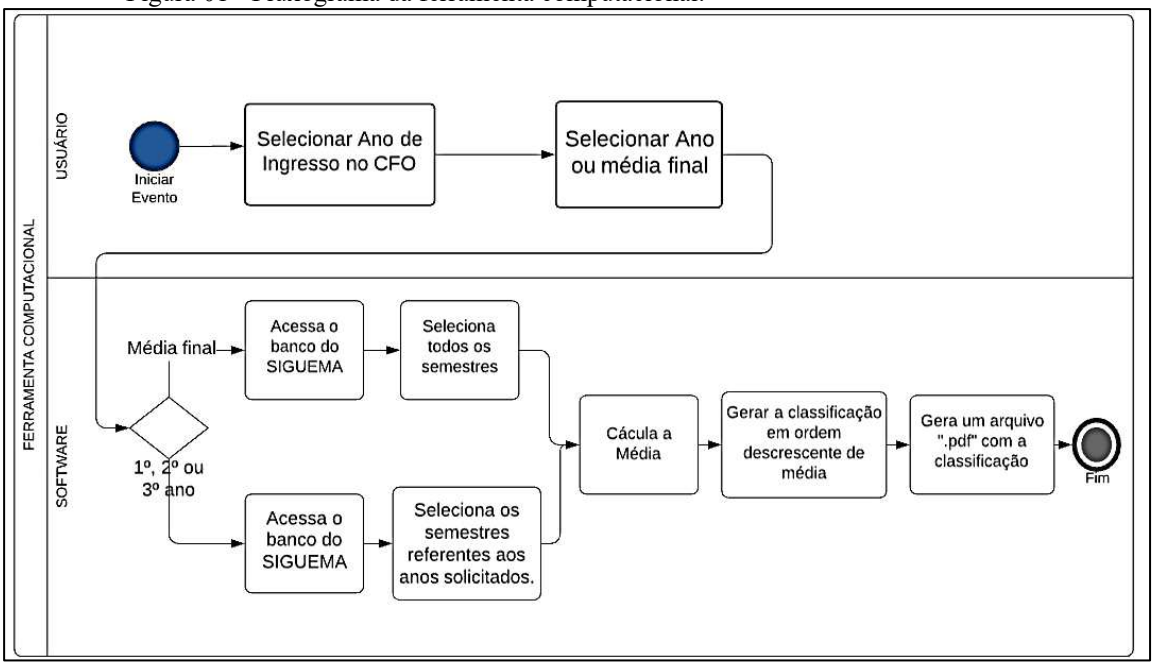
5.1 Construção da ferramenta computacional

Para realizar o cálculo da classificação do CFO, é necessário desenvolver um protótipo que se aproxima do produto final desejado, por meio do uso linguagem de programação Java, para escrever o código que realizará as ações desejadas. E também, com uso do gerenciador de banco de dados o PostgreSQL para armazenar, processar e manipular os dados.

5.1.1 Projeto

Para que se possa entender o funcionamento da ferramenta e aquilo se espera como produto final, ou seja, a classificação dos cadetes por ordem decrescente em um documento, tem-se o diagrama abaixo, Figura 1, com o fluxo de informação que será executado no funcionamento do *software*.

Figura 01 - Fluxograma da ferramenta computacional.



Fonte: Elaborado pelo autor (2020).

O usuário irá indicar qual o ano de ingresso no curso, isso será usado como identificador da turma que o aluno faz parte. Após isso será indicado o ano para o qual se deseja calcular a média, considerando que o CFO tem três anos e para cada ano se calcula uma média. Assim, ao selecionar o ano o programa irá fazer o acesso ao banco de dados do SIGUEMA

Desta forma, a arquitetura de software utilizada é a modelo em três camadas. Temos as seguintes camadas: camada de apresentação, camada de negócio e camada de dados. Onde a camada de apresentação, que será a interface, nesta há interação direta da ferramenta com o usuário, onde são feitas as requisições e consultas de dados. Na ferramenta em questão, na interface, o usuário seleciona o ano de ingresso e ano letivo ou média final. Na interface, serão mostradas mensagens confirmando que o processo foi feito por completo da forma prevista ou com algum erro, caso tenha erros, estes também serão mostrados.

Nesta camada não há interação com o usuário, os dados são voláteis e para que sejam mantidos, devem ser armazenados na camada de dados.
A ferramenta construída possuirá dois tipos de dados: dados advindos do banco de dados do SIGUEMA (alunos e médias) e dados que serão próprios da ferramenta (pesos e antiguidades). Os dados coletados do SIGUEMA deverão ser primeiramente armazenados para que então sejam feitas as manipulações.

Na camada de dados, se tem as classes que são responsáveis por armazenar as informações no banco de dados, além disso as classes também realizarão consultas, requisições e manipulações.

A linguagem utilizada é a JAVA, em virtude dos recursos disponíveis, dentre eles a orientação a objetos, característica que permite a fácil manipulação e criação de classes e de objetos para executar as diferentes ações durante todo processo. A ferramenta será desenvolvida em um *software* que auxilia o desenvolvimento de aplicações, chamado de Ambiente de Desenvolvimento Integrado (IDE). Para a ferramenta em questão se utilizará a IDE da NetBeans.

O banco de dados utilizado é o *PostgreSQL*, uma vez que este também é utilizado pelo sistema acadêmico da UEMA, o que facilita o acesso aos dados e a sua manipulação. O banco de dados do SIGUEMA armazena as notas dos alunos, além de dados pessoais e dados referentes a situação do aluno no curso da UEMA. A ferramenta acessará o banco e de acordo com o ano de entrada, irá ler e salvar os dados referentes ao aluno, principalmente as disciplinas cursadas, média final de cada disciplina, período e ano em que foi cursado a disciplina.

O PostgreSQL é um sistema gerenciador de banco de dados objeto relacional (CARVALHO, 2017, p. 10-11), e foi utilizado para armazenar as informações dos alunos do Curso de Formação de Oficiais com notas, disciplinas, pesos e outros itens. O acesso ao banco de dados do SIGUEMA foi concedido por meio do Núcleo de Tecnologia da Informação da Universidade (NTI-UEMA).

5.1.2 Implementação

O banco de dados do SIGUEMA disponibiliza os dados dos alunos por meio de um JSON. De acordo com Gama (2011), “JSON é basicamente um formato leve de troca de informações/dados entre sistemas”. O JSON contém informações numéricas e textuais, desse modo é possível discriminar nomes, códigos, notas e outras informações em apenas um arquivo, ou seja, um JSON com diversas informações de diversos tipos. Abaixo, Figura 2, vemos a forma como os dados são disponibilizados pelo sistema para serem lidos pela ferramenta.

Figura 02 - JSON com dados dos alunos.


```
CFO_CBMMA_turma_2017.txt - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
  },
  {
    "aluno": {
      "id_pessoa": [redacted],
      "nome": "RENNAN SILVA BARROS",
      "cpf_cnpj": "[redacted],
      "cpf_cnpj_hash": "68bdb33ba41121efe991d9499f596ffb",
      "email": "rennan.barros@hotmail.com",
      "data_nascimento": "1992-03-18",
      "sexo": "M",
      "endereco": null,
      "nome_pai": "[redacted],
      "nome_mae": [redacted],
      "numero_certificado_militar": "",
      "funcionario": false
    },
    "disciplinas": [
      {
        "id_matricula_componente": 2476760,
        "id_situacao_matricula": 4,
        "media_final": "9.40",
        "numero_faltas": 0,
        "detalhe_disciplina": {
          "id_componente_detalhes": 4354286,
          "id_componente": 22216,
          "nome": "LEITURA E PRODUCAO TEXTUAL",
          "codigo": "ASLNCUE016"
        },
        "ano": 2017,
```


Fonte: Elaborado pelo autor (2020).

E a partir daí o código em Java é capaz de acessar o banco, podendo retirar as informações necessárias para realizar o cálculo. O programa acessa ao banco de dados, faz a leitura das informações disposta da forma ilustrada a cima. A aplicação acessa o banco por meio da classe denominada “conexao.java”, ilustrada abaixo na Figura 3:

Figura 03 - Classe de conexão com o Banco do SIGUEMA.

```
14  import java.sql.SQLException;
15  import java.util.ArrayList;
16  import java.util.List;
17
18
19  public class conexao {
20
21
22
23      public static void main(String args[]) throws IOException, SQLException, JSONException {
24
25          String url ="http://new-sigws.intranet.uema.br/api/cfobpma?ano_ingresso=2017";
26              HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
27              String a = [redacted];
28              String b = [redacted]
29              conn.setRequestProperty( "appname", a);
30              conn.setRequestProperty( "apptoken", b);
31              conn.setRequestMethod("GET");
32              conn.setRequestProperty("Accept", "application/json");
33              if (conn.getResponseCode() != 200) {
34                  System.out.println("Erro " + conn.getResponseCode() + " ao obter dados da URL " + url);}
35
36          BufferedReader br = new BufferedReader(new InputStreamReader((conn.getInputStream())));
37
38              String output = "";
39              String line;
40              while ((line = br.readLine()) != null) {
41                  output += line;
42              }
43              conn.disconnect();
44
```

Fonte: Elaborado pelo autor (2020).

A conexão é realizada por meio de um link (“http://new-sigws.intranet.uema.br/api/cfobpma?ano_ingresso=2017"), onde é passado o ano de ingresso e chave de acesso para que se possa obter tais informações. Após a conexão com a URL, as informações recebidas são inseridas na variável “output”, a partir dela será inserido as informações no banco. O ano de ingresso utilizado foi o ano de 2017, uma vez que é ano de ingresso do autor da pesquisa em questão. Como a interface da ferramenta não foi desenvolvida, a seleção do ano de ingresso é alterada diretamente no próprio código, como visto a cima na Figura 3, é descrito no link de acesso a turma para qual se deseja calcular a média. Nesse caso a turma que ingressou em 2017 conforme a Figura 3.

Após extrair os dados do SIGUEMA é necessário armazenar tais dados para que possam ser trabalhados, desse modo, os dados extraídos e inseridos na variável “output” são enviados para o banco de dados do PostgreSQL. O JSON produzido e filtrado, dessa forma só é enviado para o banco os dados essenciais. Apenas o “id_pessoa”, “nome”, “cod_aluno” (recebe o mesmo número que “id_pessoa”), “código” da disciplina, “nome” da disciplina, “media_final”, “periodo” e “ano”. Para tanto foi desenvolvida a classe “lerArquivo” com os métodos “preencher_aluno” e “preencher_disciplina”, como vemos abaixo, na Figura 4 e 5:

Figura 04 - Método para preenchimento de aluno.

```
public int  preencher_aluno(JSONArray json) throws SQLException, JSONException {
    conectaBanco banco = new conectaBanco();

    for (int i = 0; i <json.length(); i++) {
        JSONObject alunos = json.getJSONObject(i);
        JSONObject list_alunos = alunos.getJSONObject("aluno");

         int cod_aluno = list_alunos.getInt("id_pessoa");
         String nome = list_alunos.getString("nome");

        aluno list_aluno = new aluno();
        list_aluno.setCod_aluno(cod_aluno);
        list_aluno.setNome(nome);
        // conectaBanco.inserir_aluno(cod_aluno, nome);
        b = banco.inserir_aluno(list_aluno);
        //System.out.println(b);

    }
    return b;

}
```

Fonte: Elaborado pelo autor (2020).

Dessa forma, o método acima é capaz de ler o JSON armazenado em “output” e é capaz de separar de cada aluno com suas respectivas disciplinas.

Figura 05 - Método para preencher disciplinas.

```
public int preencher_disciplina(int b,JSONArray json) throws SQLException, JSONException {
    conectaBanco banco = new conectaBanco();
    inserePeso inserirP = new inserePeso();
    for (int i = 0; i <json.length(); i++) {
        JSONObject objeto = json.getJSONObject(i);

        JSONObject list_alunos = objeto.getJSONObject("aluno");
        int cod_aluno = list_alunos.getInt("id_pessoa");

        JSONArray disciplina2 = objeto.optJSONArray("disciplinas");

        for (int k = 0; k < disciplina2.length(); k++) {

            JSONObject list_disciplina = disciplina2.optJSONObject(k);

            JSONObject detalhe_disc = list_disciplina.optJSONObject("detalhe_disciplina");
            String cod_disc = detalhe_disc.getString("codigo");

            String nome_disc = detalhe_disc.getString("nome");

            double media_final = list_disciplina.optDouble("media_final");
            int ano = list_disciplina.getInt("ano");
```

Fonte: Elaborado pelo autor (2020).

Desse modo, após armazenar todas as informações necessárias, o programa acessa as tabelas criadas no PostgreSQL, denominadas “aluno” e “disciplina”, e extrai os dados de acordo com o segundo comando do usuário. O usuário tem quatro opções, um dos três anos do curso e a opção do cálculo da nota geral do CFO, de acordo com está opção a ferramenta seleciona dois semestres referentes ao ano selecionado ou vai selecionar todos os semestres.

Assim para realizar o cálculo do ano, o programa primeiro precisa calcular a média dos dois semestres do ano desejado. A classe “calculaRank” gera a média de cada semestre por meio do método “calcularNota”, existe um método desse para cada semestre, todos semelhantes, mudando a apenas o período e o ano da disciplina cursada, o parâmetro é passado dentro da função “if” que realiza o filtro para o semestre desejado, conforme a Figura 6 abaixo:

Figura 06 - Método que calcula a média do semestre.

```
12  ,ist<rank1> calcularNota(LTDaluno ltAluno, LTDdisciplina ltDisciplina) {
13
14  .uno> allAluno = ltAluno.getLista();
15  .sciplina> allDisciplina = ltDisciplina.getLista2();
16  ink1> listaRank = new ArrayList<>();
17
18    i < allAluno.size(); i++) {
19
20   = new rank1();
21   dividendo = 0;
22  visor = 0;
23  ı flag = false;
24
25    for (int j = 0; j < allDisciplina.size(); j++)
26    {
27    if (allAluno.get(i).getCod_aluno() == allDisciplina.get(j).getCod_aluno_disc()&&allDisciplina.get(j).g
28
29    dividendo += allDisciplina.get(j).getMedia_final() * allDisciplina.get(j).getPeso();
30    divisor += allDisciplina.get(j).getPeso();
31    flag = true;
32
33
```

Fonte: Elaborado pelo autor (2020).

Após isso a média ponderada de cada semestre é gerada por meio da solicitação da classe “calculaAnos”, Figura 7, onde é realizado a média simples das médias de cada semestre. A classe “caculaAnos” solicita a execução do método “calculaNota” para dois semestres, recebe as duas médias e faz a média simples entre elas. Desse modo é gerado a classificação do 1º, 2º e 3º ano.

Figura 07 - Método que calcula a média do ano letivo.

```
15          calculaRank rankk = new calculaRank();
16
17
18      public void primeiroAno() throws SQLException {
19
20      List<rank1> listRank = rankk.calcularNota(ltAluno, ltDisciplina);
21      List<rank1> listRank2 = rankk.calcularNota2(ltAluno, ltDisciplina);
22      abrirConexao();
23      for (int l = 0; l < listRank.size(); l++) {
24
25          if (listRank.get(l).getCod_aluno()==listRank2.get(l).getCod_aluno()) {
26
27              double m1 = listRank.get(l).getMedia1();
28              double m2 = listRank2.get(l).getMedia1();
29
30              double N1= (m1+m2)/2;
31
32              String sql = "INSERT INTO public.rankum(cod_aluno, nome, media1)VALUES (?, ?, ?);";
33              PreparedStatement stm = conexao.prepareStatement(sql);
34              stm.setInt(1, listRank.get(l).getCod_aluno());
35              stm.setString(2, listRank.get(l).getNome());
36              stm.setDouble(3, N1);
37
38              stm.executeUpdate();
39
40      }}
41      fecharConexao();
42      }
43
44  public void segundoAno() throws SQLException {
45
```

Fonte: Elaborado pelo autor (2020).

Portanto, a média é calculada e armazenada no banco de dados em uma tabela denominada “rankum”, para a solicitação de média do 1º ano. No caso da média final, a ferramenta extrai do banco as informações referentes aos três anos, contidas nas tabelas “rankum”, “rankdois” e “ranktres”. Usamos apenas duas tabelas em virtude do última letivo

que ainda não está completo. A partir de então realiza o cálculo da média aritmética simples com as três médias de cada aluno, nesse caso apenas duas. Para tanto executa-se a classe “AntiguidadeFinal” e por meio do método “mediaFinal” (Figura 8), assim é calculada a média e em seguida armazenada no banco de dados na tabela “mediafinal” por meio da instrução em SQL para inserir dados no banco.

Figura 08 - Método que calcula a média final.

```
109  public void mediaFinal() throws SQLException {
110
111      AntiguidadeFinal Antifinal = new AntiguidadeFinal();
112
113      List<rank1> allrank1 = Antifinal.getLista11();
114      List<rank1> allrank2 = Antifinal.getLista22();
115      //List<rank1> allrank3 = Antigo.getLista3();
116
117          abrirConexao();
118          for (int l = 0; l < allrank1.size(); l++) {
119
120              if (allrank1.get(l).getCod_aluno()==allrank2.get(l).getCod_aluno()) {
121  // em caso de três médias usaria esse: if (allrank1.get(l).getCod_aluno()==allrank2.get(l).getCod_aluno()&
122                  double m1 = allrank1.get(l).getMedia1();
123                  double m2 = allrank2.get(l).getMedia1();
124                  //double m3 = allrank3.get(l).getMedia1();
125
126                  double N= (m1+m2)/2;
127
128                  String sql = "INSERT INTO public.mediafinal(cod_aluno, nome, mediafinal)VALUES (?, ?, ?);";
129                  PreparedStatement stm = conexao.prepareStatement(sql);
130                  stm.setInt(1, allrank1.get(l).getCod_aluno());
131                  stm.setString(2, allrank1.get(l).getNome());
132                  stm.setDouble(3, N);
133
134                  stm.executeUpdate();
```

Fonte: Elaborado pelo Autor (2020).

Por fim, após ser gerado a classificação do 1º, 2º, 3º ano ou média final, esta é enviada ao banco de dados e armazenada em uma tabela específica.

Desse modo, com todos os dados do cálculo já armazenados no banco, entra em ação a última classe, para um dos três, denominada “AntiguidadeCFO”, é nela que é realizada a leitura do banco de dados com as médias já calculadas e também é realizada a classificação por ordem decrescente de média. A classe realiza tal ação por meio do método “AntiguidadeANO”, Figura 9, que por sua vez extrai as informações do banco, por meio do método “getlista” da classe em questão, coloca os dados em uma variável do tipo lista e, por fim, criar um documento “.pdf” e descreve nele o código do aluno, nome completo do aluno e sua média.

Figura 09 - Método que gera o ".pdf" com a média do ano.

```
104   public static void AntiguidadeANO(AntiguidadeCFO antiguidadeCfo) {
105
106           // criação do documento
107           Document document = new Document();
108           try {
109
110   PdfWriter.getInstance(document, new FileOutputStream("C:\\Users\\Felipe Carmo Sousa\\Documents\\RANK_CFO1.pdf"));
111               document.open();
112
113               // adicionando um parágrafo no documento
114
115   document.add(new Paragraph("Universidade Estadual do Maranhão\n Academia de Bombeiros\n Curso de Formação de Ofic
116   + "Classificação (Antiguidade) 1° ANO - CFO- Turma 2017\n\nORD \t\t Código Aluno\t\t   Nome\t\t\t   Media"));
117
118               List<rank1> allrank = antiguidadeCfo.getLista();
119               for (int i = 0; i < allrank.size(); i++) {
120                   document.add(new Paragraph(i+1+"°      "+allrank.get(i).getCod_aluno()+ "\t    "
121                           + allrank.get(i).getNome()+ "\t\t    "
122                       + allrank.get(i).getMedia1()));
123
124               }
125               System.out.println("Documento gerado com sucesso!");
126   }
          catch(DocumentException de) {
128               System.err.println(de.getMessage());
129           }
130           catch(IOException ioe) {
```

Fonte: Elaborado pelo autor (2020).

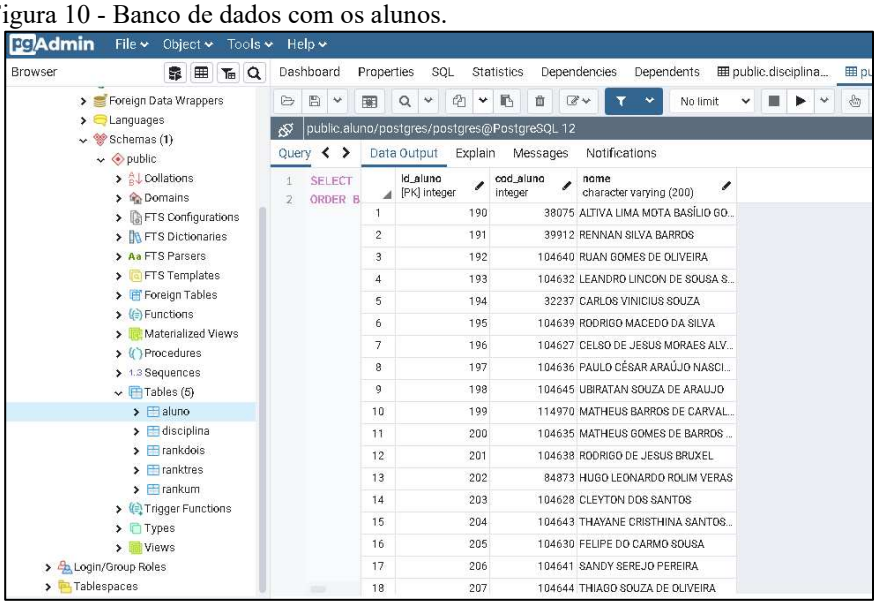
E, para a média final, temos a ação da classe denominada “AntiguidadeFinal”, é nela que é realizada a leitura do banco de dados com as médias já calculadas e também é realizada a classificação por ordem decrescente de média final, com uso do método “AntiguidadeFim”. Por fim, assim como na classe “AntiguidadeAno”, a classe “AntiguidadeFim” gera um arquivo “.pdf” com a classificação dos cadetes.

Desta forma, o código foi divido ao todo em 13 classes: Classe principal (“conexao”), classes que realizam cálculos (“calculaRank” e “calculaAnos”), classes que geram listas e inserem ou selecionam informações no banco (“LTDaluno”, “LTDdisciplina”, “lerArquivo” e “inserePeso”), classes que criam objetos (“aluno”, “disciplina” e rank1”), classe de conexão com o banco de dados do PostgreSQL (“conectaBanco”) e a classe que gerar o arquivo “.pdf” com a classificação dos cadetes por ano (“AntiguidadeCFO”) e a classificação final (“AntiguidadeFim”). Todas as classes que compõem o software encontram-se no Apêndice A.

5.1.3 Teste

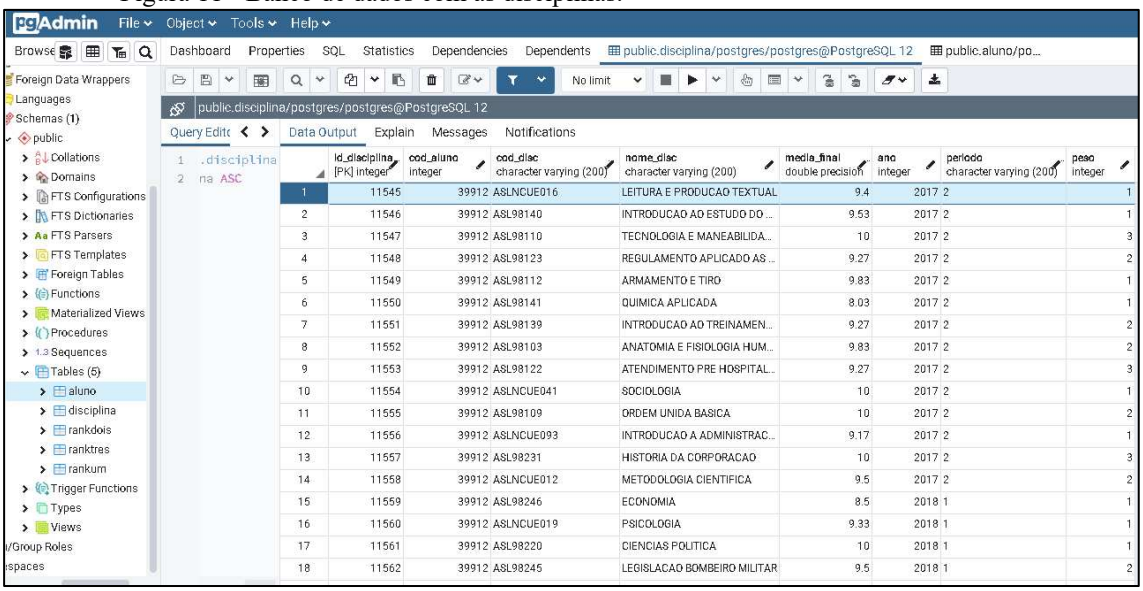
Em virtude de não existir ainda uma interface, a ferramenta foi construída e executada de acordo com a alteração direta no código das informações do ano de ingresso e ano para o qual se deseja calcular a antiguidade. Desse modo, ao executarmos a classe principal que acessa a URL do SIGUEMA e também o método que faz a inserção dos dados no banco, obtemos o resultado mostrado na Figura 10 e 11:

Figura 10 - Banco de dados com os alunos.



Fonte: Elaborado pelo autor (2020).

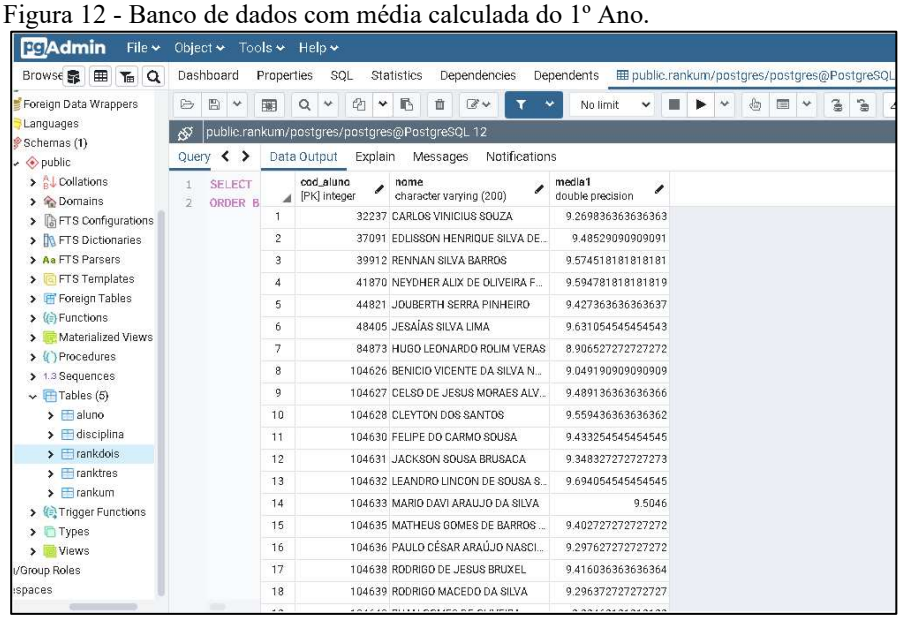
Figura 11 - Banco de dados com as disciplinas.



Fonte: Elaborado pelo Autor (2020).

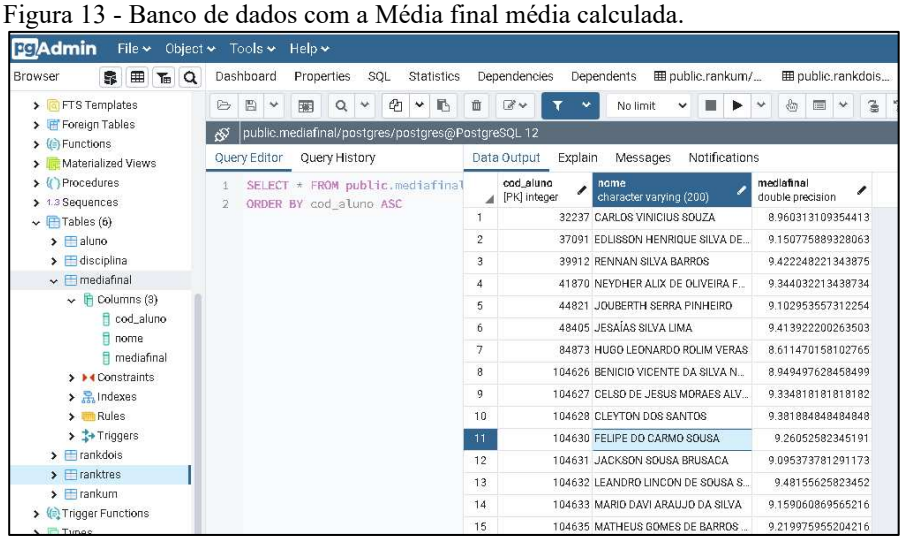
Com isso, o programa inicia a próxima etapa que é a fase de cálculo. A ferramenta faz o cálculo da média dos semestres referentes ao ano solicitado e com isso faz o cálculo da média do ano, caso seja solicitado apenas um dos anos. Caso seja solicitado a média final a ferramenta faz o cálculo de todos os seis semestres e em seguida faz o cálculo dos três anos e, por fim, faz a média final. Nos dois casos a ferramenta faz a inserção no banco da lista de alunos com as médias anuais (Figura 12) ou média final (Figura 13):

Figura 12 - Banco de dados com média calculada do 1º Ano.



Fonte: Elaborado pelo Autor (2020).

Figura 13 - Banco de dados com a Média final média calculada.

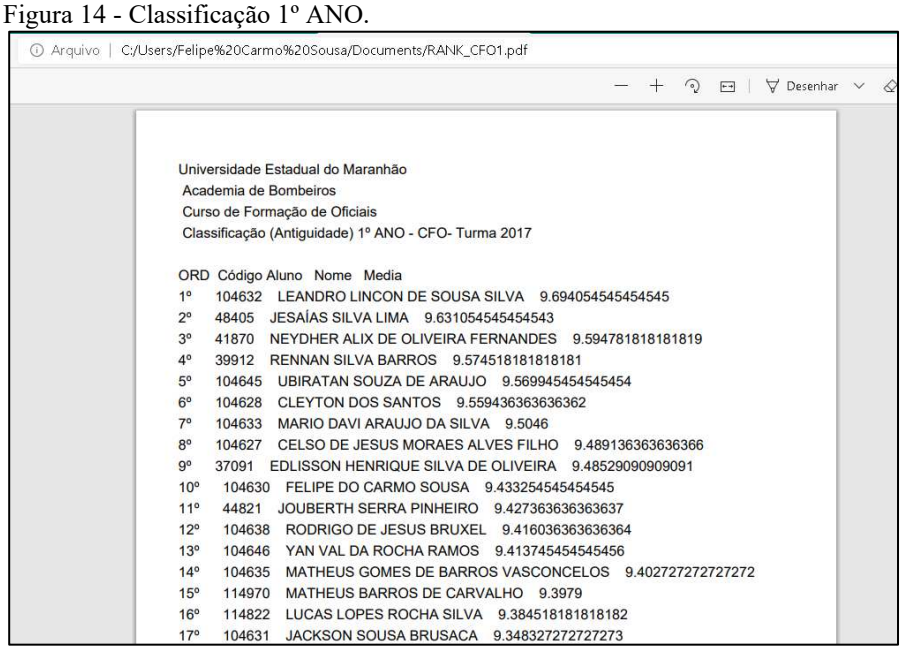


Fonte: Elaborado pelo Autor (2020).

Após estes procedimentos, a última fase é de criação do arquivo em ".pdf" com código, nome e média do aluno, e também classificados por ordem decrescente de média. O documento é gerado para os dois casos, para a média anual (figura 12) e para média final (Figura 13). A média final não foi possível ser gerada em virtude da não conclusão das disciplinas do 6º período, por conta da suspensão do calendário acadêmico da UEMA. Desse modo, faremos o cálculo final apenas considerando o 1º e 2º ano, para demonstrar que a ferramenta é capaz de fazer o cálculo, todavia considera-se que o cálculo será o mesmo tendo duas ou três médias, ou seja, usar apenas duas médias não comprometerá o cálculo quando for feito com três médias.

Abaixo temos os documentos gerados. Na figura 14, o arquivo em ".pdf" com a classificação do 1º ano do CFO por ordem decrescente de nota:

Figura 14 - Classificação 1º ANO.



Arquivo | C:/Users/Felipe%20Carmo%20Sousa/Documents/RANK_CFO1.pdf

Universidade Estadual do Maranhão
Academia de Bombeiros
Curso de Formação de Oficiais
Classificação (Antiguidade) 1º ANO - CFO- Turma 2017

ORD	Código Aluno	Nome	Media
1º	104632	LEANDRO LINCON DE SOUSA SILVA	9.694054545454545
2º	48405	JESAÍAS SILVA LIMA	9.631054545454543
3º	41870	NEYDHER ALIX DE OLIVEIRA FERNANDES	9.594781818181819
4º	39912	RENNAN SILVA BARROS	9.574518181818181
5º	104645	UBIRATAN SOUZA DE ARAUJO	9.569945454545454
6º	104628	CLEYTON DOS SANTOS	9.559436363636362
7º	104633	MARIO DAVI ARAUJO DA SILVA	9.5046
8º	104627	CELSO DE JESUS MORAES ALVES FILHO	9.489136363636366
9º	37091	EDLISSON HENRIQUE SILVA DE OLIVEIRA	9.48529090909091
10º	104630	FELIPE DO CARMO SOUSA	9.433254545454545
11º	44821	JOUBERTH SERRA PINHEIRO	9.427363636363637
12º	104638	RODRIGO DE JESUS BRUXEL	9.416036363636364
13º	104646	YAN VAL DA ROCHA RAMOS	9.413745454545456
14º	104635	MATHEUS GOMES DE BARROS VASCONCELOS	9.402727272727272
15º	114970	MATHEUS BARROS DE CARVALHO	9.3979
16º	114822	LUCAS LOPES ROCHA SILVA	9.384518181818182
17º	104631	JACKSON SOUSA BRUSACA	9.348327272727273

Fonte: Elaborado pelo autor (2020).

Na figura 15, o arquivo em ".pdf" com a classificação final do CFO por ordem decrescente de nota:

Figura 15 - Classificação Final.

Arquivo | C:/Users/Felipe%20Carmo%20Sousa/Documents/RANKCFO.pdf

Universidade Estadual do Maranhão
Academia de Bombeiros
Curso de Formação de Oficiais
Classificação (Antiguidade) Final - CFO- Turma 2017

ORD	Código Aluno	Nome	Media Final
1º	104632	LEANDRO LINCON DE SOUSA SILVA	9.48155625823452
2º	39912	RENNAN SILVA BARROS	9.422248221343875
3º	48405	JESAÍAS SILVA LIMA	9.413922200263503
4º	104628	CLEYTON DOS SANTOS	9.381884848484848
5º	104645	UBIRATAN SOUZA DE ARAUJO	9.361382147562582
6º	41870	NEYDHER ALIX DE OLIVEIRA FERNANDES	9.344032213438734
7º	104627	CELSO DE JESUS MORAES ALVES FILHO	9.334818181818182
8º	104630	FELIPE DO CARMO SOUSA	9.26052582345191
9º	104635	MATHEUS GOMES DE BARROS VASCONCELOS	9.219975955204216
10º	114970	MATHEUS BARROS DE CARVALHO	9.197163768115942
11º	104633	MARIO DAVI ARAUJO DA SILVA	9.159060869565216
12º	37091	EDLISSON HENRIQUE SILVA DE OLIVEIRA	9.150775889328063
13º	104640	RUAN GOMES DE OLIVEIRA	9.108601778656126
14º	114822	LUCAS LOPES ROCHA SILVA	9.105726482213438
15º	44821	JOUBERTH SERRA PINHEIRO	9.102953557312254
16º	104631	JACKSON SOUSA BRUSACA	9.095373781291173
17º	104636	PAULO CÉSAR ARAÚJO NASCIMENTO JÚNIOR	9.083933201581027

Fonte: Elaborado pelo autor (2020).

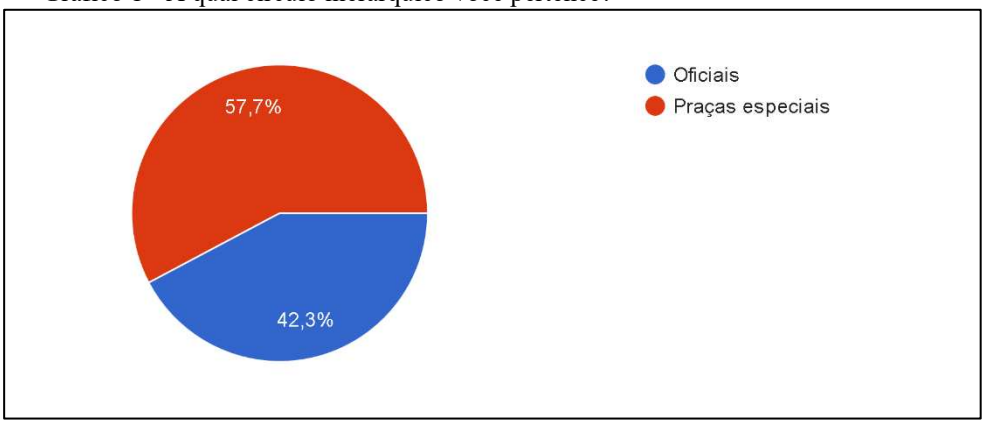
5.2 Análise e Discussão dos Resultados da Pesquisa de Levantamento de Demanda

O questionário preparado foi aplicado a 71 militares que já se formaram ou ainda estão em processo de formação no CFO. Engloba o efetivo de oficiais e praças especiais (cadetes e aspirantes a oficial BM) para que se possa conhecer a real necessidade de uma

ferramenta como esta e como tal ferramenta poderia trazer melhoria para o processo como um todo.

Na primeira pergunta elaborada é verificado o nível hierárquico da pessoa questionada, uma vez que o cálculo da antiguidade tem importância tanto para alunos em curso quanto para os oficiais já formados, como já foi demonstrado.

Gráfico 1 - A qual círculo hierárquico você pertence?

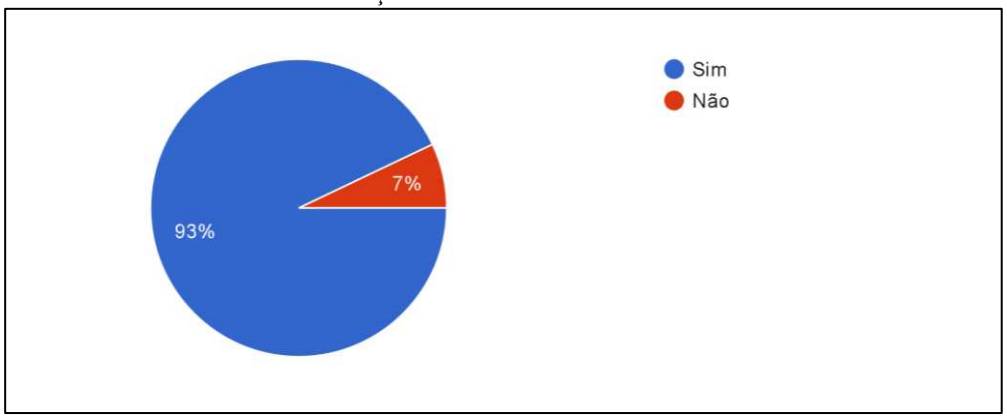


Fonte: Elaborado pelo autor (2020).

Como é possível ver, os dados mostram que 42,3% são oficiais formados na ABMJM e 57,7% são praças especiais.

Na segunda pergunta verifica-se a opinião dos questionados quanto a confiança no processo de cálculo, considerando a eficiência e a segurança do processo, enquanto um processo totalmente automatizado, ou seja, realizado de forma completa por um software de computador.

Gráfico 2 – Na sua opinião, o processo de cálculo da antiguidade seria mais eficiente e seguro se fosse realizado sem a intervenção humana?



Fonte: Elaborado pelo autor (2020).

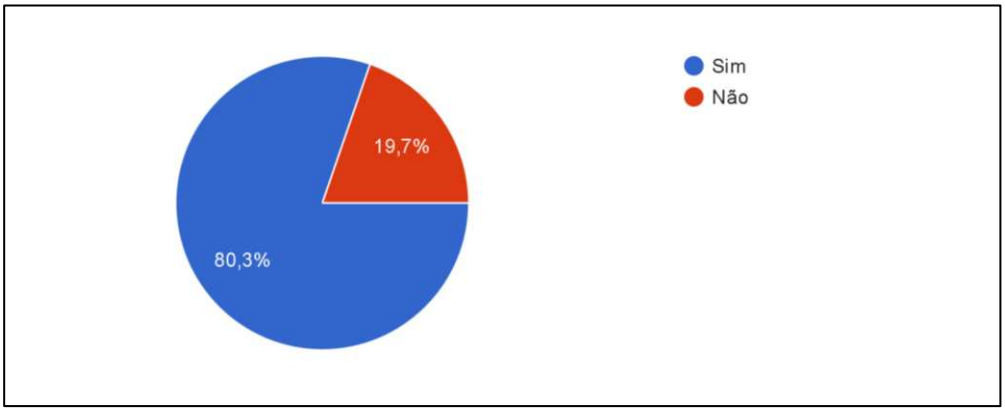
Como se pode observar, 93% dos entrevistados acreditam que a automatização do processo trará mais transparência ao processo e 07% acreditam que não trará mais transparência ao processo. Desse modo, verifica-se que as expectativas foram atendidas. Assim como se

esperava, a maior parte das pessoas questionadas, 93%, acreditam que o uso da ferramenta trará maior transparência para o processo.

Desse modo, observa-se que o referencial teórico levantando confirma a opinião dos entrevistados, uma vez que, para Cavalcanti (2017), a automatização de processos vai garantir que o processo seja executado como um todo, além disso proporciona maior eficiência, pois diminui os erros que podem acontecer ao longo do processo e preserva a integridade do processo. Ainda para Cavalcanti (2017), automatizar permite diminuir significativamente a possibilidade de fraude, pois minimiza-se a intervenção humano ao longo do processo.

Na terceira pergunta, verifica-se a opinião dos questionados quanto a diminuição da demanda de processos judiciais, considerando erros ao longo do processo.

Gráfico 3 – Sobre processos judiciais, acerca deste tema, em sua opinião, essa demanda diminuiria?



Fonte: Elaborado pelo autor (2020).

Como se pode observar, 80,3% dos questionados acreditam que com o uso da ferramenta computacional o número de demanda judicial diminuiria. Enquanto, 19,7% acredita que a demanda aumentaria. Assim, observa-se que as expectativas foram atendidas. Assim como esperava-se, a maioria dos questionados, 80,3%, acreditam que a implementação da ferramenta trará diminuição no número de processos judiciais, uma vez que o processo computacional minimiza grandemente a quantidade de erros ao longo de todo processo.

Em virtude de ser um processo complexo e importante, a presença de erros possibilita a existência de demandas judiciais, entretanto, com a automatizar os erros são mitigados, por sua vez as demandas judiciais também serão mitigadas, considerando a relação que existe entre os elementos. Desse modo, com bem nos assegura Cavalcanti (2017), um processo automatizado permite o controle total de todas as atividades desenvolvidas, e assim, o processo será executado com a maior precisão possível garantindo que os resultados sejam apresentados de acordo com o esperado.

Portanto, com base nos dados apresentados, pode-se verificar que há a necessidade de tal ferramenta computacional para calcular e gerar a classificação dos cadetes. Os resultados apontam que a implementação de tal ferramenta é muito importante e traria resultados significativos para os envolvidos no processo.

5.3 Análise e Discussão da entrevista

Considerando que o procedimento de cálculo atual realizado pela ABMJM, foi realizada uma entrevista com a Oficial responsável pela produção da Antiguidade do CFO, ou seja, pela classificação dos cadetes. A entrevista foi realizada com a 1º Tenente QOCBM Gilzimary de Jesus Sobrinho Privado, foi realizada uma entrevista por pautas, uma vez que foi apresentado um rol de perguntas abertas. Segue abaixo as perguntas e a resposta da entrevistada.

Abaixo tem-se a transcrição da entrevista:

Foi informado o título da pesquisa, assim como o objetivo do presente trabalho. Em primeiro lugar, foi perguntado: "De que forma é feito o cálculo atualmente?", a entrevistada respondeu: "Ele é feito manualmente e por meio do software Excel. Manualmente, em relação a coleta das notas no sistema e nas atas de notas da ABMJM. E também é manual no que se refere ao processo de inserção das notas nas planilhas do Excel. E, no software Excel, é realizado o cálculo das médias semestrais e anuais, por meio das inserções de fórmulas". Em seguida foi perguntado: "Quanto tempo, em média, leva o processo de cálculo?", a entrevistada respondeu: "Atualmente, são dois dias. Pois solicito um dia para que seja inserida as notas no Excel e o segundo dia para que seja realizada a revisão das notas, para verificar possíveis erros de digitação ou cálculo."

Por fim, foi perguntado: "Em vista disso, na sua opinião, os serviços seriam reduzidos caso a ferramenta fosse implementada?", a entrevistada respondeu: "Sim, com a implementação da ferramenta o serviço seria extremamente reduzido. Haveria uma diminuição significativa dos esforços no processo.". Além disso, a entrevistada ainda deu a seguinte sugestão: "Se fosse possível, seria interessante que o processo seja público a cada semestre e não a cada ano letivo, como é atualmente. Ou também, se possível, que seja disponibilizado de forma permanente e seja atualizada a cada inserção de nota no sistema, para que os alunos acompanhem em tempo real a classificação."

Em vista disso, com base no que diz Bueno (2014), é perceptível que o processo permitirá que os administradores tomem decisões com base nos resultados e assim, possam

empregar da melhor forma seus recursos, diminuindo a intervenção humana no processo e alocando os recursos humanos para outras atividades afins ou não. E por fim, será possível obter sempre melhores resultados, uma vez que, em vista do que foi descrito o processo será mais rápido e eficiente, como bem nos assegura Srour (1998), a Revolução Digital trouxe melhorias significativas para os processos manuais repetitivos ao torna-los automatizados.

Portanto, com base nas informações colhidas na entrevista podemos verificar a importância da ferramenta computacional para automatizar este processo. Como observado, o processo é feito manualmente e, sendo assim, é mais suscetível que ocorram erros de digitação ou de cálculo. Além disso, é possível comprovar que o tempo de produção da classificação é elevado, contudo com a implementação da ferramenta tal processo seria realizado quase que instantaneamente. Além disso, os serviços empregados para que a classificação seja gerada, com visto, seria drasticamente reduzido, conforme dito e conforme analisado nos estudos. Dessa forma, podemos dizer que os dados coletados na entrevista estão de acordo com a aquilo que foi estudado, corroborando as hipóteses levantadas ao longo da pesquisa.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento do presente estudo provou a importância da união entre a academia e a UEMA para o desenvolvimento de tecnologia voltada para o aprimoramento de atividades. Neste caso em especifico, a importância de automatizar processos manuais com uso de tecnologia para seu aperfeiçoamento e aumento da eficiência para que os resultados fossem obtidos com maior confiabilidade e precisão, além disso demonstrou-se como uma ferramenta computacional, no caso um *software*, permitiu aumentar a eficiência do cálculo e tempo de construção da classificação dos cadetes do Curso de Formação de Oficiais.

Chegou-se a um *software* capaz de gerar quatro classificações, sendo três de cada letivo cursado e a quarta a nota final considerando todas as médias já calculadas. O *software* desenvolvido proporcionará maior confiabilidade no processo de obtenção da classificação dos cadetes, uma vez que no processo de seleção de notas e de cálculo não há mais intervenção humana, desse modo os erros são diminuídos drasticamente. Além disso, a economia de recursos humanos e materiais aumenta, pois todo processo será realizado pela ferramenta, o usuário apenas deverá informar o ano de entrada dos cadetes e selecionar o ano letivo para o qual quer que seja gerada a classificação.

Ficou demonstrado a importância de automatizar processos e usar a tecnologia para substituir o homem em ações repetitivas. Diante das legislações apresentadas ficou evidente a grande importância da classificação para vida do militar, enquanto cadete e após a conclusão do curso. Além disso, a ferramenta foi desenvolvida com êxito e, portanto, fica evidente que os objetivos propostos foram alcançados.

A linguagem de programação Java foi de fundamental importância devido a suas características e também a facilidade de utilizar os seus recursos para desenvolvimento de *software*. Por ser uma linguagem orientada a objetos foi possível desenvolver uma ferramenta com fácil manipulação, uma vez que é possível modificar em partes do código sem prejudicar as demais partes. Além do fato de ser uma linguagem moderna e que é usada em todo tipo de tecnologia seja computacional ou em ferramentas eletrônicas do cotidiano.

O banco de dados utilizado foi de grande valia, pois os dados extraídos do sistema e tratados pela ferramenta foram armazenados em grande escala em um banco. O banco permitiu não só o armazenamento, mas como também permitiu a reutilização das informações guardadas para pudessem ser trabalhadas e geradas novas informações. O banco de dados utilizado foi o PostgreSQL, pois se trata de um banco com alto desempenho

Devido à grande importância do tema, torna-se necessário o aprimoramento da ferramenta desenvolvida, uma vez que o processo de implantação da ferramenta em sua forma final e completa ainda está por fazer. É necessária uma interface de comunicação que filtre os usuários e também as informações desejadas.

Nesse sentido, o desenvolvimento da ferramenta computacional auxilia a geração da classificação dos cadetes, demonstrando assim a sua importância para os alunos do Curso de Formação de Oficiais e aprimora o processo, tornando-o mais seguro, preciso e eficiente.

7 REFERÊNCIAS

ACADEMIA DE BOMBEIROS MILITAR JOSUÉ MONTELLO (ABMJM). **Normas Gerais de Ação do Corpo de Alunos.** Maranhão. 2010.

BRASIL. Lei Nº 6.880, de 9 de dezembro de 1980 - **Estatuto Dos Militares** - – Brasília, DF – 1980.

BROCKE, Jan Vom; ROSEMANN, Michael. **Metodologia de Pesquisa**. Rio Grande do Sul: AMGH, 2013. 617 p.

BUENO, Alexei. **Informática para estudantes de secretariado**. 1. ed. Mirassol, São Paulo: Edição do Autor, 2014.

CALEUM. **O que é Java**. [*S. l.*]: CALEUM, 2019. Disponível em: https://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/. Acesso em: 4 dez. 2019.

CAMPOS, André L.N.; CAMPOS, Artur F. **Java Start**. 1. ed. Rio De Janeiro: Sistema Moderno De Informação, 2018. 212 p. ISBN 978-85-45563-00-6.

CARVALHO, Vinicius. **POSTGRESQL**: Banco de Dados Para Aplicações Web Modernas. 1. ed. [*S. l.*]: Casa Do Código, 2017. 204 p. *E-book*.

CASTELLS, Manuel. **A sociedade em rede**. São Paulo: Paz e Terra, 1999.

CAVALCANTI, Rubens. **Modelagem de Processos de Negócios: roteiro para realização de projetos de modelagem de processos de negócios**. Rio de Janeiro: Brasport, 2017.

DATE, C.J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Elsevier, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.

FINEGAN, Edward; LIGUORI, Robert. **Java Se 8: Guia De Estudos Para O Exame 1Z0-808**. São Paulo: Bookman, 2018. 564 p.

FREIRE DE SÁ, Loamy. **Compras em empresas lucrativas**. São Paulo: Biblioteca 24horas, 2015. 99 p.

FURGERI, Sérgio. **Java Ensino Didático: Desenvolvimento E Implementação De Aplicações**. São Paulo: Saraiva, 2018. 328 p.

FURTADO, Vasco. **Tecnologia e Gestão da informação na Segurança pública**. Rio de Janeiro: Garamond, 2002.

GAMA, Alexandre. **O que é um JSON.** 2011. Disponível em: https://www.devmedia.com.br/o-que-e-json/23166. Acesso em 10/07/2020.

GASPAROTTO, Henrique Machado. **Os 4 pilares da Programação Orientada a Objetos**. [*S. l.*], 2014. Disponível em: https://www.devmedia.com.br/os-4-pilares-da-programacao-orientada-a-objetos/9264. Acesso em: 4 dez. 2019

GIL, A. C. **Métodos e técnicas de pesquisa social**. 6. ed. São Paulo: Atlas, 2008.

LORENZI, Fabiana; LOPES, Arthur Vargas. **Linguagem de programação Pascal**. Rio Grande do Sul: Ulbra, 2000. 124 p.

MANZANO, José Augusto N. G.; JUNIOR, Roberto Afonso da Costa. **Programação de computadores com Java**. 1. ed. São Paulo: Saraiva, 2014.

MARANHÃO. Lei N.º 6.513, de 30 de novembro de 1995. **Estatuto dos Policiais-Militares da Polícia Militar do Maranhão e dar outras providências.** São Luís, MA, nov. 1995.

MARTINS, José Carlos Cordeiro. **Técnicas para gerenciamento de projetos de software**. Rio de Janeiro: Brasport, 2007.

MENDES, Douglas Rocha. **Programação Java Com Ênfase Em Orientação A Objetos**. 1. ed. São Paulo: NOVATEC, 2009. 456 p. ISBN 978-85-7522-176-1.

MOREIRA NETO, Oziel. **Entendendo e dominando o Java**. São Paulo: Digerati Books, 2009. 416 p.

PALUDO, Augustinho Vicente. **Administração pública: teoria e questões**. Rio de Janeiro: Elsevier, 2010.

PINTO, Santa Rita Bento. **VBA 2007 na prática**. São Paulo: Digerati, 2009. 128 p.

PRODANOV, Cléber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. Novo Hamburgo: Feevale, 2013.

RAMARKRISHNAN, Raghu; GEHKRE, Johannes. **Sistema de gerenciamento de banco de dados**. 3. ed. Porto Alegre: AMGH, 2011.

RAMPAZZO, Lino. **Metodologia científica: para alunos dos cursos de graduação e pós-graduação**. 3. ed. São Paulo: Loyola, 2005.

REISSWITZ, Flavia. **Análise de Sistemas Vol. X**. Santa Catarina: Clube dos Autores, 2009. 183 p.

RICARTE, I, L. **Programação Orientada a Objetos: Uma Abordagem com Java**. Faculdade de engenharia elétrica e de computação UNICAMP. Campinas 2001.

SCHILDT, Herbert. **Java para Iniciantes**. 6. ed. Rio Grande do Sul: Bookman, 2015. 704 p.

SROUR, Robert H. **Poder, Cultura e Ética nas Organizações**. 5 ed. Rio de Janeiro: Campus, 1998.

APÊNDICES

APÊNDICE A – CÓDIGO JAVA DA FERRAMENTA COMPUTACIONAL

```
package notas;

import java.io.BufferedReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.SQLException;
import java.util.List;
import org.json.JSONArray;
import org.json.JSONException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;


public class conexao {



    public static void main(String args[])
            throws IOException, SQLException, JSONException {

        String url ="http://new-sigws.intranet.uema.br/"
                + "api/cfobpma?ano_ingresso=2017";

        HttpURLConnection conn = (HttpURLConnection)
                new URL(url).openConnection();
        String a = [redacted];
        String b = [redacted];
        conn.setRequestProperty( "appname", a);
        conn.setRequestProperty( "apptoken", b);
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Accept", "application/json");
        if (conn.getResponseCode() != 200) {
        System.out.println("Erro " + conn.getResponseCode() +
                " ao obter dados da URL " + url);}

        BufferedReader br = new BufferedReader
            (new InputStreamReader((conn.getInputStream())));

        String output = "";
```

```
        String line;
        while ((line = br.readLine()) != null) {
        output += line;
        }
        conn.disconnect();

       //Acessa o Json e insere no banco

       lerArquivo arquivo = new lerArquivo();
       JSONArray leitor_json = arquivo.leitor_json(output);

       conectaBanco.abrirConexao();
       int c= arquivo.preencher_aluno(leitor_json);
      // System.out.println(b);
       arquivo.preencher_disciplina(c, leitor_json);
       conectaBanco.fecharConexao();


//Faz o cálculo e salva a calssificação no banco
        calculaAnos rankfull = new calculaAnos ();
        rankfull.primeiroAno();
        rankfull.segundoAno();
        rankfull.terceiroAno();
//Envia para o banco a classificação final do CFO
        AntiguidadeFinal Antigo1 = new AntiguidadeFinal();
        Antigo1.mediaFinal();

//Classifica por ordem decrescente em um arquivo .pdf de um dos anos do CFO
        AntiguidadeCFO antiguidadeCfo = new AntiguidadeCFO();
        antiguidadeCfo.AntiguidadeANO(antiguidadeCfo);

//Gera a classificação final por ordem decrescente em um arquivo .pdf
        Antigo1.AntiguidadeFim(Antigo1);



    }
}
```

```
package notas;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;


public class conectaBanco {
    public static Connection conexao;
    public PreparedStatement stm1,stm, sql;
    public ResultSet rs;
    private static final String driver = "org.postgresql.Driver";
    private static final String
            url = "jdbc:postgresql://localhost:5432/postgres";
    private static final String usuario = "postgres";
    private static String senha = "123456";
    public int a, idGerado, idGerado2;

    public static void abrirConexao()  {
try {
            System.setProperty("jdbc.Drivers",driver);
            conexao = DriverManager.getConnection(url,usuario,senha);
            System.out.println("Conectado ao banco!");
            } catch (Exception e) {
            System.out.println("Erro ao conectar ao banco! "+e);
            }

            }

    public void executaSQL (String sql) {
        try {
            stm = (PreparedStatement) conexao.createStatement
        (rs.TYPE_SCROLL_INSENSITIVE, rs.CONCUR_READ_ONLY );
            rs = stm.executeQuery(sql);
        } catch (SQLException ex) {
        }
    }
```

```
    public static void fecharConexao() throws SQLException{
        try{
        conexao.close();
        System.out.println("Banco fechado!");
        }
        catch (SQLException z){ System.out.println(""+z);}
    }


public int inserir_aluno(aluno aluno) throws SQLException {
    try {

String sql = "INSERT into aluno (id_aluno,cod_aluno,nome) "
        + "VALUES (DEFAULT, ?, ?);";

        PreparedStatement stm = conexao.prepareStatement
        (sql, PreparedStatement.RETURN_GENERATED_KEYS);

        stm.setInt(1, aluno.getCod_aluno());
        stm.setString(2, aluno.getNome());

        stm.executeUpdate();
        System.out.println("Aluno Cadastrado");

        rs = stm.getGeneratedKeys();
        rs.next();
        idGerado = rs.getInt(1);
        //System.out.println(idGerado);
        aluno.setId_aluno(idGerado);
        stm.close();
        } catch (Exception e) {
            System.out.println(e);
        }

        return idGerado;
    }



public int inserir_disciplina(int idGerado,disciplina disciplina)
                throws SQLException {

        try {
```

```
String sql = "INSERT into disciplina (id_disciplina, cod_aluno,cod_disc, "
        + "nome_disc, media_final, ano, periodo,peso) "
        + "VALUES (DEFAULT, ?,?,?,?,?,?,?);";

            PreparedStatement stm = conexao.prepareStatement
                (sql,Statement.RETURN_GENERATED_KEYS);
            stm.setInt(1, disciplina.getCod_aluno_disc());
            stm.setString(2, disciplina.getCod_disc());
            stm.setString(3, disciplina.getNome_disc());
            stm.setDouble(4, disciplina.getMedia_final());
            stm.setInt(5, disciplina.getAno());
            stm.setInt(6, disciplina.getPeriodo());
            stm.setInt(7, disciplina.getPeso());

            stm.executeUpdate();

            ResultSet rsl = stm.getGeneratedKeys();
            rsl.next();
            idGerado2 = rsl.getInt(1);
            disciplina.setId_disciplina(idGerado2);
            System.out.println("Disciplina cadastrada com sucesso");
            stm.close();

        } catch (Exception e) {
            System.out.println(e);
        } return idGerado2;
    }
}
```

```
package notas;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfWriter;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import static notas.conectaBanco.abrirConexao;
import static notas.conectaBanco.conexao;
import static notas.conectaBanco.fecharConexao;


public class AntiguidadeFinal {

 public List<rank1> getLista11() {
    String sql = "SELECT *FROM public.rankum ORDER BY cod_aluno";
    List<rank1> listaCFO11 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media1"));
listaCFO11.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
```

```
        return listaCFO11;

 }


 public List<rank1> getLista22() {
    String sql = "SELECT *FROM public.rankdois ORDER BY cod_aluno";
    List<rank1> listaCFO22 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media2"));
listaCFO22.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO22;

 }

 public List<rank1> getLista33() {
    String sql = "SELECT *FROM public.ranktres ORDER BY cod_aluno";
    List<rank1> listaCFO33 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {
```

```
                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media3"));
listaCFO33.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO33;

 }




public void mediaFinal() throws SQLException {

 AntiguidadeFinal Antifinal = new AntiguidadeFinal();

 List<rank1> allrank1 = Antifinal.getLista11();
 List<rank1> allrank2 = Antifinal.getLista22();
 //List<rank1> allrank3 = Antigo.getLista3();

    abrirConexao();
    for (int l = 0; l < allrank1.size(); l++) {

if (allrank1.get(l).getCod_aluno()==allrank2.get(l).getCod_aluno()) {
// em caso de três médias usaria esse:
//if (allrank1.get(l).getCod_aluno()==allrank2.get(l).getCod_aluno()
//+&&allrank2.get(l).getCod_aluno()==allrank3.get(l).getCod_aluno())

            double m1 = allrank1.get(l).getMedia1();
            double m2 = allrank2.get(l).getMedia1();
            //double m3 = allrank3.get(l).getMedia1();

            double N= (m1+m2)/2;

String sql = "INSERT INTO public.mediafinal"
        + "(cod_aluno, nome, mediafinal)VALUES (?, ?, ?);";
```

```
            PreparedStatement stm = conexao.prepareStatement(sql);
            stm.setInt(1, allrank1.get(l).getCod_aluno());
            stm.setString(2, allrank1.get(l).getNome());
            stm.setDouble(3, N);

            stm.executeUpdate();

    }}
fecharConexao();
}

public List<rank1> getLista4() {
    String sql = "SELECT *FROM public.mediafinal ORDER BY mediafinal DESC";
    List<rank1> listaCFO = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("mediafinal"));
listaCFO.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO;

 }



public static void AntiguidadeFim(AntiguidadeFinal Antifim) {

        // criação do documento
        Document document = new Document();
```

```
        try {

 PdfWriter.getInstance(document, new FileOutputStream
        ("C:\\Users\\Felipe Carmo Sousa\\Documents\\RANKCFO.pdf"));
            document.open();

            // adicionando um parágrafo no documento

 document.add(new Paragraph("Universidade Estadual do Maranhão\n "
         + "Academia de Bombeiros\n Curso de Formação de Oficiais\n "
 + "Classificação (Antiguidade) Final - CFO- Turma 2017\n\nORD \t\t "
         + "Código Aluno\t\t   Nome\t\t\t   Media Final"));

    List<rank1> allrank = Antifim.getLista4();
    for (int i = 0; i < allrank.size(); i++) {
    document.add(new Paragraph(i+1+"º  "
            + ""+allrank.get(i).getCod_aluno()+ "\t"
+ allrank.get(i).getNome()+ "\t\t "
+ allrank.get(i).getMedia1()));

            }
            System.out.println("Documento gerado com sucesso!");
}
        catch(DocumentException de) {
            System.err.println(de.getMessage());
        }
        catch(IOException ioe) {
            System.err.println(ioe.getMessage());
        }
        document.close();
    }
 }
```

```
package notas;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import java.io.FileOutputStream;
import java.io.IOException;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfWriter;
import static notas.conectaBanco.*;

public class AntiguidadeCFO {

 public List<rank1> getLista() {
    String sql = "SELECT *FROM public.rankum ORDER BY media1 DESC";
    List<rank1> listaCFO1 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media1"));
listaCFO1.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO1;

 }
```

```
 public List<rank1> getLista2() {
    String sql = "SELECT *FROM public.rankdois ORDER BY media2 DESC";
    List<rank1> listaCFO2 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media2"));
listaCFO2.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO2;

 }

 public List<rank1> getLista3() {
    String sql = "SELECT *FROM public.ranktres ORDER BY media3 DESC";
    List<rank1> listaCFO3 = new ArrayList<>();

try {
            abrirConexao();
            PreparedStatement stm = conexao.prepareStatement(sql);
            ResultSet rs = stm.executeQuery();
            fecharConexao();

            while (rs.next()) {

                rank1 obj = new rank1();
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
obj.setMedia1(rs.getDouble("media3"));
```

```
listaCFO3.add(obj);

}
    }
catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return listaCFO3;

 }





 public static void AntiguidadeANO(AntiguidadeCFO antiguidadeCfo) {

        // criação do documento
        Document document = new Document();
        try {

 PdfWriter.getInstance(document, new FileOutputStream
       ("C:\\Users\\Felipe Carmo Sousa\\Documents\\RANK_CFO1.pdf"));
            document.open();

            // adicionando um parágrafo no documento

 document.add(new Paragraph("Universidade Estadual do Maranhão\n "
         + "Academia de Bombeiros\n Curso de Formação de Oficiais\n "
 + "Classificação (Antiguidade) 1° ANO - CFO- Turma 2017\n\n"
         + "ORD \t\t Código Aluno\t\t   Nome\t\t\t   Media"));

            List<rank1> allrank = antiguidadeCfo.getLista();
            for (int i = 0; i < allrank.size(); i++) {
             document.add(new Paragraph(i+1+"°     "
                         + " "+allrank.get(i).getCod_aluno()+ "\t    "
           + allrank.get(i).getNome()+ "\t\t    "
              + allrank.get(i).getMedia1()));

            }
            System.out.println("Documento gerado com sucesso!");
}
        catch(DocumentException de) {
            System.err.println(de.getMessage());
```

```
        }
        catch(IOException ioe) {
            System.err.println(ioe.getMessage());
        }
        document.close();
    }



}
```

```
package notas;
import java.io.FileReader;
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import org.json.JSONTokener;

public class lerArquivo {
int b,c;

 public JSONArray leitor_json(String path)
         throws IOException, JSONException {

     JSONArray json = new JSONArray
        (new JSONTokener(new FileReader(path)));

     return json;}

public int  preencher_aluno(JSONArray json)
        throws SQLException, JSONException {
    conectaBanco banco = new conectaBanco();

    for (int i = 0; i <json.length(); i++) {
        JSONObject alunos = json.getJSONObject(i);
        JSONObject list_alunos = alunos.getJSONObject("aluno");

         int cod_aluno = list_alunos.getInt("id_pessoa");
         String nome = list_alunos.getString("nome");

        aluno list_aluno = new aluno();
        list_aluno.setCod_aluno(cod_aluno);
        list_aluno.setNome(nome);

        b = banco.inserir_aluno(list_aluno);

       }
    return b;
     }

public int preencher_disciplina(int b,JSONArray json)
```

```
        throws SQLException, JSONException {

    conectaBanco banco = new conectaBanco();
    inserePeso inserirP = new inserePeso();
    for (int i = 0; i <json.length(); i++) {
        JSONObject objeto = json.getJSONObject(i);

        JSONObject list_alunos = objeto.getJSONObject("aluno");
        int cod_aluno = list_alunos.getInt("id_pessoa");

        JSONArray disciplina2 = objeto.optJSONArray("disciplinas");

        for (int k = 0; k < disciplina2.length(); k++) {

        JSONObject list_disciplina = disciplina2.optJSONObject(k);

        JSONObject detalhe_disc =
                list_disciplina.optJSONObject("detalhe_disciplina");

        String cod_disc = detalhe_disc.getString("codigo");

        String nome_disc = detalhe_disc.getString("nome");

        double media_final = list_disciplina.optDouble("media_final");
        int ano = list_disciplina.getInt("ano");
        int periodo = list_disciplina.getInt("periodo");

        disciplina disc = new disciplina();
        disc.setCod_aluno_disc(cod_aluno);
        disc.setCod_disc(cod_disc);
        disc.setNome_disc(nome_disc);
        disc.setMedia_final(media_final);
        disc.setAno(ano);
        disc.setPeriodo(periodo);

            int pi = inserirP.inserirPeso(cod_disc);
            disc.setPeso(pi);

            c = banco.inserir_disciplina(b, disc);
               }
}     return c;
}}
```

```
package notas;


import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import static notas.conectaBanco.*;


public class LTDaluno{

public List<aluno> getLista() {
    String sql = "SELECT * FROM public.aluno";
    List<aluno> lista = new ArrayList<>();

        try {

        abrirConexao();
        PreparedStatement stm = conexao.prepareStatement(sql);
        ResultSet rs = stm.executeQuery();
        fecharConexao();

while (rs.next()) {

aluno obj = new aluno();
obj.setId_aluno(rs.getInt("id_aluno"));
obj.setCod_aluno(rs.getInt("cod_aluno"));
obj.setNome(rs.getString("nome"));
lista.add(obj);

}
}
catch (Exception e) {
System.out.println(e.getMessage());
        }
return lista;
}}
```

```
package notas;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import static notas.conectaBanco.*;


public class LTDdisciplina{

public List<disciplina> getLista2() {
    String sql = "SELECT * FROM public.disciplina";
    List<disciplina> lista2 = new ArrayList<>();

        try {

        abrirConexao();
        PreparedStatement stm = conexao.prepareStatement(sql);
        ResultSet rs = stm.executeQuery();
        fecharConexao();

while (rs.next()) {

disciplina obj = new disciplina();
obj.setId_disciplina(rs.getInt("id_disciplina"));
obj.setCod_aluno_disc(rs.getInt("cod_aluno"));
obj.setCod_disc(rs.getString("cod_disc"));
obj.setNome_disc(rs.getString("cod_disc"));
        obj.setMedia_final(rs.getDouble("media_final"));
        obj.setAno(rs.getInt("ano"));
        obj.setPeriodo(rs.getInt("periodo"));
        obj.setPeso(rs.getInt("peso"));

        lista2.add(obj);

}}
catch (Exception e) {
System.out.println(e.getMessage());
        }
return lista2;
}
}
```

```
package notas;


public class aluno {
    int id_aluno;
    int cod_aluno;
    private String nome;



    private void aluno (int id_aluno,int cod_aluno, String nome) {
        this.cod_aluno = cod_aluno;
        this.nome = nome;

}

    public int getId_aluno() {
        return id_aluno;
    }

    public void setId_aluno(int id_aluno) {
        this.id_aluno = id_aluno;
    }


    public int getCod_aluno() {
        return cod_aluno;
    }

    public void setCod_aluno(int cod_aluno) {
        this.cod_aluno = cod_aluno;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

C:/Users/Felipe Carmo Sousa/Documents/NetBeansProjects/Notas/src/notas/calculaRank.java

```
package notas;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import java.io.FileOutputStream;
import java.io.IOException;

public class calculaRank {

public static List<rank1> calcularNota
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2017&&
                        allDisciplina.get(j).getPeriodo()==2) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();

divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);
```

```
listaRank.add(rank);

}
}
    return listaRank;
}


public static List<rank1> calcularNota2
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2018&&
                        allDisciplina.get(j).getPeriodo()==1) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();

divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);

listaRank.add(rank);
```

```
}
}
    return listaRank;
}



public static List<rank1> calcularNota3
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2018&&
                        allDisciplina.get(j).getPeriodo()==2) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();

divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);

listaRank.add(rank);
```

```
}
}
    return listaRank;
}

public static List<rank1> calcularNota4
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2019&&
                        allDisciplina.get(j).getPeriodo()==1) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();

divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);

listaRank.add(rank);

}
}
    return listaRank;
```

```
}

public static List<rank1> calcularNota5
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2019&&
                        allDisciplina.get(j).getPeriodo()==2) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();

divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);

listaRank.add(rank);

}
}
    return listaRank;
}

public static List<rank1> calcularNota6
```

```
        (LTDaluno ltAluno, LTDdisciplina ltDisciplina) {

List<aluno> allAluno = ltAluno.getLista();
List<disciplina> allDisciplina = ltDisciplina.getLista2();
List<rank1> listaRank = new ArrayList<>();

for (int i = 0; i < allAluno.size(); i++) {

    rank1 rank = new rank1();
double dividendo = 0;
int divisor = 0;
boolean flag = false;

        for (int j = 0; j < allDisciplina.size(); j++)
{
                if (allAluno.get(i).getCod_aluno() ==
                        allDisciplina.get(j).getCod_aluno_disc()&&
                        allDisciplina.get(j).getAno()==2020&&
                        allDisciplina.get(j).getPeriodo()==1) {

dividendo += allDisciplina.get(j).getMedia_final()
                        * allDisciplina.get(j).getPeso();
divisor += allDisciplina.get(j).getPeso();
flag = true;
    }}

if (flag){

rank.setCod_aluno(allAluno.get(i).getCod_aluno());
rank.setNome(allAluno.get(i).getNome());
rank.setMedia1(dividendo/divisor);

listaRank.add(rank);

}
}
    return listaRank;
}
}
```

```
package notas;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import static notas.conectaBanco.*;


public class calculaAnos {
       LTDaluno ltAluno = new LTDaluno();
       LTDdisciplina ltDisciplina = new LTDdisciplina();

       calculaRank rankk = new calculaRank();


    public void primeiroAno() throws SQLException {

    List<rank1> listRank = rankk.calcularNota(ltAluno, ltDisciplina);
    List<rank1> listRank2 = rankk.calcularNota2(ltAluno, ltDisciplina);
    abrirConexao();
    for (int l = 0; l < listRank.size(); l++) {

        if (listRank.get(l).getCod_aluno()
                ==listRank2.get(l).getCod_aluno()) {

            double m1 = listRank.get(l).getMedia1();
            double m2 = listRank2.get(l).getMedia1();

            double N1= (m1+m2)/2;

            String sql = "INSERT INTO public.rankum"
                    + "(cod_aluno, nome, media1)VALUES (?, ?, ?);";
            PreparedStatement stm = conexao.prepareStatement(sql);
            stm.setInt(1, listRank.get(l).getCod_aluno());
            stm.setString(2, listRank.get(l).getNome());
            stm.setDouble(3, N1);

            stm.executeUpdate();

    }}
    fecharConexao();
    }
```

```
public void segundoAno() throws SQLException {

    List<rank1> listRank3 = rankk.calcularNota3(ltAluno, ltDisciplina);
    List<rank1> listRank4 = rankk.calcularNota4(ltAluno, ltDisciplina);
    abrirConexao();
    for (int l = 0; l < listRank3.size(); l++) {

        if (listRank3.get(l).getCod_aluno()
                ==listRank4.get(l).getCod_aluno()) {

            double m1 = listRank3.get(l).getMedia1();
            double m2 = listRank4.get(l).getMedia1();

            double N2= (m1+m2)/2;

            String sql = "INSERT INTO public.rankdois"
                    + "(cod_aluno, nome, media2)VALUES (?, ?, ?);";
            PreparedStatement stm = conexao.prepareStatement(sql);
            stm.setInt(1, listRank3.get(l).getCod_aluno());
            stm.setString(2, listRank3.get(l).getNome());
            stm.setDouble(3, N2);

            stm.executeUpdate();

    }} fecharConexao();
}

public void terceiroAno() throws SQLException {

    List<rank1> listRank5 = rankk.calcularNota5(ltAluno, ltDisciplina);
    List<rank1> listRank6 = rankk.calcularNota6(ltAluno, ltDisciplina);
    abrirConexao();
    for (int l = 0; l < listRank5.size(); l++) {

        if (listRank5.get(l).getCod_aluno()==
                listRank6.get(l).getCod_aluno()) {

            double m1 = listRank5.get(l).getMedia1();
            double m2 = listRank6.get(l).getMedia1();

            double N3= (m1+m2)/2;

            String sql = "INSERT INTO public.ranktres"
```

```
                    + "(cod_aluno, nome, media3)VALUES (?, ?, ?);";
            PreparedStatement stm = conexao.prepareStatement(sql);
            stm.setInt(1, listRank5.get(l).getCod_aluno());
            stm.setString(2, listRank5.get(l).getNome());
            stm.setDouble(3, N3);

            stm.executeUpdate();

    }}
fecharConexao();
}}
```

```
package notas;

import java.sql.ResultSet;


public class disciplina {
    private int id_disciplina;
    int cod_aluno_disc;
    private String cod_disc;
    private String nome_disc;
    double media_final;
    int ano;
    int periodo;
    private int peso;

    public void disciplina(int id_disciplina, int cod_aluno_disc ,
            String cod_disc, String nome_disc,
            double media_final, int ano,int peso) {

        this.id_disciplina = id_disciplina;
        this.cod_aluno_disc = cod_aluno_disc;
        this.cod_disc = cod_disc;
        this.nome_disc = nome_disc;
        this.media_final = media_final;
        this.ano = ano;
        this.peso= peso;
    }

    public int getId_disciplina() {
        return id_disciplina;
    }


    public void setId_disciplina(int id_disciplina) {
        this.id_disciplina = id_disciplina;
    }

    public String getCod_disc() {
        return cod_disc;
    }

    public void setCod_disc(String cod_disc) {
```

```
        this.cod_disc = cod_disc;
    }

    public String getNome_disc() {
        return nome_disc;
    }

    public void setNome_disc(String nome_disc) {
        this.nome_disc = nome_disc;
    }

    public double getMedia_final() {
        return media_final;
    }

    public void setMedia_final(double media_final) {
        this.media_final = media_final;
    }

    public int getAno() {
        return ano;
    }

    public void setAno(int ano) {
        this.ano = ano;
    }

    public int getPeriodo() {
        return periodo;
    }

    public void setPeriodo(int periodo) {
        this.periodo = periodo;
    }

    public int getCod_aluno_disc() {
        return cod_aluno_disc;
    }

    public void setCod_aluno_disc(int cod_aluno_disc) {
        this.cod_aluno_disc = cod_aluno_disc;
    }
```

```
  public int getPeso() {
      return peso;
  }


  public void setPeso(int peso) {
      this.peso = peso;
  }

}
```

```
package notas;


public class rank1 {
     private int cod_aluno;
     private String nome;
     private double media1;

private void rank1 (int cod_aluno,String nome, double media1) {
        this.cod_aluno = cod_aluno;
        this.nome = nome;
        this.media1= media1;
        }


    public int getCod_aluno() {
        return cod_aluno;
    }
public void setCod_aluno(int cod_aluno) {
        this.cod_aluno = cod_aluno;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public double getMedia1() {
        return media1;
    }

    public void setMedia1(double media1) {
        this.media1 = media1;
    }



}
```

```
package notas;


public class inserePeso {

    int peso;
public int inserirPeso(String cod_disc ){


    if (cod_disc.equals("ASLNCUE016")){
     int peso=1;
     return peso;
    } else if (cod_disc.equals("ASL98140")){
     int peso=1;
     return peso;
     } else if (cod_disc.equals("ASL98110")){
    int peso=3;
    return peso;
se if (cod_disc.equals("ASL98123")){ int
peso=2;
return peso;
se if (cod_disc.equals("ASL98112")){ int
peso=1;
return peso;
se if (cod_disc.equals("ASL98141")){ int
peso=1;
return peso;
se if (cod_disc.equals("ASL98139")){ int
peso=2;
return peso;
lse if (cod_disc.equals("ASL98103")){ int
peso=2;
return peso;
se if (cod_disc.equals("ASL98122")){ int
peso=3;
return peso;
se if (cod_disc.equals("ASLNCUE041")){ int
peso=1;
return peso;
se if (cod_disc.equals("ASL98109")){ int
peso=2;
return peso;
se if (cod_disc.equals("ASLNCUE093")){ int
peso=1;
```

```
return peso;
se if (cod_disc.equals("ASL98231")){ int
peso=3;
return peso;
se if (cod_disc.equals("ASLNCUE012")){ int
peso=2;
return peso;
se if (cod_disc.equals("ASL98246")){ int
peso=1;
return peso;
se if (cod_disc.equals("ASLNCUE019")){ int
peso=1;
return peso;
e if (cod_disc.equals("ASL98220")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98245")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98240")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98244")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98238")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98107")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98247")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98243")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98242")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98233")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98248")){
```

```
int peso=1; return
peso;
}else if (cod_disc.equals("ASL98239")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98358")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98341")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98450")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98353")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98361")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASL98342")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASLNCUE008")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98218")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98350")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98359")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98348")){ int
peso=3;
return peso;
}else if (cod_disc.equals("ASL98102")){ int
peso=1;
return peso;
}else if (cod_disc.equals("ASL98347")){ int
peso=3;
return peso;
```

```
}else if (cod_disc.equals("ASL98360")){ int
peso=2;
return peso;
}else if (cod_disc.equals("ASLNCUE066")){ int
peso=1;
return peso;
         else if (cod_disc.equals("ASL98699")){
int peso=1;
return peso;
}         else if (cod_disc.equals("ASL98352")){
int peso=2;
return peso;
}         else if (cod_disc.equals("ASL98340")){
int peso=2;
return peso;
}         else if (cod_disc.equals("ASL98577")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98460")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98678")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98461")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98565")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98467")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98466")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98462")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98702")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98564")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98684")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98574")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98579")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98566")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98682")){
```

```
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98571")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98570")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98465")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98458")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98578")){
int peso=1;return peso;
}         else if (cod_disc.equals("ASL98701")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98683")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98688")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98694")){
int peso=3;return peso;
}         else if (cod_disc.equals("ASL98696")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98689")){
int peso=1;return peso;
}         else if (cod_disc.equals("ASL98697")){
int peso=2;return peso;
}         else if (cod_disc.equals("ASL98679")){
int peso=3;return peso;
}
       else {System.out.println("não encontrado"); }
        return peso;
    }
}
```

APÊNDICE B – PESQUISA DE LEVANTAMENTO DE DEMANDA PARA VERIFICAR A OPINIÃO DAS PRAÇAS ESPECIAIS E OFICIAS SOBRE A FERRAMENTA COMPUTACIONAL PARA CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO.

PERFIL DO ENTREVISTADO

A QUAL CÍRCULO HIERÁRQUICO VOCÊ PERTENCE:

( ) Oficiais

( ) Praças Especiais

VISÃO DO ENTREVISTADO

01. A qual círculo hierárquico você pertence:

( ) Oficiais

( ) Praças Especiais

02. Na sua opinião, o processo de cálculo da antiguidade seria mais eficiente e seguro se fosse realizado sem a intervenção humana?

( ) Sim

( ) Não

03. Sobre processos judiciais, acerca deste tema, em sua opinião, essa demanda diminuiria?

( ) Sim

( ) Não

APÊNDICE C – TABULAÇÃO DE DADOS DA PESQUISA

PERFIL DO ENTREVISTADO	
A QUAL CÍRCULO HIERÁRQUICO VOCÊ PERTENCE:	**QUANTIDADE DE RESPOSTAS OBTIDAS**
OFICIAIS	30
PRAÇAS ESPECIAIS	41
VISÃO DO ENTREVISTADO	
A QUAL CÍRCULO HIERÁRQUICO VOCÊ PERTENCE:	**QUANTIDADE DE RESPOSTAS OBTIDAS**
OFICIAIS	30
PRAÇAS ESPECIAIS	41
NA SUA OPINIÃO, O PROCESSO DE CÁLCULO DA ANTIGUIDADE SERIA MAIS EFICIENTE E SEGURO SE FOSSE REALIZADO SEM A INTERVENÇÃO HUMANA?	**QUANTIDADE DE RESPOSTAS OBTIDAS**
SIM	66
NÃO	5
SOBRE PROCESSOS JUDICIAIS, ACERCA DESTE TEMA, EM SUA OPINIÃO, ESSA DEMANDA DIMINUIRIA?	**QUANTIDADE DE RESPOSTAS OBTIDAS**
SIM	57
NÃO	14

ANEXOS

ANEXO A - PORTARIA Nº 06/2017/ABMJM/CBMMA – PESOS DAS DISCIPLINAS DO CFO BM

ESTADO DO MARANHÃO
CORPO DE BOMBEIROS MILITAR DO MARANHÃO
DIRETORIA DE ENSINO E PESQUISA
ACADEMIA DE BOMBEIRO MILITAR "JOSUÉ MONTELLO"

PORTARIA Nº 06/2017/ABMJM/CBMMA

O COMANDANTE DA ACADEMIA DE BOMBEIROS MILITAR "JOSUÉ MONTELLO", no uso de suas atribuíções legais e,

Considerando que a cada 05 (cinco) anos o Projeto Pedagógico do Curso de Formação de Oficiais Bombeiros Militar é revisado pelo Conselho Estadual de Educação do Maranhão;

Considerando que no ano de 2014 foi aprovado o atual Projeto Pedagógico do Curso de Formação de Oficiais Bombeiro Militar;

Considerando que no atual Projeto foram feitas mudanças na estrutura curricular, com inserção e supressão de disciplinas;

Considerando que para o cálculo da antiguidade do CFO - BM usa-se um peso para cada disciplna, e que no ano de 2014 foram atribuídos pesos as novas disciplinas do Projeto Pedagógico atual, porém não houve a publicação desses pesos. Para tanto,

R E S O L V E:

Art. 1º - Tornar público a manutenção das disciplinas que mudaram apenas a nomenclatura, conforme abaixo:

Projeto 2009	Projeto 2014
Administração de Empresas	Introdução a Administração
Química	Química Aplicada
Teoria Econômica	Economia
Calculo Diferencial e Integral	Cálculo Diferencial e Integral Univariável
Termodinâmica	Termodinâmica aplicada
Eletricidade	Eletricidade Aplicada

Inglês	Língua inglesa
Informática	Informática aplicada a projetos
Prática esportiva	Prática desportiva
Administração Financeira e Orçamento Público	Execução Financeira e Orçamentária
Prática Processual	Prática de Processo Administrativo
Mecânica dos Fluídos – Hidráulica	Mecânica dos Fluidos Aplicada

Art. 2º - Dar publicidade ao ato da relação de pesos das Disciplinas inseridas no Projeto Pedagógico 2014, segundo abaixo:

ORD.	DISCIPLINA	PESO
1	Gestão de Pessoas	1
2	Gestão de Operações e Logística	3
3	Métodos e Técnicas de pesquisa	1
4	Língua brasileira de sinais – Libras	1
5	Metodologia do Ensino	1
6	Salvamento Veicular	3
7	Mergulho Autônomo Básico	3
8	Direito Civil	2
9	Geografia física	1
10	Processo Administrativo	1

Art. 3° - Tornar público todos os pesos que estão sendo utilizados nas disciplinas constantes no Projeto Pedagógico do Curso de Formação de Oficiais Bombeiros Militar desde 2014, conforme descrito em seu ANEXO ÚNICO.

Publique-se e Cumpra-se.

ACADEMIA DE BOMBEIROS MILITAR "JOSUÉ MONTELLO", EM SÃO LUÍS - MA, AOS VINTE E UM DIAS DO MÊS DE SETEMBRO DE DOIS MIL E DEZESSETE.

José Raimundo **Costa Filho** – TC QOCBM
Comandante da ABMJM

ANEXO ÚNICO

RELAÇÃO GERAL DE PESOS DAS DISCIPLINAS CONSTANTES NO PROJETO PEDAGÓGICO/2014 DO CFO/BM

ORD.	DISCIPLINAS OBRIGATÓRIAS	C.H.	PESO	PERÍODO
1	Introdução a Administração	60	1	1º
2	Introdução ao Estudo do Direito	60		
3	Sociologia	60		
4	Química Aplicada	60		
5	Leitura e Produção Textual	60		
6	Armamento e Tiro	60		
7	Metodologia Científica	60	2	
8	Anatomia e Fisiologia Humana	60		
9	Introdução ao TFM	60		
10	Ordem Unida Básica	60		
11	Regulamento Aplicado ao CBMMA	45		
12	Tecnologia e Maneabilidade de Incêndio	90	3	
13	Atendimento Pré Hospitalar	90		
14	História da Corporação	45		
1	Economia	60	1	2°
2	Ciências Política	60		
3	Psicologia	60		
4	Gestão de Pessoas	60		
5	Física Geral	60		
6	Calculo Diferencial e Integral Univariável	60		
7	Direitos Humanos	45		
8	Etiqueta Social	45		
9	Direito Constitucional	60	2	
10	Correspondência Militar	45		
11	Telecomunicação Militar	45		
12	Legislação Bombeiro Militar	60		
13	Salvamento Terrestre	90	3	
14	Salvamento Aquático	90		

1	Desenho Técnico	60	1	**3º**
2	Estatística	60		
3	Calculo Vetorial	60		
4	Termodinâmica Aplicada	60		
5	Direito Penal Comum e Militar	60		
6	Eletricidade Aplicada	60	2	
7	Ordem Unida de Comando	60		
8	Treinamento Físico Militar	60		
9	Motomecanização	45		
10	Gestão de Operação e Logística	60	3	
11	Combate a Incêndio Florestal	60		
12	Estratégia e tática de Combate a Incêndio	60		
13	Salvamento em Altura	90		
14	Produtos Perigosos	60		
1	Direito Ambiental	60	2	**4º**
2	Ética e Cidadania	60		
3	Mecânica dos Fluídos Aplicada	60		
4	Higiene e Segurança do Trabalho	60		
5	Resistência dos Materiais	60		
6	Perícias de Incêndio e Explosões	45		
7	Chefia e Liderança	60		
8	Segurança Física das Instalações	60		
9	Treinamento Físico Militar Aplicado	60		
10	Defesa Civil	60	3	
11	Sobrevivência em Regiões Inóspitas	120		
1	Métodos e Técnicas de Pesquisa	60	1	
2	Mecânica Técnica	60	2	**5º**
3	Práticas Desportivas	60		
4	Administração Bombeiro Militar	45		
5	Gestão de Projetos	60		

6	Processo Penal Comum e Militar	60		
7	Estágio Supervisionado	90		
8	Direito Administrativo	60		
9	Comando em Operações Bombeiro Militar	60	3	
10	Atividades Aeroportuárias	45		
1	Defesa Pessoal	60	1	**6º**
2	Execução Financeira e Orçamentária	60	2	
3	Prática de Processo Administrativo	60		
4	Inteligência e Contra Inteligência	60		
5	Análise de Projeto	60	3	
6	Atendimento Pré-Hospitalar Aplicado	60		
7	TCC	-		

ORD.	DISCIPLINAS OPTATIVAS	C.H.	PESO
1	Língua Inglesa Instrumental	60	1
2	Língua brasileira de sinais – Libras	60	1
3	Informática Aplicada a Projetos	60	1
4	Metodologia do Ensino	60	1
5	Salvamento Veicular	60	3
6	Mergulho Autônomo Básico	60	3
7	Geografia Física	60	1
8	Criminalística Aplicada	60	1
9	Processo Administrativo	60	1
10	Direito Civil	60	2

José Raimundo **Costa Filho** – TC QOCBM
Comandante da ABMJM

ANEXO B - DECLARAÇÃO DE ORIGINALIDADE

DECLARAÇÃO DE ORIGINALIDADE

1. Eu, Aspirante a Oficial BM, aluno do CFO – BM, Felipe do Carmo Sousa declaro para todos os fins que meu trabalho de fim de curso intitulado DESENVOLVIMENTO DE UMA FERRAMENTA COMPUTACIONAL PARA O CÁLCULO DO COEFICIENTE CLASSIFICATÓRIO DO CURSO DE FORMAÇÃO DE OFICIAIS BM é um documento original elaborador e produzido por mim.

Dados do Orientador:

Nome/Grau/Hierarquia: Prof. Dr. Mauro Sérgio Silva Pinto

Filiação/Instituição: Universidade Estadual do Maranhão

E-mail: maurossp@yahoo.com.br

Telefones: +55 98 98896-4585

DISCENTE

CPF: 064.678.913-92