



UNIVERSIDADE FEDERAL DO AMAZONAS
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

IsenSys - Processador de Solicitações de Isenção de Taxa de Inscrição em Concursos Públicos

Felipe André Souza da Silva

Manaus - AM
Novembro de 2023

Felipe André Souza da Silva

IsenSys - Processador de Solicitações de Isenção de Taxa de Inscrição em Concursos Públicos

Monografia de Graduação apresentada à Faculdade de Tecnologia da UFAM como requisito parcial para a obtenção do grau de bacharel em Engenharia da Computação.

Orientador:

Dr. Edson Nascimento Silva Júnior

Universidade Federal do Amazonas

Manaus - AM

Novembro de 2023

Ficha Catalográfica


Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S729i Souza da Silva, Felipe André
IsenSys - Processador de solicitações de isenção de taxa de
inscrição em concursos públicos / Felipe André Souza da Silva .
2023
80 f.: il. color; 31 cm.


Orientador: Edson Nascimento Silva Júnior
TCC de Graduação (Engenharia da Computação) - Universidade
Federal do Amazonas.

1. taxa de inscrição. 2. isenção. 3. concurso público. 4. Java. I.
Silva Júnior, Edson Nascimento. II. Universidade Federal do
Amazonas III. Título


Monografia de Graduação sob o título *IsenSys - Processador de Solicitações de Isenção de Taxa de Inscrição em Concursos Públicos* apresentada por Felipe André Souza da Silva e aceita pela Faculdade de Tecnologia da Universidade Federal do Amazonas, sendo aprovada por todos os membros da banca examinadora abaixo especificada:

Documento assinado digitalmente
 **EDSON NASCIMENTO SILVA JÚNIOR**
Data: 13/11/2023 12:25:28-0300
Verifique em <https://validar.iti.gov.br>

Dr. Edson Nascimento Silva Júnior
Instituto de Computação
Universidade Federal do Amazonas

Documento assinado digitalmente
 **FABIOLA GUERRA NAKAMURA**
Data: 13/11/2023 12:34:04-0300
Verifique em <https://validar.iti.gov.br>

Dra. Fabíola Guerra Nakamura
Instituto de Computação
Universidade Federal do Amazonas

Documento assinado digitalmente
 **JOSE FRANCISCO DE MAGALHAES NETTO**
Data: 14/11/2023 16:21:19-0300
Verifique em <https://validar.iti.gov.br>

Dr. José Francisco Magalhães Netto
Instituto de Computação
Universidade Federal do Amazonas

Manaus - AM, 08 de Novembro de 2023.

À minha mãe, mulher guerreira e fonte de inspiração e forças para conclusão deste curso de graduação.

Agradecimentos

Em primeiro lugar quero agradecer à minha família que é pequena, mas super unida, minha mãe Adriana, aos meus irmãos Anna Beatriz e Maurício Júnior (*in memoriam*) e nossa amiga Nira, pelo amor incondicional, apoio emocional e encorajamento constante que me proporcionaram durante todos esses anos. Vocês são minha âncora e minha maior motivação.

Aos meus grandes amigos: Alexsandro Evangelista, Arianne Kaist, Hítalo Viana, Leonardo Pinheiro, Maurianne Kaist, Milena Lizandra, Munhoz, Naiara, Nilba, Thiago Lima e Will que compartilharam comigo as alegrias, desafios e preocupações deste percurso, agradeço pela amizade, colaboração, apoio mútuo e por não me permitir enlouquecer.

Também expresso minha gratidão ao meu orientador, Prof. Edson Jr. pela orientação, PACIÊNCIA, dedicação e perseverança ao longo deste processo. Suas orientações, *insights* e sua motivação foram fundamentais para o desenvolvimento deste trabalho e para o meu crescimento como acadêmico e profissional.

Por último, mas não menos importante, quero agradecer a todos os membros do corpo docente, distribuídos nos mais diversos departamentos por onde passei como estudante, cujo conhecimento e ensinamentos moldaram minha jornada acadêmica. Suas aulas foram inspiradoras e enriquecedoras, contribuindo muito para meu crescimento acadêmico.

A conclusão deste ciclo não teria sido possível sem a contribuição de cada uma dessas pessoas, e por isso, expresso minha profunda gratidão a todos. O conhecimento adquirido e as experiências vividas durante esta jornada acadêmica são inestimáveis, e estou ansioso para aplicá-los em meu futuro profissional.

Muito obrigado a todos por fazerem parte desta conquista!

Software is a great combination between artistry and engineering.

Bill Gates

IsenSys - Processador de Solicitações de Isenção de Taxa de Inscrição em Concursos Públicos

Autor: Felipe André Souza da Silva

Orientador: Dr. Edson Nascimento Silva Júnior

Resumo

Este documento versa sobre o desenvolvimento de um aplicativo computacional para processar solicitações de isenção de taxa de inscrição em concursos públicos de acordo com as normativas do Ministério do Desenvolvimento Social do Brasil e os interesses da Comissão Permanente de Concursos da UFAM. O sistema, que opera coletivamente com o Sistema de Isenção de Taxa de Concurso, do Ministério do Desenvolvimento Social do Brasil, permite que um órgão gestor prepare dados pessoais de candidatos solicitantes de isenção de taxa de inscrição para envio ao sistema do Ministério do Desenvolvimento e, após o processamento de tais solicitações pelo sistema, gere documentos de publicação de resultados e relatórios com o objetivo de garantir a lisura e transparência deste processo tão democrático. No desenvolvimento foi utilizada a linguagem de programação *Java* e tecnologias de grande consolidação no mercado como o *Jasper Reports*, para geração de relatórios e o *Apache POI*, adicionando suporte a arquivos do *Microsoft Excel*.

Palavras-chave: taxa de inscrição, isenção, concurso público, Java.

IsenSys - A Processor of Public Examination Subscription Fee Requests for Exemption

Author: Felipe André Souza da Silva

Supervisor: Dr. Edson Nascimento Silva Júnior

Abstract

This document relates to the development of a computer application that facilitates applying for the waiving of fees when sitting public examinations. This is in accordance with the guidelines set by the Brazilian Ministry of Social Development and the requirements of the UFAM Permanent Commission for Examinations. The app system works with the System of Exemption Fees of the Ministry, allowing anyone concerned to handle the personal data of candidates applying for exemption, in order for it to be sent to the Ministry's system. After this process, it creates a public notice and reports, guaranteeing smoothness and transparency in the democratic process. *Java* programming language was used in the development of the app. Other compatible technologies, such as *Jasper Reports*, was used to generate reports and the *Apache POI*, for the processing of *Microsoft Excel* files.

Keywords: examination fee, exemption, public examination, Java.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Logomarca do <i>Java</i> | 19 |
| Figura 2 – Logo do gerente de projetos <i>Apache Maven</i> | 21 |
| Figura 3 – Logo da biblioteca <i>Apache POI</i> | 21 |
| Figura 4 – <i>Eclipse IDE</i> | 22 |
| Figura 5 – Suíte de desenvolvimento <i>JasperReports®</i> | 23 |
| Figura 6 – Fluxograma das Etapas de Isenção | 26 |
| Figura 7 – Diagrama de Atividades do Módulo de Envio | 29 |
| Figura 8 – Diagrama de Atividades do SISTAC | 30 |
| Figura 9 – Diagrama de Atividades do Módulo de Retorno Preliminar . . | 31 |
| Figura 10 – Diagrama de Atividades do Módulo de Retorno Definitivo . . . | 32 |
| Figura 11 – Diagrama de Casos de Uso do <i>IsenSys</i> | 33 |
| Figura 12 – Tela Inicial do <i>IsenSys</i> | 33 |
| Figura 13 – Tela de Configurações do Sistema | 34 |
| Figura 14 – Modelagem da entidade <i>Candidato</i> | 36 |
| Figura 15 – Diagrama de Classe de <i>CandidatoBuilder</i> | 36 |
| Figura 16 – Diagrama de Classe de <i>RowParseException</i> | 38 |
| Figura 17 – Diagrama de Classe de <i>ParseResult</i> | 39 |
| Figura 18 – Diagrama de Classe de <i>CSVSheetReader</i> | 40 |
| Figura 19 – Diagrama de Classe de <i>ExcelSheetReader</i> | 41 |
| Figura 20 – Diagrama de Classe de <i>CSVSheetWriter</i> | 42 |
| Figura 21 – Diagrama de Classe de <i>ExcelSheetWriter</i> | 42 |
| Figura 22 – Diagrama do Fluxo de Dados das Solicitações | 43 |
| Figura 23 – Tela do Módulo de Envio | 44 |
| Figura 24 – Tela do Módulo de Envio (arquivo carregado) | 45 |
| Figura 25 – Tela do Módulo de Envio (preenchida) + Arquivos de Saída . . | 46 |
| Figura 26 – Modelagem da Entidade <i>Retorno</i> | 47 |
| Figura 27 – Diagrama de classe de <i>ListaRetornos</i> | 47 |
| Figura 28 – Diagrama de Classe de <i>Compilacao</i> | 48 |
| Figura 29 – Modelagem da Entidade <i>Situacao</i> | 48 |
| Figura 30 – Diagrama de Classe de <i>SituacaoDAO</i> | 48 |

| | |
|---|----|
| Figura 31 – Diagramas de Classe dos Geradores de Relatório | 49 |
| Figura 32 – Diagrama do Fluxo de Dados no Retorno Preliminar | 50 |
| Figura 33 – Tela do Módulo de Retorno Preliminar | 51 |
| Figura 34 – Tela do Módulo de Retorno Preliminar (preenchida) | 52 |
| Figura 35 – Arquivos de Saída do Módulo de Retorno Preliminar | 52 |
| Figura 36 – Diagrama do Fluxo de Dados no Retorno Definitivo | 53 |
| Figura 37 – Tela do Módulo de Retorno Definitivo | 54 |
| Figura 38 – Diagrama de Classe de uma Similaridade | 55 |
| Figura 39 – Diagrama de Classe de <i>JaroWinkler</i> | 56 |
| Figura 40 – Diagrama do Fluxo de Dados no Relatório | 56 |
| Figura 41 – Diagrama Comparativo entre o SISTAC e IsenSys | 60 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 – Dados pessoais de candidatos e seus formatos | 35 |
| Tabela 2 – Tempos de Processamento de Dados [Preliminar] | 58 |
| Tabela 3 – Tempos de Processamento de Dados [Definitivo] | 59 |
| Tabela 4 – Comparativo de Tempo de Processamento do Arquivo de Envio | 60 |

Lista de abreviaturas e siglas

| | |
|----------|---|
| CadÚnico | Cadastro Único para Programas Sociais do Governo Federal |
| CNPJ | Cadastro Nacional de Pessoas Jurídicas |
| CSV | <i>Comma-separated values</i> - valores separados por vírgula |
| COMPEC | Comissão Permanente de Concursos da UFAM |
| IDE | <i>Integrated Development Environment</i> - ambiente de desenvolvimento integrado |
| IRPF | Imposto sobre a Renda das Pessoas Físicas |
| IRPJ | Imposto sobre a Renda das Pessoas Jurídicas |
| NIS | Número de Identificação Social |
| REDOME | Registro Brasileiro de Doadores Voluntários de Medula Óssea |
| SISTAC | Sistema de Isenção de Taxa de Concurso |
| UFAM | Universidade Federal do Amazonas |

Sumário

| | | |
|------------|--|-----------|
| 1 | INTRODUÇÃO | 16 |
| 1.1 | Objetivos | 17 |
| 1.1.1 | Objetivo Geral | 17 |
| 1.1.2 | Objetivos Específicos | 17 |
| 1.2 | Organização da Monografia | 17 |
| 2 | REFERENCIAL TEÓRICO | 19 |
| 2.1 | Java | 19 |
| 2.2 | Apache Commons Text™ | 20 |
| 2.3 | Apache Maven | 20 |
| 2.4 | Apache POI | 21 |
| 2.5 | Eclipse IDE | 21 |
| 2.6 | JasperReports® | 22 |
| 2.7 | Jaro-Winkler | 23 |
| 2.8 | Joda-Time | 24 |
| 2.9 | Processo de Isenção | 25 |
| 3 | DESENVOLVIMENTO DO ISENSYS | 27 |
| 3.1 | Tela Inicial | 33 |
| 3.2 | Tela de Configurações do Sistema | 34 |
| 3.3 | Desenvolvimento do Módulo de Envio | 35 |
| 3.3.1 | Modelagem de um Candidato | 35 |
| 3.3.2 | A classe <i>CandidatoBuilder</i> | 36 |
| 3.3.3 | A classe de exceção <i>FieldParseException</i> | 38 |
| 3.3.4 | A classe de exceção <i>RowParseException</i> | 38 |
| 3.3.5 | A classe <i>ParseResult</i> | 38 |
| 3.3.6 | Importadores de Dados de Candidato | 39 |
| 3.3.6.1 | O importador <i>CSVSheetReader</i> | 40 |
| 3.3.6.2 | O importador <i>ExcelSheetReader</i> | 41 |
| 3.3.7 | Exportadores de Dados | 41 |

| | | |
|------------|--|-----------|
| 3.3.7.1 | O exportador <i>CSVSheetWriter</i> | 41 |
| 3.3.7.2 | O exportador <i>ExcelSheetWriter</i> | 42 |
| 3.3.8 | Integração do Módulo de Envio | 42 |
| 3.3.9 | Interface Gráfica do Módulo de Envio | 44 |
| 3.4 | Desenvolvimento dos Módulos de Retorno | 46 |
| 3.4.1 | Modelagem de um Retorno | 46 |
| 3.4.2 | A classe <i>ListaRetornos</i> | 47 |
| 3.4.3 | A classe <i>Compilacao</i> | 47 |
| 3.4.4 | Modelagem de uma Situacao | 48 |
| 3.4.5 | A classe <i>SituacaoDAO</i> | 48 |
| 3.4.6 | Geradores de Relatório | 49 |
| 3.4.7 | Integração dos Módulos de Retorno | 49 |
| 3.4.7.1 | Retorno Preliminar | 49 |
| 3.4.7.2 | Tela do Módulo de Retorno Preliminar | 50 |
| 3.4.7.3 | Retorno Definitivo | 53 |
| 3.4.7.4 | Tela do Módulo de Retorno Definitivo | 54 |
| 3.5 | Relatório de Similaridade | 55 |
| 3.5.1 | Modelagem de uma Similaridade | 55 |
| 3.5.2 | A classe <i>JaroWinkler</i> | 56 |
| 3.5.3 | Montagem do Relatório de Similaridade | 56 |
| 3.6 | Diretório de Recursos do <i>IsenSys</i> | 57 |
| 4 | IMPLEMENTAÇÃO E RESULTADOS | 58 |
| 5 | CONCLUSÕES | 61 |
| 5.1 | Considerações Finais | 61 |
| 5.2 | Limitações e Propostas de Atualizações | 61 |
| | REFERÊNCIAS | 63 |
| | ANEXO A – RESULTADO DEFINITIVO DA ISENÇÃO DO PSC | |
| | 2023 (APENAS ALGUMAS PÁGINAS) | 65 |

| | |
|---|-----------|
| ANEXO B – RELATÓRIO DE SIMILARIDADE APLICADO NO | |
| PSC 2023 (APENAS ALGUMAS PÁGINAS) | 72 |
| ANEXO C – RELATÓRIO DE ESTATÍSTICAS DE PROCESSA- | |
| MENTO (APENAS ALGUMAS PÁGINAS) | 75 |

1 Introdução

Com a missão de cultivar o saber em todas as áreas do conhecimento por meio do ensino, pesquisa e da extensão, a Universidade Federal do Amazonas (UFAM) é uma das principais portas de entrada para o desenvolvimento pessoal e intelectual, contando com cerca de 29.000 alunos e 3.400 servidores distribuídos em seis *campi* ao redor do Estado do Amazonas, em 2023.

Tomando como objeto de estudo e inspiração para este trabalho, um setor específico desta universidade foi adotado: a Comissão Permanente de Concursos (COMPEC), que é um órgão suplementar responsável pela execução dos principais processos seletivos de graduação e concursos para provimento de cargos da universidade.

Uma das tarefas mais democráticas e delicadas executadas por este setor é o processo de isenção de pagamento de taxa de inscrição em concursos e processos seletivos.

Atualmente a COMPEC, como qualquer outro órgão do poder executivo do Brasil, adota três tipos de modalidades de isenção: por cadastro no Registro Brasileiro de Doadores Voluntários de Medula Óssea (REDOME), por comprovação de baixa renda e curso de nível médio de forma gratuita e por meio do Cadastro Único para Programas Sociais do Governo Federal (CadÚnico).

Um dos desafios enfrentados pela COMPEC é a gerência e correto processamento das solicitações de isenção, de forma a não prejudicar os candidatos, tampouco a imagem da UFAM e do funcionalismo público. Para ilustrar, apenas em 2023 a COMPEC realizou 10 concursos, mobilizando ao total 39.289 candidatos, dos quais 18.709 solicitaram isenção de taxa de inscrição, onde 8.947 deles tiveram o benefício concedido.

Inicialmente a COMPEC realizava todo o processo de forma artesanal, logo, com o intuito de automatizar e otimizar tal processo, este trabalho apresenta uma aplicação de computador capaz de analisar dados, processar e confeccionar relatórios para publicação, tomando como objeto de estudo a modalidade de isenção mais volumosa em termos de solicitação: a modalidade CadÚnico, regulamentada pelo Decreto nº 6.593, de 2 de outubro de 2008 [1].

A aplicação, denominada *IsenSys*, procura ainda fornecer uma interface simples e objetiva, com dicas e tratamentos de forma a instruir intuitivamente sua utilização ao usuário, tomando ainda como alicerce no seu desenvolvimento, os cinco princípios fundamentais da Administração Pública do Brasil: legalidade, impessoalidade, moralidade, publicidade e eficiência.

1.1 Objetivos

1.1.1 Objetivo Geral

Esta monografia possui como objetivo apresentar um aplicativo processador de solicitações de isenção de taxa de inscrição de acordo com a regulamentação do CadÚnico [1], de forma a permitir agilidade e acurácia nos resultados, por parte de um órgão gestor do Poder Executivo do Brasil.

1.1.2 Objetivos Específicos

- Implementar à risca o motor do sistema (*backend*) utilizando as normativas do manual do SISTAC [2], visando erradicar os erros de processamento dos lotes de envio;
- Utilizar das boas práticas de programação para tornar o projeto simples e permitir colaboração;
- Implementar uma interface gráfica simples e intuitiva, com dicas e tratamentos de exceções, visando agilidade e fluidez na interação com o sistema.

1.2 Organização da Monografia

Esta monografia possui a seguinte estrutura de capítulos:

- **Capítulo 2:** aborda os fundamentos teóricos, principais tecnologias e bibliotecas utilizadas no desenvolvimento do *IsenSys*;

- **Capítulo 3:** detalha o processo de desenvolvimento do *IsenSys*, por meio de diagramas de alto nível, detalhamento de funcionalidades e exibição da interface gráfica;
- **Capítulo 4:** descreve resultados e métricas que permitem ter um comparativo entre o antes e depois da implementação do *IsenSys*;
- **Capítulo 5:** apresenta as considerações finais, conclusão e futuras atualizações no *IsenSys*.

2 Referencial Teórico

Este capítulo referencia as principais ferramentas e tecnologias utilizadas no desenvolvimento do *IsenSys*, tais como linguagem de programação, bibliotecas de funções e o ambiente de desenvolvimento. A última seção deste capítulo [2.9] explica e detalha o funcionamento do processo de isenção adotado pela COMPEC.

2.1 Java

A linguagem de programação Java [3] é uma das mais bem conceituadas e utilizadas ao redor do mundo. Concebida em meados de 1995 pela empresa *Sun Microsystems*, tem conquistado o mundo pela sua simplicidade, boa curva de aprendizagem, forma de organização e versatilidade entre os vários sistemas operacionais e dispositivos.

Por ser uma linguagem independente de plataforma, ela permite que uma mesma aplicação possa ser executada em diversos equipamentos portando sistemas operacionais de diversas arquiteturas, sem a necessidade de adaptação ou reconstrução de código por parte do desenvolvedor, comportamento que torna suas aplicações escaláveis e robustas.

Figura 1 – Logomarca do *Java*



Fonte: <<https://www.oracle.com/br/java/technologies/java-se-glance.html>>

Atualmente mantida pela empresa *Oracle Corporation*, a linguagem continua sendo livre e gratuita para utilização pessoal e para algumas classes de aplicações, e possui uma rica e extensa comunidade de suporte e documentação. As atualizações regulares também são gratuitas e sempre trazem otimizações, novas funcionalidades e melhorias de segurança.

Provavelmente o leitor já tenha utilizado algumas das aplicações implementadas utilizando a tecnologia Java, programas como: declaração de imposto sobre a renda (IRPF e IRPJ), Processo Judicial Eletrônico (PJe), *MATLAB*, famoso no mundo da engenharia e até o próprio sistema *Android*, um dos principais sistemas operacionais para dispositivos móveis.

2.2 Apache Commons Text™

O *Apache Commons Text*™ [4] é uma biblioteca livre (*open-source*) escrita em linguagem Java, desenvolvida e mantida pela *Apache Software Foundation*, que é uma organização mundial sem fins lucrativos criada para dar suporte de desenvolvimento a projetos de programação de código aberto, ou seja, livres para todos.

A biblioteca oferece funcionalidades para manipulação e processamento de texto, tornando tarefas comuns de manipulação de strings mais eficientes e simples para desenvolvedores Java, algumas são: manipulação e escapamento de *Strings*, algoritmos de distância e similaridade, tokenização, geração de senhas aleatórias, formatação em geral, dentre muitas outras.

2.3 Apache Maven

Até o presente momento foram citadas algumas bibliotecas utilizadas como complemento de código para a linguagem Java, prática muito comum entre os desenvolvedores em qualquer linguagem de programação. Porém, muitas vezes sua gerência pode ser complicada se realizada de forma manual, pois bibliotecas resultam em arquivos, que podem sofrer corrompimento ou passar por atualizações de versão.

O *Apache Maven* [5] surge como uma ferramenta de gerenciamento e compreensão de projetos escritos em Java, onde é possível organizar bibliotecas e publicá-las de forma gratuita para utilização pela comunidade de desenvolvimento. Podemos esperar também funcionalidades como atualizações, verificação de integridade e autoinstalação.

A utilização deste gerente é muito simples, tendo em vista que muitas suítes de desenvolvimento oferecem suporte nativo ao *Maven*. Após instalada, basta organizar as dependências (bibliotecas) do projeto em um arquivo específico denominado 'pom.xml', salvar e esperar que o próprio *Maven* configure as bibliotecas no projeto.

Figura 2 – Logo do gerente de projetos *Apache Maven*



Fonte: <<https://maven.apache.org>>

2.4 Apache POI

O *Apache POI* [6] também é uma biblioteca gratuita escrita utilizando a linguagem Java, que adiciona suporte de leitura e escrita de dados em documentos do *Microsoft Office* diretamente das aplicações Java, sem a necessidade de aquisição e instalação dos aplicativos da *Microsoft*.

Figura 3 – Logo da biblioteca *Apache POI*



Fonte: <<https://poi.apache.org>>

2.5 Eclipse IDE

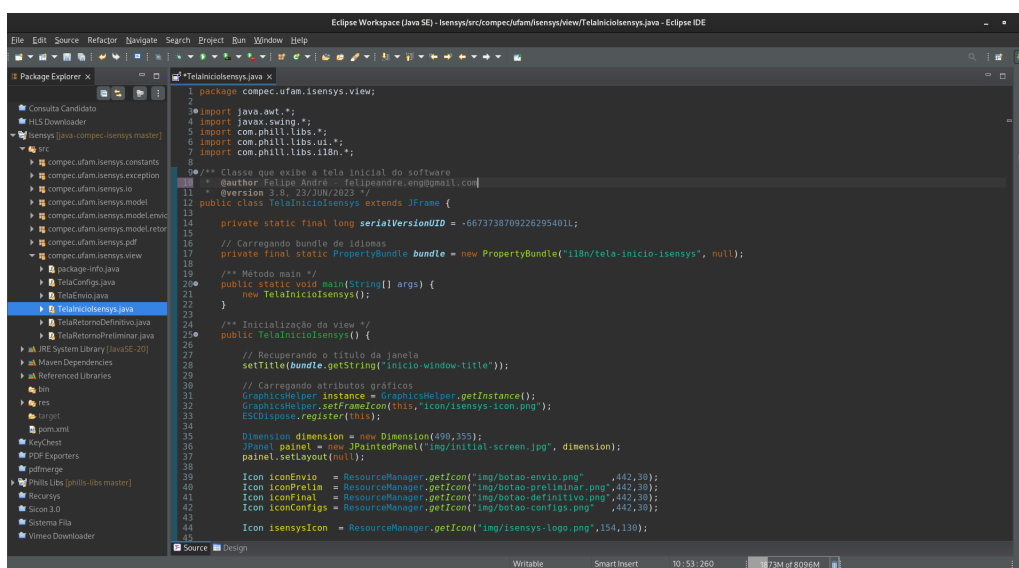
No mundo do desenvolvimento de software, agilidade e escalabilidade estão entre as qualidades mais requisitadas nas linguagens. Como forma de satisfazer tais demandas, existe no mercado uma vasta gama do que chamamos de *IDE - integrated development environment* ou ambiente de desenvolvimento integrado, amplamente aceitos pela comunidade.

Inicialmente o projeto *Eclipse* [7] foi concebido pensando no desenvolvimento utilizando a linguagem Java, mas atualmente suporta extensões (*plugins*)

que adicionam suporte a várias outras linguagens, tais como *PHP*, *Python*, *Arduino* etc e ainda, por ser de código aberto, pode ser modificado para outras finalidades.

O aplicativo adiciona funcionalidades que auxiliam muito no desenvolvimento em uma linguagem, tais como autocomplemento de código, exibição de sugestões, dicas e tratamentos de erros, verificador de sintaxe e semântica e até integração com outras tecnologias como versionadores de código e o *Apache Maven* [2.3].

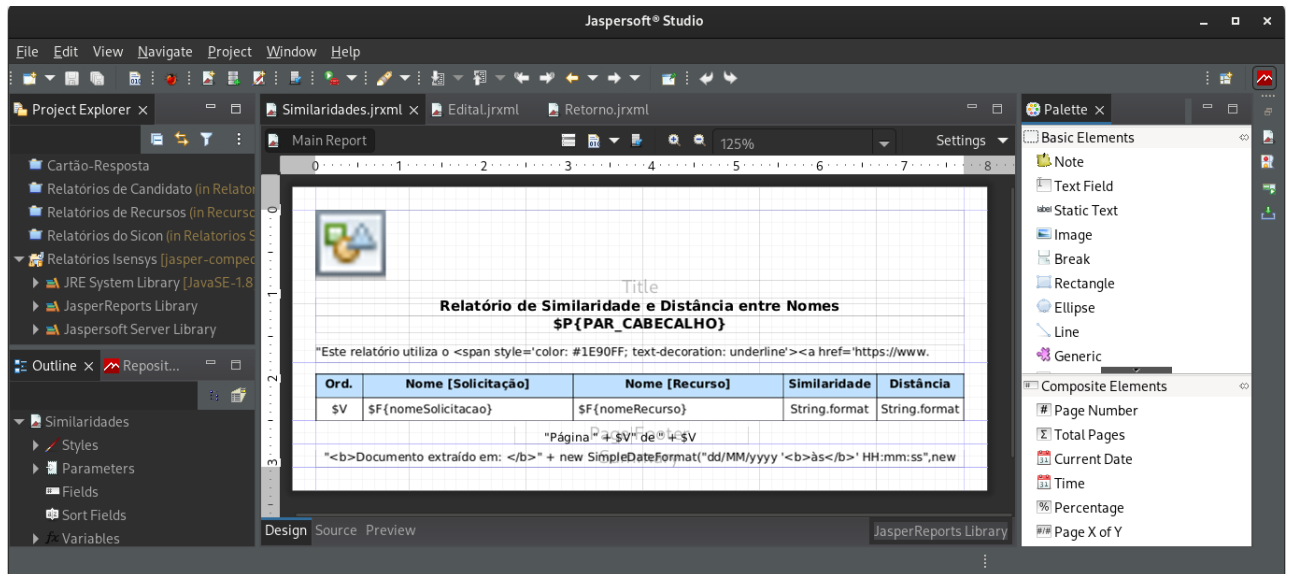
Figura 4 – Eclipse IDE



2.6 JasperReports®

A biblioteca *JasperReports*® [8] é gratuita e uma das mais robustas e populares utilizadas para criação de relatórios para aplicações escritas em Java. Possui sua própria suíte de design, onde o desenvolvedor pode rapidamente configurar um novo relatório com poucos passos, seguindo guias e instruções diretamente na interface gráfica do editor.

Por ser desenvolvida utilizando a linguagem Java, ela também apresenta o mesmo comportamento e visualização em qualquer sistema operacional onde está disponível, produzindo relatórios de alta qualidade e escalabilidade, oferecendo opções de exportação para outros formatos conhecidos, tais como *PDF*, *Web* e *Microsoft Word*.

Figura 5 – Suíte de desenvolvimento *JasperReports®*

2.7 Jaro-Winkler

O algoritmo de *Jaro-Winkler* [9] é uma técnica de comparação de strings que mede a semelhança entre duas sequências de caracteres. Ele é frequentemente utilizado para encontrar correspondências aproximadas entre strings, especialmente em aplicações de deduplicação de dados e pesquisa de registros. A seguir, encontra-se um resumo da implementação do algoritmo.

Dados: Duas strings, s_1 e s_2

Passo 1: Encontre o comprimento das strings:

$$\text{len}_1 = \text{comprimento}(s_1)$$

$$\text{len}_2 = \text{comprimento}(s_2)$$

Passo 2: Defina o valor de m como o número de caracteres em comum (correspondentes) entre as duas strings, considerando uma janela de proximidade:

$$\text{Janela} = \lfloor \max(\text{len}_1, \text{len}_2) / 2 \rfloor - 1$$

m = número de caracteres em comum dentro da janela

Passo 3: Calcule o número de transposições:

$$t = \frac{1}{2} \times (\text{número de transposições de caracteres entre } s_1 \text{ e } s_2)$$

Passo 4: Calcule a medida de similaridade de Jaro:

$$\text{Jaro_Sim} = \frac{m}{\text{len}_1} + \frac{m}{\text{len}_2} + \frac{m - t}{m}$$

Passo 5: Calcule o fator de ajuste de prefixo Winkler:

$$\text{Prefix_Factor} = 0$$

Prefix_Len = número de caracteres idênticos no início das strings

Se Prefix_Len > 4, limite Prefix_Len a 4.

$$\text{Prefix_Factor} = \text{Prefix_Len} \times 0.1 \times (1 - \text{Jaro_Sim})$$

Passo 6: Calcule a medida de similaridade de Jaro-Winkler:

$$\text{Jaro_Winkler_Sim} = \text{Jaro_Sim} + \text{Prefix_Factor}$$

Passo 7: A medida de similaridade final entre s_1 e s_2 é dada por Jaro_Winkler_Sim.

O valor resultante de Jaro_Winkler_Sim está dentro do intervalo [0, 1], onde 0 indica nenhuma semelhança e 1 indica correspondência perfeita. Quanto maior o valor, maior a semelhança entre as duas strings.

2.8 Joda-Time

A biblioteca *Joda-Time* [10] facilita praticamente qualquer operação e manipulação com datas em um sistema Java, tarefa que é deveras complicada utilizando apenas os recursos nativos da linguagem. Com uma boa documentação e as mais diversas classes para cálculos, formatação e operações com data, tem conquistado muitos desenvolvedores a utilizá-la em seus projetos.

Felizmente, *Joda-Time* é uma biblioteca livre para utilização pessoal e comercial, licenciada sobre a *Licença Apache 2.0*. Também é considerada consolidada e estável, estando até os dias atuais recebendo atualizações de otimização e novas funcionalidades.

2.9 Processo de Isenção

Compreender o processo de isenção de taxa de inscrição é parte fundamental para entender a importância e seriedade empenhados no desenvolvimento do *IsenSys*. Durante esta seção, serão abordadas de forma detalhada e objetiva as fases e o *modus operandi* do processamento das solicitações de isenção de taxa de inscrição, a ser referenciada no decorrer da monografia apenas como *solicitações de isenção* ou ainda *solicitações*.

A implementação do processo de isenção de taxa de inscrição a candidatos de concursos e processos seletivos por parte da COMPEC, segue as normativas impostas do Decreto nº 6.593, de 2 de outubro de 2008 [1] e no manual de envio e recebimento de solicitações de inscrição do SISTAC [2].

Como fonte de insumo e também como entrega do resultado final, o sistema denominado "PSConcursos", da UFAM foi utilizado. É ele quem concentra dados dos candidatos a concursos da universidade bem como solicitações de isenção e inscrição. O *IsenSys*, por sua vez, é capaz de importar as solicitações geradas pelo PSConcursos e realizar seu trabalho.

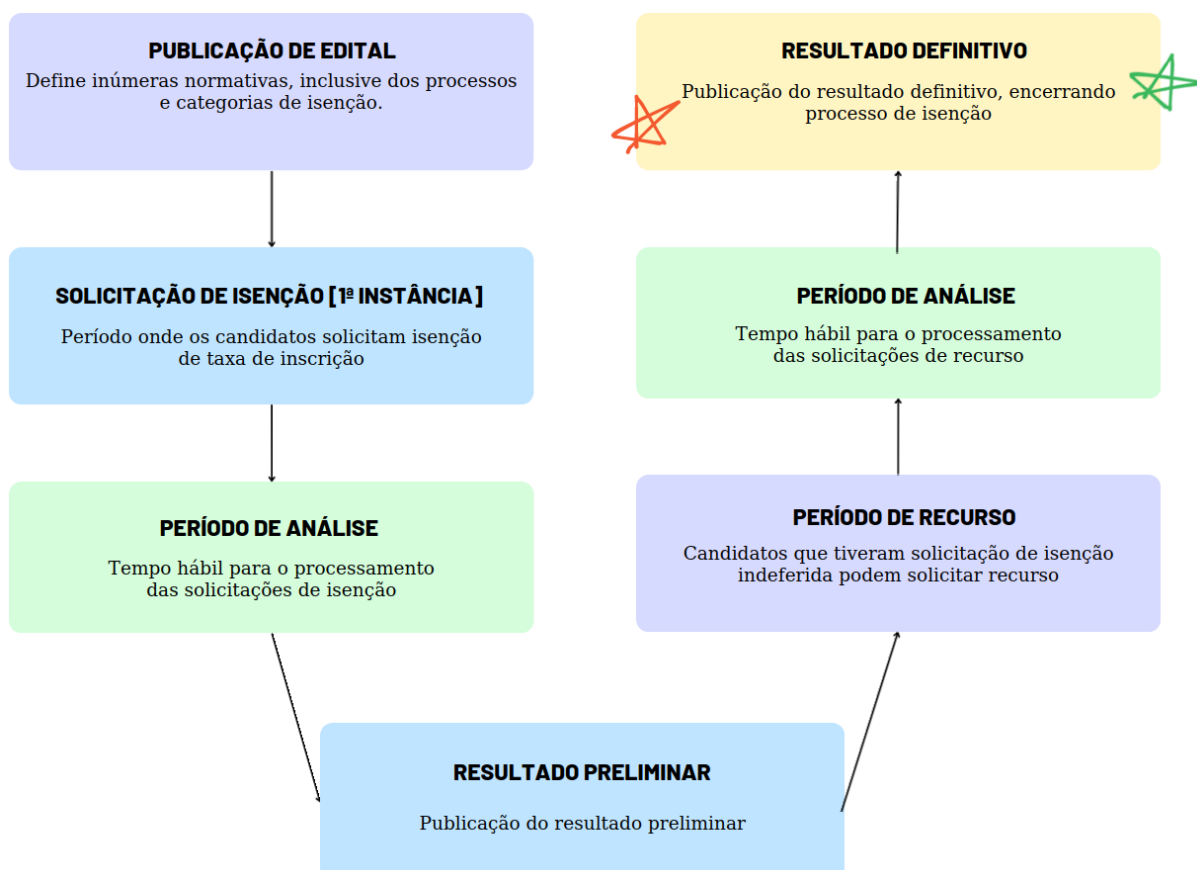
Em cada edital regido pela COMPEC, o processo de isenção é a primeira etapa a ser realizada e sua operacionalização respeita o seguinte fluxo:

1. **Publicação de Edital:** onde são definidas as regras gerais do concurso ou processo seletivo;
2. **Período de Solicitação de Isenção [1ª instância]:** geralmente dois dias úteis, onde os candidatos que têm interesse em solicitar isenção, o fazem por meio do preenchimento de um formulário de solicitação no sistema PSConcursos;
3. **Período de Análise:** é o tempo que a COMPEC possui para processar as solicitações, geralmente entre 3 a 4 dias úteis. O PSConcursos disponibiliza um arquivo *csv* com todas as solicitações obtidas no período editalício, insumo para envio ao SISTAC que, por sua vez, analisa as solicitações e retorna também um arquivo *csv*, útil para a confecção do resultado preliminar;

4. **Publicação do Resultado Preliminar:** aqui é tornado público o resultado do processamento das solicitações em 1ª instância;
5. **Período de Recurso [2ª instância]:** é a oportunidade que o candidato que teve sua solicitação indeferida em 1ª instância, recorrer, preenchendo um formulário de recurso no sistema PSConcursos. Vale ressaltar que só podem recorrer os candidatos que solicitaram isenção em 1ª instância!
6. **Período de Análise:** é o tempo que a COMPEC possui para processar os recursos, geralmente entre 2 a 3 dias úteis. Aqui é realizado o mesmo processo descrito no primeiro período de análise;
7. **Publicação do Resultado Definitivo:** publicação do resultado definitivo, juntamente com a carga de informações no PSConcursos, encerrando o processo de isenção.

O fluxograma a seguir ilustra de uma forma mais concisa este processo.

Figura 6 – Fluxograma das Etapas de Isenção



3 Desenvolvimento do *IsenSys*

Introduzidas as principais ferramentas utilizadas no projeto, aprofundar-se-á no processo de desenvolvimento do aplicativo *IsenSys*. Conceitos como especificações de sistema, requisitos para utilização e formas de aplicação serão demonstrados de maneira objetiva.

O desenvolvimento do motor do sistema (*backend*) é regido pelas normas de formato de arquivo e diretivas definidas no documento de orientações gerais do SISTAC [11] e no manual de envio e recebimento de arquivos [2], versão 10.0, publicada em 24/06/2016.

Os requisitos funcionais (RF) do sistema são:

- O sistema deverá permitir a importação de dados de candidatos através de arquivos do tipo *csv* ou planilhas do *Microsoft Excel* (RF01);
- O sistema deverá permitir o cadastro e edição de dados do órgão gestor (RF02);
- O sistema deverá realizar testes de integridade nos dados de solicitações importadas (RF03);
- O sistema deverá gerar arquivos de importação para o SISTAC (RF04);
- O sistema só pode entregar um arquivo de importação com os dados de solicitações em sua plena completude e integridade (RF05);
- Dados de solicitações que foram informadas de forma incompleta ou inválida serão considerados erros e deverão ser exportados em uma planilha (RF06);
- O sistema deverá permitir a importação do arquivo de retorno do SISTAC (RF07);
- A partir do arquivo de retorno do SISTAC, o sistema deverá produzir relatórios prontos para análise e publicação (RF08);

- O sistema também deve permitir a importação da planilha com erros, para inclusão nos relatórios públicos (RF09);
- O sistema deve gerar relatórios de estatísticas de solicitações deferidas e indeferidas (RF10);
- O sistema deverá gerar um relatório de similaridade entre nomes de candidatos recursantes (RF11).

Os requisitos não-funcionais (RF) do sistema são:

- A linguagem *Java Standard Edition (Java SE)* foi utilizada no desenvolvimento da aplicação (RNF01);
- A versão mínima da *Java Virtual Machine (JVM)* a ser utilizada é a 15 (RNF02);
- A suíte de interface gráfica utilizada é o *Java Swing* (RNF03);
- Por questões de simplicidade, todas as telas foram construídas utilizando layout absoluto com janela não redimensionável (RNF04);
- A versão do *Apache POI* utilizada é a 5.2.3, lançada em 17/09/2022 (RNF05);
- A versão do *JasperReports®* utilizada é a 6.26.0, lançada em 11/09/2023 (RNF06);
- O sistema pode ser executado em qualquer computador com suporte mínimo ao *Java Runtime Environment - JRE 15*, mínimo de 1GB de memória RAM disponível e monitor com resolução mínima de 800x600 (RNF07);
- Como limitação do *JRE 15*, somente máquinas com arquitetura de CPU de 64-bits são suportadas pelo sistema (RNF08).

O sistema conta com três módulos distintos: um dedicado a preparar os dados de candidatos para envio ao SISTAC e os outros dois dedicados a processar os arquivos de retorno do SISTAC e produzir relatórios de publicação preliminar e definitivo. A seguir estão ilustrados os diagramas de atividades do sistema, trazendo uma visão geral do funcionamento do *IsenSys*.

Figura 7 – Diagrama de Atividades do Módulo de Envio

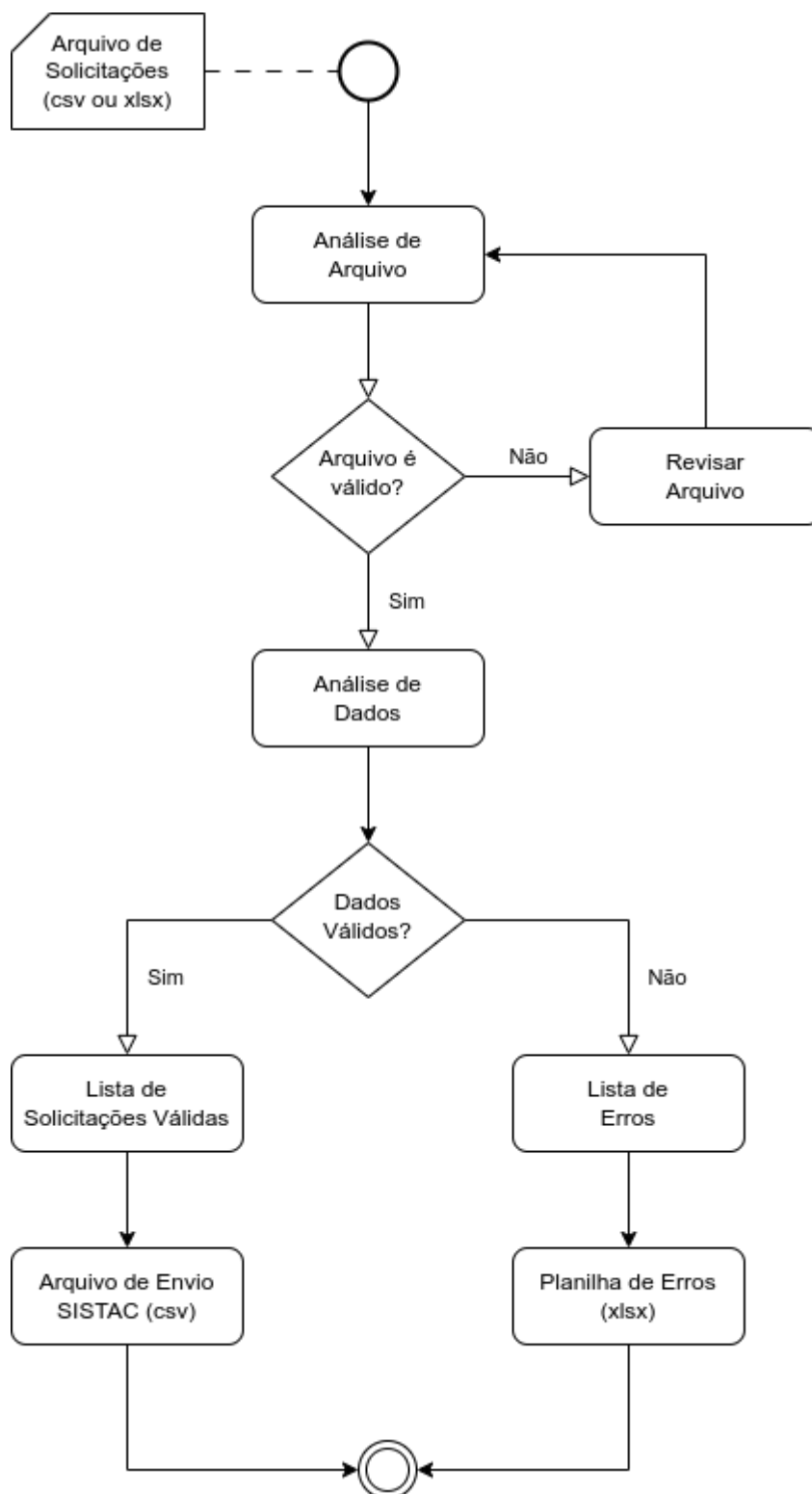


Figura 8 – Diagrama de Atividades do SISTAC

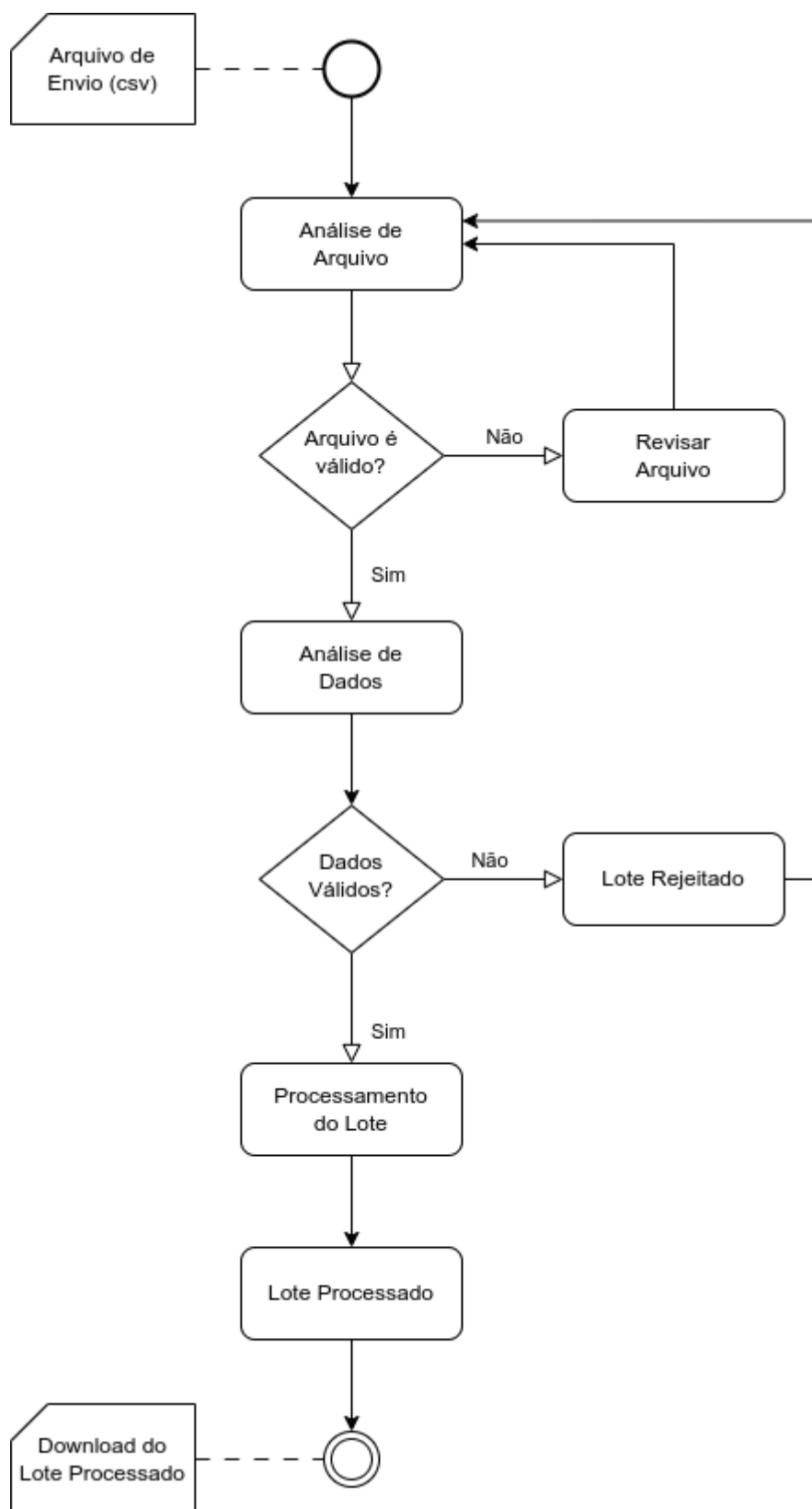


Figura 9 – Diagrama de Atividades do Módulo de Retorno Preliminar

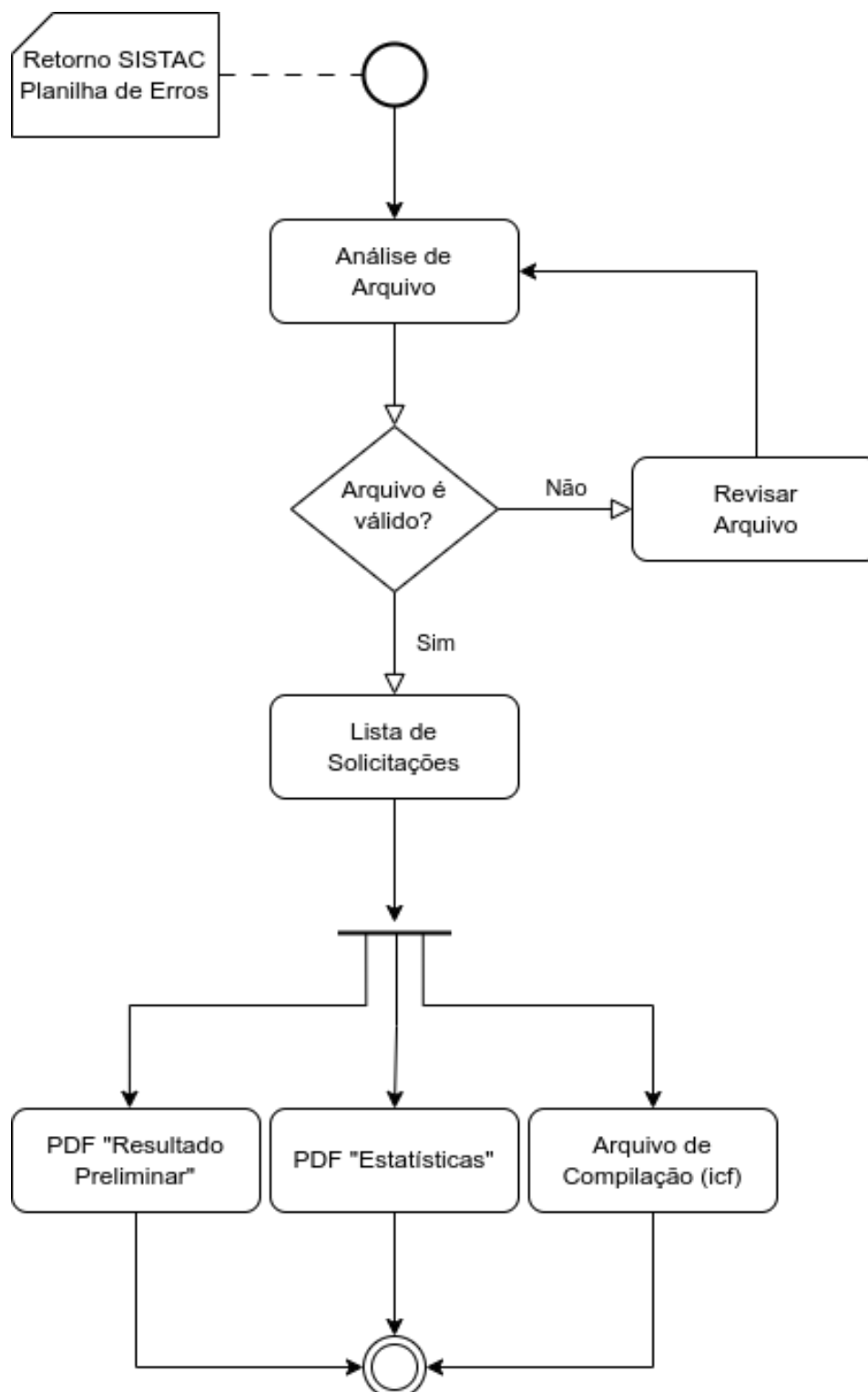


Figura 10 – Diagrama de Atividades do Módulo de Retorno Definitivo

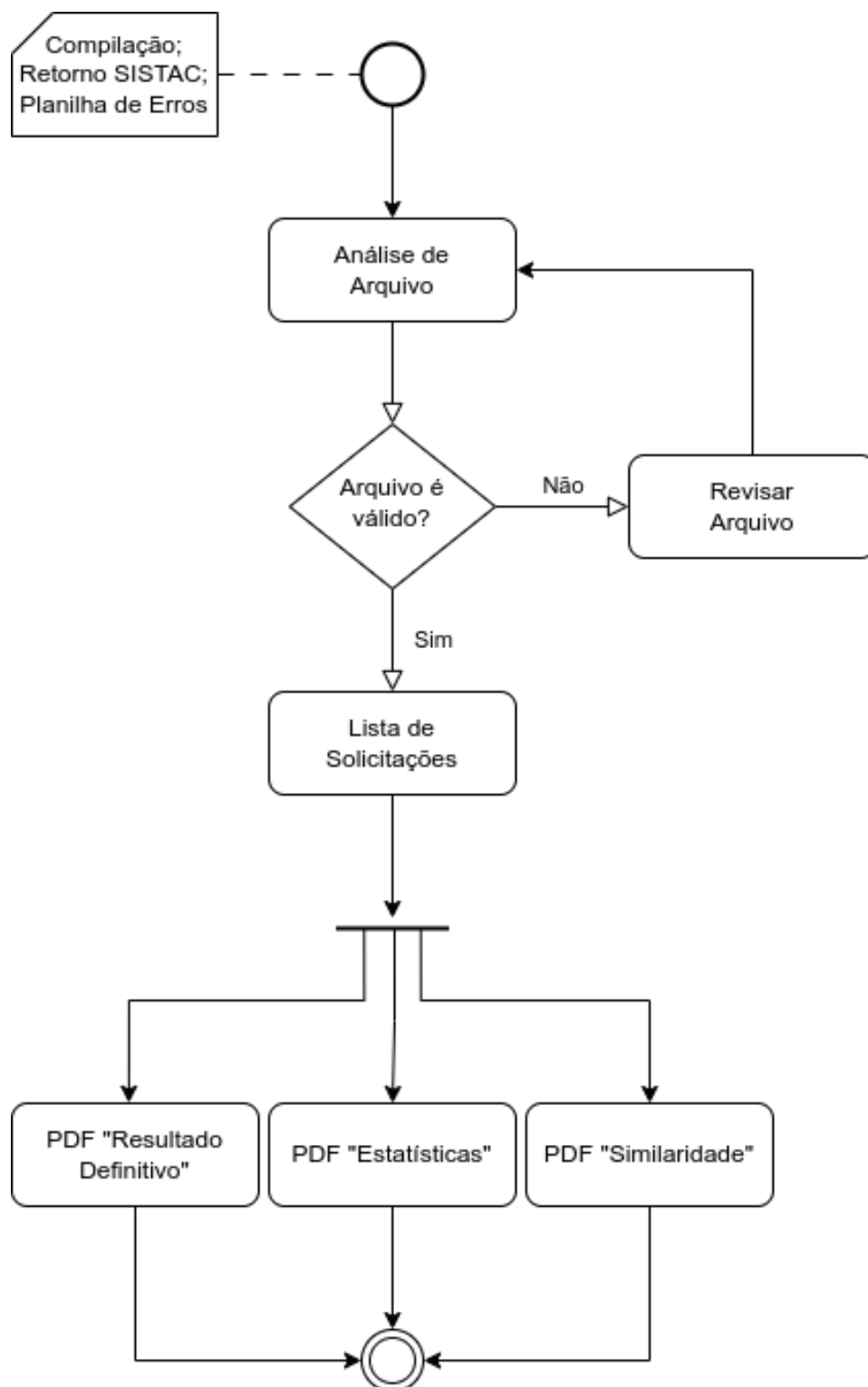
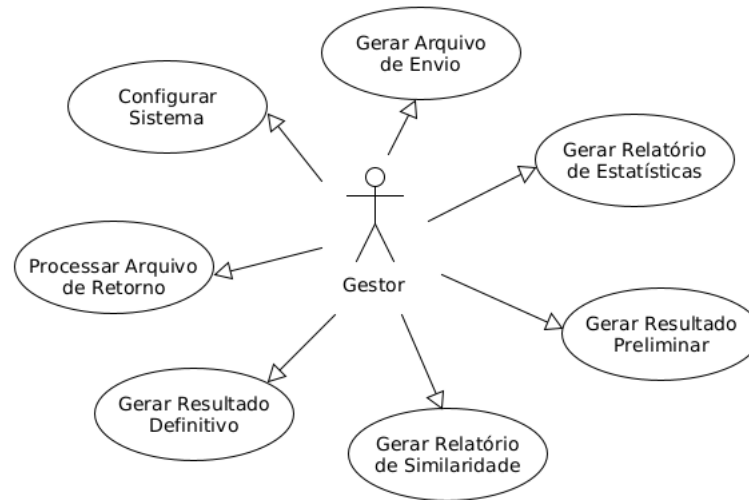
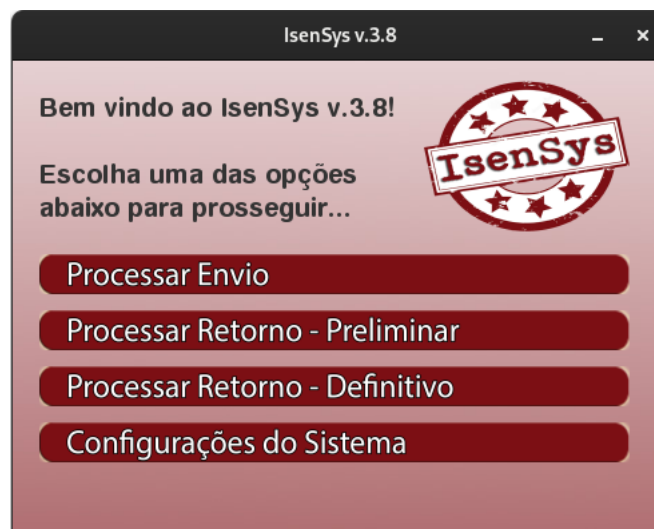


Figura 11 – Diagrama de Casos de Uso do *IsenSys*

As seções a seguir contém um estudo mais aprofundado sobre a implementação de cada módulo.

3.1 Tela Inicial

A tela inicial do *IsenSys* agrupa botões que dão acesso a todas as funcionalidades implementadas no sistema. Na primeira utilização é recomendável que o usuário configure os dados do órgão gestor, por meio do botão 'Configurações do Sistema'. A figura a seguir mostra a tela inicial do *IsenSys*.

Figura 12 – Tela Inicial do *IsenSys*


3.2 Tela de Configurações do Sistema

Algumas configurações são necessárias para o correto funcionamento do *IsenSys*. São elas:

- **CNPJ:** indica o número de CNPJ do órgão gestor;
- **Nome Fantasia:** indica o nome fantasia do órgão gestor, contendo no máximo 100 caracteres, segundo o manual do SISTAC [2];
- **Razão Social:** indica a razão social do órgão gestor, também contendo no máximo 100 caracteres, segundo o manual do SISTAC [2];
- **Índices da Planilha de Importação:** indica os índices dos campos de dados dispostos no arquivo de importação de solicitações. Tais índices começam em '0' e vão incrementando a cada coluna. Tomando como exemplo um arquivo de importação do *Microsoft Excel*, a coluna 'A', tem o índice '0', a coluna 'C' tem o índice '2' e assim sucessivamente. Todos os índices devem ser preenchidos.

As configurações são salvas em um arquivo binário contido no diretório de recursos do sistema [3.6], sob o caminho '*config/program.dat*'.

Figura 13 – Tela de Configurações do Sistema



IsenSys v.3.8 - Configurações do Sistema

Dados da Instituição

CNPJ:

Nome Fantasia:

Razão Social:

Índices da Planilha de Importação

| Nome | NIS | Dt. Nascimento | Sexo | RG | Data Emissão RG | Órgão Emissor RG | CPF | Nome da Mãe |
|------|-----|----------------|------|----|-----------------|------------------|-----|-------------|
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Ícones: Recarregar, Salvar

3.3 Desenvolvimento do Módulo de Envio

Esta seção tem por objetivo detalhar o desenvolvimento do motor (*backend*) de preparação de dados de solicitações de isenção para envio ao SISTAC. Primeiramente precisamos entender como está distribuído o fluxo de informações pelo sistema. Para isto, serão detalhadas as entidades principais e auxiliares deste módulo.

3.3.1 Modelagem de um Candidato

Os dados pessoais de um candidato são o objeto principal deste sistema, pois compõem uma solicitação de isenção que, após análise pelo SISTAC, tem uma resposta de deferimento ou não. Segundo o manual do SISTAC [2], os dados pessoais de candidatos necessários para o processamento estão dispostos na tabela a seguir.

Tabela 1 – Dados pessoais de candidatos e seus formatos

| Campo | Descrição | Máximo de Caracteres | Tipo | Formato |
|-----------------|---|-----------------------------|--------------|----------------|
| Nome | Nome completo do candidato sem caracteres especiais e sem abreviações | 100 | Texto | |
| NIS | Número de identificação social do candidato | 11 | Numérico | |
| Data de Nascim. | Data de nascimento do candidato | 8 | Numérico | ddmmaaaa |
| Sexo | Sexo do candidato | 1 | Texto | M ou F |
| RG | Número do Documento de Identidade do candidato | 16 | Alfanumérico | |
| Data de Emissão | Data de emissão do Documento de Identidade | 8 | Numérico | ddmmaaaa |
| Sigla RG | Sigla do órgão emissor do Documento de Identidade | 30 | Alfanumérico | |
| CPF | Número do CPF do candidato | 11 | Numérico | |
| Nome da Mãe | Nome completo da mãe do candidato, sem caracteres especiais e sem abreviações | 100 | Texto | |

De posse dos dados e tipos, foi concebida a modelagem da entidade *Candidato*, de acordo com o diagrama a seguir.

Figura 14 – Modelagem da entidade *Candidato*

| Candidato |
|---|
| +nome: String +nis: String +dataNascimento: DateTime +sexo: char +rg: String +dataEmissaoRG: DateTime +orgaoEmissorRG: String +cpf: String +nomeMae: String |
| |

De acordo com o diagrama de classe da entidade *Candidato*, nota-se que esta apenas armazena dados. As validações nos campos são garantidas por uma classe intermediária, detalhada na seção seguinte.

3.3.2 A classe *CandidatoBuilder*

Esta classe é responsável por montar um *Candidato* com os dados extraídos de um arquivo de entrada. Durante esse processo ela realiza uma série de validações nos dados, e caso haja pelo menos uma inconsistência, uma exceção com detalhes desta inconsistência é lançada, caso contrário, significa que foi possível construir um objeto respeitando todos os requisitos de formatos do SISTAC [2]. Segue o diagrama de classe do *CandidatoBuilder*.

Figura 15 – Diagrama de Classe de *CandidatoBuilder*

| CandidatoBuilder |
|---|
| |
| +build() +parseNome() +parseNIS() +parseData() +parseSexo() +parseRG() +parseOrgao() +parseCPF() |

Os métodos *parse...()*, são responsáveis por realizar validações específicas em cada campo de dados de um *Colaborador* e lançam exceções detalhadas

para que, posteriormente, tanto o órgão gestor dos dados quanto o candidato possam ter conhecimento de quais campos enviaram fora de formato e se ainda cabe algum recurso. Eis a metodologia implementada em cada campo de uma solicitação:

- **Nome [candidato e mãe]:** são removidos acentos, cedilhas, múltiplos espaços e quaisquer outros caracteres que não sejam alfabéticos. Se após o processamento, o nome for vazio ou possuir mais que 100 caracteres, um erro é gerado neste campo;
- **NIS:** o número de identificação social (NIS) pode ser verificado por meio de um algoritmo que utiliza os próprios dígitos de verificação do NIS para validá-lo. No *IsenSys*, esta implementação consta na biblioteca *Phills Libs* [12]. Caso o algoritmo não consiga validar o NIS, um erro é gerado nesse campo;
- **Data [nascimento e emissão de RG]:** se uma data vazia ou inválida for informada, a data de "01/01/2000" é inferida, logo, erros não são gerados nesse campo;
- **Sexo:** caso o sexo informado não seja 'M' ou 'F', um erro é gerado nesse campo;
- **Nº de RG:** são extraídos apenas caracteres alfanuméricos e, caso a resultante seja vazia ou possuir mais que 16 caracteres, um erro é gerado nesse campo;
- **Órgão Emissor:** são extraídos apenas caracteres alfanuméricos e, caso a resultante seja vazia ou possuir mais que 30 caracteres, o órgão "SSP" é inferido, logo, erros não são gerados nesse campo;
- **CPF:** o número de CPF pode ser verificado por meio de um algoritmo que utiliza os próprios dígitos de verificação do CPF para validá-lo. No *IsenSys*, esta implementação consta na biblioteca *Phills Libs* [12]. Caso o algoritmo não consiga validar o CPF, um erro é gerado nesse campo.

Para registrar inconsistências em cada um dos campos de dados de um *Candidato*, é utilizada a classe de exceção *FieldParseException* que, por sua

vez, é incorporada à classe de exceção *RowParseException*, montada com todas as exceções percebidas pelo método *build()*.

3.3.3 A classe de exceção *FieldParseException*

Como forma de descentralizar as tratativas de validação de dados de candidatos, a classe de exceção *FieldParseException* é responsável por armazenar informações sobre o motivo de um campo não ter sido validado e qual campo gerou esta exceção. Esta classe estende a superclasse *Exception* e monta uma *String* formatada com o motivo e o nome do campo inválido.

3.3.4 A classe de exceção *RowParseException*

Com o objetivo de concentrar todas as exceções de validação dos campos de dados de um *Candidato*, a classe de exceção *RowParseException* armazena tais exceções em uma lista encadeada e ainda adiciona informações que ajudam a identificar qual foi a solicitação que gerou a(s) exceção(ões) e ainda em qual posição do arquivo de entrada ele está.

A seguir é possível visualizar o diagrama de classe de *RowParseException*.

Figura 16 – Diagrama de Classe de *RowParseException*

| RowParseException |
|--|
| +linha: int +nis: String +cpf: String +nome: String +listaExcecoes: List<FieldParseException> |
| +addException() +hasException() +getMessage() +getErrorSummaryArray() +getErrorSummaryString() |

3.3.5 A classe *ParseResult*

Esta classe é responsável por concentrar o resultado da extração de dados do(s) arquivo(s) de entrada em duas listas:

1. **Lista de Candidato:** onde são armazenados apenas dados de candidatos solicitantes de isenção que passaram com sucesso por todas as validações de campos definidas na classe *CandidatoBuilder* [3.3.2];
2. **Lista de RowParseException:** onde são armazenados os dados de solicitações julgados inválidos pela classe *CandidatoBuilder* [3.3.2].

A seguir podemos compreender melhor a classe por meio de seu diagrama.

Figura 17 – Diagrama de Classe de *ParseResult*

| ParseResult |
|---|
| +listaCandidatos: List<Candidato> |
| +listaExcecoes: List<RowParseException> |
| +addCandidato() |
| +addExcecao() |
| +getListaCandidatos() |
| +getListaExcecoes() |
| +sortLists() |

3.3.6 Importadores de Dados de Candidato

O *IsenSys* é capaz de importar dados pessoais de candidatos solicitantes de isenção em dois formatos:

1. **Arquivo (.csv):** que é um arquivo de texto puro (sem formatação), contendo um cabeçalho na sua primeira linha e os dados pessoais requeridos pelo sistema nas outras linhas. O arquivo deve estar codificado em UTF-8, com separador por tabulação, vírgula ou ponto-e-vírgula;
2. **Planilha do Microsoft Excel (.xlsx):** a planilha deve conter um cabeçalho na primeira linha com os nomes dos campos e nas demais linhas os dados dos candidatos solicitantes de isenção.

Nos dois casos a ordem da disposição dos dados é extremamente importante para a correta importação. Tanto as colunas do arquivo *csv* quanto as da planilha do *Microsoft Excel* devem respeitar a seguinte ordem:

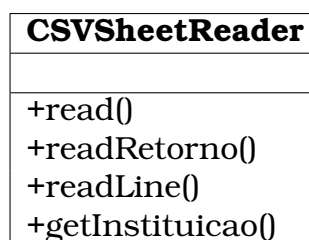
1. Nome completo;
2. NIS;
3. Data de nascimento;
4. Sexo;
5. Número de RG;
6. Data de emissão do RG;
7. Órgão emissor do RG;
8. CPF;
9. Nome completo da mãe.

Para cada tipo de arquivo foi implementado um importador, contendo as especificidades de tratamento de cada formato. Os dados extraídos pelos importadores são então enviados ao *CandidatoBuilder* [3.3.2] que irá construir um objeto *Candidato* [3.3.1], tornando todo o processo de tratativa de arquivos transparente às classes superiores.

3.3.6.1 O importador *CSVSheetReader*

Para o arquivo no formato *.csv*, temos o importador descrito na classe *CSVSheetReader*, que é útil tanto para o módulo de envio, através do método *read()*, quanto pelo módulo de retorno, através do método *readRetorno()*. A princípio o papel deste importador é detectar o tipo de separador do arquivo *csv* e extrair os dados linha-a-linha. A seguir temos seu diagrama de classe.

Figura 18 – Diagrama de Classe de *CSVSheetReader*



3.3.6.2 O importador *ExcelSheetReader*

Este importador também é comum aos três módulos do sistema, mas em momentos distintos. Sua função no módulo de envio é iterar sobre as linhas e colunas de uma planilha com os dados de candidatos solicitantes de isenção e entregá-los ao *CandidatoBuilder* [3.3.2]. Seu diagrama de classe está disposto na figura a seguir.

Figura 19 – Diagrama de Classe de *ExcelSheetReader*

| ExcelSheetReader |
|---|
| |
| +read() +readErros() +readLine() +getCellContent() |

3.3.7 Exportadores de Dados

O estado final do processo de importação de dados dos candidatos solicitantes consiste na concentração deles na classe *ParseResult* [3.3.5]. A partir daqui, os dados dos candidatos considerados válidos pela classe *CandidatoBuilder* [3.3.2] estão prontos para serem exportados para o arquivo no formato de envio do SISTAC [2]. Os dados de solicitações inválidas são exportados para uma planilha do *Microsoft Excel*, para futuro processamento nos módulos de retorno.

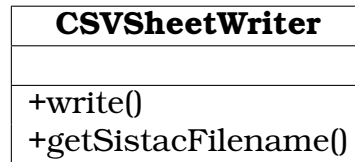
3.3.7.1 O exportador *CSVSheetWriter*

Esta classe tem como função exportar os dados de candidatos válidos para o(s) arquivo(s) no formato de envio do SISTAC [2]. Na primeira linha do arquivo é impresso o cabeçalho com alguns dados do órgão gestor (configurados previamente) e nas demais, os dados são dispostos de acordo com a formatação exigida no manual.

Empiricamente foi descoberto que o SISTAC tem um limite máximo de solicitações por arquivo de envio definido em 2.000 unidades. Portanto,

o exportador automaticamente gera outros arquivos na sequência caso a quantidade de solicitações ultrapasse este limite.

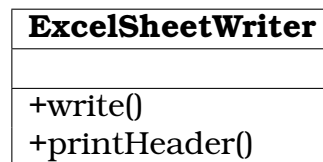
Figura 20 – Diagrama de Classe de *CSVSheetWriter*



3.3.7.2 O exportador *ExcelSheetWriter*

Esta classe tem a função de exportar os dados de solicitações inválidas para uma planilha do *Microsoft Excel*, onde a primeira linha contém um cabeçalho com títulos das colunas de erros e nas demais linhas, os dados de identificação do candidato e a lista de campos que foram considerados inválidos pela classe *CandidatoBuilder* [3.3.2].

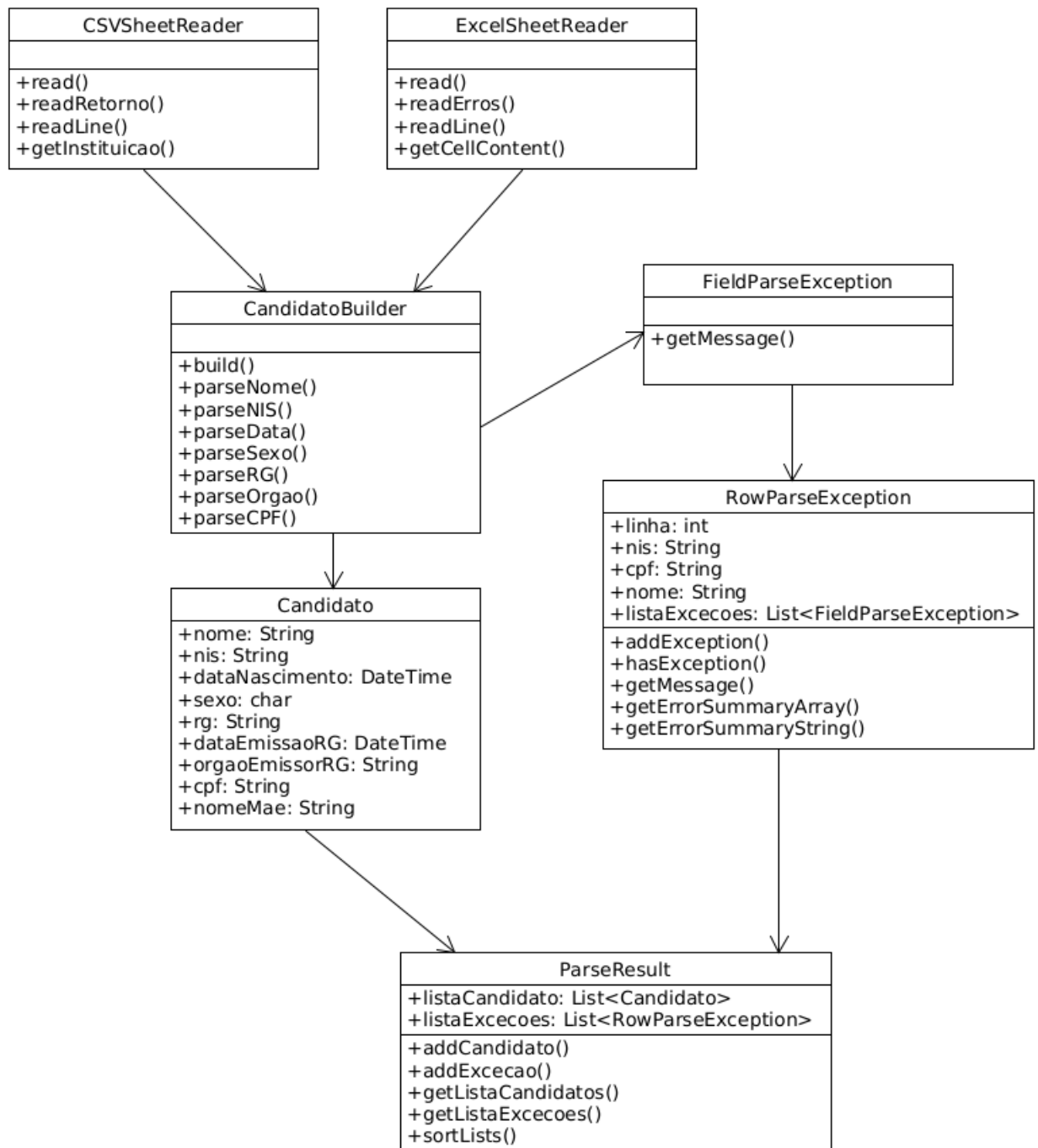
Figura 21 – Diagrama de Classe de *ExcelSheetWriter*



3.3.8 Integração do Módulo de Envio

Após conhecer individualmente todos os agentes envolvidos no módulo de envio, far-se-á uma análise em conjunto de todos eles, possibilitando compreender o fluxo das informações e os estágios do processo de preparação dos dados das solicitações para envio ao SISTAC. A seguir temos uma visão geral do fluxo dos dados de uma solicitação de isenção, desde o arquivo de origem até o estágio onde todos estão devidamente processados.

Figura 22 – Diagrama do Fluxo de Dados das Solicitações



Em resumo, os dados dos candidatos solicitantes são extraídos de acordo com o tipo de arquivo de origem, pelas classes *CSVSheetReader* e *ExcelSheetReader* que, por sua vez, chamam a classe *CandidatoBuilder* para validar os dados extraídos e, caso sejam 100% válidos, um novo objeto *Candidato* é gerado, do contrário, uma exceção do tipo *RowParseException* é lançada com informações de todos os campos inválidos (por meio da classe *FieldParseExcep-*

tion). Por fim, os dados são concentrados no objeto *ParseResult* e estão prontos para exportação.

3.3.9 Interface Gráfica do Módulo de Envio

Como premissas no desenvolvimento do *frontend* do *IsenSys*, simplicidade e usabilidade sobressaem-se, mas sem perder a essência de uma aplicação robusta e consolidada. Na figura a seguir, está ilustrada a única tela do módulo de envio.

Figura 23 – Tela do Módulo de Envio

IsenSys v.3.8 - Exportação

Dados da Instituição

CNPJ: 01.234.567/0000-04

Nome Fantasia: NOME FANTASIA DO ORGAO GESTOR

Razão Social: RAZAO SOCIAL DO ORGAO GESTOR

Arquivo de Entrada

Nome:   

Arquivos de Saída

Num. do Edital: Sequência: 1  

Pasta de Saída:  



No painel 'Dados da Instituição', estão dispostas algumas informações exigidas no cabeçalho do arquivo de envio ao SISTAC, tais como CNPJ, nome fantasia e razão social do órgão gestor.

No painel 'Arquivo de Entrada' é possível selecionar o arquivo de origem dos dados de solicitações dos candidatos, atualmente planilhas do *Microsoft Excel* e arquivos *csv* são suportados. Também é possível recarregar o arquivo previamente selecionado ou limpar sua seleção. Se um arquivo válido foi selecionado, algumas informações sobre o carregamento são mostradas na porção inferior deste painel, como ilustra a figura a seguir.

Figura 24 – Tela do Módulo de Envio (arquivo carregado)

A interface 'IsenSys v.3.8 - Exportação' apresenta os seguintes elementos:

- Dados da Instituição:**
 - CNPJ: 01.234.567/0000-04
 - Nome Fantasia: NOME FANTASIA DO ORGAO GESTOR
 - Razão Social: RAZAO SOCIAL DO ORGAO GESTOR
- Arquivo de Entrada:**
 - Nome: solicitacoes.csv (com ícones de refresh, limpar e lupa)
 - Solicitações: 7651 (OK) 1 (ERRO) 7652 (TOTAL)
- Arquivos de Saída:**
 - Num. do Edital: [campo vazio]
 - Sequência: 1 (com setas de incremento/decremento)
 - Pasta de Saída: [campo vazio] (com ícones de lupa e limpar)
- Ícone de salvar (disquete) no canto inferior direito.

Os dois primeiros campos do painel 'Arquivos de Saída' (Num. do Edital e Sequência) também são exigências descritas no manual de envio do SISTAC [2]. O campo 'Num. do Edital' faz parte do nome do arquivo de envio, juntamente com a identificação de sua sequência. Cada arquivo possui uma sequência geralmente iniciada em '1'. Se dois arquivos são gerados, temos então as sequências '1' e '2'. Sequências são reiniciadas a cada novo dia. Ainda neste painel, podemos escolher o diretório de escrita dos arquivos ou limpar sua seleção.

Por fim, após todos os dados serem fornecidos na tela, a exportação pode ser realizada clicando no ícone de salvar (disquete). As figuras a seguir ilustram a saída dos arquivos após alimentar a tela com algumas informações.

Figura 25 – Tela do Módulo de Envio (preenchida) + Arquivos de Saída



De posse dos arquivos exportados, o envio já pode ser realizado à plataforma do SISTAC. A planilha gerada pode ser consultada de forma avulsa, mas é recomendável mantê-la inalterada, pois é necessária para a montagem dos relatórios de publicação nos módulos de retorno.

3.4 Desenvolvimento dos Módulos de Retorno

Esta seção tem por objetivo detalhar o desenvolvimento do *backend* de processamento do(s) arquivo(s) de retorno do SISTAC. A partir dele(s) é possível gerar relatórios com resultados e algumas métricas e estatísticas úteis ao órgão gestor, todos prontos para publicação.

Utilizando a mesma metodologia abordada na seção anterior, vamos começar compreendendo o fluxo de informações, conhecendo as classes envolvidas no processo e, posteriormente, a interface gráfica.

3.4.1 Modelagem de um Retorno

Para estes módulos, apenas alguns dados das solicitações dos candidatos são aproveitados. Temos então a concepção da classe *Retorno*, de acordo com a especificação descrita na figura a seguir:

Figura 26 – Modelagem da Entidade *Retorno*

| Retorno |
|---|
| +situacao: char +nome: String +nis: String +cpf: String +motivo: int +nomeAnterior: String |
| +deferre() +isDeferido() +compareTo() |

3.4.2 A classe *ListaRetornos*

Esta classe concentra todos os *Retorno*'s extraídos do arquivo de retorno do SISTAC, com adição dos atributos do órgão gestor e dados do edital de publicação. Será detalhado ao decorrer desta monografia os dois módulos de retorno, que são: retorno preliminar [3.4.7.1] e retorno definitivo [3.4.7.3]. Por enquanto, temos que esta classe armazena dados úteis aos dois módulos.

Figura 27 – Diagrama de classe de *ListaRetornos*

| ListaRetornos |
|---|
| +listaRetornos: List<Retorno> +cnpj: String +nomeFantasia: String +razaoSocial: String +edital: String +dataEdital: String +cabecalho: String |
| +get() +size() +getList() +clone() +add() +update() +sort() |

3.4.3 A classe *Compilacao*

Esta classe possui a incumbência de gravar ou carregar um arquivo com a extensão (.icf - *IsenSys Compilation File*) objetos da classe *ListaRetornos*, útil

no processamento do retorno definitivo. Eis o seu diagrama de classe:

Figura 28 – Diagrama de Classe de *Compilacao*

| Compilacao |
|--------------------|
| |
| +save() +load() |

3.4.4 Modelagem de uma Situacao

De acordo com o manual do SISTAC [2], existem algumas situações de indeferimento definidas. No *IsenSys*, tais situações são armazenadas em um arquivo (*csv*) com codificação UTF-8 no diretório de recursos do sistema [3.6]. Durante a confecção dos editais de publicação, as situações são carregadas do arquivo e enviadas à classe geradora de relatórios, juntamente com os dados dos retornos.

Figura 29 – Modelagem da Entidade *Situacao*

| Situacao |
|--|
| +id: String +motivo: String +descricao: String |
| |

3.4.5 A classe *SituacaoDAO*

Basicamente esta classe é responsável por carregar as situações a partir do arquivo '*situacoes.csv*', contido no diretório de recursos do *IsenSys*, para uma lista de *Situacao* [3.4.4], útil na confecção dos relatórios de publicação.

Figura 30 – Diagrama de Classe de *SituacaoDAO*

| SituacaoDAO |
|--------------------|
| |
| +load() |

3.4.6 Geradores de Relatório

O *IsenSys* implementa três tipos de relatórios a partir dos dados provenientes do(s) arquivo(s) de retorno do SISTAC. Cada uma das classes que os implementam contam com o método *'export()'*, que é capaz de compilar e exportar o relatório para o formato PDF. A figura a seguir ilustra as estruturas de classe dos três geradores.

Figura 31 – Diagramas de Classe dos Geradores de Relatório

| PDFEstatisticas | PDFResultado | PDFSimilaridade |
|------------------------|---------------------|------------------------|
| | | |
| +export() | +export() | +export() |

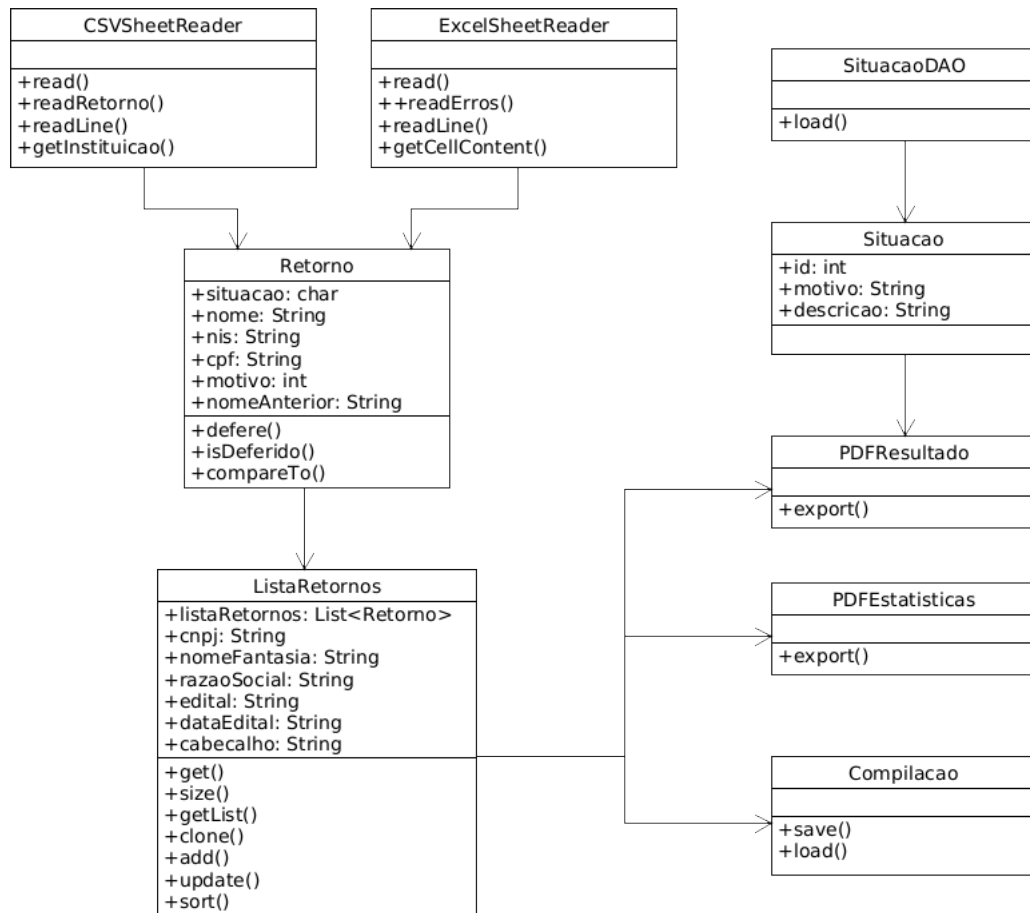
3.4.7 Integração dos Módulos de Retorno

Conhecendo individualmente os agentes envolvidos no módulo de retorno, podemos realizar uma abordagem integrada dos dois tipos de retorno: retorno preliminar e retorno definitivo.

3.4.7.1 Retorno Preliminar

Este é o módulo destinado ao processamento do resultado preliminar das solicitações de isenção. A figura a seguir ilustra o fluxo das informações através do módulo.

Figura 32 – Diagrama do Fluxo de Dados no Retorno Preliminar



Começando pelos arquivos de entrada, as classes *CSVSheetReader* e *ExcelSheetReader* carregam o(s) arquivo(s) de retorno do SISTAC e planilha de erros, respectivamente, produzindo então objetos da classe *Retorno* que são agrupados em uma lista na classe *ListaRetornos*.

Na exportação, os dados armazenados na classe *ListaRetornos* são enviados aos geradores de PDF implementados em *PDFEstatisticas* e *PDFResultado* e, por fim, salvos em um arquivo binário denominado de 'compilação', por meio da classe *Compilacao*.

3.4.7.2 Tela do Módulo de Retorno Preliminar

A figura a seguir ilustra a implementação da interface gráfica do módulo de retorno preliminar do *IsenSys*.

Figura 33 – Tela do Módulo de Retorno Preliminar

IsenSys v.3.8 - Resultado Preliminar



Dados da Instituição



CNPJ: 01.234.567/0000-04

Nome Fantasia: NOME FANTASIA DO ORGAO GESTOR


Razão Social: RAZAO SOCIAL DO ORGAO GESTOR

Arquivos de Entrada



Retorno Sistac:  


Planilha Erros:  

Edital

Cabeçalho: 

Arquivos de Saída

Diretório:  



No painel 'Dados da Instituição', estão dispostas algumas informações exigidas no cabeçalho do arquivo de envio ao SISTAC [2], tais como CNPJ, nome fantasia e razão social do órgão gestor.

Em 'Arquivos de Entrada' é possível selecionar o(s) arquivo(s) de retorno do SISTAC e a planilha contendo os erros de validação (gerada no módulo de envio). O usuário também é capaz de limpar a seleção do(s) arquivo(s). A seguir estão enumerados alguns detalhes sobre os arquivos:

- **Retorno Sistac:** o usuário deve selecionar SEMPRE o primeiro arquivo de retorno do SISTAC, caso haja mais de um, o carregamento dos demais arquivos é feito de forma automática, basta que estejam no mesmo diretório do primeiro arquivo;
- **Planilha Erros:** aqui o usuário deve selecionar a planilha de erros gerada pelo módulo de envio do *IsenSys*;

Se um arquivo válido foi selecionado, algumas informações sobre o carregamento são mostradas na porção inferior deste painel, como ilustra a figura a seguir.

Figura 34 – Tela do Módulo de Retorno Preliminar (preenchida)

IsenSys v.3.8 - Resultado Preliminar

Dados da Instituição

CNPJ: 01.234.567/0000-04

Nome Fantasia: NOME FANTASIA DO ORGAO GESTOR

Razão Social: RAZAO SOCIAL DO ORGAO GESTOR

Arquivos de Entrada

Retorno Sistac: RETORNO_01234567000004_012023_01082023_005.txt

Planilha Erros: ERROS_01234567000004_012023_01082023.xlsx

Análise do Arquivo

Deferidos: 2836 Indeferidos: 4816 Total: 7652

Edital

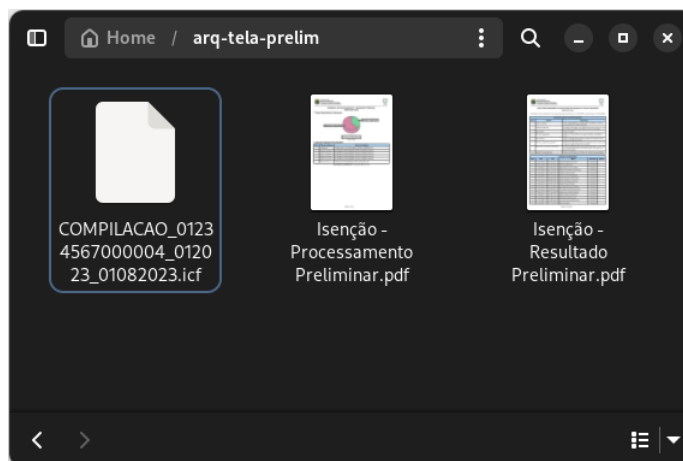
Cabeçalho: Edital de Teste

Arquivos de Saída

Diretório: /home/felipe/arq-tela-prelim

O nome do edital pode ser informado no campo de texto do painel 'Edital' e no painel 'Arquivos de Saída' é possível selecionar o diretório para saída dos arquivos gerados pelo módulo (resultados, relatório de estatísticas e arquivo de compilação), como mostra a figura a seguir.

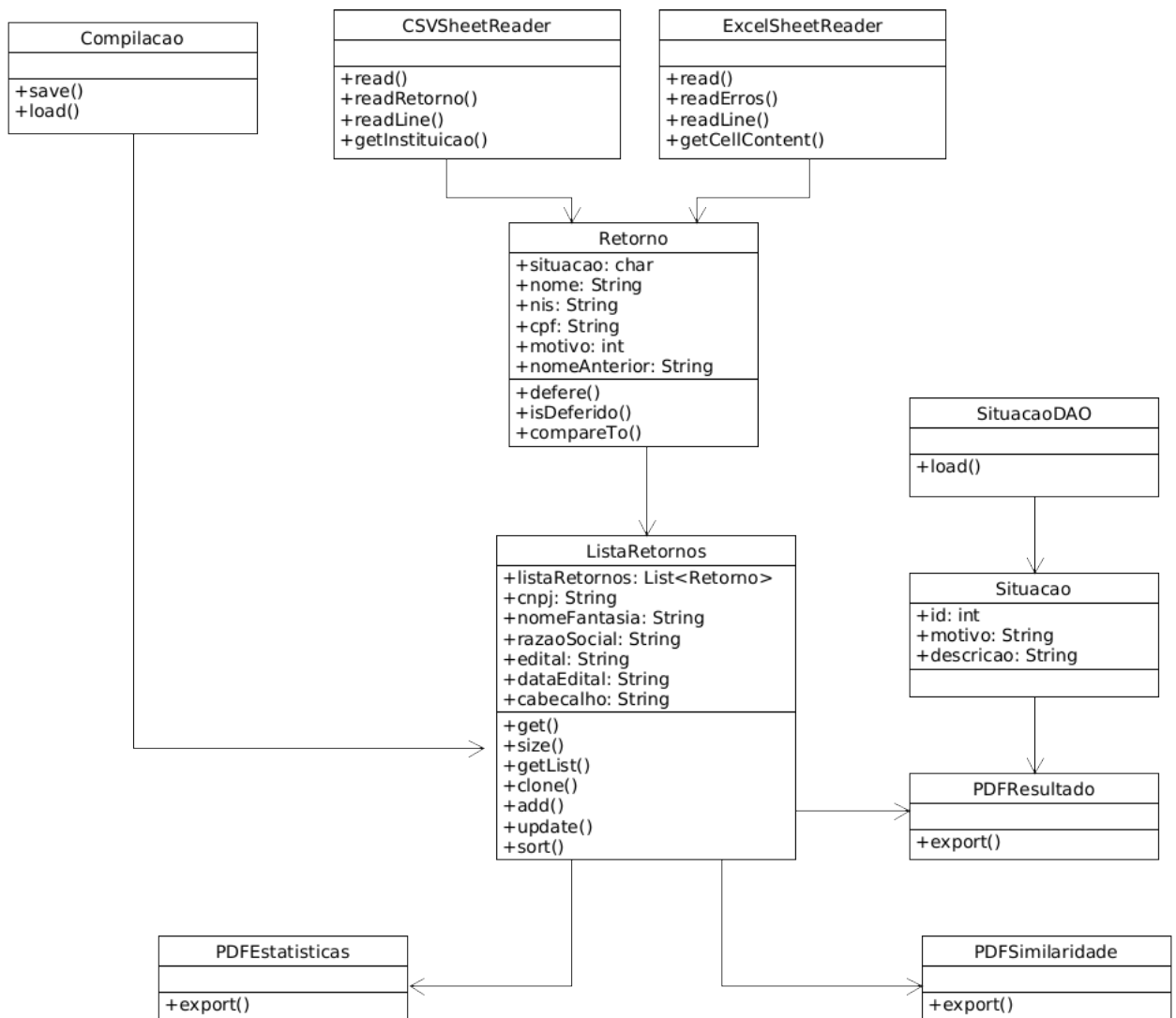
Figura 35 – Arquivos de Saída do Módulo de Retorno Preliminar



3.4.7.3 Retorno Definitivo

Este é o módulo destinado ao processamento do resultado definitivo. É capaz de recuperar as informações do módulo de retorno preliminar, através de um arquivo de compilação (gerado pela classe *Compilacao* [3.4.3]) e importar o(s) novo(s) arquivo(s) de retorno do SISTAC.

Figura 36 – Diagrama do Fluxo de Dados no Retorno Definitivo



Para o processamento do retorno definitivo, a principal classe alimentadora é a *Compilacao* [3.4.3] que carrega os dados provenientes do módulo de resultado preliminar e, a princípio, já estão prontos para serem exportados como resultado definitivo.

3.4.7.4 Tela do Módulo de Retorno Definitivo

A figura a seguir ilustra a implementação da interface gráfica do módulo de retorno definitivo do *IsenSys*.

Figura 37 – Tela do Módulo de Retorno Definitivo

IsenSys v.3.8 - Resultado Definitivo

Dados da Instituição

CNPJ: 01.234.567/0000-04

Nome Fantasia: NOME FANTASIA DO ORGAO GESTOR

Razão Social: RAZAO SOCIAL DO ORGAO GESTOR

Resultado Preliminar

Compilação:

Arquivos de Entrada

Retorno Sistac:

Planilha Erros:

Edital

Cabeçalho:

Arquivos de Saída

Diretório:

Ao carregar o arquivo de compilação, algumas informações são exibidas na porção inferior do painel 'Resultado Preliminar'. O cabeçalho do edital também é carregado.

Da mesma forma que no módulo do resultado preliminar, o arquivo de retorno a ser selecionado deve ser sempre o primeiro, os demais são carregados automaticamente desde que estejam no mesmo diretório do primeiro arquivo.

Se algum arquivo de entrada for selecionado, tanto de retorno quanto de erros, os dados são mesclados com os provenientes do resultado preliminar da seguinte forma:

- **Retorno Sistac:** se o candidato retornado tiver sua solicitação **indeferida**, apenas seu status é atualizado na lista preexistente. Caso tenha sido **deferida**, além de ter seu estado alterado, o novo deferido fará parte de uma nova lista, para futuro cálculo de similaridade [3.5];
- **Erros:** aqui todos os erros são mesclados.

3.5 Relatório de Similaridade

Uma necessidade especial tornou necessária a implementação de um relatório que calculasse o quão similar é o nome de um candidato na fase de solicitação (1ª instância) e na fase de recurso (2ª instância).

Empiricamente foi detectado que um candidato qualquer que teve sua solicitação de isenção indeferida em primeira instância e recursou trocando seus dados pessoais pelos de alguém que tem ciência de isenção garantida, teve recurso deferido pelo SISTAC.

Como forma de evitar que tais candidatos praticantes deste ato ilícito tivessem isenção deferida sem ciência do órgão gestor, foi desenvolvido um relatório chamado de 'Relatório de Similaridade e Distância entre Nomes', que computa porcentagens das referidas métricas utilizando o algoritmo de *Jaro-Winkler* [2.7], entre o nome informado na fase de solicitação e recurso, pelo candidato.

O algoritmo é executado tomando como base candidatos que tiveram solicitação de isenção indeferida em primeira instância, mas deferida após recurso, gerando um relatório com o comparativo entre nomes e as duas métricas (similaridade e distância), ordenado pela ordem crescente de similaridade.

Algumas classes estão envolvidas na confecção deste relatório, são elas: *Similaridade* e *JaroWinkler*, além do gerador de PDF: *PDFSimilaridade*.

3.5.1 Modelagem de uma Similaridade

Esta classe contém alguns atributos úteis para confecção do relatório de similaridade entre nomes. A figura a seguir ilustra seu diagrama de classe.

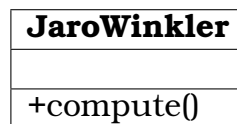
Figura 38 – Diagrama de Classe de uma Similaridade

| Similaridade |
|--------------------------|
| +nomeSolicitacao: String |
| +nomeRecurso: String |
| +distancia: double |
| +similaridade: double |
| |

3.5.2 A classe *JaroWinkler*

É uma classe auxiliar que computa o algoritmo de *Jaro-Winkler* [2.7] nos dados dos candidatos recursantes. Para cálculo das métricas de distância e similaridade, a biblioteca *Apache Commons Text*TM [2.2] foi utilizada. A seguir seu diagrama de classe.

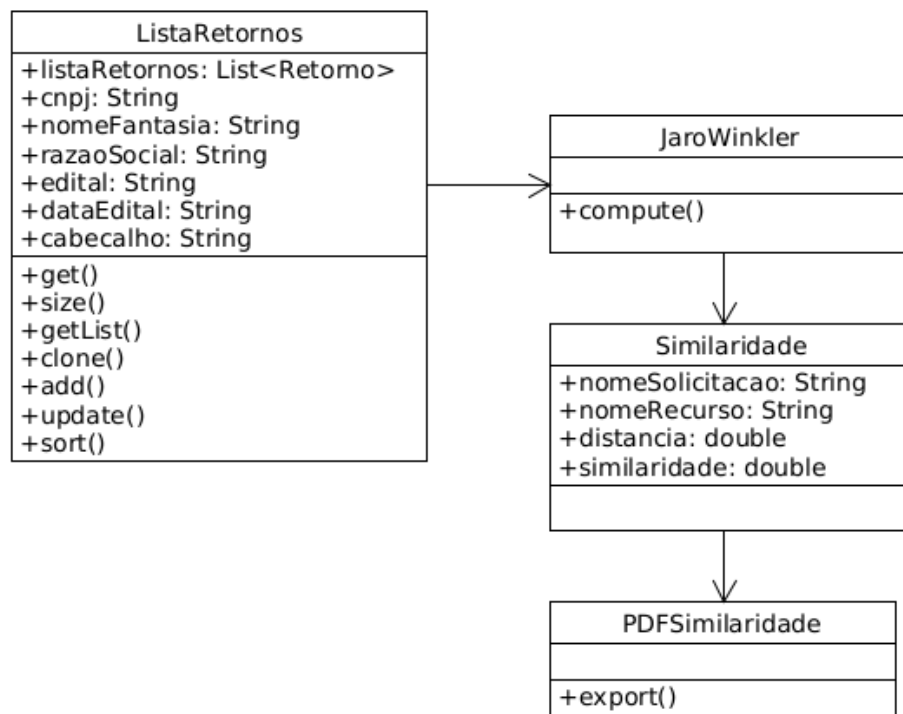
Figura 39 – Diagrama de Classe de *JaroWinkler*



3.5.3 Montagem do Relatório de Similaridade

Realizando uma abordagem integrada, conhecendo os agentes envolvidos na produção do relatório, temos o seguinte diagrama.

Figura 40 – Diagrama do Fluxo de Dados no Relatório



De uma forma prática e funcional, os dados dos candidatos solicitantes, provenientes de *ListaRetornos*, são encaminhados à classe *JaroWinkler* para computação das métricas de distância e similaridade e salvamento em objetos

da classe *Similaridade*. Por fim, *PDFSimilaridade* aproveita estes dados para a confecção do relatório.

3.6 Diretório de Recursos do *IsenSys*

O *IsenSys* conta com alguns arquivos de configuração e recursos que estão dispostos em um diretório chamado '*res*', existente no mesmo nível de diretório do arquivo executável do *IsenSys*.

A seguir estão listados os recursos e seus caminhos relativos ao diretório raiz da aplicação.

- ***config/program.dat***: contém as configurações globais do sistema;
- ***fonts/fonts-extension.jar***: fontes utilizadas na confecção dos relatórios;
- ***i18n/*.lng***: arquivos contendo as *strings* de interface do sistema;
- ***icon/*.png***: contém os ícones utilizados nas telas;
- ***img/***: diretório contendo as imagens de fundo das telas;
- ***relatorios/*.jasper***: arquivos-modelos dos relatórios;
- ***situacoes/situacoes.csv***: arquivo contendo as situações de indeferimento.

Para o correto funcionamento do sistema é recomendável que o usuário não altere o conteúdo destes arquivos manualmente, tampouco os exclua, renomeie ou os troque de diretório.

4 Implementação e Resultados

A versão corrente do sistema (v.3.8) foi implementada no início de Junho de 2023 na COMPEC, com o objetivo de ter como piloto o processamento das solicitações de isenção do Processo Seletivo Contínuo - PSC 2024 - Etapas 1 e 2, fruto dos Editais de nº [13] e [14]/GR/UFAM.

Por pura coincidência, o PSC 2024 é a primeira edição a possuir a modalidade de isenção via CadÚnico logo nas primeiras etapas do processo, opção que só era concedida aos candidatos da 3ª etapa.

Segundo os editais que regem o processo seletivo, o período de solicitações de isenção de taxa de inscrição foi entre as 10h do dia 01/06/2023 até as 17h do dia seguinte, onde foram recebidas **7.652 solicitações na 1ª etapa e 3.052 na 2ª etapa**, totalizando incríveis **10.704 solicitações**.

A computação das solicitações foi realizada em um computador da marca Lenovo, modelo M93p com as seguintes especificações:

- Memória RAM DDR3-1600, com capacidade total de 16GB;
- Processador Intel® Core™ i5-4570 de 3.60 GHz;
- Sistema operacional Arch Linux de 64 bits, com *engine* gráfica GNOME;
- Monitor padrão da marca HP, modelo V225hz, com resolução de 1920x1080.

Seguindo o fluxo padrão do processo de isenção, começamos pela preparação de dados para envio ao SISTAC, através da Tela de Envio. Toda a interação com a interface gráfica se mostrou estável e não foi detectado nenhum erro de execução ou travamentos, mesmo com alto volume de dados.

Tabela 2 – Tempos de Processamento de Dados [Preliminar]

| Etapa | Qtd. de Solicitações | Tempo de Execução (ms) | Tempo de Exportação de Relatórios (s) |
|--------------|-----------------------------|-------------------------------|--|
| 1 | 7.652 | 1.500 | 2.7 |
| 2 | 3.052 | 780 | 1.25 |

Tabela 3 – Tempos de Processamento de Dados [Definitivo]

| Etapa | Qtd. de Recursos | Tempo de Execução (ms) | Tempo de Exportação de Relatórios (s) |
|--------------|-------------------------|-------------------------------|--|
| 1 | 357 | 250 | 3.1 |
| 2 | 157 | 138 | 2.07 |

Como ilustrado em números, o *IsenSys* conseguiu reduzir drasticamente o tempo de processamento das solicitações, processo realizado por humanos antes da implementação do sistema. Antes havia uma equipe que, de posse dos dados de solicitação, realizavam a confecção dos arquivos de envio ao SISTAC de forma artesanal, processo lento que se estendia por dias e era extremamente suscetível a erros.

Além da confecção manual dos arquivos *csv*, perdia-se muito tempo no processo de envio de lotes ao SISTAC pois, ao detectar o primeiro erro de preenchimento de arquivo, cancela todo o envio do lote e sinaliza apenas o erro corrente. Logo, corrigia-se manualmente o erro e uma nova submissão era realizada, até que o lote fosse totalmente recebido, sem erros.

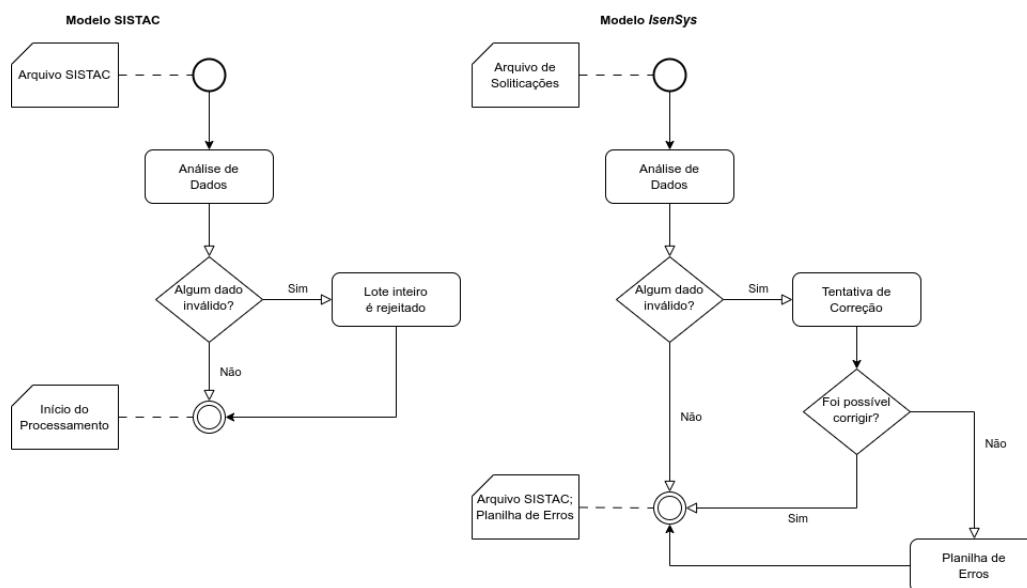
Em concursos com maior volume de solicitações estima-se que pelo menos 25 a 30 tentativas de envio eram realizadas à plataforma. É muito provável que esta realidade estenda-se em mais órgãos do Poder Executivo utilizadores do SISTAC e é quando o *IsenSys* pode servir como aplicação institucionalizada, visando eficiência e colaboração no serviço público.

A otimização implementada no *IsenSys* faz com que os dados das solicitações passem por filtros analistas de dados, implementados de acordo com as normativas do manual do SISTAC [2]. Ao detectar alguma disparidade entre o dado e o formato aceito, o *IsenSys* tenta ajustá-lo e caso não seja possível, o separa em uma lista de erros e o processamento continua.

Ao fim do processo, um arquivo de envio 100% válido no formato do SISTAC é gerado e os erros, separados em uma planilha, para futura confecção dos relatórios de resultados preliminar e definitivo.

O diagrama a seguir ilustra o pré-processamento de dados realizado tanto pelo SISTAC quanto pelo *IsenSys*. Podemos verificar que, ao primeiro erro encontrado no arquivo de solicitações, o SISTAC rejeita o lote por completo.

Figura 41 – Diagrama Comparativo entre o SISTAC e IsenSys



Graças à implementação do *IsenSys*, a COMPEC conseguiu reduzir os tempos de processamento de solicitações de isenção de dias para segundos. Atualmente, pouco tempo após o término do prazo de solicitação ou recurso, os resultados preliminares ou definitivos já podem ser confeccionados.

A tabela a seguir traz um comparativo entre o tempo necessário para confeccionar um arquivo de envio válido ao SISTAC, utilizando o processo manual e o *IsenSys*. Vale ressaltar que nos dois processos, apenas o *IsenSys* foi utilizado, as quantidades de dias foram calculadas com base em concursos anteriores de porte semelhante realizados pela COMPEC, utilizando as mesmas metodologias.

Tabela 4 – Comparativo de Tempo de Processamento do Arquivo de Envio

| Processo | Qtd. Solicitações | Preparo Manual (dias) | Preparo no IsenSys (s) |
|-----------------|--------------------------|------------------------------|-------------------------------|
| PSC 2023 | 11.477 | 15 | 5 |
| PSTEC 2023 | 5.020 | 13 | 4 |

Vimos que, com a implementação do *IsenSys*, foi possível reduzir custos com contratação temporária de pessoal e recursos como impressão em papel, tonner de impressora e energia elétrica, tornando o serviço eficiente e com resultado preciso com alguns cliques e em alguns segundos.

5 Conclusões

5.1 Considerações Finais

Visando valores como agilidade, lisura e transparência no processamento das solicitações de isenção de taxa de inscrição na modalidade CadÚnico, o *IsenSys* surgiu como agente concretizador destes anseios, possibilitando que um órgão gestor consiga preparar dados para envio ao SISTAC e, após seu processo de análise, gerar relatórios de publicação de resultados e relatórios com estatísticas e métricas.

Os resultados obtidos foram de grande satisfação tanto pela COMPEC, quanto pela sociedade, que tiveram um retorno ágil e preciso, respeitando todos os requisitos dispostos nos editais do concurso piloto e instruções normativas sobre o processo de isenção, de acordo com os manuais do SISTAC.

Pode-se afirmar então que o objetivo principal do *IsenSys* foi atingido. A aplicação foi construída utilizando uma linguagem de programação amplamente aceita e suportada no mercado, respeitando as boas práticas de programação e convenções internacionais, mantendo sempre a organização e comentários no código-fonte de forma a facilitar futuras colaborações.

5.2 Limitações e Propostas de Atualizações

Não há nada que exista que não possa ser melhorado e, partindo desse pressuposto, algumas otimizações e adições de novos recursos já foram mapeadas durante o processo de desenvolvimento e que, devido ao curto tempo para implementação e conclusão desta monografia, ainda não puderam ser incorporados ao projeto. Listar-se-á algumas:

- Estender o suporte do *IsenSys* à máquinas de 32-bits, com a utilização do JDK 8;
- Implementar layouts dinâmicos na interface gráfica, de forma a permitir o redimensionamento das janelas;

- Permitir que o número de edital contenha caracteres alfanuméricos de forma a abranger esta atualização recente do SISTAC;
- Adicionar interfaces gráficas para gerência dos recursos do sistema tais como logomarca e arquivo de situações;
- Utilizar banco de dados para salvar configurações;
- Integrar as outras modalidades de isenção de taxa de inscrição, como a modalidade de análise documental, definida pela Lei nº 12.799, de 10 de abril de 2013 [15], e doadores de medula óssea, regulamentado pela Lei nº 13.656, de 30 de abril de 2018 [16].

O código-fonte em sua íntegra está disponível no repositório GitHub sob os links:

- **IsenSys:** <<https://github.com/icompe-felipe/java-compec-isensys>>
- **Phills Libs (dependência):** <<https://github.com/icompe-felipe/phills-libs>>
- **Relatórios:** <<https://github.com/icompe-felipe/jasper-compec-isensys>>

Referências

- 1 REPÚBLICA, P. da. *Decreto nº 6.593, de 2 de outubro de 2008*. 2023. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/decreto/d6593.htm>. Acesso em: 18 de Setembro de 2023. Citado 3 vezes nas páginas 16, 17 e 25.
- 2 FOME, M. do Desenvolvimento Social e C. *Padrão para Envio das informações de candidatos para isenção de pagamento da taxa de inscrição em Concursos públicos realizados no âmbito do Poder Executivo Federal*. 2023. Disponível em: <http://aplicacoes.mds.gov.br/sistac/publico/arquivos/Manual_Envio_Recebimento.pdf>. Acesso em: 15 de Setembro de 2023. Citado 11 vezes nas páginas 17, 25, 27, 34, 35, 36, 41, 45, 48, 51 e 59.
- 3 ORACLE. *O que é tecnologia Java e por que preciso dela?* 2023. Disponível em: <https://www.java.com/pt-BR/download/help/whatis_java.html>. Acesso em: 09 de Setembro de 2023. Citado na página 19.
- 4 FOUNDATION, A. *Commons Text*. 2023. Disponível em: <<https://commons.apache.org/proper/commons-text/>>. Acesso em: 12 de Setembro de 2023. Citado na página 20.
- 5 FOUNDATION, T. A. S. *Welcome to Apache Maven*. 2023. Disponível em: <<https://maven.apache.org/>>. Acesso em: 10 de Setembro de 2023. Citado na página 20.
- 6 FOUNDATION, T. A. S. *Apache POI - the Java API for Microsoft Documents*. 2023. Disponível em: <<https://poi.apache.org/>>. Acesso em: 10 de Setembro de 2023. Citado na página 21.
- 7 FOUNDATION, E. *About the Eclipse Foundation*. 2023. Disponível em: <<https://www.eclipse.org/org/>>. Acesso em: 12 de Setembro de 2023. Citado na página 21.
- 8 JASPERSOFT. *JasperReports Library*. 2023. Disponível em: <<https://community.jaspersoft.com/project/jasperreports-library>>. Acesso em: 10 de Setembro de 2023. Citado na página 22.
- 9 STATOLOGY. *An Introduction to Jaro-Winkler Similarity (Definition Example)*. 2023. Disponível em: <<https://www.statology.org/jaro-winkler-similarity>>. Acesso em: 10 de Outubro de 2023. Citado na página 23.
- 10 JODA-TIME. *Why Joda Time?* 2023. Disponível em: <<https://www.joda.org/joda-time>>. Acesso em: 10 de Setembro de 2023. Citado na página 24.
- 11 CIDADANIA, M. da. *Orientações Gerais do Sistema de Isenção de Taxas de Concursos (SISTAC)*. 2023. Disponível em: <http://aplicacoes.mds.gov.br/sistac/publico/arquivos/Orientacoes_Gerais_2019_novo.pdf;jsessionid=46BA47F8F87D3A66614467AA04817B68>. Acesso em: 15 de Setembro de 2023. Citado na página 27.

- 12 SILVA, F. A. S. da. *Biblioteca Phills Libs*. 2023. Disponível em: <https://github.com/icomf-felipe/phills-libs>. Acesso em: 12 de Setembro de 2023. Citado na página 37.
- 13 COMPEC-UFAM. *Editais nº 13/2023-GR, de 25 de abril de 2023 - Processo Seletivo Contínuo - PSC 2023 - Etapa 1, Projeto 2024*. 2023. Disponível em: <https://edoc.ufam.edu.br/bitstream/123456789/6870/1/Editais%2013%20de%202023.pdf>. Acesso em: 25 de Setembro de 2023. Citado na página 58.
- 14 COMPEC-UFAM. *Editais nº 14/2023-GR, de 25 de abril de 2023 - Processo Seletivo Contínuo - PSC 2023 - Etapa 2, Projeto 2024*. 2023. Disponível em: <https://edoc.ufam.edu.br/bitstream/123456789/6871/1/Editais%2014%20de%202023.pdf>. Acesso em: 25 de Setembro de 2023. Citado na página 58.
- 15 REPÚBLICA, P. da. *Decreto nº 12.799, de 10 de abril de 2013*. 2023. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/lei/112799.htm. Acesso em: 18 de Setembro de 2023. Citado na página 62.
- 16 REPÚBLICA, P. da. *Lei nº 13.656, de 30 de abril de 2018*. 2023. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113656.htm. Acesso em: 21 de Setembro de 2023. Citado na página 62.

ANEXO A – Resultado Definitivo da
Isenção do PSC 2023 (apenas algumas
páginas)



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



RESULTADO DEFINITIVO DA SOLICITAÇÃO DE ISENÇÃO DA TAXA DE INSCRIÇÃO
Processo Seletivo Contínuo - PSC 2023 - Etapa 1 [Retificado em 25/06/2023]

O Cadastro Único (CadÚnico) está regulamentado pelo [Decreto nº 11.016/2022](#) e pelo [Decreto nº 6.593/2008](#).

| Legenda de Erros de Processamento | | |
|-----------------------------------|---------------------------------|---|
| Cód. | Motivo | Descrição |
| 1 | Dados inválidos | Envio impossibilitado ao órgão gestor (CadÚnico) devido a inconsistência nos dados pessoais |
| 2 | Dados divergentes | Os dados informados na solicitação de isenção ou de recurso de isenção não estão associados ao CPF do candidato |
| 3 | NIS inválido | O NIS é inválido |
| 4 | NIS não cadastrado | O NIS não foi identificado na base de dados do Cadastro Único |
| 5 | NIS excluído | O NIS foi identificado na base de dados do Cadastro Único, porém com status excluído |
| 6 | NIS com renda fora do perfil | O NIS foi identificado na base do Cadastro Único, porém com renda per capita familiar fora do perfil |
| 7 | NIS não é o da pessoa informada | O NIS foi identificado na base do Cadastro Único, com renda per capita familiar dentro do perfil, porém não pertence a pessoa informada |
| 13 | Cadastro desatualizado | Pessoa identificada em família com cadastro desatualizado |

| Listagem de Resultados | | | | | |
|------------------------|---------------|----------------|--------------------------------|----------|--------|
| Ord. | NIS | CPF | Nome | Situação | Motivo |
| 1 | ***.225.264.* | ***.644.642.** | Abdiel Rodrigues da Costa | Deferido | |
| 2 | ***.561.166.* | ***.672.472.** | Abel de Souza Lima | Deferido | |
| 3 | ***.694.918.* | ***.249.792.** | Abgail Barroso de Carvalho | Deferido | |
| 4 | ***.354.039.* | ***.237.812.** | Abigail Sarah Moura dos Santos | Deferido | |
| 5 | ***.250.104.* | ***.538.792.** | Abnego Costolio Inacio | Deferido | |
| 6 | ***.919.986.* | ***.575.702.** | Abraao de Lima Magalhaes | Deferido | |
| 7 | ***.242.171.* | ***.001.502.** | Abraao Marlison Araujo Silva | Deferido | |
| 8 | ***.067.440.* | ***.507.352.** | Abraao Rodrigues da Costa | Deferido | |
| 9 | ***.402.314.* | ***.975.142.** | Abraao Silva de Castro | Deferido | |
| 10 | ***.476.707.* | ***.114.502.** | Acenildo Ferreira dos Santos | Deferido | |
| 11 | ***.463.030.* | ***.663.292.** | Ada Izabele Figueredo da Silva | Deferido | |
| 12 | ***.660.710.* | ***.321.992.** | Adamys dos Santos Campos | Deferido | |
| 13 | ***.106.555.* | ***.897.172.** | Adelmo Oliveira de Souza | Deferido | |
| 14 | ***.350.109.* | ***.230.722.** | Adila Sissa da Silva Lima | Deferido | |
| 15 | ***.714.088.* | ***.402.842.** | Adria Cavalcante Nascimento | Deferido | |
| 16 | ***.188.400.* | ***.127.302.** | Adria Faia Pinto | Deferido | |

| Ord. | NIS | CPF | Nome | Situação | Motivo |
|------|----------------|----------------|-----------------------------------|----------|--------|
| 17 | ***.714.392.** | ***.898.852-** | Adria Oliveira Ferreira | Deferido | |
| 18 | ***.408.807.** | ***.170.072-** | Adria Vitoria Damasceno Furtado | Deferido | |
| 19 | ***.402.855.** | ***.747.172-** | Adriah Vitoria Silva Maia | Deferido | |
| 20 | ***.234.783.** | ***.848.872-** | Adrian Colares Portilho | Deferido | |
| 21 | ***.683.603.** | ***.649.362-** | Adrian Daniel Ramos Martins | Deferido | |
| 22 | ***.351.585.** | ***.270.172-** | Adriana de Moura Rodrigues | Deferido | |
| 23 | ***.408.812.** | ***.305.422-** | Adriana Gomes Guerreiro | Deferido | |
| 24 | ***.655.547.** | ***.902.952-** | Adriana Pontes Barros | Deferido | |
| 25 | ***.334.549.** | ***.977.842-** | Adriane de Oliveira Seixas | Deferido | |
| 26 | ***.715.896.** | ***.314.292-** | Adriane Miranda Araujo | Deferido | |
| 27 | ***.747.129.** | ***.177.212-** | Adriane Neves Seda | Deferido | |
| 28 | ***.955.848.** | ***.978.482-** | Adriano Araujo de Souza | Deferido | |
| 29 | ***.389.901.** | ***.443.562-** | Adriano Marcel Silva de Souza | Deferido | |
| 30 | ***.778.816.** | ***.781.202-** | Adriano Melo de Paula | Deferido | |
| 31 | ***.403.636.** | ***.425.722-** | Adriano Monteiro Rolim | Deferido | |
| 32 | ***.468.957.** | ***.903.772-** | Adriano Nunes de Souza | Deferido | |
| 33 | ***.687.286.** | ***.644.732-** | Adriano Queiroz da Cruz | Deferido | |
| 34 | ***.213.985.** | ***.079.282-** | Adriano Vieira da Silva Junior | Deferido | |
| 35 | ***.379.386.** | ***.776.222-** | Adriel da Silva Bezerra | Deferido | |
| 36 | ***.092.078.** | ***.345.162-** | Adriel da Silva Vidinho | Deferido | |
| 37 | ***.111.828.** | ***.138.782-** | Adriel Marques de Souza | Deferido | |
| 38 | ***.408.828.** | ***.029.632-** | Adrielle Braga da Silva | Deferido | |
| 39 | ***.340.587.** | ***.760.282-** | Adrielle Carvalho Lima | Deferido | |
| 40 | ***.657.819.** | ***.360.752-** | Adrielle da Silva dos Santos | Deferido | |
| 41 | ***.594.286.** | ***.915.342-** | Adrielle Ferreira da Costa | Deferido | |
| 42 | ***.114.086.** | ***.686.822-** | Adrielle Socorro Trindade Barros | Deferido | |
| 43 | ***.709.085.** | ***.902.912-** | Adrielle Lopes de Oliveira | Deferido | |
| 44 | ***.706.487.** | ***.038.172-** | Adrielly Lopes de Amorim | Deferido | |
| 45 | ***.666.706.** | ***.054.952-** | Adrielly Silva de Souza | Deferido | |
| 46 | ***.363.346.** | ***.211.062-** | Adriely Ramires Nascimento | Deferido | |
| 47 | ***.255.857.** | ***.781.182-** | Adriene Cristina da Silva Castro | Deferido | |
| 48 | ***.551.437.** | ***.830.692-** | Adrinne da Silva Mota | Deferido | |
| 49 | ***.499.516.** | ***.223.422-** | Adrya Luanny Batista Viana | Deferido | |
| 50 | ***.717.369.** | ***.496.482-** | Adrya Rebeca de Oliveira Nogueira | Deferido | |
| 51 | ***.287.090.** | ***.711.172-** | Adryana de Oliveira Lopes | Deferido | |
| 52 | ***.308.031.** | ***.482.232-** | Adryellen Oliveira de Carvalho | Deferido | |
| 53 | ***.545.964.** | ***.884.962-** | Adson Levi Rodrigues Rocha | Deferido | |
| 54 | ***.004.101.** | ***.974.032-** | Adson Militao Fidelis | Deferido | |

| Ord. | NIS | CPF | Nome | Situação | Motivo |
|------|----------------|----------------|--------------------------------------|------------|--------|
| 3008 | ***.375.459.** | ***.295.092-** | Yasmim Moreira Auzier | Deferido | |
| 3009 | ***.717.204.** | ***.195.712-** | Yasmim Rocha da Costa | Deferido | |
| 3010 | ***.290.025.** | ***.162.652-** | Yasmim Victoria Fernandes de Lima | Deferido | |
| 3011 | ***.482.619.** | ***.785.002-** | Yasmin Almeida da Silva | Deferido | |
| 3012 | ***.117.782.** | ***.118.762-** | Yasmin Andrade Correia | Deferido | |
| 3013 | ***.396.668.** | ***.342.772-** | Yasmin Cardoso Nunes | Deferido | |
| 3014 | ***.081.705.** | ***.038.292-** | Yasmin da Silva Figueiredo | Deferido | |
| 3015 | ***.137.595.** | ***.244.562-** | Yasmin Gabriely Silva de Araujo | Deferido | |
| 3016 | ***.560.625.** | ***.656.752-** | Yasmin Gloria Costa de Aguiar Silva | Deferido | |
| 3017 | ***.375.466.** | ***.683.542-** | Yasmin Larissa de Souza Cruz | Deferido | |
| 3018 | ***.327.735.** | ***.355.392-** | Yasmin Lima dos Santos | Deferido | |
| 3019 | ***.427.011.** | ***.621.192-** | Yasmin Luiza Medeiros Lemos | Deferido | |
| 3020 | ***.760.857.** | ***.504.882-** | Yasmin Pio Souza | Deferido | |
| 3021 | ***.436.097.** | ***.436.972-** | Yasmin Souza da Silva | Deferido | |
| 3022 | ***.327.207.** | ***.588.402-** | Yeda Creuzy Ribeiro Sodre | Deferido | |
| 3023 | ***.344.182.** | ***.549.512-** | Yego Pimentel Batista | Deferido | |
| 3024 | ***.081.318.** | ***.247.192-** | Ynaiara Alfredo Florentino | Deferido | |
| 3025 | ***.623.968.** | ***.651.262-** | Yohanna Victoria Torres da Silva | Deferido | |
| 3026 | ***.564.837.** | ***.226.932-** | Yohise Marina Lopez Gonzalez | Deferido | |
| 3027 | ***.358.380.** | ***.583.002-** | Yris Fabiane Ribeiro Valente | Deferido | |
| 3028 | ***.853.257.** | ***.036.082-** | Yulieth de Los Angeles Garcia Moreno | Deferido | |
| 3029 | ***.406.161.** | ***.570.272-** | Yuri de Paula Brito | Deferido | |
| 3030 | ***.599.012.** | ***.711.762-** | Zamily de Paula Cunha | Deferido | |
| 3031 | ***.599.016.** | ***.711.912-** | Zemily de Paula Cunha | Deferido | |
| 3032 | ***.796.288.** | ***.285.352-** | Zimmer Gabriel Del Aguila Yactayo | Deferido | |
| 3033 | ***.273.150.** | ***.016.532-** | 05801653295 | Indeferido | 1 |
| 3034 | ***.478.250.** | ***.601.312-** | Aaao Auzier Anselmo Junior | Indeferido | 13 |
| 3035 | ***.627.910.** | ***.799.492-** | Abel Dias Leite | Indeferido | 7 |
| 3036 | ***.979.280.** | ***.474.142-** | Abel Seixas Bisneto | Indeferido | 7 |
| 3037 | ***.690.948.** | ***.537.522-** | Abigail Yasmim Falcao Barbosa | Indeferido | 4 |
| 3038 | ***.962.630.** | ***.415.962-** | Abraao Caue da Silva Pantoja | Indeferido | 7 |
| 3039 | ***.436.880.** | ***.003.852-** | Abraao da Silva Lima | Indeferido | 7 |
| 3040 | ***.047.187.** | ***.561.862-** | Abraao Paes de Faria | Indeferido | 3 |
| 3041 | ***.372.211.** | ***.428.262-** | Abraao Tacafais de Lima | Indeferido | 3 |
| 3042 | ***.430.653.** | ***.236.172-** | Abraham Del Jesus Hidalgo Boada | Indeferido | 3 |
| 3043 | ***.117.755.** | ***.043.832-** | Ackson Gabriel Silva da Silva | Indeferido | 3 |
| 3044 | ***.369.834.** | ***.042.742-** | Adam Iohannson da Silva Romao | Indeferido | 3 |
| 3045 | ***.774.288.** | ***.060.262-** | Adam Lucas Lemos Fonseca | Indeferido | 4 |

| Ord. | NIS | CPF | Nome | Situação | Motivo |
|------|---------------|----------------|--|------------|--------|
| 3046 | ***.633.314.* | ***.634.492.** | Adan Henrique Vasconcelos dos Santos | Indeferido | 4 |
| 3047 | ***.687.952.* | ***.737.752.** | Ademar Batista Moreira Neto | Indeferido | 3 |
| 3048 | ***.154.682.* | ***.154.682.** | Adenilson Jordao Romana | Indeferido | 3 |
| 3049 | ***.212.752.* | ***.715.192.** | Adenilson Tavares e Tavares | Indeferido | 7 |
| 3050 | ***.251.950.* | ***.946.622.** | Adiel Gustavo da Silva Xavier | Indeferido | 7 |
| 3051 | ***.477.856.* | ***.928.112.** | Adila Aimee Gomes Andrade | Indeferido | 3 |
| 3052 | ***.333.616.* | ***.943.662.** | Adilson Machado Mourao | Indeferido | 3 |
| 3053 | ***.212.054.* | ***.543.372.** | Adley Paiva Melo | Indeferido | 3 |
| 3054 | ***.213.797.* | ***.982.322.** | Adonias da Silva Feitosa | Indeferido | 7 |
| 3055 | ***.505.696.* | ***.805.732.** | Adria Alves Brito | Indeferido | 7 |
| 3056 | ***.859.930.* | ***.185.432.** | Adria Caroline Lima da Silva | Indeferido | 7 |
| 3057 | ***.842.472.* | ***.842.472.** | Adria Kariny Pereira Oliveira | Indeferido | 3 |
| 3058 | ***.096.126.* | ***.628.172.** | Adria Raiane Vieira Cavalcante | Indeferido | 7 |
| 3059 | ***.304.047.* | ***.802.032.** | Adriana Curica de Souza | Indeferido | 7 |
| 3060 | ***.482.524.* | ***.426.662.** | Adriane Moraes Tavares | Indeferido | 7 |
| 3061 | ***.370.677.* | ***.521.222.** | Adriane Paiva Pinto | Indeferido | 13 |
| 3062 | ***.341.602.* | ***.311.092.** | Adriane Stefanelly Siqueira Ramires | Indeferido | 7 |
| 3063 | ***.111.810.* | ***.332.912.** | Adriano | Indeferido | 7 |
| 3064 | ***.169.400.* | ***.382.832.** | Adriano Cesar Belem Jefferson | Indeferido | 4 |
| 3065 | ***.061.305.* | ***.970.532.** | Adriano da Silva Martins | Indeferido | 7 |
| 3066 | ***.065.519.* | ***.890.772.** | Adriano Elerson de Carvalho Trindade | Indeferido | 7 |
| 3067 | ***.823.992.* | ***.013.922.** | Adriano Ferreira Medeiros | Indeferido | 7 |
| 3068 | ***.822.083.* | ***.374.972.** | Adriano Galvao Jacauna | Indeferido | 3 |
| 3069 | ***.449.432.* | ***.948.742.** | Adriano Henrique de Lima Noriega | Indeferido | 7 |
| 3070 | ***.000.350.* | ***.721.132.** | Adriano Josias da Silva Rengifo | Indeferido | 7 |
| 3071 | ***.582.996.* | ***.837.272.** | Adriel Bentes Godinho | Indeferido | 3 |
| 3072 | ***.394.745.* | ***.188.282.** | Adriel Lima Neves | Indeferido | 7 |
| 3073 | ***.558.220.* | ***.699.762.** | Adriel Marques Tavares | Indeferido | 7 |
| 3074 | ***.987.169.* | ***.112.382.** | Adriele Brito da Silva | Indeferido | 7 |
| 3075 | ***.908.456.* | ***.706.402.** | Adriele Sales Basques | Indeferido | 3 |
| 3076 | ***.582.228.* | ***.644.032.** | Adrielle Brasil da Silva | Indeferido | 7 |
| 3077 | ***.726.374.* | ***.162.512.** | Adrielle Goes Castro | Indeferido | 7 |
| 3078 | ***.561.270.* | ***.070.512.** | Adriellen da Costa Nunes | Indeferido | 7 |
| 3079 | ***.096.313.* | ***.361.462.** | Adrielly dos Santos Rocha | Indeferido | 7 |
| 3080 | ***.377.110.* | ***.479.022.** | Adriely Cauane Silva Pereira | Indeferido | 7 |
| 3081 | ***.005.902.* | ***.005.902.** | Adriely Rocha de Souza | Indeferido | 3 |
| 3082 | ***.085.820.* | ***.700.562.** | Adriely Silva Gama | Indeferido | 7 |
| 3083 | ***.618.970.* | ***.877.702.** | Adrielyson Guilherme Costa de Oliveira | Indeferido | 7 |

| Ord. | NIS | CPF | Nome | Situação | Motivo |
|------|----------------|----------------|---|------------|--------|
| 7611 | ***.390.270.** | ***.637.952-** | Yasmin Barroso do Nascimento Rodrigues | Indeferido | 4 |
| 7612 | ***.581.810.** | ***.502.482-** | Yasmin Beatriz Cordeiro de Oliveira | Indeferido | 7 |
| 7613 | ***.095.919.** | ***.793.362-** | Yasmin da Costa Maricaú | Indeferido | 7 |
| 7614 | ***.722.100.** | ***.068.392-** | Yasmin da Silva e Silva | Indeferido | 7 |
| 7615 | ***.652.161.** | ***.304.652-** | Yasmin da Silva Ferreira | Indeferido | 3 |
| 7616 | ***.276.610.** | ***.877.742-** | Yasmin de Souza Cardoso | Indeferido | 4 |
| 7617 | ***.165.197.** | ***.058.872-** | Yasmin dos Santos Souza | Indeferido | 7 |
| 7618 | ***.263.607.** | ***.443.972-** | Yasmin Elias da Costa | Indeferido | 4 |
| 7619 | ***.891.085.** | ***.222.322-** | Yasmin Gabriela Silva dos Santos | Indeferido | 7 |
| 7620 | ***.815.922.** | ***.815.922-** | Yasmin Gabrielle Marques Pety | Indeferido | 3 |
| 7621 | ***.408.692.** | ***.408.692-** | Yasmin Gabrielly Ribeiro dos Santos | Indeferido | 3 |
| 7622 | ***.476.145.** | ***.504.732-** | Yasmin Lopes Santos | Indeferido | 5 |
| 7623 | ***.355.264.** | ***.735.612-** | Yasmin Monteiro dos Santos | Indeferido | 7 |
| 7624 | ***.578.114.** | ***.940.202-** | Yasmin Moraes da Mota | Indeferido | 3 |
| 7625 | ***.405.887.** | ***.735.742-** | Yasmin Nerily de Araujo Alencar Repolho | Indeferido | 3 |
| 7626 | ***.920.200.** | ***.345.962-** | Yasmin Pinto de Sales | Indeferido | 7 |
| 7627 | ***.393.393.** | ***.329.222-** | Yasmin Rebecca de Souza Dutra | Indeferido | 7 |
| 7628 | ***.829.660.** | ***.275.882-** | Yasmin Rodrigues dos Santos | Indeferido | 4 |
| 7629 | ***.014.702.** | ***.825.862-** | Yasmin Santos de Assis | Indeferido | 7 |
| 7630 | ***.799.000.** | ***.137.112-** | Yasmin Sayuri de Albuquerque Sobrinho | Indeferido | 3 |
| 7631 | ***.845.500.** | ***.705.062-** | Yasmin Silva Simplicio | Indeferido | 4 |
| 7632 | ***.174.680.** | ***.542.702-** | Yasmin Victoria dos Anjos Ladislau | Indeferido | 7 |
| 7633 | ***.818.659.** | ***.596.072-** | Yasmin Victoria Dr Oliveira Cavalcante | Indeferido | 7 |
| 7634 | ***.537.374.** | ***.010.182-** | Yasmin Vitoria Marinho da Rosa | Indeferido | 7 |
| 7635 | ***.813.639.** | ***.570.762-** | Yasmin Vitoria Prestes Chaves | Indeferido | 7 |
| 7636 | ***.324.652.** | ***.771.812-** | Yasmin Zagury Rodrigues | Indeferido | 7 |
| 7637 | ***.355.549.** | ***.065.532-** | Yasna de Sousa da Silva | Indeferido | 7 |
| 7638 | ***.601.484.** | ***.548.072-** | Ycaro Araujo dos Santos | Indeferido | 7 |
| 7639 | ***.578.121.** | ***.897.702-** | Ycaro Lima Lopes | Indeferido | 4 |
| 7640 | ***.342.020.** | ***.960.602-** | Yetzabeth Carolina Bandres Bandres | Indeferido | 7 |
| 7641 | ***.301.883.** | ***.988.152-** | Ygor Jose Lopes Porfirio | Indeferido | 3 |
| 7642 | ***.094.122.** | ***.815.632-** | Yohane Pinto Lima | Indeferido | 4 |
| 7643 | ***.324.020.** | ***.550.022-** | Yule Agape Alves de Almeida | Indeferido | 7 |
| 7644 | ***.054.103.** | ***.711.382-** | Yurhi Prestes da Silva | Indeferido | 7 |
| 7645 | ***.939.442.** | ***.427.902-** | Yuri Renato da Silva Marinho | Indeferido | 7 |
| 7646 | ***.107.278.** | ***.089.432-** | Yuri Santos de Sousa | Indeferido | 7 |
| 7647 | ***.584.954.** | ***.232.382-** | Yzana Railana Pinheiro dos Santos | Indeferido | 7 |

| Ord. | NIS | CPF | Nome | Situação | Motivo |
|------|-----------------|-----------------|------------------------------|------------|--------|
| 7648 | ***.352.701.*-* | ***.622.442-**- | Zacarias Martins de Oliveira | Indeferido | 7 |
| 7649 | ***.318.327.*-* | ***.703.632-**- | Zaiane Azevedo e Costa | Indeferido | 7 |
| 7650 | ***.973.268.*-* | ***.004.912-**- | Zamira Dasilva Colares | Indeferido | 4 |
| 7651 | ***.382.351.*-* | ***.734.302-**- | Zian Satana Moraes | Indeferido | 3 |
| 7652 | ***.923.952.*-* | ***.923.952-**- | Zimaely da Cruz Araujo | Indeferido | 3 |

ANEXO B – Relatório de Similaridade
Aplicado no PSC 2023 (apenas algumas
páginas)



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



Relatório de Similaridade e Distância entre Nomes Processo Seletivo Contínuo - PSC 2023 - Etapa 1

Este relatório utiliza o [algoritmo de Jaro-Winkler](#) para cálculo das similaridades entre os nomes de candidatos solicitantes de isenção de taxa de inscrição. Em resumo, o algoritmo compara os dois nomes e calcula uma porcentagem de semelhança entre eles.

* **Similaridade:** quanto mais próximo de 100%, significa que houve pouca alteração entre o nome informado na solicitação de isenção e o nome informado no recurso. Similaridades abaixo de 75% podem significar uma grande mudança no nome!

* **Distância:** quanto mais próximo de 0%, significa que houve pouca alteração entre o nome informado na solicitação de isenção e o nome informado no recurso. Distâncias acima de 25% podem significar uma grande mudança no nome!

Obs.: para fins de cálculo, são envolvidos apenas os candidatos deferidos após a solicitação de recurso.

| Ord. | Nome [Solicitação] | Nome [Recurso] | Similaridade | Distância |
|------|--------------------------------|--------------------------------|--------------|-----------|
| 1 | Gabriella Costa Prata | Josciane Silva Costa | 57.10 % | 42.90 % |
| 2 | Hanny Hadassa Monteiro Martins | Maria Eliza Alves Monteiro | 58.33 % | 41.67 % |
| 3 | Thais Teixeira da Silva | Claudecirla Oliveira Teixeira | 61.37 % | 38.63 % |
| 4 | Thaissa Gomes de Souza | Luziete da Silva Gomes | 61.47 % | 38.53 % |
| 5 | Dilacy Nathalia Silva do | Angelica da Silva e Silva | 62.73 % | 37.27 % |
| 6 | Pedro Chagas Nunes | Pedro Chagas da Costa | 88.89 % | 11.11 % |
| 7 | Naylla Taynne Rosas da Silva | Naylla Tayanne Rosas da Silva | 93.24 % | 6.76 % |
| 8 | Evelyn Mariany Castro Lima | Evelyn Mayara Castro Lima | 95.16 % | 4.84 % |
| 9 | Pablo Willian de Souza | Pablo Willian de Souza Farias | 95.17 % | 4.83 % |
| 10 | Kenisson Marques Pereira | Kenisson Marques Pereira | 95.87 % | 4.13 % |
| 11 | Sydon Mourao Sousa | Sydon Mourao Souza | 97.78 % | 2.22 % |
| 12 | Carine Freitas Cavalcante | Carine Freitas Cavalcante | 98.40 % | 1.60 % |
| 13 | Viviane Tavares Pimentel | Viviane Tavares Pimentel | 99.20 % | 0.80 % |
| 14 | Hillary de Arujo de Aquino | Hillary de Araujo de Aquino | 99.26 % | 0.74 % |
| 15 | Julliana do Nascimento Sousa | Julliana do Nascimento Sousa | 99.29 % | 0.71 % |
| 16 | Adriane de Oliveira Seixas | Adriane de Oliveira Seixas | 100.00 % | 0.00 % |
| 17 | Adrielly Silva de Souza | Adrielly Silva de Souza | 100.00 % | 0.00 % |
| 18 | Adryana de Oliveira Lopes | Adryana de Oliveira Lopes | 100.00 % | 0.00 % |
| 19 | Agape Vitoria Lopes da Costa | Agape Vitoria Lopes da Costa | 100.00 % | 0.00 % |
| 20 | Agata Malrine Martins da Silva | Agata Malrine Martins da Silva | 100.00 % | 0.00 % |
| 21 | Agatha Rayana Otaviano de | Agatha Rayana Otaviano de | 100.00 % | 0.00 % |
| 22 | Alan Rodrigues Braga | Alan Rodrigues Braga | 100.00 % | 0.00 % |
| 23 | Alana Costa Queiroz | Alana Costa Queiroz | 100.00 % | 0.00 % |
| 24 | Alciane Ferreira Vieira | Alciane Ferreira Vieira | 100.00 % | 0.00 % |
| 25 | Alessandro Farias da Costa | Alessandro Farias da Costa | 100.00 % | 0.00 % |
| 26 | Alexa Leticia de Freitas dos | Alexa Leticia de Freitas dos | 100.00 % | 0.00 % |

| Ord. | Nome [Solicitação] | Nome [Recurso] | Similaridade | Distância |
|------|----------------------------------|----------------------------------|--------------|-----------|
| 179 | Samuel de Lima Soares | Samuel de Lima Soares | 100.00 % | 0.00 % |
| 180 | Sara Eliza Smith da Silva | Sara Eliza Smith da Silva | 100.00 % | 0.00 % |
| 181 | Sara Vieira de Castro Cunha | Sara Vieira de Castro Cunha | 100.00 % | 0.00 % |
| 182 | Sidelmar de Azevedo de Almeida | Sidelmar de Azevedo de Almeida | 100.00 % | 0.00 % |
| 183 | Soraya Rodrigues Galvao | Soraya Rodrigues Galvao | 100.00 % | 0.00 % |
| 184 | Tais Pereira Bastos | Tais Pereira Bastos | 100.00 % | 0.00 % |
| 185 | Thais da Silva de Sa | Thais da Silva de Sa | 100.00 % | 0.00 % |
| 186 | Thaiza Gomes da Silva | Thaiza Gomes da Silva | 100.00 % | 0.00 % |
| 187 | Thayla Mirela Pontes Alfaia | Thayla Mirela Pontes Alfaia | 100.00 % | 0.00 % |
| 188 | Thayna Cunha da Costa | Thayna Cunha da Costa | 100.00 % | 0.00 % |
| 189 | Ully Akebi Ferreira Pedrosa | Ully Akebi Ferreira Pedrosa | 100.00 % | 0.00 % |
| 190 | Valciney Pinheiro Gaspar | Valciney Pinheiro Gaspar | 100.00 % | 0.00 % |
| 191 | Valeria Bastos Canavarro | Valeria Bastos Canavarro | 100.00 % | 0.00 % |
| 192 | Victoria Hatcherian Lobo | Victoria Hatcherian Lobo | 100.00 % | 0.00 % |
| 193 | Victoria Queiroz da Silva | Victoria Queiroz da Silva | 100.00 % | 0.00 % |
| 194 | Vitor Ribeiro Soares | Vitor Ribeiro Soares | 100.00 % | 0.00 % |
| 195 | Vitoria de Oliveira Santos | Vitoria de Oliveira Santos | 100.00 % | 0.00 % |
| 196 | Vitoria Regina Moreira de | Vitoria Regina Moreira de | 100.00 % | 0.00 % |
| 197 | Vivian Miranda de Castro | Vivian Miranda de Castro | 100.00 % | 0.00 % |
| 198 | Viviane Nobre de Lima | Viviane Nobre de Lima | 100.00 % | 0.00 % |
| 199 | Wakson Ramos Batista | Wakson Ramos Batista | 100.00 % | 0.00 % |
| 200 | Yara Camila Costa da Silva | Yara Camila Costa da Silva | 100.00 % | 0.00 % |
| 201 | Yohanna Victoria Torres da Silva | Yohanna Victoria Torres da Silva | 100.00 % | 0.00 % |

Documento extraído em: 24/06/2023 às 23:35:47

ANEXO C – Relatório de Estatísticas de Processamento (apenas algumas páginas)



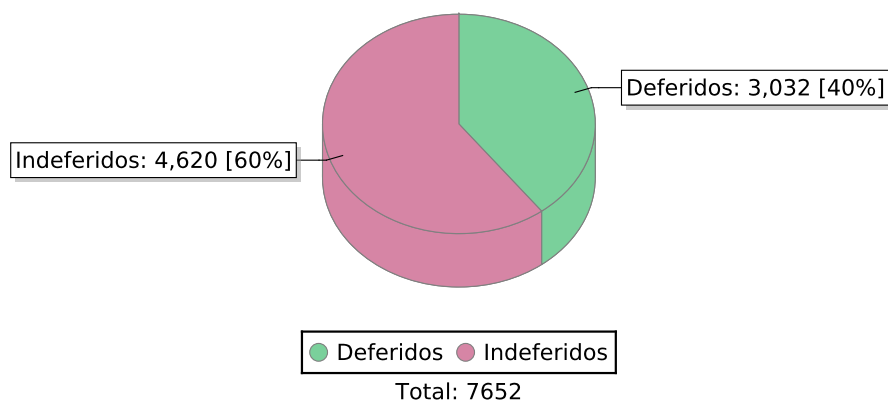
Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



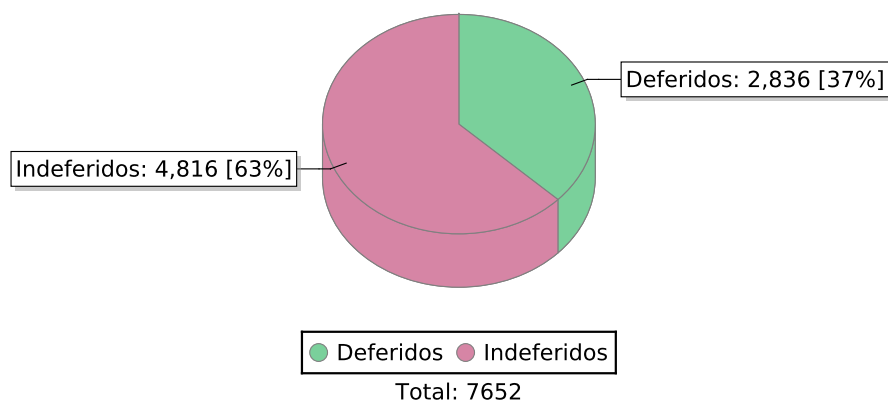
UFAM

Relatório de Processamento - Resultado Definitivo
Processo Seletivo Contínuo - PSC 2023 - Etapa 1 [Retificado em 25/06/2023]

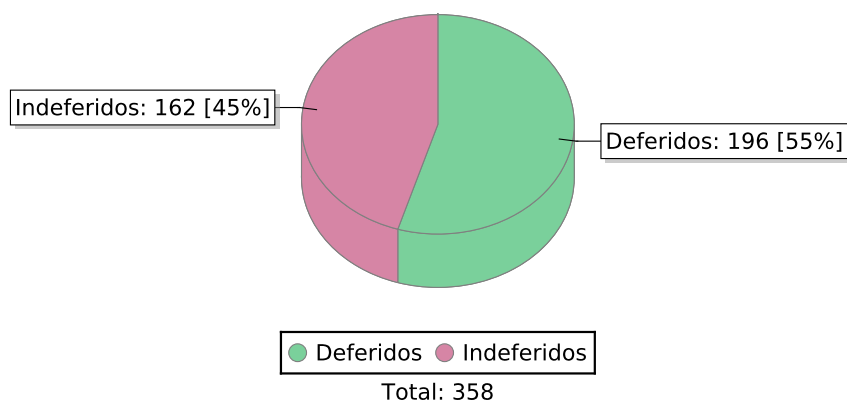
*** Totais [Resultado Definitivo]**



*** Totais [Antes do Recurso]**



*** Totais [Recurso]**





Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



*** Lista de Novos Deferidos**

| Ord. | Nome | CPF | NIS |
|------|-------------------------------------|----------------|-----------------|
| 1 | Adriane de Oliveira Seixas | ***.977.842-** | ***.334.549.*-* |
| 2 | Adrielly Silva de Souza | ***.054.952-** | ***.666.706.*-* |
| 3 | Adryana de Oliveira Lopes | ***.711.172-** | ***.287.090.*-* |
| 4 | Agape Vitoria Lopes da Costa | ***.200.602-** | ***.180.890.*-* |
| 5 | Agata Malrine Martins da Silva | ***.783.332-** | ***.192.111.*-* |
| 6 | Agatha Rayana Otaviano de Souza | ***.935.532-** | ***.999.744.*-* |
| 7 | Alan Rodrigues Braga | ***.688.112-** | ***.208.872.*-* |
| 8 | Alana Costa Queiroz | ***.677.932-** | ***.471.906.*-* |
| 9 | Alciane Ferreira Vieira | ***.640.342-** | ***.166.472.*-* |
| 10 | Alessandro Farias da Costa | ***.686.532-** | ***.523.199.*-* |
| 11 | Alexa Leticia de Freitas dos Santos | ***.415.272-** | ***.403.686.*-* |
| 12 | Ana Beatriz Ozaki Teixeira | ***.258.932-** | ***.340.899.*-* |
| 13 | Ana Clara Viana Lima | ***.751.662-** | ***.069.300.*-* |
| 14 | Ana Francisca da Costa Lima | ***.925.472-** | ***.147.973.*-* |
| 15 | Ana Gabriela Penha Tavares | ***.804.412-** | ***.780.356.*-* |
| 16 | Ana Julia Lago Xavier | ***.754.232-** | ***.678.703.*-* |
| 17 | Analice Ferreira | ***.046.432-** | ***.331.177.*-* |
| 18 | Analice Silva de Souza | ***.677.762-** | ***.270.246.*-* |
| 19 | Andre Lucas de Oliveira Cordeiro | ***.148.292-** | ***.293.252.*-* |
| 20 | Andre Rodrigues de Lima Filho | ***.522.532-** | ***.504.863.*-* |
| 21 | Andreize da Silva Xavier | ***.567.352-** | ***.336.621.*-* |
| 22 | Andrielly Cavalcante da Silva | ***.802.452-** | ***.028.889.*-* |
| 23 | Andryo Romualdo de Freitas | ***.762.542-** | ***.549.167.*-* |
| 24 | Anna Caroliny Souza Ramos | ***.179.462-** | ***.309.128.*-* |
| 25 | Arnando de Souza Xavier | ***.557.432-** | ***.446.407.*-* |
| 26 | Ayna Martins da Silva | ***.619.542-** | ***.456.748.*-* |
| 27 | Beatriz dos Santos Ricardo | ***.715.222-** | ***.530.521.*-* |
| 28 | Bianca Andrade Marques | ***.628.022-** | ***.161.674.*-* |
| 29 | Camila Barbosa da Silva | ***.007.142-** | ***.445.224.*-* |
| 30 | Carine Freitas Cavalcante | ***.164.752-** | ***.252.123.*-* |
| 31 | Carine Victoria Costa da Silva | ***.815.342-** | ***.393.806.*-* |
| 32 | Cassio Peres Maia | ***.160.592-** | ***.027.629.*-* |
| 33 | Celso Cardoso da Costa Neto | ***.797.262-** | ***.800.569.*-* |



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



| Ord. | Nome | CPF | NIS |
|------|------------------------------------|----------------|-----------------|
| 170 | Ryan Campos da Silva | ***.004.302-** | ***.814.573.*.* |
| 171 | Sabrina Ferreira Teixeira | ***.354.622-** | ***.903.186.*.* |
| 172 | Samuel de Lima Soares | ***.513.192-** | ***.424.390.*.* |
| 173 | Sara Eliza Smith da Silva | ***.359.452-** | ***.617.886.*.* |
| 174 | Sara Vieira de Castro Cunha | ***.787.272-** | ***.704.201.*.* |
| 175 | Sidelmar de Azevedo de Almeida | ***.495.152-** | ***.055.055.*.* |
| 176 | Soraya Rodrigues Galvao | ***.264.202-** | ***.533.675.*.* |
| 177 | Sydon Mourao Souza | ***.964.012-** | ***.762.613.*.* |
| 178 | Tais Pereira Bastos | ***.812.302-** | ***.781.894.*.* |
| 179 | Thais da Silva de Sa | ***.982.532-** | ***.688.874.*.* |
| 180 | Thaiza Gomes da Silva | ***.790.182-** | ***.471.355.*.* |
| 181 | Thayla Mirela Pontes Alfaia | ***.408.402-** | ***.131.052.*.* |
| 182 | Thayna Cunha da Costa | ***.888.072-** | ***.831.941.*.* |
| 183 | Uly Akebi Ferreira Pedrosa | ***.158.432-** | ***.887.349.*.* |
| 184 | Valciney Pinheiro Gaspar | ***.814.742-** | ***.459.528.*.* |
| 185 | Valeria Bastos Canavarro | ***.225.862-** | ***.676.114.*.* |
| 186 | Victoria Hatcherian Lobo | ***.357.002-** | ***.521.504.*.* |
| 187 | Victoria Queiroz da Silva | ***.901.622-** | ***.928.672.*.* |
| 188 | Vitor Ribeiro Soares | ***.176.022-** | ***.031.631.*.* |
| 189 | Vitoria de Oliveira Santos | ***.836.882-** | ***.983.644.*.* |
| 190 | Vitoria Regina Moreira de Carvalho | ***.637.072-** | ***.715.020.*.* |
| 191 | Vivian Miranda de Castro | ***.066.342-** | ***.444.147.*.* |
| 192 | Viviane Nobre de Lima | ***.642.192-** | ***.812.251.*.* |
| 193 | Viviane Tavares Pimentel | ***.274.082-** | ***.658.175.*.* |
| 194 | Wakson Ramos Batista | ***.634.452-** | ***.457.319.*.* |
| 195 | Yara Camila Costa da Silva | ***.531.542-** | ***.121.560.*.* |
| 196 | Yohanna Victoria Torres da Silva | ***.651.262-** | ***.623.968.*.* |

*** Lista de Recursantes Indeferidos**

| Ord. | Nome | CPF | NIS |
|------|-------------------------------|----------------|-----------------|
| 1 | Adam Iohannson da Silva Romao | ***.042.742-** | ***.369.834.*.* |
| 2 | Adam Lucas Lemos Fonseca | ***.060.262-** | ***.774.288.*.* |
| 3 | Adrya Nunes de Sales | ***.110.912-** | ***.349.936.*.* |
| 4 | Alaene Maciel da Silva | ***.213.182-** | ***.214.099.*.* |
| 5 | Alex Romaina Fermino | ***.076.922-** | ***.004.959.*.* |



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



| Ord. | Nome | CPF | NIS |
|------|---|----------------|-----------------|
| 6 | Ana Beatriz Agostinho Gomes | ***.581.722-** | ***.249.551.*.* |
| 7 | Ana Beatriz Sena de Lima | ***.919.432-** | ***.459.090.*.* |
| 8 | Ana Carolina Farias de Souza | ***.552.952-** | ***.906.051.*.* |
| 9 | Ana Clara Oliveira de Castro | ***.777.862-** | ***.136.954.*.* |
| 10 | Ana Flavia Seixas Bahia | ***.563.152-** | ***.212.076.*.* |
| 11 | Ana Karoliny Castilha da Silva | ***.163.662-** | ***.254.337.*.* |
| 12 | Ana Mirelle Goncalves Ferreira | ***.850.262-** | ***.149.711.*.* |
| 13 | Anderson de Sousa dos Santos | ***.468.692-** | ***.374.134.*.* |
| 14 | Andreina Xavier Lopes | ***.068.872-** | ***.380.439.*.* |
| 15 | Andrey Raymundo de Souza de Araujo | ***.428.092-** | ***.589.778.*.* |
| 16 | Andrya Vitoria de Souza Macedo | ***.800.032-** | ***.719.550.*.* |
| 17 | Angelica da Silva e Silva | ***.671.722-** | ***.151.825.*.* |
| 18 | Anny Gabryelly Rosas Pena | ***.786.122-** | ***.731.163.*.* |
| 19 | Athos Agnaldo Sousa da Silva | ***.852.352-** | ***.237.766.*.* |
| 20 | Augusto Ythalo Monteiro Gonzaga | ***.768.672-** | ***.077.064.*.* |
| 21 | Beatriz Lopes de Sa | ***.561.272-** | ***.499.899.*.* |
| 22 | Bianca Castro Ribeiro | ***.891.852-** | ***.315.820.*.* |
| 23 | Boehner Manoel Ramos Batista | ***.909.632-** | ***.295.497.*.* |
| 24 | Camilly Gabriele Pessoa de Franca | ***.707.572-** | ***.707.572.*.* |
| 25 | Carla dos Santos Ramos | ***.455.862-** | ***.591.576.*.* |
| 26 | Carla Ferreira de Oliveira | ***.425.022-** | ***.229.276.*.* |
| 27 | Carla Isabele Silva Furtado de Benedito | ***.418.712-** | ***.871.731.*.* |
| 28 | Carlos Alberto de Brito Moxi | ***.725.112-** | ***.356.413.*.* |
| 29 | Carlos Eduardo Barbosa Lima | ***.938.392-** | ***.159.415.*.* |
| 30 | Clara Moraes Cavalcante | ***.855.822-** | ***.858.689.*.* |
| 31 | Clarice Moraes Cavalcante | ***.835.112-** | ***.858.692.*.* |
| 32 | Claudecira Oliveira Teixeira | ***.969.192-** | ***.115.150.*.* |
| 33 | Danyel Felipe Alves de Souza | ***.521.812-** | ***.093.412.*.* |
| 34 | Davi Wemerson Andrade Pinheiro | ***.626.862-** | ***.787.703.*.* |
| 35 | Debora Andrade de Souza | ***.882.492-** | ***.198.035.*.* |
| 36 | Deivila Emanuelle Coello Perrone | ***.254.792-** | ***.204.518.*.* |
| 37 | Eduarda Araujo da Silva | ***.006.392-** | ***.249.671.*.* |
| 38 | Eduardo Gomes Barros | ***.001.692-** | ***.224.532.*.* |
| 39 | Elane Ferreira dos Santos | ***.798.832-** | ***.418.114.*.* |



Poder Executivo
Ministério da Educação
Universidade Federal do Amazonas
Comissão Permanente de Concursos



| Ord. | Nome | CPF | NIS |
|------|--|----------------|-----------------|
| 142 | Thayla Kamilly de Souza Bentes | ***.617.852-** | ***.056.778.*.* |
| 143 | Thiago dos Santos Lopes | ***.232.482-** | ***.216.380.*.* |
| 144 | Tiago Paixao Lima | ***.224.732-** | ***.621.765.*.* |
| 145 | Valdenor Menezes da Fonseca | ***.961.852-** | ***.498.444.*.* |
| 146 | Victoria Gabriele Menezes de Azevedo | ***.756.772-** | ***.744.007.*.* |
| 147 | Vitor Daniel de Azevedo Liarte | ***.241.172-** | ***.008.810.*.* |
| 148 | Vitor Gabriel Pereira da Silva | ***.632.152-** | ***.400.348.*.* |
| 149 | Vitoria Santos Gabriel | ***.222.192-** | ***.118.602.*.* |
| 150 | Walter Souza Lima Bisneto | ***.204.082-** | ***.200.313.*.* |
| 151 | Wendel Adson Lira Dantas | ***.007.002-** | ***.341.082.*.* |
| 152 | Wiglesson Soares da Silva | ***.833.992-** | ***.785.309.*.* |
| 153 | William Vasconcelos de Oliveira | ***.646.382-** | ***.759.084.*.* |
| 154 | Wisley da Silva Campos | ***.709.332-** | ***.432.548.*.* |
| 155 | Wylham Vitor Estevam da Silva | ***.989.142-** | ***.303.399.*.* |
| 156 | Yan de Lima Guedes | ***.140.152-** | ***.234.920.*.* |
| 157 | Yan de Souza Aguiar | ***.681.522-** | ***.574.913.*.* |
| 158 | Yasmim Stefany Oliveira Vieira | ***.355.432-** | ***.703.422.*.* |
| 159 | Yasmin Rebecca de Souza Dutra | ***.329.222-** | ***.393.393.*.* |
| 160 | Yasmin Victoria dos Anjos Ladislau | ***.542.702-** | ***.174.680.*.* |
| 161 | Yasmin Victoria Dr Oliveira Cavalcante | ***.596.072-** | ***.818.659.*.* |
| 162 | Viviane Pedrosa Freitas | ***.295.752-** | NÃO TENHO |

* Lista de Arquivos Processados

| Ord. | Tipo de Arquivo | Nome do Arquivo |
|------|-------------------|--|
| 1 | Compilação | COMPILACAO_04378626000197_132023_05062023.icf |
| 2 | Retorno do Sistac | RETORNO_04378626000197_132023_25062023_001.txt |
| 3 | Planilha de Erros | ERROS_04378626000197_132023_25062023.xlsx |

Documento extraído em: 24/06/2023 às 23:54:43