

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

EDUARDA LARA WALDEMAR

**CLASSIFICAÇÃO AUTOMÁTICA DE SERVIDORES INSCRITOS EM
EDITAIS DE REMOÇÃO COM MONITORAMENTO DE VAGAS
REMANESCENTES PARA SECRETARIA MUNICIPAL DE
EDUCAÇÃO E CULTURA DA CIDADE DE GUARAPUAVA -
PARANÁ**

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

GUARAPUAVA
2022

EDUARDA LARA WALDEMAR

**CLASSIFICAÇÃO AUTOMÁTICA DE SERVIDORES INSCRITOS EM
EDITAIS DE REMOÇÃO COM MONITORAMENTO DE VAGAS
REMANESCENTES PARA SECRETARIA MUNICIPAL DE
EDUCAÇÃO E CULTURA DA CIDADE DE GUARAPUAVA -
PARANÁ**

Monografia de Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Diego marczal
Universidade Tecnológica Federal do Paraná
- Campus Guarapuava

Coorientador: Prof. Dr. Andres Jessé Porfirio
Universidade Tecnológica Federal do Paraná
- Campus Guarapuava

GUARAPUAVA
2022



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

EDUARDA LARA WALDEMAR

**CLASSIFICAÇÃO AUTOMÁTICA DE SERVIDORES INSCRITOS EM EDITAIS DE REMOÇÃO
DA SECRETARIA MUNICIPAL DE EDUCAÇÃO E CULTURA DA CIDADE DE GUARAPUAVA -
PARANÁ**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título
de Tecnólogo em Sistemas para Internet do Curso
de Tecnologia em Sistemas para Internet da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data da aprovação: 15/dezembro/2022

Prof. Diego Marczal
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Renata Luiza Stange
Mestre
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Luciano Ogiboski
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Dênis Lucas Silva
Mestre
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

GUARAPUAVA
2022

Dedico esta monografia a minha vó, por todo o amor que compartilhou comigo em vida.

Agradeço a minha família por todo apoio desde o início, respeitando sempre todas as escolhas feitas em minha vida. Agradeço especialmente a minha vó (in memoriam) que sempre acreditou e me apoiou em tudo que foi preciso, lembrando sempre da importância dos estudos e da educação nas nossas vidas mesmo sem ter tido a mesma oportunidade. À minha mãe (in memoriam) que não teve a chance de me acompanhar em grande parte da minha vida mas sempre esteve presente em minhas orações.

Aos meus amigos pela amizade e todo o apoio emocional, principalmente a Amanda Karoline que além de colega se tornou uma das melhores amigas da minha vida toda. Ao meu namorado Kaio que esteve comigo em todos os momentos dessa reta final, sempre me incentivando e cuidando de mim, presente nos momentos felizes e também nos difíceis.

Agradecimentos ao meu orientador por todos os ensinamentos, por confiar em mim e me auxiliar em tudo que foi necessário. Ao meu coorientador por toda ajuda. Aos meus professores por todo o conhecimento compartilhado durante os anos. Agradeço também aos meus colegas de curso e de trabalho, por toda a troca de conhecimento entre nós todos os dias.

Eu sou a continuação de um sonho, da minha mãe, do meu pai de todos que vieram antes de mim, eu sou a continuação de um sonho, da minha vó, do meu vô, quem sangrou pra gente poder sorrir.”(BK, 2022).

RESUMO

WALDEMAR, Eduarda Lara. Classificação automática de servidores inscritos em editais de remoção com monitoramento de vagas remanescentes para Secretaria Municipal de Educação e Cultura da Cidade de Guarapuava - Paraná. 2022. 43 f. Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2022.

As publicações de editais da Secretária de Educação da Prefeitura Municipal de Guarapuava referentes a remoção de servidores, acontece algumas vezes ao ano e envolve uma média de 2500 (dois mil e quinhentos) funcionários. Estes editais, publicam as vagas nas unidades de trabalho disponíveis, onde o servidor pode se inscrever de acordo com seu interesse e cargo. Para cada vaga individual podem ocorrer várias inscrições, então se faz necessário classificá-las em ordem de prioridade de acordo com critérios pré-estabelecidos pela secretaria, como, por exemplo, o maior tempo de serviço, titulação, formação do servidor e tempo de serviço na unidade. Atualmente, essa classificação é feita manualmente, tornando o processo lento, pois os encarregados devem verificar os dados de cada candidato, somar seus pontos e usá-los para categorizar os requerentes. Pensando em uma forma de trazer uma melhoria para o processo e buscando automatizá-lo, o sistema verifica os critérios de cada funcionário, soma seus pontos e fornece, assim, uma ordenação dos inscritos no edital. Durante a remoção, há alteração na quantidade de vagas sendo necessário o monitoramento e atualização de vagas disponíveis, é possível também o controle das vagas que irão ser liberadas quando um funcionário assumir outra lotação. Com todos os inscritos já ordenados, o administrador poderá visualizar facilmente quem terá prioridade para assumir a vaga e os inscritos terão conhecimento das novas vagas disponíveis atualizadas.

Palavras-chave: Gerenciamento. Remoção. Ordenação. Sistemas. Editais.

ABSTRACT

WALDEMAR, Eduarda Lara. Automatic classification of servers registered in removal notices with monitoring of remaining vacancies for the Municipal Department of Education and Culture of the City of Guarapuava - Paraná. 2022. 43 f. Trabalho de Conclusão de Curso de graduação – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2022.

The publication of servants removal notices by the Secretary of Education of the Guarapuava City occurs several times a year and involves an average of 2500 (two thousand five hundred) employees. These notices, publish vacancies in the available work units, where the server can enroll according to your interest and position. For each individual vacancy, several applications may occur, so if it is necessary to classify them in order of priority according to criteria pre-established by the secretariat, such as the longest length of service, title, employee training and length of service in the unit. Currently, this classification is done manually, which makes the process slow, as incumbents must verify the data of each candidate, add up their points and use them to categorize applicants. Thinking of a way to bring an improvement to the process and seeking to automate it, the system verifies the criteria of each employee, adds up their points and thus provides an ordering of those enrolled in the edict. During the removal, there is a change in the number of vacancies, requiring monitoring and updating of vacancies available, it is also possible to control the vacancies that will be released when an employee assumes another assignment. With all subscribers already sorted, the administrator will be able to easily visualize who will have priority to assume the vacancy and the subscribers will be aware of new vacancies updated.

Keywords: Management. Removal. Classification. Systems. Edicts.

LISTA DE FIGURAS

Figura 1 – Tela de Listagem de Editais da Área Administrativa	4
Figura 2 – Tela de Cadastro de um Edital da área Administrativa	5
Figura 3 – Listagem de Editais da Área Pública	6
Figura 4 – Visualização dos Editais na Área do Servidor	7
Figura 5 – Formulário de Inscrição em Edital de Remoção na Área do Servidor	8
Figura 6 – Fragmento do Banco de Dados do Sistema da SEMEC	9
Figura 7 – Exemplo do Uso do Trello com o Método Kanban	15
Figura 8 – Exemplo do Gráfico Burdown	16
Figura 9 – Exemplo do Gráfico Burnup	16
Figura 10 – Fluxo de Desenvolvimento Proposto	18
Figura 11 – Fluxo de ações do sistema	21
Figura 12 – Fluxo de desenvolvimento	24
Figura 13 – Protótipo da Tela para Seleção de Unidade	25
Figura 14 – Protótipo de Tela para Visualização de Inscritos	26
Figura 15 – Protótipo de Tela para Visualização da Pontuação	27
Figura 16 – Cadastro de Vagas para um Edital	28
Figura 17 – Lista de Unidades Cadastradas no Edital.	28
Figura 18 – Relacionamento Entre os Dados Complementares do Servidor e sua Formação.	31
Figura 19 – Tela Cadastro de Nova Formação	32
Figura 20 – Tela Cadastro de Nova Formação para um Servidor	33
Figura 21 – Tabela Classificação	35
Figura 22 – Lista de unidades cadastradas no edital.	37
Figura 23 – Tela Classificações	37
Figura 24 – Tela Classificações após assumir uma vaga	39
Figura 25 – Banco de Dados	40
Figura 26 – Tabela de relacionamento entre edital e unidades	41

LISTA DE ABREVIATURAS E SIGLAS

SEMEC	Secretaria de Educação e Cultura da Prefeitura Municipal de Guarapuava
TSI	Tecnologia em Sistemas para Internet
UTFPR-GP	Universidade Tecnológica Federal do Paraná - Campus Guarapuava
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
MYSQL	My Structured Query Language
PHPUnit	Hypertext Preprocessor Unit
SQL	Structured Query Language
PHPStan	Hypertext Preprocessor Stan
PHPMD	Hypertext Preprocessor Mess Detector
PHPCS	Hypertext Preprocessor Coding Standards
CI	Continuous Integration
CD	Continuous Delivery

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
2 – SISTEMA DE GESTÃO DA VIDA FUNCIONAL DE SERVIDORES DA SEMEC	3
2.1 PRINCIPAIS FUNCIONALIDADES DO SISTEMA	3
2.1.1 Área Administrativa	3
2.1.2 Área Pública	5
2.1.3 Área Do Servidor	6
2.1.4 Banco de Dados	8
2.2 TECNOLOGIA UTILIZADAS	9
3 – METODOLOGIA DE DESENVOLVIMENTO	13
3.1 SCRUM	13
3.2 APLICAÇÃO DO SCRUM NO PRESENTE PROJETO	17
3.3 FLUXO DE DESENVOLVIMENTO	17
4 – DESENVOLVIMENTO	19
4.1 DOMÍNIO DO PROBLEMA	19
4.1.1 O Processo Manual de Classificação	19
4.1.2 Solução Proposta	20
4.2 PROCESSO DE DESENVOLVIMENTO	22
4.2.1 O uso do SCRUM	22
4.2.2 User Stories	22
4.2.3 Fluxo de Desenvolvimento Aplicado	23
4.3 PROTÓTIPOS DE TELA	24
4.4 VAGAS DISPONÍVEIS EM UM EDITAL	27
4.5 SOLUÇÃO IMPLEMENTADA	29
4.5.1 Pontuação do servidor	29
4.5.2 Algoritmo para a Classificação dos Servidores	34
4.5.3 Apresentação da Classificação dos Servidores	36
4.5.4 Admissão do Servidor a uma Vaga	38
4.5.5 Monitoramento e Disponibilização das Vagas Remanescentes	40
4.5.6 Reclassificação dos Servidores para as Novas Vagas	41

5 – CONSIDERAÇÕES FINAIS	42
Referências	43

1 INTRODUÇÃO

A Secretaria de Educação e Cultura da Prefeitura Municipal de Guarapuava (SEMEC) é responsável por 43 escolas em todo município que consiste por 2500 a 3000 servidores atuando, informações mencionadas pela responsável Elaine. Uma de suas responsabilidades é conduzir a movimentação de servidores entre as escolas, por exemplo, a permuta de servidores.

Os locais de trabalho dos servidores da secretaria são nomeados de unidades durante esses processos e as movimentações são realizadas entre elas de acordo com critérios estabelecidos, sendo a SEMEC responsável por garantir a homogeneidade do processo. Isto demanda tempo e esforço dos servidores, com o agravante que a execução do processo é realizada de forma manual.

Buscando dar maior fluidez e dinamismo a este processo de movimentação de servidores, a SEMEC, em 2019, realizou uma parceria com o curso de Tecnologia em Sistemas para Internet (TSI) da Universidade Tecnologia Federal do Paraná, câmpus Guarapuava (UTFPR-GP). A partir disso, deu-se início ao projeto de extensão Sistema para Gestão de Servidores da Secretaria Municipal de Educação de Guarapuava, cujo objetivo é automatizar todo o processo de movimentação dos servidores da SEMEC com o projeto do [Sistema. . . \(2022\)](#). Neste projeto, foi desenvolvido um sistema web que permite essa automatização. Os módulos responsáveis pelos cadastros, publicação de editais e inscrições foram desenvolvidos durante o projeto, mas ainda não estão sendo usados pela secretaria.

O desenvolvimento foi realizado do ano 2019 a 2021, com participação de três alunas da universidade sob orientação de 2 professores do curso de TSI. Como as tecnologias utilizadas no sistema já estavam pré-definidas, as alunas participantes não tinham conhecimento sobre elas e com isso houve uma necessidade de estudo das tecnologias. Inicialmente, foi realizada uma reunião com a responsável pela secretaria da época para o levantamento de requisitos necessários para o início do projeto, onde foram informadas todas as funcionalidades que deveriam existir. Os módulos inicialmente desenvolvidos foram os de cadastro de usuários contendo informações sobre ele e seu histórico de trabalho na secretaria, divididos em dois tipos: administradores e servidores. O cadastro relacionado aos locais de trabalho dos servidores nomeados como unidades, contém sua categoria para organizá-las, também a publicação de editais, possibilitando a inscrição realizada por servidores nos mesmos.

O sistema gerencia a troca de servidores entre os diversos locais de trabalho, para isso, permite a divulgação de vagas, abertura de editais, realização de inscrições pelos interessados, além do gerenciamento interno para determinar quais inscritos terão prioridade para as vagas disponíveis. Uma das dificuldades do processo, da forma como é feito atualmente, é classificar os inscritos de acordo com os critérios, como tempo de serviço no município, titulação e tempo de serviço na escola. Esta etapa do processo, no entanto, é realizada de forma manual, em que o servidor responsável precisa analisar todas as informações dos inscritos e ordená-los conforme

os critérios estabelecidos para o edital. Isso torna a análise dos documentos lenta, além de dificultar a classificação dos inscritos e definir quem poderá assumir as vagas disponíveis.

Outra necessidade é o monitoramento das vagas que ficarão disponíveis durante o processo, deixando-as disponíveis para outros servidores. Quando um servidor assume uma vaga, abre-se sua antiga vaga em seu local de trabalho definido como unidade, podendo ser assumida por outro servidor que se inscreveu no mesmo edital, considerando que até o momento do desenvolvimento do trabalho este poderá escolher até 3 unidades de interesse.

Portanto, foi desenvolvida a automatização do processo de classificação dos inscritos em um edital, assim como o gerenciamento da viabilidade de servidores aptos a assumirem novas vagas que surgiram em um mesmo edital. Os principais desafios do projeto dizem respeito a compreensão de todo o processo, transposição dele para o sistema, a dificuldade de comunicação entre a proponente e os responsáveis na prefeitura, por último, o trabalho com um problema real.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo deste trabalho é automatizar a classificação dos servidores inscritos em um edital, seja para as vagas previamente disponíveis ou para aquelas que podem surgir durante o processo de admissão dos servidores selecionados.

1.1.2 Objetivos Específicos

- Definir e implementar os cálculos para pontuação do servidor em cada um dos critérios pré-estabelecidos pela SEMEC;
- Implementação do algoritmo para a classificação dos servidores;
- Implementação da apresentação da classificação dos servidores inscritos em um edital;
- Permitir que o administrador admita um servidor a uma determinada vaga;
- Monitorar a disponibilidade de vagas que podem surgir durante o processo de admissão dos servidores selecionados;
- Registrar e distinguir as novas vagas das vagas previamente inscritas no edital;
- Reclassificação dos servidores inscritos em um edital para as vagas que surgem durante o processo.

2 SISTEMA DE GESTÃO DA VIDA FUNCIONAL DE SERVIDORES DA SEMEC

Ao início desse trabalho, já haviam algumas partes do sistema desenvolvidas, nessa seção serão mencionadas apenas as que fazem parte diretamente e afetam nosso módulo.

O administrador é responsável pelo cadastro de novos servidores sendo eles administradores ou não, realiza também a publicação de novos editais para receber as inscrições dos servidores interessados em novas lotações. Na área do servidor está disponível a visualização dos editais abertos e fechados, onde ele pode realizar sua inscrição nos editais que estiverem em aberto escolhendo qual matrícula será usada para esse processo e suas unidades de interesse para remoção. Já os editais que se encontram fechados disponibilizam apenas a consulta de suas informações.

2.1 PRINCIPAIS FUNCIONALIDADES DO SISTEMA

2.1.1 Área Administrativa

A área administrativa atualmente conta com algumas áreas já desenvolvidas, em seu perfil o administrador pode atualizar e alterar seu perfil com confirmação de senha. O cadastro, atualização e remoção de outros administradores também se encontram disponíveis. Os servidores são carregados semi-automáticos de uma base direta da prefeitura e com isso temos a listagem de servidores. As unidades podem ser cadastradas, alteradas e excluídas, também é possível cadastrar as categorias das unidades e relacioná-las.

O sistema permite a visualização das movimentações dos servidores, onde mostra quais unidades o servidor já atuou. Se tem também o contrato do servidor cadastrado e que a partir dele se realiza cadastro complementar, onde constam suas informações como a sua formação e carga horária de trabalho. O contrato, também conhecido como matrícula, será usado para as inscrições em editais publicados pelo administrador.

O administrador pode cadastrar novos editais que serão publicados como mostra a Figura 1, neles constam informações como seu título, descrição e sua data de início e fim, onde serão disponibilizados em uma listagem como consta na Figura 2, também é possível sua atualização e exclusão. Para os editais em abertos, os servidores podem realizar as inscrições, os administradores conseguem ter a visualização das mesmas.

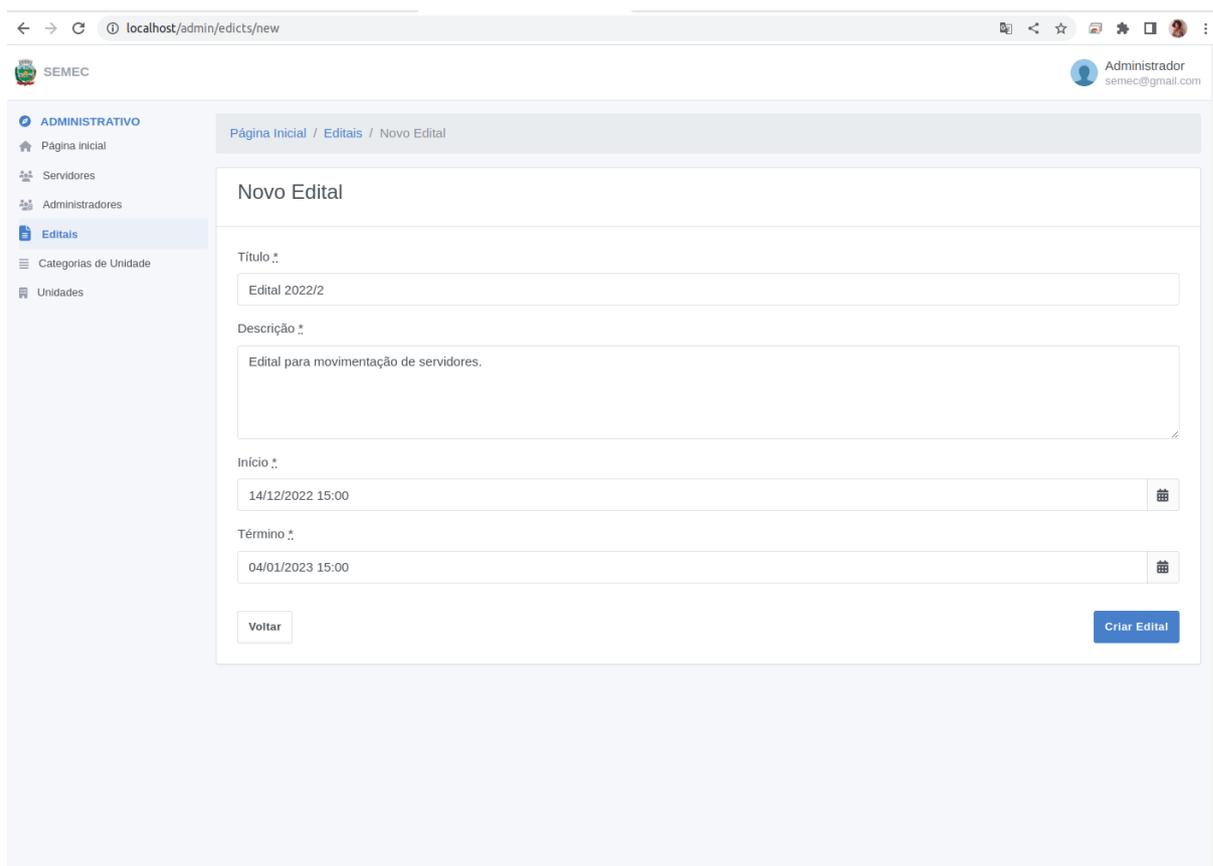
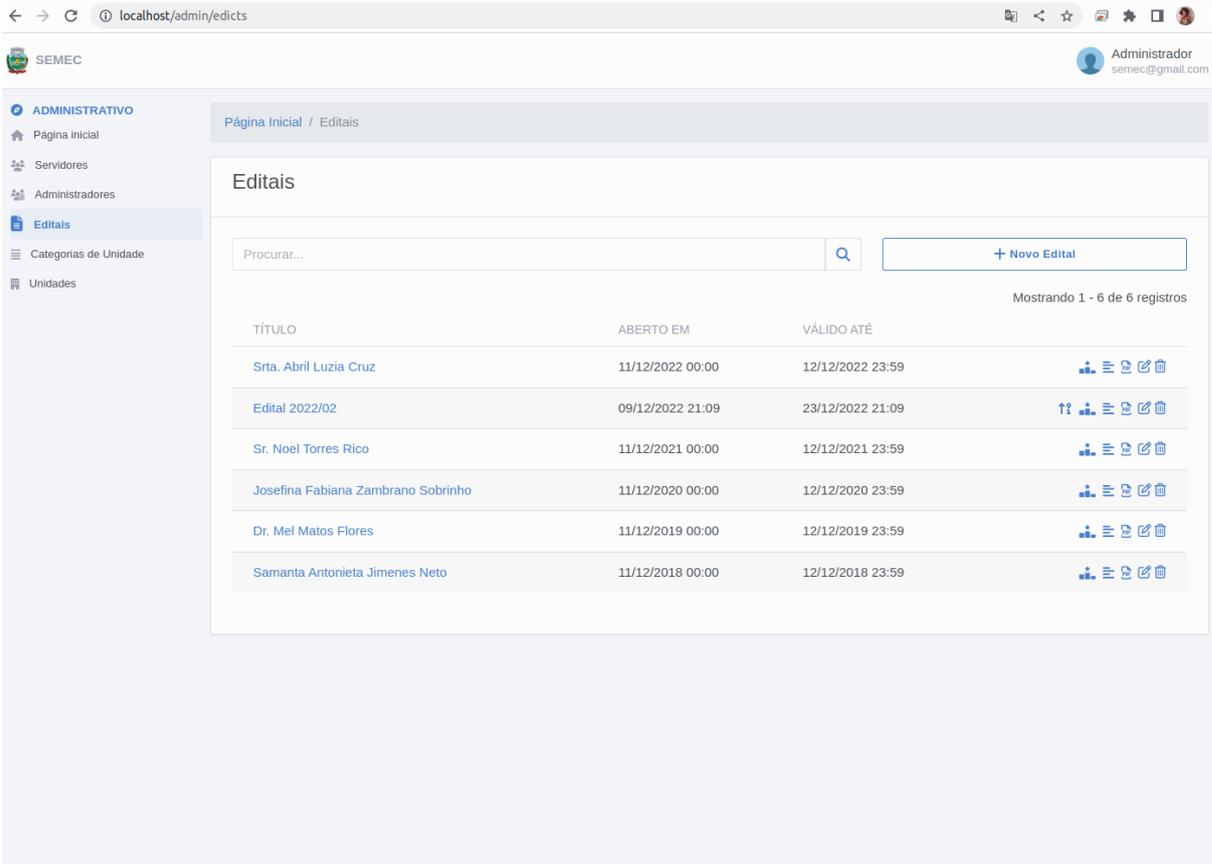


Figura 1 – Tela de Listagem de Editais da Área Administrativa

Fonte: Sistema de Gestão da Vida Funcional dos Servidores da SEMEC



The screenshot shows a web browser window with the URL localhost/admin/editais. The page title is 'Editais' and it displays a table of edicts. The table has three main columns: TÍTULO, ABERTO EM, and VÁLIDO ATÉ. There are six records listed. Each record has a set of icons for actions like edit, delete, and print. A search bar is located at the top left of the table area, and a '+ Novo Edital' button is at the top right. The text 'Mostrando 1 - 6 de 6 registros' is displayed below the table.

TÍTULO	ABERTO EM	VÁLIDO ATÉ	
Srta. Abril Luzia Cruz	11/12/2022 00:00	12/12/2022 23:59	  
Edital 2022/02	09/12/2022 21:09	23/12/2022 21:09	  
Sr. Noel Torres Rico	11/12/2021 00:00	12/12/2021 23:59	  
Josefina Fabiana Zambrano Sobrinho	11/12/2020 00:00	12/12/2020 23:59	  
Dr. Mel Matos Flores	11/12/2019 00:00	12/12/2019 23:59	  
Samanta Antonieta Jimenes Neto	11/12/2018 00:00	12/12/2018 23:59	  

Figura 2 – Tela de Cadastro de um Edital da área Administrativa

Fonte: Sistema de Gestão da Vida Funcional dos Servidores da SEMEC

2.1.2 Área Pública

Na área pública do sistema, o usuário sem acesso ao sistema pode visualizar informações gerais sobre a SEMEC, a disponibilização dos editais, sejam eles abertos ou encerrados com o seu nome e sua data de duração, categorizados pelo ano em que ocorreu, isso está disponível na Figura 3. Por ela também se tem acesso a área de acesso do servidor e administrador.

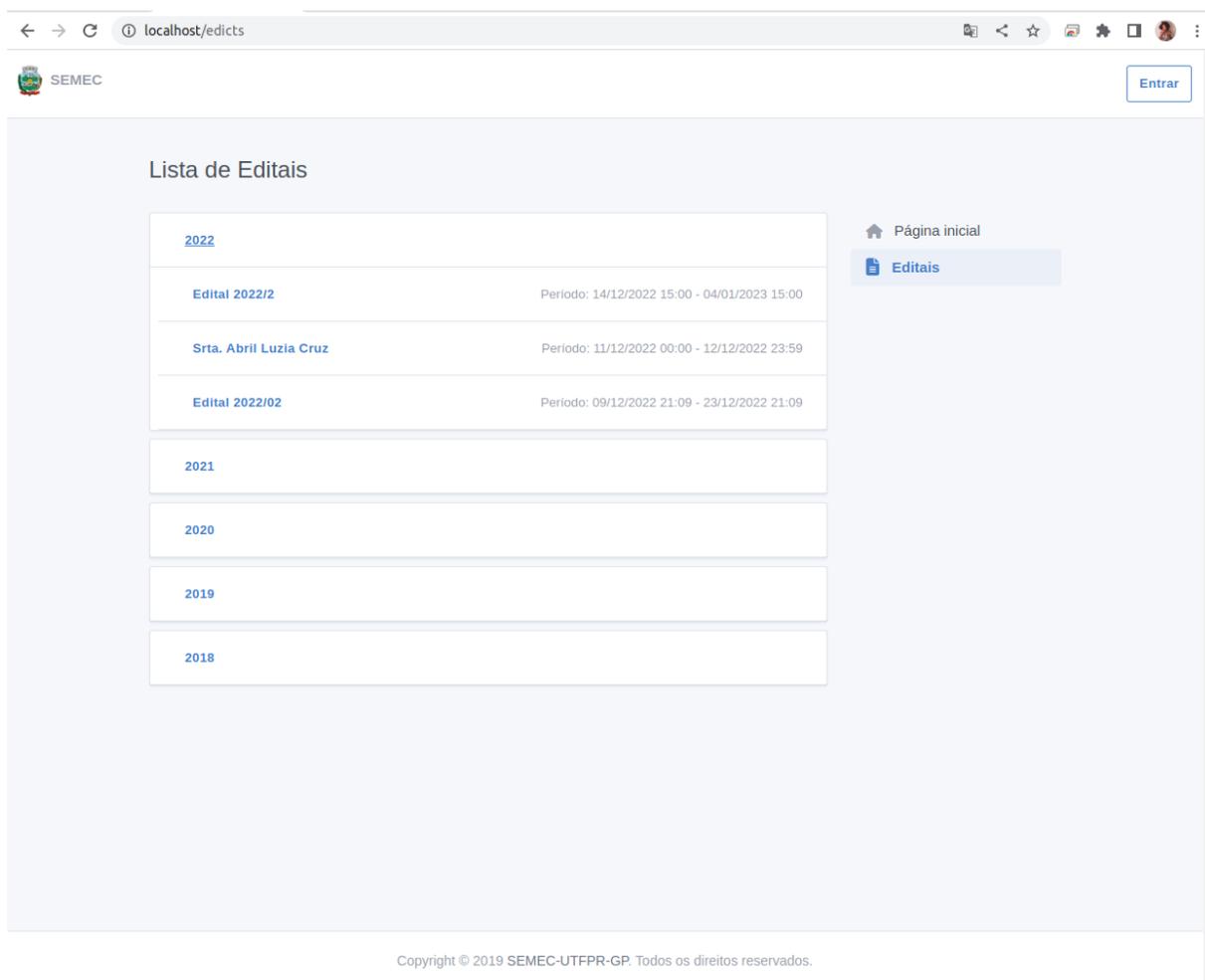
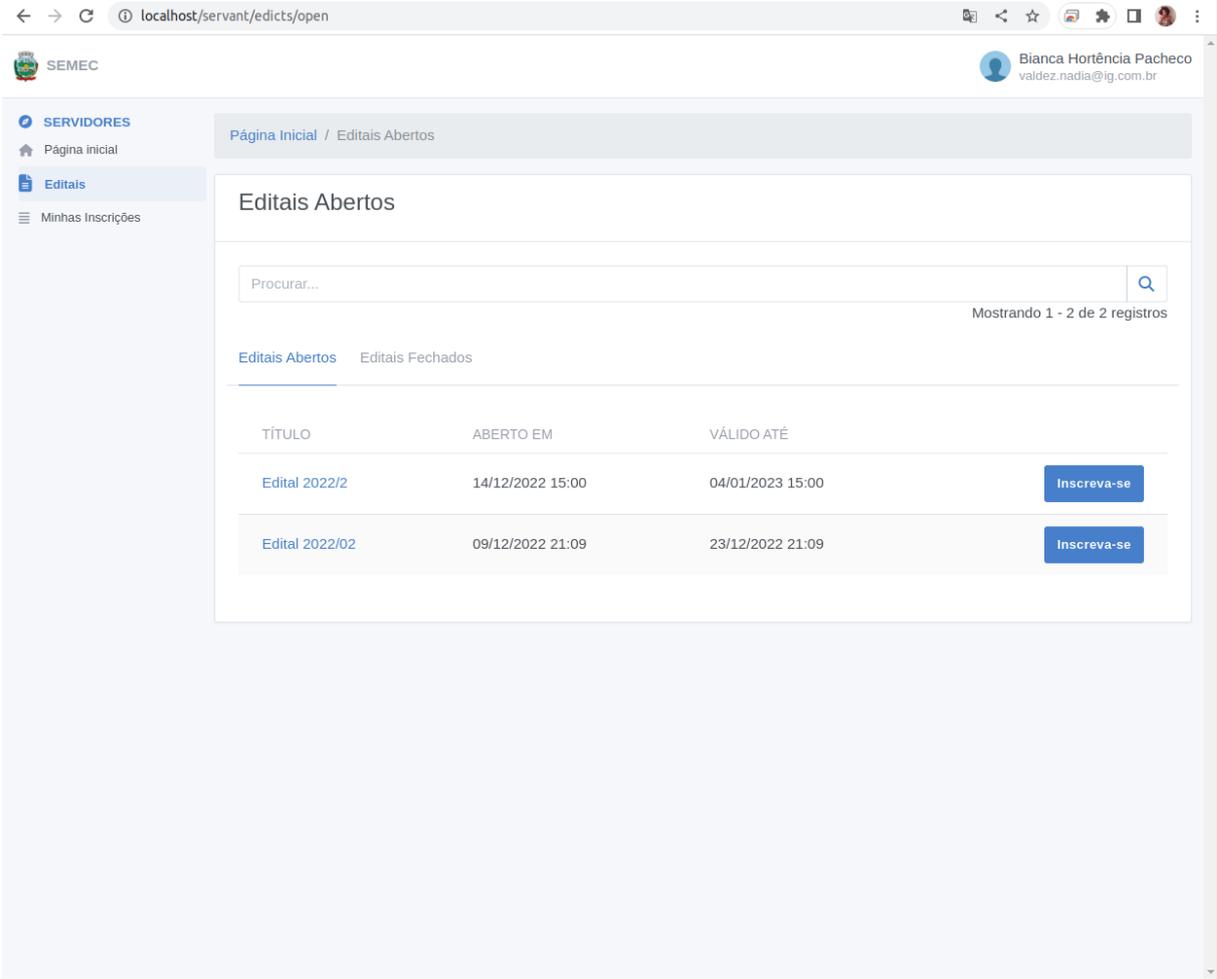


Figura 3 – Listagem de Editais da Área Pública

Fonte: Sistema de Gestão da Vida Funcional dos Servidores da SEMEC

2.1.3 Área Do Servidor

Na área do servidor já desenvolvida está disponível para o usuário sua atualização de foto de perfil e sua alteração de senha, mediante a confirmação de senha. Está disponível também a visualização dos editais abertos e encerrados com suas informações como seu título, descrição e data de duração, a apresentação dos editais ao servidor é mostrada na Figura 4.



The screenshot shows a web browser window with the URL `localhost/servant/edicts/open`. The page title is "Editais Abertos" and the breadcrumb is "Página Inicial / Editais Abertos". The user is logged in as Bianca Hortência Pacheco (valdez.nadia@ig.com.br). The sidebar contains "SERVIDORES", "Página inicial", "Editais", and "Minhas Inscrições". The main content area has a search bar labeled "Procurar..." and a "Mostrando 1 - 2 de 2 registros" indicator. Below the search bar are tabs for "Editais Abertos" and "Editais Fechados". A table lists two open edicts:

TÍTULO	ABERTO EM	VÁLIDO ATÉ	
Edital 2022/2	14/12/2022 15:00	04/01/2023 15:00	Inscreva-se
Edital 2022/02	09/12/2022 21:09	23/12/2022 21:09	Inscreva-se

Figura 4 – Visualização dos Editais na Área do Servidor

Fonte: Sistema de Gestão da Vida Funcional dos Servidores da SEMEC

Os editais em aberto, permitem que o servidor se inscreva para a remoção no formulário presente da Figura 5 onde ele irá selecionar sua matrícula, a qual é o contrato que ele deseja usar para esse edital, lembrando que cada contrato terá seu próprio cadastro complementar, seu tipo de remoção e as três unidades que ele tem interesse para assumir uma vaga e o motivo de sua remoção.

The image shows a web browser window displaying a form for registration in a public notice. The browser's address bar shows the URL: localhost/servant/edicts/233/inscriptions/new. The page header includes the SEMEC logo and the user's name, Bianca Hortência Pacheco, with the email valdez.nadia@ig.com.br. The breadcrumb trail is: Página Inicial / Editais Abertos / Edital #233 / Nova Inscrição. The form title is 'Inscreva-se no Edital Edital 2022/2'. The form fields are: 'Matricula' with a dropdown menu showing '36'; 'Tipo de remoção' with a dropdown menu showing 'Pedro Hernani Faria'; 'Unidade atual' with a text input field containing 'Sr. Ricardo Romero Neto'; 'Unidade de interesse' with a multi-select dropdown menu showing 'Dr. Luciana Vasques Neto', 'Clara Mia Serna Neto', and 'Carolina Rodrigues Toledo Neto'; and 'Motivo' with a text area containing 'Teste'. At the bottom of the form, there are two buttons: 'Voltar' and 'Enviar'.

Figura 5 – Formulário de Inscrição em Edital de Remoção na Área do Servidor

Fonte: Sistema de Gestão da Vida Funcional dos Servidores da SEMEC

2.1.4 Banco de Dados

O banco de dados do sistema antes do trabalho contava com 16 tabelas, serão citadas apenas as que estarão envolvidas diretamente com o trabalho desenvolvido, ilustradas na Figura 6. Tabela de servidores, *servants*, armazena alguns dados do servidor, seu contrato é encontrado na tabela *contracts*, onde um usuário pode ter mais de um contrato e número de registro nomeado como matrícula, mas a tabela *servant complementary datas* vinculado diretamente ao contrato, contém o cadastro complementar de cada usuário levando sua formação e carga horária.

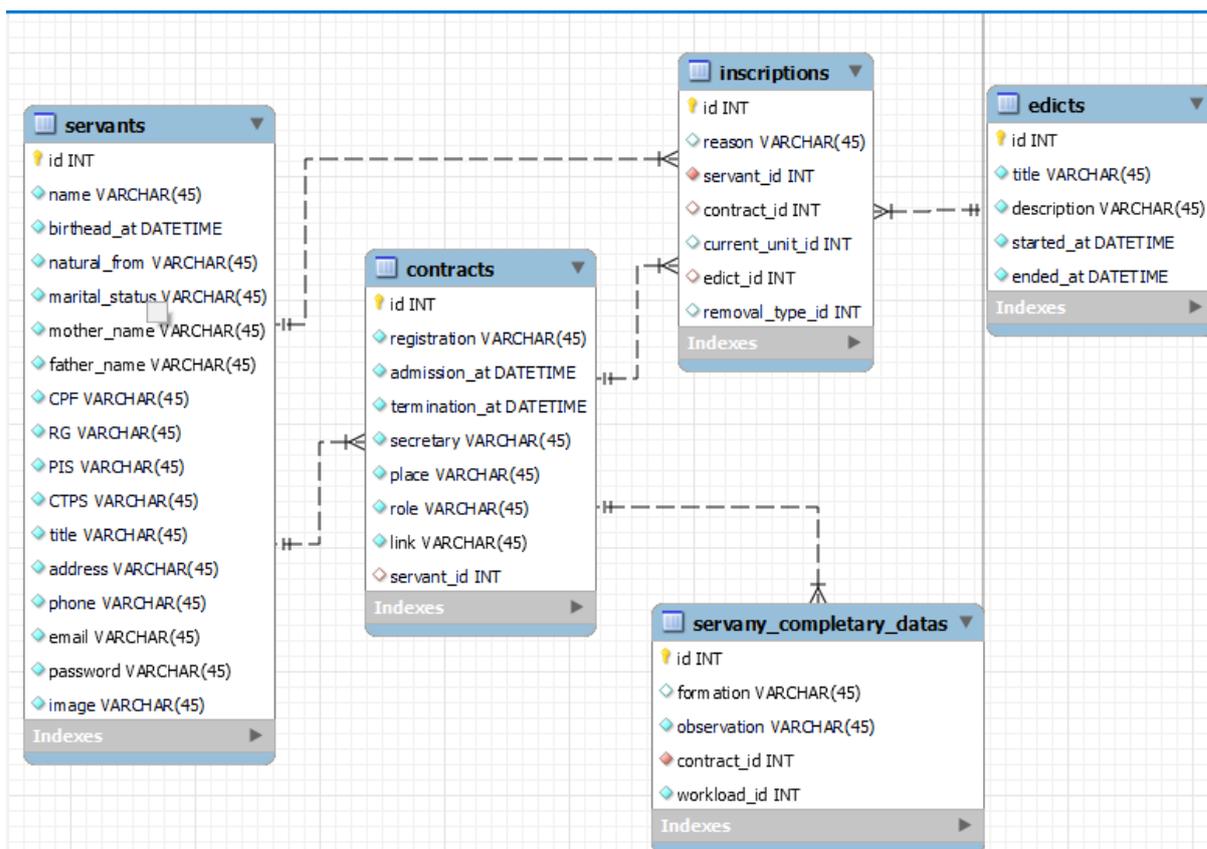


Figura 6 – Fragmento do Banco de Dados do Sistema da SEMEC

Fonte: Autor

A tabela de inscrições nomeada com *inscriptions* é responsável por armazenar as inscrições realizadas pelos servidores nos editais abertos, ela mantém uma relação direta com a tabela de servidores, contratos, unidades, editais e tipo de remoção, armazena também o motivo da inscrição.

Os editais são armazenados na *edicts*, nas inscrições é armazenado o edital o qual foi feito a inscrição, assim é identificável o edital a qual ela pertence. A *units*, tabela onde são armazenadas as unidades onde contém seu nome e endereço também terão papel importante no desenvolvimento do projeto, por elas se tem a informação dos locais disponíveis para movimentações dos servidores.

2.2 TECNOLOGIA UTILIZADAS

Neste capítulo, apresentam-se as tecnologias utilizadas para o desenvolvimento do projeto. Opta-se por tecnologias já utilizadas em outros sistemas da prefeitura, tais quais foram escolhidas por eles. Outras, por serem ferramentas de código aberto, geram uma grande comunidade onde pode-se tirar dúvidas, além de uma documentação mais completa.

Para melhorar o desempenho da construção do front-end (aparência do site) foi

utilizado o framework Bootstrap¹, com ele o desenvolvimento se torna mais dinâmico, pois possui componentes e plugins prontos e estilizados com o uso de HTML, CSS e JavaScript, além disso, a ferramenta é de código aberto, tornando ainda mais fácil o acesso. O Tabler.io² é um template voltado para áreas administrativas desenvolvido com o Bootstrap, com ele, páginas e elementos mais comuns nessa categoria de sistema já estão prontos e pode-se aproveitá-los realizando poucas alterações.

Como framework para desenvolvimento de aplicações web foi utilizado o Laravel³, escrito da linguagem de programação PHP, esse reúne várias bibliotecas com métodos e configurações predefinidas para o desenvolvimento de aplicações web de maneira rápida e eficiente. O PHP⁴ é uma das linguagens mais populares para o desenvolvimento web, sendo escolhida pela prefeitura, devido à familiaridade de seus funcionários de Tecnologia da Informação (TI).

Para o banco de dados utilizou o MYSQL⁵, um gerenciador de banco de dados que usa a linguagem SQL, ele nos traz diversas vantagens, mas as principais são o fato de ser open source, possuir uma grande comunidade, facilitando as buscas por informações, além de ser muito confiável.

Com a necessidade de executar o ambiente de desenvolvimento em mais de uma máquina de uma forma que não precisasse alterar e realizar instalações localmente utilizamos o Docker⁶. A plataforma utiliza containers que funcionam como ambientes isolados para criar, alterar e executá-los conforme as necessidades do projeto. Para auxiliar na orquestração dos containers e executá-los usa-se o Docker Compose⁷ onde define suas necessidades em um arquivo.

Para testes automatizados, foi utilizado o framework PHPUnit⁸, com ele, ao invés de testar pequenos trechos da aplicação manualmente, escrevesse um código conforme a necessidade da funcionalidade a ser testada e a executamos. Com o uso de “asserts” consegue-se saber se ele está retornando o valor esperado. O uso dos testes automatizados permitem que a cada nova funcionalidade adicionada, sejam feitos testes de regressão, garantindo que nenhuma outra funcionalidade foi quebrada com a adição de novos códigos.

Há, também, a forma automatizada para realizar testes que simulam o comportamento de um usuário no navegador: preenchendo formulários, realizando requisições e outras ações de nossa escolha, interagindo com a página pelo navegador. Garantindo, assim, que o sistema funciona como deveria, ou conforme os testes escritos. Para isso, é utilizado o pacote Dusk, que permite de uma forma facilitada, simular a interação do cliente com a aplicação.

¹Site oficial: <<https://getbootstrap.com>>

²Site oficial: <<https://tabler.io>>

³Site oficial: <<https://laravel.com/>>

⁴Site oficial: <<https://www.php.net/>>

⁵Site oficial: <<https://www.mysql.com>>

⁶Site oficial: <<https://www.docker.com>>

⁷Site oficial: <<https://docs.docker.com/compose>>

⁸Site oficial: <<https://phpunit.de>>

Para manter um padrão de qualidade do código são utilizados verificadores de código como o PHPStan⁹, com a execução, ele irá “escanear” o código-fonte e retornar possíveis erros e incongruências baseados em padrões pré-definidos na comunidade. O ESLint (ESLINT, 2023) tem a mesma funcionalidade, mas é voltado para a linguagem Javascript. Outro verificador é conhecido como PHP-CS¹⁰, tem o objetivo de aplicar padrões na codificação e estilos de código nos arquivos PHP, detecta os trechos fora desse padrão e os corrige. Podendo-se aplicar regras já existentes na comunidade ou definir seu próprio estilo de configuração através de um arquivo de configuração. Com o mesmo intuito das outras ferramentas, também é utilizado o PHPMD¹¹ (PHP Med Detector) é uma ferramenta de análise, que permite encontrar problemas e analisar a qualidade do código, localizando bugs, métodos, propriedades e parâmetros que não estão sendo utilizados, melhorando a qualidade de seu código.

Para versionamento de código é utilizado o Git¹², com ele é possível ter um histórico das alterações no código e controlar suas versões e mudanças, isso facilita o trabalho em equipe podendo ter várias pessoas trabalhando em alterações diferentes. O código está hospedado na plataforma Github¹³, ele permite que todos os programadores envolvidos no projeto possam contribuir no projeto de modo organizado em ramificações que só serão adicionadas na versão principal com aprovação de outros participantes.

Algumas ações são automatizadas com o Github Actions¹⁴, criando rotinas necessárias para a execução do código dispensando o uso de ferramentas externas para iniciar a CI/CD (Continuous Integration/ Integração Contínua e Continuous Delivery/Entrega Contínua) CI/CD (2019), realizando a build e executando os testes e verificadores de código, executadas conforme o que foi definido nas ações que ocorrem no repositório, por exemplo, quando executa um pull request (ação executada quando contribuidor pede ao administrador do repositório para revisar seu código antes de juntar ao projeto) pode ser realizado automaticamente um comando com teste para as funcionalidades novas.

Com o uso de todas essas tecnologias e testes realizados por usuários evitamos ao máximo erros em sua versão final, poupando tempo com algumas partes do desenvolvimento sem precisar realizar tarefas do zero, também previne problemas futuros como “bugs” que podem ser evitados se existir testes corretos. A descrição resumida de todas essas tecnologias se encontram no quadro 1.

As tecnologias utilizadas durante o projeto já estavam sendo usadas desde o início do projeto, escolhidas pela prefeitura foram o Mysql e o PHP, por já estarem presentes em outros sistemas e eles terem um conhecimento sobre elas e outras por serem de código-aberto. Não foram necessários novas ferramentas, apenas um novo estudo sobre elas.

⁹Site oficial: <<https://phpstan.org>>

¹⁰PHP-CS

¹¹Site oficial: <<https://phpmd.org>>

¹²Site oficial: <<https://git-scm.com>>

¹³Site oficial: <<https://github.com>>

¹⁴Site oficial: <<https://docs.github.com/pt/actions>>

Tecnologia	Descrição	Justificativa de uso
Bootstrap	Framework para interface para aplicações com o uso de HTML, CSS e Javascript	Torna o desenvolvimento mais dinâmico com seus componentes e plugins estilizados.
Tabler.io	Template administrativo desenvolvido com Bootstrap.	Reúne bibliotecas com interfaces pré-definidas de áreas administrativas as quais podemos aproveitar
Laravel	Framework para desenvolvimento web com várias bibliotecas para otimizar os recursos do php	Auxilia o desenvolvimento, tornando mais rápido, organizado e eficiente.
PHP	É uma linguagem de programação para desenvolvimento web.	Foi escolhida pela prefeitura por causa da familiaridade dos funcionários da tecnologia da informação.
MYSQL	Gerenciador de banco de dados que utiliza a linguagem SQL.	Se trata de algo open source, com uma grande comunidade facilitando sua aplicação.
Docker	É uma forma de virtualização da aplicação, conhecido por containers, onde criamos e alteramos de acordo com a nossa necessidade de dependências.	Facilita a execução do sistema quando necessário em qualquer máquina.
Docker Compose	Orquestrador de containers docker, executando as ações dentro deles.	Auxiliar no orquestramento dos containers.
PHPUnit	Framework utilizado para testes unitários da linguagem php.	Testar pequenos trechos de código durante o desenvolvimento.
Laravel Dusk	Automação das ações testando a API.	Teste de browser das funcionalidades.
PHPStan	Encontra erros no código sem precisar executá-lo.	Padronizar e evitar erros no código.
PHP-CS	Aplicação de padrões para códigos php.	Padronizar e evitar erros no código.
PHPMD	Procura vários problemas em potencial, como bugs, expressões complicadas e métodos parâmetros não utilizados.	Padronizar e evitar erros no código.
Github	Plataforma para hospedagem de código com controle de versões.	Armazenar a aplicação.
Github Actions	Criação de rotinas necessárias para a execução do código, iniciando a CI e CD	Realizar o build executando as ações necessárias.

Quadro 1 – Tecnologias utilizadas.

Fonte: Autor

3 METODOLOGIA DE DESENVOLVIMENTO

3.1 SCRUM

Para o desenvolvimento do projeto era utilizado a metodologia Scrum, que consiste em uma estrutura ágil para gerenciamento e planejamento de projetos de Software. Segundo [Sutherland \(2014\)](#), o Scrum é uma estrutura utilizada na criação e manutenção de sistemas complexos. Sendo composto por papéis (Product Owner, Scrum Master e Development Team), artefatos (Product Backlog, Sprint Backlog e Product Increment) e eventos (Sprint, Sprint Planning, Daily Scrum Meeting, Sprint Review e Sprint Retrospective).

Ele é composto por pequenos ciclos de atividades, chamados de sprints, tais quais são planejados com um tempo definido conforme a complexidade da atividade a ser realizada. Com isso, evitamos a correção do produto apenas quando estiver finalizado, podendo realizá-la a cada Sprint finalizada.

O Product Owner é responsável por definir e manter a integridade das novas funcionalidades do sistema e, também, garante a qualidade das entregas, sendo o único que pode aceitar as tarefas como concluídas. Normalmente, é o proprietário do produto ou o membro do time que tem mais conhecimento sobre o produto, responsável em escrever as histórias e priorizá-las em backlog - termo explicado posteriormente. O Scrum Master é responsável pela aplicação do Scrum. Seu papel é apresentar a teoria, práticas e regras ao time, ele irá facilitar os processos conforme as habilidades ao time e orientar o Product Owner para a criação do Product Backlog, também irá resolver conflitos que possam aparecer durante o processo.

Development Team, são os responsáveis no desenvolvimento do produto, eles irão inserir os itens do Backlog no software, esse time não será composto necessariamente apenas por programadores, mas por todas as pessoas ligadas a necessidade do produto. O Product Backlog é o conjunto das histórias que darão origem ao produto, ordenando-as por prioridades, o Product Owner é responsável por verificar e realizar a manutenção dessas histórias se tornando algo dinâmico. As Dailys Scrum são reuniões diárias para conseguir ter conhecimento do andamento, são citadas as atividades do time que foram e serão realizadas e se há algum impedimento para a realização de uma atividade que possa ser auxiliado pelo time.

Para uma melhor organização do tempo das tarefas são criadas as Sprints, as durações devem variar de 1 a 4 semanas, assim se tem maior controle de quanto tempo as atividades levarão para ser concluídas de acordo com sua complexidade, a Sprint planning é uma reunião realizada antes de um Sprint Backlog. Após o encerramento da reunião de cada Sprint, o Product Owner sugere as mudanças que serão feitas conforme a prioridade, essa fase é a Sprint Review onde a equipe mostra o que conseguiu realizar durante a Sprint, a reunião é aberta e a equipe decide que tarefas podem ser consideradas concluídas. Na Sprint Retrospective reúne-se a equipe e pensam no que deu certo e o que poderia ter sido melhorado para a próxima Sprint,

nessa fase olha-se o processo e realizar um feedback e procurar uma solução ou aprimoramento e realiza-se o Increment no produto.

Para definir as funcionalidades do sistema em um processo Scrum, são usadas histórias (Do inglês User Stories) que descrevem as funcionalidades conforme as necessidades do cliente, são como casos de uso, especificando o ator e objetivo das tarefas. As histórias precisam ser independentes entre elas, agregando valor ao usuário. Ao escrever essas histórias é importante ser usado a abordagem INVEST, onde há alguns critérios para manter uma boa qualidade na sua criação e a importância de "investir em boas User Stories (SABBAGH, 2022)". São esses:

- Independent (Independente): Stories devem ser independentes umas das outras para serem trabalhadas em qualquer ordem e sem carregarmos mais uma do que a outra;
- Negotiable (negociável): O resultado precisa ser negociável, ou seja, escrito de forma colaborativa entre cliente, time e Scrum master, atendendo as necessidades do produto da melhor forma;
- Valuable (Valiosa): As Stories devem ter um valor perceptível para o usuário final ou cliente, priorizando as regras e valores de negócio e não a das tecnologias;
- Estimable (Estimável): Toda Story deve ser possível estimar ou dimensionar para priorizar ou não conforme o seu tempo de desenvolvimento;
- Small (Pequena): Procurar sempre escrever Stories pequenas para conseguir estimar melhor seu tempo de desenvolvimento, é recomendável 3 a 4 dias de trabalho, incluindo todo o processo até o status pronto;
- Testable (Testável): Todo Story deve ser testado, seja ele funcional ou não funcional.

Smart são critérios para seguir para criação de metas como as Sprints, com esse formato garante que ao definir metas terá clareza sobre o que está sendo realizado, quando e como:

- Specific (específico) diz que ao definir metas, deve-se ser específico sobre o que deseja;
- Measurable (mensurável) usando as métricas corretas consegue-se determinar como você irá cumprir as metas, isso era tornar tudo mais tangível, um projeto de meses será possível definir marcos considerando as tarefas que serão realizadas;
- Achievable (alcançável) esse tópico é sobre você focar a importância para tornar a meta alcançável, podendo exigir uma mudança de atitudes e aprendizado de novas habilidades com o objetivo de inspirar e motivar a equipe;
- Relevant (Relevante) se refere em focar em algo que faça sentido de acordo com os objetivos de negócio, ou seja, alinhar suas metas com os objetivos de negócios;
- Time-bound (definido pelo tempo), trace suas metas com prazos para finalizadas, isso será útil para definirmos quanto tempo esse projeto irá levar para ser concluído.

As histórias são escritas conforme as ações que o usuário irá executar, que serão transformados em requisitos necessários para construir o produto, por exemplo, visualizar os inscritos no edital: Como usuário administrador eu quero visualizar todos os inscritos de um edital, como usuário administrador eu quero classificar usuários inscritos em um edital.

Baseando-se nessa história as tarefas serão as seguintes: Implementar uma tela que retorne os inscritos separados por editais que foram publicados. Implementar um método que liste inscritos no edital escolhido e ordene conforme os requisitos de classificação.

Como já foi citado, as histórias e tarefas devem ser independentes, para que estejam claras e seu desenvolvimento possa ser realizado isoladamente, dando mais autonomia ao time.

Muitas ferramentas são usadas para aplicar o método Scrum em projetos, dois principais são Kanban Board e Burndown (BURNDOWN..., 2019). O Kanban é um método que nasceu em uma fábrica da Toyota e se popularizou na metodologia ágil, como ele melhora a visualização do trabalho, o quadro usa cartões e colunas para controlar o fluxo de trabalho, normalmente essas colunas são divididas em “A fazer”, “Execução” e “Pronto”, o Kanban basicamente irá traduzir as Sprints e transformá-las em cartões para organizar as tarefas de acordo seu estado nessas colunas, a Figura 7 nos mostra um exemplo de uso deste método.

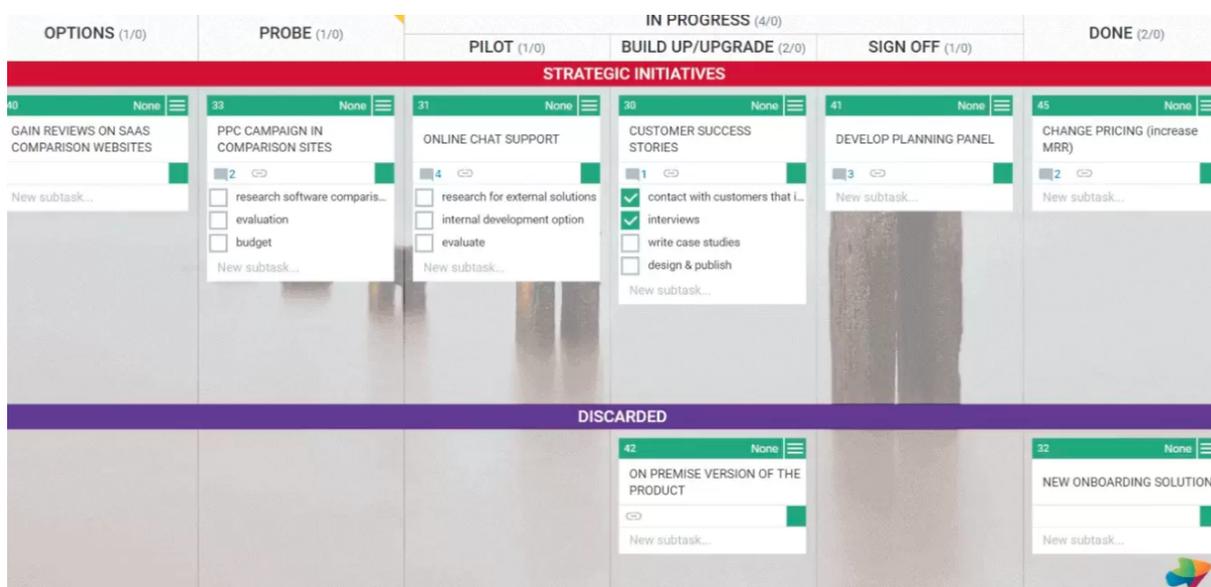


Figura 7 – Exemplo do Uso do Trello com o Método Kanban

Fonte: Autor

O Burndown oferece uma visualização rápida de como anda o desempenho do time de acordo com o cronograma que foi estabelecido em forma de gráfico (8). O gráfico apresenta dois eixos, um apresenta o tempo de uma Sprint e o outro o esforço que o time deve ter para completá-la. Normalmente, no final do expediente o time vê o que foi finalizado e atualiza o gráfico conforme os pontos da tarefa.

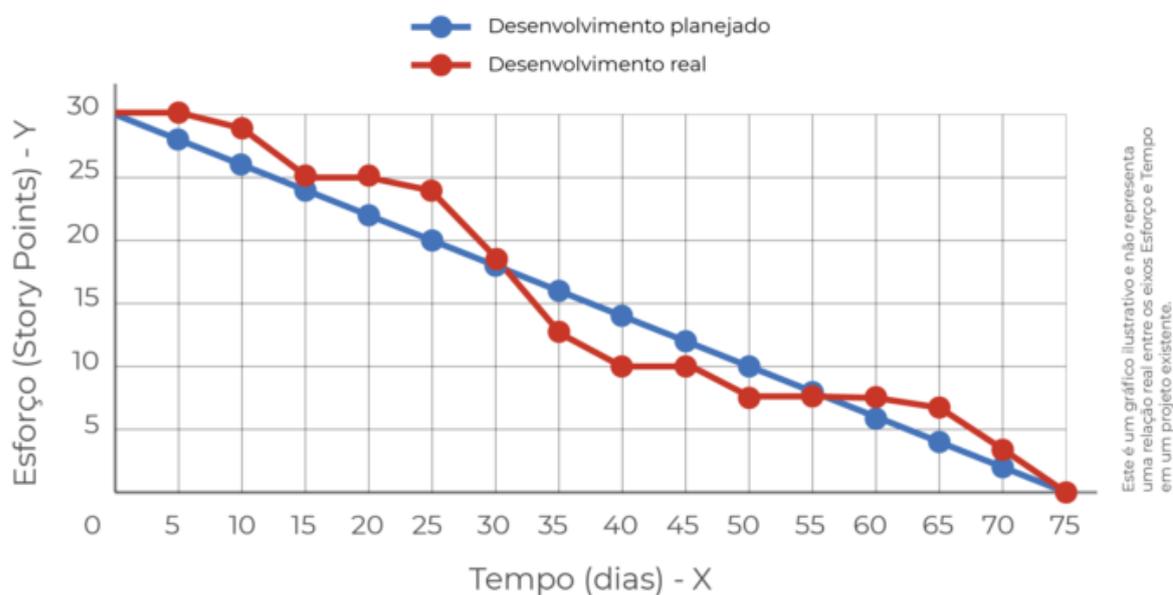


Figura 8 – Exemplo do Gráfico Burdown

Fonte: Autor

No gráfico de Burnup presente na 9, o eixo Y vertical exhibe onde a equipe está ao longo da Sprint, mostrando seu progresso enquanto a linha sobe, o que reflete uma linha ideal é uma linha reta. Nesse tipo de gráfico consegue-se verificar o planejamento durante as semanas analisando o quanto a linha já subiu e o quanto falta para alcançar o planejamento.

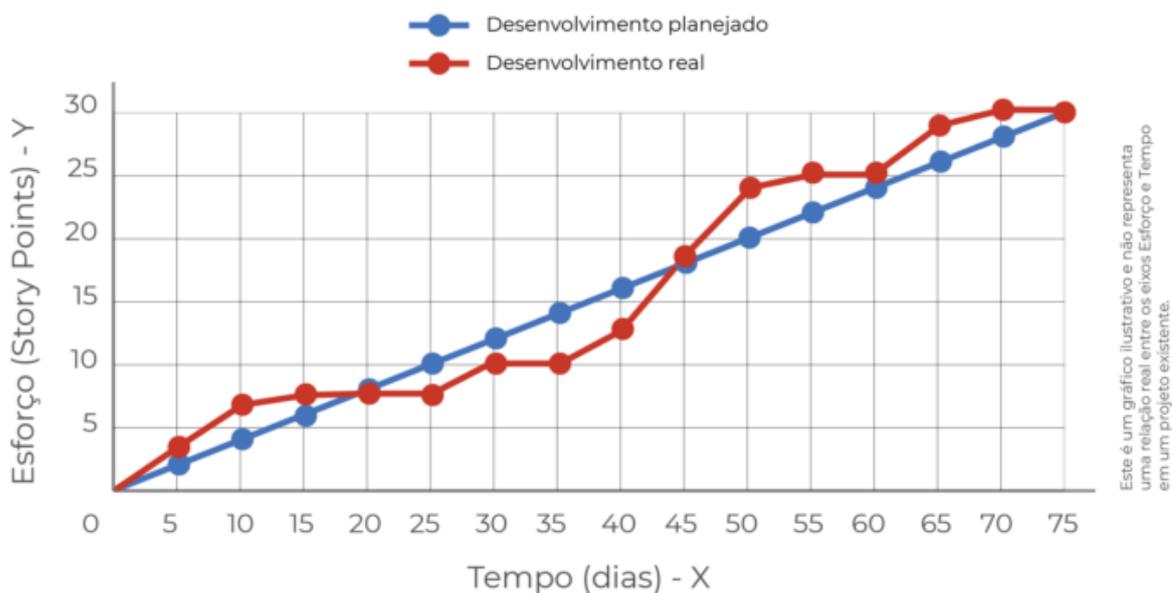


Figura 9 – Exemplo do Gráfico Burnup

Fonte: Autor

Os dois gráficos aparentemente tem o mesmo objetivo, ilustrar o desempenho e o progresso da equipe durante a Sprint, enquanto o de Burndown nos mostra a redução de acordo com a conclusão das tarefas. O de Burnup parte do zero e cresce pelo mesmo motivo: representam a mesma informação mas de maneira invertida.

3.2 APLICAÇÃO DO SCRUM NO PRESENTE PROJETO

A aplicação do Scrum será adaptada de acordo com as necessidades e realidade do desenvolvimento, as sprints serão realizadas de 15 em 15 dias com reuniões semanais para verificar o andamento das mesmas.

O Product Owner será o funcionário representante da SEMEC que estará presente em todo o desenvolvimento, responsável por definir as demandas do sistema e também irá verificar as sprints e aceitá-las se acaso estiverem concluídas. Professor orientador responsável por orientar o Development Team nas regras de negócio e orientar o Product Owner na criação do Product Backlog, assumirá o papel de Scrum Master.

Nas primeiras etapas do projeto, o time de desenvolvimento era composto por três acadêmicas. Nas novas implementações assumindo o papel do Development Team, a acadêmica, autora deste trabalho, dará continuidade sendo responsável por todo o desenvolvimento e entregas desse módulo de acordo com as sprints criadas.

Product Backlog está sendo utilizado desde o início do projeto. Toda sprint está sendo baseada nesses conjuntos de histórias criadas pelo Product Owner, infelizmente, Dailys Scrum não serão realizadas diariamente devido ao tempo disponível de todas as partes. A fase de Sprint Review será feita de 15 em 15 dias e após essa etapa realizasse a Sprint Retrospective.

Sempre buscando a melhor forma para criar as histórias, será necessário basear-se em abordagens INVEST e Smart, como já citado, seus critérios auxiliam na criação de histórias de qualidade para as Sprints - que facilitarão todo o processo de desenvolvimento.

Com o método Kanban como uma ferramenta da metodologia ágil, utiliza-se o Trello, tal como nas outras etapas será mantido o mesmo padrão. Ainda não estão definidas que ferramentas serão utilizadas para os gráficos Burdown, sua criação oferece uma visualização rápida do estado do nosso cronograma, se ele está de acordo com o que foi planejado.

3.3 FLUXO DE DESENVOLVIMENTO

O processo de desenvolvimento proposto inicialmente era baseado no método Scrum, onde na realização das reuniões se definia os requisitos necessários a serem desenvolvidos. Com os requisitos já definidos, é possível a criação de User Stories que serão usados como base para a realização das tarefas, após definir as tarefas é escolhida uma para iniciar o desenvolvimento.

Ao iniciar, o desenvolvedor abre uma nova branch e dá início a sua tarefa, quando finalizada submete a branch para o github e abre uma nova pull request e iniciando automatica-

mente o processo de building, onde ele irá também executar também seus testes e verificações de qualidade de código.

Com a conclusão dos testes alcançando sucesso, a nova funcionalidade é testada pelo orientador do projeto e se estiver correto é realizado a subida para o ambiente de produção, onde estará disponível para o responsável da secretaria verificar. Caso aprovado, inicia-se uma nova tarefa, se não é retornado ao desenvolvimento. O fluxo se encontra ilustrado na Figura 10.

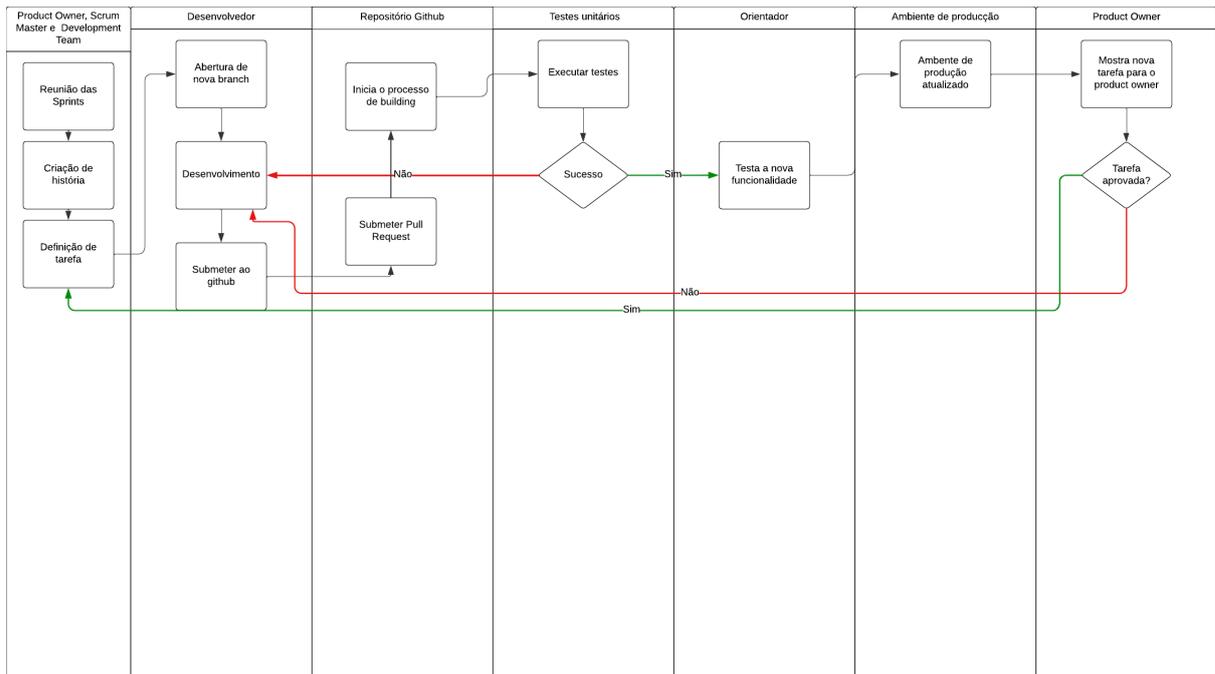


Figura 10 – Fluxo de Desenvolvimento Proposto

Fonte: Autor

4 DESENVOLVIMENTO

4.1 DOMÍNIO DO PROBLEMA

O problema e o domínio do problema serão apresentados nesta seção. Será exemplificado como é feito todo o processo manual de classificação realizado pela secretaria e a solução proposta por este trabalho.

4.1.1 O Processo Manual de Classificação

O processo de classificação, atualmente, é realizado de maneira manual por ao menos dois servidores. Os editais de remoção são publicados no site da Prefeitura Municipal de Guarapuava, com os detalhes legais, cronograma das etapas, regras relacionadas ao processo e a quantidade de vagas disponíveis para cada unidade.

Ao publicar um edital, é disponibilizado um formulário por meio do google forms, para preenchimento dos servidores onde é possível registrar a matrícula, tipo de remoção, unidade atual e sua unidade de interesse, sendo permitido a escolha de até três unidades. Em caso de formulários duplicados, que ocorrem quando o servidor cadastrou seu interesse mais de uma vez, será considerado o último atualizado por ele.

Os responsáveis pelo processo recebem as inscrições e separam conforme as vagas de interesse, após são organizadas pelas unidades, sendo que o administrador verifica separadamente cada inscrição feita. Pela matrícula se tem acesso às informações do servidor, estas são registradas nos documentos físicos disponíveis na secretaria, devido a isso, para verificar os dados do servidor deve-se que procurar entre as, milhares de fichas guardadas na secretaria, os dados contidos nesses documentos resultam na pontuação do servidor e na sua classificação.

A pontuação é calculada separadamente pelos critérios preestabelecidos pela secretaria e publicados no edital, atualmente esse processo é realizado totalmente de forma manual, o funcionário responsável realiza a somatória de pontos manualmente de cada inscrito, consultando a documentação física do servidor, os critérios atuais são:

- Maior tempo de serviço no município;
- Maior titulação;
- Maior tempo de serviço na escola.

A primeira comparação é realizada pelos pontos do tempo de serviço no município em dias, se acaso houver empate, é verificado a titulação e quantos pontos ela vale para o critério e o compara com a formação dos outros concorrentes. Na ocorrência de um novo empate, é averiguado se os servidores já trabalharam na unidade e se sim quantos dias foram. Se após todos os critérios verificados e os inscritos ainda se encontrarem empatados, o mais velho assume.

No ato da inscrição, o servidor pode adicionar interesse a vagas que não são menci-

onadas no edital, pois durante o processo elas podem surgir após um inscrito assumir uma vaga distinta da sua. Quando surge uma nova vaga, se deve verificar novamente quais usuários tem interesse na nova unidade e repetir todo o processo, demandando muito tempo e trabalho repetitivo do administrador.

4.1.2 Solução Proposta

A solução proposta visa a automatização do processo de classificação, diminuindo consideravelmente o trabalho do administrador no processo. O cálculo das pontuações dos requisitos preestabelecidos serão realizados no ato da inscrição e armazenados para as comparações futuras para a classificação dos servidores.

Ao realizar a inscrição, o sistema irá acessar os dados do servidor armazenados previamente em seu contrato vinculado a inscrição, as classificações serão separadas de acordo com suas unidades de interesse. O primeiro requisito é calculado por quantos dias se passaram da data de admissão encontrada no contrato até a data da realização da inscrição e armazenada.

Para a formação, será armazenado previamente seu nome e sua pontuação definida pela secretaria, quando se criar o cadastro complementar é adicionado essa formação ao servidor e será acessada para calcular a pontuação referente a ela.

O sistema contém registros das movimentações dos servidores que já se encontravam desenvolvidas, as movimentações são referentes as quais unidades o servidor já trabalhou durante o seu tempo de trabalho no município. Se o usuário contém uma movimentação na unidade que ele demonstrou interesse, será calculado os dias que seu contrato ficou vinculado e armazena para a comparação.

Com os cálculos das pontuações realizados a classificação se torna fácil, o algoritmo será um método de ordenação que será apresentado nas próximas seções aplicado em todos os critérios de forma separada, em caso de empate, o sistema irá ordenar pelo próximo critério. Após ordenado é apresentado ao administrador quem pode assumir a vaga.

As vagas serão distintas por dois tipos, as cadastradas diretamente no edital onde o administrador adiciona qual a unidade e quantas vagas estarão disponíveis. As vagas remanescentes são as vagas que serão liberadas quando um usuário assumir a vaga e poderão também serem assumidas pelos servidores. Ao assumir a vaga, o sistema verifica se ainda contém vagas disponíveis na unidade e se encontrar o inscrito é retirado da classificação e a quantidade de vagas disponíveis é diminuída e a vaga remanescente da unidade atual será liberada.

O administrador verifica se ainda há vagas e inscritos interessados por ela, se acaso houver, todo o processo é repetido até a data de término do edital. Se não existirem mais vagas sendo elas cadastradas ou liberadas, o administrador encerra o edital. Todo o fluxo explicado é ilustrado na figura 11.

Com o desenvolvimento de todo esse processo mencionado conseguimos solucionar o problema com a classificação e automatizá-la. O problema com as vagas disponíveis e seu monitoramento consequentemente também será resolvido.

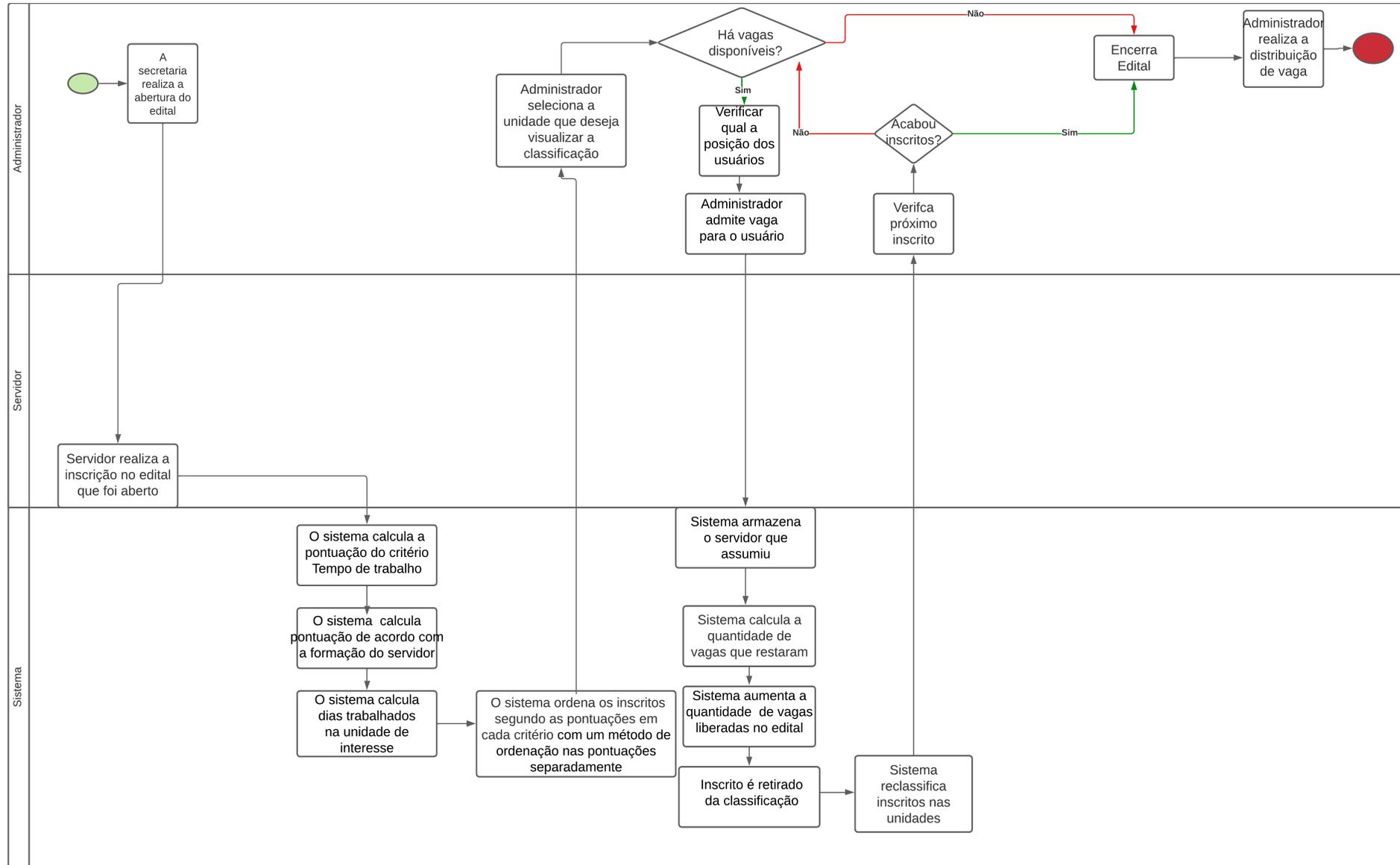


Figura 11 – Fluxo de ações do sistema

Fonte: Autor

4.2 PROCESSO DE DESENVOLVIMENTO

4.2.1 O uso do SCRUM

Inicialmente foi proposto o uso da metodologia ágil Scrum para o desenvolvimento, mas não foi utilizado por motivos de tempo e organização. As sprints, de 15 em 15 dias, não ocorreram, foram substituídas por reuniões feitas após o desenvolvimento de uma tarefa proposta, sem um tempo fixo e que variava conforme a complexidade da tarefa desenvolvida.

O único método utilizado no desenvolvimento pertencente ao SCRUM é o kanban, onde se transforma as tarefas em cartões que se movimentam entre as colunas normalmente divididas entre "A fazer", "Execução" e "Pronto" informando em qual estado a tarefa se encontra, melhorando a visualização do andamento do trabalho.

4.2.2 User Stories

Durante o projeto foi levantado os requisitos dos sistemas com o responsável pelo projeto na SEMEC no formato de *user stories*, pensando nas ações do administrador realizará no sistema. As *user stories* propostas, inicialmente, estão presentes no Quadro 2.

1	Listar inscritos por unidade	Como administrador, gostaria de listar todos os inscritos de uma unidade.
2	Visualizar requisitos de cada inscrito	Como administrador, gostaria de visualizar os requisitos de cada inscrito.
3	Classificar inscritos de cada unidade	Como administrador, gostaria de classificar os inscritos da unidade selecionada.
4	Verificar número de vagas	Como administrador, gostaria de verificar quantas vagas há disponível.
5	Atualizar vagas automaticamente	Como administrador, gostaria que o sistema atualizasse as vagas de forma automática.
6	Verificar se inscrito assumiu vaga	Como administrador, gostaria de verificar se o inscrito assumiu a vaga.

Quadro 2 – User Stories propostas inicialmente.

Fonte: O autor

No decorrer do projeto as *user stories* foram divididas em tarefas, sendo essas apresentadas na listagem a seguir:

1. Permitir que o administrador cadastre vagas em um edital, o número de vagas devem estar relacionadas com as unidades disponíveis no edital.
2. Adicionar na área administrativa o cadastro, atualização, visualização e remoção de formação dos servidores e suas pontuações.
3. Alterar o cadastro complementar do servidor para que a formação não seja um campo aberto e sim selecionada a partir do cadastro de formações.
4. Após a inscrição do servidor calcular e registrar a pontuação do critério Tempo de Trabalho na SEMEC.
5. Após a inscrição do servidor calcular e registrar a pontuação do critério Formação.
6. Após a inscrição do servidor calcular e registrar a pontuação do critério Tempo de Trabalho na Unidade de Interesse para as 3 unidades selecionadas pelo servidor.
7. Registrar a classificação dos servidores conforme sua pontuação, salvar os critérios e os pontos obtidos pelo servidor.
8. Na área administrativa, para cada edital, permitir o administrador listar as unidades e as vagas disponíveis.
9. Na área administrativa, para cada unidade e um edital, permitir o administrador listar os usuários inscritos já classificados. Deve-se apresentar suas pontuações em cada critério e a prioridade do servidor para aquela unidade, qual ordem de escolha do servidor para unidade classificada.
10. Na área administrativa, para a classificação de cada unidade, permitir que o administrador admita um servidor a uma vaga.
11. Na área administrativa, para a classificação de cada unidade, permitir que o administrador visualize os servidores inscritos separadamente daqueles que assumiram uma vaga.
12. O servidor que assumiu uma vaga não deve aparecer na classificação das outras unidades.
13. Na área administrativa, para a classificação de cada unidade, permitir que o administrador visualize a quantidade de vagas na unidade, a quantidade de vagas disponíveis e quantidade de servidores que assumiram vagas na unidade.
14. Quando um servidor é admitido em uma vaga, disponibilizar no edital a vaga em que ele pertencia (vaga remanescente), diferenciando da vaga cadastrada no edital.
15. Na área administrativa, para a classificação de cada unidade, permitir que o administrador visualize a quantidade de vagas remanescentes.
16. Caso a vaga remanescente não seja de uma unidade registrada no edital, criar um registro para a vaga nessa unidade e classificar os servidores inscritos nessa nova unidade.

4.2.3 Fluxo de Desenvolvimento Aplicado

Para se entender o fluxo de desenvolvimento usado no trabalho foi ilustrado na Figura 12, onde se mostra o papel do desenvolvedor, orientador e os ambientes usados durante o processo.

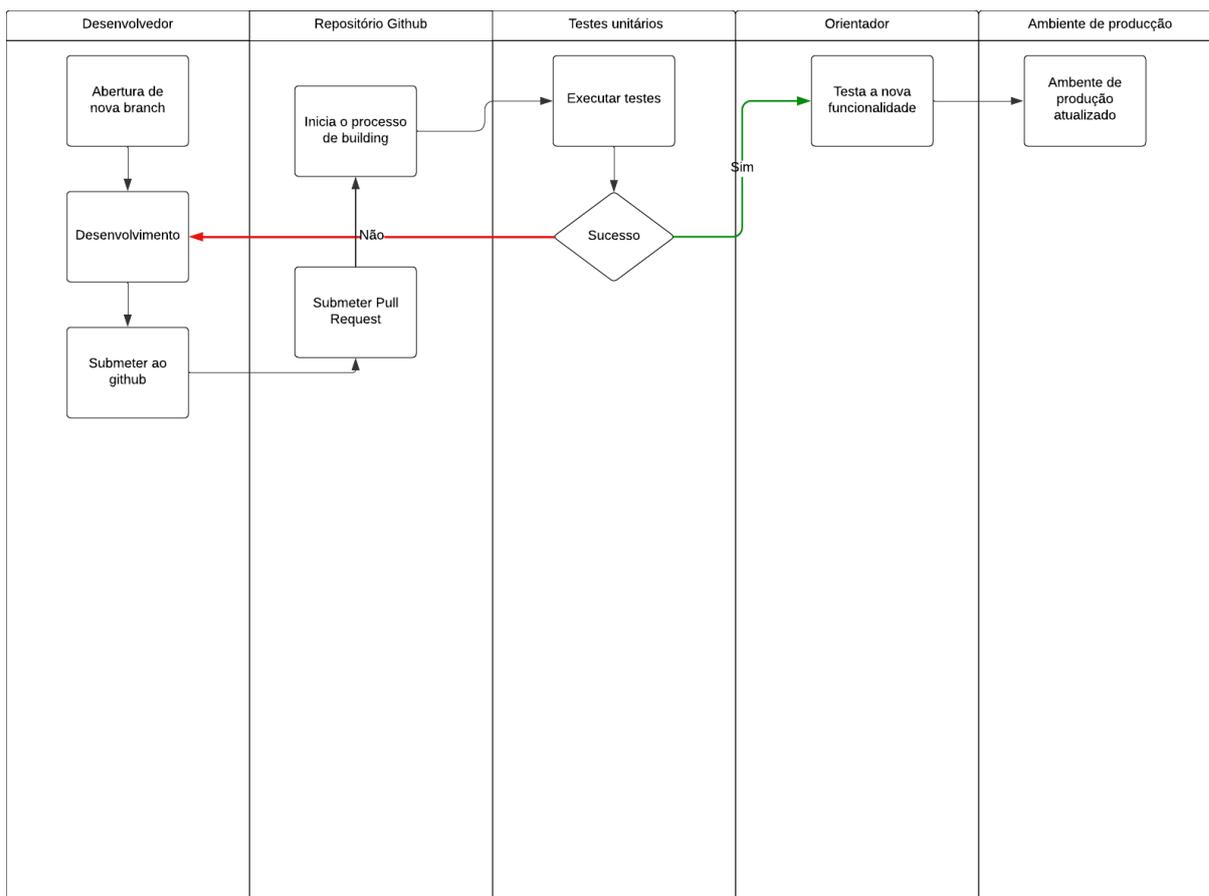


Figura 12 – Fluxo de desenvolvimento

Fonte: Autor

4.3 PROTÓTIPOS DE TELA

No início do desenvolvimento do projeto, foi pensando em telas que trouxesse a informação para o usuário de forma clara e sem a necessidade de muitas ações realizadas pelo administrador, realizando também um desenvolvimento reaproveitando as telas já existentes no sistema e sua padronização.

Durante o processo de desenvolvimento dos protótipos de tela, foi pensado em uma separação de classificação por unidades, como mostra o protótipo da Figura 13, onde contém a possibilidade de pesquisar e selecionar a unidade que deseja visualizar a classificação. O campo nomeado como “Selecione uma unidade” irá carregar uma lista com todas as unidades com vagas disponíveis no edital, sejam elas cadastradas ou liberadas.

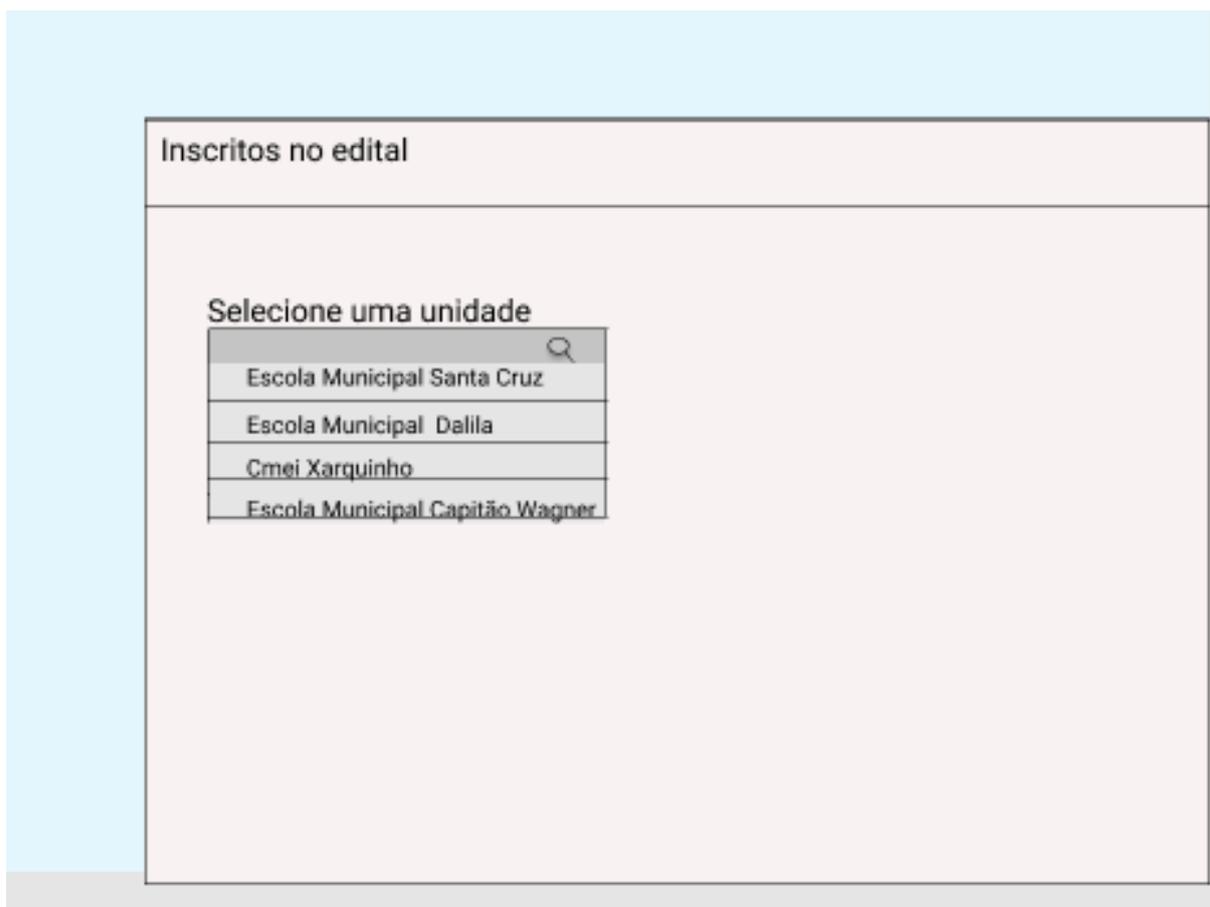


Figura 13 – Protótipo da Tela para Seleção de Unidade

Fonte: Autor

Ao selecionar uma unidade, o sistema irá trazer a lista de inscritos nessa unidade, com seu nome e um botão para a visualizar as informações da inscrição do servidor. Inicialmente, havia se pensando em um botão que se acionado é responsável por classificar as inscrições ordenando-os por seus requisitos, isso pode ser visualizado no protótipo da Figura 14.

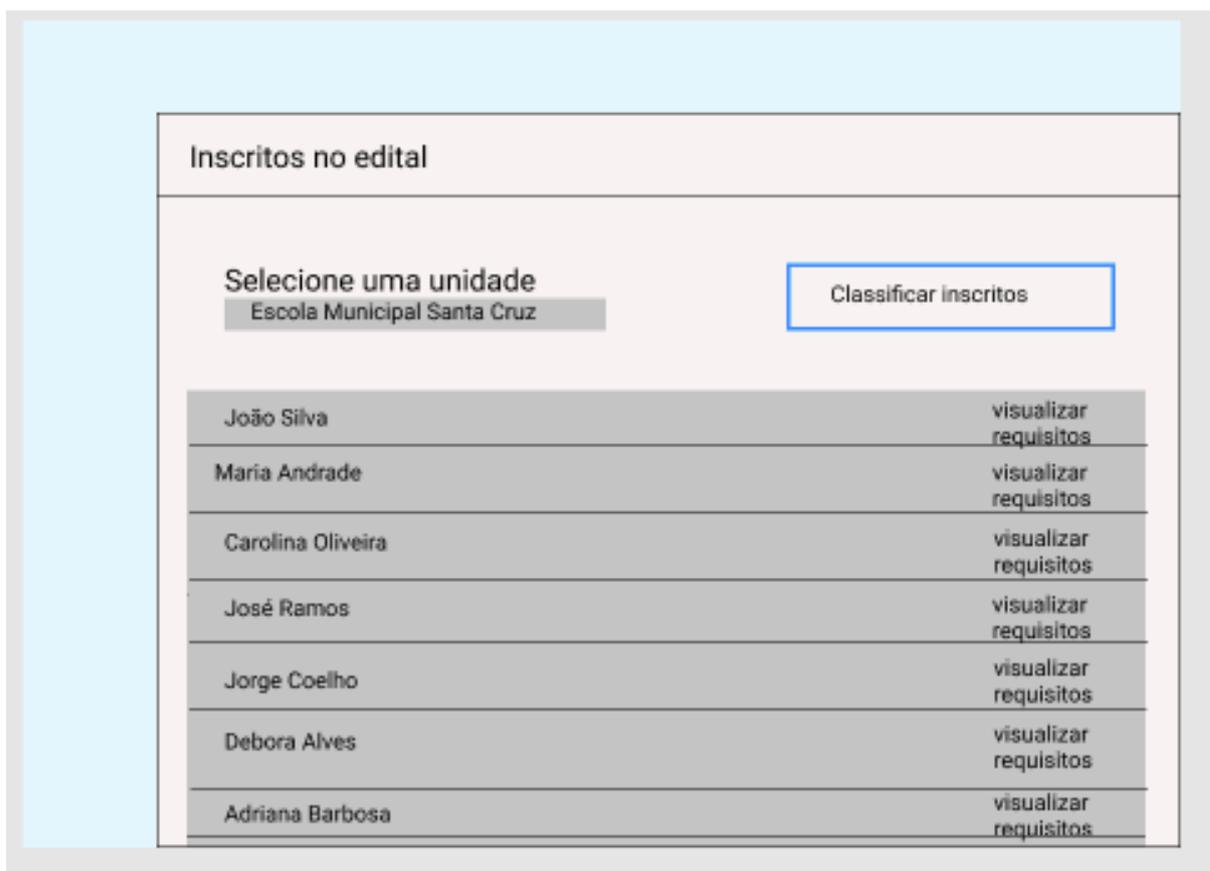


Figura 14 – Protótipo de Tela para Visualização de Inscritos

Fonte: Autor

O terceiro protótipo, exposto na Figura 15, traz ao administrador as informações do servidor relacionadas aos critérios, como o tempo de serviço dele no município, sua formação e seu tempo de serviço na unidade. Assim, o administrador consegue compreender o porquê da sua posição na classificação geral da unidade.

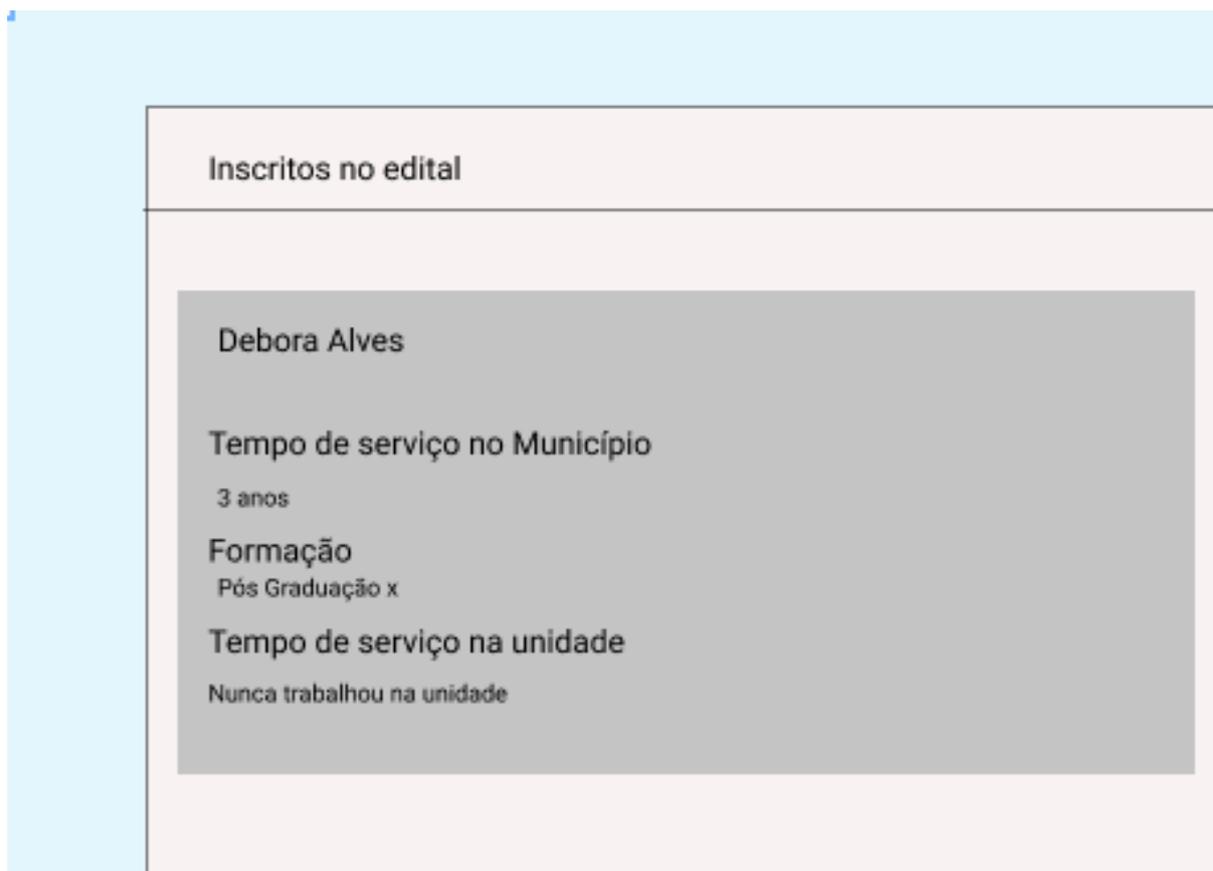


Figura 15 – Protótipo de Tela para Visualização da Pontuação

Fonte: Autor

4.4 VAGAS DISPONÍVEIS EM UM EDITAL

Uma das funcionalidades necessárias, não disponível no sistema, e que foi implementada, é o registro das vagas em um edital. Para isso, foram desenvolvidas as telas representadas nas figuras 16 e 17. Por meio destas, o administrador pode vincular as vagas a uma unidade e a um edital. Funcionalidade necessária para classificação dos servidores de acordo com as unidades disponíveis.

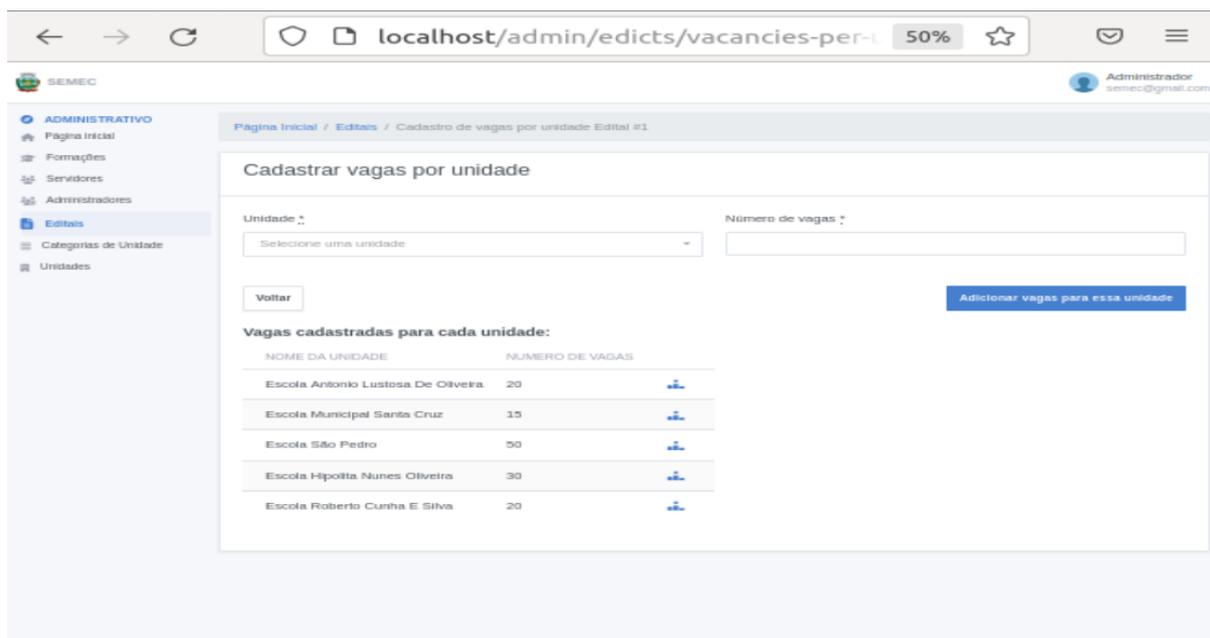


Figura 16 – Cadastro de Vagas para um Edital
 Fonte: Sistema de Gestão de Servidores da SEMEC

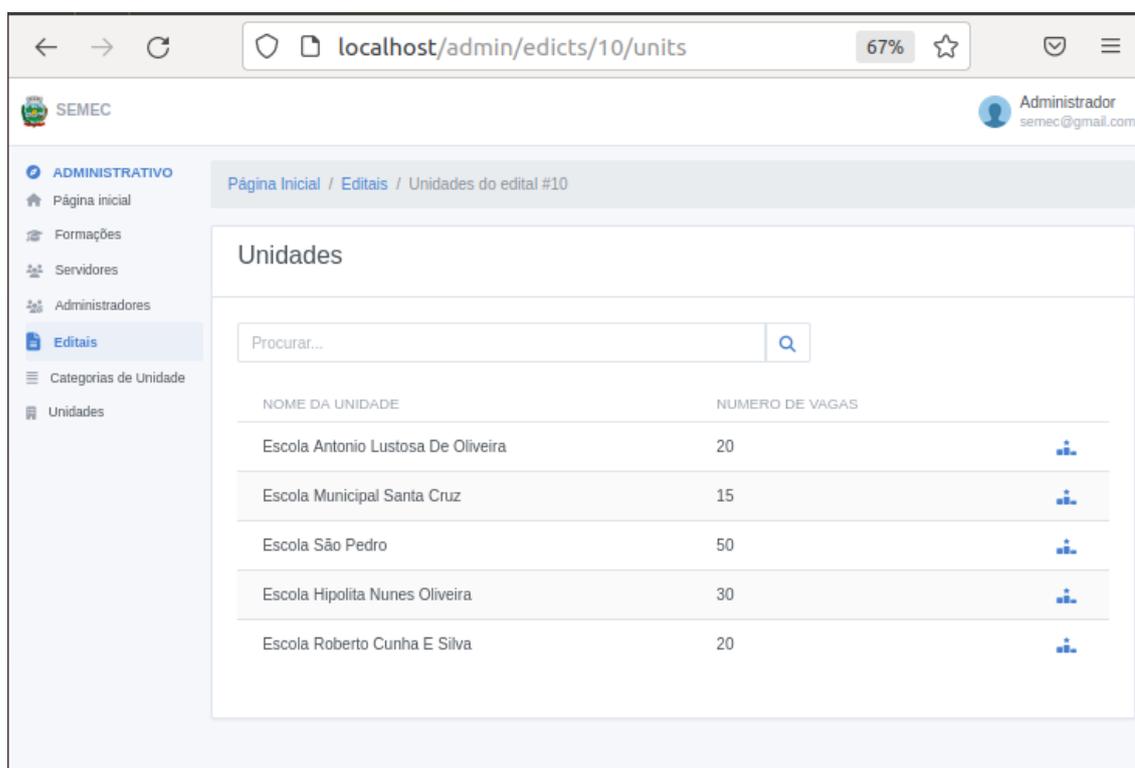


Figura 17 – Lista de Unidades Cadastradas no Edital.
 Fonte: Sistema de Gestão de Servidores da SEMEC

4.5 SOLUÇÃO IMPLEMENTADA

Nesta seção é apresentada a solução desenvolvida para o problema abordado por este trabalho. Assim, será retomado cada objetivo e então demonstrado como este foi resolvido. Esta seção está organizada da seguinte maneira: cada subseção apresenta um objetivo, sua descrição, a solução, destacando o pseudocódigo do algoritmo escrito, telas implementadas, quando existirem, mudanças necessárias no banco de dados e, por fim, uma discussão dos pontos relevantes.

4.5.1 Pontuação do servidor

Esta subseção visa apresentar o trabalho desenvolvido para cumprir com o primeiro objetivo, a saber: *Definir e implementar os cálculos para pontuação do servidor em cada um dos critérios pré-estabelecidos pela SEMEC*. A pontuação do servidor é utilizada para determinar sua classificação em um edital de remoção, sendo está estabelecida por três critérios, são eles:

- Tempo de trabalho na SEMEC;
- Formação;
- Tempo de trabalho na unidade de interesse.

O tempo de trabalho na SEMEC, é o primeiro critério a ser calculado, este é determinado conforme o registro da matrícula do servidor na secretaria. Sendo contado em dias, o primeiro dia é a data de admissão do servidor e último dia a data de abertura do edital. Deve-se ter atenção neste ponto, para que a data final seja a mesma para todos os inscritos, assim, estabeleceu a data de abertura do edital para garantir o cálculo correto do tempo para todos os inscritos. O Algoritmo 1 apresenta o pseudo do código do cálculo realizado para este critério.

Algoritmo 1: Calculo do critério Tempo de trabalho na SEMEC

```

admissionAt ← Select from contracts table to the current servant inscription
edictStartedAt ← Select from edicts table to the current servant inscription edict

admissionAtIndDays ← convertDateToDays(admissionAt)
edictStartedInDays ← convertDateToDays(edictStartedAt)

daysWorked = edictStartedInDays – admissionAtIndDays
store(daysWorked)

```

Conforme apresentado no Algoritmo 1, para cálculo desse critério é necessário recuperar a data de admissão do servidor do banco de dados. Esta informação está registrada na tabela *contracts*, coluna *admissionAt*. Destaca-se que um servidor pode possuir vários contratos de admissão, por isso, existe uma relação 1 para N entre as tabelas *servants* e *contracts*. Para diferenciar os vários contratos, o servidor informar a matrícula referente ao contrato que deseja utilizar para se inscrever em um edital de remoção.

Em um segundo momento, é necessário definir qual data será utilizada para determinar o fim período de trabalho, usado para o cálculo do tempo do servidor na SEMEC. Assim, conforme mencionada anteriormente, é usado a data de abertura do edital, de modo a garantir a mesma data final para todos os inscritos. Essa informação é recuperada da tabela *edicts*, coluna *startedAt* do banco de dados, isso para o edital em questão.

Então, essas duas datas são convertidas para dias, e após é realizado subtração da data de abertura do edital pela data de início do servidor na SEMEC para a matrícula escolhida. Assim, têm-se os dias trabalhos do servidor, sendo está a sua pontuação neste critério. Por fim, essa informação é salva no banco de dados em uma coluna específica da tabela referente a classificação dos servidores no edital, essa tabela é detalhada na seção 4.5.2.

O segundo critério diz respeito a formação do servidor, sendo que o servidor com maior formação terá uma maior pontuação. Para determinar a pontuação neste critério foi necessário alterar tanto o banco de dados quanto a aplicação. Na versão original do sistema a formação era cadastrada em um campo de texto aberto, dificultado relacionar uma pontuação com a formação. Assim, foi necessário criar uma tabela para registrar a formação, esta foi chamada *formations*, utilizada para registro da formação e de sua pontuação. Então foi realizada uma relação 1xN com a tabela *servant_complementary_datas*, relacionada nesta tabela a formação do servidor e a partir dela recuperada para o registro da sua pontuação, os detalhes dessas tabelas pode ser vistas na Figura 18. Adicionalmente foi implementado no sistema a funcionalidade para registros e edição da formação do servidor, isto pode ser visualizado na Figura 19.

Algoritmo 2: Critério Formação

```

inscription ← Select from inscriptions table to the current servant inscription
contract ← inscription → contract

servantComplementaryData ← inscription → contract →
  servantComplementaryData

scoreFormation ← servantComplementaryData → formation →
  scoreFormation

store(scoreFormation)

```

O novo campo e a nova tabela *formations*, alterações realizadas para atender a necessidade para uma pontuação do terceiro critério, são ilustradas no diagrama da Figura 18. A tabela contém dois campos, o valor de *name* será o nome da nova formação cadastrada pelo usuário e sua pontuação será definida por ele e armazenada para um futuro cálculo por meio do seu relacionamento 1xN com a tabela *servant_complementary_datas* apresentada ao seu lado.

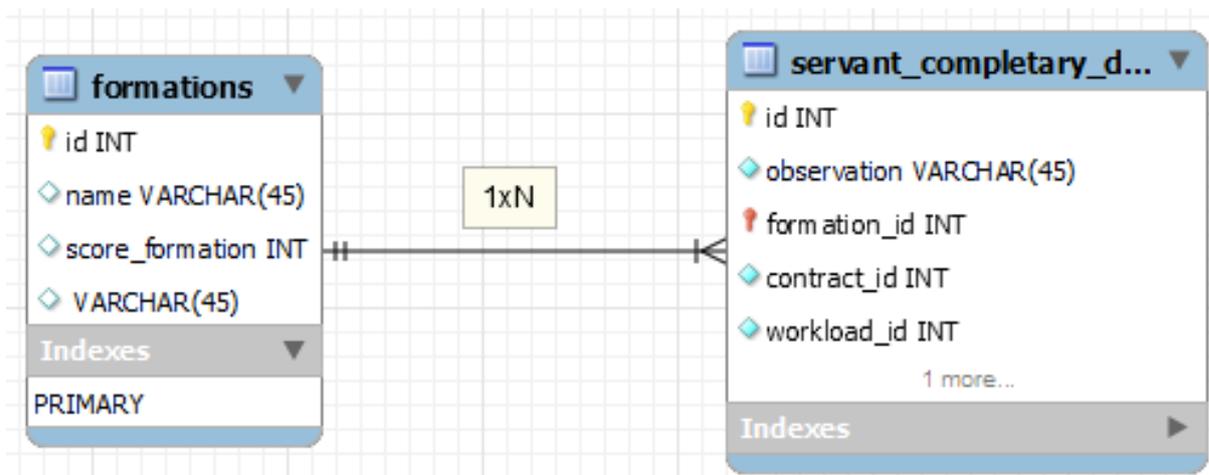


Figura 18 – Relacionamento Entre os Dados Complementares do Servidor e sua Formação.
 Fonte: Autor

Para criação de novas formações com pontuações, o sistema permite a realização do cadastro na tela presente na Figura 19, adicionando o nome e a pontuação desejada para essa formação. Após cadastrada, a formação ficará disponível na listagem de formações que serão usadas no cadastro complementar do servidor. O campo formação presente no formulário da Figura 20 tela do cadastro complementar, é do tipo seleção onde se pode selecionar uma formação e relacionar ela ao cadastro.

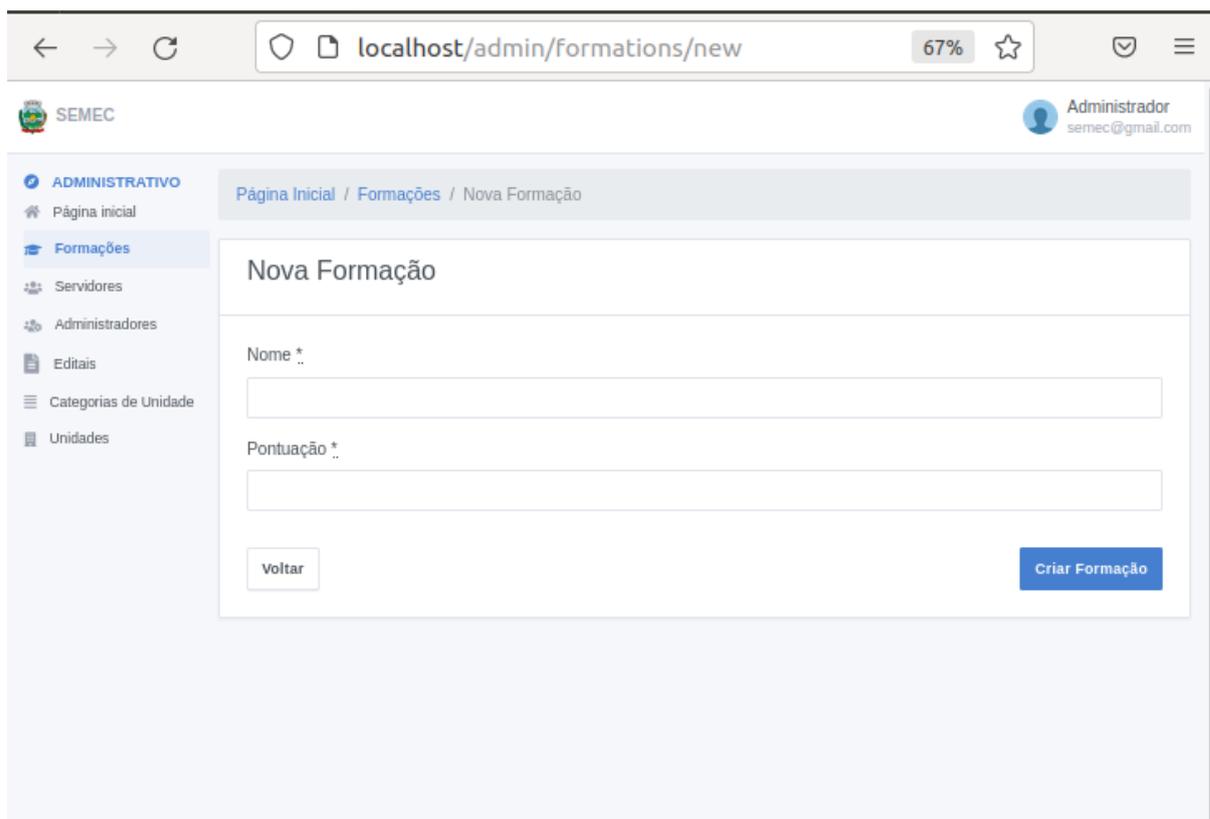
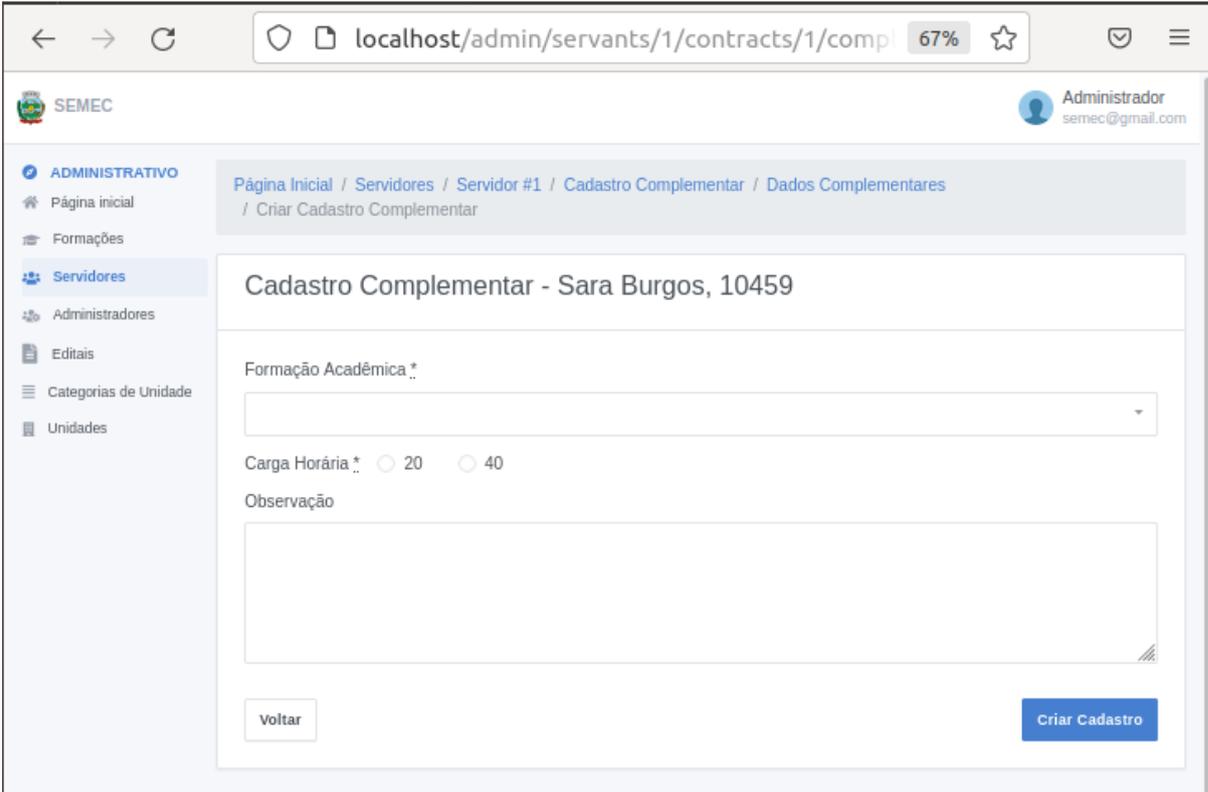


Figura 19 – Tela Cadastro de Nova Formação

Fonte: Sistema de Gestão de Servidores da SEMEC



The screenshot displays a web browser window with the URL `localhost/admin/servants/1/contracts/1/comp`. The page title is "Cadastro Complementar - Sara Burgos, 10459". The form contains the following elements:

- Formação Acadêmica:** A dropdown menu.
- Carga Horária:** Radio buttons for 20 and 40.
- Observação:** A large text area.
- Buttons:** "Voltar" (left) and "Criar Cadastro" (right).

The sidebar on the left lists the following menu items: ADMINISTRATIVO, Página inicial, Formações, Servidores (highlighted), Administradores, Editais, Categorias de Unidade, and Unidades. The top right corner shows the user "Administrador semec@gmail.com".

Figura 20 – Tela Cadastro de Nova Formação para um Servidor

Fonte: Sistema de Gestão de Servidores da SEMEC

O Algoritmo 2 apresenta o acesso à pontuação do servidor e seu registro para posterior classificação. Primeiramente recupera-se a inscrição do servidor registrada na tabela *inscriptions*, posteriormente a partir da inscrição têm-se acesso ao contrato, tabela *contracts*, do servidor, por meio deste, acessa seu cadastro complementar do servidor, tabela *servant_complementary_data*, e por fim, pelo relacionamento com a tabela *formations*, tem-se acesso à coluna *scoreFormation*, onde está registrado a pontuação da formação em questão. Assim, uma cópia dessa pontuação é realizada e salva na tabela de classificação, detalhada na seção 4.5.2. Para recuperar em qual formação foi pontuada, também é realizada uma cópia do nome da formação. Isso evita que em futuras modificações da pontuação na tabela *formations* não afetem a apresentação da classificação edital já concluído.

A pontuação do terceiro critério, leva-se em consideração o tempo de trabalho do funcionário nas unidades de interesse escolhidas no ato de sua inscrição de um edital. Neste momento, o servidor pode escolher até 3 unidades de interesse para remoção. Assim, é necessário verificar se este funcionário já trabalhou em alguma dessas unidades e, caso afirmativo, deve-se calcular o tempo, em dias, em cada uma dessas unidades para então registrar essas informações. Neste ponto, vale destacar que a classificação do servidor em um edital é realizada por unidade nas quais disponibilizadas vagas, assim é necessário determinar essa pontuação para cada uma das unidades de interesse do servidor.

O sistema criado para SEMEC, mantém o registro de todas as movimentações do

funcionário realizadas pelo seu contrato, assim, caso seja encontrada uma movimentação referente a ele e sua unidade de interesse, é contado quantos dias se passaram da data de admissão até data que seu contrato terminou na unidade, se não houver registros relacionados a pontuação desse critério será zero.

Para registro dessas informações, foi criada uma coluna na tabela de classificação para pontuação de tempo de trabalho na unidade de interesse, como a classificação do servidor é por unidade, cada unidade terá um registro na tabela de classificação relacionando, assim, sua pontuação nessa unidade. Caso exista mais de uma movimentação que registre a passagem do servidor pela unidade, a pontuação torna-se a soma dos dois períodos. Caso o servidor nunca tenha trabalhado na unidade de interesse, sua pontuação é zero. Um exemplo do Algoritmo utilizado para esse critério pode ser observado no Algoritmo 3.

Algoritmo 3: Critério Tempo em Unidades de Interesse

inscription ← Select from inscriptions table to the current servant inscription

interestUnits ← *inscription* → *interestUnits*

while *exists interestUnits* **do**

if *servant worked in the interest unit* **then**

while *exists period of work* **do**

 | *workedDays* + = *endedDay* – *startedDay*

end

 store worked days at classification table relation with unit, edict and inscription.

else

 | *workedDays* = 0

end

 store worked days at classification table relation with unit, edict and inscription.

end

4.5.2 Algoritmo para a Classificação dos Servidores

Nesta subseção é descrito o objetivo *Implementação do algoritmo para a classificação dos servidores*, o qual visa determinar como será realizada a classificação dos servidores. A classificação de servidores é definida pela verificação das pontuações dos critérios na exata ordem:

1. Pontuação tempo de trabalho na SEMEC;
2. Pontuação formação;
3. Pontuação tempo de trabalho na unidade de interesse.

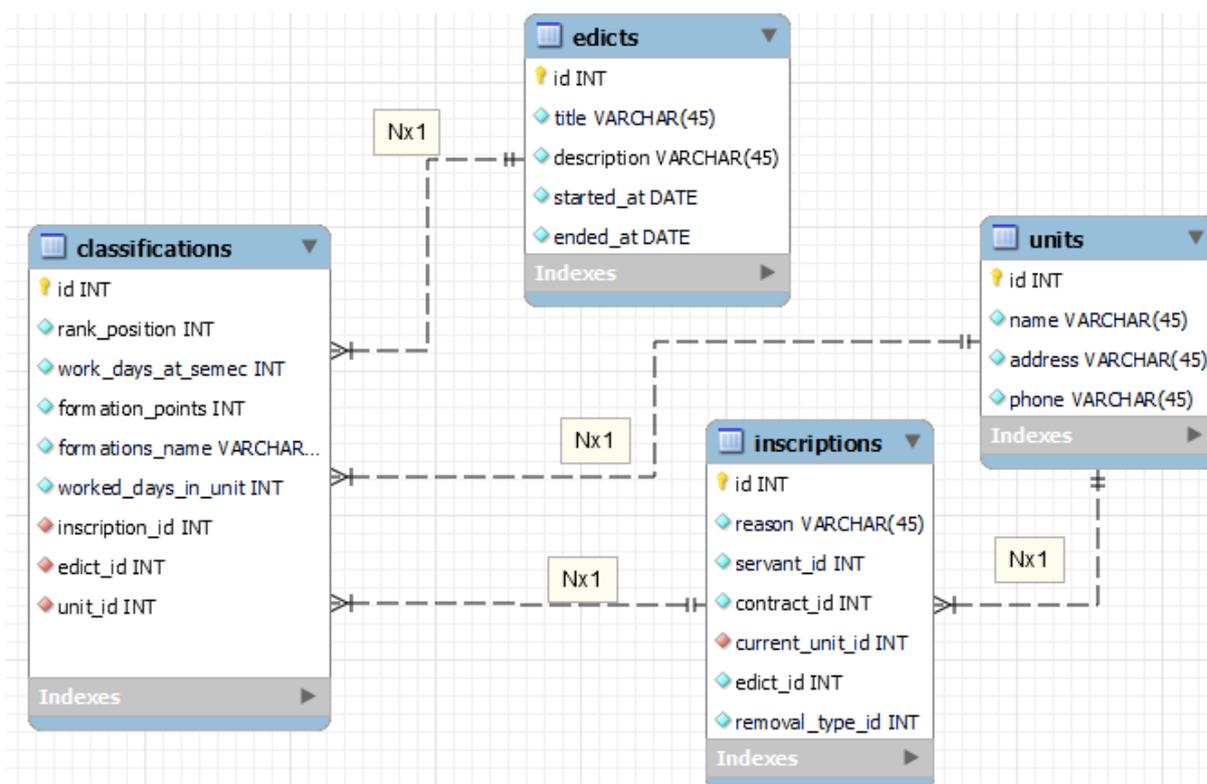


Figura 21 – Tabela Classificação

Fonte: Autor

Na tabela *classifications*, presente na Figura 21, é onde se encontram as pontuações armazenadas referentes a pontuação do servidor inscrito em um edital, sua ordenação e relacionamentos com as tabelas que contém informações relevantes para a classificação. A tabela *inscriptions* designada ao cadastro das inscrições permite se ter acesso as principais informações referentes ao servidor. As classificações ocorrem separadamente para cada unidade, devido a isso, se torna importante um relacionamento diretamente com a unidade, tabela *units*, outro relacionamento que a tabela mantém é com o edital, apesar de as classificações serem feitas por unidade elas só são válidas para o edital em que o servidor se inscreveu.

O campo *position rank* é responsável por armazenar a posição em que o servidor se encontra, este é atualizado a cada inscrição de um servidor no edital, e somente após os cálculos da pontuação e ordenação dos inscritos pelos três critérios na devida ordem. Para ordenar os inscritos é usado o comando ORDER BY do banco de dados. Assim, ordenação baseada nos cálculos dos critérios de pontuação fica simples. O Algoritmo 4 ilustra como é realizada a ordenação pelo sistema.

Algoritmo 4: Algoritmo de classificação

```
classifications ← SELECT * FROM classifications
WHERE edict_id = id from current edict AND
WHERE unit_id = ID from current unit
ORDER BY workedDays, formationPoints, workedDaysUnit DESC

while exists classifications do
| update column rank with index + 1
end
```

Os registros presentes em *classifications* são filtrados pela unidade e o edital que pertencem. A ordenação é iniciada pelos dias trabalhados armazenado no campo *worked_days_at_semec*, inscritos com maior pontuação ficarão nas primeiras posições. Em caso de empate, se compara o segundo critério presente do campo *formation_points* referente a pontuação da formação, servidores empatados pelo primeiro critério e com maior pontuação no segundo critério irão se classificar a frente.

No caso de empate nos dois critérios, um novo desempate é realizado baseado no campo *worked_days_unit* onde se encontra a quantidade de dias trabalhados do servidor na unidade. Em casos de empate em todos os critérios mencionados, o administrador verifica a idade dos empatados para o mais velho assumir.

Assim, o método usado para ordenar os inscritos após todos os cálculos das pontuações é simplificado, com o uso do ORDER BY DESC consegue-se a ordenação com o uso de cada coluna de pontuação separadamente e na ordem dos critérios estabelecida, como mostra o Algoritmo 3, sendo que os dados são ordenados de forma decrescente.

4.5.3 Apresentação da Classificação dos Servidores

Nesta subseção é descrito o objetivo *Implementação da apresentação da classificação dos servidores inscritos em um edital*, o qual visa apresentar a classificação dos servidores ao administrador do sistema.

A classificação do servidor é realizada para cada unidade, assim primeiramente o administrador seleciona a unidade que deseja ver os servidores inscritos e classificados, essa listagem pode ser observada na Figura 22. Na tela de apresentação da classificação por unidade presente na Figura 23, pode-se visualizar os inscritos que escolheram a unidade sendo ordenados após a classificação realizada na etapa da seção 4.5.2.

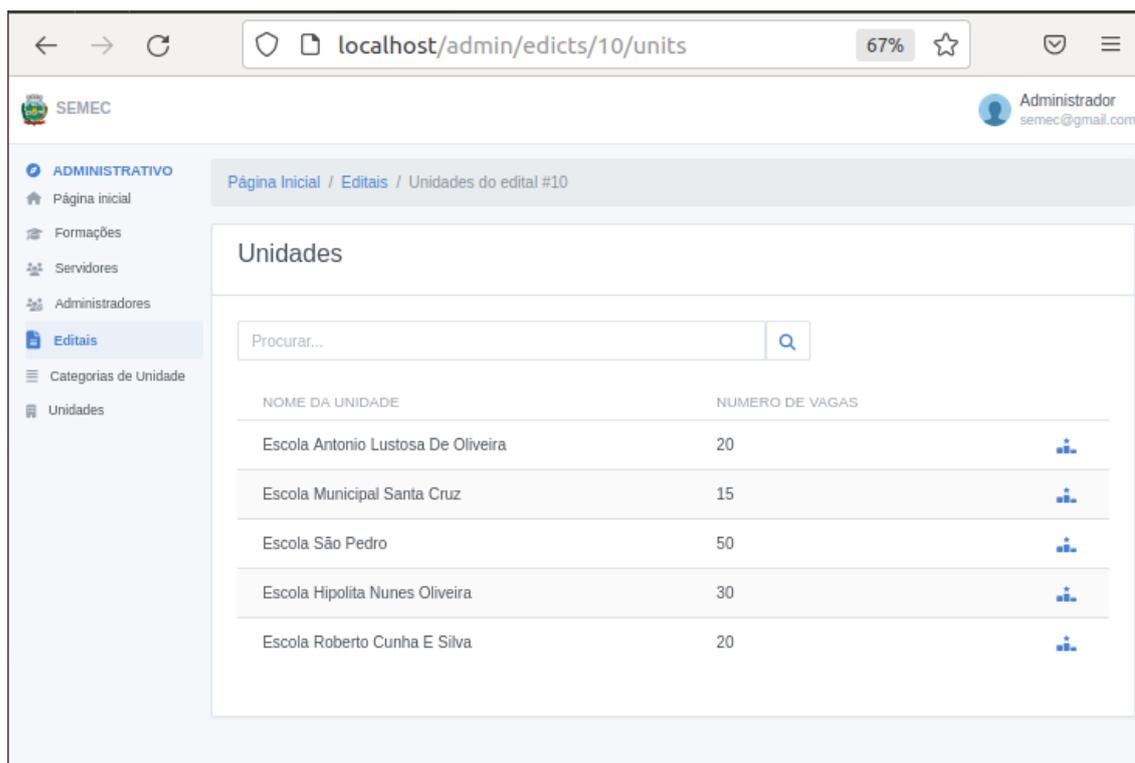


Figura 22 – Lista de unidades cadastradas no edital.

Fonte: Sistema de Gestão de Servidores da SEMEC

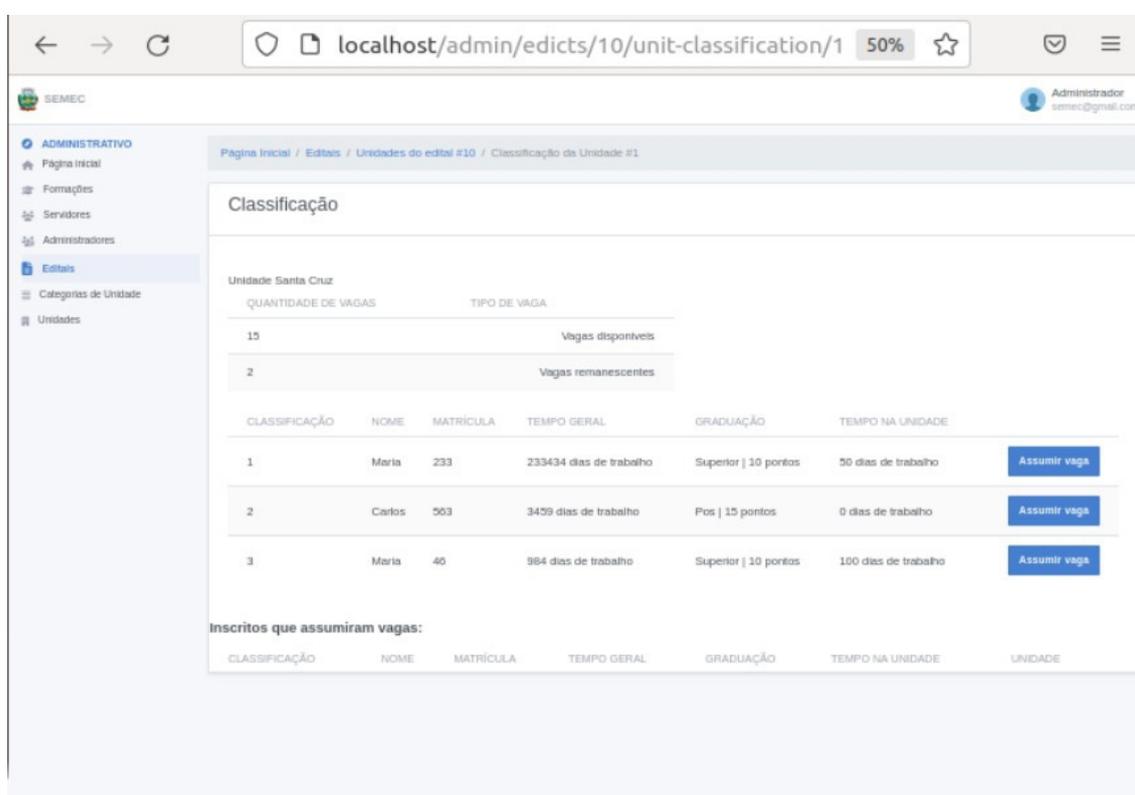


Figura 23 – Tela Classificações

Fonte: Sistema de Gestão de Servidores da SEMEC

Os inscritos, como consta na imagem, estão listados com suas informações necessárias para saber o porquê de sua posição. Na listagem está presente a coluna Classificação responsável por mostrar a posição do inscrito entre seus concorrentes, as próximas são o nome do inscrito e o seu tempo de trabalho na SEMEC que está presente na coluna nomeada como Tempo Geral, o próximo critério que se refere a formação é exibida com seu nome e sua pontuação. A coluna tempo de trabalho na unidade é apresentada com seus dias calculados, se o inscrito nunca trabalhou na unidade seu valor é zerado.

Com a visualização da listagem, o administrador consegue verificar qual inscrito está a frente dos concorrentes e a sua pontuação em comparação aos outros e dessa forma saber quem irá assumir as vagas disponíveis.

4.5.4 Admissão do Servidor a uma Vaga

Nesta subseção é descrito o objetivo *Permitir que o administrador admita um servidor a uma determinada vaga*, o qual visa o administrador atribuir uma vaga a um servidor.

Na listagem da classificação disponível ao administrador disponibiliza-se o botão responsável pela ação de admissão de vaga, ele se encontrará no registro do servidor indicando ao administrador quem deve assumir uma das vagas que se encontram disponíveis na unidade. Se acaso existir vaga disponível, o administrador realiza a admissão da vaga do primeiro colocado através do botão.

Após o funcionário assumir a vaga ele é retirado da classificação e adicionado em outra listagem como exemplificamos na Figura 24, onde se é armazenado sua posição e os pontos calculados de cada critério e a coluna indicando o nome da nova unidade mantendo o histórico para onde ele foi. A ação realizada afeta diretamente o total de vagas disponíveis da unidade, com a admissão feita o valor atualiza e diminui uma, por exemplo, se eram 10 vagas disponíveis agora terá apenas 9.

The screenshot shows a web application interface for SEMEC. The browser address bar is `localhost/admin/edicts/10/unit-classification/1`. The page title is 'Classificação'. The breadcrumb trail is 'Página Inicial / Editais / Unidades do edital #10 / Classificação da Unidade #1'. The left sidebar contains a menu with 'Editais' selected. The main content area shows 'Unidade Santa Cruz' with a summary of vacancies: 14 available, 2 remaining, and 1 assumed. Below this is a table of classified candidates with columns for classification, name, registration number, general time, graduation, and time in the unit. Each row has an 'Assumir vaga' button. At the bottom, there is a section for 'Inscritos que assumiram vagas:' with a table showing one candidate, Lucas, who assumed a vacancy.

QUANTIDADE DE VAGAS	TIPO DE VAGA
14	Vagas disponíveis
2	Vagas remanescentes
1	Vaga assumida

CLASSIFICAÇÃO	NOME	MATRÍCULA	TEMPO GERAL	GRADUAÇÃO	TEMPO NA UNIDADE	
1	Maria	233	233434 dias de trabalho	Superior 10 pontos	50 dias de trabalho	Assumir vaga
2	Carlos	563	3459 dias de trabalho	Pos 15 pontos	0 dias de trabalho	Assumir vaga
3	Maria	45	984 dias de trabalho	Superior 10 pontos	100 dias de trabalho	Assumir vaga

Inscritos que assumiram vagas:

CLASSIFICAÇÃO	NOME	MATRÍCULA	TEMPO GERAL	GRADUAÇÃO	TEMPO NA UNIDADE	UNIDADE	
2	Lucas	5056	4500 dias de trabalho	Superior 10 pontos	50 dias de trabalho	Santa Cruz	Assumiu a vaga

Figura 24 – Tela Classificações após assumir uma vaga

Fonte: Sistema de Gestão de Servidores da SEMEC

Para indicar que o servidor assumiu uma vaga ou não em mesmo edital foi adicionada a coluna *admission* do tipo booleano e a *admissionAt* do tipo timestamp na tabela de *classifications*. A primeira, quando true, indica que o servidor já assumiu a vaga, e não deve ser listados nas classificações das unidades do edital no qual assumiu a vaga. Como o servidor só pode assumir uma vaga por edital, está coluna é suficiente para essa informação. A segunda coluna é utilizada para indicar quando foi atribuído esta vaga ao servidor.

Para registrar em qual unidade o servidor assumiu a vaga, foi adicionada uma coluna do tipo Array na tabela *edict_units*, chamada *admissioned_servant_ids*, na qual é armazenado os ids dos servidores que assumiram as vagas para a unidade e edital em questão. Desse modo, também é apresentado o número de vagas assumidas subtraindo a quantidade de ids dessa coluna do total de vagas na unidade. O uso da coluna do tipo Array se deu pelo motivo de evitar mais uma relação NXN para as vagas assumidas, desse modo não precisa percorrer toda uma tabela para encontrar essas informações, pois já estão disponíveis na tabela de relação das unidades com o edital. A Figura 25, apresenta um fragmento do banco de dados com essas modificações.

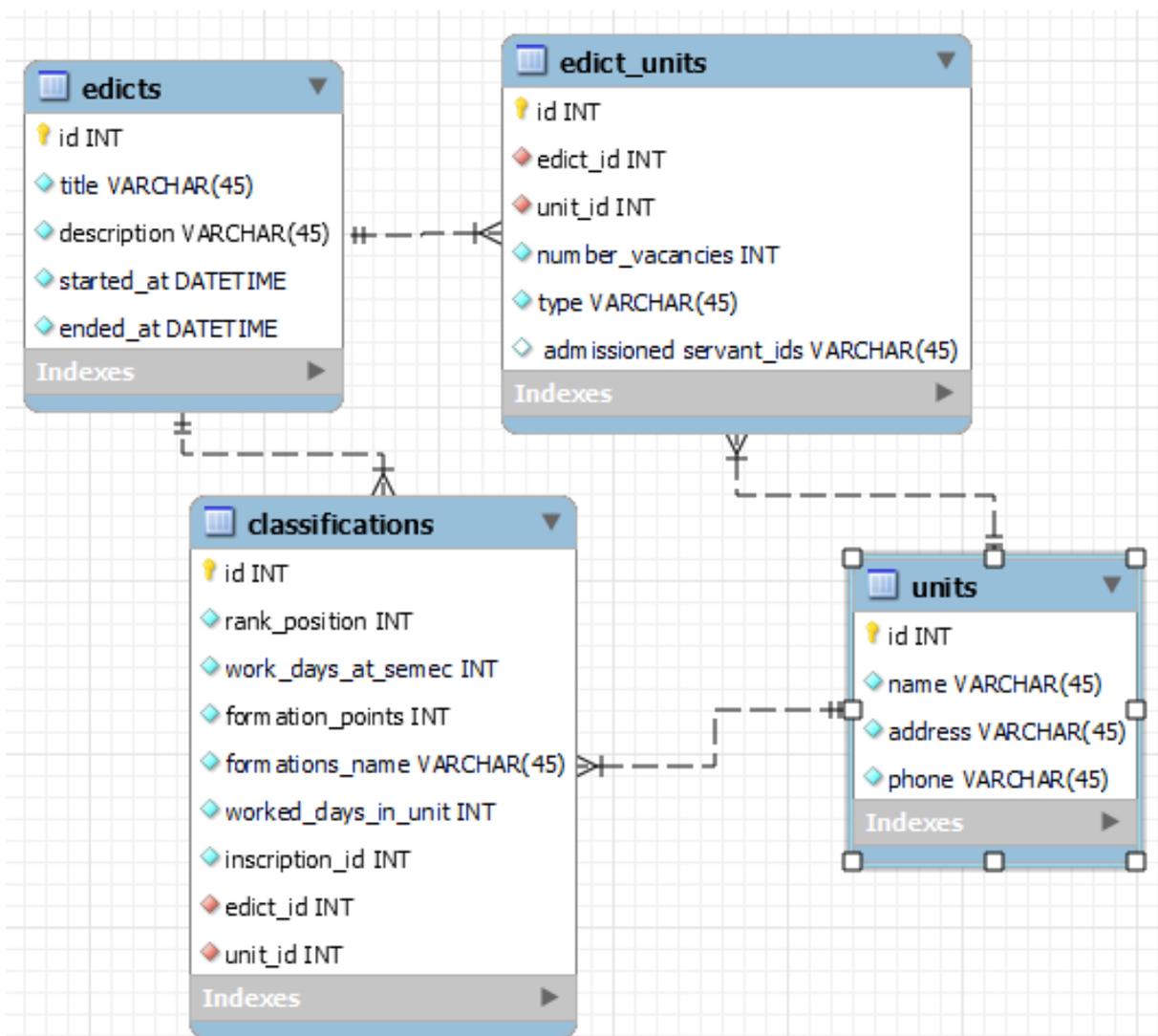


Figura 25 – Banco de Dados

Fonte: Autor

Vale destacar que as devidas validações para assumir a vaga foram implementadas, são elas:

- Somente pode assumir uma vaga, se esta existir;
- Servidor pode assumir apenas uma vaga por inscrição no edital;
- Após assumir uma vaga, o servidor não é mais apresentado nas classificações daquele edital. Apenas na listagem de vagas assumidas da unidade em que assumiu a vaga.

4.5.5 Monitoramento e Disponibilização das Vagas Remanescentes

Nesta subseção é descrito como foram atingidos os objetivos, *Monitorar a disponibilidade de vagas que podem surgir durante o processo de admissão dos servidores e Registrar e distinguir as novas vagas das vagas previamente inscritas no edital.*

Quando um servidor assume uma vaga em edital, a sua vaga se torna disponível para os demais servidores. Assim, após a admissão o sistema foi implementado um mecanismo que

insere um novo registro na tabela *edict_units* referente para unidade do servidor foi previamente admitido em outra vaga. Assim, outro servidor pode assumir essa vaga. Para distinguir das vagas originais do edital foi criado uma coluna adicional na tabela *edict_units* do tipo enum nomeada como *type_of_vacancy*, está pode receber dois valores, *released* e *registered*. O primeiro indica que é uma vaga do tipo liberada, o segundo que é uma vaga registrada no edital.

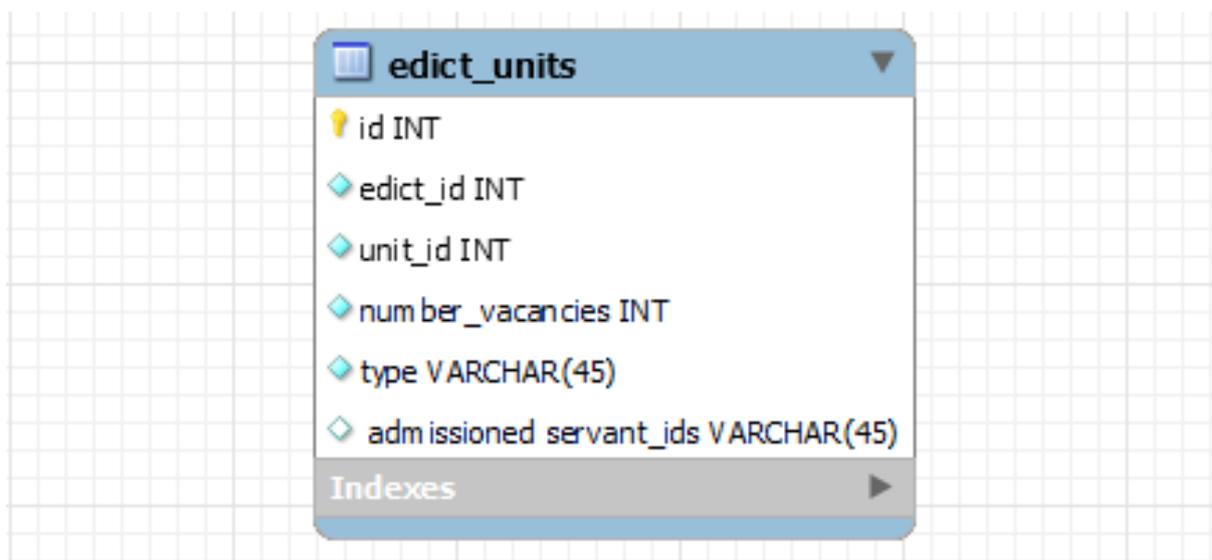


Figura 26 – Tabela de relacionamento entre edital e unidades

Fonte: Autor

Com essa informação, a listagem das unidades do edital, na qual o administrador seleciona a unidade para verificar a classificação, é dividida em duas tabelas, uma para as unidades com vagas registradas no edital e outra para as unidades com vagas liberadas durante o edital, o que pode ser observado na Figura 26. Permitindo, assim, a classificação dos servidores em uma nova unidade e sua posterior admissão pelo administrador do sistema.

4.5.6 Reclassificação dos Servidores para as Novas Vagas

Nesta subseção é descrito como foi atingido o objetivo Reclassificação dos servidores inscritos em um edital para as vagas que surgem durante o processo.

Com o surgimento de novas vagas no edital, é necessário verificar em que unidade a vaga surgiu, se for para uma unidade presente no edital não é necessário refazer a classificação, apenas disponibilizar essa vaga para aquela unidade. Porém, caso seja para unidade que não está presente no edital, é necessário a reclassificação dos servidores para esta nova unidade.

A reclassificação acontece da mesma forma que a classificação, pela inclusão de mais um registro na tabela classificações relacionando o servidor, edital e unidade. Porém, é necessário recalcular o tempo de cada servidor na unidade em que a vaga surgiu, caso ele já tenha trabalhado nesta unidade, de modo a garantir a classificação correta dos servidores.

5 CONSIDERAÇÕES FINAIS

O principal foco desse trabalho foi automatizar o processo de classificação de servidores inscritos em editais da SEMEC, o mesmo vem sendo feito a anos manualmente dificultando e muitas vezes atrasando a divulgação dos resultados do edital. Considerando o total de funcionários que estarão envolvidos no edital, torna-se algo trabalhoso, com automatização algo que levaria dias irá ser resolvido em menos de uma hora.

As principais dificuldades encontradas foram a administração de tempo para o desenvolvimento, a lógica para desenvolver uma classificação que englobe todos os requisitos sem necessitar de alterações maiores em partes do projeto que já estão feitos e sincronizar todas as atualizações de acordo com as movimentações feitas pelos servidores e administrador.

Reconhecendo a precariedade desse processo atualmente, percebe-se a necessidade de um sistema como esse, onde irá suprir os principais problemas encontrados quando acontece um lançamento de um novo edital. Com isso, poupamos tempo e trabalhos desnecessários para os funcionários responsáveis.

Referências

BURNDOWN e Burnup no Scrum: como avaliar o desempenho da sua equipe. 2019. Disponível em: <<https://www.flowup.me/blog/burndown-e-burnup/>>. Acesso em: 1 set. 2021. Citado na página 15.

CI/CD. 2019. Disponível em: <<https://medium.com/katalon-studio/continuous-delivery-vs-continuous-deployment-where-to-draw-the-line-3d220033dfb6>>. Acesso em: 28 jan. 2023. Citado na página 11.

ESLINT. 2023. Disponível em: <<https://eslint.org/>>. Acesso em: 28 jan. 2023. Citado na página 11.

SABBAGH, R. **Scrum: Gestão Ágil para Projetos de Sucesso**. [S.l.: s.n.], 2022. Acesso em: 31 jan. 2023. Citado na página 14.

SISTEMA para Gestão de Servidores da Secretaria Municipal de Educação e Cultura de Guarapuava. 2022. Disponível em: <<https://semec.tsi.pro.br/>>. Acesso em: 01 fev. 2023. Citado na página 1.

SUTHERLAND, J. **Scrum: The art of doing twice the work in half the time**. [S.l.: s.n.], 2014. Citado na página 13.