

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO SUL
CAMPUS RESTINGA**

**SISTEMA DE COLETA E PROCESSAMENTO DE
FREQUÊNCIAS PARA PAGAMENTO DOS AUXÍLIOS
ESTUDANTIS**

ADRIANO PRADO CAVALHEIRO

**Porto Alegre
2021**

ADRIANO PRADO CAVALHEIRO

**SISTEMA DE COLETA E PROCESSAMENTO DE
FREQUÊNCIA PARA PAGAMENTO DOS AUXÍLIOS
ESTUDANTIS**

Trabalho de Conclusão de Curso apresentado, junto ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal Educação, Ciência e Tecnologia do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Gleison Samuel do Nascimento.

**Porto Alegre
2021**

ADRIANO PRADO CAVALHEIRO

**SISTEMA DE COLETA E PROCESSAMENTO DE
FREQUÊNCIAS PARA PAGAMENTO DOS AUXÍLIOS
ESTUDANTIS**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para a obtenção do grau
de Tecnólogo em Análise e Desenvolvimento
de Sistemas.

Orientador: Prof. Gleison Samuel do
Nascimento.

Aprovado em ____, ____.

Orientador – Professor Gleison S. do Nascimento

Membro da Banca - Professor Diego Moreira da Rosa

Membro da Banca – Professor Pedro Chaves da Rocha

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL

Reitor: Prof. Júlio Xandro Heck

Pró-Reitora de Ensino: Prof. Lucas Coradini

Diretor-geral do *Campus* Restinga: Prof. Rudinei Müller

Coordenador do CST em Análise e Desenvolvimento de Sistemas: Prof. Iuri Albandes Cunha Gomes

Bibliotecária-chefe do *Campus* Restinga: Paula Porto Pedone

Dedico este trabalho a minha esposa Kelly Daiana
Vargas Serpa, amiga e companheira de todas as
horas.

AGRADECIMENTOS

Os agradecimentos principais: são direcionados aos professores, que contribuíram para o meu desenvolvimento acadêmico; aos colaboradores do IFRS – Restinga, que se dedicam para o funcionamento e o zelo da nossa instituição; aos meus colegas de trabalho, pelo incentivo e inspiração; a Márcia Pedroso, servidora do Setor de Assistência Estudantil, por toda a colaboração durante o projeto; a minha amada mãe Noeli Lopes Prado e ao meu amado irmão Anderson Prado Cavaleiro, por todo o apoio e amor ao longo dos anos; ao meu orientador Prof. Gleison S. do Nascimento, que foi fundamental para a realização deste trabalho e a Dra. Flávia Wagner, nossas seções têm me ajudado muito.

Agradecimentos especiais: são direcionados a minha amada esposa Kelly Daiana, pois sem ela nada disso seria possível.

“A verdade só pode ser encontrada em um só lugar: o código. Somente o código pode verdadeiramente lhe dizer o que ele faz.”

(Robert C. Martin)

RESUMO

Devido a importância do programa de assistência estudantil, que visa a manutenção de estudantes de baixa renda nas instituições de ensino, através do pagamento de auxílios. Sendo a frequência dos beneficiários nas aulas, requisito indispensável para o recebimento do auxílio cabe aos setores de assistência estudantil de cada câmpus do Instituto Federal do Rio Grande do Sul executar este controle mensalmente, a fim de enviar o relatório contendo os estudantes beneficiários do programa aptos ao recebimento do auxílio no mês referência, para o setor financeiro.

Portanto, a proposta deste trabalho é o desenvolvimento de um sistema de informação para automatizar a coleta e processamento das frequências dos estudantes beneficiários dos auxílios estudantis.

O objetivo é desonerar os servidores, que atuam de forma manual no processamento dessas informações, além da mitigação de riscos e redução do tempo empreendido na tarefa.

Como estudo de caso para o desenvolvimento do sistema de informação, foi mapeado o processo de pagamentos de auxílios estudantis, realizado no setor de Assistência Estudantil do IFRS - Campus Restinga, sendo implementada a automação a partir deste processo de trabalho.

O levantamento de requisitos do sistema contou com entrevistas, questionários e teleconferência com servidores da assistência estudantil. A partir das informações obtidas foi possível elaborar: diagrama de casos de uso, descrição detalhada dos casos de uso, diagrama de classe, diagrama entidade relacionamento e prototipagem de telas, que por sua vez geraram os artefatos necessários para estruturação do sistema.

A fase de desenvolvimento teve início após consulta de viabilidade para implantação do sistema, junto ao setor de Tecnologia da Informação do IFRS - Campus Restinga. Com a consulta, foi possível realizar a definição das tecnologias com base naquilo que já é utilizado pelo setor. Optou-se então pelo desenvolvimento com a linguagem de programação PHP, servidor de aplicação Apache e banco de dados MySQL. A fim de agregar eficiência ao desenvolvimento foram utilizados os *frameworks Laravel e Bootstrap*.

Testes com os usuários do setor de assistência estudantil foram realizados com o propósito de rastrear eventuais erros e identificar possíveis melhorias com base na experiência dos usuários, antes do lançamento do sistema em produção. Todos estes fatores foram importantes para a avaliação e conclusão deste estudo.

Palavra-chave: Sistema de informação, assistência estudantil, processamento de frequências.

ABSTRACT

Due to the importance of the student assistance program, which aims to maintain low-income students in educational institutions, through the payment of subsidies. Since the frequency of beneficiaries in classes, an indispensable requirement for receiving the aid is up to the student assistance sectors of each campus of the Federal Institute of Rio Grande do Sul to perform this control monthly, to send the report containing the students who are eligible for the program eligible to receive the aid in the reference month for the financial sector.

Therefore, the purpose of this work is the development of an information system, to automate the collection and processing of the frequencies of students receiving student aid.

The goal is too exonerative the servers, which act manually in the processing of this information, in addition to mitigating risks and reducing the time spent on the task.

As a case study for the development of the information system, the process of student aid payments was mapped, carried out in the Student Assistance sector of IFRS - Restinga Campus, and automation was implemented from this work process.

The system requirements survey included interviews, questionnaires and teleconferencing with student assistance servers. From the information obtained it was possible to elaborate use case diagram, detailed description of use cases, class diagram, entity relationship diagram and screen prototyping, which in turn generated the artifacts necessary for structuring the system.

The development phase began after feasibility consultation for the implementation of the system, with the Information Technology sector of IFRS - Restinga Campus. With the consultation, it was possible to make the definition of technologies based on what is already used by the sector. It was then decided to develop with the programming language PHP, Apache application server and MySQL as database. In order to add efficiency to the development, Laravel and Bootstrap frameworks were used.

Testing with users in the student assistance sector was conducted to track for possible errors and identify possible improvements based on user experience prior to the release of the system in production. All these factors were important for the evaluation and conclusion of this study.

Keywords: *Information system, student assistance, frequency processing.*

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo do processo de pagamento dos auxílios.	19
Figura 2 – Diagrama casos de uso.	21
Figura 3 – Detalhamento caso de uso Realizar login.	22
Figura 4 – Protótipos em HTML.	23
Figura 5 – Diagrama de classes.	23
Figura 6 – Diagrama Entidade Relacionamento.	24
Figura 7 – Arquitetura cliente servidor.	25
Figura 8 – Estrutura de <i>frameworks</i>	26
Figura 9 – Estrutura de pastas <i>Laravel</i>	26
Figura 10 – Padrão MVC.	27
Figura 11 – Migração Usuário.	28
Figura 12 – Modelo Administrador.	28
Figura 13 – Rotas de autenticação.	29
Figura 14 – Controle Administrador.	29
Figura 15 – Index Administrador.	30
Figura 16 – Estrutura de <i>templates Blade</i>	31
Figura 17 – Tela <i>login</i>	31
Figura 18 – Telas para cadastros.	32
Figura 19 – Telas de listagem e gerência.	32
Figura 20 – Telas para cadastros de beneficiários.	33
Figura 21 – Tela de listagem de beneficiários.	33
Figura 22 – Tela para upload das planilhas de frequências.	34
Figura 23 – Relação de infrequentes.	34
Figura 24 – Telas para notificações.	35
Figura 25 – Tela para justificativa.	35
Figura 26 – Alterada situação para pagamento autorizado.	36
Figura 27 – Tela de listagem de beneficiários aptos a receber auxílio.	36
Figura 28 – Impressões gerais.	38

LISTA DE TABELAS

Tabela 1 – Comparação entre os sistemas.	17
--	----

LISTA DE ABREVIATURAS E SIGLAS

AE	Assistência Estudantil
COVID-19	Coronavírus
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IFES	Instituições Federais de Ensino Superior
IFRS	Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
JSF	Java Server Faces
JSP	Java Server Pages
MEC	Ministério da Educação
MVC	Modew-View-Controller
ORM	Object-relational mapping
PHP	Hypertext Preprocessor
PNAES	Política Nacional de Assistência Estudantil
SAE	Servidor da Assistência Estudantil
SCPF	Sistema de Coleta e Processamento de Frequências
SQL	Structured Query Language

SUMÁRIO

1	INTRODUÇÃO	14
2	TRABALHOS RELACIONADOS	16
3	PROCESSO DE NEGÓCIO DE PAGAMENTO DOS AUXÍLIOS	19
4	SOLUÇÃO CONCEITUAL	21
4.1	Levantamento e análise dos requisitos	21
4.1.1	Diagrama de casos de uso.	21
4.1.2	Descrição detalhada de casos de uso	22
4.1.3	Prototipação de telas	22
4.1.4	Diagrama de classes	23
4.1.5	Diagrama entidade relacionamento.	24
5	DESENVOLVIMENTO	25
5.1	Implementação da Camada de Modelo	27
5.2	Implementação da Camada de Controle	28
5.3	Implementação da Camada de Visualização	30
5.4	Execução do sistema	31
6	TESTES	37
7	CONCLUSÃO	39
	REFERÊNCIAS	40
	APÊNDICES	42
	APÊNDICE A – DETALHAMENTO TECNOLOGIAS UTILIZADAS	43
A.1	Linguagem de Programação	43
A.1.1	PHP	43
A.2	HTML	43
A.3	CSS	43
A.4	Javascript	43
A.5	Padrões de Projeto	44
A.5.1	Modelo	44
A.5.2	Visão	44
A.5.3	Controle	44

A.6	Frameworks	44
A.6.1	Bootstrap	44
A.6.2	Laravel	45
A.6.2.1	Blade	45
A.6.2.2	Eloquent ORM	45
A.6.2.3	Migrations	45
A.6.2.4	Artisan Console	45
	APÊNDICE B – DETALHAMENTO DOS CASOS DE USO	46
	ANEXOS	70
	ANEXO A – DOCUMENTO DE TESTE COM USUÁRIOS	71

1 INTRODUÇÃO

Com o objetivo de modernizar o país no início dos anos 30 o governo passou a incentivar o proletariado brasileiro a educação, tendo em vista que, educação superior era apenas para aqueles que possuíam recursos para estudar no exterior. Desta forma surgiram as primeiras universidades no Brasil e com elas a necessidade de assistência à moradia e alimentação a estudantes egressos da classe trabalhadora. Essas ações eram realizadas pelas instituições de ensino com recursos limitados (DUTRA; SANTOS, 2017).

Com isso, estudantes de baixa renda passaram a ocupar cada vez mais vagas nas Instituições Federais de Ensino Superior (IFES), as instituições passaram a pleitear recursos para investir em programas de assistência estudantil (AE), visando a manutenção destes estudantes até a graduação. Apesar dos esforços da comunidade acadêmica, a AE não era tratado como necessidade de política pública, somente em 2007 foi instituído o Programa Nacional de Assistência Estudantil (Pnaes), pela Portaria Normativa do Ministério da Educação (MEC) nº39/2007, e regulamentado pelo Decreto nº7.234/2010, considerado marco histórico para a assistência estudantil (ESTRADA; RADAELLI, 2017).

O Pnaes tem como objetivo, garantir a estudantes de baixa renda o acesso às IFES e a permanência destes estudantes até a conclusão do curso, por meio da distribuição de incentivos na forma de auxílios estudantis à moradia, alimentação, transporte, assistência à saúde, inclusão digital e atividades culturais, como forma de minimizar as desigualdades sociais do país (MACEDO; ABRANCHES, 2018).

A participação do estudante no programa é de responsabilidade da instituição superior no qual o aluno deve estar devidamente matriculado, cabe a instituição por meio de editais realizar a seleção dos alunos e garantir o repasse dos benefícios aos beneficiários. Aos candidatos por sua vez, para participação no Pnaes, devem acompanhar os prazos estabelecidos para entrega dos documentos necessários, previstos em edital e responder a um questionário socioeconômico. Atendendo as exigências, os alunos tornam-se beneficiários do Pnaes, passando a receber mensalmente os benefícios. Porém, a condição para o aluno beneficiário garantir a manutenção do seu auxílio, é a de manter sua frequência nas aulas, sob pena de suspensão do auxílio. Este controle é atribuição dos servidores do setor de assistência estudantil da instituição (BRASIL, MINISTÉRIO DA EDUCAÇÃO, 2013). Neste contexto, fica evidente a importância da tarefa de verificação das frequências.

Com a definição do tema, Sistema de Coleta e Processamento de Frequências e com dados obtidos durante a revisão literária, identificou-se o problema: pagamento dos auxílios estudantis nas IFES. A partir disso, elaborou-se a seguinte questão: como, um sistema de coleta e processamento de frequências, pode auxiliar no processo de pagamento dos auxílios estudantis?

O presente estudo tem como objetivo responder esta questão. Para isso, propõe-se o desenvolvimento de um sistema que realize a coleta e processamento de frequências para o pagamentos de auxílios estudantis. Com a motivação de tornar este processo mais eficiente nas

instituições de educação superior, minimizando consideravelmente os riscos de erros, além de diminuir o tempo despendido no processo.

Partindo do fato que existe um número considerável de Institutos Federais chegando às centenas em todo o território nacional (BRASIL, MINISTÉRIO DA EDUCAÇÃO, 2018?), o setor de Assistência Estudantil do IFRS – Campus Restinga, foi escolhido como referência e limite para a análise do projeto.

O setor de Assistência Estudantil faz parte do setor de Ensino do IFRS - Campus Restinga, e tem por princípios, o enfrentamento às desigualdades sociais por meio da ampliação de acesso à educação para estudantes de baixa renda. O objetivo dos auxílios é permitir que alunos de baixa renda tenham condições de permanência na instituição até a conclusão do curso, de forma respeitosa, diversificada e inclusiva. Os estudantes participantes do programa, podem solicitar serviços como: acompanhamentos psicológicos, pedagógicos e de serviço social. Além de auxílios de permanência e moradia. Somente no ano 2020 o programa teve um total de 488 auxílios deferidos (IFRS. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL - CAMPUS RESTINGA, 2021?a). Os estudantes do IFRS - Restinga atendidos pelo programa, têm sua maioria, em famílias com renda per capita de até 0,5 salário-mínimo (IFRS. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL - CAMPUS RESTINGA, 2021?b).

Atualmente, a tarefa de processamento das frequências dos beneficiários do programa de assistência estudantil no IFRS - Campus Restinga, é realizada de forma manual, com a necessidade de alocação de pessoal e de tempo para a execução da tarefa. A verificação das frequências é indispensável para a realização dos pagamentos e tem demandado tempo significativo para sua execução, dificultando o processo de pagamento dos auxílios.

Este projeto ambiciona, com a implantação do sistema para o processamento de frequências, desonerar os servidores do setor de Assistência Estudantil, permitindo-lhes direcionar parcela maior de tempo em suas demais atribuições, tais como: atendimentos pedagógicos, psicológicos e os de serviços sociais. Para melhor atender a comunidade.

2 TRABALHOS RELACIONADOS

A partir da questão que este estudo visa responder, foi realizado o levantamento dos trabalhos relacionados. Com a utilização de questões elaboradas de acordo com a questão problema, e utilizando as ferramentas de pesquisa, Google Acadêmico¹ e o sistema de bibliotecas do IFRS - Pergamum². Como resultado do levantamento, três trabalhos relacionados à questão problema, foram selecionados para estudo.

SAAMS - Sistema para Assistência Estudantil e Acompanhamento Acadêmico, aborda o estudo realizado para a implantação de um sistema para assistência estudantil e acompanhamento acadêmico, além da concepção do sistema. O estudo surgiu com a necessidade, apresentada pelo setor de Coordenadoria de acompanhamento acadêmico do IFRS – Campus Porto Alegre, de um sistema para automatização das tarefas relacionadas à assistência estudantil e acompanhamento acadêmico, tendo em vista que as tarefas do setor eram realizadas por meio de planilhas ou documentos em texto (CARDOSO, 2019?).

A proposta do sistema visa a facilitação de alguns processos de gestão acadêmica e de assistência estudantil.

O módulo de assistência abrange a publicação de editais referentes à assistência estudantil, concessão e manutenção de auxílios. Também permite a submissão de documentos, inscrição para concorrer ao auxílio e o preenchimento de formulário socioeconômico por parte dos alunos. Além de, processos de acompanhamento acadêmico, tais como, monitoramento de aproveitamento e evasão, controle de atendimentos e gerenciamento do histórico sociopedagógico dos alunos.

SCAE – Sistema Coordenadoria Assistência Estudantil, sistema para gerenciamento de benefícios para a Coordenadoria de Assistência Estudantil com o uso de formulários dinâmicos. A demanda surgiu com a observação da quantidade de documentos que os candidatos precisavam entregar na inscrição, para concorrer aos auxílios estudantis, o que exigia tempo, esforço e o emprego de recursos humanos para o setor de assistência estudantil do Campus Farroupilha. A proposta de solução foi a implantação de um sistema para automatizar e gerenciar as tarefas de publicação de editais e inscrições de alunos via sistema, por meio de formulários dinâmicos (BREDA, 2017).

GD-PNAES - Gerenciamento de Distribuição do Programa Nacional de Assistência Estudantil, trata-se de, estudo de caso realizado no Campus Porto Nacional, pertencente ao Instituto Federal do Tocantins, sobre o impacto da implantação de uma ferramenta computacional que visa auxiliar no processo de distribuição de bolsas do PNAES. O objetivo é levantar dados sobre as dificuldades encontradas na implantação de ferramentas para a utilização de estudantes em condições de baixa renda (ANAIS DO WORKSHOP DE INFORMÁTICA NA ESCOLA, out. 2015).

O GD-PNAES foi construído na arquitetura *web* com módulo administrativo, que executa

¹ <https://scholar.google.com.br/>

² https://biblioteca.ifrs.edu.br/pergamum_ifrs/biblioteca/

processos de: publicações de bolsas, emissão de relatórios, controle de avaliação entre outros relacionados a avaliação socioeconômica do candidato. O módulo aluno permite ao candidato: acompanhamento de cronogramas, inscrição nas bolsas e anexo de documentos necessários para concorrer às bolsas. O estudo teve três avaliações, sendo duas entrevistando alunos que participaram do processo de seleção divididos em duas etapas, e a terceira entrevistando a equipe responsável pela oferta das bolsas. O sistema foi avaliado de forma positiva, com maior aprovação pela equipe responsável, já entre os alunos não houve consenso, embora a maioria tenha sido favorável ao sistema.

Os estudos relacionados apontaram que os sistemas, embora realizem tarefas distintas, têm o mesmo objetivo, que é tornar os processos de assistência estudantil eficientes. Porém, nenhum faz referência a tarefa de coleta e processamento das frequências, tarefa que mensalmente dispõe de tempo considerável dos servidores na sua execução.

O foco do Sistema de Coleta e Processamento de Frequências (SCPF), é auxiliar na tarefa de processamento das frequências dos alunos, que ocorre mensalmente e é fundamental para pagamentos dos auxílios. Diferente dos sistemas estudados, que tem seu foco no processo de oferta e seleção das bolsas.

Comparativo das características encontradas em cada sistema pesquisado, além de relacioná-las com o sistema proposto neste projeto (SCPF) Tabela 1.

Tabela 1 – Comparação entre os sistemas.

Características	SCPF	SAMMS	SCAE	GD-PNAES
Plataforma da aplicação	<i>Web</i>	<i>Web</i>	<i>Web</i>	<i>Web</i>
Banco de dados	MySql	PostgreSQL	PostgreSQL	MySql
Linguagem de programação	PHP	Java	Python	PHP
<i>Template engine</i>	Blade	JSP	Django	?
Padrão de arquitetura	MVC	MVC	MVC	?

Características	SCPF	SAMMS	SCAE	GD-PNAES
<i>Framework backend</i>	Laravel	Não	Django	?
<i>Framework frontend</i>	<i>Bootstrap</i>	JSF	<i>Bootstrap</i>	?
Comunicação com outros sistemas	Não	Sim	Sim	Não
Proposta do sistema	Manutenção dos auxílios: automatizar o processo de verificação das frequências dos beneficiários	Seleção de beneficiários: automatizar o processo de seleção de beneficiários	Seleção de beneficiários: automatizar o processo de seleção de beneficiários	Seleção de beneficiários: automatizar o processo de seleção de beneficiários

Fonte: Próprio autor

3 PROCESSO DE NEGÓCIO DE PAGAMENTO DOS AUXÍLIOS

Para que possam existir os processos de pagamentos dos auxílios estudantis, há a necessidade do processo de seleção dos beneficiários. Os processos de seleção são semestrais. Atualmente os resultados da seleção são adicionados em planilhas e por meio delas é realizado o controle dos beneficiários.

A frequência em aula é requisito indispensável para a manutenção dos pagamentos dos auxílios. Cabe aos servidores do setor de assistência estudantil realizarem mensalmente a verificação das frequências dos beneficiários.

Os servidores devem gerar os relatórios das frequências dos alunos nos sistemas acadêmicos da instituição, os relatórios trazem as frequências de todos os alunos, seja beneficiário ou não. Os servidores então se dividem e realizam a validação das frequências, identificando os beneficiários na lista de frequências e cruzando as informações com a lista de controle de beneficiários dos auxílios.

Para casos de beneficiários em situação de infrequência, criam outra planilha contendo estes beneficiários, a partir disto, as coordenações dos cursos são notificadas, para que informem ao setor de AE se há justificativas para a infrequência dos respectivos beneficiários. Caso haja justificativas, estas são arquivadas para controle e os beneficiários passam a estarem aptos para receber o auxílio. Desta forma é criada uma planilha com os beneficiários aptos para que seja enviada ao setor financeiro. O processo Figura 1, leva em média, quatro dias.

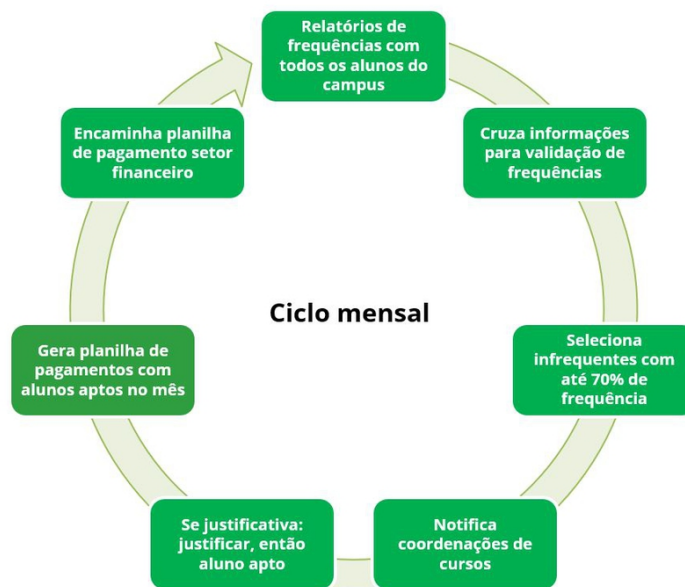


Figura 1 – Fluxo do processo de pagamento dos auxílios.

Fonte: Próprio autor.

A proposta é automatizar a tarefa de processamento e verificação de frequências, pois esta se mostrou consideravelmente onerosa durante o processo de pagamentos dos auxílios. Com isso, espera-se uma significativa redução do tempo despendido nos processos de pagamentos dos

auxílios estudantis. Visando beneficiar, tanto os servidores, quanto os beneficiários dos auxílios estudantis.

Para automatizar a tarefa de verificação das frequências, ambiciona-se o desenvolvimento de um sistema que possibilite aos servidores do setor de AE, realizarem a submissão dos relatórios de frequências, possibilitando a submissão de múltiplos arquivos. O sistema deverá cruzar as informações contidas nos relatórios de frequências dos alunos, com as informações dos beneficiários cadastrados no banco de dados. Com isso, verificar se há algum aluno infrequente que seja beneficiário, caso não haja, será possível ao servidor gerar a guia de pagamento com as informações dos beneficiários aptos ao recebimento do auxílio e enviá-la para o setor financeiro.

Para casos de infrequência, será possível ao servidor selecioná-los e notificar as respectivas coordenações, para que, verifiquem se existem justificativas. Havendo justificativas, estas serão submetidas ao sistema para armazenamento na base de dados, relacionadas aos respectivos beneficiários para o controle. Os beneficiários que tiverem a infrequência justificada passam da situação de não aptos, para a situação aptos ao recebimento dos auxílios.

4 SOLUÇÃO CONCEITUAL

Para o desenvolvimento de um sistema é necessário que sigamos alguns conceitos básicos:

A engenharia de software engloba processos, métodos e ferramentas que possibilitam a construção de sistemas complexos baseados em computador dentro do prazo e com qualidade. O processo de software incorpora cinco atividades estruturais: comunicação, planejamento, modelagem, construção e emprego; e elas se aplicam a todos os projetos de software. A prática da engenharia de software é uma atividade de resolução de problemas que segue um conjunto de princípios básicos. (PRESSMAN, 2011)

4.1 Levantamento e análise dos requisitos

Nas seções a seguir estão dispostos os artefatos gerados durante a fase de elicitação do sistema, desta forma percebe-se a importância das atividades estruturais.

4.1.1 Diagrama de casos de uso.

Para melhor compreensão das interações dos usuários com o sistema, elaborou-se a representação através do diagrama de casos de uso Figura 2.

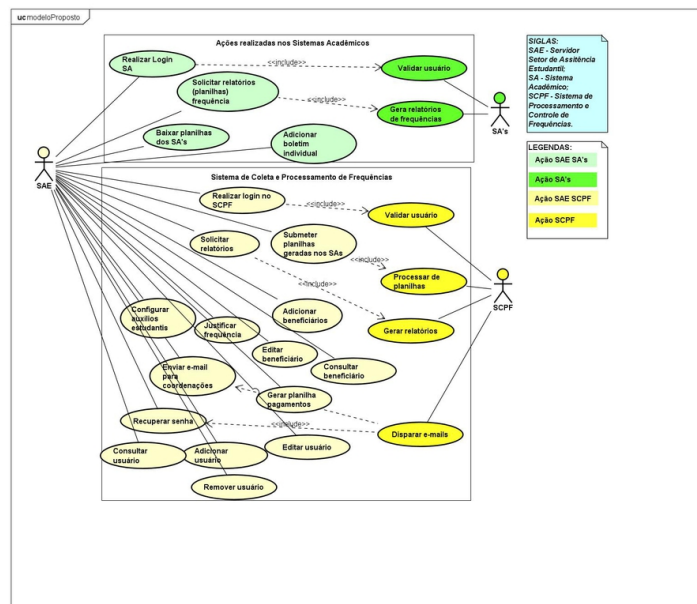


Figura 2 – Diagrama casos de uso.

Fonte: Próprio autor.

4.1.2 Descrição detalhada de casos de uso

Detalhamento do caso de uso “Realizar login” Figura 3. Demais casos de uso Apêndice A.

SAE (Servidor Assistência Estudantil)
SCPF (Sistema Controle e Processamento de Frequências)

Nome do Caso de Uso	Realizar Login
Resumo (Objetivo)	SCPF deve permitir que SAE previamente cadastrado acesse o sistema e suas funcionalidades
Atores	SAE, SCPF
Pré-condições	Necessário SAE ter cadastro no SCPF
Fluxo Principal	P1. SCPF apresenta a tela para “realizar login” P2. SAE informa email e senha P3. SAE clica botão “Entrar” P4. SCPF “valida SAE” P5. SCPF exibe página de listar beneficiários
Fluxo Alternativo	FA1. No P1. SAE com sessão ativa, SCPF retorna P5 FA2. No P2. SAE informa dados inválidos a) SCPF exibe mensagem “Dados incorretos” b) SCPF retorna P1 FA3. No P2. SAE não informa dados obrigatórios a) SCPF exibe mensagem “Dados obrigatórios não foram informados” b) SCPF retorna P1 . FA4. No P4. SCPF não encontra dados do SAE. a) SCPF exibe mensagem “SAE não cadastrado na base de dados” b) SCPF retorna P1 FA5. No P2. SAE não lembra senha a) SAE clica no botão “recuperar senha” b) SCPF exibe tela recuperar senha c) SAE informa email cadastrado d) SCPF envia senha aleatório para email do SAE e) SCPF retorna P1
Pós-condições	- SAE deve informar dados previamente cadastrados no base de dados do SCPF - Caso SAE informe dados válidos SCPF deve autenticar SAE e apresentar tela de listagem de beneficiários pois essa é a tela principal do SCPF - Caso o SAE informe dados inválidos ou não informe nenhum dados o SCPF deve exibir uma mensagem para o SAE que não deve sair da tela de login - Caso o SAE não lembre sua senha, SCPF deve permitir que usuário recupere a senha
Requisitos Não Funcionais	RNF1. Em caso de recuperação de senha, deve ser enviada para o email cadastrado do SAE
Autor	Adriano Prado Cavalheiro
Data	12/03/2021
Regras de Validação	- Dados não nulos - Dados correspondem cadastro na base de dados
Observações	

Figura 3 – Detalhamento caso de uso Realizar login.

Fonte: Próprio autor.

4.1.3 Prototipação de telas

Para validação do conceito da solução proposta, foi criado um protótipo dinâmico em HTML (A.2), o que proporcionou a melhor compreensão das expectativas dos usuários para com o sistema.

A seguir é possível conferir algumas das imagens do protótipo Figura 4, para compreensão do fluxo proposto pelo sistema.

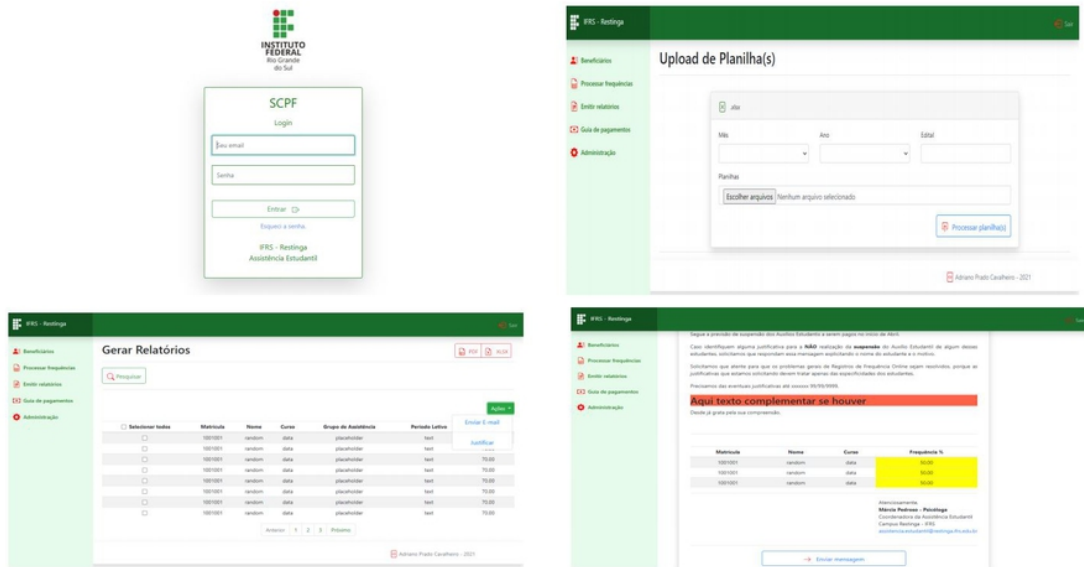


Figura 4 – Protótipos em HTML.
 Fonte: Próprio autor.

4.1.4 Diagrama de classes

Com a validação dos casos de uso e protótipo, foi possível realizar a modelagem das classes de modelo que comporão o sistema. Conforme apresentado na Figura 5.

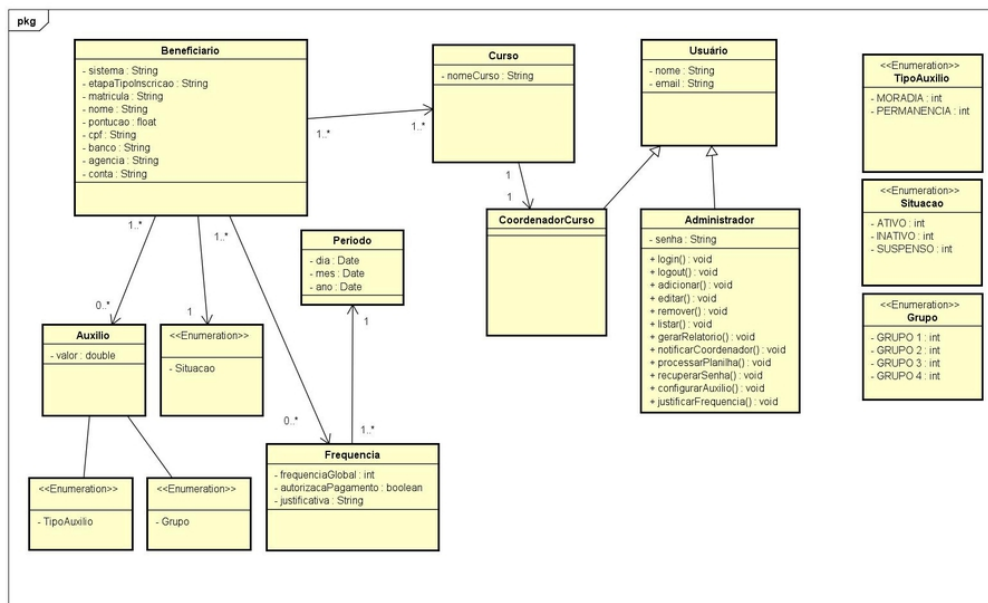


Figura 5 – Diagrama de classes.
 Fonte: Próprio autor.

4.1.5 Diagrama entidade relacionamento.

Com base na modelagem de classes, foi elaborado o seguinte diagrama de entidade e relacionamento Figura 6, que possibilita a visualização dos relacionamentos entre as tabelas do banco de dados.

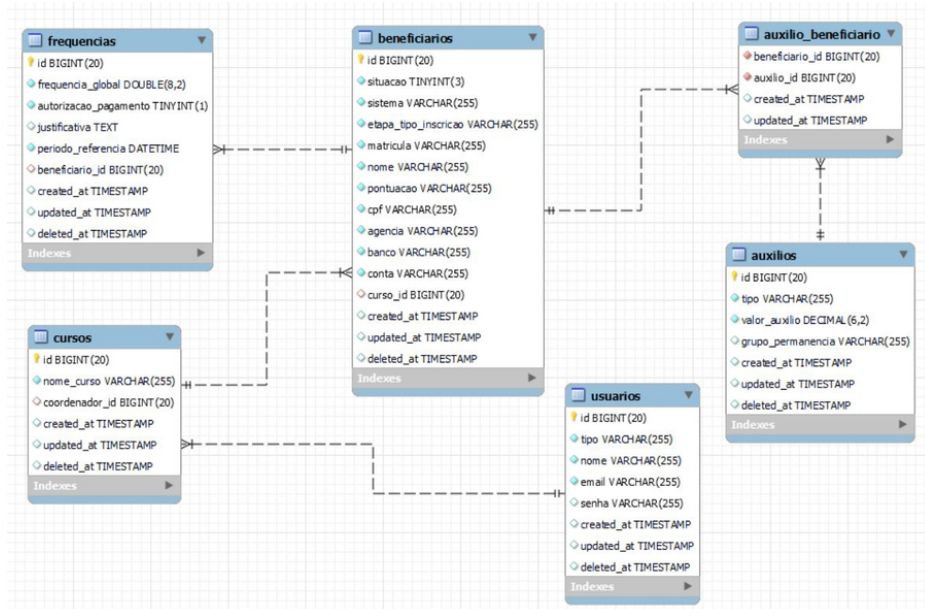


Figura 6 – Diagrama Entidade Relacionamento.

Fonte: Próprio autor.

Até o final desta etapa, foram empregadas três das cinco atividades estruturais citadas por Pressman: comunicação, atividades de planejamento e modelagem.

Os resultados destas atividades da estruturação, serviram como base para a fase de desenvolvimento, assunto da próxima seção. Já as atividades de construção e emprego, serviram como norte até a conclusão do projeto.

5 DESENVOLVIMENTO

O cenário atual foi decisivo para a escolha da plataforma da aplicação. Devido a necessidade do distanciamento social como medida de controle da pandemia de COVID-19 (AQUINO *et al.*, 2020), a modalidade de trabalho remoto tem ganhado espaço. Portanto, optou-se pela arquitetura *web* Figura7, para desenvolvimento deste projeto. Por se tratar de uma aplicação servida de forma centralizada, onde basicamente, o usuário precisa de um computador com acesso à *Internet*³ e um navegador para realizar suas tarefas.

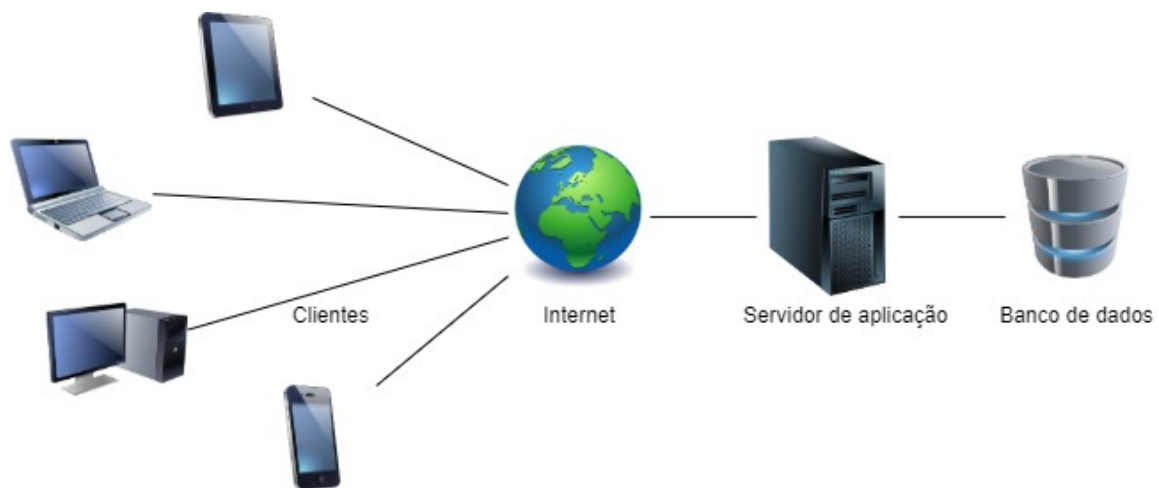


Figura 7 – Arquitetura cliente servidor.

Fonte: Próprio autor.

O desenvolvimento de sistemas envolve uma série de tecnologias. As principais tecnologias que compõem o sistema deste projeto podem ser conferidas no Apêndice A.

A escolha da linguagem de programação (A.1) PHP (A.1.1) para o desenvolvimento do sistema, se deu após consulta ao setor de Tecnologia da Informação do IFRS - Campus Restinga. Informaram que os sistemas disponíveis no Campus são, em sua maioria, em PHP. Portanto, a escolha objetivou adequar o sistema à estrutura do Campus visando facilitar a implantação.

Para agregar ao desenvolvimento boas práticas, paradigma de orientação a objetos, padrões de projeto, segurança, estilização, e eficiência, optou-se pela utilização de *frameworks* (A.6), o que facilitou em muito a parte de codificação. O *Laravel* (A.6.2), é responsável pelo provimento do *back-end* da aplicação, conta também com a biblioteca de *templates Blade* (A.6.2.1), para tratamento de dados no *front-end*. O *Bootstrap* (A.6.1), responsável pela estilização das páginas da aplicação, trata-se de um *framework front-end*. A estrutura dos principais *frameworks* do projeto Figura 8.

³ <https://www.w3.org/People/Frystyk/thesis/TcpIp.html>

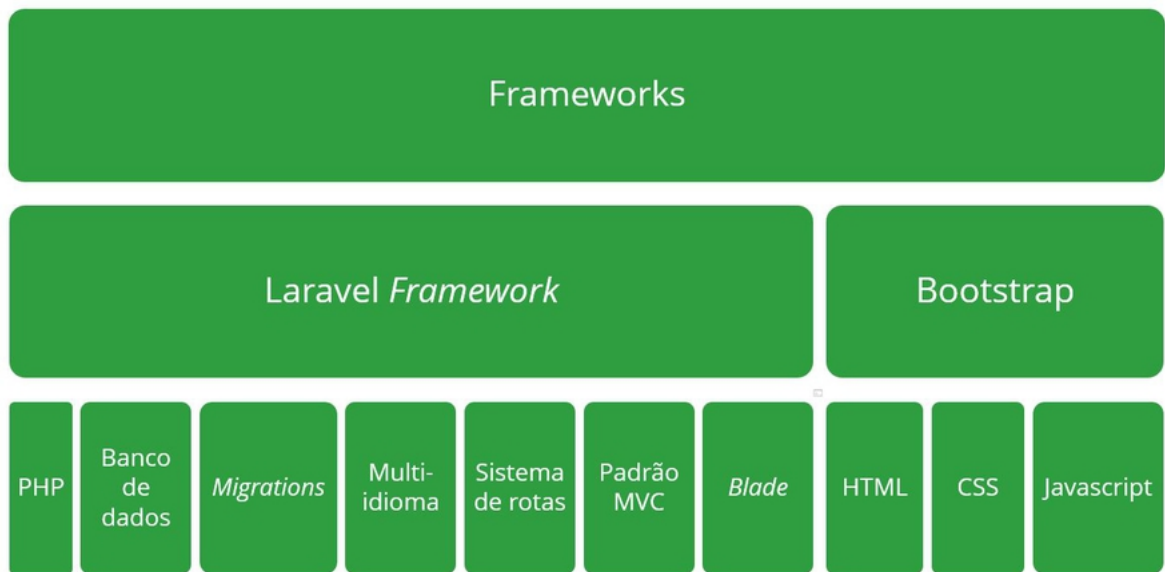


Figura 8 – Estrutura de *frameworks*.

Fonte: Próprio autor.

Estruturação das pastas e arquivos do código fonte do projeto Figura 9: na pasta *app* ficam as classes de modelo e controle, além de outras estruturas importantes para o *back-end*; na *resources* os arquivos referentes a parte de visualização; a exemplo da pasta *templates* que fica dentro da *resources*, nesta pasta por convenção ficam os arquivos que são compartilhados entre todas as páginas do sistema, assim é possível reaproveitar código através da inclusão de arquivos noutras páginas, isso é possível com a utilização do *Blade*. Ainda na mesma figura; é possível perceber a pasta *routes*, nela se encontra o arquivo *web.php*, responsável por comportar as rotas da aplicação.

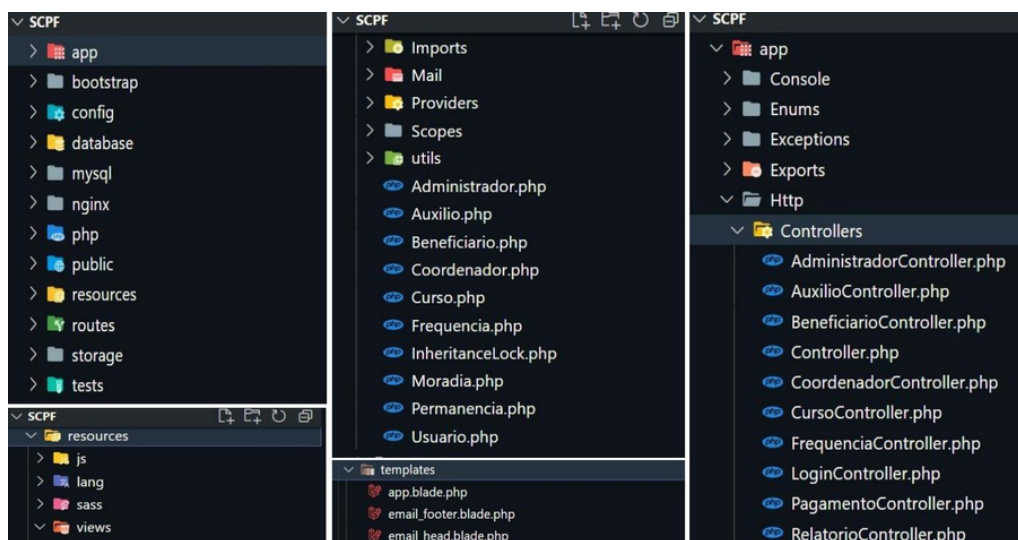


Figura 9 – Estrutura de pastas *Laravel*.

Fonte: Próprio autor.

A estrutura de pastas fornecidas pelo *Laravel*, permite a organização do código a fim de implementar o padrão de projeto MVC (*Model-View-Controller*) (A.5), padrão este, que visa a separação de responsabilidades em camadas Figura 10.

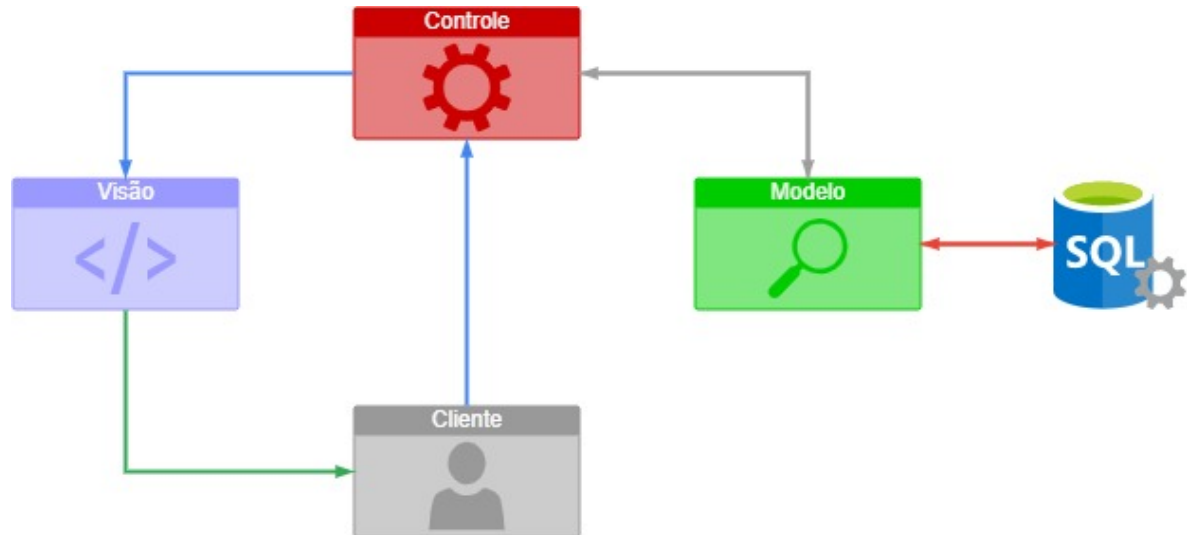


Figura 10 – Padrão MVC.

Fonte: Próprio autor.

5.1 Implementação da Camada de Modelo

Na sequência são apresentados fragmentos da codificação, com a finalidade de exemplificar o fluxo de desenvolvimento do sistema, com o emprego do *Laravel* (A.6.2).

Fragmento da classe de migração “*CreateUsuariosTable*” Figura 11. Esta classe permite que informemos ao *Eloquent ORM* (A.6.2.2), quais os tipos de dados que serão salvos na base de dados e os relacionamentos entre as entidades. Com sua utilização, é possível o versionamento da base de dados, através de *Migrations* (A.6.2.3).

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsuariosTable extends Migration
{
    public function up()
    {
        Schema::create('usuarios', function (Blueprint $table) {
            $table->id();
            $table->string('tipo');
            $table->string('nome');
            $table->string('email')->unique();
            $table->string('senha')->nullable();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('usuarios');
    }
}

```

Figura 11 – Migração Usuário.
 Fonte: Próprio autor.

Fragmento da classe de modelo Administrador Figura 12. Desta forma, representamos a tabela da base de dados através de uma classe PHP.

```

<?php

namespace App;

class Administrador extends Usuario
{
    use InheritanceLock;

    protected $table = 'usuarios';

    protected $fillable = [
        'nome', 'email', 'senha',
    ];
}

```

Figura 12 – Modelo Administrador.
 Fonte: Próprio autor.

5.2 Implementação da Camada de Controle

O fragmento Figura 13, faz parte do arquivo responsável pelo roteamento da aplicação. As rotas fazem a ligação e transporte de dados entre as camadas da aplicação, elas permitem

que a camada de visualização realize uma consulta na base de dados. Desta forma, a camada de modelo trata a requisição e o controle fica a cargo de retornar o resultado, devolvendo os dados requisitados para a camada de visualização.

```
<?php
use Illuminate\Support\Facades\Route;

Route::get('/', ['as' => 'login', 'uses' => 'LoginController@index']);
Route::post('/validar_usuario', ['as' => 'validar_usuario', 'uses' => 'LoginController@validarLogin']);
Route::get('/usuario_logout', ['as' => 'usuario_logout', 'uses' => 'LoginController@realizarLogout']);
Route::get('/recuperar_senha', ['as' => 'recuperar_senha', 'uses' => 'AdministradorController@exibirPaginaRecuperarSenha']);
Route::post('/solicitar_senha', ['as' => 'solicitar_senha', 'uses' => 'AdministradorController@criarNovaSenha']);
```

Figura 13 – Rotas de autenticação.

Fonte: Próprio autor.

Fragmento da classe controladora “AdministradorController” Figura 14. Podemos ver o método *index*, que tem a função de realizar uma consulta na base de dados e devolver para a *view index.blade.php*, localizada na pasta “resources/usuarios/administrador”. Perceba no método, como é realizada a consulta no banco de dados. Em uma linha, o *Eloquent* nos devolve uma lista de administradores paginada e ordenada sem a utilização de código SQL⁴ (*Structured Query Language*). O *Eloquent* tem ainda, uma série de métodos que nos permite lidar com o banco de dados de maneira bastante prática, em relação às consultas com código SQL.

```
<?php

namespace App\Http\Controllers;

use App\Administrador;
use App\Mail\EmailRecuperarSenha;
use App\Usuario;
use App\Utils\NovaSenha;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Facades\Mail;

class AdministradorController extends Controller
{
    public function index()
    {
        $usuarios = Administrador::orderBy('nome', 'ASC')->paginate(10);
        return view('usuarios.administrador.index', compact('usuarios'));
    }
}
```

Figura 14 – Controle Administrador.

Fonte: Próprio autor.

⁴ <https://www.w3schools.com/sql/>

5.3 Implementação da Camada de Visualização

Representação da *view index*, responsável por listar os administradores da aplicação Figura 15. Neste arquivo é tratada a lista de administradores, originada com a consulta à base de dados. No código é possível perceber a utilização do *Bootstrap* (A.6.1), através de suas classes. Com a utilização do *Blade* (A.6.2.1) é possível realizar uma interação do tipo *foreach* na lista administradores, percebe-se também a utilização de estruturas condicionais. Desta forma, não há “poluição” de código PHP em nossa camada de visualização.

```
@extends('templates.app')
@section('content')
    <div class="d-flex justify-content-between flex-wrap flex-md-nowrap align-items-center pt-3 pb-2 mb-3 border-bottom">
        <h1 class="h2">Administrador(es)</h1>
        <div class="btn-toolbar mb-2 mb-md-0">
            <a href="{{ route('admins.create') }}" class="btn btn-md btn-outline-primary">
                <i class="bi bi-plus"></i>&nbsp;
                Adicionar administrador
            </a>
        </div>
    </div>
    <br>
    @include('includes.alertas')
    @if (!count($usuarios) > 0)
        <div class="alert alert-warning text-center" role="alert">
            <strong>Não há registro na base de dados!</strong>
        </div>
    @else
        <div class="row" style="margin-bottom: 5%;">
            @foreach ($usuarios as $usuario)
                <div class="col-sm-6" style="margin-top: 15px;">
                    <div class="card">
                        <div class="card-body">
                            <h5 class="card-title">{{ $usuario->nome }}</h5>
                            <p class="card-text">{{ $usuario->email }}</p>
                            @if ($usuario->id === session('usuario_logado')['id'])
```

Figura 15 – Index Administrador.

Fonte: Próprio autor.

Mais uma demonstração da utilidade do *Blade* para o desenvolvimento Figura 16. Aqui temos a representação de como foi empregada a estrutura de *templates* do *Blade*.

```

1 @include('templates.head')
2
3 @include('templates.nav-top')
4
5 @include('templates.nav-side')
6
7 @yield('content')
8
9 @include('templates.footer')
10

```

```

1 <nav class="navbar navbar-dark st
2 <a class="navbar-brand col-m
3 
9 <ul class="navbar-nav px-3">
10 <li class="nav-item text-
11 <a class="nav-link" h
12 </li>
13 </ul>
14 </nav>
15

```

```

1 <div class="container return">
2 <div class="row align-items-c
3 <div class="col-4">
4 <a type="button" titl
5 <i class="bi bi-a
6 </a>
7 </div>
8 <div class="col-8 ">
9 <p class="font-weight
10 title="Visita
11 Assistência B
12 </p>
13 </div>
14 </div>
15 </main>
16 </div>
17 </div>
18
19 <script src="{{ asset('js/app.js
20 <script>
21 $(document).ready(function()
22 $("#filtro_busca").on("ke
23 var value = $(this).v
24 $(".filtro_tabela tr
25 $(this).toggle($
26 });
27

```

Figura 16 – Estrutura de *templates Blade*.

Fonte: Próprio autor.

Ainda na fase de desenvolvimento, foram implementados métodos de notificação por email, *upload*, processamento de arquivos em massa e relatórios nas extensões, PDF e XLSX.

5.4 Execução do sistema

Nesta seção é apresentado o cadastro de beneficiários eo fluxo principal do sistema. Ou seja, a exemplificação do processo de validação de frequências. O sistema conta ainda com cadastro e gerência de administradores do sistema, cursos, coordenações e auxílios.

Tela de *login* Figura 17, para acesso ao sistema;

Figura 17 – Tela *login*.

Fonte: Próprio autor.

Após o primeiro acesso, o administrador deverá realizar os cadastros de auxílios, cursos e coordenadores, utilizando os formulários apresentados na Figura 18;

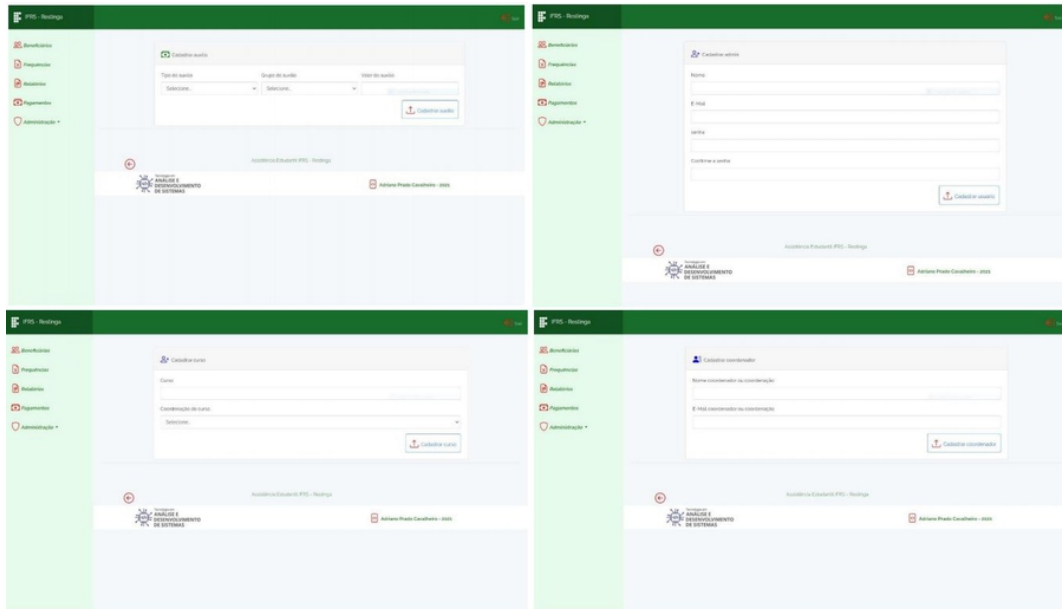


Figura 18 – Telas para cadastros.

Fonte: Próprio autor.

Listagem de cursos, auxílios, coordenadores e administradores Figura 19;

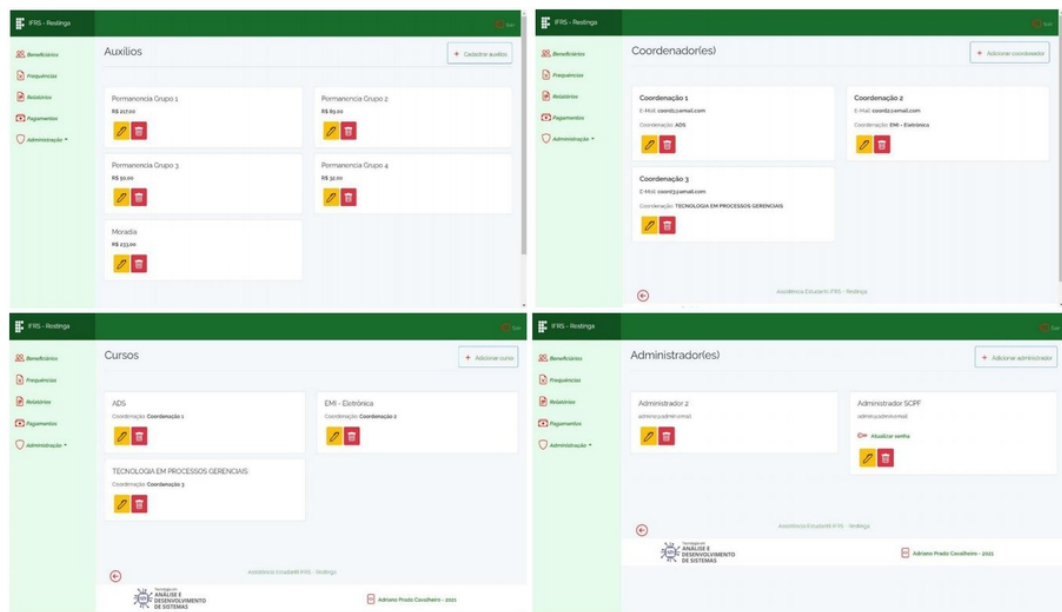


Figura 19 – Telas de listagem e gerência.

Fonte: Próprio autor.

Cadastro de beneficiários, este processo será realizado a cada nova seleção de beneficiários. O cadastramento pode ser realizado de duas maneiras: em lote, através de *upload* de planilhas ou com formulário para cadastro Figura 20;

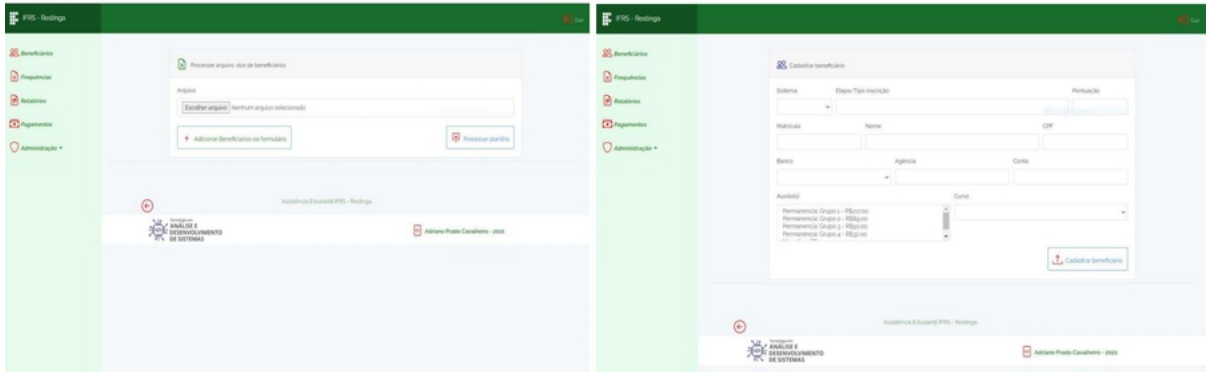


Figura 20 – Telas para cadastros de beneficiários.

Fonte: Próprio autor.

A tela de listagem dos beneficiários Figura 21, contém botões para ações de alterar a situação, detalhar as informações e para a edição de beneficiários. Ainda nesta tela, é possível visualizar o botão para cadastro de beneficiários e os filtros para pesquisa.

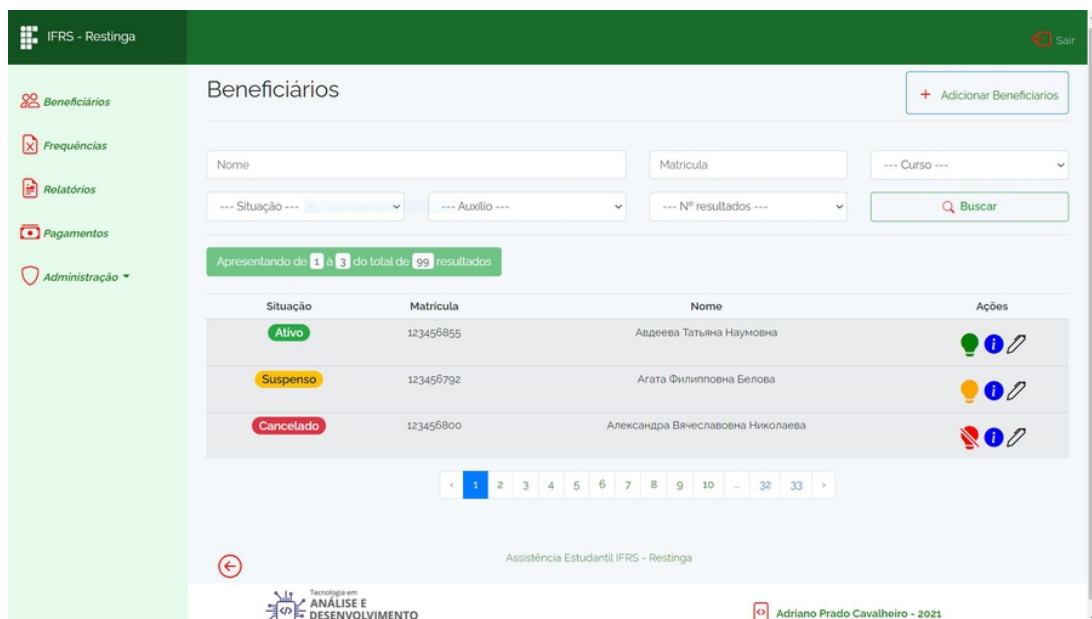


Figura 21 – Tela de listagem de beneficiários.

Fonte: Próprio autor.

O usuário de posse dos relatórios das frequências, gerados nos sistemas acadêmicos, selecionará no menu à esquerda o item “Planilhas” e com isso será apresentada a seguinte tela Figura 21, onde poderá realizar o *upload* de múltiplos arquivos para o devido processamento;

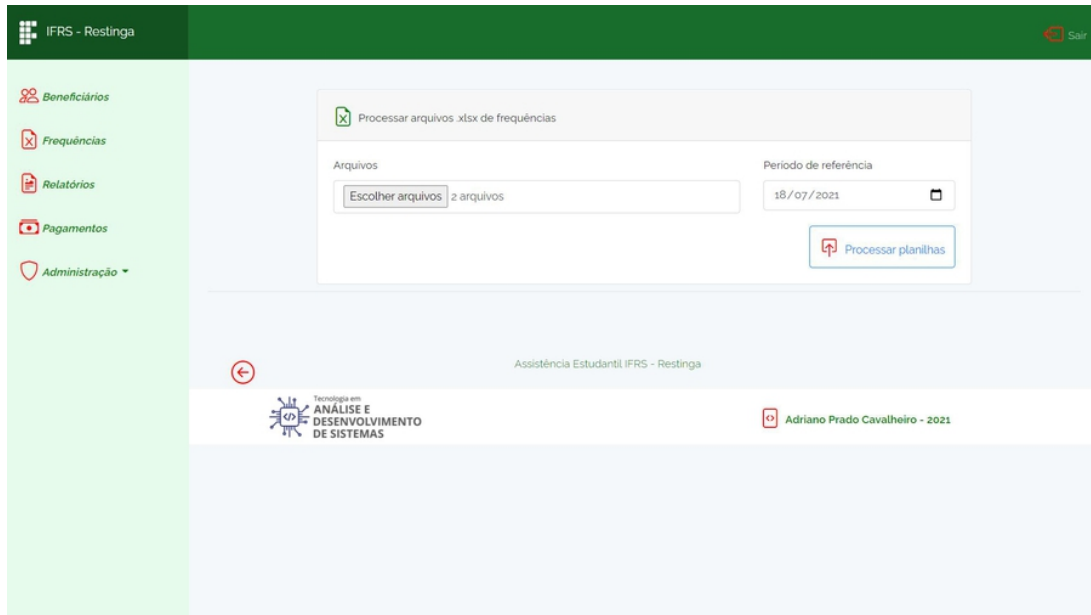


Figura 22 – Tela para upload das planilhas de frequências.

Fonte: Próprio autor.

Após o *upload* das planilhas, o sistema deverá realizar o processamento, identificando entre estudantes, aqueles que recebem benefícios, verificará também entre estes, se há infrequentes. Identificados casos de beneficiários em situação de infrequência, estes terão a sua situação alterada pelo sistema para “Suspenso” e a autorização de pagamentos alterada para “Não autorizado”, conforme Figura 23.

O usuário poderá selecionar beneficiários em situação de infrequência, filtrando por cursos, para que a respectiva coordenação seja notificada, o usuário deverá realizar a seleção dos beneficiários e no botão ações, escolher opção “Enviar E-mail”;

Período	Situação	Sistema	Curso	Matricula	Nome	CPF	Frequência	Ações
08/2021	Suspenso	SIGAA	TECNOLOGIA EM ELETRÔNICA INDUSTRIAL	123456860	Блинова Вера Мироновна	30351007210	24%	Enviar E-mail, Justificar, Justificativa
08/2021	Suspenso	SIGAA	TECNOLOGIA EM ELETRÔNICA INDUSTRIAL	123456851	г-жа Артемьева Ксения Васильевна	96572884520	57%	Enviar E-mail, Justificar, Justificativa
08/2021	Suspenso	SIGAA	TECNOLOGIA EM ELETRÔNICA INDUSTRIAL	123456861	Осипов Вячеслав Изотович	67625177300	40%	Enviar E-mail, Justificar, Justificativa
08/2021	Suspenso	SIGAA	TECNOLOGIA EM ELETRÔNICA INDUSTRIAL	123456853	Суворов Андроник Демьянович	30342757903	36%	Enviar E-mail, Justificar, Justificativa
08/2021	Suspenso	SIGAA	TECNOLOGIA EM ELETRÔNICA INDUSTRIAL	123456850	Фрол Антипович Волков	76575473326	16%	Enviar E-mail, Justificar, Justificativa

Figura 23 – Relação de infrequentes.

Fonte: Próprio autor.

Para notificar as coordenações, o usuário deverá selecionar a coordenação desejada, o respectivo assunto e opcionalmente complementar o texto padrão da notificação, com o preenchimento do campo complemento.

Clicando no botão “Prévia”, pode-se conferir as informações da notificação, estando estas condizentes com o desejado clicar no botão “Enviar mensagem” Figura 24;

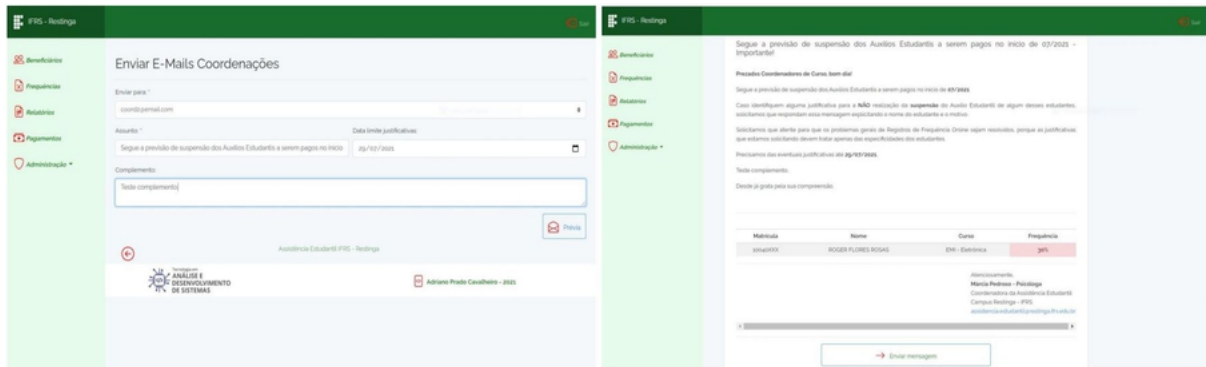


Figura 24 – Telas para notificações.

Fonte: Próprio autor.

Após o retorno das coordenações, informando se há justificativas para as infrequências, o usuário poderá selecionar todos aqueles com justificativa e no botão ações, selecionar a opção “Justificar”. Será apresentada a seguinte tela Figura 25, onde o usuário deverá informar as justificativas e clicar no “botão Justificar”;

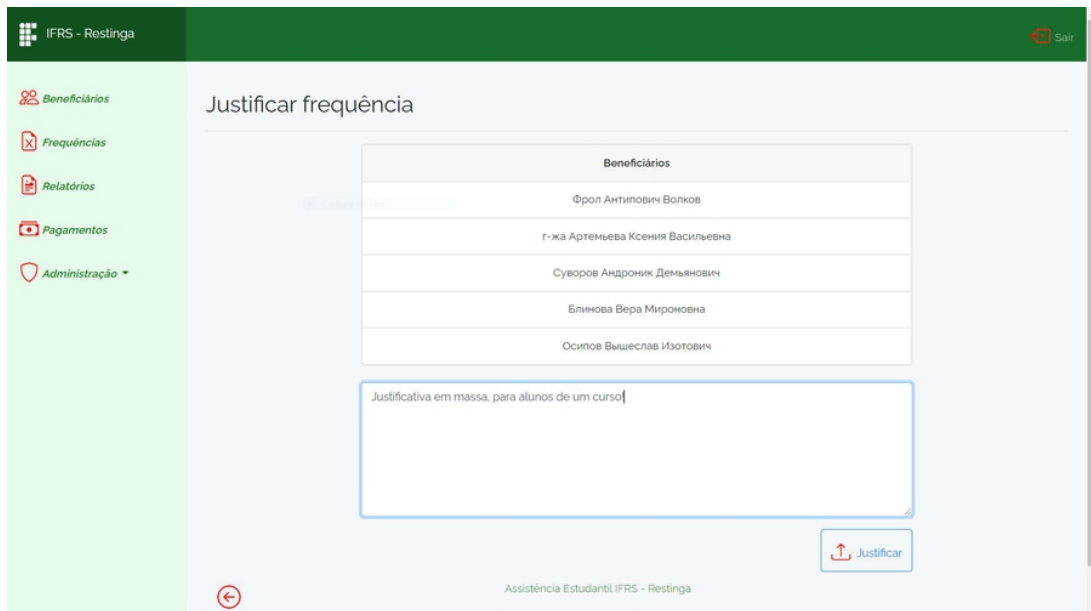


Figura 25 – Tela para justificativa.

Fonte: Próprio autor.

Aqueles beneficiários que tiverem a infrequência justificada, passam para a situação de pagamento autorizado, ficando o registro da justificativa informada de acordo com o mês referente ao benefício Figura 26;

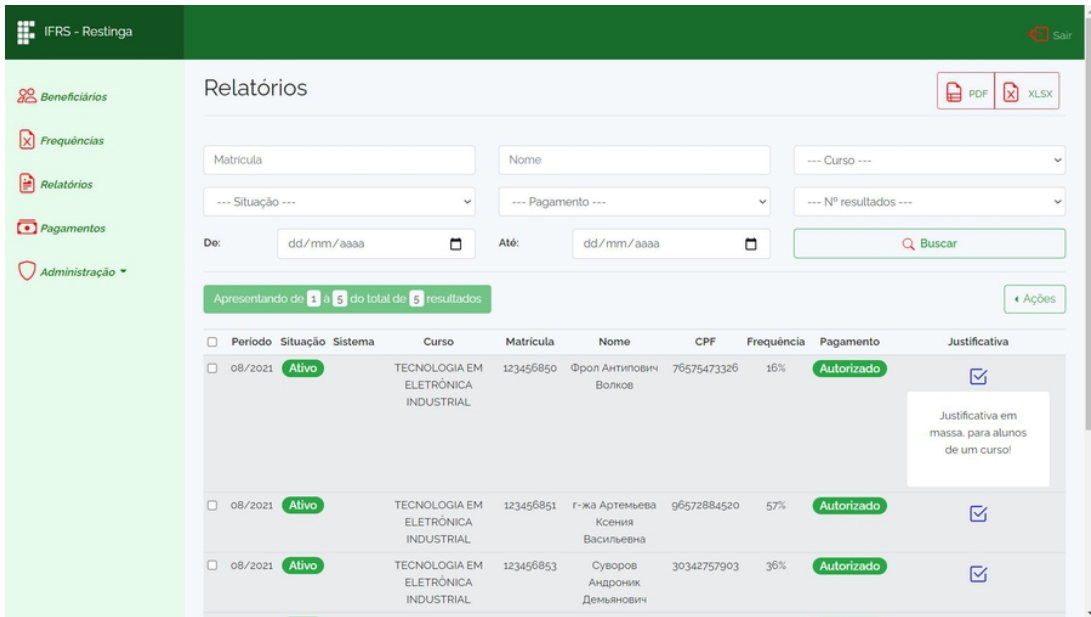


Figura 26 – Alterada situação para pagamento autorizado.

Fonte: Próprio autor.

A partir deste processo será possível realizar a verificação dos beneficiários aptos ao recebimento dos auxílios, conforme Figura 27, sendo este também o passo seguinte para casos onde não existirem beneficiários infrequentes, após o processamento das frequências;

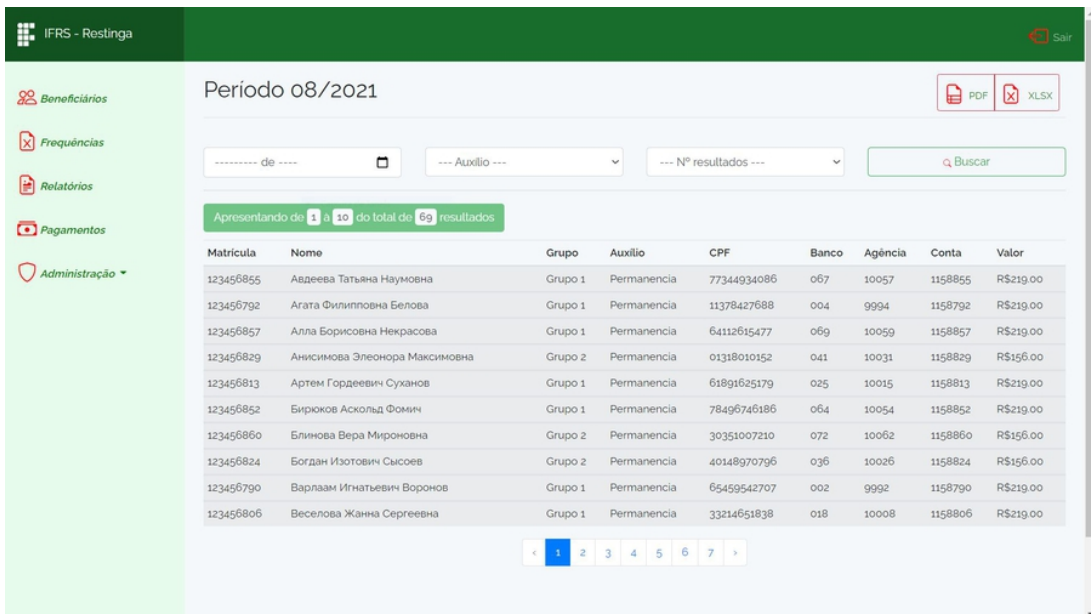


Figura 27 – Tela de listagem de beneficiários aptos a receber auxílio.

Fonte: Próprio autor.

Finalizado o processo de validação de frequências dos beneficiários, passo importante para o processo de pagamento dos auxílios estudantis, o usuário poderá gerar a planilha de pagamentos do mês referência, selecionando o “botão XLSX” para ser enviado ao setor financeiro. Finalizando a atuação dos servidores do setor de AE no processo.

6 TESTES

A fim de corrigir *bugs*, realizar ajustes e validar o sistema, foram promovidos testes com os usuários, ao todo três sessões de testes foram executadas com a participação da coordenadora do setor de AE.

O sistema foi disponibilizado em ambiente para testes, onde a usuária recebeu orientações para acesso e livre utilização. Ao final de cada seção, foram realizados encontros remotos para a coleta de informações. Desta forma, foi possível realizar ajustes de maneira eficiente e assertiva.

Para a terceira e última seção de testes, foi elaborado um roteiro composto por questões e tarefas, objetivando orientar durante a utilização do sistema e coletar as impressões da usuária e sua avaliação.

- **Exemplo de tarefa:**

Página: Geral

Objetivo: Entender se o objetivo do sistema está claro

Descrição da tarefa: **Navegue por alguns minutos pelo site, para conhecer o ambiente.**

Você sabe explicar sobre o que se trata este site?

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais.

- **Exemplo de questão:**

Quão satisfeito você ficou com o sistema?

- Muito satisfeito
- Nada satisfeito
- Satisfeito

Com base nas respostas da usuária, foi elaborado o seguinte gráfico Figura 28.



Figura 28 – Impressões gerais.

Fonte: Próprio autor.

O teste teve duração de duas horas e trinta minutos, realizado remotamente. A usuária pôde testar o sistema utilizando dados reais e assim realizar avaliação efetiva, sobre a usabilidade e o comportamento do sistema durante o processamento das planilhas. Ao todo foram cadastrados 423 beneficiários via planilha e processados mais de 800 registros contidos nas planilhas geradas nos sistemas acadêmicos, para a verificação das frequências. O documento resultante do teste está disponível no Anexo A.

7 CONCLUSÃO

O Sistema de Coleta e Processamento de Frequências para Pagamento dos Auxílios Estudantis permite aos servidores da AE executar a tarefa de verificação das frequências dos beneficiários, que atualmente despende tempo e recursos consideráveis, em poucos segundos. permite também controle e eficiência na gerência dos auxílios estudantis.

Com base na avaliação dos usuários, durante a fase de testes, conclui-se que o sistema tem potencial para proporcionar aos servidores da AE maior eficiência no processo de pagamentos dos auxílios estudantis.

Para trabalhos futuros: integração com os sistemas de Inscrições de Auxílios Estudantis, SIA, SIGAA. inclusão de dashboard, exportar relatórios para arquivo .doc, opção para remover e editar justificativas, podem agregar maior valor ao sistema.

A elaboração deste estudo de caso, foi possível não só, com o emprego dos conhecimentos adquiridos no decorrer do curso, mas também com o aprimoramento da capacidade de resolver problemas, sendo este o cotidiano na área da tecnologia da informação. Portanto, é possível concluir que o curso de Tecnologia em Análise e Desenvolvimento de Sistemas, prepara profissionais para desenvolver sistemas e os capacita, apresentando-lhes os meios e os incentivando na busca por soluções.

REFERÊNCIAS

ANAIS DO WORKSHOP DE INFORMÁTICA NA ESCOLA, XXI., out. 2015, [S.l.].

Impactos da implantação do sistema informatizado para processo de seleção do PNAES no IFTO Campus Porto Nacional. ISSN 2316-6541. Disponível em: <https://br-ie.org/pub/index.php/wie/article/view/5084>. Acesso em: 22 mar. 2021.

AQUINO, Estela M. L. *et al.* Medidas de distanciamento social no controle da pandemia de COVID-19: potenciais impactos e desafios no Brasil. **Ciência & Saúde Coletiva**, ABRASCO - Associação Brasileira de Saúde Coletiva, v. 25, n. 16, p. 2423 – 2446, 06 2020. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-81232020006702423&lng=en&nrm=iso.

BRASIL, MINISTÉRIO DA EDUCAÇÃO. Resolução nº 086, de 03 de dezembro de 2013. Resolve aprovar. **A Política de Assistência Estudantil do IFRS**, 2013. Disponível em: https://ifrs.edu.br/wp-content/uploads/2017/09/Resolucao_86_13.pdf. Acesso em: 22 mar. 2021.

BRASIL, MINISTÉRIO DA EDUCAÇÃO. **Instituições da Rede Federal: Conheça as características das instituições que compõem a Rede Federal de Educação Profissional, Científica e Tecnológica e a lista de suas unidades.** 2018? Disponível em: <http://portal.mec.gov.br/rede-federal-inicial/instituicoes>. Acesso em: 22 mar. 2021.

BREDA, Helder. **SCAE: um sistema de gerenciamento de benefícios para a coordenadoria de assistência estudantil com o uso de formulários dinâmicos.** 2017. Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) — Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Farroupilha. Disponível em: <https://pergamum.ifrs.edu.br/pergamumweb/vinculos/000055/0000553a.pdf>. Acesso em: 22 mar. 2021.

CARDOSO, Alex Vargas. **Desenvolvimento do SAAMS-Sistema para assistência estudantil e acompanhamento acadêmico.** 2019? Monografia (Tecnologia em Sistemas para Internet) — Instituto Federal do Rio Grande do Sul Campus Porto Alegre. Disponível em: <http://atom.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/1/3/a/13ad9e126899fef031ad5d36d08edab61fda9bb1f514689a6e6ee25da5b749e7/desenvolvimento-saams-sistema-revisado-banca.pdf>. Acesso em: 22 mar. 2021.

DUTRA, Natália Gomes dos Reis; SANTOS, Maria de Fátima de Souza. Assistência estudantil sob múltiplos olhares: a disputa de concepções. **Ensaio: Avaliação e Políticas Públicas em Educação**, Fundação CESGRANRIO, v. 25, n. 94, p. 148 – 181, 03 2017. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-40362017000100148&lng=en&nrm=iso.

ESTRADA, Adrian Alvarez; RADAELLI, Andressa. A política de assistência estudantil em uma universidade pública; a perspectiva estudantil. **Revista on line de Política e Gestão Educacional**, Araraquara, n. 16, fev. 2017. ISSN 1519-9029. Disponível em: <https://periodicos.fclar.unesp.br/rpge/article/view/9359>. Acesso em: 22 mar. 2021.

GAMMA, Erich *et al.* **Padrões de Projetos: Soluções Reutilizáveis de software orientado a objeto.** São Paulo: Bookman, 2009.

IFRS. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL - CAMPUS RESTINGA. Assistência Estudantil. **Relatório Quantitativo da Assistência Estudantil**, Porto Alegre, 2021?a. Disponível em: <https://docs.google.com/spreadsheets/d/11QtD0exXIV6DwHsPjf2s2wJBMhXpQC3xps-ATkWoKSQ/edit#gid=1537681913>. Acesso em: 22 mar. 2021.

IFRS. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO SUL - CAMPUS RESTINGA. Assistência Estudantil: Perfil dos Beneficiários de Auxílio Estudantil. Porto Alegre, 2021?b. Disponível em: <https://ifrs.edu.br/restinga/wp-content/uploads/sites/5/2020/02/Perfil-dos-Benefici%C3%A1rios-de-Aux%C3%ADlio-Estudantil-IFRS-Campus-Restinga.pdf>. Acesso em: 22 mar. 2021.

MACEDO, Juliana Cavalcanti; ABRANCHES, Ana de Fátima Sousa. POLÍTICA DE ASSISTÊNCIA ESTUDANTIL: REPERCUSSÕES SOBRE A SUA CONTRIBUIÇÃO. **Jornal de Políticas Educacionais**, [S.l.], v. 12, n. 10, jun 2018. ISSN 1981-1969. Disponível em: <https://revistas.ufpr.br/jpe/rt/captureCite/58615>. Acesso em: 22 mar. 2021.

PRESSMAN, Roger S. **Engenharia de software**: Uma abordagem profissional. 7ª. ed. Porto Alegre: AMGH, 2011.

Apêndices

APÊNDICE A – DETALHAMENTO TECNOLOGIAS UTILIZADAS

A.1 Linguagem de Programação

Para o desenvolvimento de aplicações *web* são utilizadas linguagens de programação, ou seja, blocos de instruções lógicas que são executadas nos servidores de aplicações e são responsáveis pelo processamento de informações.

A linguagem de programação utilizada neste projeto é o PHP⁵ (*Hypertext Preprocessor*) em sua versão 7.2.

A.1.1 PHP

A linguagem PHP conta com suporte a Orientação a Objetos desde a sua versão 3 sendo aprimorada nas versões posteriores. Uma linguagem orientada a objetos possibilita o desenvolvimento de aplicações robustas, devido a possibilidade do reaproveitamento, segurança, manutenibilidade e elegância do código. A orientação a objetos é um paradigma de programação que permite a representação do mundo real em forma de objetos computacionais, esses objetos são provenientes de modelos, compostos por linhas de código, denominados classes. Além disso, o PHP é uma linguagem de código aberto e gratuita.

A.2 HTML

O HTML⁶ é a linguagem de marcação que é utilizada para a criação de documentos de hipertexto. É composta por *tags* que seguem um padrão e que envolvem o conteúdo que se deseja adicionar aos documentos. Quando requisitados estes documentos por meio da requisição de um usuário, o navegador é responsável por interpretar o documento HTML e apresentar o conteúdo na tela.

A.3 CSS

O CSS⁷ (*Cascading Style Sheets*) é a linguagem de estilos utilizada em conjunto com o HTML e serve para formatação de documentos de hipertexto e permitir que tenham um visual atraente para os usuários.

A.4 Javascript

Javascript⁸ é uma linguagem de programação que pode rodar no lado do cliente, com isso ela é muito utilizada para adicionar dinamicidade as páginas atuando em conjunto com HTML e CSS e permitindo uma melhor interação dos usuários com conteúdo *web*, além de desonerar o servidor da aplicação, pois com o Javascript é possível adicionar lógica e ações que não serão processadas pelo servidor, mas sim pelo navegador do usuário.

⁵ https://www.php.net/manual/pt_BR/langref.php

⁶ <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

⁷ <https://developer.mozilla.org/pt-BR/docs/Web/CSS>

⁸ <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

A.5 Padrões de Projeto

Padrões de projeto são empregados com o objetivo de facilitar o desenvolvimento de softwares orientados a objetos, por meio de arquiteturas modulares que diminuem a complexidade de projetos de *software*, agregando segurança e organização aos projetos, além de eficiência no desenvolvimento.

Neste projeto será utilizada a arquitetura MVC (*Model-View-Controller*), Modelo, Visão e Controle. Esta arquitetura permite separar o projeto em três camadas Figura 10, isolando as camadas de dados, regras de negócio e visualização umas das outras (GAMMA *et al.*, 2009).

A.5.1 Modelo

A camada de modelo é responsável pela parte lógica da aplicação, onde ficam as regras de negócio, essa camada também tem responsabilidade pela manipulação e acesso a dados.

A.5.2 Visão

A camada de visualização é responsável pela apresentação dos dados do sistema permitindo ao usuário a interação sobre essas informações e visualização.

A.5.3 Controle

A camada de controle tem por finalidade realizar a comunicação entre as partes de modelo e visualização, sendo responsável pela filtragem e tráfego dessas informações.

A.6 Frameworks

No desenvolvimento de *softwares* os *frameworks* são utilizados para tornar a tarefa de desenvolvimento mais prática e rápida, Têm como finalidade diminuir as configurações necessárias para se iniciar um projeto, além de prover estruturas que são comuns a vários projetos evitando a reescrita de códigos e configurações. Por exemplo, alguns *frameworks backend*, fornecem suporte a banco de dados, padrões de projeto, servidor de aplicação embutido para desenvolvimento, *templates*, entre outros.

Alguns *frameworks* de código aberto serão utilizados com o objetivo de otimizar o desenvolvimento desta aplicação.

A.6.1 Bootstrap

O Bootstrap⁹ é um *framework* que permite agilizar a criação da parte visual da aplicação, foi escolhido para este projeto pela facilidade de reuso de componentes e por proporcionar componentes que englobam HTML, CSS e Javascript, permitindo adicionar a aplicação estilo e ação com poucas linhas de código. Portanto, o Bootstrap nos fornece pacotes que podem facilmente ser adicionados ao código por meio de classes inseridas em *tags* HTML, proporcionando estilo, dinamismo e responsividade por meio de componentes reutilizáveis.

⁹ <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

A.6.2 Laravel

O Laravel¹⁰ é um *framework* que contém uma série de ferramentas e funcionalidades para a construção de aplicações *web* em linguagem PHP. Possibilita a escrita de código de forma otimizada, orientada a objetos e elegante. Provê em poucos minutos uma estrutura de projeto robusta, com arquitetura MVC, sem a necessidade de grandes configurações, permitindo que o desenvolvedor se preocupe apenas com as regras de negócio da aplicação. A seguir apresenta-se uma lista de algumas ferramentas que são provisionadas pela estrutura do Laravel e que serão utilizadas no projeto.

A.6.2.1 Blade

Blade¹¹ é um *template engine* que possibilita a inserção de dados do modelo sem a necessidade de misturar *tags* HTML com PHP, além de tornar o código limpo, também conta com estruturas condicionais e permite a modularização do código HTML em vários arquivos.

Antes das *templates engines* para apresentar dados vindos do *backend* em uma tabela era necessário a utilização de código *backend* em documentos HTML para criar a estrutura de repetição responsável por realizar a leitura dos dados e inserir nas *tags* HTML, com as *templates*, ainda fazemos uso de estruturas de repetição, mas com um código muito enxuto, as *templates* são utilizadas como se fossem uma camada de abstração sobre a linguagem de *backend*.

A.6.2.2 Eloquent ORM

O Eloquent ORM¹² (*Object-relational mapping*) é um *framework* que implementa o padrão *ActiveRecord* para banco de dados relacionais, permitindo a conversão de modelos de classe em tabelas no banco de dados, além disso fornece a aplicação uma série de funcionalidades para acesso e manipulação de dados. Esta manipulação pode ser realizada sem a utilização de SQL (*Structured Query Language*), que é uma linguagem estruturada específica para banco de dados relacional, podemos utilizar apenas código PHP devido aos métodos de manipulação de dados fornecidos pelo Eloquent.

A.6.2.3 Migrations

Outra funcionalidade provida pelo Laravel são as Migrations¹³, que oferecem a possibilidade de versionamento da base de dados da aplicação por meio de comandos executados por linha de comando.

A.6.2.4 Artisan Console

Artisan¹⁴ é uma ferramenta de linha de comando que permite a execução de uma série de comandos, permitindo facilidades tais como a criação de projetos Laravel, criação de migrações de base de dados, criação de modelos, controles, dentre outros, além de permitir a execução de um servidor *web* para o desenvolvimento da aplicação.

¹⁰ <https://laravel.com/docs/7.x/releases#laravel-7>

¹¹ <https://laravel.com/docs/7.x/blade#introduction>

¹² <https://laravel.com/docs/7.x/eloquent-resources#introduction>

¹³ <https://laravel.com/docs/7.x/migrations#introduction>

¹⁴ <https://laravel.com/docs/7.x/artisan#introduction>

APÊNDICE B – DETALHAMENTO DOS CASOS DE USO

Nome do Caso de Uso	Realizar Login
Resumo (Objetivo)	SCPF deve permitir que SAE previamente cadastrado acesse o sistema e suas funcionalidades
Atores	SAE, SCPF
Pré-condições	Necessário SAE ter cadastro no SCPF
Fluxo Principal	<p>P1. SCPF apresenta a tela para “realizar login”</p> <p>P2. SAE informa email e senha</p> <p>P3. SAE clica botão “Entrar”</p> <p>P4. SCPF “valida SAE”</p> <p>P5. SCPF exibe página de listar beneficiários</p> <p>FA1. No P1. SAE com sessão ativa, SCPF retorna P5</p> <p>FA2. No P2. SAE informa dados inválidos</p> <p style="padding-left: 40px;">1) SCPF exibe mensagem “Dados incorretos”</p> <p style="padding-left: 40px;">2) SCPF retorna P1</p> <p>FA3. No P2. SAE não informa dados obrigatórios</p> <p style="padding-left: 40px;">1) SCPF exibe mensagem “Dados obrigatórios não foram informados”</p> <p style="padding-left: 40px;">2) SCPF retorna P1.</p> <p>FA4. No P4. SCPF não encontra dados do SAE.</p> <p style="padding-left: 40px;">1) SCPF exibe mensagem “SAE não cadastrado na base de dados”</p> <p style="padding-left: 40px;">2) SCPF retorna P1</p> <p>FA5. No P2. SAE não lembra senha</p> <p style="padding-left: 40px;">1) SAE clica no botão “recuperar senha”</p> <p style="padding-left: 40px;">2) SCPF exibe tela recuperar senha</p> <p style="padding-left: 40px;">3) SAE informa email cadastrado</p> <p style="padding-left: 40px;">4) SCPF envia senha aleatório para email do SAE</p> <p style="padding-left: 40px;">5) SCPF retorna P1</p>
Fluxo Alternativo	
Pós-condições	

Nome do Caso de Uso	Realizar Login
Requisitos Não Funcionais	<ul style="list-style-type: none"> - SAE deve informar dados previamente cadastrados no base de dados do SCPF - Caso SAE informe dados válidos SCPF deve autenticar SAE e apresentar tela de listagem de beneficiários pois essa é a tela principal do SCPF - Caso o SAE informe dados inválidos ou não informe nenhum dados o SCPF deve exibir uma mensagem para o SAE que não deve sair da tela de login - Caso o SAE não lembre sua senha, SCPF deve permitir que usuário recupere a senha
Requisitos Não Funcionais	RNF1. Em caso de recuperação de senha, deve ser enviada para o email cadastrado do SAE
Autor	Adriano Prado Cavalheiro
Data	12/03/2021
Regras de Validação	<ul style="list-style-type: none"> - Dados não nulos - Dados correspondem cadastro na base de dados
Observações	

Nome do Caso de Uso	Adicionar beneficiários
Resumo (Objetivo)	SCPF deve permitir ao SAE adicionar beneficiários dos auxílios assistências por meio de planilhas excel, SCPF deve permitir o cadastro de beneficiários via formulário opcionalmente
Atores	SAE, SCPF
Pré-condições	Listagem dos beneficiários contemplados com o auxílio estudantil

Nome do Caso de Uso	Adicionar beneficiários
Fluxo Principal	<p>P1. SAE clica no botão “adicionar beneficiários”</p> <p>P2. SCPF exibe tela para submissão de planilhas</p> <p>P3. SAE seleciona planilha para submissão em seu ambiente local</p> <p>P4. SAE clica no botão “Processar planilha”</p> <p>P5. SCPF salva beneficiários na base de dados</p>
Fluxo Alternativo	<p>FA1. No P3. Caso planilha não seja no formato correto</p> <ol style="list-style-type: none">1) SCPF deve informar com mensagem “Planilha em formato incorreto”2) SCPF retorna P2 <p>FA2. No P3. Caso SAE submeter formulario vazio</p> <ol style="list-style-type: none">1) SCPF deve apresentar mensagem “Campo obrigatório”2) SCPF retorna P2 <p>FA3. No P1. Caso SAE precise inserir beneficiário via formulário</p> <ol style="list-style-type: none">1) SCPF deve apresentar no P2 botão para apresentar formulário de adição de beneficiário2) SAE deve preencher campos com informações validas3) SAE deve submeter formulário4) SCPF deve salvar beneficiário na base de dados

Nome do Caso de Uso	Adicionar beneficiários
Pós-condições	
Regras de Negócio	<ul style="list-style-type: none"> - SCPF deve permitir que SAE adicione beneficiários via planilhas opção principal - SCPF deve permitir que SAE adicione beneficiário via formulário como opção alternativa - Se planilha não tiver extensão correta sistema deve exibir mensagem “Extensão de planilha inválida” - Se algum dado não atender a especificação sistema deve exibir mensagem “dados inválidos” - Sistema deve salvar beneficiários na base de dados
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	12/03/2021
Regras de Validação	<ul style="list-style-type: none"> - Dados beneficiários não nulos - Planilha formato xlsx - Planilha preenchida padrão válido
Observações	

Nome do Caso de Uso	Consultar beneficiário
Resumo (Objetivo)	SCPF deve permitir SAE pesquisar beneficiários por filtros: nome, matricula, curso, periodo letivo
Atores	SAE, SCPF

Nome do Caso de Uso	Consultar beneficiário
Pré-condições	Beneficiários cadastrados na base de dados
Fluxo Principal	<p>P1. SCPF deve exibir campos para possibilitar pesquisa de beneficiários</p> <p>P2. SAE deve informar na consulta o respectivo filtro</p> <p>P3. SAE deve clicar no botão “Pesquisar”</p> <p>P4. SCPF deve exibir resultados da consulta</p>
Fluxo Alternativo	<p>FA1. No P4. Caso SAE não preencha nenhum filtro</p> <ul style="list-style-type: none">1) SCPF deve exibir mensagem “Nenhum filtro foi preenchido”2) SCPF retorna P1 <p>FA2. No P4. Caso SAE informe filtro que não corresponda a nenhum resultado na base de dados</p> <ul style="list-style-type: none">1) SCPF deve exibir mensagem “Pesquisa não corresponde”2) SCPF retorna P1
Pós-condições	
Regras de Negócio	<ul style="list-style-type: none">- SCPF deve permitir que SAE pesquise dados de beneficiários- SCPF deve permitir SAE informar filtros para a pesquisa
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	<ul style="list-style-type: none">- No mínimo um campo de consulta deve ser preenchido
Observações	

Nome do Caso de Uso	Editar beneficiário
Resumo (Objetivo)	SCPF deve permitir que SAE edite informações de beneficiários
Atores	SAE, SCPF
Pré-condições	Beneficiário cadastrado na base de dados
Fluxo Principal	<p>P1. SCPF deve exibir na listagem de beneficiários botão “Editar beneficiário”</p> <p>P2. SAE clica no botão “Editar beneficiário”</p> <p>P3. SCPF deve apresentar formulário com dados do beneficiário preenchidos</p> <p>P4. SAE pode editar um ou todos os dados</p> <p>P5. SAE clica no botão “Atualizar beneficiário”</p> <p>P6. SCPF deve salvar beneficiário com dados atualizados</p> <p>P7. SCPF retorna P1</p>
Fluxo Alternativo	<p>FA1. No P1. Caso não exista nenhum beneficiário na base de dados</p> <p>1) SCPF deve exibir mensagem “nenhum beneficiário cadastrado na base de dados”</p> <p>FA2 No P5. Caso SAE informe campo em branco</p> <p>1) SCPF deve exibir mensagem “Campo obrigatório não preenchido”</p> <p>2) SCPF retorna P3</p>

Nome do Caso de Uso	Editar beneficiário
Pós-condições	- Beneficiário com dados atualizados na base de dados
Regras de Negócio	- SCPF deve permitir que SAE edite dados de beneficiário cadastrados na base de dados
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	- Dados beneficiário não nulos - Dados correspondem a beneficiário na base de dados
Observações	

Nome do Caso de Uso	Submeter planilhas
Resumo (Objetivo)	SCPF deve permitir SAE adicionar planilhas com informações de frequências dos alunos para processamento, como resultado SCPF deve ser capaz de validar de beneficiários têm frequência mínima para o recebimento de auxílio estudantil correspondente ao seu grupo de auxílio e tipo de auxílio
Atores	SAE, SCPF
Pré-condições	- Planilhas geradas nos sistemas acadêmicos com informações sobre frequência dos alunos

Nome do Caso de Uso	Submeter planilhas
Fluxo Principal	<p>P1. SCPF deve exibir formulário para submissão de arquivo</p> <p>P2. SAE deve selecionar em seu ambiente local uma ou mais planilhas com a relação de frequência dos alunos</p> <p>P3. SAE clica no botão “Submeter planilhas”</p> <p>P4. SCPF deve processar planilhas comparando com os beneficiários dos auxílios</p> <p>P5. SCPF deve exibir informações sobre beneficiários aptos e não aptos para pagamentos de auxílios</p> <p>FA1. No P3. Caso SAE não selecione planilhas</p> <ol style="list-style-type: none">1) SCPF deve exibir mensagem “Campo obrigatório”2) SCPF deve retornar P1 <p>FA2. No P3. Caso planilha não corresponda ao formato válido</p> <ol style="list-style-type: none">1) SCPF deve exibir mensagem “Formato de arquivo inválido”2) SCPF deve retornar P1 <p>FA3. No P3. Caso colunas planilha não corresponda a padrão válido</p> <ol style="list-style-type: none">1) SCPF deve exibir mensagem “Arquivo não corresponde padrão válido”2) SCPF deve retornar P1.
Fluxo Alternativo	

Nome do Caso de Uso	Submeter planilhas
Pós-condições	- Permitir que SAE visualize beneficiários aptos e não aptos a receber pagamentos de auxílios
Regras de Negócio	- SAE deve submeter ao menos uma planilha com padrão válido no SCPF - SCPF deve processar planilhas comparando alunos beneficiários cadastrados e ativos na base de dados com alunos contidos na planilha submetida - SCPF deve apresentar beneficiários aptos e não aptos ao pagamento do auxílio
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	- Arquivo deve ter formato xlsx - Padrão de colunas deve ser no formato que SCPF esteja apto a processar - No mínimo um arquivo deve ser submetido
Observações	

Nome do Caso de Uso	Solicitar relatórios
Resumo (Objetivo)	SCPF deve permitir a SAE que gere relatórios
Atores	SAE, SCPF
Pré-condições	- SCPF deve conter dados na base de dados - SAE deve pesquisar beneficiários

Nome do Caso de Uso	Solicitar relatórios
Fluxo Principal	<p>P1. No resultado de CU. Pesquisar beneficiários, SCPF deve exibir botão “Gerar relatórios”</p> <p>P2. SAE clica no botão “Gerar relatórios”</p> <p>P3. Download automático do relatório</p> <p>P4. SCPF retorna P1</p>
Fluxo Alternativo	
Pós-condições	- Relatório disponível no ambiente local do SAE
Regras de Negócio	<ul style="list-style-type: none"> - SAE pesquisa beneficiarios informando filtros desejados - SCPF deve permitir que SAE gere relatório dos resultados das pesquisas - SCPF deve permitir SAE gerar relatório xlsx e PDF - Relatório deve ser baixado na máquina do SAE
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	
Observações	

Nome do Caso de Uso	Justificar frequência
Resumo (Objetivo)	SCPF deve permitir a SAE justificar a frequência de beneficiários que não tenham atingido frequência mínima no mês e que tenham justificativa comprovada

Nome do Caso de Uso	Justificar frequência
Atores	SAE, SCPF
Pré-condições	- Beneficiário abaixo da frequência mínima - Beneficiário com justificativa comprovada
Fluxo Principal	P1. SAE verifica se coordenadores enviaram justificativa de infrequentes P2. SAE consulta beneficiário com frequência justificada no tela de pesquisa do SCPF P3. SCPF retorna beneficiário consultado P4. SAE adiciona justificativa P5. SCPF salva justificativa na base de dados P6. Beneficiário removido lista de infrequentes P7. Beneficiário apto para receber auxílio
Fluxo Alternativo	FA1. No P2. Caso não haja infrequentes com justificativas 1) SAE deve prosseguir com a geração da CU. Gerar planilha pagamentos
Pós-condições	- Justificativas salvas na base de dados e relacionadas aos beneficiários - Beneficiários aptos a receber auxílio
Regras de Negócio	- Beneficiários com frequência justificada - SCPF deve permitir que SAE adicione justificativa ao beneficiário
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021

Nome do Caso de Uso Justificar frequência

Regras de Validação - Justificativa não pode ser nula

Observações

Nome do Caso de Uso Configurar auxílios estudantis

Resumo (Objetivo) Os auxílios são divididos em dois tipos, moradia e permanência, um beneficiário pode receber um ou os dois tipos. Os auxílios são categorizados por grupos de pagamento cada beneficiário é atendido por apenas um grupo. Os valores de pagamentos variam de acordo com o grupo e podem sofrer alterações. O SCPF deve permitir ao SAE alterar os valores de pagamentos dos grupos

Atores SAE, SCPF

Pré-condições - Grupos cadastrados na base de dados

Fluxo Principal P1. SCPF deve exibir tela para consulta de auxílio
P2. SAE consulta auxílio que deseja configurar
P3. SCPF exibe formulário para edição de auxílio com dados do retorno preenchidos
P4. SAE submete auxilio configurado
P5. SCPF salva auxilio atualizado
P6. SCPF retorna P1.

Nome do Caso de Uso	Configurar auxílios estudantis
	FA1. No P2. Caso SAE não informe dados para consulta
	1) SCPF exibe mensagem “Campo obrigatório”
	2) SCPF retorna P1
	FA2. No P2. Caso auxilio não conste na base de dados
	1) SCPF exibe mensagem “Auxilio não cadastrado na base de dados”
	2) SCPF retorna P1
Fluxo Alternativo	FA3. No P4. Caso SAE não informe algum dado obrigatório
	1) SCPF exibe mensagem “Campo obrigatório”
	2) SCPF retorna P3
	FA4. No P5. Caso SAE não informe dado válido
	1) SCPF exibe mensagem “Dado inválido”
	2) SCPF retorna P3

Nome do Caso de Uso Configurar auxílios estudantis

Pós-condições Auxilio atualizado na base de dados

Regras de Negócio - SCPF deve permitir que SAE edite dados de auxílio

Requisitos
Não
Funcionais

Autor Adriano Prado Cavalheiro

Data 13/03/2021

Regras de Validação - Campos não nulos

Observações

Nome do Caso de Uso Gerar planilha pagamentos

Resumo (Objetivo) SCPF deve permitir que SAE gere planilhas de pagamentos de auxílios mensais, SAE deve baixar planilha no formato xlsx na sua máquina

Atores SAE, SCPF

Pré-condições - Beneficiários cadastrados
- CU. Submeter planilhas
- CU. Justificar frequência

Nome do Caso de Uso	Gerar planilha pagamentos
Fluxo Principal	<p>P1. SCPF deve exibir listagem de beneficiários aptos para receber auxílios</p> <p>P2. SCPF deve exibir opção " Gerar Planilha" de pagamento</p> <p>P3. SAE seleciona opção " Gerar Planilha"</p> <p>P4. Download automático de planilha na máquina do SAE</p> <p>P5. Retorna P1</p>
Fluxo Alternativo	<p>FA1. No P1. Caso não tenha beneficiários aptos</p> <p>1) SCPF deve exibir mensagem "Nenhum beneficiário apto"</p>
Pós-condições	- Planilha de pagamentos disponível na máquina do SAE
Regras de Negócio	<p>- Planilha deve ser gerada em formato xlsx</p> <p>- Planilha não pode ser gerada sem dados</p>
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	
Observações	

Nome do Caso de Uso	Enviar email coordenações
Resumo (Objetivo)	SCPF deve permitir que SAE notifique coordenadores de cursos sobre infrequentes do mês

Nome do Caso de Uso	Enviar email coordenações
Atores	SAE, SCPF
Pré-condições	- CU. Processar planilhas - Beneficiários infrequentes
Fluxo Principal	P1. SCPF deve permitir que SAE selecione beneficiários infrequentes para informar a coordenação do respectivo curso P2. SCPF deve exibir opção de “Notificar coordenação” P3. SCPF deve exibir formulário para informar assunto e coordenação a ser notificada e corpo notificação P4. SAE clica botão “Prévia” P5. SCPF exibe prévia notificação para SAE contendo texto padrão mais texto do corpo e relação de infrequentes selecionados P6. SAE submete notificação P7. Retorna P1
Fluxo Alternativo	
Pós-condições	Recebimento de resposta das coordenações notificadas
Regras de Negócio	- Se houver algum beneficiário infrequente SAE deve notificar coordenações para validar frequência - Após validação coordenações informam SAE se há justificativas
Requisitos Não Funcionais	RNF1. Notificação deve ser via email
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	

Nome do Caso de Uso	Enviar email coordenações
Observações	Resposta de coordenações devem ser retornadas para email do setor de Assistência Estudantil

Nome do Caso de Uso	Adicionar usuário
Resumo (Objetivo)	SCPF deve permitir que SAE cadastre usuário SAE na base de dados
Atores	SAE, SCPF
Pré-condições	Ao menos um usuário SAE deve estar cadastrado na base de dados
Fluxo Principal	<p>P1. SCPF exibe botão “Adicionar usuário”</p> <p>P2. SAE clica botão “Adicionar usuário”</p> <p>P3. SCPF exibe formulário para adição de usuário SAE</p> <p>P4. SAE informa dados usuário</p> <p>P5. SAE clica botão “Salvar”</p> <p>P6. Retorna CU. Listar usuários</p>
Fluxo Alternativo	<p>FA1. No P5. SAE não informa dados obrigatórios</p> <ol style="list-style-type: none">1) SCPF exibe mensagem “Campos obrigatórios”2) Retorna P3 <p>FA2. No P5. SAE não informa nenhum dado</p> <ol style="list-style-type: none">1) SCPF exibe mensagem “Usuário não pode ser nulo”2) Retorna P3

Nome do Caso de Uso	Adicionar usuário
Pós-condições	- Novo SAE cadastrado na base de dados
Regras de Negócio	- Cadastro deve ter nome, email e senha
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	- Todos os campos devem ser obrigatórios - Não pode ser cadastrado nulo
Observações	

Nome do Caso de Uso	Editar usuário
Resumo (Objetivo)	SCPF deve permitir edição de SAE
Atores	SAE, SCPF
Pré-condições	- SAE cadastrado na base de dados
Fluxo Principal	<p>P1. SCPF exibe opção para edição de SAE</p> <p>P2. SAE seleciona opção de edição</p> <p>P3. SCPF exibe formulário com os respectivos dados de SAE preenchidos</p> <p>P4. SAE edita dado desejado</p> <p>P5. SAE clica botão “Atualizar”</p> <p>P6. SCPF salva SAE atualizado na base de dados</p> <p>P7. Retorna P1</p> <p>FA1. No P5. Dados obrigatórios não informados</p>
Fluxo Alternativo	<p>1) SCPF exibe mensagem “Dados obrigatórios”</p> <p>2) Retorna P3</p>
Pós-condições	Dados SAE atualizados na base de dados
Regras de Negócio	
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro

Nome do Caso de Uso	Editar usuário
---------------------	----------------

Data	13/03/2021
------	------------

Regras de Validação	- Todos os campos obrigatórios - SAE não nulo
---------------------	--

Observações	
-------------	--

Nome do Caso de Uso	Listar usuários
---------------------	-----------------

Resumo (Objetivo)	SCPF deve exibir em uma tela todos os usuários SAE cadastrados na base de dados
-------------------	---

Atores	SAE, SCPFI
--------	------------

Pré-condições	SAE cadastrado na base de dados
---------------	---------------------------------

Fluxo Principal	P1. SAE seleciona item no menu “Listar usuários” P2. SCPF exibe usuários SAE cadastrados na base de dados
-----------------	--

Fluxo Alternativo	
-------------------	--

Pós-condições	Todos os usuários SAE exibidos para SAE
---------------	---

Regras de Negócio	
-------------------	--

Requisitos Não Funcionais	
---------------------------	--

Nome do Caso de Uso	Listar usuários
---------------------	-----------------

Autor	Adriano Prado Cavalheiro
-------	--------------------------

Data	13/03/2021
------	------------

Regras de Validação	
---------------------	--

Observações	
-------------	--

Nome do Caso de Uso	Remover usuário
---------------------	-----------------

Resumo (Objetivo)	SCPF deve permitir remoção de SAE
-------------------	-----------------------------------

Atores	SAE, SCPF
--------	-----------

Pré-condições	- SAE cadastrado na base de dados
---------------	-----------------------------------

Fluxo Principal	P1. SCPF exibe opção para remoção de SAE P2. SAE seleciona opção de remoção P3. SAE clica botão “remover” P4. SCPF exibe mensagem “Tem certeza que deseja remover?” P5. SAE responde positivo P4 P6. SCPF remove SAE da base de dados P7. Retorna P1
-----------------	--

Nome do Caso de Uso	Remover usuário
Fluxo Alternativo	FA1. No P5. Caso resposta negativa P4 1) Retorna P1
Pós-condições	SAE removido da base de dados
Regras de Negócio	
Requisitos Não Funcionais	
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	
Observações	

Nome do Caso de Uso	Recuperar senha
Resumo (Objetivo)	SCPF deve exibir na tela de login opção para a recuperação de senha, usuário deve receber uma senha aleatória de acesso em seu email de cadastro
Atores	SAE, SCPF
Pré-condições	- SAE cadastrado na base de dados - SAE lembrar email de cadastro

Nome do Caso de Uso	Recuperar senha
Fluxo Principal	<p>P1. SCPF exibe na tela de login opção “Recuperar senha”</p> <p>P2. SAE clica na opção “Recuperar senha”</p> <p>P3. SCPF exibe tela para SAE informar email de cadastro</p> <p>P4. SAE informa email de cadastro</p> <p>P5. SAE clica botão “Enviar”</p> <p>P6. Retorna P1</p> <p>P7. SCPF gera senha aleatória</p> <p>P8. SCPF envia notificação com a nova senha para SAE</p> <p>P9. SCPF recebe confirmação de entrega de notificação</p> <p>P10. SCPF atualiza senha SAE na base de dados</p>
Fluxo Alternativo	<p>FA1. No P5. Caso SAE informe email inválido</p> <ul style="list-style-type: none">1) SCPF exibe mensagem “Email inválido”2) Retorna P3 <p>FA2. No P5. Caso SAE não informe dados</p> <ul style="list-style-type: none">1) SCPF exibe mensagem “Email obrigatório”2) Retorna P3 <p>FA3. No P9. Caso SCPF não receba confirmação de entrega</p> <ul style="list-style-type: none">1) SCPF não atualiza senha SAE na base de dados

Nome do Caso de Uso	Recuperar senha
Pós-condições	- Acesso SAE nova senha
Regras de Negócio	- SCPF deve permitir que SAE recupere senha - Uma nova senha aleatoria deve ser gerada - Senha nova deve ser enviada para email cadastrado do SAE - SCPF deve receber confirmação de entrega da notificação - Senha deve ser atualizada na base de dados caso confirma entrega de notificação
Requisitos Não Funcionais	- SCPF deve gerar senha aleatória - Notificação via email
Autor	Adriano Prado Cavalheiro
Data	13/03/2021
Regras de Validação	- Email válido - Email cadastrado - Dado não nulo
Observações	Precisa verificar se existe como receber confirmação de entrega de email

Anexos

ANEXO A – DOCUMENTO DE TESTE COM USUÁRIOS

Neste teste de usabilidade iremos testar o Sistema de Coleta e Processamento de Frequências (SCPF).

Anotações gerais

Produto: *SCPF*

Teste nº: 1

Data: 17/08/2021

Início da sessão: 14:30

Término da sessão: 17:00

Local: Remoto

Entrevistador: Adriano Prado Cavalheiro

Observador: Professor orientador Gleison

Usuário que realizou os testes: Márcia Pedroso

Realização de Tarefas

O principal objetivo é a avaliação do sistema por parte dos usuários, se o sistema atende às expectativas, quais pontos são necessários melhorar.

Tarefa A

Página: Geral

Objetivo: Entender se o objetivo do sistema está claro

Descrição da tarefa: **Navegue por alguns minutos pelo site, para conhecer o ambiente. Você sabe explicar sobre o que se trata este site?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Sistema esteticamente agradável, instintivo, e atende pontualmente a todos as solicitações efetuadas nas reuniões realizadas com o programador.

Tarefa B

Página: Menu lateral

Objetivo: Entender se o menu do sistema está intuitivo.

Descrição da tarefa: **Sobre a disposição dos menus e suas legendas, passam as informações desejadas?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

As funcionalidades foram construídas a partir de escuta com os usuários, com reuniões para avanços, dando conta das necessidades especificadas.

Tarefa C

Página: Cadastrar coordenções

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Você deve cadastrar as coordenções dos cursos**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Fácil de cadastrar e descadastrar, assim como de editar os dados dos coordenadores, nomes e e-mails.

Tarefa D

Página: Cadastrar cursos

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Você poderá cadastrar os cursos, para esta tarefa é importante atentar para os nomes dos cursos, Estes devem ser condizentes com os arquivos gerados pelos sistemas acadêmicos**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Fácil de cadastrar e descadastrar os cursos. Atende aos propósitos.

Tarefa E

Página: Cadastrar auxílios

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Você poderá cadastrar os auxílios**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Excelente funcionalidade, fácil e instintiva de cadastrar ou descadastrar os auxílios e os valores.

Tarefa F

Página: Página adicionar administradores

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Você poderá um novo administrador para o sistema.**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

A funcionalidade atende à finalidade, sendo fácil e instintiva a forma de cadastrar ou descadastrar os administradores.

Tarefa G

Página: Página adicionar beneficiários

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Você poderá cadastrar beneficiários via arquivo e via formulário, conforme requisitos solicitados.**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Excelente funcionalidade, fácil e instintiva de cadastrar. O upload da planilha de beneficiários foi feito rapidamente e com a indicação do número de beneficiários cadastrados.

Tarefa H

Página: Página Processar frequências.

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Inserir um ou mais arquivos de frequências, consegue entender o fluxo?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

O sistema operou tranquilamente o upload de duas planilhas de dois sistemas diferentes, com mais de 800 estudantes. Subiu as duas planilhas ao mesmo tempo e super rápido.

Tarefa I

Página: Relatórios.

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Está clara a proposta desta página?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

Sim, proposta clara, esteticamente agradável e com botões instintivos, fornece a planilha pronta para verificação do status da frequência para a definição dos casos a justificar.

Tarefa J

Página: Menu de ações (Notificação e Justificativa).

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Está clara a proposta das ações, tanto justificativa, quanto envio de notificações?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais:

São ótimas funcionalidades, permite enviar e-mails às Coordenações de Curso por curso, ou no conjunto de cursos. Da mesma forma permite justificar individualmente ou clicar em conjuntos de estudantes a terem frequência justificada.

Tarefa L

Página: Página Pagamentos.

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Está clara a proposta desta página?**

Usuário conseguiu realizar a tarefa?

- Sim, com facilidade
- Sim, com dificuldade
- Não conseguiu realizar a tarefa

Impressões gerais.

Tarefa M

Objetivo: Entender se está fácil para o usuário dar sua opinião.

Descrição da tarefa: **Utilizar o sistema como lhe aprouver, não esqueça de anotar todas as observações que surgirem.**

Impressões gerais:

Fornece a planilha pronta, a partir das demais informações, com efetividade.

Perguntas:

O sistema atende suas expectativas?

- Sim

- Não

Quão satisfeito você ficou com o sistema:

- Muito satisfeito

- Nada satisfeito

- Satisfeito

O sistema atende a proposta de agilizar o processo de pagamento dos auxílios estudantis no campus Restinga do IFRS?

- Sim

- Não

Conclusão

Momento para rever como o usuário se sentiu realizando as tarefas, se tem algo a adicionar para contribuir com a melhoria do site.

Se não tiver tocado nos seguintes assuntos, é possível perguntar diretamente ao usuário o que ele achou sobre:

- Realizar uma busca no sistema
- Dar sua opinião sobre o sistema
- Testar a acessibilidade do sistema
- Encontrar serviços disponíveis no sistema
- Sobre gerar folha de pagamentos
- Sobre as entradas, se atendem às necessidades do setor
- Informar sobre a questão de padrão dos arquivos para processamento
- Comentar sobre quaisquer pontos que não foram comentados