



Módulo de Aluno Especial do SiGPos - Sistema de Gerenciamento da Pós-Graduação do IC

*M. Martins L. Mendonça C. Rubira A. Braga
P. Sérgio*

Relatório Técnico - IC-PFG-21-24
Projeto Final de Graduação
2021 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

AGRADECIMENTOS

Às nossas famílias, pelo carinho e apoio.
À equipe do SiGPos, pela dedicação, atenção e suporte.

Sumário

1	Introdução	3
1.1	Contexto	3
1.2	Problema	4
1.3	Solução	4
2	Fundamentos	4
2.1	O SiGPos	4
2.2	O sistema Castelo	5
2.3	Java e o framework Spring Boot	6
2.4	JavaScript e o framework VueJS	8
2.5	MySQL	8
2.6	Metodologia Agile	9
2.7	User Stories e Requisitos do SiGPos	9
3	Visão geral da solução desenvolvida	11
3.1	Escopo da Solução	11
3.2	Arquitetura do SiGPos	12
4	User Story 1 - Candidatar a Aluno Especial pela primeira vez	13
5	User story 2 - Candidatar a Aluno Especial pela segunda vez	16
6	User story 3 - Acessar painel do sistema para visualizar informações	18
7	Resultados obtidos	19
7.1	Avaliação da Solução	19
7.2	Trabalhos Futuros	20
8	Dificuldades Encontradas e Lições Aprendidas	20
9	Conclusões	21

Módulo de Aluno Especial do SiGPos - Sistema de Gerenciamento da Pós-Graduação do IC

Marcelo Biagi Martins Lucas Rodrigues Mendonça
Cecília Mary Fischer Rubira* Alexandre Melo Braga*
Paulo Sérgio do Nascimento*

Resumo

O Módulo de Aluno Especial do SiGPos foi desenvolvido para substituir o sistema *Castelo* no gerenciamento de alunos especiais, centralizando as atividades acadêmicas da Pós-Graduação do IC em uma única aplicação.

Além do problema da falta de unificação entre as partes, o sistema *Castelo* era pouco seguro e pouco otimizado, forçando os alunos especiais a inserirem novamente todos os seus dados a cada novo semestre.

Para resolver o problema, foi desenvolvida uma solução dentro da plataforma do SiGPos, utilizando tecnologias atuais e implementando validação de campos e persistência de dados, eliminando assim a necessidade de preenchimento de informações redundantes a cada matrícula e tornando o processo mais simples e direto.

Apesar de notáveis as melhorias implementadas durante este projeto, seu foco principal foi o usuário. Para ser colocado em produção, o sistema ainda precisa de uma interface administrativa, que será implementada em interações futuras com foco no administrador.

Dessa forma, este trabalho apresenta o primeiro passo na modernização do processo de matrícula de aluno especial do IC e os resultados obtidos nessa primeira interação, bem como as dificuldades encontradas e lições aprendidas.

1 Introdução

1.1 Contexto

O desenvolvimento do Módulo de Aluno Especial do SiGPos foi gerenciado por uma equipe de cinco indivíduos: dois desenvolvedores (L. Mendonça e M. Martins), responsáveis pela implementação das novas funcionalidades e dos novos recursos do sistema; uma orientadora (C. Rubira), que também assumiu o papel de proprietária do sistema ou *Project Owner*, solicitando modificações e analisando os resultados obtidos; um gerente de projeto ou *Scrum Master* (A. Braga), responsável pela organização do cronograma de desenvolvimento e pelo acompanhamento dos avanços e das dificuldades encontradas no decorrer do projeto; e um analista técnico do IC (P. Nascimento), com o objetivo de fornecer suporte para os desenvolvedores e realizar a homologação das mudanças.

*Instituto de Computação, Universidade Estadual de Campinas, 13081-970 Campinas, SP.

1.2 Problema

Alunos especiais são indivíduos que não possuem vínculo estudantil com a Unicamp, mas desejam se matricular em disciplinas de graduação ou pós-graduação. Para que seu pedido de matrícula seja aceito, cada instituto é responsável por realizar um processo seletivo que definirá se o indivíduo está ou não apto para cursar a disciplina, com base em critérios como análise de currículo, análise de histórico acadêmico e análise de vínculo empregatício em área diretamente relacionada à disciplina desejada.

Até o início de 2020, a única maneira de um aluno especial se matricular em uma disciplina do IC era preenchendo um formulário de cadastro hospedado no domínio *castelo.ic*. O fato deste formulário estar em um domínio independente tornava o processo descentralizado, tanto para os usuários quanto para os docentes e administradores do sistema, visto que a maioria das funcionalidades de gerenciamento acadêmico da pós-graduação do IC já estão concentradas no SiGPos. Além disso, a falta de unificação entre os sistemas gerava retrabalho por parte dos alunos, que precisavam inserir todo semestre a mesma lista de dados e informações pessoais. Ademais, os docentes possuíam pouco controle sobre os dados dos candidatos a aluno especial, tornando o processo extremamente manual. Por fim, faltava clareza acerca das responsabilidades e dos deveres de cada plataforma.

1.3 Solução

Para resolver o problema, foi desenvolvido uma nova solução baseada no sistema *Castelo*, desta vez dentro da plataforma do SiGPos. Essa solução, chamada de Módulo do Aluno Especial, resgata as funcionalidades e responsabilidades do sistema antigo (*Castelo*) e adiciona novas funcionalidades, sendo mais seguro, mais intuitivo e mais robusto. Dessa forma, não seria mais necessária a segregação dos alunos especiais em outra plataforma, eliminando o sistema extra e resolvendo assim o problema de descentralização.

2 Fundamentos

2.1 O SiGPos

O Sistema de Gestão Acadêmica para Pós-Graduação (SiGPos)[1] tem como objetivo facilitar e agilizar as atividades acadêmicas da Pós-Graduação do IC, tanto da parte dos alunos, quanto da parte administrativa. Para isso, o SiGPos mantém uma base de dados própria, que pode ser utilizada para análises diversas e construção de métricas e gráficos com as informações sobre a vida acadêmica das partes envolvidas. Entre essas análises, podemos citar as rotinas administrativas da coordenação e o acompanhamento da vida acadêmica dos alunos, desde a sua inscrição nos processos de seleção da pós-graduação até a conclusão do curso.

O SiGPos não é responsável por controlar notas e frequência dos alunos, visto que esta função já é desempenhada pelo sistema da DAC - Diretoria Acadêmica da Unicamp. Da mesma maneira, quaisquer atividades que não sejam acadêmicas também fogem do escopo do sistema.

Existem módulos no sistema que contemplam funcionalidades de regulamentação necessárias à pós-graduação do IC. São eles: Gerenciamento, Inscrição do Candidato, Recomendação de Avaliadores, Montagem de Banca de Avaliação, Processo de Avaliação, Vida Acadêmica do Aluno Regular, Módulo do Orientador, Seleção do Pós-Doutorado, Pós-Doutorado, Controle de Bolsas, Solicitação de Auxílio para Participação em Evento Científico e, com o fim deste projeto, Seleção do Aluno Especial e Vida Acadêmica do Aluno Especial.

Quanto à implementação, o SiGPos é dividido em back-end e front-end. O primeiro é responsável por gerenciar as regras de negócio do sistema e armazenar os dados necessários. Já o segundo, por exibir os resultados obtidos ao usuário final, bem como apresentar a interface visual do sistema. O back-end foi desenvolvido utilizando a linguagem Java, com auxílio do framework Spring Boot, enquanto seu banco de dados foi implementado com o serviço de gerenciamento MySQL. O front-end, por sua vez, foi implementado na linguagem JavaScript, utilizando o framework VueJS. A tela inicial do SiGPos está na figura 1.

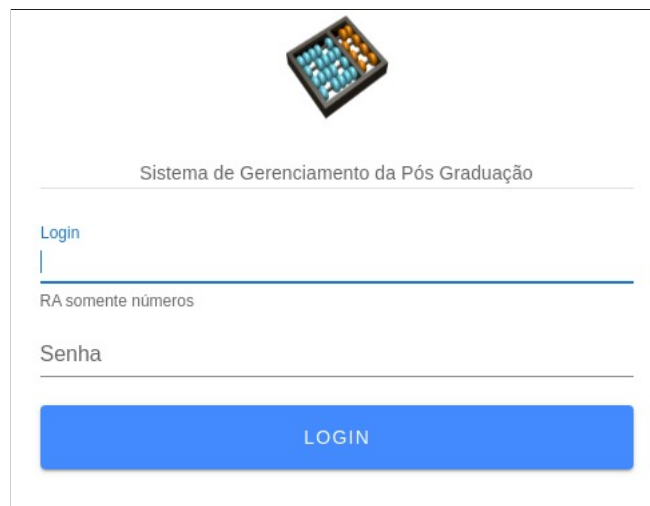
A imagem mostra a tela inicial do sistema SiGPos. No topo centralizado, há um ícone de uma caixa de jogos de tabuleiro. Abaixo dele, o título "Sistema de Gerenciamento da Pós Graduação" é exibido. O formulário de login contém um campo rotulado "Login" com uma barra de texto vazia. Abaixo dele, há uma restrição "RA somente números". O campo de senha é rotulado "Senha" e também possui uma barra de texto vazia. Na base do formulário, um botão azul com o texto "LOGIN" em branco está centralizado.

Figura 1: Tela inicial do SiGPos

2.2 O sistema Castelo

O sistema Castelo[2] é uma página auxiliar pouco segura e pouco otimizada, através da qual o IC realiza as inscrições de alunos especiais. Formado apenas por um formulário composto por uma extensa página HTML de layout completamente desatualizado em relação aos padrões atuais, a aplicação recebe os dados dos candidatos a aluno especial e os insere em uma planilha externa. Entre as informações requeridas pelo formulário estão: a formação acadêmica completa do candidato, dois endereços completos, sendo um deles na região de Campinas e outro fora, as disciplinas que o aluno tem interesse em se candidatar e muitas perguntas de cunho pessoal. Ao completar o formulário, os alunos são direcionados para uma segunda página, que permite o upload de arquivos, como histórico escolar e cartas de recomendação.

A aplicação não informa irregularidades no preenchimento de campos e nem faz qualquer tipo de validação, o que além de transferir a responsabilidade de validação dos dados para o usuário, permite a inserção de dados fraudulentos, como e-mails de terceiros e inserção de dados inválidos. Além disso, a falta de padronização dificulta a análise e a filtragem por parte dos administradores do sistema, pois um mesmo dado pode ser inserido de diversas formas, utilizando acrônimos ou espaçamentos irregulares, diferenciando maiúsculas de minúsculas ou desconsiderando a acentuação. Erros durante o envio são punidos com a desconsideração total da inscrição e o candidato tem que preencher novamente toda a extensa lista de informações caso erre algum campo, não sendo possível editar a inscrição após o envio.

Exemplos de alguns campos requeridos no formulário *Castelo* estão presentes nas figuras 2 e 3.

RA (se tiver)	<input type="text"/>
Nome:	<input type="text"/>
RG/RNE:	<input type="text"/>
UF:	<input type="text"/>
CPF:	<input type="text"/>
Sexo:	<input type="text" value="Masculino"/>
Tipo de Aluno:	<input type="text" value="(Escolha uma opção)"/>
Estado Civil:	<input type="text" value="Solteiro"/>
Nome do Pai:	<input type="text"/>
Nome da Mãe:	<input type="text"/>
Data de Nascimento (dd/mm/aaaa):	<input type="text"/>
Cidade:	<input type="text"/>
Estado:	<input type="text"/>
Nacionalidade:	<input type="text" value="Brasileira"/>
Naturalizado:	<input type="text" value="Não"/>

Figura 2: Tabela para inserção de dados pessoais *Castelo*

2.3 Java e o framework Spring Boot

Java[3] é uma linguagem de programação orientada a objetos que, devido a sua robustez e portabilidade, se tornou uma das linguagens mais populares do mundo nos dias atuais, sendo a segunda mais buscada[4] conforme uma pesquisa realizada pelo índice PYPL.

A linguagem foi projetada seguindo os pilares[5] de: concorrência, que é a capacidade de executar expressões simultaneamente; divisão em classes, ou seja, todas as funções, métodos e demais definições estão contidas em classes independentes; orientação de objetos e inde-

Preferência	Disciplinas
1	(selecione uma opção) ▼
2	(selecione uma opção) ▼
Só escolha mais que 2 disciplinas se foi explicitamente orientado a fazer isto	
3	(selecione uma opção) ▼
4	(selecione uma opção) ▼
5	(selecione uma opção) ▼
Número máximo de disciplinas que pretende cursar:	1 ▼

Figura 3: Tabela para escolha das disciplinas *Castelo*

pendência, pois seu código é compilado para um bytecode, que é capaz de ser executado em qualquer sistema que possua uma Java Virtual Machine (JVM) sem a necessidade de recompilação. Java ainda possui uma sintaxe simples e similar à C/C++, atualizações constantes e uma vasta comunidade, sendo uma opção adequada e viável para implementação do sistema.

Desenvolver uma aplicação web apenas com Java puro, ou também chamado *Java vanilla*, é uma tarefa complexa que consome muitas horas. Existem bibliotecas e frameworks que concentram as funcionalidades comuns de um sistema web de maneira mais simples, como o Spring, presente no SiGPos e também no Módulo de Aluno Especial.

O Spring[6][7] possui código aberto e simplifica tarefas comumente realizadas na linguagem Java, como a conversão de classes em tabelas do banco de dados. Para isso, ele está estruturado em pacotes que podem ser adicionados como dependências do sistema. Entre esses, podemos citar o Spring Data JDBC - responsável por gerenciar a conexão entre o Java e o banco de dados - e o Spring Security, responsável pela segurança do usuário e de seus dados, como o login e as senhas.

Baseando-se na arquitetura MVC, o Spring trabalha com anotações que definem cada tipo de componente do sistema. Ao incluir uma anotação em uma classe, o framework define uma configuração pré-definida para ela, não sendo necessário nenhum tipo de implementação manual. Além disso, o Spring complementa a linguagem Java através da inclusão de um container de injeção de dependências, tornando certos objetos responsáveis pelo controle do ciclo de vida de outros (criação, inicialização e configuração).

O Spring Boot[8] é uma extensão do Spring que fornece um website[9] capaz de realizar toda a organização inicial dos arquivos do código, inclusive das suas dependências, eliminando a necessidade do usuário de alterar esses arquivos. Isso permite que o desenvolvedor não se preocupe tanto com as configurações de arquivos XML ou similares, já que o próprio Spring Boot fica encarregado dessa configuração.

Para o desenvolvimento do Módulo do Aluno Especial, as dependências utilizadas foram:

- Lombok: framework que reduz a verbosidade do código, eliminando a necessidade do usuário codificar os Getters, Setters, ToString, Construtores e Builders de uma classe Java
- Spring Data JDBC e Spring Data JPA: provém a interação entre classes e o banco de dados, fornecendo ao usuário a possibilidade de criar e alterar tabelas, queries e

métodos CRUD - criação, leitura, atualização e remoção de dados do banco

- Spring Security: gerencia a autenticação e o login de diversas formas (classes específicas, em memória ou via banco de dados com JDBC)
- Spring Validator: fornece anotações que verificam certos parâmetros dos dados, evitando a criação de cláusulas condicionais, como a anotação "@NotNull", que impede o dado anotado de ser nulo, por meio da emissão de uma exceção
- MySQL Connector: realiza a conexão entre o driver de MySQL com o Java

2.4 JavaScript e o framework VueJS

A linguagem de programação JavaScript[10], juntamente com o HTML e o CSS, é uma das ferramentas que constituem o cerne[11] do desenvolvimento web. Desenvolvida como uma linguagem multi-paradigma, ela possui suporte à orientação a objetos, mas também pode ser utilizada como uma linguagem funcional. JavaScript é uma linguagem interpretada e projetada para lidar com as ações do usuário no front-end, ou seja, na parte das interações do cliente com a interface do sistema.

Com relação à sintaxe, JavaScript é bastante similar a Java/C/C++, sendo uma linguagem de rápido aprendizado e muito poderosa. É a principal forma de interação com o Document Object Model (DOM), que representa a página vista pelo utilizador, de forma que os programas escritos possam alterar o conteúdo, estilo e a estrutura do documento. Sem o DOM, JavaScript não seria capaz de interagir com as telas da forma como faz atualmente. Assim como Java, JavaScript também possui frameworks que facilitam o seu desenvolvimento, como AngularJS, ReactJS e o VueJS[12].

Para o SiGPos e o Módulo de Aluno Especial, foi escolhido o Vue, que é um framework de código aberto utilizado para o desenvolvimento de interfaces de usuário e aplicações dinâmicas de página única. Em Vue, o usuário é capaz de criar componentes, que são trechos de código reutilizáveis e modularizados e as páginas criadas possuem carregamento reativo, isto é, são alteradas conforme o código muda, sem a necessidade de recarregar a página frequentemente. Dessa forma, o desenvolvimento fica mais rápido e mais organizado.

Vue utiliza uma sintaxe similar a HTML para conectar seus dados ao DOM. Sua sintaxe concatena o HTML, CSS e JavaScript de uma página em um único arquivo, simplificando a navegação do desenvolvedor e tornando a relação entre o código mais clara. Devido a esses fatores, VueJS é um framework altamente escalável e rápido, além de não pertencer a nenhuma empresa privada, como o Angular (da Google) e o React (do Facebook).

2.5 MySQL

O MySQL[13] é um sistema de gerenciamento de bancos de dados de código aberto[14], lançado em 1995 pela MySQL AB. Através dele, é possível gerenciar bancos relacionais, ou seja, bancos cujas tabelas possuem linhas e colunas com ID's únicos em que os dados de uma podem estar relacionados com dados de outra. O MySQL é, atualmente, gerenciado pela Oracle e, apesar de possuir uma versão Enterprise, que é paga, sua versão community é totalmente gratuita.

Conhecido por ter uma curva de aprendizado pequena e por ser escalável e veloz, o MySQL é um dos componentes essenciais da pilha *LAMP*[15], que denota as soluções mais comuns encontradas nos aplicativos da web de maneira geral, juntamente com *L-Linux*, *A-Apache* e *P - PHP/Perl/Python*. Além disso, o MySQL é utilizado por grandes empresas de desenvolvimento de software, entre elas: Facebook, Twitter e Youtube, mostrando a confiança do mercado nesse sistema.

2.6 Metodologia Agile

As metodologias ágeis surgiram em 2001, após a divulgação do Manifesto para o Desenvolvimento Ágil de Software[16]. Criado em um contexto de crise na indústria, consequência da bolha dot-com, que eclodiu nos anos 2000, os idealizadores do manifesto notaram que as metodologias da época não eram efetivas para se trabalhar com um setor tão diferente e inovador como o de software. Após a falência de diversas empresas, causadas pelos fracos resultados operacionais, surgiram novas maneiras de organizar grupos para desenvolvimento. Entre elas, o Scrum[17], que foi parcialmente utilizado na execução deste projeto.

Criado em 1995, o Scrum é um framework que utiliza o pensamento ágil para desenvolver, entregar e sustentar projetos. Caracterizado por ser capaz de gerenciar trabalhos complexos, o Scrum divide o projeto em ciclos denominados *sprints*, que devem ter entre uma e quatro semanas de duração. Nesse tempo, um conjunto de atividades deve ser definido e executado. Para organização, utiliza-se um *backlog de produtos*, a fim de gerenciar quais funcionalidades serão implementadas. No início de cada sprint, o time deve realizar o *sprint planning* e planejar a sprint que será iniciada. Essa parte de controle do backlog é responsabilidade do *Scrum Master*, que deve gerenciar as tarefas e ministrar a *sprint review meeting*, ou seja, a reunião do final da sprint onde são realizadas as demonstrações ao cliente *Project Owner*, a retrospectiva da sprint e o planejamento das próximas.

A maior parte dos princípios citados foram utilizados no desenvolvimento do projeto. As sprints possuíam entre uma e duas semanas, contavam com encontros semanais para review e planejamento das demais. No final de cada reunião, as tarefas eram definidas e divididas entre os dois desenvolvedores, sendo que a cada semana, eles demonstravam uma funcionalidade implementada no sistema.

2.7 User Stories e Requisitos do SiGPos

Além dos princípios já citados de metodologias ágeis, também foi empregado ao projeto o uso de User Stories[18], ou *histórias de usuário*. Uma user story descreve, em poucas palavras, um dos requisitos do sistema de acordo com o ponto de vista de um usuário. Apesar de não ser parte fundamental do Scrum, essa prática costuma ser muito utilizada nos mais diversos tipos de projeto, pois facilita a comunicação entre o time e a compreensão de um assunto por seus integrantes. Como um produto desenvolvido com Scrum é descrito por itens no Product Backlog, cada um deles pode ser representado por uma User Story. A partir da seção 4, as histórias de usuário empregadas no projeto serão descritas com mais detalhes, incluindo suas implementações.

Além disso, o Módulo do Aluno Especial foi construído com base em um documento

interno da equipe do SiGPos chamado *Documento de Especificação de Requisitos SiGPos*, de 2016, que especifica todos os requisitos dos módulos planejados na época, de forma que cada módulo tenha, descrito nesse documento, todas as suas funcionalidades obrigatórias, opcionais e desejáveis. Como o modelo não passa por atualizações desde sua criação, alguns requisitos extras foram adicionados.

Os requisitos foram classificados utilizando uma prática muito comum em Engenharia de Software: a divisão entre requisitos funcionais e não-funcionais[19][20]. Essa prática, busca descrever os requerimentos do usuário e as features do produto (funcionais) e também as expectativas do usuário e as propriedades do produto (não-funcionais). Na prática, os primeiros descrevem as funcionalidades do sistema (login, regras de negócio, páginas do sistema), enquanto que os últimos definem propriedades e restrições do sistema (segurança, desempenho, controle de acesso, persistência de dados). No caso do SiGPos, os requisitos são:

Requisitos Funcionais (RF)

1. [RF01] Cadastro no sistema: O candidato deve ser capaz de se cadastrar no sistema utilizando seu e-mail. (Obrigatório)
2. [RF02] Envio de documentos: O candidato deve ser capaz de enviar, alterar e visualizar arquivos relativos à sua inscrição, como históricos e carta de apresentação. (Obrigatório)
3. [RF03] Ver disciplinas ofertadas no semestre: O candidato pode visualizar as disciplinas disponíveis para matrícula no semestre. (Obrigatório)
4. [RF04] Submissão de pedido de matrícula em disciplina: O candidato pode solicitar matrícula em uma disciplina disponível para inscrição. (Obrigatório)
5. [RF05] Relatório de status dos pedidos de matrícula em disciplinas: O candidato pode consultar o status do pedido de matrícula nas disciplinas que solicitou inscrição. (Desejável)
6. [RF06] O candidato a aluno especial pode se inscrever em no máximo 3 disciplinas. (Desejável)
7. [RF07] O sistema deve enviar um e-mail de confirmação antes de aceitar o cadastro do usuário. (Obrigatório)
8. [RF08] Alteração de cadastro: o candidato deve ser capaz de alterar informações sobre seu cadastro após aprovação.(Obrigatório)
9. [RF09] Desistência de pedido de matrícula nas disciplinas: O candidato pode remover um pedido de matrícula (desistência) caso o mesmo ainda esteja sendo avaliado. (Opcional)
10. [RF10] Histórico de pedidos de matrícula anteriores: O candidato pode consultar o resultado dos pedidos de matrículas de semestres anteriores. (Opcional)

11. [RF11] Gerar relatório de inscritos por disciplina: O avaliador pode gerar um relatório com resumo dos candidatos que solicitaram inscrição por disciplina. (Obrigatório)
12. [RF12] Aceitar ou recusar pedido de matrícula do aluno: O avaliador pode aceitar ou recusar o pedido de matrícula de um candidato. (Obrigatório)
13. [RF013] Informar candidato sobre resultado: O sistema deve enviar uma mensagem para o e-mail do candidato informando mudanças no status do seu pedido de matrícula. (Desejável)
14. [RF014] Manutenção de disciplinas disponíveis para alunos especiais: O coordenador do processo seletivo deve cadastrar, editar e remover disciplinas disponíveis no processo seletivo, bem como alterar seus detalhes. (Obrigatório)
15. [RF015] Associar um tutor para um aluno: O coordenador deve atribuir um tutor para o aluno aceito como aluno especial. (Opcional)
16. [RF016] Consultar disciplinas: O aluno especial pode consultar as disciplinas que foram aceitas, recusadas, ou estão em análise, bem como as já cursadas. (Desejável)
17. [RF017] Consultar dados do tutor: O aluno especial, pode consultar os dados do seu tutor, tais como nome, sala e e-mail. (Opcional)
18. [RF018] Atualizar dados dos alunos: O sistema deve atualizar automaticamente os dados dos alunos. (Obrigatório)
19. [RF019] Acesso a dados do aluno: Os gerentes do sistema podem consultar os dados dos alunos especiais. (Desejável)

Requisitos Não-funcionais (RNF)

1. [RNF01] O sistema deve manter os dados e arquivos fornecidos pelo candidato, de modo que o mesmo não tenha que preencher os dados novamente no caso de uma nova inscrição.
2. [RNF02] O sistema deve possuir autenticação, de forma a prover segurança para os usuários e uma forma de controlar o acesso ao sistema.
3. [RNF03] O módulo deve possuir interface para conexão com o sistema já existente da DAC.

3 Visão geral da solução desenvolvida

3.1 Escopo da Solução

Os requisitos que envolvem a parte de cadastro e envio de e-mail de segurança são novos em relação ao sistema Castelo, sendo essas funcionalidades exclusivas do SiGPos. Como citado na seção 2.2, o sistema antigo possuía poucas validações de dados e não validava e-mails,

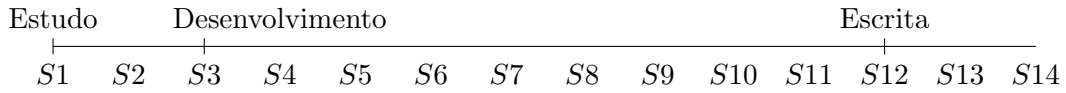


Figura 4: Linha do tempo semanal do desenvolvimento deste trabalho

CPFs e endereços. Além disso, para esse projeto, o escopo de requisitos foi limitado e os desenvolvedores focaram nas funcionalidades que tangem o aluno especial como usuário, enquanto que as funcionalidades administrativas serão desenvolvidas em projetos futuros de outros alunos, como descrito na seção 7.2. Dessa forma, os requisitos que foram colocados em prática são: RF01 até RF08 e RNF01 até RNF03.

Quanto ao desenvolvimento do módulo, a figura 4 retrata uma linha do tempo aproximada do projeto e mostra qual etapa estava sendo desenvolvida em uma determinada semana. Nas semanas S1 e S2, os desenvolvedores focaram nos estudos, visando a compreensão das características do SiGPos e buscando aprender os frameworks Spring Boot e VueJS. A partir da terceira semana, o desenvolvimento foi iniciado, focando na tela de cadastro nas semanas S3 à S6 e na tela inicial do candidato da S7 à S11. Da S12 em diante, os alunos ficaram focados em redigir o texto final do PFG.

3.2 Arquitetura do SiGPos

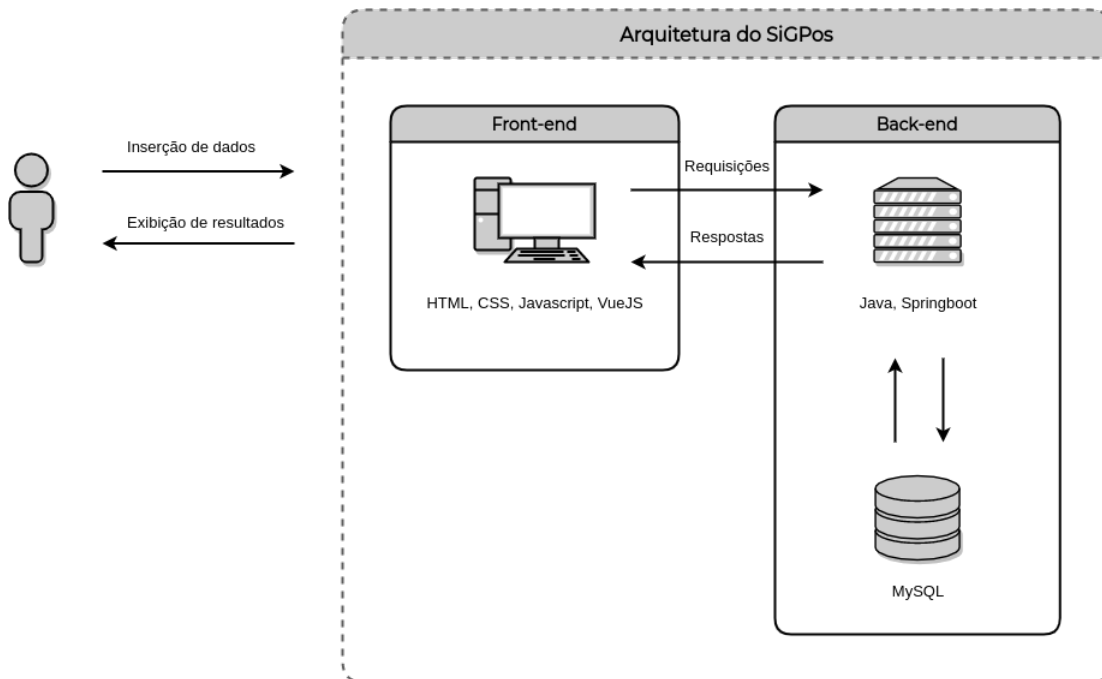


Figura 5: Diagrama de Arquitetura do SiGPos

A arquitetura do sistema foi desenvolvida conforme a representação na figura 5, composta por dois módulos, aqui denominados front-end e back-end. O primeiro módulo é responsável pela interface com o usuário e por exibir na tela os resultados de suas requisições, enquanto o segundo módulo é responsável pelas regras de negócio do sistema, bem como pela comunicação com o banco de dados para armazenamento das informações de cada candidato a aluno especial. Os módulos foram desenvolvidos, respectivamente, na linguagem JavaScript, utilizando o framework VueJS e na linguagem Java, utilizando o framework Spring Boot.

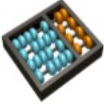
4 User Story 1 - Candidatar a Aluno Especial pela primeira vez

A User Story gerada durante o projeto que se refere a essa seção é: "Como um candidato a Aluno Especial, eu quero poder me **cadastrar no sistema** pela primeira vez, para que eu possa **enviar um pedido** de me tornar um Aluno Especial em uma ou mais disciplinas" e os requisitos implementados foram: RF01, RF07, RNF01 e RNF02.

Foi realizada uma adaptação do formulário do sistema Castelo (Seção 2.2), seguindo o layout do SiGPos, com modificações que visavam simplificar o preenchimento do usuário e evitar erros de digitação e discrepância de dados. Um overview da página de inscrição do candidato a aluno especial está na figura 6.

As principais mudanças da tela de cadastro desenvolvida em relação ao sistema antigo são:

- Criação de um cadastro de fato, dado que o sistema Castelo não possuía essa funcionalidade.
- Utilização da API da *ViaCEP*[21] para a inserção dos endereços do usuário. Ao pressionar TAB após a inserção do CEP, a API completa o nome da rua, a cidade e o estado para o candidato. Dessa forma, erros de digitação ou abreviações de endereço são evitados. O resultado está na figura 7.
- Caso o candidato selecione seu país de origem como Brasil, será requisitado o número do seu CPF. Do contrário, será requisitado o número do seu RNE/CRNM, o Registro Nacional de Estrangeiros, caso o usuário não possua nacionalidade brasileira. Os resultados estão nas figuras 8 e 9.
- A remoção da obrigatoriedade do candidato adicionar dois endereços, um em Campinas e o de sua cidade natal, se houver, tornando essa funcionalidade opcional.
- Criação de campos com a tag *select* do Vue, que fornecem uma lista de opções para o usuário, evitando erros de digitação. Um dos exemplos está na escolha da instituição de ensino superior cursada pelo candidato, que apresenta uma lista com as universidades mais comuns. Essa funcionalidade também foi utilizada em diversos campos de perguntas pessoais, como objetivo da inscrição, UF do candidato e país.



Inscrição de aluno especial - IC Unicamp

Dados pessoais

RA (se tiver)

Nome completo*

E-mail*

Senha*

Pais de origem*

Brasil ▼

CPF*

Figura 6: Página de inscrição do candidato a aluno especial

CEP*

01001-001

UF*

SP ▼

Cidade*

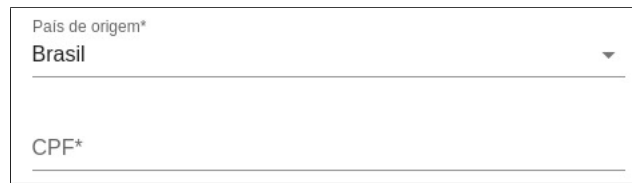
São Paulo

Rua/Avenida*

Praça da Sé

Figura 7: API da ViaCEP completando o endereço

- A inclusão de máscaras no front-end, através da biblioteca *v-mask*[22] do Vue, e verificações no back-end para garantir a validação dos dados. Dessa forma, o usuário é forçado a digitar um e-mail válido, um CPF que exista e que possua 11 dígitos, todas as datas no formato *dd/mm/yyyy*, telefones com DDD e o CEP no formato *dddd-dd*, por exemplo. Caso contrário, o sistema não permite seu avanço.



Pais de origem*
Brasil ▼

CPF*

Figura 8: Candidato brasileiro



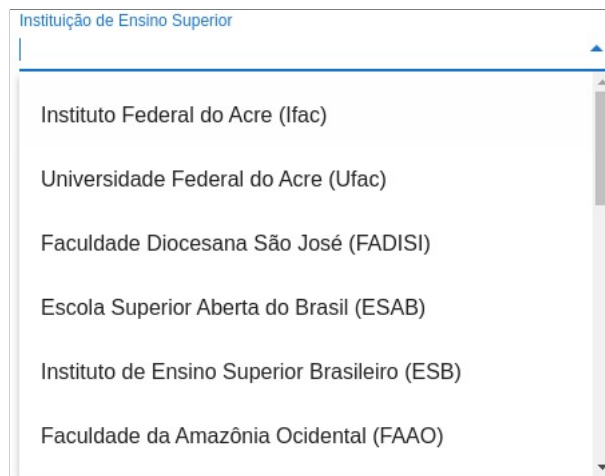
Pais de origem*
Afeganistão ▼

Adquiriu nacionalidade brasileira?*

Não ▼

RNE/CRNM*

Figura 9: Candidato estrangeiro sem nacionalidade brasileira



Instituição de Ensino Superior

- Instituto Federal do Acre (Ifac)
- Universidade Federal do Acre (Ufac)
- Faculdade Diocesana São José (FADISI)
- Escola Superior Aberta do Brasil (ESAB)
- Instituto de Ensino Superior Brasileiro (ESB)
- Faculdade da Amazônia Ocidental (FAAO)

Figura 10: Lista de seleção de instituição de ensino superior

- Após o preenchimento do formulário de inscrição, o candidato recebe um e-mail de confirmação com um token único e apenas será capaz de acessar o sistema após sua validação.
- Busca no banco de dados do e-mail fornecido pelo candidato. Caso ele já exista no banco, o usuário não poderá se cadastrar novamente e um novo token será enviado para ele, conforme mostrado na figura 12. O e-mail recebido pelo candidato está representado na figura 11.

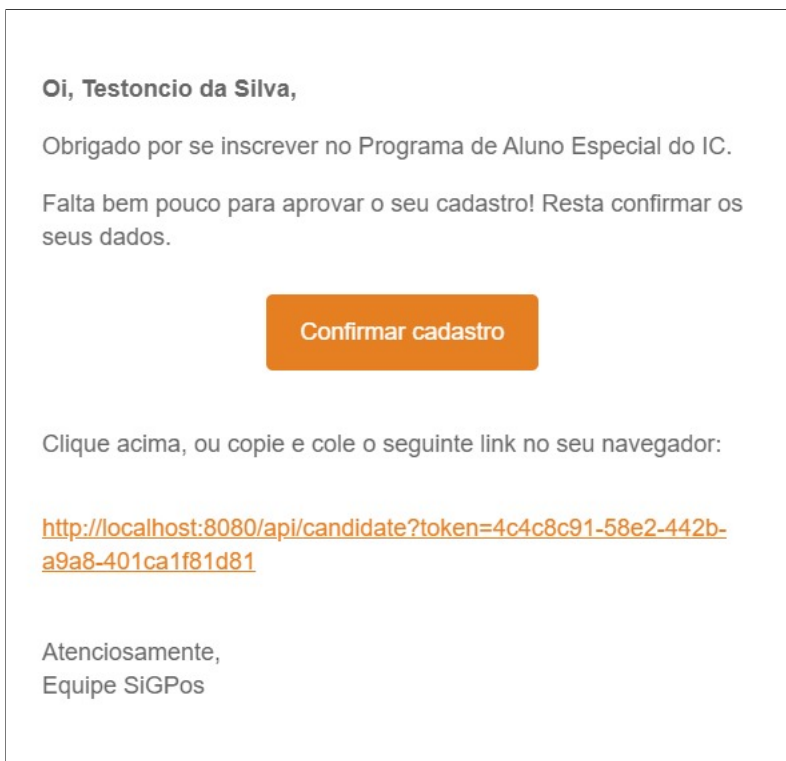


Figura 11: E-mail de confirmação enviado ao candidato

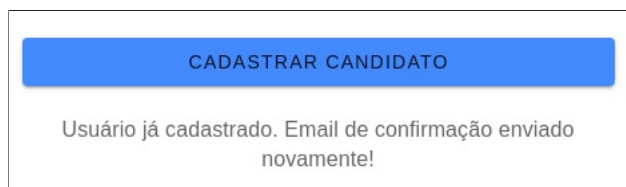


Figura 12: Aviso de usuário já cadastrado

5 User story 2 - Candidatar a Aluno Especial pela segunda vez

A User Story referente a essa seção é "Como um candidato a Aluno Especial, eu quero poder **complementar meu cadastro**, caso eu já tenha me cadastrado anteriormente, para que eu **não precise preencher o formulário** de inscrição todos os semestres" e os requisitos implementados foram: RF02 até RF06, RNF01 até RNF03.

No módulo de aluno especial criado neste projeto, foi implementada a persistência de dados. Dessa forma, o cadastro do usuário se mantém e ele não precisa preencher seus dados novamente nos semestres seguintes. Partindo do login, o candidato aceito será redirecionado à página inicial do candidato, disposta na figura 13. Nela, ele poderá selecionar nos menus

laterais, algumas opções que lhe permitem complementar seu cadastro.

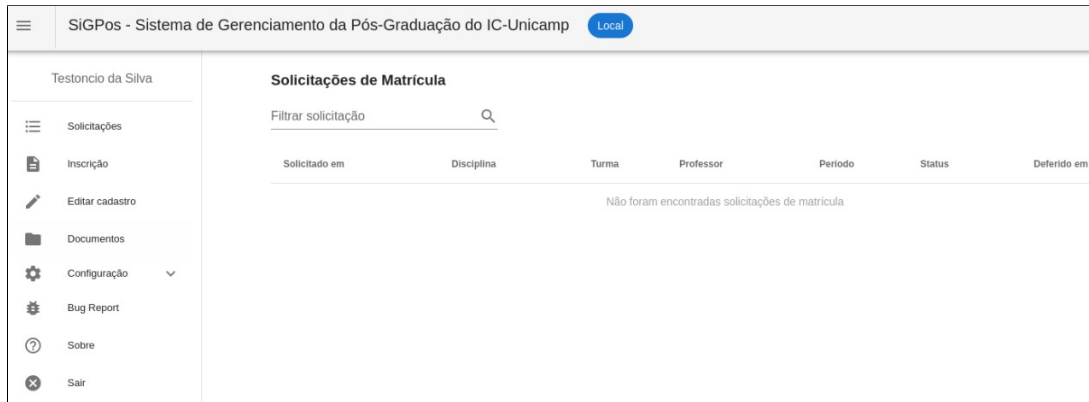


Figura 13: Tela inicial do candidato

No menu de *Inscrição*, retratado na figura 14, o candidato poderá selecionar as disciplinas em que deseja pleitear uma vaga, adicionar seus dados do POSCOMP[23] (Exame Nacional para Ingresso na Pós-Graduação em Computação), caso possua e justificar o motivo da sua inscrição. Além disso, tanto nesse menu, quanto na aba de "Documentos", o candidato pode fazer o upload dos documentos necessários, como histórico e carta de recomendação.

Inscrição

Área para realização de inscrição na modalidade de aluno especial, submissão de documentos pertinentes e informações complementares necessárias à solicitação de matrícula.

Documentos

Submissão de documentação prevista no regulamento. Ex.: Histórico, carta de recomendação, cursos externos e etc. Obs.: São permitidos apenas documentos no formato pdf com o tamanho máximo de 8mb.

Opções

Selecione as disciplinas, com a mais prioritária à esquerda e... Objetivo da inscrição 0 / 200

POSCOMP

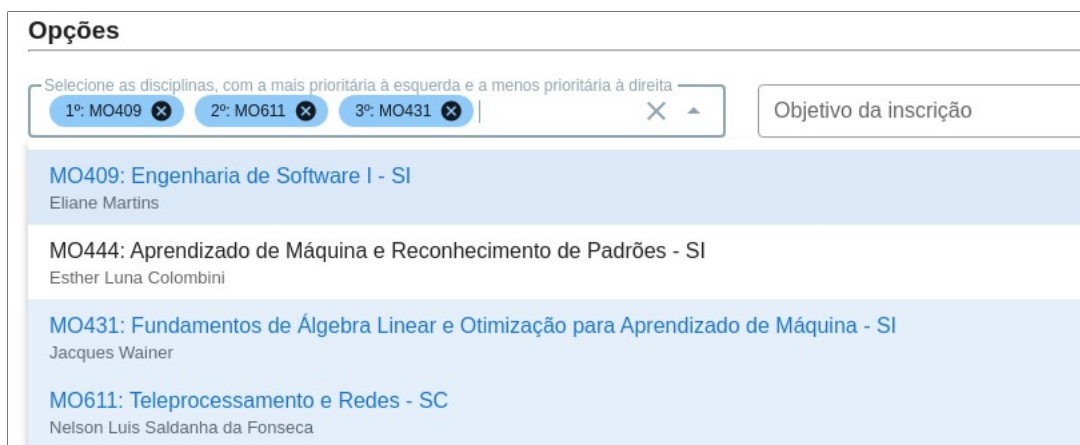
O [POSCOMP](#) é um exame nacional para ingresso na pós-graduação em Ciência da Computação. O exame é aplicado anualmente no segundo semestre. Recomendamos a todos os candidatos a aluno especial já graduados que façam o POSCOMP. Como o exame é aplicado apenas no segundo semestre, sabemos que vários candidatos não tiveram a oportunidade de realizá-lo e, por isso mesmo, o POSCOMP não é considerado obrigatório para avaliação. No entanto, recomendamos que todos os candidatos se programem para realizar o POSCOMP na próxima oportunidade possível.

Número de inscrição Ano de Realização :

Figura 14: Tela de inscrição do candidato

Para a tabela de disciplinas, a equipe do SiGPos desenvolveu uma API interna que se

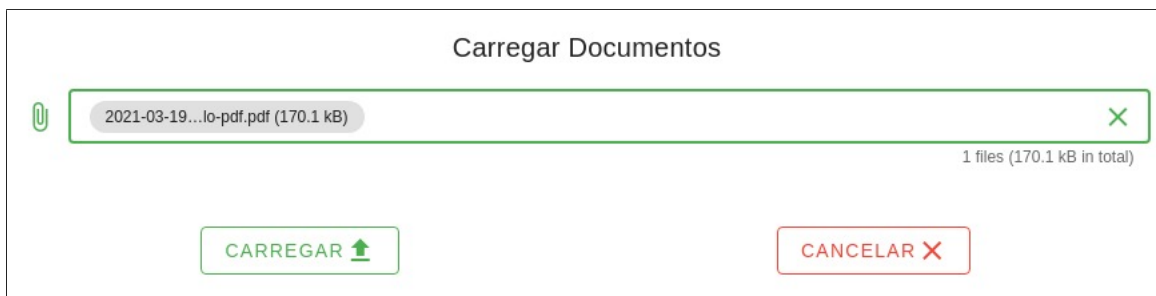
comunica com a DAC. Assim, o aluno pode escolher as matérias que mais lhe interessam, de acordo com cada semestre, na ordem de preferência, conforme demonstrado na figura 15. Já para o envio de documentos, a equipe também desenvolveu uma API que trata da comunicação entre o SiGPos e o Google Drive, armazenando e gerenciando o controle de acesso aos arquivos, como visto na figura 16.



The screenshot shows a web interface titled "Opções". At the top, there is a selection bar with the instruction "Selecione as disciplinas, com a mais prioritária à esquerda e a menos prioritária à direita". The bar contains three selected courses: "1º: MO409", "2º: MO611", and "3º: MO431", each with a close button (X). To the right of the bar is a button labeled "Objetivo da inscrição". Below the bar is a list of course options:

- MO409: Engenharia de Software I - SI
Eliane Martins
- MO444: Aprendizado de Máquina e Reconhecimento de Padrões - SI
Esther Luna Colombini
- MO431: Fundamentos de Álgebra Linear e Otimização para Aprendizado de Máquina - SI
Jacques Wainer
- MO611: Teleprocessamento e Redes - SC
Nelson Luis Saldanha da Fonseca

Figura 15: Tela de escolha das disciplinas em ordem de preferência do candidato



The screenshot shows a web interface titled "Carregar Documentos". It features a file upload area with a paperclip icon on the left and a close button (X) on the right. A file named "2021-03-19...lo-pdf.pdf (170.1 kB)" is shown as uploaded. Below the upload area, there are two buttons: "CARREGAR" with an upward arrow icon and "CANCELAR" with a close icon (X). The text "1 files (170.1 kB in total)" is displayed at the bottom right of the upload area.

Figura 16: Tela de envio dos documentos para inscrição

6 User story 3 - Acessar painel do sistema para visualizar informações

A User Story referente a essa seção é "Como um candidato a Aluno Especial, eu quero ser capaz de **alterar meus dados**, caso necessário" e os requisitos implementados foram: RF08 e RNF01.

No sistema Castelo, não havia persistência de dados. Dessa forma, caso um usuário precisasse trocar alguma informação depois de realizar seu cadastro, seria necessário contactar o suporte do sistema ou reenviar as informações. Portanto, o desenvolvimento do Módulo

do Aluno Especial do SiGPos buscou resolver essa questão através da criação de uma aba de edição de cadastro.

Clicando em "Editar Cadastro", um formulário similar ao de cadastro de aluno especial é mostrado ao usuário, porém com seus dados já preenchidos. Se o mesmo desejar, é possível alterar qualquer um dos campos presentes e, ao clicar no botão de editar cadastro, os dados são verificados novamente e atualizados no banco, mantendo a persistência sem prejudicar a segurança do sistema, pois o back-end assegura que apenas o candidato pode alterar seus próprios dados. A edição está retratada na figura 17.

Editar cadastro de aluno especial - IC Unicamp

Dados pessoais

RA (se tiver)
AAAAAAA

Nome completo*
Testoncio da Silva

E-mail*
marcelobmartins211@gmail.com

Senha*

Pais de origem*
Brasil

CPF*
892.824.390-46

Figura 17: Tela de edição do cadastro do candidato

7 Resultados obtidos

7.1 Avaliação da Solução

Segundo o plano de trabalho enviado no início do semestre, era esperado que ao final do projeto o SiGPos fosse capaz de: (1) cadastrar Alunos Especiais, (2) auxiliá-los quanto às rotinas acadêmicas do IC, (3) facilitar e agilizar as atividades, tanto da administração quanto dos próprios alunos e (4) manter uma base de dados própria com informações que podem ser utilizadas para melhorar ainda mais a qualidade do programa de Aluno Especial.

O item (1) foi contemplado na seção 4, através do formulário de cadastro construído. Os itens (2) e (3) foram contemplados através da remoção da necessidade de preencher dados similares todos os semestres, também mencionada na seção 4, bem como a dinamização

de campos e remoção de informações redundantes do formulário Castelo, mencionado na seção 2.2. Quanto ao item (4), o sistema não trouxe, até o momento, vantagens gerenciais à administração do ponto de vista de análise de dados, apenas para armazenamento destes, que era o previsto no início do desenvolvimento.

O projeto foi desenvolvido ao longo de 14 semanas, sendo dedicadas em torno de 13 horas semanais de trabalho, totalizando 182 horas de trabalho. Destas, estão incluídas: duas horas semanais de reuniões técnicas e reuniões de alinhamento de projeto e 11 horas de estudos das tecnologias, escrita de código, implementação e versionamento.

7.2 Trabalhos Futuros

Como citado na seção 3, este projeto teve escopo limitado e, portanto, algumas funcionalidades serão desenvolvidas no futuro pela equipe do SiGPos, ou por outros alunos de graduação e iniciação científica.

Futuramente, fora do escopo deste trabalho, espera-se que a equipe do SiGPos possa utilizar os dados dos alunos especiais cadastrados para operações como envio de e-mails de oportunidades de bolsa, análise de áreas curriculares dos alunos e microgerenciamento por parte dos professores na hora de avaliar se o aluno está apto para cursar a disciplina, a fim de ter uma solução mais robusta e completa para o problema atacado.

Além disso, os requisitos fora do escopo deste projeto deverão ser implementados nos próximos, sendo que uma das funcionalidades do sistema Castelo que não foi contemplada neste desenvolvimento é a planilha de controle do administrador, que é responsável por sumarizar os resultados das inscrições e concatenar os dados dos candidatos em um arquivo, para que os responsáveis decidam quais alunos conseguirão determinada vaga. Atualmente, essa escolha é manual e, portanto, um dos trabalhos futuros será gerar essa planilha a fim de automatizar a seleção dos candidatos.

8 Dificuldades Encontradas e Lições Aprendidas

Durante a fase inicial de familiarização com o sistema, houve dependência da gestão de conhecimento do técnico responsável pelas implementações anteriores, devido à documentação praticamente inexistente e ao código bastante redundante e desorganizado. Para resolver esse problema, foi sugerido adicionar à rotina de reuniões semanais uma reunião extra, de uma hora de duração, para eventuais dúvidas e demonstrações com o técnico, o que foi suficiente para resolver as questões necessárias. Além disso, o técnico se mostrou sempre solícito a resolver quaisquer dúvidas no quesito passagem de conhecimento, o que foi primordial. Uma das lições aprendidas está na questão organizacional do código, dado que o desenvolvimento deve ser realizado de forma clara e objetiva, a fim de evitar gasto extra de tempo com reuniões de gestão de conhecimento.

Durante o desenvolvimento da funcionalidade de cadastro, surgiram dúvidas quanto ao conteúdo dos campos, já que o responsável pelo sistema Castelo (seção 2.2) não se fez presente nas reuniões e se manteve inacessível. Esse fato culminou em aproximadamente duas semanas de retrabalho, pois os campos implementados inicialmente precisaram ser modificados ou incluídos após o parecer deste, tanto na interface, quanto na criação das

tabelas no banco de dados. Outra lição aprendida pela equipe diz respeito à disponibilidade de terceiros, que devem ser contactados e incluídos nas reuniões desde o início, buscando não retroceder o desenvolvimento após o início deste.

Por fim, o sistema de repositório do IC utilizado, o GitLab [24], por diversas vezes esteve indisponível, não sendo possível consultar o versionamento de código e tampouco subir as mudanças implementadas. A solução adotada foi a realização de mudanças de forma local e aguardo por parte da equipe do retorno do GitLab, que geralmente levava entre um a três dias. Fica de lição à equipe ter uma ferramenta de compartilhamento de código alternativa entre os desenvolvedores, para casos de dificuldade de acesso ao sistema ou emergências.

9 Conclusões

Conforme as funcionalidades implementadas, pode-se concluir que a incorporação do Módulo do Aluno Especial ao SiGPos contribui com a unificação dos sistemas de controle do Instituto de Computação da Unicamp e diminuem a burocracia, centralizando as responsabilidades em uma única equipe. Dessa forma, esse novo componente será capaz de evitar retrabalho dos responsáveis e realizar o papel do sistema Castelo, cuja utilização, nos próximos semestres, não será mais necessária.

Além disso, é notável o aperfeiçoamento de design desde o sistema Castelo, visto que o novo módulo é composto por uma interface mais moderna e de acordo com os padrões atuais, além de utilizar tecnologias recentes e líderes de mercado, diminuindo o risco das funcionalidades do sistema tornarem-se defasadas em um futuro próximo. Somado a isso, a implementação de cadastro de usuários e armazenamento das informações dos mesmos no banco simplificam a inscrição de alunos especiais nas matérias dos próximos semestres, evitando a inserção de informações similares em cada matrícula. Por fim, a integração com o sistema da DAC e as diversas verificações realizadas com a autenticação tornaram a inscrição de alunos especiais mais segura e robusta.

Até a conclusão deste projeto, o módulo ainda não está disponível para os usuários. No entanto, a equipe pretende torná-lo a ferramenta oficial de controle de alunos especiais já no segundo semestre de 2021. Com isso, o time do SiGPos será capaz de realizar testes e colher opiniões sobre o novo sistema. Portanto, o uso futuro do módulo será a melhor maneira de avaliar a qualidade da solução desenvolvida.

Referências

- [1] SiGPos, <https://sigpos.ic.unicamp.br/#/>, 10 07 2021.
- [2] Ficha de Inscrição do Aluno Especial, <https://castelo.ic.unicamp.br/AlunoEspecial?id=2550fb1d-5cb8-473a-b238-d33d2f63d9cb>, 10 07 2021.
- [3] Site oficial do Java, <https://www.java.com/pt-BR/>, 10 07 2021.
- [4] Popularidade das linguagens de programação, <https://pypl.github.io/PYPL.html>, 10 07 2021.

- [5] Sobre o Java, <https://www.edureka.co/blog/what-is-java/>, 10 07 2021.
- [6] Site oficial do Spring, <https://spring.io/>, 10 07 2021.
- [7] Sobre o Spring, <https://spring.io/projects/spring-framework>, 10 07 2021.
- [8] Sobre o Spring Boot, <https://spring.io/projects/spring-boot>, 10 07 2021.
- [9] Spring Boot Initializer, <https://start.spring.io/>, 10 07 2021.
- [10] Site oficial JavaScript, <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>, 10 07 2021.
- [11] D. Flanagan, *JavaScript - The Definitive Guide (6 ed.)*, pp. 1 (Maio 2021).
- [12] Site oficial do VueJS, <https://vuejs.org/>, 10 07 2021.
- [13] Site oficial do MySQL, <https://www.mysql.com/>, 10 07 2021.
- [14] Repositório oficial do MySQL no GitHub, <https://github.com/mysql/mysql-server>, 10 07 2021.
- [15] A pilha LAMP explicada, <https://www.ibm.com/cloud/learn/lamp-stack-explained>, 10 07 2021.
- [16] Site oficial do Manifesto Ágil, <https://agilemanifesto.org/iso/ptbr/manifesto.html>, 10 07 2021.
- [17] Sobre o Scrum, <https://www.atlassian.com/br/agile/scrum>, 10 07 2021.
- [18] O que é User Story, <https://k21.global/tudo-sobre/agilidade/user-stories/o-que-e-user-story>, 10 07 2021.
- [19] Functional vs Non-Functional Requirements: The Definitive Guide, <https://qracorp.com/functional-vs-non-functional-requirements/#:~:text=While%20functional%20requirements%20define%20what,the%20system%20should%20do%20it.&text=Non%2Dfunctional%20requirements%20define%20system,that%20affect%20the%20user%20experience.>, 10 07 2021.
- [20] Requisitos Funcionais e Requisitos Não-Funcionais, https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf, 10 07 2021.
- [21] API ViaCEP, <https://viacep.com.br/>, 10 07 2021.
- [22] V-Mask, <https://www.npmjs.com/package/v-mask>, 10 07 2021.
- [23] Site oficial POSCOMP, https://www.sbc.org.br/index.php?option=com_content&view=article&layout=edit&id=458, 10 07 2021.
- [24] Site oficial GitLab IC, <https://gitlab.ic.unicamp.br/>, 10 07 2021.