

Desenvolvimento do SAAMS - Sistema para assistência estudantil e acompanhamento acadêmico

Trabalho de Conclusão do Curso de Tecnologia em Sistemas para Internet

Alex Vargas Cardoso
Orientador: Prof. André Peres

¹Instituto Federal do Rio Grande do Sul
Campus Porto Alegre
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

alex.cardoso@poa.ifrs.edu.br, andre.peres@poa.ifrs.edu.br

Resumo. *O presente trabalho apresenta o estudo, concepção e desenvolvimento de um sistema computacional que automatize os processos necessários às atividades de assistência estudantil e acompanhamento acadêmico realizadas no Instituto Federal do Rio Grande do Sul – campus Porto Alegre. Desenvolvido em arquitetura cliente/servidor, para ambiente web, utilizando recursos da plataforma Java e banco de dados PostgreSQL. Os resultados obtidos pelo presente trabalho cobrem a modelagem e implementação de estrutura básica do sistema e implementação de casos de uso referentes a acompanhamento acadêmico.*

Palavras-chave: sistema de gestão para assistência estudantil, gestão de auxílios acadêmicos, acompanhamento acadêmico.

1. Introdução

A equipe da Coordenadoria de Acompanhamento Acadêmico (CAAd) do Instituto Federal do Rio Grande do Sul (IFRS) – campus Porto Alegre, identificou através de sua atuação junto aos alunos da instituição a necessidade de que um sistema informatizado possa acolher seus processos de assistência estudantil e acompanhamento acadêmico. O referido setor solicitou auxílio para pesquisa, implantação, ou desenvolvimento de um *software* que atenda suas necessidades operacionais diárias. Tal solução é aguardada pela equipe desde meados de 2014, quando foi demonstrada necessidade junto ao setor de informática do campus.

Todos os processos da CAAd até o presente, são operacionalizados com apoio de planilhas eletrônicas ou documentos de texto. Segundo experiência da equipe, o uso de tais ferramentas mostra-se pouco produtor para realizar processos e controles, organizar, sistematizar, catalogar, e acessar informações de forma eficaz, levando inevitavelmente a operações repetitivas, informações conflitantes ou inconsistentes. Face a uma equipe reduzida e demanda crescente, necessitam apoio de um aplicativo que permita a atuar em duas frentes: na área de assistência estudantil, em processos como publicização de editais, inscrição, caracterização socioeconômica, classificação dos candidatos, concessão e manutenção de auxílios estudantis; e no acompanhamento acadêmico dos alunos da instituição, no monitoramento de aproveitamento e evasão, controle de atendimentos, gerenciamento do histórico sociopedagógico do aluno e, em especial daqueles indivíduos em situação de maior vulnerabilidade. Como fator de relevância, os processos entre equipes

análogas a CAAAd no IFRS, possuem organização e processos similares, e eventual solução que contemple o campus Porto Alegre, pode tornar-se alternativa viável para implantação nos demais câmpus da instituição.

O desenvolvimento dos trabalhos inicia desde uma pesquisa preliminar comparativa das características dos principais sistemas informatizados para assistência estudantil e acompanhamento acadêmico em uso nos Institutos Federais (IFs), até a concepção e desenvolvimento de um sistema próprio, o Sistema de Assistência Estudantil e Acompanhamento Acadêmico (SAAMS), desenvolvido em arquitetura cliente/servidor para uso em ambiente *web* e utilização voltada primariamente para estações de trabalho (*Desktops*).

Este trabalho divide-se da seguinte maneira: a Seção 2 apresenta uma contextualização histórica sobre os IFs, assistência estudantil, acompanhamento acadêmico e marcos regulatórios importantes. A seção 3 trata sobre as necessidades identificadas, na forma de uma visão geral dos requisitos que o sistema visa atender, bem como a metodologia adotada para a obtenção dos requisitos. Na seção 4 são identificados os softwares que foram comparados em busca de uma solução preexistente. Em seguida (seção 5), detalha-se o projeto do sistema desenvolvido: modelagem, processo de desenvolvimento, arquitetura, tecnologias empregadas, resultados já obtidos e discussão. Por último, as considerações finais são abordadas na seção 6.

2. Entendendo o macrocontexto: os Institutos Federais, a assistência estudantil e o acompanhamento acadêmico

De acordo com o Programa Nacional de Assistência Estudantil – PNAES [BRASIL, 2010]¹, a assistência estudantil é uma ação do Estado que busca democratizar e minimizar as desigualdades sociais, ampliar e garantir o acesso, bem como o êxito dos indivíduos nas instituições federais de ensino. Visando alcançar seus propósitos, realiza ações, projetos e programas focados no acompanhamento pedagógico universal e no apoio a estudantes em situação de vulnerabilidade socioeconômica.

Por sua vez, os IFs foram criados através de uma mudança iniciada em 2005 [PACHECO; PEREIRA; SOBRINHO, 2010], na qual os Centros Federais de Educação Tecnológica, as Escolas Técnicas e Agrotécnicas Federais e Escolas Técnicas vinculadas às Universidades Federais passaram a constituir-se na forma de organizações multicurriculares e multicampi, compartilhando estruturas organizacionais e administrativas.

Ações de assistência estudantil, antes mais comuns em Universidades Federais, passaram a fazer parte da realidade dos IFs após a inclusão destas instituições no PNAES. Tornou-se necessário que estes então criassem suas próprias políticas de assistência, conforme Taufick [2014]. Apesar de possuir ações de apoio desde sua criação em 2008, o IFRS teve aprovada sua política de assistência estudantil no final de 2013 [IFRS, 2013]. Desde então criou-se um processo de normatização das ações dos núcleos de assistência dos seus câmpus, que resultou na publicação da Instrução Normativa nº005/2015 pela Pró-reitoria de Ensino (PROEN) [IFRS, 2015b], de forma a estabelecer diretrizes para o funcionamento do Programa de Auxílios estudantis em toda a instituição.

Por acompanhamento acadêmico deve-se entender, segundo a definição da proposta

¹Institucionalizado através da Portaria nº39 de 2007 do Ministério da Educação, que viria a tornar-se o Decreto Federal 7.234/10

de trabalho da CAAd [IFRS, 2017], atividades que transcendam o combate a retenção e evasão, e que possibilitem acompanhamento integral do aluno e seu acolhimento, ou ainda, o apoio pedagógico com vistas a "acompanhar os sujeitos nos seus diferentes contextos, de forma a contribuir para a ampliação e consolidação da cidadania, promover a inclusão social, desenvolver ações de promoção de saúde e incentivar a participação e o respeito à diversidade"[IFRS, 2015a].

2.1. Caracterização da demanda

Segundo dados referentes ao mês de maio de 2018 que foram apurados dos sistemas acadêmicos SIGAA e SCA em uso, o campus possui aproximadamente 1900 alunos², e recebe cerca de 350 ingressantes por semestre³. Por volta de 75% desse público é proveniente da rede pública de ensino, e 55% com renda *per capita* familiar de até 1,5 salários mínimos, estes últimos compondo o principal público alvo da CAAd. Em dados divulgados pela equipe, por semestre são analisados ao redor de 400 processos de concessão de auxílios⁴, e o montante financeiro administrado anualmente aproxima-se de R\$1 milhão de reais⁵. São efetuados cerca de 30 atendimentos mensais, individuais ou em grupo, os quais desdobram-se em inúmeras ações. A CAAd possui uma equipe de 05 pessoas para realizar todas as suas atividades no campus. Ao todo, o IFRS possui 17 câmpus onde estudam por volta de 19 mil alunos⁶.

3. O modelo de assistência estudantil e acompanhamento acadêmico apresentado pela CAAd

Imaginou-se que após compreender o domínio e efetuar um levantamento de requisitos preliminar, pudesse ser identificado um *software* pré-existente que atendesse ao negócio. Contudo, uma vez levantados os principais aspectos necessários, e efetuada comparação das características dos principais softwares presentes nos IFs para a área, foi concluído que era necessário projetar uma solução customizada.

3.1. Metodologia de Levantamento de requisitos

A identificação dos requisitos empregou metodologia baseada na proposta de Presman [2009], fazendo uso das seguintes ferramentas: reuniões, *brainstormings* e análise documental, onde foram obtidas informações dos principais *stakeholders*, de regulamentações, dos editais de assistência estudantil referentes ao programa de auxílios, e de artefatos já em uso – planilhas e documentos, com finalidade de detalhar os processos e mapeá-los para requisitos de *software*.

As reuniões realizadas com a equipe da CAAd, contaram com a presença de uma assistente social, uma psicóloga e uma técnica em assuntos educacionais. Expuseram nestes encontros os processos que realizam e seus detalhamentos, regras do negócio, dados que armazenam, relatórios e processamentos que mais lhe são importantes e laboriosos, bem como expectativas funcionais e não funcionais relativas a solução necessária, como segurança e usabilidade.

²Alunos do Ensino Técnico Subsequente, Tecnológico e Superior.

³Edital de chamamento em <http://ingresso.ifrs.edu.br>, semestre 2018/1. Acesso em 01/06/2018.

⁴Resultado do Edital 064/2017 de 19/01/2018. Constam 256 renovações e 173 novos ingressantes.

⁵Valores empenhados relativos ao ano de 2017: R\$932,962,00.

⁶Sobre o IFRS - <https://ifrs.edu.br/institucional/sobre>. Acesso em 01/06/2018.

Um instrumento importante apresentado pela equipe para detalhar o funcionamento da assistência estudantil, foi a Instrução Normativa nº05/2015 da Pró-Reitoria de Ensino (PROEN), que é regulamento interno da instituição para o tema. Este documento foi recentemente atualizado através da expedição das Instruções Normativas 04/2018 e 05/2018 da PROEN.

3.2. Requisitos e regras de negócio identificados

Dos processos da CAAd que possuem maior volume de trabalho, complexidade e regras de negócio dizem respeito à execução da assistência estudantil e seu programa de auxílios. Em linhas gerais, trata-se de prover acesso a editais e informações complementares, bem como meios para que os alunos possam inscrever-se, preencham questionários de caracterização socioeconômica que os posicionará em listas de classificação para concessão dos auxílios. A concessão evidentemente somente se dará após constatar-se que os documentos apresentados estiverem em conformidade com as informações prestadas nos questionários, e após entrevista com a equipe, a qual deve possuir mecanismo de gerenciamento para revisar as respostas, classificar, conceder ou revogar benefícios. Deve-se ainda poder gerenciar os questionários, benefícios e suas configurações, bem como emitir relatórios diversos.

De igual importância, as funcionalidades para acompanhamento acadêmico levantadas baseiam-se principalmente em manter o agendamento e o cadastro das informações de atendimentos individuais ou em grupo, bem como de seus encaminhamentos. Também é necessário registrar o acolhimento inicial do aluno e notas em seu histórico pedagógico, manter uma lista com alunos que necessitem atenção especial por situação de vulnerabilidade, bem como da necessidade de gerar gráficos, relatórios e alertas de monitoramento. O monitoramento analisa principalmente a frequência e aproveitamento do aluno, com intuito de orientar ações mormente em relação a retenção, controle de evasão, e aumento do êxito dos estudantes [IFRS, 2013]. Alertas de monitoramento ainda são úteis para gerir a manutenção do aluno no programa de auxílios, uma vez que é obrigatório que mantenha-se frequente e com boas notas, bem como com vínculo ativo na instituição. Deve-se poder ainda manter no sistema arquivos sensíveis relacionados com o aluno, como pareceres pedagógicos, de saúde, etc.

O sistema deve prover ainda meios seguros de manutenção dos dados, uma vez que muitos deles tratam de situações pessoais e familiares de alunos, implementando métodos de autenticação e segurança bem como o gerenciamento de usuários e grupos que possuam acesso a áreas segmentadas do sistema. Deve ainda permitir configurar parâmetros relativos ao seu funcionamento, úteis principalmente de forma a alterar a aplicação das regras de negócio, como por exemplo, a maneira como os benefícios são disponibilizados em editais, seja por benefício unico ou benefícios separados. Do mesmo modo é importante que o sistema apresente mecanismo para obter dados ou comunicar-se com distintos sistemas de gestão acadêmica, de maneira a adequar-se a realidade atual do campus que é de transição entre sistemas, bem como implementar a comunicação com o principal sistema de gestão acadêmica em uso no campus atualmente. Ainda, que possa obter dados acadêmicos de um sistema, e de forma complementar, dados de monitoramento de outro que possua atualizações mais frequentes, de modo a oferecer melhor qualidade e acurácia nas informações, principalmente relativas a acompanhamento acadêmico. A listagem dos requisitos identificados são apresentados no Anexo B.

4. Estudo de soluções: softwares existentes

Os *softwares* foram comparados a nível de características gerais e funcionalidades presentes, e foram escolhidos dentre os que encontram-se em uso no cenário dos IFs na área de gestão acadêmica: o BAE desenvolvido pelo IFRS campus Canoas; o SUAP do Instituto Federal do Rio Grande do Norte (IFRN); SIA, da Universidade Federal de Rio Grande (FURG) em versão adaptada pela Reitoria do IFRS; AssisEXT, do Instituto Federal de Minas Gerais (IFMG); e por último o SIGAA, da Universidade Federal do Rio Grande do Norte (UFRN), em implantação no IFRS.

As características e descrição ou funcionamento das funcionalidades comparadas dos *softwares* foram obtidas a partir de manuais de usuário ou operação, na *internet* ou nos sites dos próprios desenvolvedores, e em alguns casos da utilização do software já presente na instituição ou de instalação específica para este fim. No Anexo A, são apresentadas os principais requisitos levantados para a solução proposta e a forma como os softwares pesquisados os implementam.

A dispensa dos softwares identificados deu-se por não realizarem em solução única os requisitos levantados, ou porque necessitariam de modificações variadas para atendê-los, o que mostrou-se pouco atraente frente a uma solução customizada. Cabe salientar ainda que as instruções normativas da PROEN citadas anteriormente, determinam diretrizes de negócio de forma bastante específica, o que em parte restringe a adoção de ferramenta genérica.

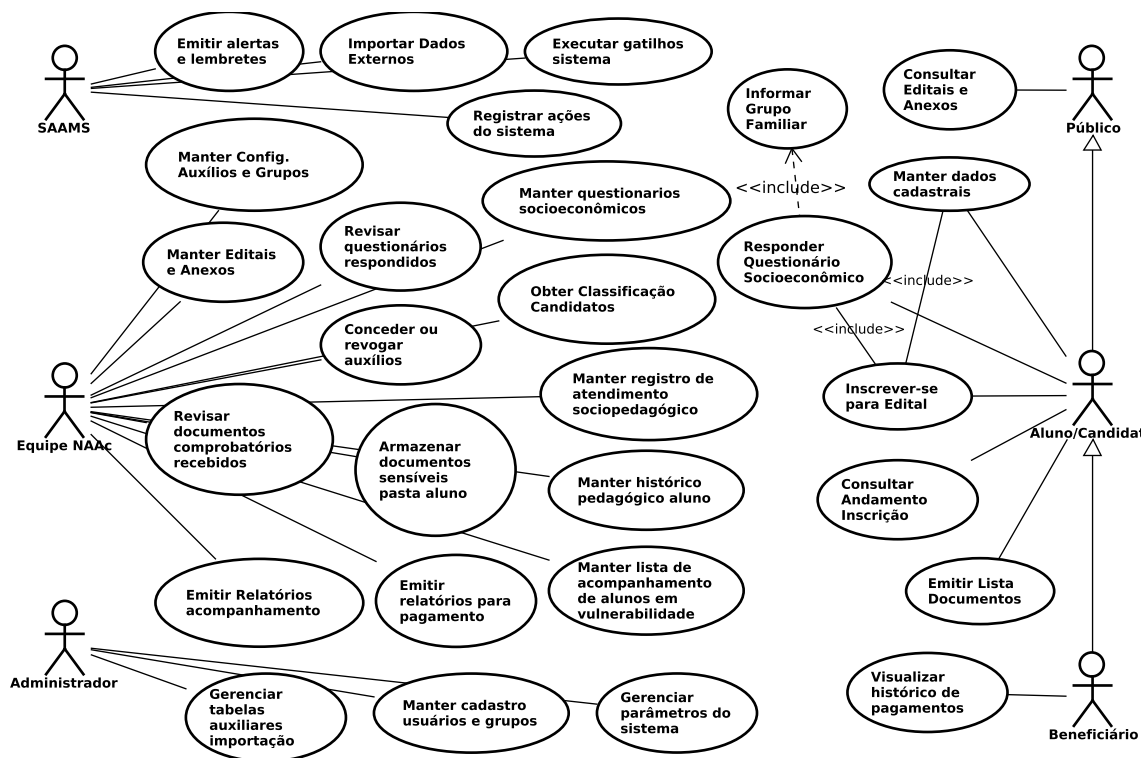


Figura 1. Caso de uso de alto nível do SAAMS

5. Desenvolvimento do SAAMS

O sistema desenvolvido apresenta uma solução customizada que contempla o modelo de negócio da CAAd, para assistência estudantil e acompanhamento acadêmico. Trata-se de um sistema predominantemente orientado a transação e entradas de usuário conforme categorizado por Pressman [2009], de arquitetura cliente/servidor em ambiente *web* e com foco de uso a partir de estações de trabalho.

5.1. Modelagem

Uma vez que a eliciação dos principais requisitos funcionais foi efetuada previamente para a comparação dos sistemas existentes, modelou-se o diagrama de caso de uso geral, como já apresentado na Figura 1. Procedeu-se com a identificação das regras de negócio, e a melhor compreensão do cenário viabilizou a modelagem do banco de dados, demonstrado em diagrama lógico de tabelas, vide Anexo C.

5.2. Processo de desenvolvimento

O modelo de desenvolvimento adotado baseia-se em incremento e iteração, utiliza práticas de Modelagem Ágil [AMBLER, 2004], como Modelagem Incremental, Documentação Contínua, *Just Barely Good Enough Models and Documents*⁷, de forma a entregar funcionalidades e obter uma versão inicial mais rapidamente. Assim, as funcionalidades são entregues em versões simplificadas e ao longo do ciclo de vida do software adicionados aperfeiçoamentos e controles mais elaborados.

Durante a fase de implementação foram conduzidas reuniões para validação e verificações funcionais, bem como de aceitação junto a equipe da CAAd.

5.2.1. Prototipação

Durante as fases de modelagem e implementação do sistema foi utilizada a técnica de prototipação com o uso do *framework Java Server Faces* (JSF). O uso dessa técnica e instrumento, permitiu a partir da utilização de componentes preconcebidos, construir, aprimorar e validar rapidamente a navegação entre as telas do programa, bem como as informações a serem apresentadas ao usuário. A vantagem de utilizar essa ferramenta, foi que uma vez produzidos os protótipos, puderam ser reutilizados na construção propriamente dita do aplicativo.

5.3. Arquitetura do sistema

A solução desenvolvida conta com uma área pública para acesso da comunidade em geral, onde editais do programa de auxílios devem estar publicamente disponíveis. Existem no sistema duas áreas de acesso restrito a usuários cadastrados: uma na qual podem ingressar a equipe da CAAd e administradores, e outra destinada a alunos autenticados. Cada área provê acesso a funções diferenciadas. As figuras 2 e 3 a seguir, demonstram respectivamente a área pública – onde podem ser consultados os editais e documentos anexos, e a capa da área restrita a alunos autenticados.

Procedimentos agendados possibilitam execuções de tarefas, como a obtenção de dados sistemas acadêmicos, lançamentos dos auxílios em intervalos regulares, emissão de

⁷JBGE - "modelos e documentos apenas bons o suficiente"(tradução nossa)

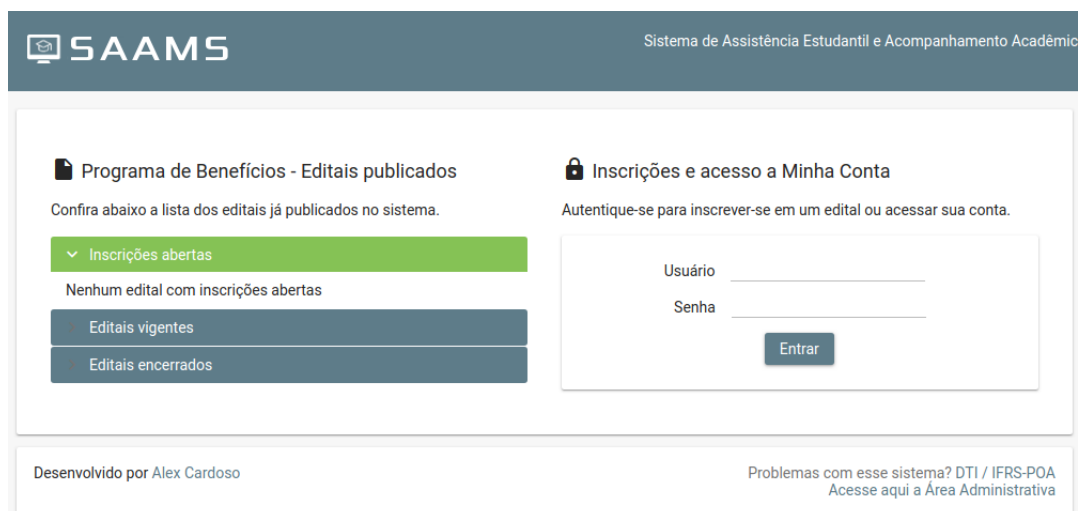


Figura 2. Área Pública do sistema



Figura 3. Página inicial da Área Restrita a Alunos

alertas ou notificações. A obtenção de dados a partir de outros sistemas, foi estruturada de forma que possa-se agregar módulos de comunicação específicos para cada sistema a obter dados. Na figura 4, a seguir, é apresentado o conceito arquitetural de implantação do sistema.

5.3.1. Estrutura do projeto e tecnologias adotadas

O projeto do software é estruturado em camadas, utilizando o padrão *Model-View-Controller* (MVC) onde cada camada respectivamente é responsável por visão, controle e modelo de dados [GAMMA et al., 2006].

Para sua implementação, optou-se por fazer uso de uma linguagem de amplo uso no mercado, orientada a objetos e que oferecesse boas ferramentas de produtividade. A linguagem Java⁸ foi escolhida para este fim. Por motivos semelhantes, para a persistência de dados foi adotado o banco de dados PostgreSQL⁹, por permitir inclusive a programação

⁸Java - <https://java.com/>

⁹PostgreSQL - <https://www.postgresql.org>

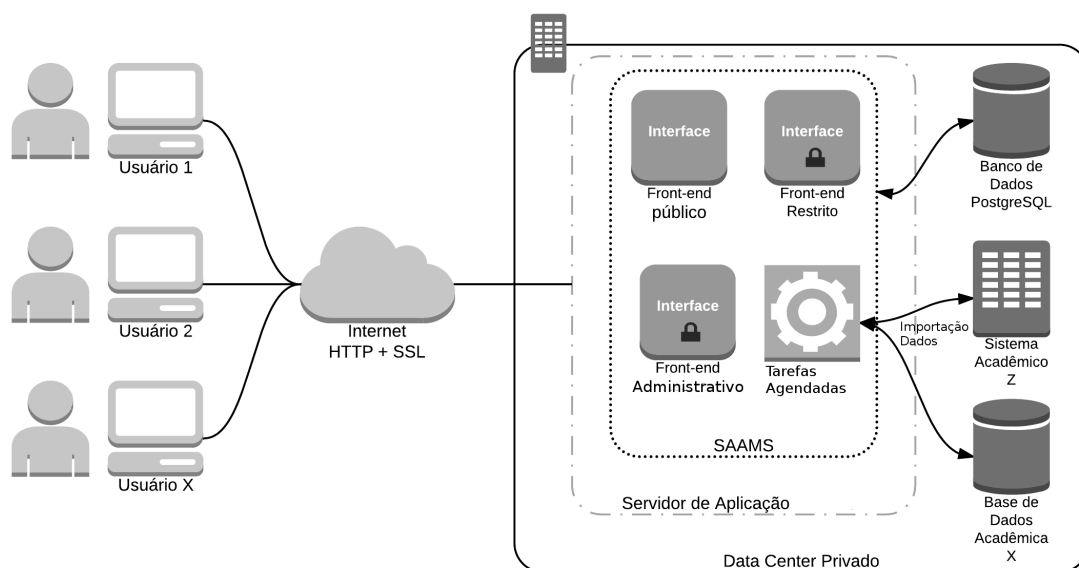


Figura 4. Arquitetura de Implantação do SAAMS

em PL/SQL, caso necessário durante o ciclo de vida do projeto.

Dentre as características providas pela plataforma Java EE¹⁰ e que influenciaram sua escolha, destaca-se a facilidade em utilizar o padrão de Injeção de Dependência (DI - *Dependency Injection*) apresentado por Silveira et al. [2012]. Esta abordagem transfere a classes fábrica a responsabilidade de localizar e inserir as referências de um objeto requerido onde o mesmo deve ser consumido, reduzindo o acoplamento entre as camadas que irão compor o sistema. Em Java EE, a Injeção de Dependência pode ser configurada através do uso de *Annotations* em determinadas classes, e a responsabilidade de implementação das classes fábrica é da plataforma e gerenciada automaticamente pelo ambiente de execução, reduzindo boa parte do esforço de codificação necessário para implementá-la.

A utilização de um Servidor de Aplicações Java EE [THE MIDDLEWARE COMPANY, 2003], simplifica a estrutura do projeto, uma vez que implementa e fornece ambiente de execução para o gerenciamento de objetos de negócio (Java Beans), a Injeção de Dependências, além de possibilitar o uso de *pools* de conexões (*JDBC Connection Pools*), cujo objetivo é melhora a performance na execução de consultas SQL [SHAIKH; PACHGHARE, 2017].

O sistema faz uso da biblioteca *UnboundID LDAP SDK*¹¹, para autenticação dos alunos no sistema junto a servidores LDAP. Como forma de garantir a confidencialidade, integridade e autenticidade dos dados entre clientes e servidor, é utilizado o protocolo de comunicação SSL¹² para o ambiente de implantação. Adicionalmente, a ferramenta *Jasper Reports*¹³, para a criação e manutenção de relatórios.

¹⁰Java EE - www.oracle.com/technetwork/java/javaee/overview

¹¹UnboundID LDAP SDK - <https://www.ldap.com/unboundid-ldap-sdk-for-java>

¹²Secure Socket Layer

¹³Jasper Reports - <http://community.jaspersoft.com/project/jasperreports-library>

5.3.2. Ambiente de Implantação

Como servidor de aplicações foi escolhido o Payara 5¹⁴, originado como um *fork* do *GlassFish*¹⁵, este último implementação de referência da Java EE desde 2005 e de grande sucesso, mas que foi recentemente descontinuado pela Oracle. O objetivo foi entregar ao projeto um ambiente que tenha suporte por um período mais longo.

5.3.3. Camada de Visão

Foi construída utilizando-se do *framework Java Server Faces* (JSF) com implementação principal através de *PrimeFaces*¹⁶; garantindo principalmente a obtenção de um nível mais abstrato de desenvolvimento na aplicação através do uso de componentes, ocultando detalhes e o fluxo do protocolo HTTP [COUTO; FOSCHINI, 2016].

5.3.4. Camada de Negócios

Para a camada de negócios, foi adotado o padrão da plataforma Java EE com a utilização de objetos de negócio implementados com *Enterprise Java Beans* (EJBs), e beans *Contexts and Dependency Injection* (CDI), este último que também fornece o contexto de injeção de dependências mais utilizado no sistema.

5.3.5. Camada de Modelo

De forma a facilitar a implementação da camada de modelo, utiliza-se de bibliotecas de mapeamento objeto-relacional *Java Persistence API* (JPA). Apesar da JPA disponibilizar um gerenciador de entidades, optou-se por manter uma camada intermediária no modelo, os *Data Access Objects* (DAOs), facilitando a manutenção e permitindo que o sistema lide com diversas fontes de dados de forma transparente, seja buscando dados em bases relacionais ou em interfaces de comunicação como *webservices*.

5.3.6. Boas práticas de projeto

Todo o código do projeto é mantido e gerenciado através de um Sistema de Controle de Versões Git¹⁷, que além de manter o controle de versões e modificações do código, possibilita o resgate de versões estáveis e ramificações no projeto [MASON, 2006].

Para simplificar o gerenciamento de dependências (bibliotecas), e automação da criação de pacotes de publicação (*build*), foi utilizado o Apache Maven¹⁸.

5.4. Demais resultados obtidos

As fases de elicitação de requisitos, de identificação das inúmeras regras de negócio, e de modelagem do banco de dados, evidenciaram que o sistema necessitaria de uma estrutura que ultrapassasse o trivial, o que em grande parte influenciou na escolha das tecnologias a serem empregadas, mas que também demonstrou a necessidade de implementar uma subestrutura interna própria que desse suporte às implementações posteriores. Os trabalhos que seguiram trataram principalmente de prover essa infraestrutura do sistema. A seguir são listados alguns dos requisitos implementados:

¹⁴Payara App Server - <https://www.payara.fish>

¹⁵GlassFish App Server - <http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html>

¹⁶PrimeFaces - <https://www.primefaces.org>

¹⁷GIT, <https://git-scm.com>

¹⁸Apache Maven - <https://maven.apache.org>

- Criação dos ambientes das áreas pública, restrita e administrativa;
- Implementação de autenticação através de LDAP para alunos, proporcionando a configuração de parâmetros através de arquivos de *properties* (visando flexibilidade), bem como a autenticação para Administradores e Equipe da CAAd através de tabelas internas do sistema.
- Criação de gerenciador de usuários, grupos e sua associação;
- Criação de mecanismo para persistência de *logs* de processamento do sistema, a ser utilizado principalmente para conferência de rotinas de importação de dados, ou de ações que necessitem de auditoria posterior, como a concessão e revogação de benefícios, modificação na classificação de candidatos, etc.;
- Criação da estrutura para consulta e gerencia de parâmetros de sistema, de forma a configurar variáveis para modificar o comportamento do sistema em situações diversas;
- Classes controladoras de cadastros, com o intuito de reduzir o tempo no processo de criação das telas de cadastramento do sistema e minimizar a replicação de códigos de tratamento. Implementadas com base em *generics* do Java e programação reflexiva. Incluem-se nestas classes, tratamentos que possibilitam automatizar cadastros de registros de tabelas que possuam relacionamento 1-N ou 1-1. Para sua realização foi utilizado o padrão *factory method*, de forma a permitir flexibilidade nas classes que estendem tais controladoras;
- Criação de componentes de interface dinâmicos na camada de visão, como listas e formulários, por motivos análogo ao item anterior;
- *Converters* e *View Services*, implementações relativas ao *framework* JSF, baseados em implementação com Java *generics*;
- Em algumas situações no sistema, registros são reservados e não podem ter os valores atualizados ou excluídos, o que é necessário por exemplo, em tabelas que permitem configurar estados para *workflows*, ou que sofrem importação de dados de sistemas externos mas que ainda devem suportar cadastro e alteração de novos registros. Foi efetuada a implementação de tratamentos para esses registros, utilizando Java *annotations* e programação reflexiva. Utilizou-se *annotations* também para tratamentos como ordenamento padrão em consultas, de forma a minimizar a geração de código *boiler plate*;
- Criado um conjunto de interfaces e implementações genéricas, de forma a permitir a criação de módulos de comunicação com sistemas externos utilizando o padrão estrutural *adapter*. A abordagem permite que os módulos sejam desenvolvidos sem dependência da estrutura interna do programa, bastando que adequem-se à interface provida, instalados no ambiente de execução (*classpath*) através de um pacote de classes (Java ARchive), e finalmente parametrizados no SAAMS para que possam ser ativados e executados;
- Implementado o controlador de comunicação para obtenção de dados de sistemas externos, fazendo uso das interfaces acima mencionadas, de forma a adquirir, validar e tornar os dados coletados homogêneos dentro dos padrões esperados pelo sistema;
- Classes para tratamento e geração de relatórios a serem utilizados no sistema;
- Classes utilitárias diversas.

Em relação aos demais requisitos levantados para o sistema já implementados, pode-se listar:

- Interface de consulta pública a editais e anexos;
- Realização do requisito funcional relativo a manutenção de editais e seus anexos;
- Atualização de dados pessoais por alunos, bem como de seus dados bancários;
- Manutenção do cadastro do aluno, de suas informações bancárias pela equipe, aprovação de modificações pendentes em seu cadastro por solicitação do aluno;
- Implementação dos demais cadastros necessários do sistema, como bancos, cursos, modalidades de curso, nível de curso, modalidade de ingresso, forma de ingresso, turno, períodos letivos, tipos de membros de grupo familiar, tipos de documentos;
- Manutenção de notas de histórico pedagógico do aluno;
- Criação do cadastro de atendimentos e manutenção dos seus desdobramentos;
- Criação de marcadores ou *labels* para atendimentos, de forma que seja possível extrair informações destes, sem expor as informações sensíveis do atendimento, criando uma forma de obter meios para qualificar e quantificar os mesmos;
- Realização do requisito funcional referente a importação de dados de sistemas acadêmicos do Sistema Acadêmico SCA. No entanto, a importação dos eventos de monitoramento – frequência e aproveitamento ainda está em desenvolvimento;

Cabe salientar que durante o processo de desenvolvimento foi instalado no campus para utilização o sistema BAE já mencionado. O mesmo implementa somente questionários para caracterização socioeconômica. Através da sua utilização, a equipe da CAAd vem gerando uma série de sugestões que serão apreciadas para melhoria da modelagem e construção dos casos de uso relativos a essa funcionalidade do SAAMS.

6. Considerações finais

Alguns aspectos são importantes em relação à usabilidade e devem ser implementados antes da liberação da primeira versão completa. São eles: a utilização de carregamento sob demanda nas listagens ou *lazy loading*; o aprimoramento das interfaces de cadastros com relação pai-filho; afinação ou (*tunning*) do servidor de aplicação para suportar SSL em todas as requisições, visto que em testes realizados as chamadas em Ajax realizadas pelo Primefaces acabou por tornar partes do sistema lento; a utilização de métodos nativos de autorização do Java EE – *Java Security* para implementar a segmentação das funções de cada usuário no sistema; a criação de melhores filtros nas listagens, possivelmente utilizando-se de Java *annotations* e *composite components* da JSF, uma vez que a utilização dos filtros no cabeçalho das tabelas limita o número de campos de filtro que podem ser utilizados.

Através de reuniões com a equipe realizadas durante o desenvolvimento do sistema, foram validadas todas as características desenvolvidas, principalmente de acompanhamento acadêmico e importação de dados do sistema SCA, de mais fácil compreensão por usuários finais. Evidenciou-se também que as funcionalidades desenvolvidas, bem como as vindouras, apresentam potencial substancial de melhora em suas condições de trabalho e atendimento, permitindo-lhes melhor organização, controle e acesso a informações.

Referências

- AMBLER, S. *Modelagem Ágil: Práticas eficazes para a programação eXtrema e o processo unificado*. [S.l.]: Bookman, 2004.
- BRASIL. Decreto nº 7234 – Dispõe sobre o Programa Nacional de Assistência Estudantil - PNAES. *Diário Oficial da União*, Brasília, 2010. URL: <http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2010/Decreto/D7234.htm>. Acessado em 01.03.2018.
- COUTO, R. S.; FOSCHINI, I. J. Análise comparativa entre dois frameworks mvc para a plataforma java ee: Jsf e vraport. *Revista TIS*, v. 4, n. 3, 2016.
- GAMMA, E. et al. *Padrões de Projetos: Soluções Reutilizáveis*. [S.l.]: Bookman, 2006.
- IFRS. Política de Assistência Estudantil do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Resolução CONSUP 086/2013. Bento Gonçalves, 2013. URL: <<https://ifrs.edu.br/wp-content/uploads/2017/07/ANEXO.pdf>>. Acessado em 01.03.2018.
- IFRS. Apresentação do Núcleo de Acompanhamento Acadêmico IFRS-POA. Porto Alegre, 2015. URL: <<http://www2.poa.ifrs.edu.br/institucional/setores/nucleo-de-acompanhamento-academico-naac>>. Acessado em 01.03.2018.
- IFRS. Instrução Normativa PROEN nº 005, de 05 de novembro de 2015. Bento Gonçalves, 2015. URL: <https://ww1.ifrs.edu.br/site/midias/arquivos/20151010143015338in_005_de_05_de_novembro_de_2015.pdf>. Acessado em 01.03.2018.
- IFRS. Página da Coordenadoria de Acompanhamento Acadêmico IFRS-POA. Porto Alegre, 2017. URL: <http://www.poa.ifrs.edu.br/index.php?option=com_content&view=article&id=2672&Itemid=564>. Acessado em 01.03.2018.
- MASON, M. *Pragmatic Version Control: Using Subversion (The Pragmatic Starter Kit Series)*. [S.l.]: Pragmatic Bookshelf, 2006.
- PACHECO, E. M.; PEREIRA, L. A. C.; SOBRINHO, M. D. Institutos federais de educação, ciência e tecnologia: limites e possibilidades. *Linhas Críticas, Revista da Faculdade de Educação da UnB, Brasília - DF*, 2010. URL: <<http://periodicos.unb.br/index.php/linhascriticas/article/viewFile/1429/1065>>. Acessado em 01.03.2018.
- PRESSMAN, R. *Engenharia de Software*. [S.l.]: McGraw Hill Brasil, 2009.
- SHAIKH, S. S.; PACHGHARE, V. K. A comparative study of database connection pooling strategy. 2017.
- SILVEIRA, P. et al. *Introdução à Arquitetura e Engenharia de Software - Uma visão sobre a plataforma Java*. Rio de Janeiro: Elsevier, 2012.
- TAUFICK, A. L. de O. L. Análise da política de assistência estudantil dos institutos federais de educação, ciência e tecnologia. *Revista Brasileira de Política e Administração da Educação, Universidade Federal de Goiás, Goiânia - GO*, 2014. URL: <<http://seer.ufrgs.br/index.php/rbpae/article/view/50020/31328>>. Acessado em 01.03.2018.
- THE MIDDLEWARE COMPANY. J2EE Best Practices. The Middleware Company, 2003. URL: <http://media.techtarget.com/tss/static/articles/content/TMCBestPractices/J2EE_BP.pdf>. Acessado em 01.03.2018.

Anexo A - Comparação de características presentes nos *softwares* identificados

	SIA	SUAP	AssisEXT	BAE	SIGAA	SAAMS
Versão	3.0	-	1.0	-	3.1.4	2.0
Data aproximada do <i>release</i>	2013	2013	2013	2017	2018	2019
Assistência Estudantil						
Publicação de editais de auxílios e anexos	Não	Não	Não	Não	Não	Sim
Inscrição para os editais de auxílios	Não	Sim	Sim	Sim	Sim	Sim
Questionários de caracterização socioeconômica	Não	Sim	Sim	Sim	Sim	Sim
Interface para construção dos questionários	Não	Sim	Não	Não	Sim	Sim
Classificação dos inscritos em grupos de vulnerabilidade	Não	Por auxílio	Por auxílio	Não	Por auxílio	Sim, com opção de segmentar por auxílio
Controle de documentos comprobatórios para concessão e manutenção dos auxílios	Não	Não	Não	Não	Documentos pré-definidos, sem parametrização	Sim
Recebimento em formato digital de documentos comprobatórios	Não	Não	Não	Não	Não	Sim
Emite avisos ou lembretes aos alunos/equipe sobre manutenção de vínculo, expiração de prazo de documentação, datas vincendas.	Não	Não	Não	Não	-	Sim
Acompanhamento Acadêmico						
Cadastro de atendimentos	Não	Sim	Não	Não	Não	Sim
Agendamentos de atendimentos	Não	Não	Não	Não	Não	Sim
Histórico pedagógico do aluno	Sim	Não	Não	Não	Sim	Sim
Monitoramento de alunos em vulnerabilidade	Não	Não	Não	Não	Não	Sim, através de alerta e lembrete
Monitoramento de eventos de frequência, aproveitamento e vínculo	Não	Não	Não	Não	Sim, integrado	Sim, emitindo alertas
Relatórios de evasão, aproveitamento e variáveis relativas ao perfil do aluno	Sim	Não	Não	Não	Sim	Sim
<i>Dashboards</i> e gráficos para monitoramento	Não	Não	Não	Não	Não	Sim
Aspectos estruturais						
Funcionamento independente de Sistema de Gestão Acadêmica	Integrado	Sim	Sim	Sim	Integrado	Sim
Comunicação para consolidação de dados de outros sistemas	Não	Não	Não	Não	Não	Sim
Orientado especificamente para Assistência e Acompanhamento Acadêmico	Não	Não	Sim	Sim	Não	Sim

Anexo B - Requisitos de software identificados a serem implementados pelo SAAMS

Área	Requisito
Assistência Estudantil	1. Consultar publicamente editais e anexos
	2. Prover meio para aluno atualizar dados de cadastro e dados para pagamento
	3. Prover preenchimento de cadastro de caracterização socioeconômica para alunos, informando grupo familiar e caracterização destes.
	4. Possibilitar cadastro de aluno a um edital de auxílios
	5. Emitir lista de documentos comprobatórios a serem entregues pelo aluno candidato
	6. Permitir ao aluno consultar o andamento da inscrição no edital
	7. Permitir ao aluno consultar os lançamentos de pagamentos e emitir recibos
	8. Permitir à equipe cadastrar editais e anexos, publicando-os
	9. Permitir à equipe construir e manter questionários de caracterização socioeconômica
	10. Permitir à equipe revisar as inscrições de alunos a um determinado edital
	11. Permitir à equipe revisar os questionários respondidos, corrigindo as informações se necessário
	12. Permitir à equipe revisar a documentação apresentada por um candidato em um edital
	13. Permitir à equipe obter classificação de candidatos em um edital de acordo com pontuação de caracterização socioeconômica
	14. Permitir à equipe manter configuração de auxílios e grupos de classificação
	15. Permitir à equipe conceder ou revogar benefício a candidato classificado
	16. Permitir à equipe emitir relatórios diversos referentes a editais: classificação, concessão de auxílios, dados bancários de alunos, ordem de pagamento, etc.
	17. Emitir à equipe e ao aluno, aviso sobre documentos comprobatórios vencidos para manutenção de auxílios
Acompanhamento Acadêmico	18. Permitir à equipe cadastrar atendimentos, bem como seus desdobramentos
	19. Permitir à equipe realizar o agendamento de atendimentos
	20. Permitir à equipe manter registros históricos pedagógicos sobre determinado aluno
	21. Prover à equipe meios de obter informações sobre os atendimentos de forma a gerar indicadores sobre os mesmos
	22. Prover à equipe o gerenciamento de uma lista de alunos em situação de vulnerabilidade para acompanhamento
	23. Emitir alertas para a equipe sobre frequência, aproveitamento, entre outros relacionados a alunos
	24. Gerenciar e armazenar documentos sensíveis relativos a atendimento de aluno: pareceres, atestados, etc.
	25. Emitir relatórios e gráficos de frequência, aproveitamento, evasão, formados, tendências, entre outras variáveis que possam ser apuradas relativas a acompanhamento acadêmico
Estrutural	26. Disponibilizar áreas segmentadas para público em geral, alunos e equipe
	27. Implementar método de autenticação de alunos utilizado internamente na instituição
	28. Permitir aos administradores gerenciar usuários e grupos
	29. Permitir aos administradores efetuarem a parametrização de aspectos do sistema
	30. Permitir aos administradores manter cadastros de informações associativas relativas a importação de dados de sistemas acadêmicos externos
	31. Manter registro de informações relativas ao funcionamento interno das rotinas do sistema
	32. Implementar estrutura que possibilite obtenção de dados de sistemas acadêmicos externos de forma plugável e independente
	33. Realizar a integração com o sistema acadêmico em uso no câmpus - SCA
	34. Implementar gatilhos de suspensão de auxílio em caso de suspensão de vínculo ou desligamento do aluno, bem como baixo aproveitamento acadêmico e frequência.

Anexo C - Diagrama lógico de tabelas do sistema

