

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

Curso de Graduação em Sistemas de Informação

Luiz Felipe Evaristo Barbosa

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA PROCESSO DE
SELEÇÃO DE ESTUDANTES DE MESTRADO EM EDUCAÇÃO DA UFVJM**

Diamantina

2018

Luiz Felipe Evaristo Barbosa

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA PROCESSO DE
SELEÇÃO DE ESTUDANTES DE MESTRADO EM EDUCAÇÃO DA UFVJM**

Trabalho de conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos exigidos para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Marcus Vinícius C. Guelpeli

Coorientadora: MSc. Jésyka M. Az. Gonçalves

Diamantina

2018

Luiz Felipe Evaristo Barbosa

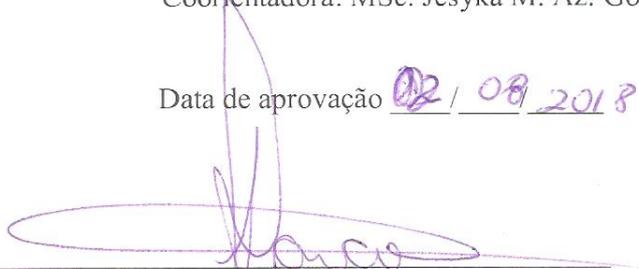
DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA PROCESSO DE SELEÇÃO DE ESTUDANTES DE MESTRADO EM EDUCAÇÃO DA UFVJM

Trabalho de conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação, como parte dos requisitos exigidos para obtenção do título de Bacharel em Sistemas de Informação.

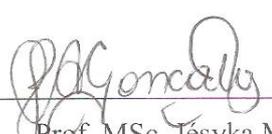
Orientador: Prof. Dr. Marcus Vinícius C. Guelpeli

Coorientadora: MSc. Jésyka M. Az. Gonçalves

Data de aprovação 02 / 08 / 2018



Prof. Dr. Marcus Vinícius Carvalho Guelpeli
Departamento de Computação – UFVJM



Prof. MSc. Jésyka Milleny Az. Gonçalves
PPGEd – UFVJM



Prof. MSc. Luís Henrique Silva Rodrigues.
Departamento de Computação – UFVJM

Diamantina

2018

“N3o deixe seus sonhos serem apenas sonhos”

Shia LaBeouf

AGRADECIMENTOS

Agradeço a Deus pelo dom da vida, por me conceder força de lutar a cada dia.

Agradeço aos meus pais, por motivar, ajudar e não me deixar desistir de alcançar meus objetivos. A minha namorada, agradeço pelos puxões de orelha e por me apoiar independente das minhas escolhas.

Agradeço aos meus amigos, colegas que fizeram os momentos dentro da faculdade mais felizes, e que me auxiliaram durante o desenvolvimento do presente trabalho e a todos que tive o prazer de conviver.

Agradeço ao meu orientador, Marcus Vinicius, por me conceder a oportunidade de fazer parte do projeto de pesquisa, que tem como resultado este trabalho e o sistema apresentado nele.

Agradeço a minha coorientadora, Jésyka, por me auxiliar durante a pesquisa e por todas as preciosas dicas.

Agradeço a todos os professores que tive na UFVJM por proporcionarem condições para o ensino e aprendizado de qualidade.

RESUMO

A internet vem se tornando uma ferramenta que torna o acesso a informação e a inscrição em processos seletivos cada vez mais fácil. O processo de seleção de alunos para o mestrado em educação, realizado anualmente, pelo Programa de Pós-graduação em Educação (PPGE) da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) foi realizado em 2016 de forma manual, catalogando e conferindo cerca de 300 inscrições. Com o intuito de aumentar o alcance e de simplificar o processo, este trabalho teve como objetivo o desenvolvimento de um sistema web para auxiliar na realização e na gestão de inscrições do processo seletivo, uma vez que o candidato necessita apenas de um computador com acesso à internet, este modelo de sistema torna a realização da inscrição mais simples. O sistema desenvolvido em equipe aplicou metodologia ágil, como *Kanban*, e *frameworks* amplamente utilizados no mercado, como *Laravel*, *Bootstrap* e *jQuery*. Como resultado, o sistema apresentado neste trabalho, foi utilizado no edital nº.29/2018, recebendo 382 inscrições e armazenando mais 4 mil documentos.

Palavras-chaves: PPGE; Laravel; Processo Seletivo; Kanban;.

ABSTRACT

The internet has become a tool that makes access to information and registration in selective processes easier and easier. The students' selection process for the master's degree in education, held annually by the Postgraduate Program in Education (PPGE) of UFVJM was conducted in 2016 by hand, cataloging and conferring about 300 entries. In order to increase the reach and to simplify the process, this work had as objective the development of a web system to assist in the accomplishment and the management of inscriptions of the selection process. Since the applicant only needs a computer with internet access, this system model makes the registration process simpler. The system developed in team applied agile methodology, such as Kanban, and frameworks widely used in the market, such as Laravel, Bootstrap and jQuery. As a result, the system presented in this work was used in bidding 29 of 2018, receiving 382 registrations and storing another 4 thousand documents.

Keywords: PPGE; Laravel; Selection Process; Kanban;

LISTA DE ILUSTRAÇÕES

Figura 1 - Requisição de uma página Web - Arquitetura Cliente Servidor.....	15
Figura 2 - Clico de Requisição MVC	17
Figura 3 - Quadro Kanban utilizado pela equipe.....	24
Figura 4 Sistema APPGIN: Home page do Administrador	27
Figura 5 - Sistema APPGIN: Homepage do perfil “Professor”	28
Figura 6 - Painel do perfil “Candidato”	29
Figura 7 - Página de informações de Edital.....	30
Figura 8 - Confirmação de Inscrição	31
Figura 9 - Formulário de inscrição, para cadastro de dados pessoais do candidato.	31
Figura 10 - Pagina com Dados Detalhados da inscrição	32
Figura 11 - Pagina com Dados Detalhados da Inscrição, continuação.....	33
Figura 12 Lista de inscrições	34

SUMÁRIO

1. INTRODUÇÃO	10
1.1. JUSTIFICATIVA	11
1.2. OBJETIVOS.....	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos.....	12
1.3. ORGANIZAÇÃO DO TEXTO	12
2. REVISÃO BIBLIOGRÁFICA	13
2.1. DESENVOLVIMENTO DE APLICAÇÕES WEB.....	13
2.2. FRAMEWORKS PARA DESENVOLVIMENTO WEB	13
2.3. ARQUITETURA CLIENTE-SERVIDOR.....	14
2.4. ARQUITETURA DE 3 CAMADAS	15
2.5. REQUISITOS DO SISTEMA	17
3. MATERIAIS E MÉTODOS.....	19
3.1. MATERIAIS	19
3.1.1 Organização da equipe.....	19
3.1.2 Ferramentas de desenvolvimento.....	20
3.2. MÉTODOS	21
3.2.1 Organização da equipe.....	21
3.2.2 Git.....	22
3.2.3 Laravel.....	23
4. RESULTADOS E DISCUSSÕES	24
4.1. RESULTADOS	24
4.1.1 Organização da equipe.....	24
4.1.2 Ferramentas de desenvolvimento.....	25
4.1.3 Apresentação do Sistema	26
5. CONSIDERAÇÕES FINAIS.....	35
REFERENCIAS	36

1. INTRODUÇÃO

Alguns programas como o Sistemas de Seleção Unificado (SISU), juntamente ao Exame Nacional do Ensino Médio (ENEM), possuem a finalidade de facilitar o ingresso ao ensino superior ao romper barreiras de acesso aos vestibulares. O SISU beneficia o candidato ao permitir que, qualquer inscrito no sistema, consiga concorrer a uma vaga em uma Instituição Federal de nível superior, pois a seleção é realizada pela internet.

O programa de Pós-Graduação em Educação (PPGE), da UFVJM conta com cerca de 40 alunos ingressados anualmente e cerca de 300 inscrições em cada processo seletivo, conforme publicações dos editais passados. Seguindo esta perspectiva, cada candidato a uma vaga de mestrado, para realização de sua inscrição, deve enviar cerca de oito documentos para cada vaga que deseja concorrer, o que gera um enorme volume de papeis e dados a serem processados pela comissão do programa.

Neste contexto, Bastos e Jaques (2010) afirmam que a internet vem ganhando espaço e é utilizada, para o entretenimento e para a educação, dentre outras finalidades. A *Web* tem se constituído uma mídia promissora, uma vez que oferece a oportunidade de interação, através de uma tecnologia naturalmente aberta e possibilita a colaboração coletiva de novas ideias e obras (LEVY, 1999).

Bastos e Jaques (2010) completam dizendo que a distância geográfica deixou de ser uma barreira para a humanidade, sendo que as principais, da atualidade, são a dificuldades econômicas e domínio de tecnologias. Contudo o número de usuários da *Web* continua aumentando, ampliando as possibilidades das aplicações para este fim.

Paula Filho (2013) define que aplicações *web* são “produtos de *softwares* ou sistemas de informática que utilizam parcialmente do protocolo *HTTP*”, protocolo de comunicação utilizado na internet, em consequência parte da interação com usuário se faz através de uma *web browser* e internet (FILHO, 2013).

Este trabalho objetiva aproveitar as vantagens de uso de um Sistema *web* para inscrições em processo seletivo e propor o desenvolvimento de uma aplicação *web* para o gerenciamento de inscrições de candidatos ao Programa de Pós-graduação em Educação (PPGE), da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM).

Algumas vantagens deste tipo de sistema, e também contribuições almejadas por este trabalho, podem ser citadas: facilidade de acesso, alta disponibilidade, transparência de informações, impacto na economia de papeis e por fim, agilidade no processo que era considerado lento principalmente por acontecer de forma manual.

A garantia da qualidade do sistema proposto será por meio da utilização de *frameworks* já consolidados no mercado e na comunidade *open-source*, fazendo o uso dos seguintes: *Laravel 5.4*, *Bootstrap 4*, *jQuery*. SGBD *MySQL*, servidor *Apache* e sistema *Linux*, além das metodologias aplicadas durante o desenvolvimento.

1.1. Justificativa

Segundo a secretaria do Programa de Pós-Graduação em Educação, da UFVJM, anualmente são realizadas cerca de 300 inscrições em cada processo seletivo, gerando uma quantidade enorme de papéis e informações a serem processadas.

O Plano de Logística Sustentável (PSL) publicado em 2013, mostra que a UFVJM está alinhada às metas globais da Organização das Nações Unidas (ONU) ao buscar melhor aproveitamento de recursos, sendo assim ele recomenda fortemente a economia e o reuso de papel. Sabe-se que a computação e a internet podem ser usadas como meio de solução na economia de papel e para facilitar o acesso do candidato ao Processo Seletivo e tornar mais fácil o gerenciamento das inscrições, por parte dos membros do PPGEd.

Neste contexto, um sistema *web* de gerenciamento de inscrições adequaria o processo seletivo mencionado às normas sugeridas pela PSL e facilitaria a rotina de processamento das informações dos candidatos inscritos.

Além disso, o mesmo favorece o acesso ao programa, uma vez que existem candidatos que deixam de concorrer por serem incapazes de entregar os documentos pessoalmente na repartição estipulada no edital de seleção do PPGEd, que normalmente é no Campus JK, da UFVJM, na cidade de Diamantina

Neste trabalho é levantado a hipótese se o sistema desenvolvido irá auxiliar o gerenciamento das inscrições pelo PPGEd, além de facilitar o acesso ao processo seletivo do programa PPGEd .

1.2. Objetivos

1.2.1 Objetivo Geral

Desenvolver um sistema *web* de gerenciamento de inscrições para os processos seletivos publicados pelo Programa de Pós-Graduação em Educação (PPGED) da UFVJM.

1.2.2 Objetivos Específicos

- Facilitar o acesso ao programa de Pós-Graduação em Educação;
- Organizar os projetos e documentos submetidos ao Comitê Avaliador;
- Estudar conceitos de Segurança e Desenvolvimento *Web* com PHP e *Frameworks*;
- Utilizar e implementar os conceitos de Integridade e Disponibilidade.

1.3. Organização do Texto

Capítulo 1 – Introdução

No primeiro capítulo deste trabalho faz-se uma apresentação da pesquisa, aduzindo seu contexto, motivações, finalidades e descrição da organização do texto aqui apresentado.

Capítulo 2 - Revisão Bibliográfica

No capítulo 2 será apresentada as principais teorias que fundamentam a execução deste trabalho. Neste serão detalhadas as ferramentas, linguagens e frameworks utilizados para a construção do sistema gerencial de inscrições.

Capítulo 3 – Materiais e Métodos

O capítulo 3 apresenta os materiais e métodos adotados neste trabalho. Serão abordados os instrumentos de coleta de dados, como o processo de Engenharia de Software que foi utilizado junto a implementação e contextualização do problema.

Capítulo 4 - Resultados e Discussões

No capítulo 4 serão apresentados o escopo do problema, junto a modelagem e requisitos levantados. Posteriormente são abordadas a explicação dos diagramas e apresentação das funcionalidades do sistema proposto.

Capítulo 5 - Considerações Finais

Por fim, neste último capítulo, serão abordadas as considerações do autor, os resultados alcançados e as limitações do sistema elaborado.

2. REVISÃO BIBLIOGRÁFICA

Este referencial teórico propõe os principais conceitos que fundamentam esta pesquisa, como: ferramentas de desenvolvimento web, metodologias para trabalho em equipe, arquiteturas de servidores, além de padrões de projetos utilizados no desenvolvimento *web*.

2.1. Desenvolvimento de Aplicações Web

Hirama (2012) ressalta que para iniciar o desenvolvimento de software, seja ele um tradicional, *mobile* ou *web*, é importante definir um canal de comunicação entre os membros da equipe, além de oferecer uma padronização que facilite a atividade de manutenção. Com estes dois pontos bem definidos, é possível que novos colaboradores sejam treinados para sua melhoria.

Para construir um simples site, ou um complexo sistema *Business-to-Business* (B2B), é necessária a utilização de ferramentas que possibilitam a consulta ao banco de dados, integração com sistemas externos e inúmeras outras funcionalidades (TEMPLE et al.,2004).

Segundo a revista eletrônica de tecnologia, Olhar Digital (2016), o número de *websites* vem crescendo a cada dia, onde, em 1994 existiam cerca de 3 mil sites e em 2014 extrapolou o 1 bilhão de páginas web. Perante este crescimento, o que atrai os leitores é a característica do site ter uso descomplicado. Este atributo foi definido por Nielsen e Hoa (2007) como usabilidade. Estes assemelham usabilidade com a rapidez que os usuários podem aprender a usar o *website* ou objeto.

Nielsen e Hoa (2007) afirmam que os usuários se tornam mais críticos com o passar do tempo, desta maneira, faz-se necessário o uso de boas técnicas de usabilidade. Devido ao alto número de *websites*, melhorias de usabilidade são úteis para adaptações de acordo com as necessidades dos leitores.

2.2. Frameworks para Desenvolvimento Web

Converse e Park (2003) definem o PHP como uma linguagem de *script* para *web* do lado do servidor, descrevendo como “*PHP: Hypertext Preprocessor*”, e podendo ser incorporado com HTML, código utilizado para construir pagina web, ou utilizar como binário. Assim, a linguagem se torna uma excelente ferramenta para construir páginas dinâmicas.se tornando uma

das mais utilizadas no mundo, com milhões de sites aplicando-a como linguagem principal, principalmente por possuir código aberto e ser gratuito (NIEDERAUER, 2011).

A criação de sistemas *web* foi favorecida pelo PHP, pois esta é uma linguagem de fácil aprendizado e aplicação. Entretanto, devido a simplicidade da linguagem, surgiram muitos programas pouco eficientes, com vulnerabilidade de ataques, além de fragilidade de código (MINETTO, 2007). Este ainda afirma que, código com difícil manutenção, misturando HTML, CSS, JavaScript e PHP, chamados de *spagetti code*, são comuns (MINETTO, 2007)

Os *Frameworks* são utilizados com o intuito de auxiliar no desenvolvimento de aplicações. Estes têm como principal objetivo agilizar o desenvolvimento e organizar a forma de implementação através do uso de padrões de projetos (Librelato e Belusso, 2014).

Padrões de Projeto e organização durante o desenvolvimento são características ressaltadas por Librelato e Belusso (2014). O autor continua chamando atenção para que aplicações comerciais usem *frameworks* já consolidados no mercado.

2.3. Arquitetura Cliente-Servidor

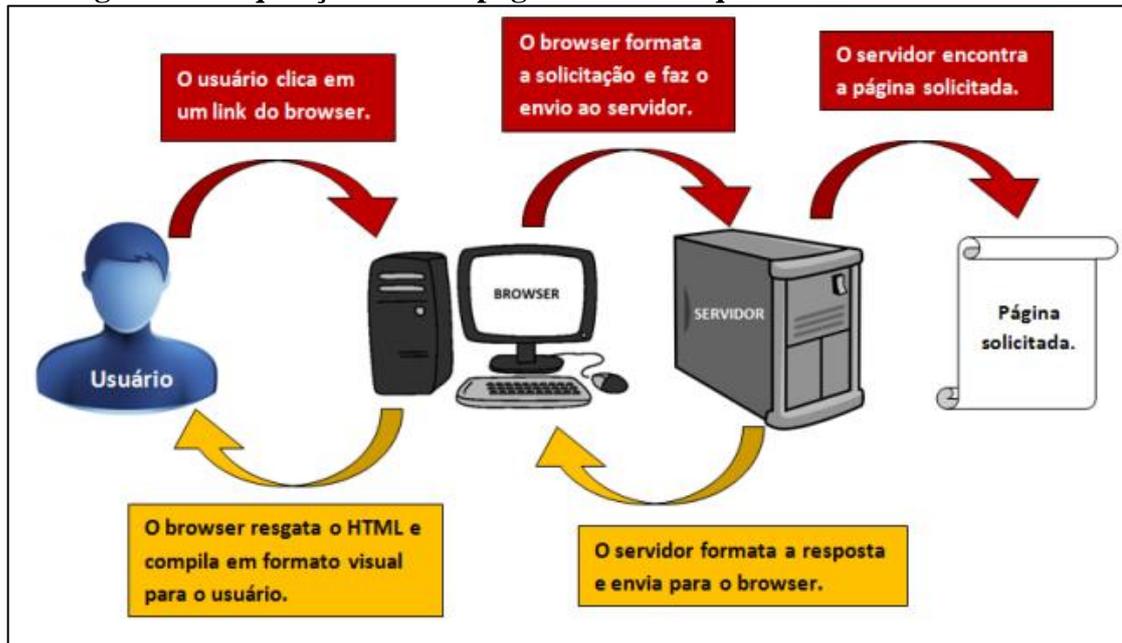
Uma aplicação *web* é caracterizada por ser acessada por um navegador ou browser *web*, onde as ações são enviadas por um cliente, *web browser*, a um servidor, que processa as informações e normalmente retorna uma página HTML (LIBRELATO E BELUSSO, 2014).

Palmeira (2015) define que a função do cliente *web* é permitir fazer solicitações ao servidor, exibindo o resultado do pedido. O autor continua definindo, que o navegador (*web browser*) é o software que se comunica com o servidor fazendo a comunicação entre o usuário e o servidor, neste caso ele é o cliente *web* do sistema (PALMEIRAS, 2015).

Segundo Rocha *et al* (2018), o usuário faz uso do *web browser*, que formata sua solicitação e envia ao servidor. Responsável por receber a solicitação e dados que o usuário enviou, o servidor é quem responde por tratar as regras de negócios, comunicar com banco de dados, comunicar com outros aplicativos, entre outras funcionalidades. Com a resposta da solicitação do usuário, o servidor envia uma página\resposta ao *web-browser* que fica responsável por mostrar ao usuário final. “Servidores disponíveis podem aceitar tais requisições, processá-las e retornar o resultado para o cliente. Apesar do conceito ser aplicado em diversos usos e aplicações, a arquitetura é praticamente a mesma” (MACEDO, 2018)

A Figura 1, a seguir, descreve o ciclo de requisição realizado entre o cliente e o servidor.

Figura 1 - Requisição de uma página Web - Arquitetura Cliente Servidor



Fonte: Palmeira, 2015, Adaptado.

Sistemas modernos consideram a arquitetura 2 como sendo a arquitetura de comunicação. Para organização do código e do servidor ele é subdividido em três camadas chamadas de MVC (*Model-View-Controller*). O MVC separa as responsabilidades de comunicação e tratamento de informações com o banco de dados, o tratamento de regras de negócio e a comunicação com o cliente (GAMMA *et al*, 2014).

A arquitetura *Model-View-Controller* delega ao *Model* a comunicação e tratamento de dados com o Sistema de Gerenciamento de Banco de Dados (SGBD), enquanto o *View* fica responsável pela renderização e tratamento de HTML\CSS\JS e o *Controller* é a entidade que aplica as regras de negócio providas do banco de dados e delega a *view* para disponibilizar ao usuário. Os conceitos de MVC apresentados são definidos por (CAPARRO *et al* 2007; LIBRELATO e BELUSO 2014; GAMMA *et al* 2014).

2.4. Arquitetura de 3 Camadas

Aplicações *Web* são geralmente uma conexão do tipo *HTTP* (FILHO, 2013). Onde certamente é feito uma requisição de recursos, como arquivos XML, fotos, páginas HTML dentre outros. Chaparro *et al.* (2007) completa que tradicionalmente estas interações na *Web* são divididas em 3 (três) camadas, onde um *model* armazena as informações e o usuário efetua chamada de funções de baixo nível para o navegador *renderizar* a resposta.

Librelato e Beluso (2014) definem a separação das 3 (três) camadas *Web* como: a camada de apresentação, que é o navegador, onde contém interface de interação com a aplicação; camada de persistência sendo o banco de dados e o SGDB, armazenando todas as informações da aplicação; e pôr fim a camada de lógica que executará a aplicação.

As camadas definidas por Librelato e Belusso (2014) são amplamente utilizadas na construção de sistemas. As três camadas são chamadas de *MVC*, onde: ‘*M*’ é definido como *Model*, responsável pela persistência e busca no banco de dados; ‘*V*’ é a camada responsável pela comunicação com o usuário conhecida como *View* e, ‘*C*’ são os *Controllers*, que cuida das regras de negócio (LIBRELATO E BELUSSO, 2014).

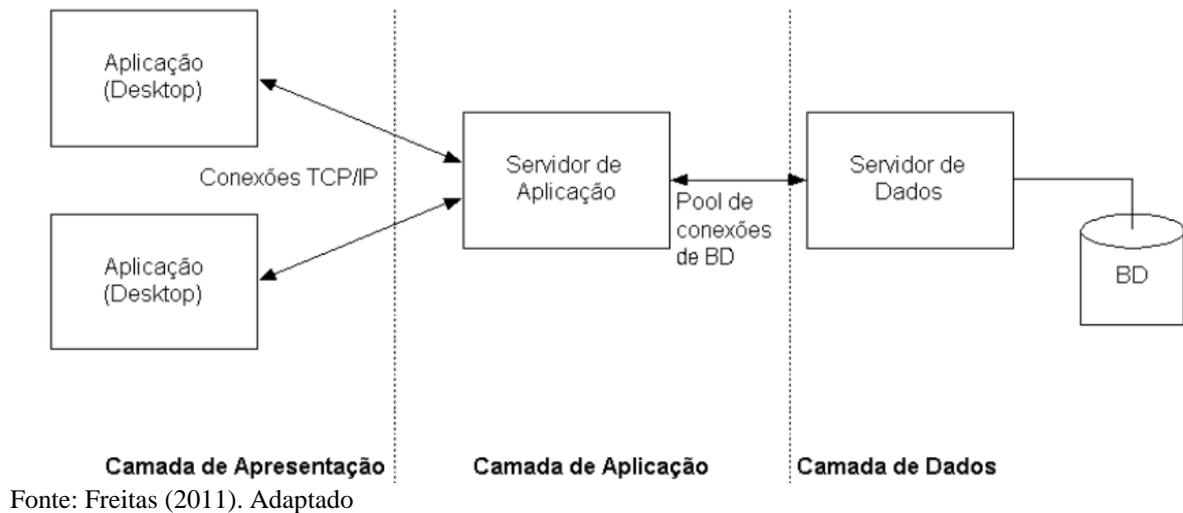
Gamma *et al.* (1994) apresenta estes conceitos dizendo que a arquitetura *MVC* foi introduzida como um padrão de projetos, em uma linguagem chamada *Smalltalk-80*, e foi amplamente utilizada em aplicações *web*.

Minetto (2007) explica que as requisições feitas pelo usuário ao sistema são enviadas ao *Controlle* que, manipula os dados usando o *Model* e invoca a *View* correta, de acordo com a ação executada ou com os resultados vindos do *Model*.

Guangchun e Hanhong (2003) completa ao afirmar que a camada *MVC* possui proporcional escalabilidade e manutenção, além de fácil personalização. Minetto (2007) afirma que o *MVC* separa a lógica e apresentação, sendo que favorece o trabalho em equipe. O trabalho em equipe é favorecido, seguindo a afirmação deste autor, que separa o *Design*, podendo trabalhar definindo os arquivos HTML, CSS e JS enquanto o DBA (*Data Base Administrator*) administra funções do banco de dados (MINETTO, 2007).

Freitas (2011) mostra como as interações ocorrem na Arquitetura *MVC* ao separar suas atribuições em um ciclo de vida, apresentado na Figura 2 a seguir.

Figura 2 - Ciclo de Requisição MVC



2.5. Requisitos do Sistema

Nardi e Falbo (2006) afirma que obter os requisitos corretos de um software é a parte mais importante e difícil durante o desenvolvimento da aplicação, e que a deficiência no tratamento de requisitos tem sido apontada como a principal causa de fracasso.

Martins e Daltrini (1999) apresentam dois grupos de problemas provindos de requisitos mal levantados: problemas acidentais e problemas essenciais. O primeiro é definido como os problemas oriundos de falta de controle, como pouco esforço para entender os requisitos, revisar e documentar; já no segundo, são os problemas que aparecem quando o usuário tem dificuldade em expressar o que ele deseja, ou quando existe uma má comunicação entre usuários e desenvolvedores.

“O caminho para o software de qualidade começa com a excelência nos requisitos.” (RIBEIRO e PIVOTO, 2002, p. 40). Os autores continuam afirmando que cada requisito deve descrever as necessidades com exatidão, não devem ser ambíguos e devem conter uma prioridade, assim o leitor (desenvolvedor) do requisito entende a necessidade e consegue fazer uma entrega com qualidade.

A capacidade de alterar o seu processo e sistema de informação que dão suporte, são um grande diferencial no mercado (JUNIOR E CAMPOS, 2008). O alinhamento entre os requisitos do software e as reais necessidades são as características que mantem a aplicação competitiva no mercado, além disso, os autores ainda afirmam que um sistema que está em um ambiente dinâmico deve aceitar mudanças para encontrar sucesso em sua execução (JUNIOR E CAMPOS, 2008).

A escolha da linguagem de programação utilizada para o desenvolvimento está muito relacionada aos requisitos, afirma BMED (2016). O autor ressalta que algumas perguntas como: “tipo de linguagem que a equipe domina?”; “Qual o tipo de sistema será desenvolvido?”; “Onde o sistema irá ser hospedado (desktop, web, mobile ...) ”, são questionamentos que devem ser feitos ao escolher uma linguagem (BMED, 2016).

3. MATERIAIS E MÉTODOS

Nesta sessão, inicialmente, será apresentado os materiais, ferramentas, *frameworks* e *plugins* utilizados para alcançar o objetivo principal desta pesquisa: o desenvolvimento do sistema, que foi realizado por uma equipe. Posteriormente serão apresentados os métodos de desenvolvimento e comunicação utilizados para melhorar a comunicação e divisão de tarefas dentro do projeto.

3.1. Materiais

Tendo em vista que o principal objetivo é o desenvolvimento de um sistema onde um candidato possa realizar o envio de documentos e dados pessoais para posterior avaliação de uma comissão, é fundamental que o sistema apresentado garanta as características de integridade e unanimidade.

Definidas pelo dicionário online Michaelis, unanimidade é a “conformidade de avaliações, julgamento”. Já integridade é a “qualidade de conduta irrepreensível, honesta, que entregue o seu estado inteiro” (MICHAELIS, 2018). Rocha, *et al* (2018) completa que para ter uma integridade de dados exige que o dado se mantenha por completo, sem sofrer alterações.

Seguindo esta perspectiva, a seguir serão descritas as ferramentas utilizadas e como a equipe utilizou e adaptou às suas necessidades.

3.1.1 Organização da equipe

3.1.1.1. Asana

O *Asana* é uma ferramenta que surgiu no mercado juntamente com outras metodologias para organização de equipes. Seguindo o site oficial, o principal objetivo do *Asana* é manter o *workflow* (fluxo de trabalho) organizado, oferecendo uma interface de simples uso e permitindo uma organização em forma de listas ou quadros. A ferramenta divide as equipes em *workspaces*, e cada *workspace* pode ser dividido em projetos, sendo o projeto um quadro ou lista de atividades a ser atribuída a um colaborador (ASANA, 2018).

Esta ferramenta é um grande atrativo para organizar os recursos do projeto, pois permite marcar uma data final, construir uma descrição, criar sub-taferas, anexar arquivos ou marcar

links externos. O *Asana* também pode ser utilizado como um quadro Kanban, tornando a organização da equipe de fácil acesso e fácil atualização.

3.1.2 Ferramentas de desenvolvimento

3.1.2.1. Git

Chacon e Straub (2014) introduziram o conceito de sistema de versionamento como um sistema de gravar mudanças em um arquivo, permitindo acessar as mudanças futuramente ou retroceder a um determinado ponto.

Os autores afirmam que muitas pessoas gravam diversos arquivos no disco, pois essa prática dificulta a identificação do arquivo atual, tendenciando a sobrescrita em um arquivo errado e conseqüentemente a perda de mudanças ou backups (CAHCON e STRAUB, 2014).

Git é um sistema criado por Linus Torvalds, para utilização no *kernel* do Linux (FARIAS, 2013). O autor destaca que este sistema possui características positivas sobre outros sistemas de desenvolvimento por ser não-linear, com desenvolvimento distribuído através de replicação de repositório.

A escolha dessa ferramenta foi feita porque, como mostram Franzolin e Custódio (2011), é amplamente utilizado em vários projetos, como Facebook, PostgreSQL e Netflix, além de possibilitar a implementação de backups e histórico de código com custos baixos.

3.1.2.2. Laravel

“Todas as linguagens de programação *server-side*, como o PHP, certamente não têm barreiras de entrada”, (BEAN, 2015, p. 2). Neste projeto é utilizado *frameworks* para a construção da aplicação, sendo que o principal utilizado é o *Laravel* em sua versão 5.4. *Laravel* utiliza o PHP como sua linguagem principal

A comunidade *Laravel*, assim como o próprio *framework*, tem as definições de estrutura e padrões de projetos, e as definições e boas práticas que foram seguidas pela equipe de desenvolvimento. A descrição dos padrões utilizados serão detalhados no decorrer do texto.

Laravel é um *framework* desenvolvido para agilizar a criação de aplicações *web*, afirmam Douglas e Marabesi (2017). Os autores chamam atenção que este *framework*

simplifica o uso e fornece funcionalidades para desenvolvedores ganharem produtividade, apesar de usar muitos componentes do *Symfony*, outro *framework* muito utilizado.

Bezerra e Schimiguel (2006) mostram que este é um dos *frameworks* mais utilizados do mundo hoje em dia, apontando como pontos positivos sua facilidade de aprendizado, com documentação clara, vastos recursos, comunidade ativa e prestativa, além da estabilidade do código.

3.1.2.3. Sentry

Para completar o framework, o projeto faz uso de um sistema externo para captura de erros: o *Sentry*. Esta é uma ferramenta *open-source* de rastreamento de erros, em sua documentação oficial é definido como um rastreador que ajuda a monitorar o desenvolvimento e reparar quebras no código em tempo real (SENTRY.IO, 2017). Construído com um design simples e de fácil aprendizagem, a ferramenta *Sentry* torna fácil a integração com várias linguagens e *frameworks*, incluindo o PHP e *Laravel*.

3.2. Métodos

3.2.1 Organização da equipe

O *Kanban* é uma estrutura popular utilizada no desenvolvimento ágil de software. Este é um subsistema Toyota de produção (STP) utilizado para controlar os estoques em processo, a produção e o suprimento de componentes, afirmam Junior e Filho (2008). A tradução literal de *Kanban* é “anotação visível”. Essa metodologia foi adotado pela equipe junto com o sistema do *Asana*.

Desenvolvido para atender a uma empresa específica, cada equipe adapta o chamado quadro *Kanban* da maneira mais pertinente e atraente para o melhor desempenho da equipe (SILVA *et al*, 2012). Em equipes físicas e de encontros diários, é afixado um quadro em uma área de fácil acesso, onde a equipe desenvolve/trabalha, com o objetivo de apresentar as atividades e seus *status*.

O “Quadro *Kanban*” é um quadro com título do projeto, data de início e equipe responsável descritos na parte superior. Este é dividido em três sessões na horizontal, cada sessão

tem o objetivo de informar a situação de uma atividade dentro do projeto. Para representar uma atividade é utilizado *post-it*, em cada *post-it* é descrito a atividade, junto com a prioridade do projeto ou requisito de funcionamento. Cada integrante do projeto, quando responsável, deve alterar o *post-it* quando houver alteração de status, mantendo toda equipe atualizada.

A divisão do quadro é um comum acordo entre a equipe, normalmente ele é dividido entre “A fazer”, “Fazendo”, “Feito”. Quando o responsável pela atividade inicia a construção da tarefa o *post-it* referente a esta atividade é movido entre as sessões. A divisão apresentada foi baseada nos argumentos de Silva *et al* (2012) que mesclou algumas metodologias do *Scrum* ao *Kanban*.

Como descrito, o *Kanban* atinge seu objetivo quando executado e atualizado por toda a equipe. O quadro em local de fácil acesso e visível a toda a equipe facilita a comunicação e trocas de informações.

3.2.2 Git

Para o desenvolvimento deste projeto, a equipe optou por utilizar a ferramenta *GIT* como um complemento à ferramenta *Asana*, auxiliando para manter o código do projeto atualizado e sincronizado. Foi acordado que os *commits*, envios de código ao servidor *GIT*, deveriam ser feitos seguindo as divisões aplicadas no *Asana*.

Uma característica que permite as equipes trabalharem separadamente é o fato de que o *GIT* é distribuído e descentralizado, sendo pontos positivos, pois permitem o desenvolvimento remoto, porém, quando é realizado a junção de códigos (*merge*) é preciso ter atenção pois pode acontecer de sobrepor código novo pelo antigo (CHACON E STRAUB,2014).

A metodologia escolhida para o versionamento de código foi assertiva, uma vez que houve poucos conflitos, e quando realizado os testes, não apareciam erros, pois os *commits* eram enviados apenas após a finalização da tarefa, evitando funções pela metade.

Além da escolha do método de envios de *commits*, foram criados 2 *Branchs* no git, sendo um a versão utilizada no servidor, onde os candidatos realizam as inscrições e o outro com o código em desenvolvimento. Com *branchs* distintos foi possível separar o sistema que estava em produção do sistema em desenvolvimento.

3.2.3 Laravel

Ferramenta de uso descomplicado, com vasta documentação. *Laravel* foi utilizado pela equipe como principal *framework* de desenvolvimento, ele conta com uma ferramenta que auxilia em tarefas rotineiras no desenvolvimento, como criação de classes e arquivos de configurações, o facilitador é chamado de *Artisan*.

Segundo a Documentação Oficial do *Laravel*, o *Artisan* é um CLI (*Command Line Interface*) com o objetivo de auxiliar o desenvolvedor a executar algumas rotinas durante o desenvolvimento do sistema (OTWELL, 2018). Uma CLI é executada através de comando em um terminal. Comando como `'make:controller'` efetua a criação de uma classe já configurada com função de *Controller* dentro do projeto.

No desenvolvimento do sistema apresentado por este trabalho foi definido alguns comandos junto aos parâmetros para criação dos *migrations*, *model*, *controllers* e *views*. A definição do comando junto aos parâmetros permitiu a equipe desenvolver o projeto com padrão definido, pois os comandos *Artisan* criam os arquivos nos diretórios definidos durante a configuração do projeto.

A biblioteca chamada de *Query Build* é uma peça importante dentro do *framework* do *Laravel*, pois permite a conexão com diversos bancos de dados, além de fazer uso do PDO para proteção de *SQL injection* (OTWELL, 2018). O *Laravel* abstrai o conceito de SQL, permitindo que o sistema se mantenha independente do Sistema de Gerenciamento de banco de Dados (SGBD). Durante o desenvolvimento foi utilizado as classes chamadas de *migrations*, que permite criar e alterar tabelas no banco através do PHP;

A comunidade que suporta o *Laravel*, assim como sua documentação oficial, indica que este faz forte uso da arquitetura MVC, além de utilizar muitos conceitos e padrões de projetos provindos da Orientação a Objetos. As classes, e funções desenvolvidas nos projetos são simples e fáceis de entender, pois herdam dos recursos oferecidos pelo *Laravel* e suas bibliotecas. As classes são minimalistas, com apenas poucas linhas de códigos, porem fazendo forte uso de Orientação a Objetos e seus padrões de projetos (OTWELL, 2018).

4. RESULTADOS E DISCUSSÕES

4.1. Resultados

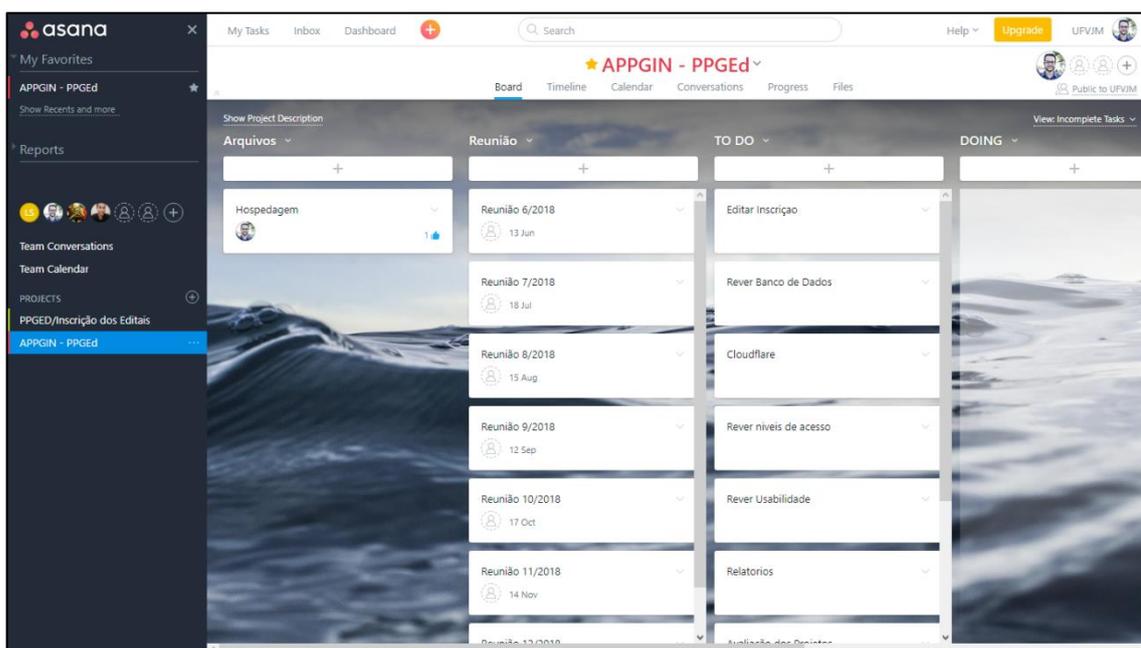
4.1.1 Organização da equipe

Em relação a organização da equipe, foi criada uma conta para acesso ao *Asana* por cada integrante da equipe de desenvolvimento. O uso do *Asana* foi feito baseado no *Kanban*, dividindo o projeto em listas de atividades. As listas foram: Arquivos gerais e Reuniões com *status* de “A fazer”, “Fazendo” e “Feito”.

Cada integrante ficou responsável por uma atividade, marcando data de entrega e movendo o cartão (atividade) de lista, conforme o desenvolvimento. Ao finalizar a tarefa o responsável pela conclusão ficou encarregado de fazer uma breve explicação da atividade ou anexar arquivos e link pertinentes para futuras consultas.

O quadro Kanban, apresentado na Figura 3, foi utilizado pela equipe e marcado com todas as reuniões pertinentes ao projeto, facilitando o agendamento e organização dos colaboradores. As tarefas, quando iniciadas, eram movidas para a lista “fazendo” e marcando o responsável. Arquivos, é a lista que armazena informações gerais, que não são tarefas, como dados relacionados aos e-mails, hospedagem web, e documentos produzidos.

Figura 3 - Quadro Kanban utilizado pela equipe



Fonte: Próprio Autor

A metodologia *Kanban* junto do *Asana* facilitou o trabalho em equipe, pois todos os colaboradores tinham acesso a todas as informações do projeto, além de manter a equipe sempre atualizada sobre o real *status* do projeto.

4.1.2 Ferramentas de desenvolvimento

4.1.2.1. GIT e *Laravel*

Como apresentado anteriormente, o sistema foi construído utilizando o *framework* em PHP - *Laravel* 5.4. A linguagem PHP é uma linguagem *server-side*, onde funciona apenas no servidor enviando respostas ao cliente. No *front-end* foi utilizado *frameworks* como *Bootstrap* CSS e *jQuery*. As ferramentas de *front-end* foram utilizadas no *template Gentelella*, sendo escolhido porque o mesmo defini um padrão de navegação, além de reunir boas práticas de UX/UI (usabilidade e experiência do usuário).

Para a junção de código e bom funcionamento da equipe, foi utilizado o GIT com a metodologia apresentada. A utilização do GIT foi assertiva, pois a equipe conseguiu se comunicar e obter versões atualizadas do sistema com exatidão. Sem a utilização de uma ferramenta com o objetivo do *GIT* não seria possível manter uma qualidade e produtividade com relação ao trabalho em equipe.

A utilização do *framework* PHP foi realizada de uma forma simples, como recomenda a documentação oficial. Junto ao *Artisan* que é o CLI (*Command Line Interface*) do *Laravel* que oferece uma série de comandos para aumentar a produtividade dentro do *framework*, como o comando *make* que oferece a criação de classes com funcionalidades básicas, como criação, busca, edição e deleção de dados providos do banco.

Iniciar a utilização de um *framework* é uma tarefa difícil, pois quando não é de conhecimento da equipe ele deve ser estudado e principalmente entender a arquitetura e funcionamento, para melhor aproveitar os recursos oferecidos. A curva de aprendizado é um grande problema, afirmam Douglas e Marabesi (2017, p 13), quanto a escolha de uma ferramenta desse tipo, o grande diferencial dos *frameworks* famosos é a comunidade que o suporta, continua os autores, oferecendo recursos e principalmente bons materiais para estudo e uso de tal ferramenta.

Os principais pontos de escolha do *Laravel* foram: Vasta documentação e suporte da comunidade, além de um grande histórico de bom desempenho, visto que ele é escolhido por

muitos sites como o *framework* mais utilizado em PHP, segundo pesquisa realizada pelo *SitePoint*, que é um website com livros e curso de desenvolvimento *web*.

4.1.3 Apresentação do Sistema

O sistema foi desenvolvido seguindo as boas práticas de programação recomendadas na documentação do *Laravel* e na documentação oficial do *Gentelella*. O sistema foi nomeado pela equipe de desenvolvimento por APPGIN, Aplicativo para Gerenciamento de Inscrições.

Para a construção do banco de dados, a documentação oficial do *Laravel* recomenda a utilização de uma ferramenta chamada *Migrations*, onde o banco é construído através da criação de uma classe em PHP, abstraindo as *queries* de construção SQL. Pelo fato do *Laravel* utilizar a biblioteca PDO, para conexão com banco de dados, este *framework* detém de compatibilidade com vários bancos de dados suportado por esta biblioteca, junto ao *Migration* a aplicação se torna independente do Banco de Dados.

Compatibilidade com diversos bancos é um ponto importante dentro do sistema, pois é como o usuário acessa os dados salvos no banco de dados. Utilizando a arquitetura *MVC*, o *Laravel* requisita ao SGBD os dados, que são processados pelo *Controller* e *renderizado* em HTML, para ser enviado ao *Web Browser*.

Para o bom funcionamento do sistema, foram criados três níveis de acesso, sendo eles: o “administrador”, que é o usuário que não tem restrição de acessos; o “professor” que é o usuário que tem acesso as inscrições e pode acessar os arquivos enviados ao sistema pelo candidato, e por fim o usuário “candidato” que tem acesso a seus dados pessoais e pode realizar inscrições em editais. Nos próximos sub tópicos serão apresentadas as visões de cada nível de usuário no sistema.

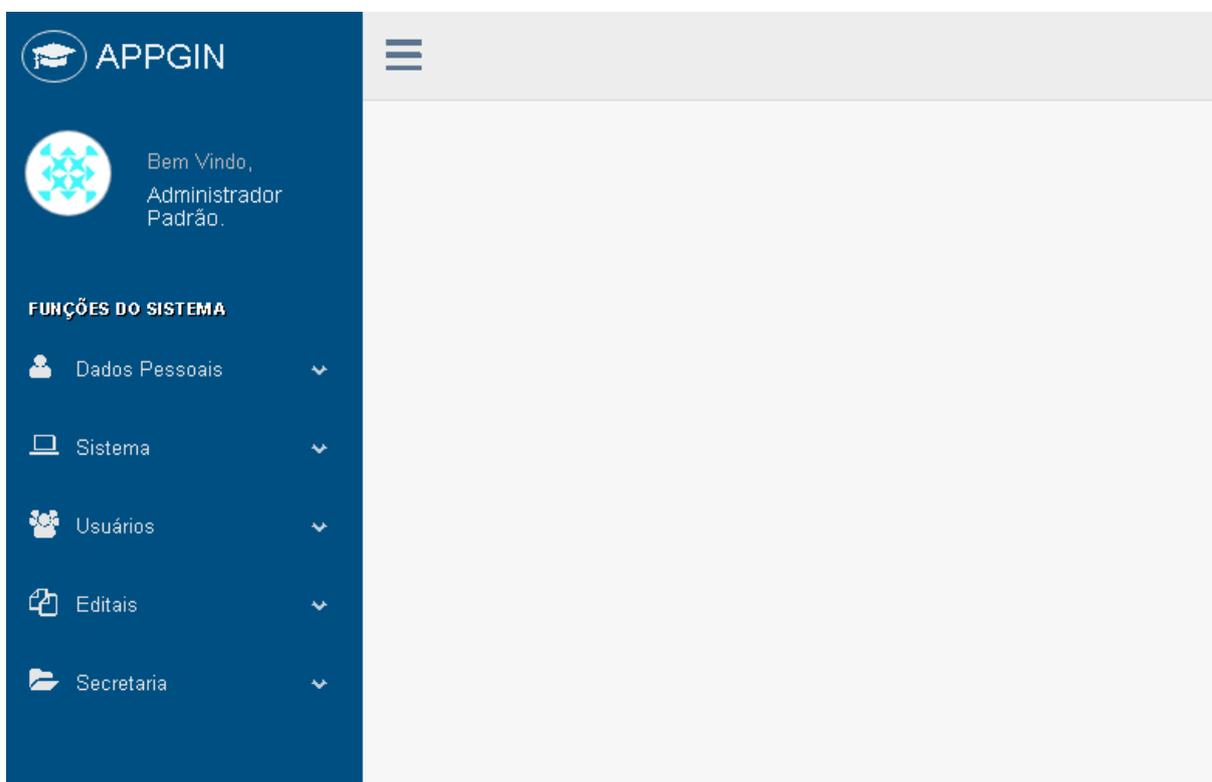
4.1.3.1. Nível de Acesso do Administrador

Para o processo seletivo acontecer ele depende de uma comissão responsável pela seleção. Os responsáveis por esta devem ter acesso total ao sistema para verificar, publicar e editar informações.

Como usuário administrador é possível cadastrar, editar e visualizar as etapas da seleção, disponibilizar arquivos úteis aos candidatos, acessar as inscrições de forma individual, cadastrar um novo professor e usuário do sistema, além de principalmente disponibilizar um edital para inscrição.

A Figura 4 mostra a *HomePage*, ou página inicial do sistema, quando acessado por um usuário do tipo Administrador.

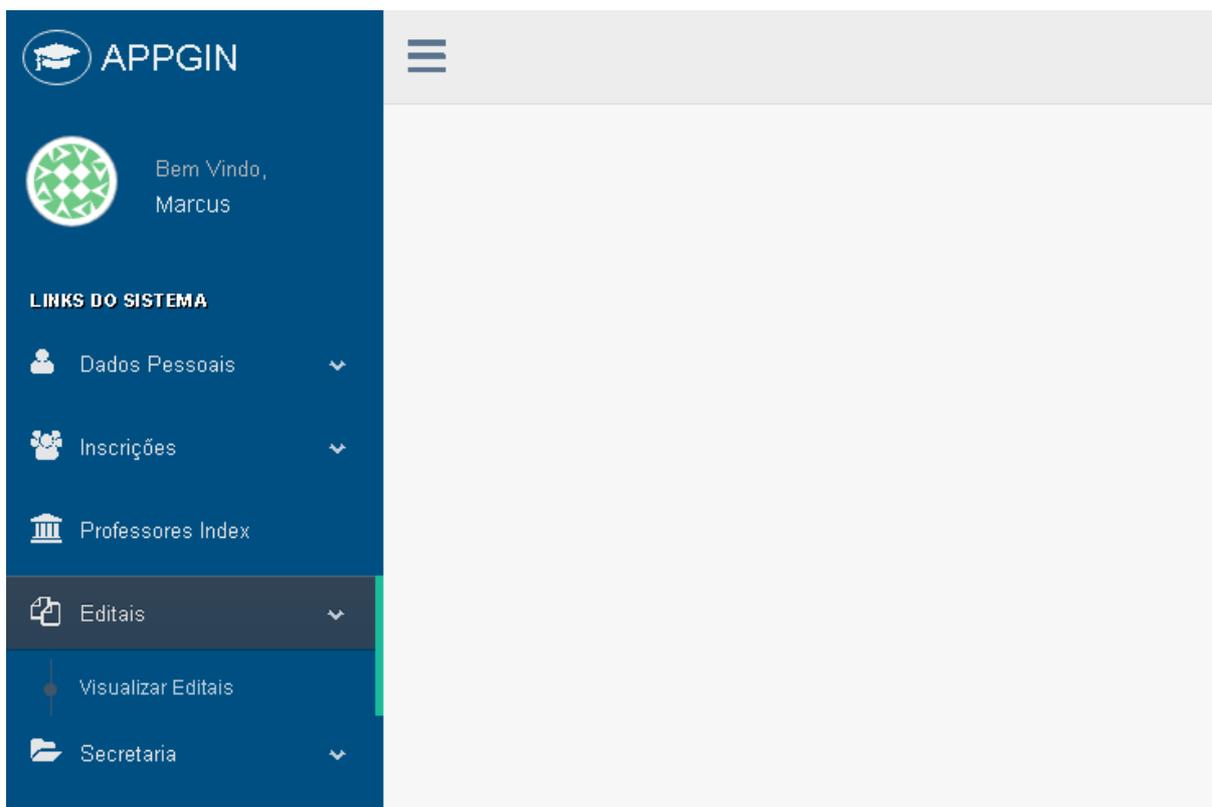
Figura 4 - Sistema APPGIN: Home page do Administrador



Fonte: Sistema APPGIN Próprio Autor

4.1.3.2. Nível de Acesso do Professor

Categorizado como o ator que revisa e avalia as informações enviadas pelos candidatos. O perfil chamado “Professor” é bloqueado a fazer inscrições em processos seletivos, e sua principal característica é acessar as inscrições de todos os candidatos. O acesso as inscrições são para fins de avaliação da inscrição e dados enviados. Com nível de acesso diferente do Administrador e candidato, a página inicial deste tipo de usuário é mostrada pela Figura 5.

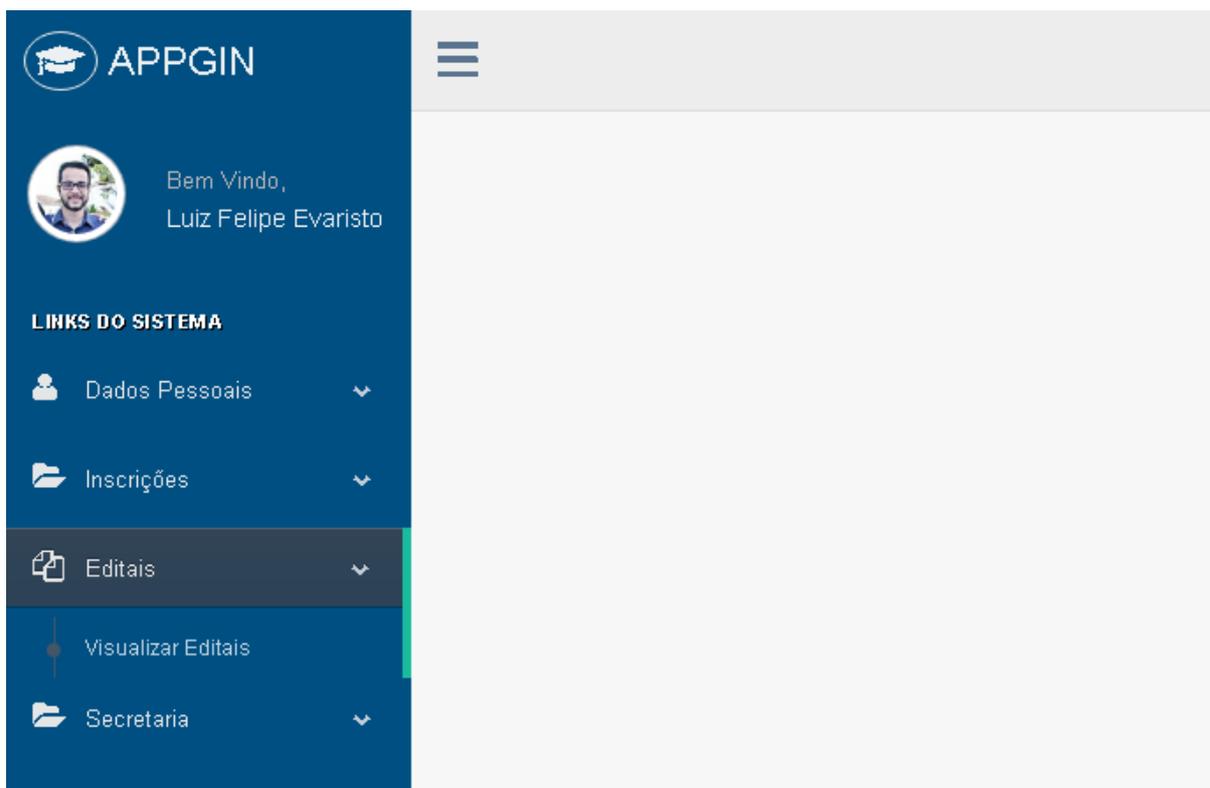
Figura 5 - Sistema APPGIN: Homepage do perfil “Professor”

Fonte: Sistema APPGIN, próprio autor.

4.1.3.3. Nível de Acesso do Candidato

O Candidato é a peça fundamental dentro do sistema, sendo o ator que realiza a inscrição em processos seletivos. O sistema deve garantir que o acesso do candidato *logado* não acesse inscrições de outros candidatos, ou realize cadastros de informações relevantes, como novo edital, dentro do sistema. Efetuar o bloqueio de outras áreas do sistema é fundamental para garantir a unanimidade e integridade dos dados enviados por cada candidato.

Ator com menor número de ações dentro do sistema, porém peça fundamental para que todo o processo de seleção e avaliação funcione corretamente. As principais ações dentro do sistema são a busca por editais e a visualização das inscrições, opções acessadas pelo painel lateral esquerdo como mostra a Figura 6.

Figura 6 - Painel do perfil “Candidato”

Fonte: Sistema APPGIN, próprio autor.

4.1.3.4. Processo de inscrição pelo candidato

Um dos pilares utilizados para a construção do sistema é a simplicidade e fácil utilização dos recursos oferecidos pelo sistema. Como afirmando anteriormente, a usabilidade é um fator muito importante, pois os usuários estão se tornando mais críticos e a cada dia requisitam aplicações que visam facilitar o cotidiano.

Para realizar uma inscrição, primeiro é necessário criar uma conta de usuário com e-mail que o usuário tem acesso, pois este e-mail pode ser utilizado para futuras comunicações. Após a criação de uma conta de usuário é possível acessar o sistema, e visualizar as informações pessoais, como documentos e inscrições passadas, além das pagina de acesso público.

A Figura 7 é um exemplo, com dados fictícios cadastrados pelo autor, para mostrar como o sistema funciona. Na parte esquerda da tela é mostrado um menu lateral, este menu se encontra presente em todo o sistema. O restante da tela são as informações apresentada pelo sistema, informações que foram cadastradas anteriormente pelos usuários administradores.

Figura 7 - Página de informações de Edital

Bem Vindo,
Usuário 1

LINKS DO SISTEMA

- Dados Pessoais
- Inscrições
- Editais
- Secretaria

Edital código, 1

Descrição

Convenio do Edital: **UFVJM**
 Ano/Semestre de egresso: **2018/2**
 Data de Início das Inscrições: **Apr 1, 2018**
 Data de Fim das Inscrições: **Jul 4, 2018**

[Download do Edital](#)

* É importante a leitura do Edital publicado para concientização dos requisitos de avaliação.
 ** Ao submeter a inscrição, você aceita os termos do presente edital e afirma que irá cumprir com os itens solicitados.

Documento requisitados em formato PDF

Documento	Descrição
CPF	Documento de CPF frente e verso
Identidade ou Passaporte	Documento de Identidade ou de Passaporte(caso de estrangeiro) Frete e verso
Projeto	Projeto sem identificação do usuário

[Realizar Inscrição no Processo Seletivo.](#)

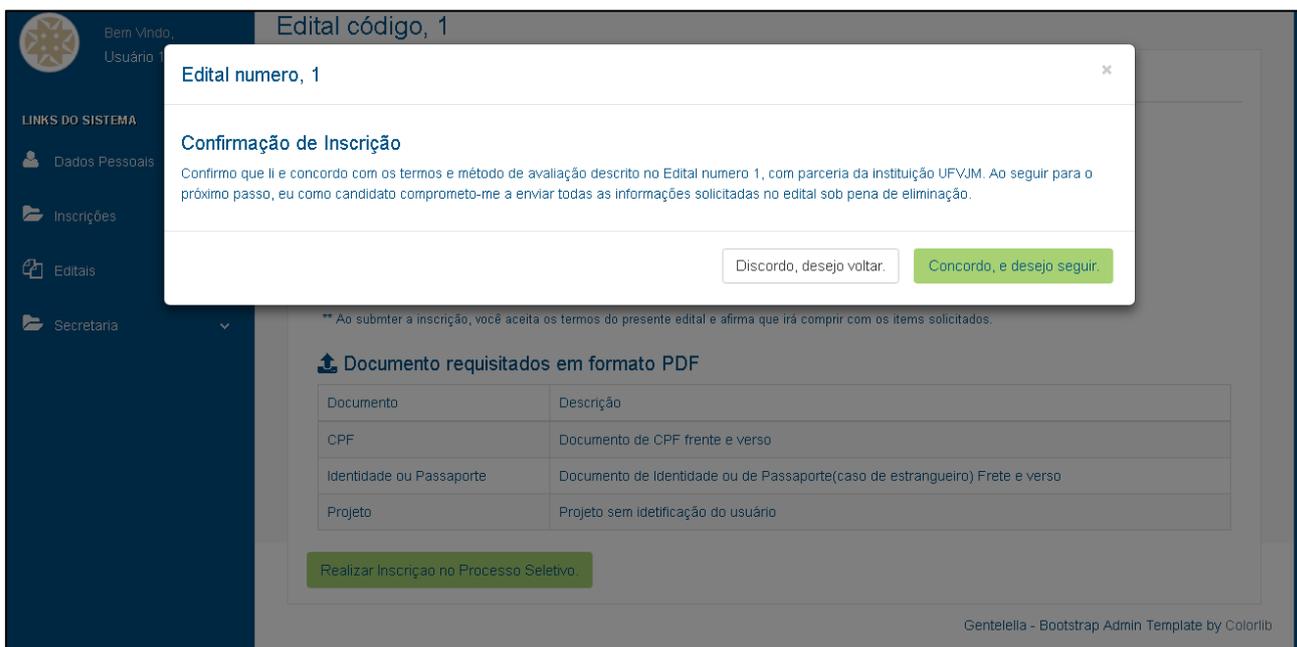
Gentelella - Bootstrap Admin Template by Colorlib

Fonte: Sistema APPGIN, próprio autor

Ao confirmar o interesse em se candidatar em um processo de seleção, amparado por um edital, presente no sistema, o sistema solicita uma confirmação, pois ele permite apenas 1 e-mail por edital. Mensagens de confirmação em pontos críticos do sistema são mostrados, como na Figura 8.

Com a confirmação do interesse em se candidatar o sistema solicita a verificação dos dados pessoais, como número de RG, CPF, nome da mãe, etc exemplificada pela Figura 9. A confirmação dos dados é importante para conferência dos documentos e no caso de aprovação futura, exportação de dados para o sistema de gestão da universidade.

Figura 8 - Confirmação de Inscrição



Fonte: Sistema APPGIN, próprio autor

Figura 9 - Formulário de inscrição, para cadastro de dados pessoais do candidato.

Fonte: Sistema APPGIN, próprio autor

Ao finalizar a verificação dos dados o candidato é encaminhado ao painel da inscrição, onde é mostrado todos os dados referente a inscrição, incluindo número de inscrição e os dados pessoais. Logo abaixo é disponibilizado a área para anexar os arquivos, como mostrado na

Figura 10 e 11. Os arquivos solicitados são cadastrados pelos Administradores do sistema ao cadastrar o edital.

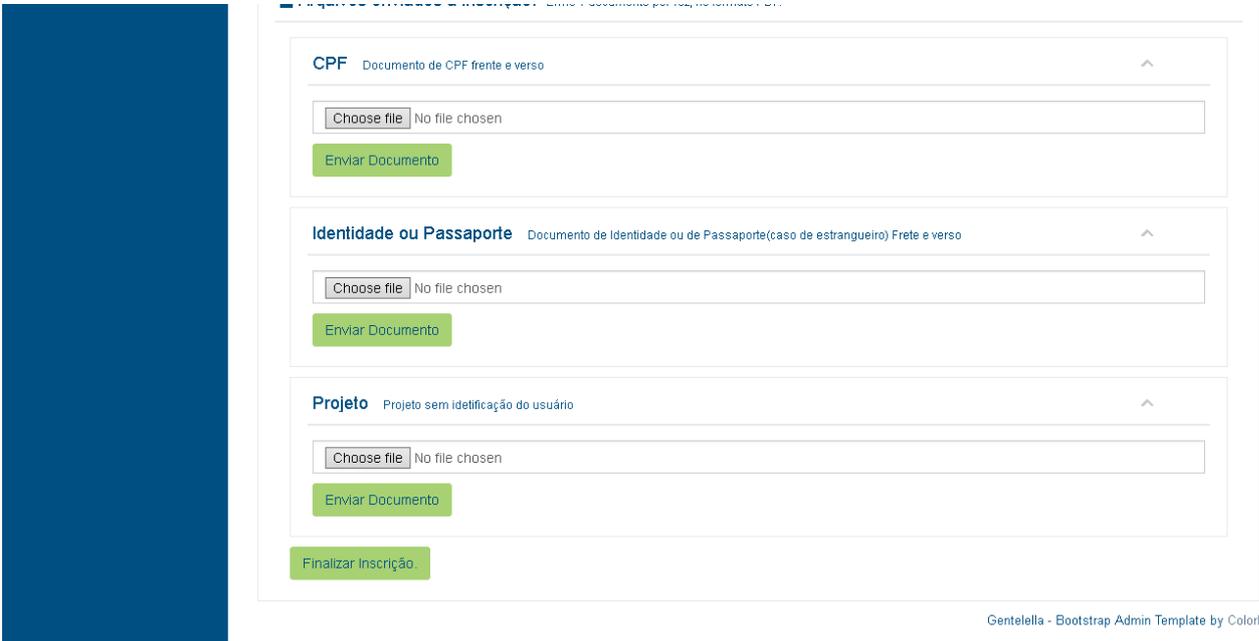
Figura 10 - Pagina com Dados Detalhados da inscrição

The screenshot displays the APPGIN system interface. On the left is a dark blue sidebar with the APPGIN logo and navigation links: 'Dados Pessoais', 'Inscrições', 'Editais', and 'Secretaria'. The main content area shows the registration ID '2018021000001' and the title 'Dados Submetidos a inscrição.'. Below this, it states 'Inscrição 2018021000001 , Inscrição não finalizada'. The data is organized into two columns of key-value pairs, each with a small icon representing the field type. A green 'Editar Dados' button is located at the bottom right of the data area.

Dados Submetidos a inscrição.	
Inscrição 2018021000001 , Inscrição não finalizada	
CPF: 12345678912	Data de Nascimento: 2018-06-04 00:00:00
Lingua Estrangeira: Inglês	Estado Civil: C
Sugestão de Avaliação: Professor Dr 1	Nome do Pai:
Nome do Candidato: Usuário 1	Orgão Emissor do RG: MG
E-mail do Candidato: usuario1@email.com	Bairro: Centro
Passaporte:	Estado: Minas Gerais
Nome da Mãe: Mãe do User1	Telefone 2: 0
Endereço : Averidade Primerio	Orgão Emissor do RG: MG
Cidade : Cidade	Data de Emissão do RG: 2014-06-02 00:00:00
Codigo Postal: 11111-111	Pais de Nascimento: Brasil
Telefone 1: 1234567	Numero do Lattes: 111111111111111111
Telefone 3: 0	Linha de Pesquisa: Linhas de Pesquisa 1
Numero do RG: 123123	Escolaridade: 1
Estado do RG: MG	Pais de Residencia: Brasil
Oferta de Vagas: Ampla Concorrência	

Fonte: Sistema APPGIN, próprio autor

Por ser uma sessão com muitas informações e requisições a apresentação da tela foi 2 dividida em 2 telas.

Figura 11 - Pagina com Dados Detalhados da Inscrição, continuação

The screenshot displays a web form for document uploads. It features three distinct sections, each with a title, a description, a file selection input, and a submit button:

- CPF**: Document of CPF front and back. Includes a "Choose file" button and an "Enviar Documento" button.
- Identidade ou Passaporte**: Document of Identity or Passport (case of foreigner) front and back. Includes a "Choose file" button and an "Enviar Documento" button.
- Projeto**: Project without user identification. Includes a "Choose file" button and an "Enviar Documento" button.

At the bottom of the form, there is a "Finalizar Inscrição" button. The footer of the page reads "Gentelella - Bootstrap Admin Template by Colorlib".

Fonte: Sistema APPGIN, próprio autor

Uma grande característica e principal diferencial do *Laravel* frente a outros *frameworks* é a compatibilidade com o *Eloquent*, uma classe ORM (*Object-Relational Mapping*), para buscas e tratamentos de dados provenientes do banco de dados. O *Eloquent* tem como principal característica facilitar as buscas simples e abstrair as *queries* mais elaboradas a simples chamadas de funções PHP. Esta ferramenta permite oferecer uma infinidade de buscas e relacionamentos de dados dentro do sistema. Na Figura 12 é mostrado uma lista de inscrições, onde o professor ou administrador pode buscar ou filtrar um determinado dado.

Figura 12 Lista de inscrições

APPGIN Bem Vindo, Administrador Padrão.

FUNÇÕES DO SISTEMA

- Dados Pessoais
- Sistema
- Usuários
- Editais
- Secretaria

Inscrições Listas de inscrições em Editais.

Inscrição	Edital	Situação 1 (Finalizada/Em andamento)	Data da Inscrição	Ações
20180229000001	29 - PRPPG/UFVJM	"Em Andamento"	"Em Análise"	May 10, 2018
20180229000002	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 10, 2018
20180229000003	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 10, 2018
20180229000004	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 11, 2018
20180229000005	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 11, 2018
20180229000006	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 11, 2018
20180229000007	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 12, 2018
20180229000008	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 13, 2018
20180229000009	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 13, 2018
20180229000010	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 14, 2018
20180229000011	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 14, 2018
20180229000012	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 14, 2018
20180229000013	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 14, 2018
20180229000014	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 15, 2018
20180229000015	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 15, 2018
20180229000016	29 - PRPPG/UFVJM	"Finalizada"	"Em Análise"	May 15, 2018

Fonte: Sistema APPGIN, próprio autor

5. CONSIDERAÇÕES FINAIS

Considerando a crescente média de inscritos por processo seletivo no PPGEd, gerando demanda de servidores públicos e a dificuldade em entregar documentos nos campi da UFVJM, se faz necessário um sistema de informação que facilite as inscrições e que permita uma melhor gestão e organização dos documentos dos candidatos que desejam ingressar no programa de mestrado profissional.

O Sistema de Gerenciamento de Inscrições foi desenvolvido a partir da justificativa de buscar melhorar o processo já existente, tornar fácil o acesso ao programa de pós-graduação e reduzir o número de papéis enviados a universidade, adequando-a às orientações da PSL da UFVJM.

Através do desenvolvimento do trabalho, pode-se concluir que os sistemas de informação tornam os processos mais rápidos e íntegros, oferecendo agilidade na gestão da informação e controle dos dados.

Foram utilizados conceitos de gerencia de projetos, como *Kanban*, modernos *frameworks* e ferramentas para gestão de equipe. Tais conceitos e ferramentas facilitaram o surgimento de um sistema moderno e alinhado aos padrões de projetos mais utilizados no mercado.

Como forma inicial de aplicação e testes, o sistema apresentado foi utilizado no processo seletivo de 2018, edital nº 29/2018. Neste processo seletivo foram realizadas 382 inscrições, e recebendo mais de 4 mil documentos. Durante este processo foram recebidos 3 inscrições de candidatos que residem fora do estado de Minas Gerais, comprovando a hipótese que o sistema ajudou os candidatos que não tem acesso a secretaria do PPGEd.

Durante este processo seletivo aconteceram picos de acessos nos últimos dias de inscrições, ocasionando lentidão na conexão. Mas observou-se que estes fatos não atrapalharam o funcionamento do sistema proposto neste trabalho, apenas demonstravam lentidão na renderização das *Views*. Logo o sistema conseguiu se manter íntegro e funcional.

Como sugestão de trabalhos futuros o APPGIN atende a primeira etapa do processo seletivo, sendo possível a expansão para as demais fases, como a de avaliação do pré-projeto. O sistema apresentado neste trabalho está preparado para atender a expansão, visto que depende de atribuir novas funcionalidades ao ator professor. Além das novas funcionalidades é importante adicionar estatísticas e informações sobre os processos abertos a páginas iniciais dos usuários tipos Professor e Administrador. Informações como, número de inscrições, relação de documentos enviados.

REFERENCIAS

- ASANA. Use Asana, 2018. Disponível em: <<https://asana.com>>. Acesso em: 01 jul. 2018.
- ASSESSORIA DE MEIO AMBIENTE. Plano de Gestão de Logística Sustentável -PLS. **UFVJM**, 2012. Disponível em: <http://www.ufvjm.edu.br/formularios/doc_view/2541.html?lang=pt_BR.utf8%2C+pt_BR.UT>. Acesso em: 10 jul. 2018.
- AZEVEDO JUNIOR, D. P. D.; CAMPOS, R. D. Definição de requisitos de software baseada numa arquitetura de modelagem de negócios. **Produção**, v. 18, n. 1, p. 26 - 46, Jan/Abr 2008.
- BASTOS, R.; JAQUES, P. A. ANTARES: Um sistema Web de consulta de rotas de ônibus com serviço público. **Revista Brasileira de Computação Aplicada**, p. 41-56, 2010.
- BEAN, M. **Laravel 5 Essentials**. [S.l.]: Packt Publishing Ltd, 2015.
- BEZERRA, PETERSON TUBINI; SCHIMIGUEL, JULIANO. **DESENVOLVIMENTO DE APLICAÇÕES MOBILE CROSS-PLATFORM UTILIZANDO PHONEGAP**. [S.l.]: OBSERVATORIO DE LA ECONOMIA LATINOAMERICANA, 2016.
- BMED, F. Como escolher uma linguagem para desenvolver um sistema. **Fabio Bmed - Blog**. Disponível em: <<http://www.fabiobmed.com.br/como-escolher-uma-linguagem-para-desenvolver-um-sistema/>>. Acesso em: jul. 2018.
- CHACON, S.; STRAUB, B. **Pro Git: Everything you need to know about Git**. [S.l.]: Apress, 2014.
- CONVERSE, T.; PARK, J. **PHP a Bíblia**. 2. ed. Rio de Janeiro: Print book, 2003.
- DOUGLAS, M.; MARABESI, M. **Aprendendo Laravel**. São Paulo: Novatec, 2017.
- EDITORA MELHORAMENTOS LTDA. Dicionário Brasileiro da Língua Portuguesa. **Michaelis**. Disponível em: <<http://michaelis.uol.com.br/busca?r=0&f=0&t=0&palavra=integridade>>. Acesso em: 2018 jul. 2018.
- FARIAS, G. S. **Reengenharia do Sistema de Administração do PPGC**. Porto Alegre: Universidade Federal do Rio Grande do Sul, 2013.
- FILHO, W. D. P. P. **Engenharia de Software**. Rio de Janeiro: LTC, 2013.
- FRANZOLIN, D. V.; CUSTÓDIO, D. D. O. **USO DE CONTROLADOR DE VERSÃO PARA DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE**. Campinas: Embrapa Monitoramento por Satélite - Resumo em anais de congresso (ALICE), 2011.
- FREITAS, M. D. Arquitetura em três camadas. **Marcel Mesmo**, 2011. Disponível em: <<http://marcelmesmo.blogspot.com/2011/10/arquitetura-em-tres-camadas.html#.W0QTa3tjMdV>>. Acesso em: 09 jul. 2018.
- GAMMA, E. et al. **Design patterns: elements of reusable object-oriented languages and systems**. [S.l.]: Addison-Wesley Reading, 1994.

- GUANGCHUN, W. L. L.; HANHONG, X. A novel web application frame developed by MVC. **ACM SIGSOFT Software Engineering Notes** , p. 28, 2003.
- HIRAMA, K. **Kechi. Engenharia de software: qualidade e produtividade com tecnologia.** [S.l.]: Elsevier Brasil, 2012.
- JUNIOR, M. L.; FILHO, M. G. Adaptações ao sistema kanban: revisão, classificação, análise e avaliação. **Gestão e Produção**, São Carlos, v. 15, n. 1, p. 173 - 188, Jan - Abr 2008.
- LEVY, P. **Collective Intelligence: Mankind's Emerging World in Cyberspace.** [S.l.]: PERSEUS BOOKS, 1999.
- LIBRELATO, P. H.; BELUSSO, W. V. **Sistema web para inscrição em atividades acadêmicas.** [S.l.]: Universidade Tecnológica Federal do Paraná, 2014.
- MACÊDO, D. Arquitetura de Aplicações em 2, 3, 4 ou N camadas. **Diego Macêdo: Um pouco de tudo sobre T.I.**, 21 maio 2018. Disponível em: <<http://www.diegomacedo.com.br/arquitetura-de-aplicacoes-em-2-3-4-ou-n-camadas/>>.
- MARTINS, L. E. G.; DALTRINI, B. M. Utilização dos Preceitos da Teoria da Atividade na Elicitação dos Requisitos do Software. **SBES - Simpósio Brasileiro de Engenharia de Software**, Florianópolis, Outubro 1999.
- MICHAELIS, D. **Michaelis On-line**, p. Michaelis, Dicionário. "Disponível em< <http://michaelis.uol.com.br>>." Acesso em 17 (2011)., 2018. Disponível em: <<http://michaelis.uol.com.br>>. Acesso em: 10 Julho 2018.
- MINETTO, E. L. **Frameworks para Desenvolvimento em PHP.** São Paulo: Novatec, 2007.
- MORALES-CHAPARRO, R. et al. MVC web design patterns and rich internet applications. **Proceedings of the Jornadas de Ingenieria del Software y Bases de Datos**, p. 39-46, 2007.
- NARDI, J. C.; FALBO, R. D. A. Uma Ontologia de Requisitos de Software. **CibSE**, p. 111 - 124, 2006.
- NIEDERAUER, J. **Desenvolvendo Websites com PHP.** 2. ed. São Paulo: Novatec Editora, v. 1, 2011.
- NIELSEN, J.; HOA, L. A. **Nielsen, Jakob, and Hoa Loranger. "Usabilidade na web: projetando websites com qualidade.** Rio de Janeiro: Campus, 2007.
- OTWELL, T. **Documentação Oficial do Laravel**, 2018. Disponível em: <<https://laravel.com/docs/5.6>>. Acesso em: 03 jul. 2018.
- PALMEIRA, T. V. V. **Como funcionam as aplicações web.** [S.l.]: [s.n.], 2015.
- RIBEIRO, C. V.; PIVOTO, F. M. F. GERENCIANDO A QUALIDADE DO SOFTWARE COM BASE EM REQUISITOS. **INICIA - Revista de Iniciação Científica da FAI**, Santa Rita do Sapucaí, v. 2, p. 37 - 43, Outubro 2002.
- ROCHA, D.; QUINTÃO, F.; MARTINS, S. A Integridade de Dados em Sistemas Automatizados. **Five Consulting**. Disponível em: <<http://www.fiveconsultoria.com/a-integridade-de-dados-em-sistemas-de-automacao-nas-industrias-de-ciencias-da-vida/>>. Acesso em: 10 jul. 2018.

SENTRY.IO. Documentação Oficial. **Sentry**, 03 Junho 2017. Disponível em: <<https://docs.sentry.io>>.

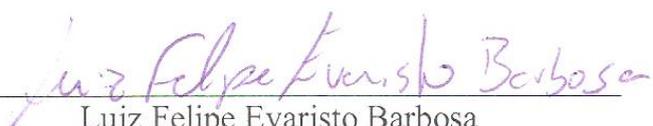
SILVA, D. V. D. S.; SANTOS, F. A. D. O.; NETO, P. S. **Os benefícios do uso de kanban na gerência de projetos de manutenção de software**. VII Simpósio Brasileiro de Sistemas de Informação. [S.l.]: Trilhas Técnicas. 2012. p. 337 - 347.

SILVA, E. Q. D.; MOREIRA, D. D. A. Um Framework de Componentes para o Desenvolvimento de Aplicações Web Robustas de Apoio à Educação. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**, p. 158-167, 2004.

TEMPLE, A. et al. Programação Web com JSP, Servlets e J2EE., 20 maio 2004. Disponível em: <<http://143.0.0.107>>.

AUTORIZAÇÃO

Autorizo a reprodução e/ou divulgação total ou parcial do presente trabalho, por qualquer meio convencional ou eletrônico, desde que citada a fonte.


Luiz Felipe Evaristo Barbosa
lfelipeeb@gmail.com

Universidade Federal dos Vales do Jequitinhonha e Mucuri – Campus JK
Rodovia MGT 367 – km 583, nº 5000, Alto da Jacuba. Diamantina- MG