



Proposta de uma aplicação para gerenciamento de bolsas de estágio e vagas de emprego.

*Wagner Rodrigo Da Silva,
Centro Universitário Academia, Juiz de Fora, MG*

*Tassio Ferezini Martins Sirqueira¹
Centro Universitário Academia, Juiz de Fora, MG*

Linha de Pesquisa: Engenharia de Software

Resumo

Este trabalho tem como objetivo apoiar os graduandos da instituição na busca de oportunidades de bolsas de estudo ou para vagas no mercado de trabalho na região, visto que há uma grande procura de profissionais que estão se qualificando para a área de tecnologia, por meio da construção de uma plataforma adequada para este fim. A plataforma foi desenvolvida para facilitar a busca por estes formandos a fim de consolidar os conhecimentos aprendidos em sala de aula com a demandas do mercado de trabalho, uma vez que esse graduando deverá sempre estar atualizado com as práticas acadêmicas e com as metodologias adotadas no mercado de trabalho.

Palavras-chave: Vaga de trabalho, oportunidades, plataforma de vagas, bolsa de estágio.

ABSTRACT

This work aims to support the undergraduate students of the institution in the search for opportunities for scholarships or for vacancies in the labor market in the region, since there is a great demand for professionals who are qualifying for the area of technology, by building a platform suitable for this purpose. The platform was developed to facilitate the search for these trainees in order to consolidate the knowledge learned in the classroom with the demands of the labor market, since this undergraduate should always be up to date with academic practices and with the methodologies adopted in the labor market.

Keywords

Job vacancy, opportunities, job platform, internship, scholarship.

¹ Docente do Curso de Sistemas de Informação do Centro Universitário Academia. Orientador.

1. Introdução

Segundo Bohlander, Snell e Sherman (2005) a grande demanda de inovação e mudança tecnológica volta-se para o fato que as habilidades valorizadas hoje podem tornar-se obsoletas amanhã. Com a constante evolução de processos produtivos e econômicos, é natural que se necessita de profissionais que estejam sintonizados e atentos às demandas que o mercado vem necessitando e exigindo para atender às suas necessidades.

Empresa ²de vários portes estão investindo ousadamente em recursos de tecnologia, e isso implica que há mais vagas em promoção no mercado, então, os indivíduos que estão no início de carreira devem estar atento às oportunidades. Ao mesmo tempo a concorrência mais acirrada por talentos significa que algumas pessoas serão atraídas para trabalhar em outras empresas concorrentes. Além disso, o trabalho remoto, onde a barreira geográfica fica inexistente, propicia mais oportunidades de escolha para os profissionais, onde e qual empresa atende suas expectativas.

Para o graduando que está no início da faculdade e tem objetivo de qualificar-se para atender a demanda que o mercado necessita, contudo, é necessário saber em qual vaga o graduando está apto para desenvolver e contribuir para a empresa, pois se o seu nível de conhecimento não atende a demanda da empresa não será uma boa oportunidade para este graduando, pois existe bastante oportunidade para todos os níveis.

Esta plataforma vem para ajudar os alunos da instituição UniAcademia, no sentido de selecionar e analisar as melhores oportunidade de bolsas de estágio ou vagas para contratação liberadas pelo mercado local ou internacional, visto que normalmente são vagas que necessitam de senioridade intermediária ou superior. A plataforma desenvolvida irá trabalhar com vagas liberadas pela instituição para

² disponível em: <https://www.indi.mg.gov.br/minas-gerais/setores-de-destaque/tics/>

disponível em:

https://www.em.com.br/app/noticia/tecnologia/2022/03/02/interna_tecnologia,1349529/minas-gerais-e-o-terceiro-estado-em-numero-de-startups-no-brasil.shtml

atender seus alunos em sua necessidade de estágio, pois há uma demanda elevada pelos formandos à procura de colocações, à medida em que vão se aproximando de sua formatura e posteriormente à conclusão do curso.

Na seção 2 apresenta-se algumas informações sobre os requisitos e tecnologias utilizados no desenvolvimento desta plataforma. Ao longo da seção 3 demonstrar sobre a arquitetura e os detalhes dela. Já na seção 4 demonstrará algumas limitações acerca da plataforma, nossas considerações finais e os trabalhos futuros com possibilidade de desenvolvimento.

2. Referencial Teórico

A plataforma deve atender as necessidades dos graduandos na busca de bolsa de estágio ou emprego em suas diferentes áreas de conhecimento, visto que a instituição visa facilitar e selecionar as melhores oportunidade para seus alunos.

Descreve-se quais as tecnologias utilizadas no desenvolvimento da plataforma e como ela está estruturada.

Como mencionada anteriormente a plataforma foi desenvolvida na linguagem de programação PHP juntamente com o **framework Laravel**³e o **template Voyager**⁴, para armazenar as informações da aplicação, foi empregado o sistema gerenciador de banco de dados PostgreSQL.

2.2 Tecnologias utilizadas

Sobre as tecnologias e software escolhidos para a criação do projeto demos preferência em usar projetos de código aberto “*Open Source*” mantido por uma comunidade ou grupo de utilizadores da tecnologia em questão.

A plataforma está voltada para o meio web tem em vista atender a dispositivos moveis e a computadores tradicionais, por esses motivos foi utilizado a linguagem *PHP* segundo o manual da linguagem

³disponível em: <https://laravel.com/>

⁴disponível em: <https://github.com/the-control-group/voyager>

O *PHP* (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de script open Source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do *HTML*.

Juntamente com *Bootstrap* ⁵ um conhecido framework de interface onde implementa recursos e padrões de *HTML*, *CSS* e *Javascript*.

Outro ponto importante para atender a dispositivos moveis foi utilizar o conceito de *Design responsivo* onde utilizamos técnicas capazes de permitir que a interface da aplicação ou de sites seja redesenhada ou adaptada para qualquer tipo de tamanho de tela onde a aplicação esteja sendo executada, tornando-a responsiva sem a perda de suas características de design e sua identidade visual para o meio em questão. Essa técnica é amplamente utilizada no mercado pois diminui o trabalho das equipes de criação e desenvolvimento pois dispensa a necessidade de criar outras versões para tamanhos de telas diferentes.

Para entender a os recursos utilizados na plataforma e sua arquitetura em que foi constituída, é necessário apresentar o *framework Laravel*, que é um *Framework Open Source* criado pelo desenvolvedor **Taylor B Otwell** ⁶sobre a *licença MIT*, com o propósito de ser uma alternativa mais avançada aos frameworks existentes no mercado, na data de seu nascimento, em meados de 2011 e seu código fonte está disponível no *GitHub* ⁷

Em uma tradução direta de sua página no *GitHub* ⁸o criador do *Laravel* o denomina da seguinte maneira:

“*Laravel* é um *framework* de aplicação web com sintaxe expressiva e elegante. Acreditamos que o desenvolvimento deve ser uma experiência agradável e criativa para ser verdadeiramente gratificante. O *Laravel* elimina a dor do desenvolvimento facilitando tarefas comuns usadas em muitos projetos da web, como:

⁵ disponível em: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

⁶ disponível em: <https://github.com/taylorotwell>

⁷ disponível em: <https://github.com/the-control-group/voyager>

⁸ disponível em: <https://github.com/>



- Mecanismo de roteamento simples e rápido.
- Poderoso contêiner de injeção de dependência.
- Vários *back-ends* para sessão e armazenamento em *cache*.
- *ORM*⁹ de banco de dados expressivo e intuitivo.
- Migrações de esquema agnóstico de banco de dados.
- Processamento de trabalho em segundo plano robusto.
- Transmissão de eventos em tempo real.
- O *Laravel* é acessível, poderoso e fornece as ferramentas necessárias para aplicativos grandes e robustos.”

Outro componente utilizado no projeto foi o template Admin Voyager¹⁰ que fornece um conjunto de ferramentas para a criação, deleção, e edição de dados. Já para o SGBDR “sistema gerenciador de banco de dados objeto relacional” utilizou-se utilizar o PostgreSQL um conhecido e famoso SGBDR referência no mercado de sistemas para o gerenciamento de dados.

2.3 Requisitos do sistema

Requisitos Funcionais

RF1 - O sistema deve permitir ao coordenador de curso cadastrar alunos , contendo a sua identificação, como o nome do curso, se possui estágio obrigatório, sua carga horária mínima, e a partir de qual período pode o aluno fazer esse tipo de estágio, se permite estágio não obrigatório, assim como os dados do supervisor de estágio do curso (nome, Email, celular).

RF2 - O sistema deve permitir à concedente de estágio se cadastrarem, informando nome da concedente, tipo (pessoa jurídica / profissional liberal), CNPJ para pessoa jurídicas, CPF e registro em conselho profissional (opcional) para pessoas físicas, endereço completo, telefone de contato, senha, Email institucional, número de empregados, nome do representante, rg, CPF, celular e Email pessoal. Todos os dados são obrigatórios.

⁹ Object relational mapping ou Mapeamento objeto relacional uma técnica de desenvolvimento onde as tabelas do banco de dados são representadas em classes e os registros das tabelas seriam instâncias dessas classe

¹⁰ disponível em: <https://github.com/the-control-group/voyager>



RF3 - O sistema deve permitir que o aluno se cadastre no aplicativo, informando nome completo, curso, matrícula, Email, celular, senha, fotografia, e se é portador ou não de deficiência (informando o tipo em caso positivo).

RF4 - O sistema deve permitir que os funcionários da Central de Estágios se cadastrem, informando nome completo, cargo, matrícula, Email, celular, senha e fotografia.

RF5 - O sistema deve permitir que concedentes de estágio disponibilizem vagas aos alunos da instituição, informando título e descrição da vaga, carga horária, turno/horários de trabalho, dias da semana, habilidades obrigatórias, habilidades desejadas, valor da bolsa (opcional), benefícios, a partir de que período o aluno deve estar cursando (opcional), cursos abrangidos, data de início e término para inscrição/envio de currículo, link para inscrição ou Email para envio de currículo. Vagas cadastradas só são divulgadas aos alunos depois que o coordenador do curso ou supervisor de estágio do curso autorizem.

RF6 - O sistema deve notificar o coordenador do curso ou o supervisor de estágio por e-mail e notificação no aplicativo quando uma vaga for cadastrada por uma concedente.

RF7 - O sistema deve permitir ao coordenador do curso ou ao supervisor de estágio consultar as vagas cadastradas por concedentes aprovando ou não a divulgação para os alunos dos cursos informados na vaga.

RF8 - O sistema deve notificar os alunos dos respectivos cursos associados à vaga quando forem aprovadas para divulgação, tanto por Email quanto por notificação no aplicativo.

RF9 - O sistema deve permitir aos alunos consultar as vagas em aberto e associadas ao seu curso, ofertadas pelas concedentes até a data de término das inscrições.

RF10 - O sistema deve permitir à central de estágios cadastrar convênios com concedentes, informando todos os dados da mesma (RF2 caso não esteja cadastrada), data de início do convênio e data de término do convênio.

RF11 - O sistema deve permitir à central de estágios emitir o termo de convênio em formato pdf para respectivas assinaturas. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.

RF12 - O sistema deve permitir à central de estágios editar o cadastro de convênio anexando documento pdf relativo a esse termo, assinado pelos interessados, bem como outros documentos em pdf associados às comprovações necessárias. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.

RF13 - O sistema deve notificar aos funcionários da central de estágios quando os termos de convênio estiverem prestes a vencer (30 dias antes), por Email e notificação no celular.

RF14 - O sistema deve permitir à concedente informar um estagiário selecionado para a central de estágios, contendo o CPF do aluno, data de início e término do estágio (não pode ser superior a 2 anos, exceto se estagiário portador de deficiência), carga horária semanal (máximo 30 horas), supervisor do estágio na empresa (nome completo, Email, rg, CPF, área de formação, celular), número da apólice de seguro do aluno (opcional), plano de atividades. O sistema deve verificar o limite de estagiários na concedente conforme legislação (arredondado para cima em caso de fração), sendo de 1 (um) a 5 (cinco) empregados: 1 (um) estagiário; de 6 (seis) a 10 (dez) empregados: até 2 (dois) estagiários; de 11 (onze) a 25 (vinte e cinco) empregados: até 5 (cinco) estagiários; acima de 25 (vinte e cinco) empregados: até 20% (vinte por cento) de estagiários. Este requisito independe se a concedente disponibilizou a vaga anteriormente no sistema. A central de estágios deve ser notificada por e-mail e notificação no aplicativo desta ação de cadastramento. Deve ficar registrado data e hora deste cadastro.

RF15 - O sistema deve permitir à central de estágios, cadastrar termo de compromisso a partir da informação de um aluno selecionado para estágio por uma concedente. Para isso, o sistema deve verificar se existe termo de convênio válido com a concedente. Também deve informar se o aluno está em situação de matriculado na instituição (preferencialmente integrado com sistema acadêmico). Deve ainda preencher o número da apólice de seguros caso não tenha sido informado pela concedente. O sistema deve indicar se o estágio é obrigatório ou não obrigatório dependendo das configurações do curso e momento do aluno no curso, permitindo alteração. O sistema deve preencher o supervisor de estágio acadêmico com o supervisor de estágio do curso, ou o coordenador do curso na falta de um supervisor de estágio cadastrado no curso. Deve ficar registrado data e hora deste cadastro.

RF16 - O sistema deve notificar o supervisor acadêmico de estágio ou o coordenador do curso na falta do primeiro, via Email e notificação no aplicativo, para aprovação do plano de atividades nos termos de compromisso cadastrados.

RF17 - O sistema deve permitir à central de estágios emitir o termo de compromisso em formato pdf para respectivas assinaturas. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.

RF18 - O sistema deve permitir à central de estágios editar o cadastro de termo de compromisso anexando documento pdf relativo a esse termo, assinado pelos interessados, bem como outros documentos em pdf associados às comprovações necessárias. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.



RF19 - O sistema deve notificar a concedente e ao estagiário quando do preenchimento do relatório semestral de atividades. Isso deve acontecer a cada seis meses de estágio ou término do semestre letivo, o que acontecer primeiro. Também deve ser emitido quando do término do estágio ou cancelamento do mesmo. O relatório semestral de atividades consta das seguintes informações: identificação das partes (concedente, supervisor na empresa, supervisor acadêmico, aluno), data de início e término, carga horária semanal, cálculo da carga horária cumprida, descrição das atividades, diferentes critérios de avaliação cadastrados para a instituição como um todo, podendo ser avaliados como muito bom: (MB), Bom: (B), Regular: (R), Não se Aplica: (NA).

RF20 - O sistema deve notificar o supervisor de estágio do curso, ou o coordenador na falta do primeiro, para que efetue aceite do relatório semestral de atividades. Somente depois disso e da assinatura do respectivo documento, as horas podem ser creditadas ao aluno, dando baixa na quantidade de horas de estágio obrigatório (quando for esse o caso).

RF21 - O sistema deve permitir à central de estágios editar o cadastro de estágio anexando documento pdf relativo aos relatórios semestrais de atividades, assinado pelos interessados. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.

RF22 - O sistema deve notificar a concedente, estagiário e central de estágios via Email e notificação no aplicativo quando o estágio estiver próximo a finalizar (30 dias antes).

RF23 - O sistema deve permitir à concedente a renovação de estágio através de cadastro de termo aditivo. Deve-se observar as mesmas regras definidas anteriormente para a celebração de aditivos. Essa solicitação deve ser notificada à central de estágios e ao estagiário via Email e notificação no aplicativo.

RF24 - O sistema deve permitir à central de estágios editar o cadastro de estágio anexando documento pdf relativo a termos aditivos, assinado pelos interessados. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.

RF25 - O sistema deve permitir à concedente o cancelamento de estágio através de cadastro de termo de cancelamento. Essa solicitação deve ser notificada à central de estágios e ao estagiário via Email e notificação no aplicativo.

RF26 - O sistema deve permitir à central de estágios editar o cadastro de estágio anexando documento pdf relativo a termos de cancelamento, assinado pelos interessados. Verificar viabilidade de integração com a plataforma *AssineBem* via *API*.



RF27 - O sistema deve permitir à central de estágios registrar mudanças na situação acadêmica do aluno, como trancamento, abandono ou formatura, com a respectiva data, passando a situação de estágios em andamento para cancelado. Verificar possibilidade de integração com sistema acadêmico via *API*.

RF28 - O sistema deve notificar a concedente via Email e notificação no aplicativo quando da mudança da situação curricular dos alunos em estágio, informando sobre o cancelamento do estágio e as providências que precisam ser tomadas relativo ao respectivo documento de cancelamento e avaliação de estágio.

RF29 - O sistema deve permitir ao coordenador, supervisor de estágio acadêmico, central de estágios e aluno consultar todos os estágios, tanto concluídos quanto em andamento, apresentando a carga horária cumprida em estágio não obrigatório e obrigatório (neste último, indicando o quanto ainda falta para cumprir).

RF30 - O sistema deve enviar Email para o coordenador do curso, supervisor de estágio acadêmico e aluno sempre que um aluno cumprir as regras para conclusão de estágio obrigatório do curso, colocando no corpo do Email uma tabela com os estágios obrigatórios cumpridos, respectiva carga horária, anexando as imagens dos comprovantes.

RF31 - O sistema deve permitir ao aluno cadastrar solicitação de aproveitamento de atividades profissionais como estágio obrigatório, informando matrícula, curso, data de início e término (opcional este último) da atividade profissional, cargo, dados da empresa (nome, CNPJ, Email, telefone, chefe imediato, CPF, celular, Email pessoal), anexando documentos pdf das páginas da carteira de trabalho, declaração da empresa e outros documentos que se fizerem necessários.

RF32 - O sistema deve notificar o supervisor de estágio acadêmico ou coordenador do curso na falta do primeiro, via Email e notificação no aplicativo, para aprovação de aproveitamento de atividades profissionais. No caso de negativa, deve ser informada uma justificativa. No caso de afirmativa, deve ser dado o estágio obrigatório como concluído e enviada notificação ao coordenador do curso e supervisor acadêmico por Email e notificação no celular com a documentação anexada. O resultado da avaliação deve ser enviado por Email e notificação no aplicativo ao aluno.

RF33 - O sistema deve permitir ao coordenador e supervisor acadêmico de estágio a consulta de informações de estágios de um determinado curso em um período (data de início e fim) informados, filtrando por todos, estágios em andamento, estágios concluídos, filtrando também por obrigatório, não obrigatório ou ambos.



RF34 - Caso um coordenador, supervisor ou aluno estejam vinculados a mais de um curso, o sistema deve permitir que cada um desses usuários possa chavear o curso, filtrando as respectivas informações.

RF35 - As funcionalidades previstas para a concedente, coordenador de curso e supervisor acadêmico devem estar disponíveis via aplicativo móvel e interface web. As funcionalidades previstas para os alunos devem estar disponíveis apenas via aplicativo móvel.

RF36- O sistema deve utilizar um banco de dados relacional padrão ANSI e gratuito, como o PostgreSQL

RF37 - Todos os acessos devem ser feitos utilizando o Email do usuário e senha criptografada no banco, utilizando também tokens de acesso.

RF38 - O aplicativo deve permitir a todos os usuários alterar seu nome, Email, celular e fotografia a qualquer tempo.

RF39 - O código fonte deve ser integralmente disponibilizado.

Requisitos Não Funcionais

RNF1 - O sistema deve ter interface para dispositivos móveis, responsiva em função do tamanho da tela, em plataforma híbrida para qualquer sistema operacional móvel.

RNF2 - A interface deve prever padrões dark/light.

RNF3 - As imagens dos documentos devem ser armazenadas separadamente, vinculadas à matrícula do aluno, preferencialmente no Firebase, e de forma protegida.

RNF4 - O sistema deve, em sendo possível, integrar base de usuários com outros sistemas administrativos da instituição. Sendo possível essa situação, substitui-se os requisitos anteriores no que diz respeito ao cadastramento e alteração de dados dos usuários.

RNF9 - O sistema deve ter um link para consulta externa à Lei 11.788/2008:
http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11788.htm

3. Arquitetura do Sistema

Através do diagrama de caso de uso representado pela figura 1 pode-se visualizar como o sistema é composto juntamente com as suas funcionalidades. Dentre as entidades terá, o funcionário; o sistema gerenciador de vaga propriamente dito; interagindo com o sistema.

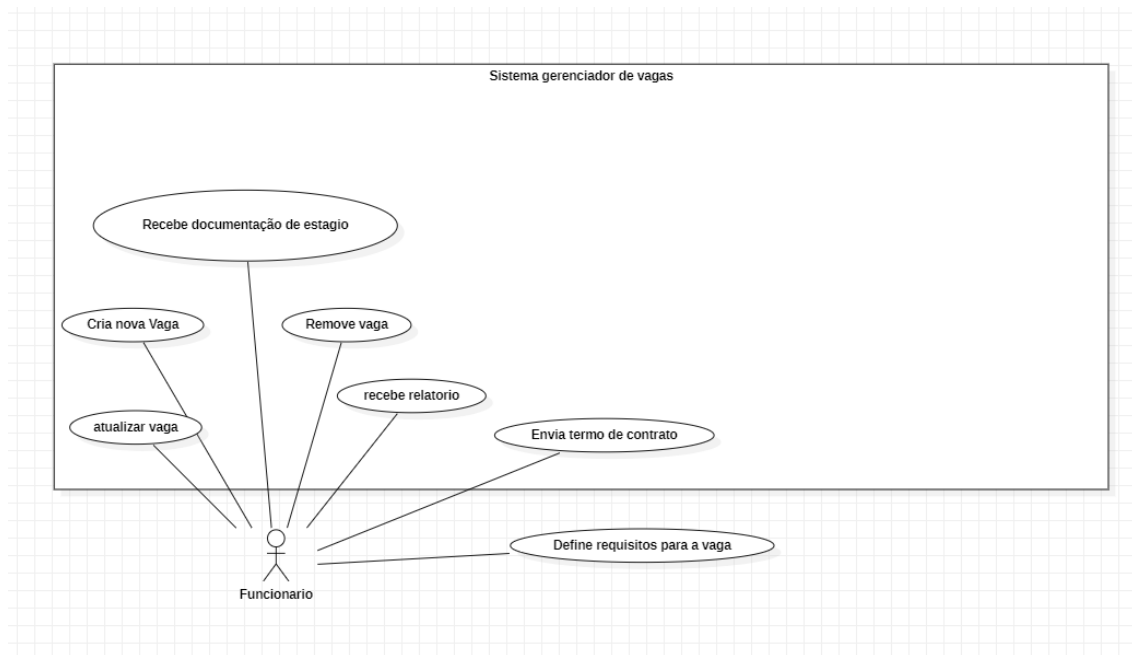


Figura 1: Diagrama de caso de uso.

Fonte: Elaboração própria

Abaixo o diagrama entidade relacionamento que representa a estrutura do banco de dados.

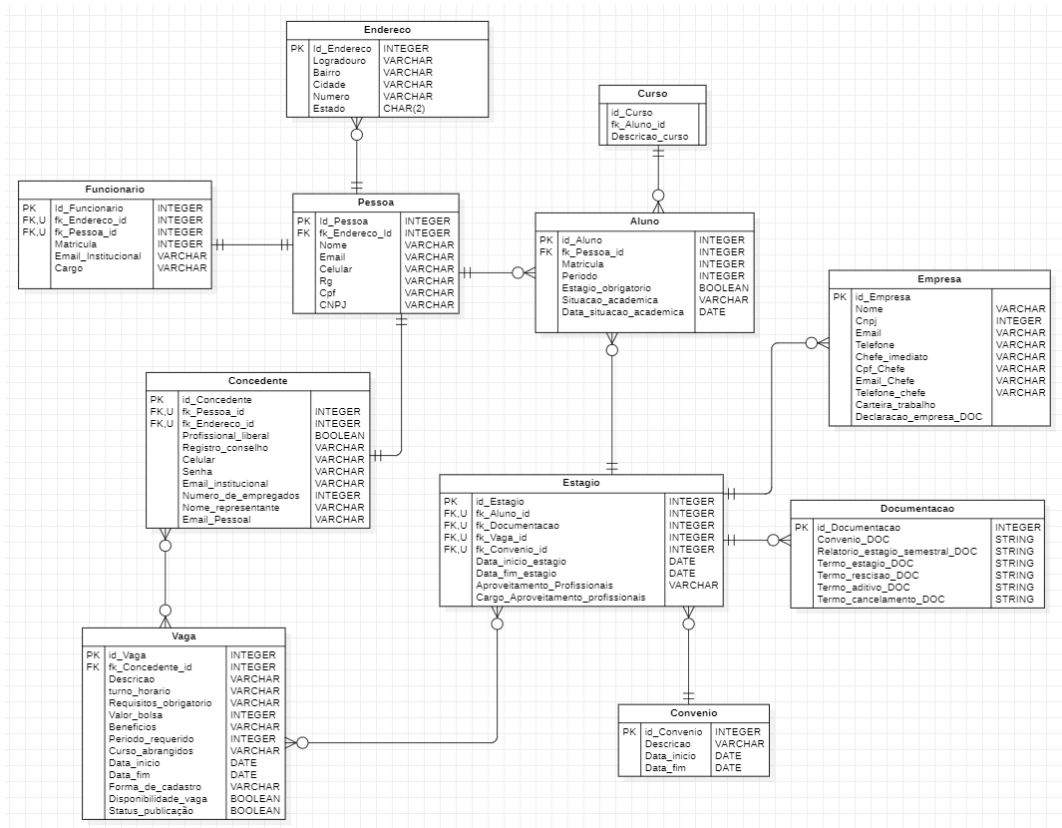


Figura 2: Página de login.

Fonte: Elaboração própria.

O projeto está estruturado em um painel admin onde será possível a customização pela própria instituição, onde não será necessário de desenvolvedores ou alunos da instituição para tarefas cotidianas como a criação, deleção de vagas e atualizações de documentos pela instituição ou empresas parceiras referente as vagas.

Para isso utiliza-se o painel admin, a equipe autorizada acessa o perfil administrador através do login e senha.

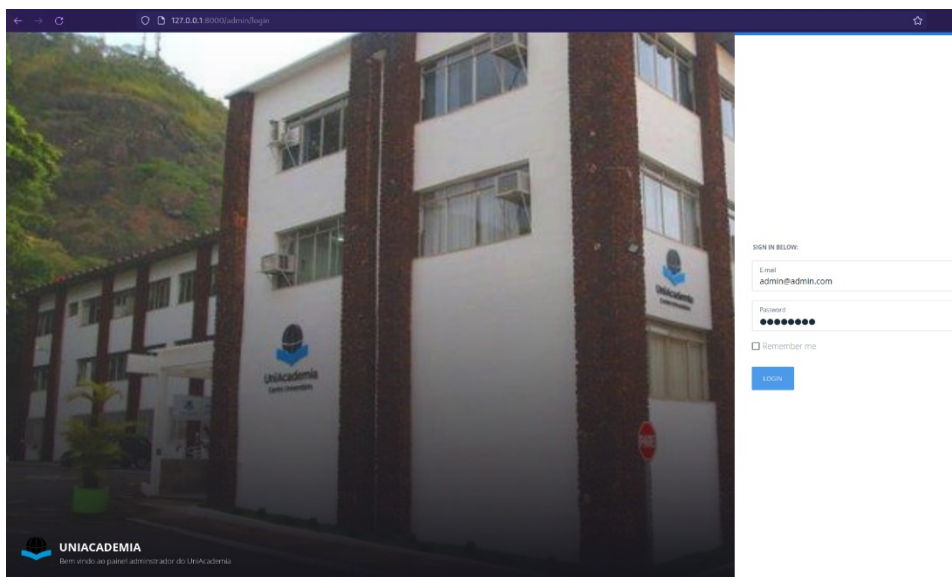


Figura 3: Página de login.

Fonte: Elaboração própria

O painel admin permite que o usuário realize três grupos de operações com suas respectivas ações secundárias, sendo elas:

- Gerenciamento do sistema:
 - Gerenciar de roles;
 - Gerenciar de usuários;
- Criação e consulta dados;
 - Gerenciamento do banco de dados
 - Criar, Editar, Atualizar e Deletar base de dados
 - Gerenciar documentação.

Gerenciamento do sistema

O usuário administrado tem acesso através de login e senha ao painel admin onde executa as operações de gerenciar de *roles*(permissões) e gerenciar de usuários, começando pela gerência de permissões acessamos o menu lateral e clicar em “Roles” (figura 4) onde será redirecionados para a página de regras de permissão (figura 5) onde é possível visualizar os

1. Nome da regra de permissão,
2. Descrição que é apresentada para o usuário
3. Ações que pode ser tomada para cada permissão como editar, visualiza e remover
4. Criar permissões
5. Exclusão de permissões

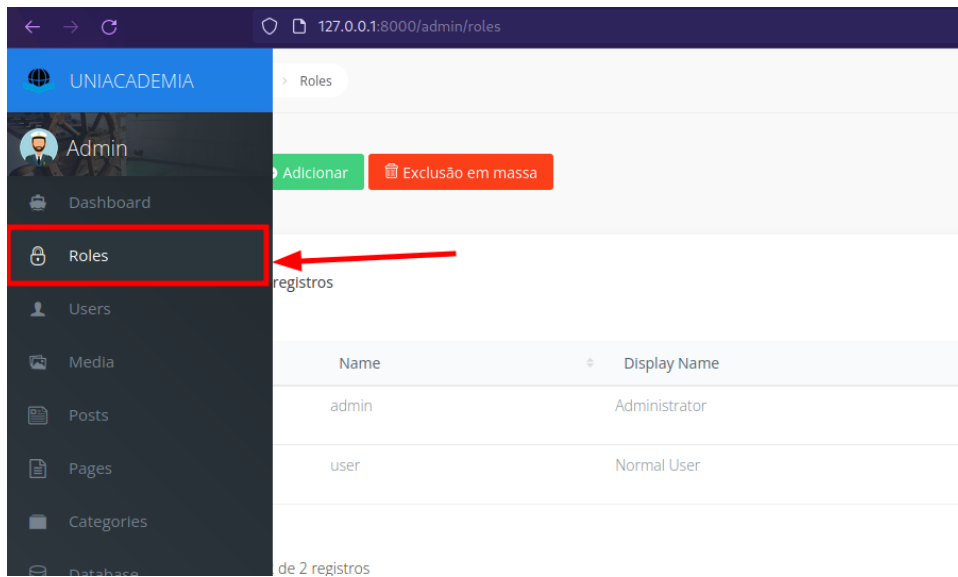


Figura 4: Menu lateral

Fonte: Elaboração própria.

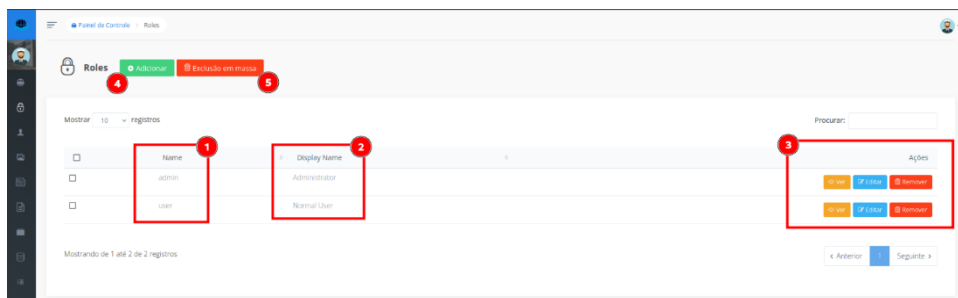


Figura 5: Página de visão geral das permissões dos usuários.

Fonte: Elaboração própria

Ao acessar a opção de editar possui a seguinte tela e opções que são permitidas para cada parte do sistema com os seus respectivos menus.



The screenshot displays the 'Edit' page for a user role in the UniAcademia system. The breadcrumb navigation at the top reads 'Painel de Controle > Roles > 2 > Edit'. The form contains the following fields and sections:

- Name:** A text input field containing the value 'user'.
- Display Name:** A text input field containing the value 'Normal User'.
- Permissões:** A section titled 'Permissões' with a sub-header 'Selecione Todos / Desmarcar todos'. It contains several groups of checkboxes:
 - Browse Admin
 - Browse Bread
 - Browse Database
 - Browse Média
 - Browse Compass
- Menus**
 - Browse Menus
 - Read Menus
 - Edit Menus
 - Add Menus
 - Delete Menus
- Roles**
 - Browse Roles
 - Read Roles
 - Edit Roles
 - Add Roles
 - Delete Roles
- Users**
 - Browse Users
 - Read Users
 - Edit Users
 - Add Users
 - Delete Users

Figura 6: Página de gerência das permissões do usuário.

Fonte: Elaboração própria

Na figura acima temos uma representam do sistema onde cada caixa de seleção está associado a um modulo e a permissão dele. Ao marcar uma das caixas de seleção a funcionalidade e ativa para o usuário escolhido anterior em nosso caso o usuário comum tem acesso a todas as funcionalidade do modulo de usuário isso significa que ele pode visualizar suas informações, inserir novas informações, editar e deletar.

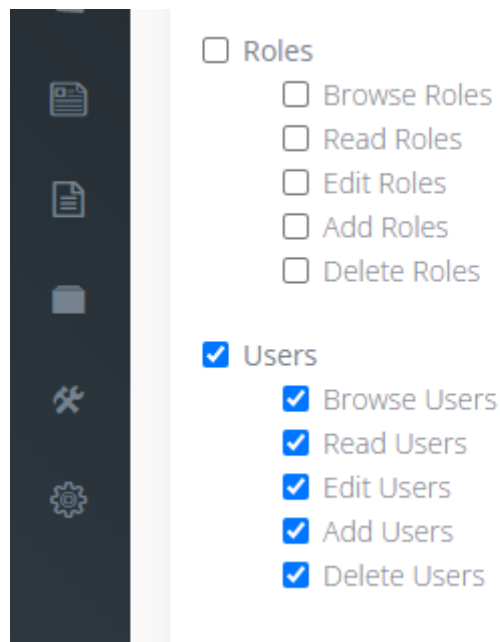


Figura 6.1: Recorte da página de gerência das permissões do usuário.

Fonte: Elaboração própria.

Ao selecionar a caixa de seleção desejada a funcionalidade está habilitada e o contrário também é válido,

Browser Roles: Usuário não navegar no modulo de permissões.

Read Roles: Usuário não pode visualizar as permissões dos demais usuários.

Edit Roles: Usuário não pode editar as permissões dos demais usuários.

Add Roles: Usuário não pode adicionar novas permissões.

Delete Roles: Usuário não pode excluir uma ou mais permissões.

Gerenciador de Usuários

Nesta página fica a gerência dos usuários onde é possível visualizar todos os usuários do sistema com suas respectivas informações como nome, Email, data de criação, avatar, função, permissões e ações que podemos tomar para cada usuário como ver suas informações, editar ou remover.

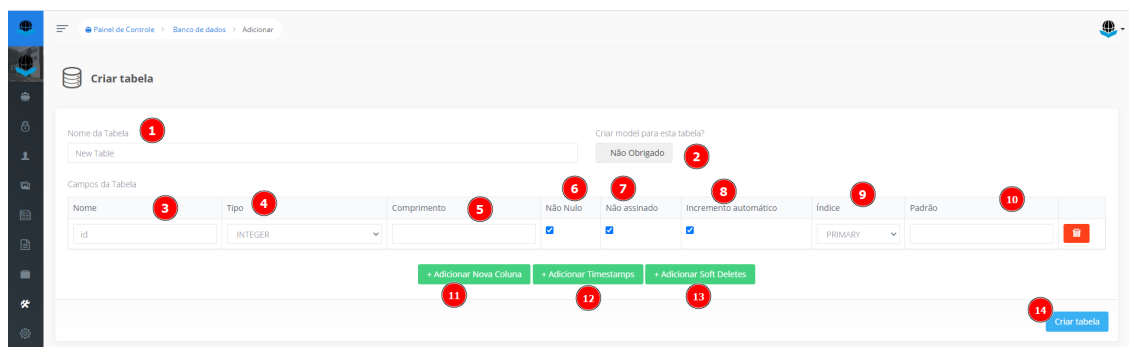


Figura 7: Página principal de gerência de banco de dados.

Fonte: Elaboração própria.

1. Nome da tabela: nome da nova tabela a ser criada.
2. Criação de model: criação do arquivo de model no framework *laravel*.
3. Nome da linha: nome da nova linha.
4. Tipo de dados: tipo de dados que a linha pode assumir.
5. Comprimento: comprimento do tipo de dados da linha.
6. Obrigoriedade do campo: da nova linha.
7. Não assinado.
8. Campo auto incremental: novos dados serão criados automaticamente e de forma incremental.
9. Índice: da linha.
10. Campo padrão.
11. Nova coluna.
12. Coluna de *timestamp*
13. Coluna de soft delete
14. Criar tabela

Gerenciador de mídias

Esta área é responsável por gerência de mídias disponível no sistema sendo fotos, vídeos e vagas.

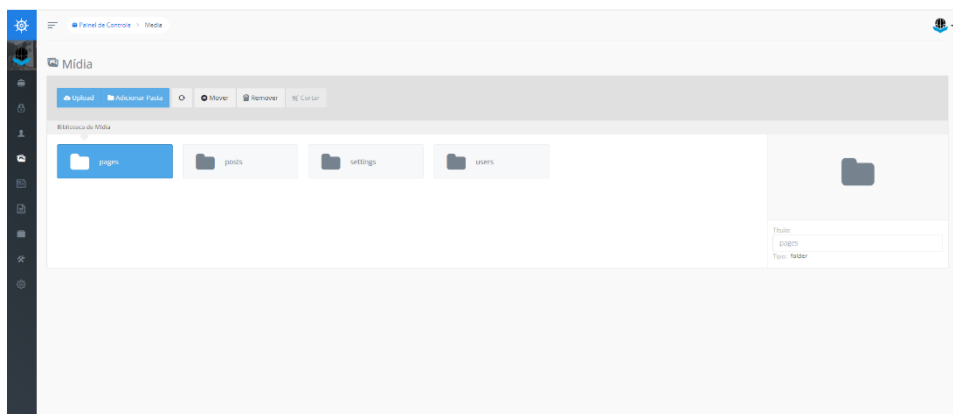


Figura 7: Página principal de gerência de banco de dados.

Fonte: Elaboração própria.

4. Considerações finais

Diante do estudo apresentado, podemos concluir que o software aqui demonstrado, com o objetivo de auxiliar os graduandos da UNIACADEMIA a encontrar vagas existentes no mercado de trabalho em Juiz de Fora e região, de modo a facilitar sua inserção no mesmo, existe diversos pontos fortes. Estes consistem em adequados processos de desenvolvimento de Gerência de mídia, Gerência de banco de dados e Gerência de permissões. este *software* poderá, contudo, sofrer alterações no sentido de otimizar seu desempenho, tendo em vista alguns aos requisitos mencionados anteriormente.

Tais requisitos consistem, por exemplo, na interface com o usuário, que pode ser de difícil compreensão, sobretudo para aqueles que possuem pouca vivência com tecnologia, necessitando de treinamento básico para a correta utilização do sistema. Outro ponto importante é que o usuário administrador possua conhecimento em Gerência e Consulta em Banco de Dados, para o correta manipulação e criação de novos módulos para o sistema.

Para resolver o problema, faz-se necessário desenvolver ferramentas capazes de atender as demandas acima citadas, como a criação de outros *templates* de interface para atender públicos sem treinamento em tecnologia. Da mesma forma, o administrador deve ser previamente selecionado para o trabalho, pois deve possuir minimamente os conhecimentos básicos para desenvolver a criação das tabelas de bancos de dados.



Acreditamos que, seguindo essas premissas, poderemos atender algumas necessidades quanto ao nosso objetivo inicial, no que concerne à aproximação da relação candidato-vaga no mercado de trabalho de Juiz de Fora e região.

Código do projeto

<https://github.com/ces-jf/SGV>

REFERÊNCIAS

PHP. **Hypertext Preprocessor, Página inicial.** Disponível em: < <https://www.php.net> >. Acesso em: 19 de outubro 2022.

PHP. **Manual da linguagem, Página inicial.** Disponível em: < https://www.php.net/manual/pt_BR/intro-whatism.php >. Acesso em: 19 de outubro 2022.

GITHUB DO VOYAGER. **VOYAGER, Página inicial.** Disponível em: <<https://github.com/the-control-group/voyager>>. Acesso em: 19 de outubro 2022.

DESIGN RESPONSIVO. **Design responsivo, Página inicial.** Disponível em: < <https://www.digitalhouse.com/br/blog/design-responsivo/> >. Acesso em: 19 de setembro 2022.

LARAVEL. **LARAVEL, Página inicial.** Disponível em: <<https://laravel.com/>>. Acesso em: 19 de outubro 2022.

POSTGRESQL. **POSTGRESQL, Página inicial.** Disponível em: <https://www.postgresql.org/>>. Acesso em: 19 de setembro 2022.

Taylor Otwell. **Criador do *Framework laravel*.** 1ª versão do Laravel foi lançada junho, 2011. Disponível em: < <https://pt.wikipedia.org/wiki/Laravel/> >. Acesso em: 09 de setembro 2022

BOHLANDER, George; SNELL, Scott; SHERMAN, Arthur. **Administração de recursos humanos.** São Paulo: Pioneira Thomson learning, 2005.