

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**PHASE - UMA APLICAÇÃO PARA AUXILIAR O ALUNO NO
PROCESSO DE ESTÁGIO**

WELIGTON FERREIRA DE ARAÚJO

**Cajazeiras
2022**

WELIGTON FERREIRA DE ARAÚJO

**PHASE - UMA APLICAÇÃO PARA AUXILIAR O ALUNO NO PROCESSO DE
ESTÁGIO**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora

Prof. Dra. Eva Maria Campos Pereira.

Coorientador

Prof. Me. Ricardo de Sousa Job.

**Cajazeiras
2022**

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

A663p Araújo, Weligton Ferreira de.
Phase : uma aplicação para auxiliar o aluno no processo de estágio
/ Weligton Ferreira de Araújo. - 2023.

69f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e
Desenvolvimento de Sistemas) - Instituto Federal de Educação,
Ciência e Tecnologia da Paraíba, Cajazeiras, 2023.

Orientador(a): Prof^ª. Dra. Eva Maria Campos Pereira.
Co-orientador(a): Prof. Me. Ricardo de Sousa Job.

1. Desenvolvimento de sistemas. 2. Sistema Phase. 3.
Desenvolvimento de pessoas. 4. Estágio supervisionado. I. Instituto
Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.45

ATA 1/2023 - CLICCOMP/UNINFO/DDE/DG/CZ/REITORIA/IFPB

**ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO (TCC)
CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS (ADS)**

Às 08h30 do dia 27 do mês de MARÇO do ano de 2023, o(a) **aluno(a) WELIGTON FERREIRA DE ARAÚJO, matrícula 201912010026**, apresentou, como parte dos requisitos para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, seu trabalho de conclusão de curso, tendo como título "**PHASE - UMA APLICAÇÃO PARA AUXILIAR O ALUNO NO PROCESSO DE ESTÁGIO**". Constituíram a banca examinadora os professores **Eva Maria Campos Pereira** (orientadora), **Leandro Luttiane da Silva Linhares** (examinador) e **Afonso Serafim Jacinto** (examinador).

Após a apresentação e as observações dos membros da Banca Examinadora, ficou definido que o trabalho foi considerado **APROVADO** com nota **90**, com a condição de que o (a) aluno (a) entregue, no prazo máximo de 30 dias, a versão final do trabalho com as correções sugeridas pelos membros da banca examinadora. Eu, **FÁBIO ABRANTES DINIZ**, Coordenador do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, lavrei a presente ata, que segue assinada digitalmente por mim e pelos membros da banca examinadora.

Cajazeiras, 29 de março de 2023.

Documento assinado eletronicamente por:

- **Fabio Abrantes Diniz, COORDENADOR(A) DE CURSOS - FUC1 - CADS-CZ**, em 29/03/2023 08:35:37.
- **Eva Maria Campos Pereira, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/03/2023 08:38:36.
- **Weligton Ferreira de Araujo, ALUNO (201912010026) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 29/03/2023 08:40:57.
- **Afonso Serafim Jacinto, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/03/2023 08:57:13.
- **Leandro Luttiane da Silva Linhares, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/03/2023 09:18:45.

Este documento foi emitido pelo SUAP em 29/03/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código: 410117
Verificador: 968474347b
Código de Autenticação:



*Dedico este trabalho a todos os estudantes,
para que através deste, haja uma contri-
buição relevante para o aprendizado e a
formação dos mesmos.*

AGRADECIMENTOS

Agradeço à Deus por conduzir minha vida durante todos esses anos, permitindo-me viver dias felizes ao lado de pessoas maravilhosas e de realizar sonhos muito especiais. Agradeço à minha família, em especial a minha mãe, por ter me incentivado e me dado forças até o presente momento. Gostaria de agradecer, também, aos meus queridos orientadores pela disposição em me direcionar durante cada etapa. Agradeço a cada amigo que me ajudou neste processo, em especial ao meu amigo de longa data e colega de curso, José Ricardo. Por fim, agradeço o apoio de alguém que amo e que é muito especial pra mim, Gaby Figueiredo.

RESUMO

O estágio é um exercício educacional supervisionado que possui como principal objetivo, possibilitar que os discentes obtenham aprendizado e experiência através da atuação no mercado de trabalho. Para tanto, é necessário consolidar um convênio entre os alunos de uma determinada instituição de ensino e uma empresa pública ou privada, por meio de um processo que permeia desde a assinatura do contrato até os resultados obtidos. No entanto, há dificuldades na busca por vagas que exijam qualidades técnicas que estejam alinhadas com o que o discente procura. Os orientadores também enfrentam dificuldades para acompanhar os seus orientandos de forma eficaz e satisfatória. O objetivo desse trabalho é desenvolver a aplicação *Web Phase*, que visa, facilitar o ingresso dos discentes no processo de estágio, fomentar a aprendizagem e a capacitação dos alunos e aperfeiçoar a assistência prestada pelos orientadores durante esse processo.

Palavras-chave: Educação. Estágio. Desenvolvimento Pessoal. Phase.

ABSTRACT

The internship is a supervised educational exercise whose main objective is to enable students to obtain learning and experience through working in the job market. Therefore, it is necessary to consolidate an agreement between the students of a particular educational institution and a public or private company, through a process that permeates from the signing of the contract to the results obtained. However, there are difficulties in the search for vacancies that require technical qualities that are in line with what the student is looking for. The supervisors also face difficulties to accompany their supervisees effectively and satisfactorily. The objective of this work is to develop the Web Phase application, which aims to facilitate the entry of students into the internship process, promote student learning and training, and improve the assistance provided by advisors during this process.

Keywords: Education. Internship. Personal Development. Phase.

LISTA DE FIGURAS

Figura 1 – Modelo da arquitetura MVC	27
Figura 2 – Modelo da arquitetura <i>Clean Architecture</i>	28
Figura 3 – Diagrama de Casos de Uso	31
Figura 4 – Modelo Entidade Relacionamento	33
Figura 5 – Diagrama de Classes	34
Figura 6 – Arquitetura da Aplicação	35
Figura 7 – Tela de login de usuários	37
Figura 8 – Tela de cadastro para alunos e professores	37
Figura 9 – Tela de cadastro para instituições e empresas	38
Figura 10 – Tela de cadastro vagas	39
Figura 11 – Tela de cadastro de habilidades do aluno	40
Figura 12 – Tela de edição do perfil do aluno	41
Figura 13 – Tela de postagem do material de apoio	42
Figura 14 – Tela do <i>feed</i> de vagas	43
Figura 15 – Tela de bate-papo	44
Figura 16 – Modelo lógico	49
Figura 17 – Tela do perfil do professor orientador	61
Figura 18 – Tela do perfil de uma empresa	62
Figura 19 – Tela do perfil do discente	63
Figura 20 – Tela do perfil do discente com medalha de ouro	64
Figura 21 – Tela do perfil do discente com medalha de prata	65
Figura 22 – Tela do perfil do discente com medalha de bronze	66
Figura 23 – Tela de projetos do discente	67
Figura 24 – Tela de cadastro de metas	68
Figura 25 – Tela de listagem de metas	69
Figura 26 – Tela do cronograma de atividades	70

LISTA DE QUADROS

Quadro 1 – Requisitos funcionais da ferramenta Phase	32
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ADS	Análise e Desenvolvimento de Sistemas
DOM	<i>Document Object Model</i>
HTML	<i>HiperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IFPB	Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model-View-Controller</i>
NBR	Norma Brasileira
ORM	<i>Object-Relational Mapping</i>
OSS	<i>Open Source Software</i>
RF	Requisito Funcional
SSG	<i>Static Site Generation</i>
SQL	<i>Structured Query Language</i>
SSR	<i>Server Side Rendering</i>
TCC	Trabalho de Conclusão do Curso
UI	<i>User Interface</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Motivação	15
1.2	Problemática	16
1.3	Objetivos	16
1.3.1	Objetivo Geral	16
1.3.2	Objetivos Específicos	17
1.4	Metodologia	17
1.4.1	Resumo do fluxo de trabalho	17
1.5	Organização do Documento	18
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Estágio Supervisionado	19
2.1.1	Definição	19
2.1.2	Objetivo	19
2.1.3	Diretrizes	19
2.1.4	Papel do Orientador	20
2.2	Gamificação	21
2.2.1	Definição	21
2.2.2	Gamificação como ferramenta de apoio ao ensino	21
2.2.3	Aplicação da Gamificação no presente trabalho	21
2.3	<i>React.js</i>	22
2.3.1	Definição	22
2.3.2	Biblioteca <i>Open Source</i>	23
2.3.3	Componentes <i>React.js</i>	23
2.3.4	Desenvolvimento <i>front-end</i>	24
2.4	<i>Next.js</i>	24

2.4.1	Definição	24
2.4.2	Vantagens do <i>Next.js</i>	25
2.4.3	Por que usar o <i>Next.js</i>?	25
2.5	Arquitetura de <i>Software</i>	26
2.5.1	<i>Model-View-Controller</i>	26
2.5.2	<i>Clean Architecture</i>	27
3	PLATAFORMA PHASE	30
3.1	Levantamento de Requisitos	30
3.2	Análise dos Requisitos	30
3.3	Modelagem Conceitual	33
3.4	Arquitetura da Aplicação	34
3.5	Telas dos principais requisitos	36
4	CONSIDERAÇÕES FINAIS	45
4.1	Trabalhos futuros	45
	REFERÊNCIAS	46
	APÊNDICE A – ARQUITETURA DO BANCO DE DADOS DO SISTEMA	48
	APÊNDICE B – PROTÓTIPOS	61

1 INTRODUÇÃO

O Estágio Supervisionado é uma fase muito importante para o desenvolvimento dos alunos, visto que, durante esse período, o discente colocará em prática o que aprendeu na sala de aula em um contexto real de uma empresa ou uma outra organização pública, ou privada.

Segundo Polzin (2019), A inclusão do estudante na esfera profissional por meio do trabalho, proporciona o crescimento profissional, aprendizado, contribui na formação profissional e pessoal, aumentando assim a responsabilidade do aluno. Esse período de amadurecimento desperta o jovem a deixar a sua área de conforto para defrontar-se com o competitivo mercado de trabalho.

Dessa forma, é de suma importância para os discentes, que aprendam e ponham em prática não apenas os conhecimentos técnicos e teóricos obtidos no curso, mas que também venham desenvolver habilidades realmente necessárias referentes à sua profissão, pois são elas que possibilitarão a sua adaptação junto ao mercado de trabalho e que serão necessárias para que o discente torne-se um profissional preparado em seus diversos aspectos.

1.1 MOTIVAÇÃO

Alguns alunos iniciam o processo de estágio supervisionado, geralmente, próximo à conclusão do curso, seja ele Técnico, Subsequente ou Superior. Durante esse processo é necessário garantir que o aluno compreenda como funciona o fluxo de trabalho de uma empresa, onde não se deve apenas pensar nas tarefas a serem cumpridas, mas também de que modo realizá-las.

Dessa forma, para promover esse aprendizado em campo, é de grande importância que os orientadores agreguem, aos seus orientandos, competências técnicas e habilidades de trabalho em grupo, visto que, durante o estágio, o discente deverá atuar em um trabalho real que traz consigo necessidades reais a serem atendidas.

Ademais, é imprescindível que o aluno compreenda, durante esse período, quais procedimentos são corretos ou mais adequados e quais não são, a fim de que suas capacidades sejam lapidadas para tornar-se um exímio profissional, contribuindo, assim, com o desenvolvimento pessoal e da sociedade.

Portanto, através deste trabalho, será desenvolvida uma ferramenta que assista às necessidades dos orientadores e orientandos, dispondo de um ambiente que possua mecanismos de diálogo, disseminação de conhecimento e de planejamento das atividades realizadas durante o estágio.

1.2 PROBLEMÁTICA

Há entre os orientadores, o grande desafio de manter-se à par da situação de cada aluno, visto que é de grande importância acompanhar a progressão do desenvolvimento dos mesmos, quais tarefas estão causando mais dificuldades, o que estão aprendendo, como estão aplicando o novo aprendizado, entre outros aspectos. Estes e outros pontos só podem ser conhecidos se os orientadores tiverem em mãos uma forma de viabilizar a agilidade de comunicação com seus alunos, bem como também, prestar apoio através de materiais didáticos que forneçam um aprendizado claro e objetivo, ao tratar de conhecimentos essenciais e basilares, como questões comportamentais, métodos ágeis, gestão do tempo e trabalho integrado.

De acordo com IFPB (2019), o aluno pode buscar informações sobre vagas de estágio através das listas divulgadas no site, na coordenação de estágio e nos murais pelo IFPB – Campus Cajazeiras, ou por meio de sua própria busca junto às empresas. Dessa forma, ao buscar por uma vaga em algum estágio, há muitos alunos que acabam escolhendo uma vaga por indicação de seus professores ou procurando por si só, o que tem se mostrado efetivo, entretanto, é desejável ter à disposição dos discentes, uma ferramenta capaz de oferecer-lhes uma forma de visualizar e encontrar fácil e antecipadamente as vagas ofertadas e o que cada uma delas requerem daqueles que a procuram, visando facilitar o ingresso dos alunos em uma vaga que detenha de requisitos compatíveis com suas habilidades.

Portanto, a ferramenta proposta neste presente trabalho almeja suprir não somente a demanda por essas buscas, como também, auxiliar os orientadores e orientandos durante esse processo.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver uma plataforma *Web* que vise auxiliar orientadores e discentes durante o processo de estágio.

1.3.2 Objetivos Específicos

Com a finalidade de atingir o objetivo geral deste trabalho, foram elencados os seguintes objetivos específicos:

- Aprofundar o conhecimento sobre o fluxo dos estágios supervisionados por instituições de ensino;
- Estudar o uso da gamificação para aumentar o desempenho dos discentes nos estágios;
- Realizar estudos sobre ferramentas que possibilitem a construção de aplicações *Web* de forma rápida e organizada;
- Aplicar arquiteturas de software que facilitem a manutenção da aplicação à medida que o escopo da mesma aumenta;

1.4 METODOLOGIA

Para a realização do trabalho proposto foi definido um fluxo de atividades. As etapas e o cronograma estão detalhados nas subseções abaixo.

1.4.1 Resumo do fluxo de trabalho

A metodologia empregada para elaboração deste trabalho foi definida seguindo um fluxo dividido nas seguintes atividades:

ATIVIDADE 01 - Identificação da ideia. Nesta etapa inicial do projeto foi definido o escopo do trabalho;

ATIVIDADE 02 - Fundamentação Teórica. Nesta etapa foi realizada uma pesquisa acerca do conteúdo abordado neste trabalho. Esta pesquisa conteve os temas: Estágio Supervisionado, Gamificação e Tecnologias *Web* para a construção de aplicações ágeis e de fácil manutenção;

ATIVIDADE 03 - Análise e Projeto. Nesta etapa houve o levantamento de requisitos por meio de encontros onde foi discutido com a orientadora e o coorientador o escopo do presente trabalho, também houve a escolha arquitetural do padrão a ser utilizado, ferramentas de desenvolvimento, gerenciamento do projeto, modelagem do banco de dados e a prototipação de telas da versão inicial do sistema;

ATIVIDADE 04 - Elaboração do documento TCC I. Nesta etapa foi apresentado o resultado inicial deste trabalho.

ATIVIDADE 05 - Implementação da *Application Programming Interface* (API) com *Node.js* e *Express*. Nesta etapa foi realizada a codificação do *back-end* da aplicação.

ATIVIDADE 06 - Implementação da interface gráfica abordando a tecnologia de desenvolvimento *Web Next.js*. Nesta etapa foi realizado o desenvolvimento das páginas que constituem a interface gráfica da aplicação e integração com a API.

ATIVIDADE 07 - Documentação final. Nesta última etapa foi realizada a escrita da versão final do documento do Trabalho de Conclusão de Curso.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em cinco seções. A primeira seção apresenta respectivamente: Introdução, Motivação, Problemática, Objetivos Gerais e Específicos e Metodologia. Na seção 2, encontra-se a fundamentação teórica que possui os seguintes temas: Estágio Supervisionado, Gamificação, *React.js*, *Next.js* e Arquitetura de *Software*. Na seção 3, está presente as informações sobre a ferramenta proposta, o diagrama conceitual de classes e a descrição das tecnologias usadas. Por fim, a seção 4 dispõe sobre as Considerações Finais e Trabalhos Futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Para o desenvolvimento da ferramenta proposta se faz necessário o aprendizado de tecnologias que abordem o escopo da aplicação. Logo abaixo serão observados os tópicos fundamentais para a validação teórica deste trabalho, são eles: Estágio Supervisionado, Gamificação, *React.js*, *Next.js* e Arquitetura de *Software*.

2.1 ESTÁGIO SUPERVISIONADO

2.1.1 Definição

O estágio compreende o feito educativo escolar supervisionado, desenvolvido no ambiente de trabalho, que visa o preparo para o trabalho produtivo de educandos que estejam frequentando o ensino regular em instituições de educação superior, de educação profissional, de ensino médio, da educação especial e dos anos finais do ensino fundamental, na modalidade profissional da educação de jovens e adultos (IFPB, 2019).

Segundo Martins (2012), o estágio é caracterizado como um procedimento didático-metodológico, que propicia a aprendizagem social, profissional e cultural a um estudante, visando à preparação para o trabalho produtivo de estudantes que estejam devidamente matriculados em instituições de ensino superior, profissional, ensino médio ou educação especial. Caracteriza-se, desse modo, como um meio de integração entre o aprendizado na escola e aplicação prática no local designado ao estágio.

2.1.2 Objetivo

De acordo com IFPB (2019), o estágio está incluso no projeto pedagógico do curso, além de fazer parte do itinerário formativo do educando e anela assegurar o aprendizado de competências próprias da atividade profissional e à contextualização curricular, pretendendo o desenvolvimento do educando para a vida cidadã e para o trabalho.

2.1.3 Diretrizes

A realização de estágios é regulamentada pela Lei nº 11.788 de 25 de setembro de 2008 e, no IFPB, pelas normas de estágio, aprovadas em 2009. Existem duas modalidades de estágio, que são: o estágio obrigatório e o não obrigatório. Estágio

obrigatório é aquele definido como tal no projeto do curso, cuja carga horária é requisito para aprovação e obtenção de diploma (IFPB, 2019).

Estágio não obrigatório é a modalidade realizada como atividade opcional, acrescida à carga horária regular e obrigatória. Para estagiar, o aluno deverá:

- Estar regularmente matriculado e frequentando regularmente o curso para o qual está se candidatando a estágio;
- No caso de estágio não obrigatório, o aluno não poderá estar no último semestre do curso;
- Não estar exercendo as funções de estagiário em outra Instituição;
- Ter disponibilidade de tempo para atender as atividades programadas e a carga horária exigida;
- Ser aprovado em seleção específica, quando for o caso.

2.1.4 Papel do Orientador

“O saber orientar está associado a uma competência didática do professor orientador, que exige, inclusive, capacidade de relacionamento com o aluno orientando.” (DIAS et al., 2011, p. 701). Dessa maneira, torna-se imprescindível que haja uma boa interação entre o orientador e o seu orientando, pois desse modo, torna-se possível o devido planejamento das atividades realizadas durante o estágio.

Cabe ao professor supervisor do estágio orientar o estudante para que ele desenvolva a maturidade necessária, evitando que fique estressado ou ansioso, pelo fato de ter pouco tempo para a realização do estágio. O grande ensinamento que deve ser deixado para o aluno é que não se pode aprender tudo de uma só vez e em tão pouco tempo, pois esse é um processo que exige tempo investido e perseverança. Assim sendo, o mais importante será analisar cada tarefa que lhe for conferida e procurar realizá-la da melhor forma possível, oferecendo, se viável, sugestões que possam contribuir para o desenvolvimento do estágio e de si próprio (ALBUQUERQUE; SILVA, 2006, p. 9).

2.2 GAMIFICAÇÃO

2.2.1 Definição

Gamificação consiste em inserir diversão, reconhecimento pessoal e/ou competição, em atividades normais de trabalho, para tanto, deve-se aplicar o uso de técnicas de jogo, elaboradas com o objetivo de envolver e motivar os funcionários, ajudando-os a atingir as suas metas (CAMENISCH, 2012 apud FORMANSKI, 2016). Ademais, segundo Kapp (2012), Gamificação é a utilização de mecânicas baseadas em *games*, estética e pensamento *gamer* para engajar as pessoas, motivar ações, promover o aprendizado e a solução de problemas.

2.2.2 Gamificação como ferramenta de apoio ao ensino

A Gamificação manifesta-se como uma oportunidade de agregar diversos mecanismos de captação do interesse dos alunos, de modo a despertar a curiosidade, levando a elementos que expandem a participação e engajamento, resultando na reinvenção do aprendizado (ORLAND et al., 2018). Desse modo, o uso da Gamificação é imprescindível para trazer engajamento durante a realização das atividades que surgirão no percorrer do estágio, incentivando a autoconfiança e a superação, além de fornecer ao estudante *feedback* constante (MURR; FERRARI, 2020).

Assim sendo, o uso da Gamificação neste trabalho, se dá, ao incorporar uma abordagem de resolução de problemas que auxilia o discente a atingir metas e objetivos enquanto exerce o seu papel.

2.2.3 Aplicação da Gamificação no presente trabalho

Tendo em vista os benefícios da Gamificação, foi abordado neste trabalho uma forma de introduzir esta estratégia visando proporcionar engajamento, aumento da produtividade e melhor organização das atividades do estágio. Para tanto, foi desenvolvido um mecanismo que recompensa os discentes ao executarem suas tarefas dentro do prazo, e, caso as atividades do estágio sejam realizadas mais rapidamente do que o estipulado pelo prazo, uma recompensa maior é obtida.

Para avaliar o desempenho ao decorrer das atividades há uma pontuação que atribui ao discente uma categoria através do tipo de medalha, tendo como base as medalhas de bronze, prata e ouro. Cada medalha simboliza o progresso atual em que o discente se encontra no processo de estágio e como tem progredido no cumprimento das atividades. Embora a medalha atual seja uma forma de identificar o progresso do discente, tal progresso evoluirá conforme o desempenho do discente.

Há também recompensas para metas alcançadas. O discente poderá desafiar-se com o intuito de desenvolver novas habilidades ou acumular novos conhecimentos que poderão ser aplicadas durante o estágio supervisionado. Dessa forma, o discente poderá acompanhar sua evolução e enxergar as conquistas alcançadas durante todo o processo do estágio.

As metas são objetivos de aprendizado de novas habilidades ou conhecimentos que serão necessários durante a prática do estágio supervisionado. O discente terá o papel de criar uma meta conforme queira, após o mesmo, identificar que atingiu o desenvolvimento esperado, deverá passar por uma rápida avaliação com o seu orientador, este por fim, retornará uma avaliação positiva quanto ao cumprimento da meta caso esta tenha sido alcançada, sendo este o caso, o orientador a marcará como concluída.

O discente só poderá criar uma meta por semana, caso a meta não seja concluída será apagada e o discente deverá criá-la novamente, neste caso, para alcançar a meta desejada pelo discente, seria necessário mais do que uma semana. Sendo assim, a melhor forma de trabalhar com as metas é enxergando-as como um objetivo maior que pode ser fragmentado em pequenas conquistas. O propósito das metas é ensinar ao discente a criar metas mais objetivas e práticas que tragam resultados satisfatórios. Portanto, as metas são uma forma de incentivar os discentes a desenvolverem novas habilidades e conhecimentos que poderão ser utilizados durante o estágio supervisionado e durante sua carreira profissional.

Todas as pontuações adquiridas durante o período de estágio servirão para realçar a dedicação dos alunos e os resultados de seu empenho. A medalha atual alcançada e as metas conquistadas podem ser exibidas nos perfis dos alunos caso queiram.

2.3 REACT.JS

2.3.1 Definição

"O *React.js* é uma biblioteca *JavaScript* declarativa, eficiente e flexível para criar interfaces com o usuário. Ele permite compor *UIs* complexas a partir de pequenos e isolados códigos chamados 'componentes'"(REACT, 2021). À partir dessa premissa básica do *React.js*¹, é possível ter mais modularidade no código, fazendo com que projetos mais complexos que sigam as boas práticas do *React.js* possibilitem uma fácil manutenção.

¹ <https://pt-br.reactjs.org/>

Inicialmente, foi desenvolvido para organização de anúncios do *Facebook*, onde eles utilizavam uma abordagem tradicional de *Model-View-Controller* do lado do cliente. Aplicativos como esses normalmente consistem em vinculação de dados bidirecional com o modelo de renderização. O *React.js* mudou a maneira como esses aplicativos foram criados, fazendo alguns avanços ousados no desenvolvimento da web (GACKENHEIMER, 2015).

2.3.2 Biblioteca *Open Source*

O *React.js* foi disponibilizado em 2013 como uma ferramenta *JavaScript* na modalidade *Open Source* (Código Aberto). E desde então, sua popularidade entre os desenvolvedores *web* tem crescido, tendo em vista o nível de aprovação observável em seu repositório do *GitHub*². Por isso, dado as diversas possibilidades fornecidas pelo *Framework*, muitas empresas começaram a utilizá-la.

“Originalmente, *Open Source* é um termo que se refere ao *Software Open Source* (OSS). Ele é um código projetado para ser acessado abertamente pelo público: todas as pessoas podem vê-lo, modificá-lo e distribuí-lo conforme suas necessidades” (REDHAT, 2019). Devido a essa iniciativa, existe atualmente inúmeros *Frameworks* e ferramentas *Open Source* disponíveis para auxiliar o desenvolvimento de *Software* ou mesmo de projetos que são disponibilizados para a comunidade e por ela mesma, além disso, há também algumas empresas como *Google*, *Meta*, *Canonical* e outras que possuem projetos *Open Source* hospedados em repositórios públicos no *GitHub*³ ou em outros serviços de repositórios remotos.

2.3.3 Componentes *React.js*

Os componentes no *React.js* tem a finalidade de separar as preocupações do *software*, acoplando as funcionalidades e transformando um ou mais elementos *HiperText Markup Language* (HTML) em um único elemento *React.js*, (FREIRES, 2019). Com os componentes *React.js* é possível refatorar o código, fazendo com que, o mesmo, seja manutenível e escalável, essa facilidade traduz-se na possibilidade do reuso desses componentes em outras partes da aplicação de forma simples.

No *React.js* pode-se optar entre *Functional Components* e *Class Components*. Porém, devido ao desenvolvimento dos *Hooks*, praticamente não se utiliza mais os *Class Components* como antes. Com a chegada dos *Hooks*, o *React.js* passou a usar mais abrangentemente o paradigma funcional dentro de seus componentes. Através dos *Hooks*, têm-se funções que, alteram estado do componente, gerenciam ações

² <https://github.com/facebook/react>

³ <https://github.com/>

que podem ocorrer durante o ciclo de vida do componente, compartilham contextos para toda a aplicação, e entre outras funcionalidades. É mantido também o conceito de imutabilidade, pois, ao invés de atualizar o valor das variáveis de estado em si, lhes são atribuídas um novo estado através da função *setState* do *Hook useState*, por exemplo.

Portanto, por meio dos *Functional Components* é possível criar um ou mais elementos HTML que podem ser replicados em qualquer parte da aplicação. Tais recursos traduzem-se em componentes funcionais, reativos, personalizáveis e reutilizáveis, resultando em uma maior facilidade na construção e integração das interfaces da aplicação.

2.3.4 Desenvolvimento *front-end*

Como já discorrido anteriormente, o *React.js* é utilizado no desenvolvimento do *front-end*, ou seja, a parte da aplicação que compõe a interface do usuário. Segundo Gackenheim (2015), O *React.js* desafia as convenções que se tornaram os padrões de fato para as melhores práticas da estrutura *JavaScript*. O *React.js* faz isso introduzindo muitos novos paradigmas e mudando o status quo do que é necessário para criar aplicativos *JavaScript* e interfaces de usuário escaláveis e sustentáveis. Juntamente com a mudança na mentalidade de desenvolvimento *front-end*, o *React.js* vem com um rico conjunto de recursos que tornam a composição de um aplicativo de página única ou interface de usuário acessível para desenvolvedores de muitos níveis de habilidade - desde aqueles que acabaram de ser apresentados ao *JavaScript*, aos experientes veteranos da *web*.

Com a utilização desse conceito de componentização é possível definir elementos que irão compor a *User Interface* (UI) da aplicação e torna-lós componentes reutilizáveis. Alguns exemplos de componentes, podem ser, por exemplo, uma tela de perfil, um *navbar* ou até mesmo um botão. Qualquer elemento ou conjunto de elementos que possuam propriedades ou que necessite manter um estado que reflita facilmente as novas informações na interface, precisam ter um nível de abstração que faça sentido junto ao conceito de componentes, conceito esse, que pode variar conforme o contexto em que for implementado.

2.4 *NEXT.JS*

2.4.1 Definição

O *Next.js* é uma estrutura *React.js* flexível que fornece blocos de construção para criar aplicativos *Web* rápidos. Essa estrutura refere-se a como o *Next.js* lida

com as ferramentas e configurações necessárias para o *React.js* e fornece estrutura, recursos e otimizações adicionais para seu aplicativo (NEXT, 2022).

O *Next.js* tem se revelado uma ferramenta poderosa tanto para quem é desenvolvedor individual ou para quem faz parte de uma equipe maior, assim sendo, os desenvolvedores conseguem tirar proveito do *React.js* e do *Next.js* para criar aplicativos *Web* totalmente interativos, altamente dinâmicos e de alto desempenho.

2.4.2 Vantagens do *Next.js*

A seguir serão apresentadas algumas vantagens que se destacam no *Framework Next.js*:

- Configuração zero: sendo necessário apenas baixar as dependências e executar um comando de inicialização;
- Híbrido: SSG e SSR: possuindo *features* de renderização de páginas como o *Server Side Rendering* (SSR) e o *Static Site Generation* (SSG) que trabalham com diferentes estratégias de como realizar o carregamento das páginas no navegador em resposta às requisições que chegam no servidor;
- Suporte ao *Typescript*: O *Typescript* tem ajudado muitos desenvolvedores em seus projetos por adicionar tipagem ao *Javascript*. Isso implica que, o código escrito pelos demais integrantes do projeto obrigatoriamente deverá manter a consistência dos tipos de dados que fluem por entre as diferentes partes do sistema, pois os eventuais erros de tipagem serão indicados em tempo de compilação, evitando assim, erros em produção que são mais difíceis de detectar com o *Javascript*. O *Typescript* facilita também no *Intellisense* do código, onde é possível ter a visualização dos atributos de um objeto, de um *type* ou de uma *interface* por exemplo.
- *API Routes*: Proporciona uma API própria onde é possível criar facilmente um *endpoint* de API como uma função, de forma que não seja necessário utilizar o servidor do *Node.js*.

2.4.3 Por que usar o *Next.js*?

A forma de trabalhar com o *Next.js* é muito similar ao *React.js*, por essa razão, a curva de aprendizado é menor. Essa escolha também se dá pelo fato de que através do *Next.js* é possível criar páginas estáticas, facilitando a indexação da aplicação para os motores de busca dos navegadores, problema que acontece com o *React.js*.

O *Next.js* soluciona este problema mediante estratégias como o SSG, onde o servidor do *Next.js* consegue devolver pro *Browser*, em resposta a requisições *Hypertext Transfer Protocol* (HTTP), todo o corpo HTML do site, dispensando também o processamento realizado normalmente pelo lado do cliente através do motor *JavaScript* presente nos navegadores.

Dessa forma, o servidor torna-se o encarregado de gerar essas páginas estáticas. Uma vez que essas páginas foram geradas, elas permanecem na *build* do projeto como uma espécie de *cache*, assim, quando uma nova requisição HTTP ocorre, o servidor já possui a página em questão armazenada em *cache*, proporcionando maior fluidez na navegação do site e maior desempenho para a aplicação *web*.

2.5 ARQUITETURA DE SOFTWARE

“A arquitetura de *software* pode ser definida como uma representação da organização de um sistema” (GARLAN; SHAW, 1994 apud ALVES, 2016). Para Martin (2017, p. 196), a arquitetura de um sistema de *software* é a forma dada a esse sistema pelos seus criadores. Essa forma está na repartição desse sistema em componentes, na organização desses componentes e nos modos como esses componentes se comunicam entre si.

Sendo assim, para produzir um *software* que cumpra bem o seu papel e seja flexível para mudanças, é necessário gastar tempo planejando como ele será construído, uma vez que, conforme o *software* cresce, aumenta-se a complexidade do mesmo. Por isso, escolher uma arquitetura que solucione adequadamente o problema em questão se faz necessário para garantir que o sistema permaneça escalável e manutenível.

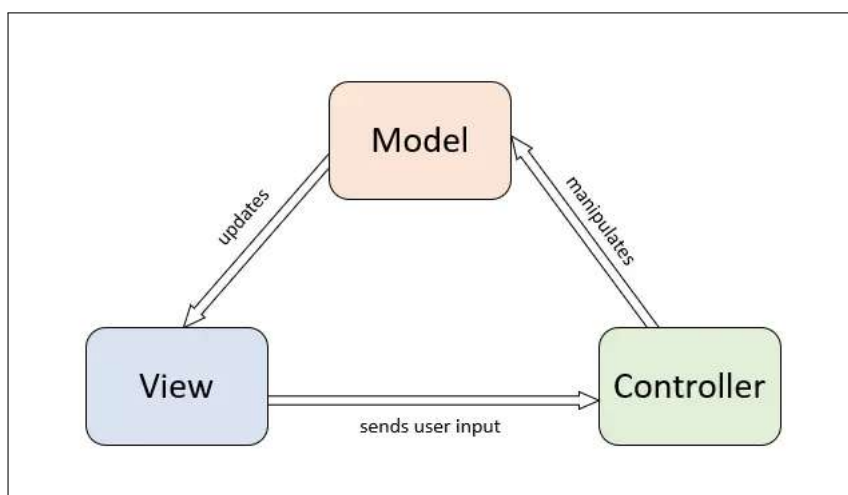
2.5.1 *Model-View-Controller*

O padrão *Model-View-Controller* (MVC) separa as camadas do projeto em três, a camada de apresentação, de lógica de negócio e de gerenciamento do fluxo da aplicação. Dessa forma, aumenta-se a capacidade de reutilização e de manutenção do projeto. A organização em camadas é o método mais adequado para atingir a independência entre os componentes. É através desta independência que pode-se alcançar eficiência, escalabilidade, reutilização e manutenibilidade (LUCIANO; ALVES, 2011).

O MVC se trata de dividir o código do *software* em segmentos funcionais para tornar a aplicação apta a executar mudanças de forma rápida, realizar testes unitários e realizar ampliações sem afetar o código já pronto, mantendo o limite entre três camadas,

a *Model*, a *View* e o *Controller* (OLIVEIRA, 2013). Deste modo, firma-se uma estratégia que consiste na divisão das preocupações do *software*, tornando o mesmo, eficiente quanto a execução das diversas tarefas.

Figura 1 – Modelo da arquitetura MVC



Fonte: (VERLAG, 2022)

A Figura 1 apresenta o padrão MVC, nela estão representadas as camadas de separação física e lógica, proposta pelo modelo. Na camada *View*, é apresentada todas as interfaces que interagem com o usuário, são elas que permitem a realização de requisições através da entrada de dados. Na camada *Controller*, se concentra toda a parte que controla o comportamento da aplicação, servindo como uma intermediária entre a *View* e o *Model*. Por fim, a camada *Model* é se responsabiliza por conter o código da ação que foi requisitada como, por exemplo, se comunicar com outra camada que realize a busca em um banco de dados (BUSCHMANN et al., 1996).

2.5.2 *Clean Architecture*

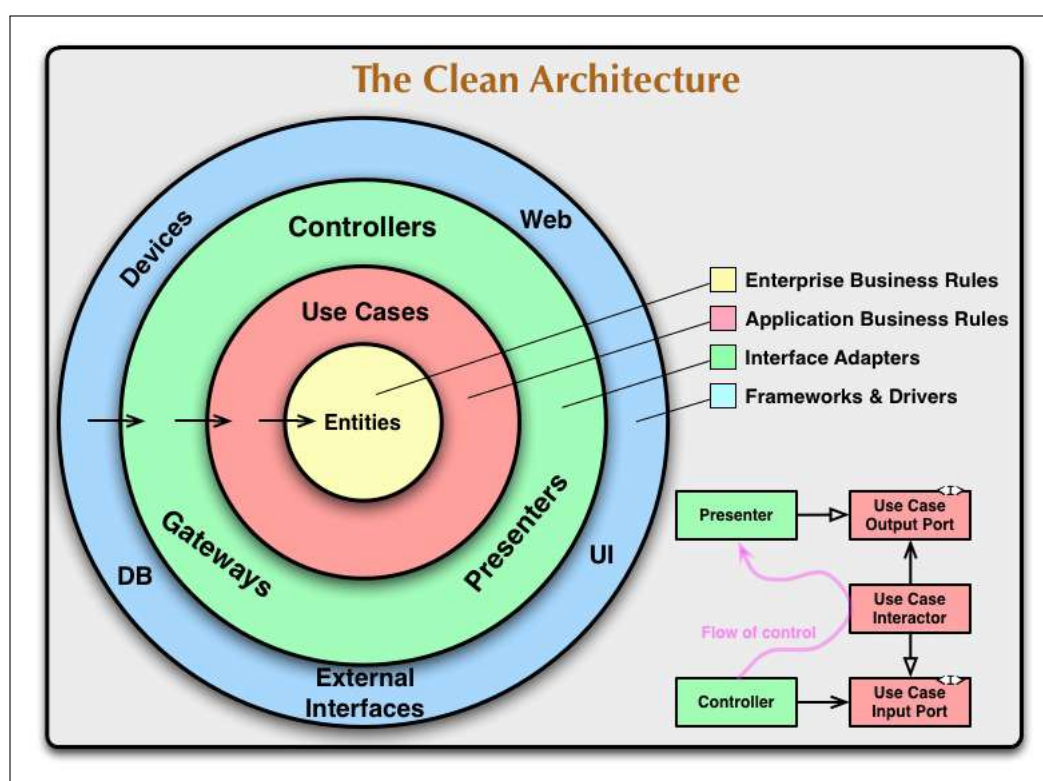
Uma boa arquitetura faz com que o sistema seja fácil de entender, desenvolver, manter e implantar. O objetivo final, entretanto, é minimizar o custo da vida útil do sistema e maximizar a produtividade do programador (MARTIN, 2017). A *Clean Architecture*, a qual é composta pelos seguintes componentes: *controllers*, *use cases*, *services*, *adapters*, *repositories* e *entities*.

Os *controllers* são funções que recebem os dados provenientes das rotas da API. Caso haja necessidade, é implementada também uma lógica de adaptação dos

dados antes de enviá-los para o seu *service* correspondente, nesses casos são criados *adapters*, que se responsabilizam por esse processo. Após isso, o *service* utiliza-se dos *use cases* para efetuar uma validação segundo a lógica de negócio adotada, em seguida, os dados são repassados para o seu respectivo *repository*, onde os mesmos, serão persistidos no banco de dados através das abstrações representadas pelas *entities*.

Na Figura 2 há um modelo que ilustra as diferentes camadas da *Clean Architecture* e suas respectivas atribuições.

Figura 2 – Modelo da arquitetura *Clean Architecture*



Fonte: (MARTIN, 2017)

Conforme a Figura 2 percebe-se que há 4 camadas, são elas: regras de negócio da empresa, regras de negócio da aplicação, interfaces e adaptadores de interface e *Frameworks* e *Drivers*. Através desta organização de camadas, a arquitetura visa produzir sistemas com as seguintes características: Independência de *Frameworks*, Testabilidade, Independência da UI, Independência do banco de dados e Independência de qualquer agência externa (MARTIN, 2017).

Posto isso, nota-se que uma boa arquitetura faz com que haja a desacoplação dos componentes, visando dirimir quaisquer dependências que possam dificultar a flexibilidade do código quanto a futuras mudanças de tecnologias e de estratégias ligadas a lógica do negócio.

3 PLATAFORMA PHASE

Phase é uma plataforma pensada para o contexto *web* e foi idealizada para auxiliar os alunos no processo de estágio, onde o seu acesso, se dá com qualquer dispositivo que possua um navegador. Outrossim, contendo a finalidade de auxiliar os discentes a ingressarem no processo de estágio e a corroborar com os professores, a ferramenta visa alcançar, de forma objetiva, viabilizar o exercício de orientação de uma forma prática e eficiente para os alunos. Para tanto, o sistema utiliza-se do conceito de gamificação, que será incorporado dentro dos mecanismos de estudo dos materiais de apoio e do cumprimento das metas estabelecidas no cronograma de atividades.

Usando-se necessariamente de um dispositivo que possua acesso à *internet*, a plataforma oferece funcionalidades como busca de vaga por filtros, organização e anexação de materiais de apoio por parte dos orientadores e a postagem de vagas pelas empresas. Essas e outras funções, constituirão a ferramenta proposta, para que a mesma, venha conferir apoio a alunos e professores de forma satisfatória.

Destinando-se a discentes de instituições de ensino que integrem a modalidade de estágio supervisionado, a ferramenta Phase busca proporcionar um ambiente que contribua com as etapas do processo de estágio, tornando o mesmo, eficiente e satisfatório para os discentes e orientadores.

Nos tópicos abaixo, serão discutidos os métodos utilizados para a realização da coleta de dados, onde os mesmos serão descritos e expostos, em seguida, será apresentada a arquitetura utilizada para o desenvolvimento do sistema e as funcionalidades já implementadas.

3.1 LEVANTAMENTO DE REQUISITOS

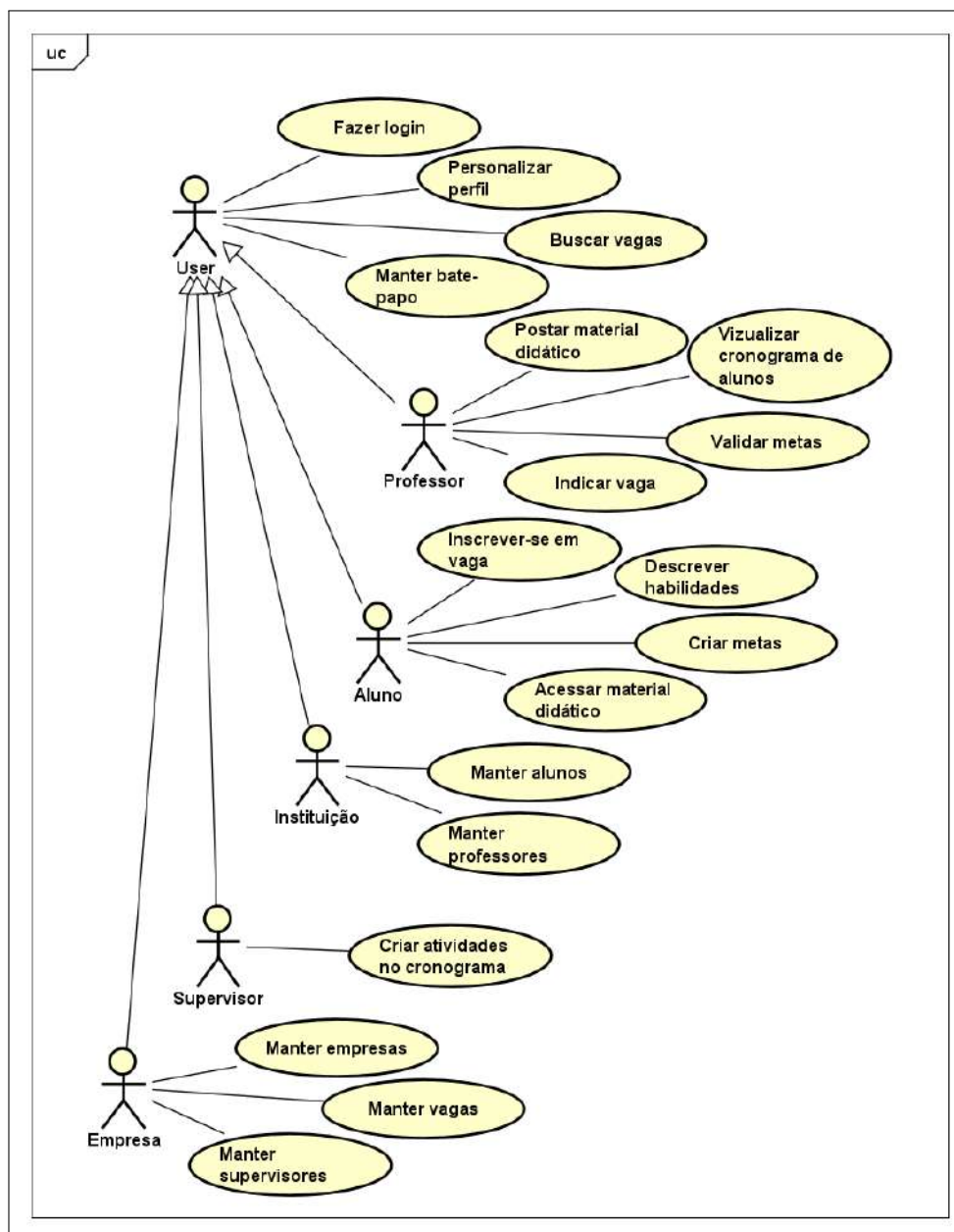
O levantamento de requisitos é uma etapa primordial que definirá quais funcionalidades atenderão as necessidades sanadas pelo sistema, para tanto, tais requisitos foram estabelecidos juntamente com dois professores do IFPB - Campus Cajazeiras.

3.2 ANÁLISE DOS REQUISITOS

Ao fim do levantamento de requisitos, houve o reconhecimento de algumas funcionalidades que serão importantes para o cumprimento do objetivo da ferramenta

proposta. A Figura 3 destaca os casos de uso levantados que representam os requisitos funcionais apresentados no Quadro 1.

Figura 3 – Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

Na Figura 3 estão ilustrados, por meio de um diagrama de casos de uso, os casos de usos que envolvem as ações do ator Usuário e de suas especializações, sendo elas: Professor, Aluno, Instituição e Empresa.

Quadro 1 – Requisitos funcionais da ferramenta Phase

ID	REQUISITO	DESCRIÇÃO
RF01	Login	O sistema deverá permitir que os usuários sejam autenticados
RF02	Manter aluno	O sistema deverá cadastrar, atualizar e remover as informações básicas dos alunos
RF03	Manter professor	O sistema deverá cadastrar, atualizar e remover as informações básicas dos professores
RF04	Manter instituição	O sistema deverá cadastrar, atualizar e remover as informações básicas das instituições
RF05	Manter empresa	O sistema deverá cadastrar, atualizar e remover as informações básicas das empresas
RF06	Manter supervisor	O sistema deverá cadastrar, atualizar e remover as informações básicas dos supervisores
RF07	Manter vagas	O sistema deverá cadastrar, atualizar e remover as informações básicas das vagas
RF08	Manter habilidades	O sistema deverá cadastrar, atualizar e remover as habilidades dos alunos
RF09	Personalizar perfil do aluno	O sistema deverá permitir a edição do perfil dos alunos que deverão expor suas habilidades técnicas, comportamentais e experiências adquiridas
RF10	Postagem de material de apoio	O sistema deverá permitir aos professores a postagem de materiais de apoio para os seus alunos
RF11	Implementar <i>Feed</i> de vagas	O sistema deverá possuir um <i>feed</i> onde serão listadas as vagas das empresas
RF12	Implementar bate-papo	O sistema deverá possuir um bate-papo para que haja possibilidade de comunicação entre os usuários
RF13	Implementar Quadro de cronograma	O sistema deverá possuir um quadro de cronograma para cada aluno
RF14	Manter atividades do Cronograma	O sistema deverá criar, atualizar e remover as atividades do cronograma
RF15	Manter metas	O sistema deverá criar, validar o <i>status</i> e excluir as metas

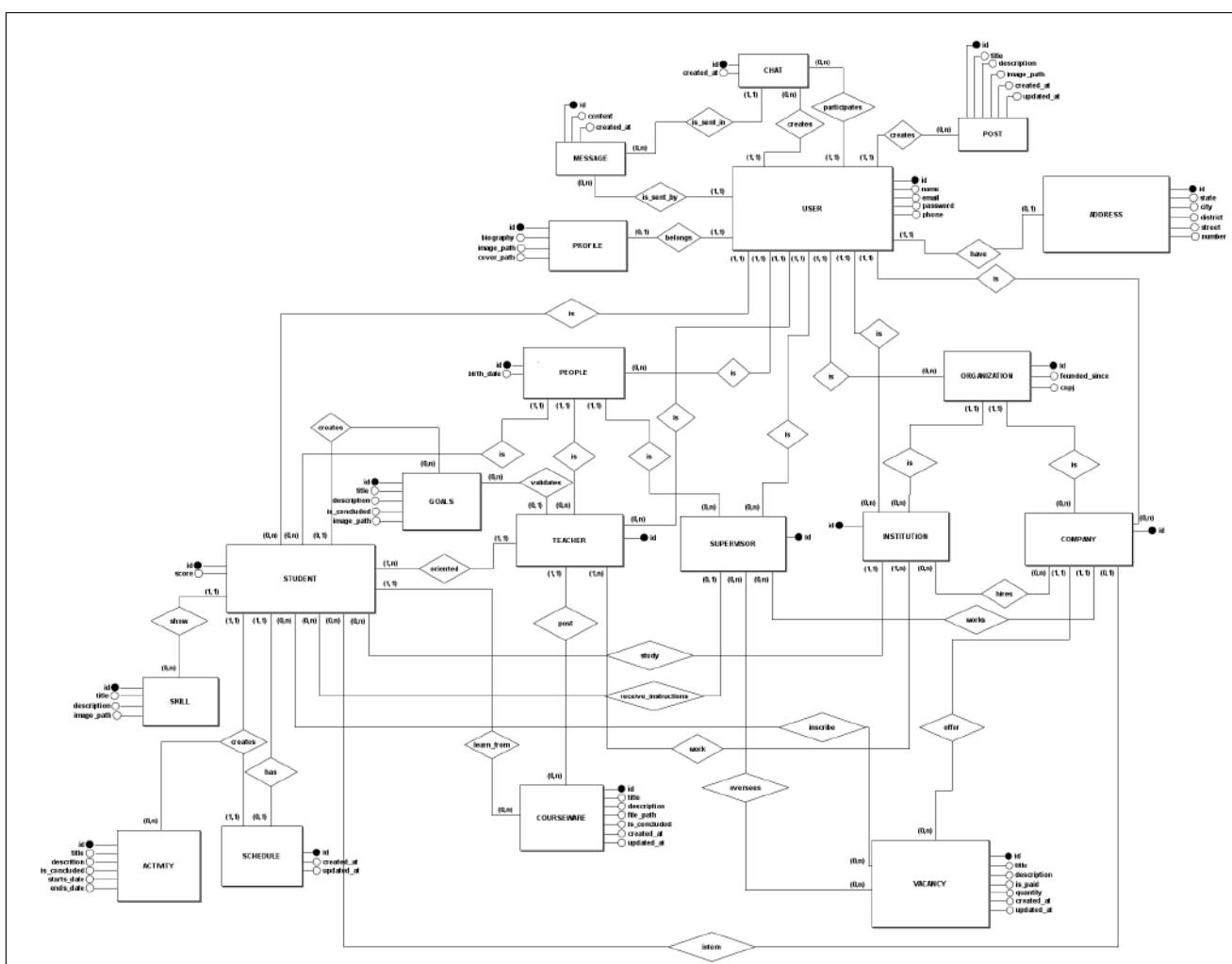
Fonte: Elaborado pelo autor

No Quadro 1 estão elencados os requisitos funcionais da ferramenta proposta. Sendo eles os requisitos esperados para a ferramenta, entretanto, pode haver o acréscimo de mais requisitos, caso haja a necessidade e seja viável.

3.3 MODELAGEM CONCEITUAL

Na Figura 4 é ilustrada a modelagem dos dados por meio da representação das entidades e de seus relacionamentos.

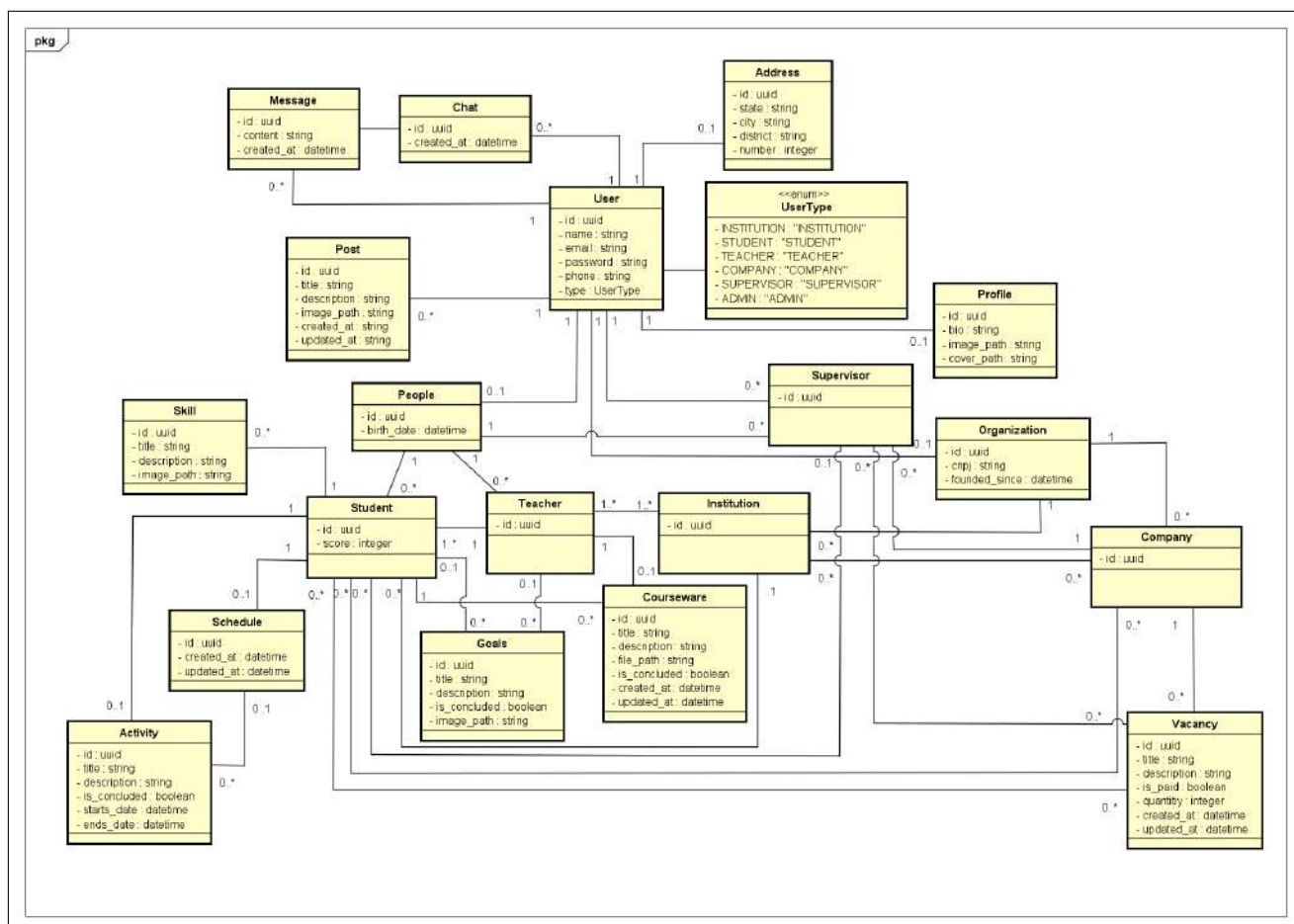
Figura 4 – Modelo Entidade Relacionamento



Fonte: Elaborado pelo autor

Na Figura 5 é apresentada a modelagem da perspectiva de um diagrama de classes, onde estão presentes: as classes, seus atributos e seus relacionamentos.

Figura 5 – Diagrama de Classes



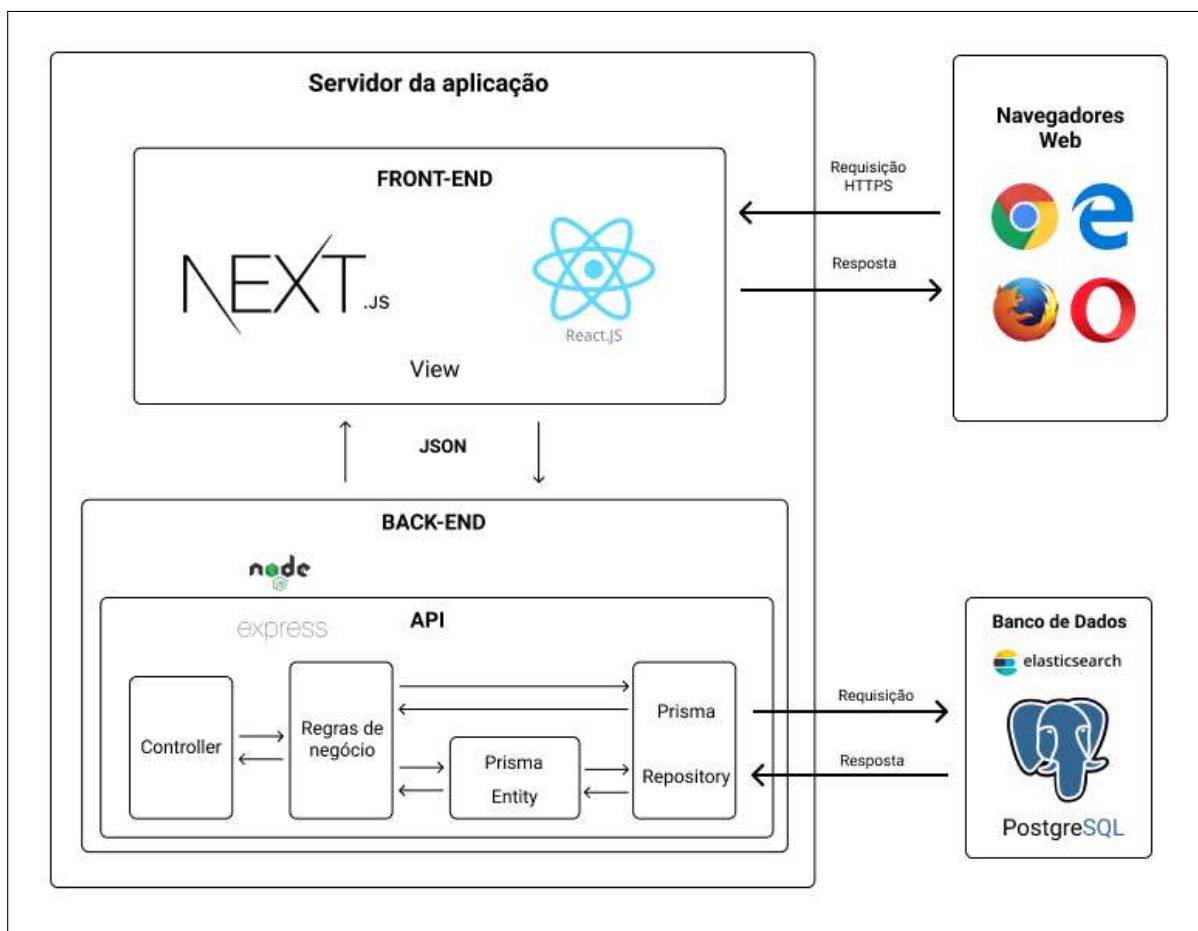
Fonte: Elaborado pelo autor

A modelagem dos dados está melhor detalhada no Apêndice A, onde estão presentes: a descrição do mini-mundo, o modelo lógico e o modelo físico.

3.4 ARQUITETURA DA APLICAÇÃO

A ferramenta proposta Phase, segue em sua arquitetura o padrão MVC em conjunto com a *Clean Architecture*, tendo em vista que as duas oferecem excelentes formas de se estruturar um projeto e considerando que elas conseguem atender a qualidade de *software*.

Figura 6 – Arquitetura da Aplicação



Fonte: Elaborado pelo autor

Ao examinar a Figura 6 acima, é possível perceber os diferentes elementos que fazem parte da arquitetura da aplicação. A mesma, é dividida entre *back-end* e *front-end*. Para o *back-end*, será utilizado o *Node.js*¹ que permite executar código *JavaScript* fora do navegador. Ainda nesse escopo, tem-se a *Application Programming Interface* (API), que será fornecida através da biblioteca *Express*², através dela, requisições HTTP ocorrerão utilizando-se de uma estrutura *JavaScript Object Notation* (JSON) para compor o corpo dos dados. Dessa forma, o *back-end* tem a função de estabelecer uma comunicação com o banco de dados, visando fornecer os dados necessários para o usuário final através da API que se utiliza de requisições HTTP contendo interface gráfica. Durante a construção da API, foi adotada a *Clean Architecture* visando a criação

¹ <https://nodejs.org/pt-br/>

² <https://expressjs.com/pt-br/>

de um sistema com segmentos bem definidos e de fácil manutenibilidade, realização de teste e modificação de tecnologias. O *back-end* será hospedado na plataforma de *deploy Heroku*³.

Para realizar a conexão entre o *back-end* e o *PostgreSQL*⁴, visando a permanência dos dados da aplicação, será utilizado o *Object-Relational Mapping (ORM) Prisma*⁵, que detém de diversas funcionalidades capazes de oferecer maior facilidade na manipulação e controle do banco de dados. Visando obter maior velocidade na busca de vagas, optou-se por utilizar a ferramenta de busca *Elasticsearch*⁶, dado que, a ferramenta é extremamente ágil. Por fim, o *front-end*, que representa a interface gráfica, será construído pelo *Framework Next.js* e será hospedado na plataforma de *deploy Vercel*⁷, criadora do *Next.js*. Será utilizado no *front-end* o padrão MVC, tendo em vista que ele favorece a separação das preocupações do código, tonando-o escalável e reutilizável.

3.5 TELAS DOS PRINCIPAIS REQUISITOS

A prototipação das telas foi realizada através do *software Figma*⁸, o qual possibilitou a implementação da UI de um modo fácil e prático. Logo abaixo serão apresentadas as telas prototipadas.

³ <https://www.heroku.com/>

⁴ <https://www.postgresql.org/>

⁵ <https://www.prisma.io/>

⁶ <https://www.elastic.co/pt/>

⁷ <https://vercel.com/>

⁸ <https://www.figma.com/>

Figura 7 – Tela de login de usuários

A imagem mostra a interface de login do sistema Phase. No topo, há uma barra azul com o nome 'Phase' e ícones de perfil, notificação e menu. O conteúdo principal é dividido em duas seções. À esquerda, uma ilustração mostra uma pessoa sentada em uma cadeira, trabalhando em um computador em uma mesa com livros e uma planta. Acima da ilustração, o texto diz: 'Olá, seja bem-vindo(a) ao Phase!' e 'Feito para você que busca obter maior aproveitamento, capacitação e aprendizado durante o processo de estágio junto ao seu orientador.' À direita, há uma seção de login com o título 'Login'. Abaixo dele, há dois campos de entrada: 'Email' e 'Senha'. Um botão azul 'Entrar' está posicionado abaixo dos campos. Abaixo do botão, há um link: 'É novo por aqui? Clique aqui e registre-se já!'.

Fonte: Elaborado pelo autor

RF01: A Figura 7 apresenta a tela responsável por realizar o login de usuários. Para realizar o login na plataforma se faz necessário ter efetuado um cadastro prévio, contudo, para realizar essa ação, o usuário deve preencher o formulário com seu E-mail e senha e então clicar no botão "Entrar".

Figura 8 – Tela de cadastro para alunos e professores

A imagem mostra a interface de cadastro do sistema Phase. No topo, há uma barra azul com o nome 'Phase' e ícones de perfil, notificação e menu. O conteúdo principal é dividido em duas seções. À esquerda, uma ilustração mostra uma pessoa sentada em uma cadeira, trabalhando em um computador em uma mesa com livros e uma planta. Acima da ilustração, o texto diz: 'Preencha os dados abaixo'. À direita, há um formulário de cadastro com vários campos de entrada: 'Nome', 'Email', 'Senha', 'Telefone', 'Rua', 'Número', 'Bairro', 'Cidade', 'Estado', 'Data de nascimento' (com máscara dd/mm/aaaa). Um botão azul 'Cadastrar' está posicionado abaixo dos campos.

Fonte: Elaborado pelo autor

RF02, RF03 e RF06: A Figura 8 apresenta a tela de cadastro de alunos, professores e supervisores. Para realizar o cadastro dessas entidades na plataforma se faz necessário que o usuário preencha o formulário de cadastro com seus dados correspondentes. Dentre esses dados, apenas Nome, E-mail, Senha e Telefone são obrigatórios, sendo assim, os demais dados são opcionais.

Figura 9 – Tela de cadastro para instituições e empresas

A imagem mostra a interface de usuário para o cadastro de instituições e empresas. No topo, há uma barra azul com o nome 'Phase' e ícones de perfil e navegação. O formulário contém o seguinte texto: 'Preencha os dados abaixo'. Os campos de entrada são:

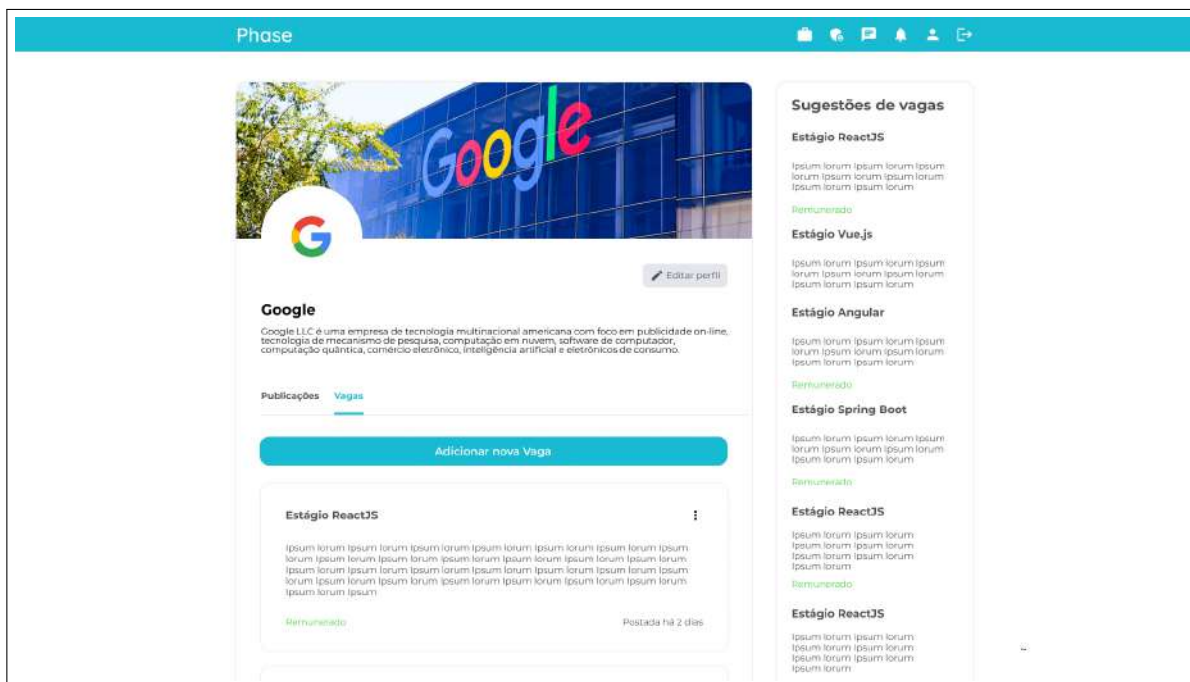
Nome	Rua	Cidade
Email	Número	Estado
Senha	Bairro	Data de fundação dd/mm/aaaa
Telefone	CNPJ	

Um botão azul 'Cadastrar' está localizado na base do formulário.

Fonte: Elaborado pelo autor

RF04 e RF05: A Figura 9 apresenta a tela de cadastro de instituições e empresas. Para a realização do cadastro dessas entidades na plataforma se faz necessário que o usuário preencha o formulário de cadastro com seus dados correspondentes. Neste caso, todos os dados presentes no formulário são obrigatórios.

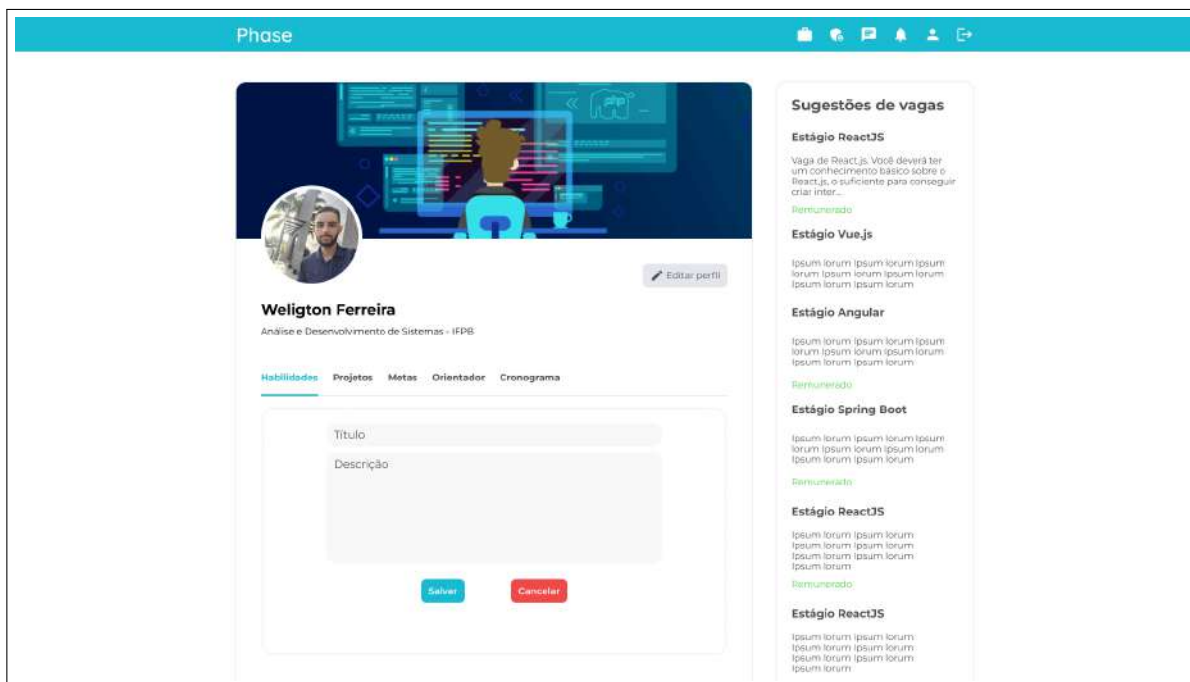
Figura 10 – Tela de cadastro vagas



Fonte: Elaborado pelo autor

RF07: A Figura 10 apresenta a tela de cadastro de vagas. Para a realização do cadastro das vagas é necessário informar o título, a descrição, uma imagem caso seja desejável, se a vaga é remunerada e a quantidade de vagas ofertadas.

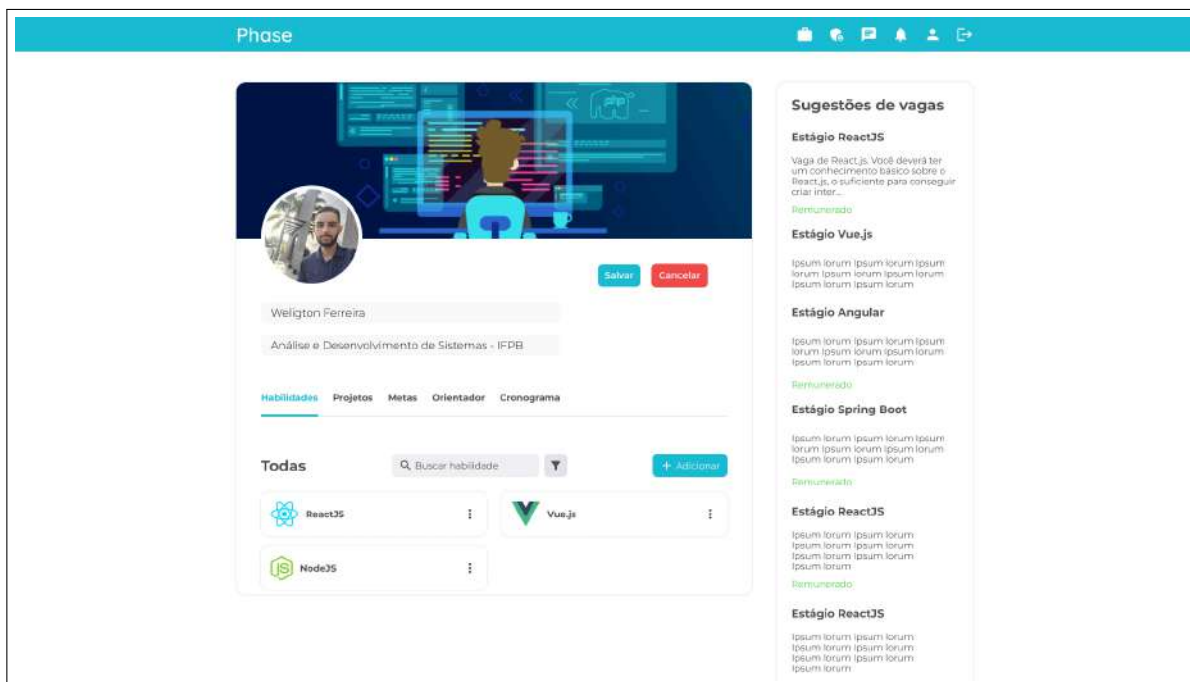
Figura 11 – Tela de cadastro de habilidades do aluno



Fonte: Elaborado pelo autor

RF08: A Figura 11 apresenta a tela de cadastro de habilidades do aluno. Para a realização do cadastro das habilidades é necessário informar o título e a descrição.

Figura 12 – Tela de edição do perfil do aluno



Fonte: Elaborado pelo autor

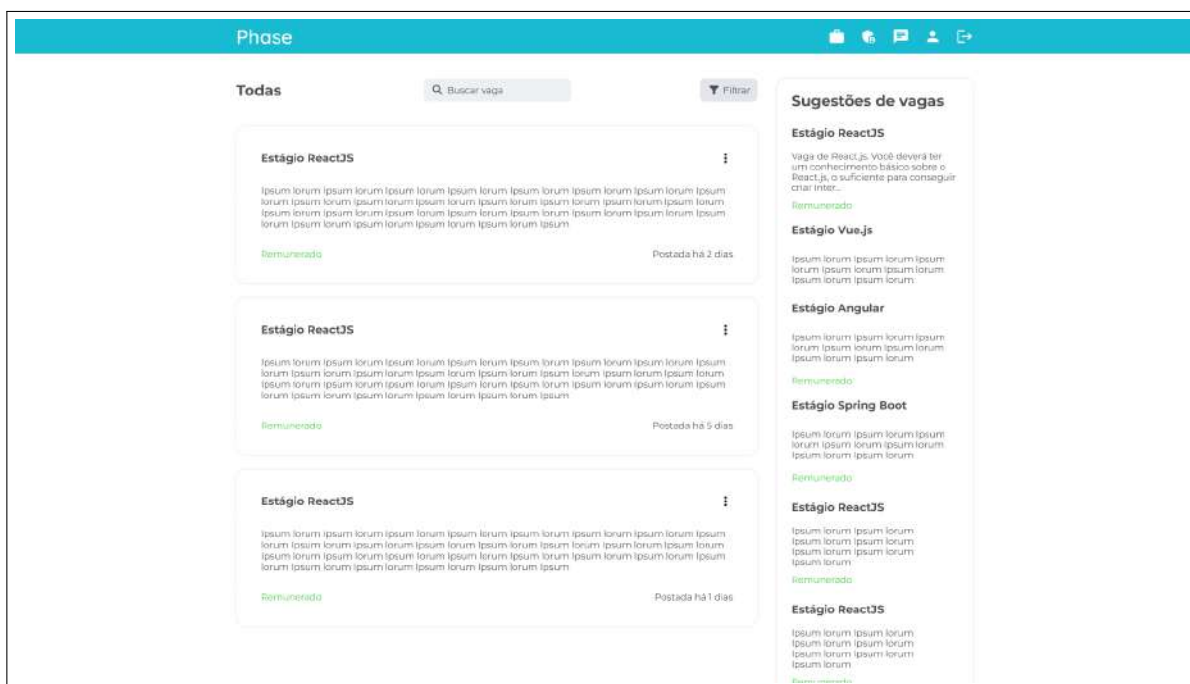
RF09: A Figura 12 apresenta a tela de edição do perfil do aluno. Para realizar esta operação o aluno deverá alterar as informações que deseja e então as informações serão atualizadas.

Figura 13 – Tela de postagem do material de apoio

The screenshot displays the 'Phase' app interface. At the top, a teal header contains the word 'Phase' and several navigation icons. Below the header, there is a large landscape image of a park. Underneath the image is a circular profile picture of a woman, followed by the name 'Eva Campos' and her title 'Professora Doutora na Instituição IFPB, Campus Cajazeiras'. To the right of the profile picture is an 'Editar perfil' button. Below the profile information, there are two tabs: 'Material de Apoio' (selected) and 'Orientandos'. The 'Material de Apoio' tab contains a form with the following elements: a 'Título' input field, a larger 'Descrição' text area, an 'Anexar arquivo' section with icons for 'Foto', 'Video', and 'Arquivo', and two buttons at the bottom: 'Salvar' (blue) and 'Cancelar' (red). To the right of the form, there is a 'Sugestões de vagas' section listing several job opportunities, each with a title, a short description, and a 'Remunerado' status.

Fonte: Elaborado pelo autor

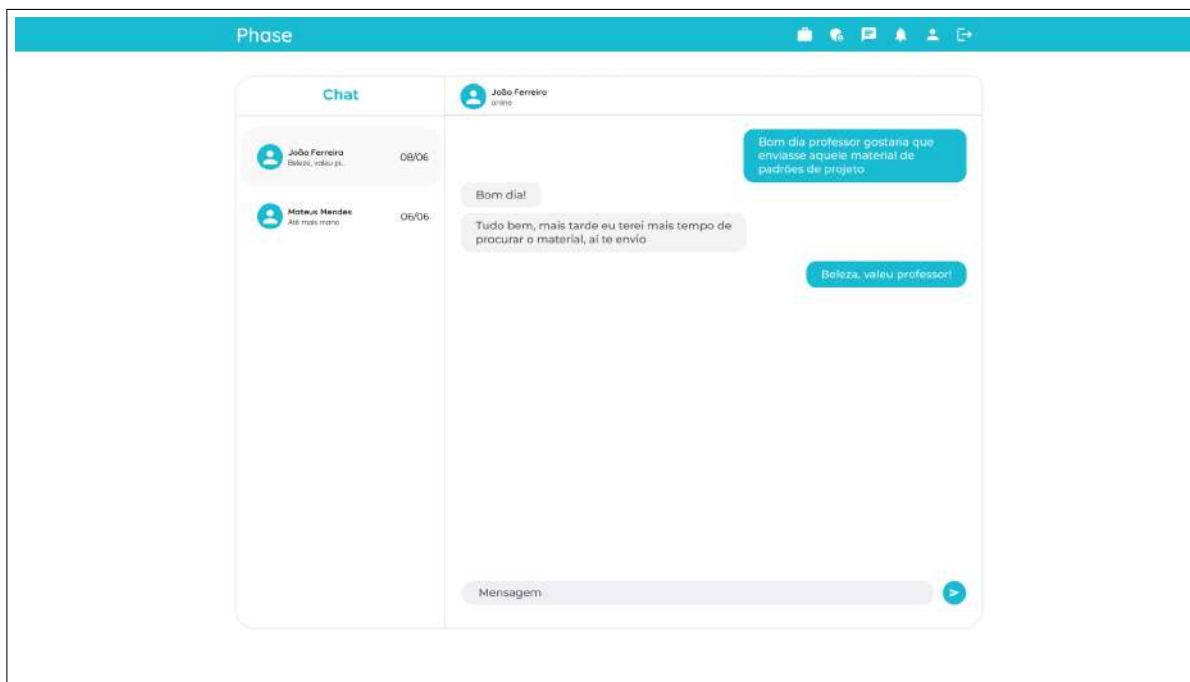
RF10: A Figura 13 apresenta a tela de postagem do material de apoio por parte do professor. Para realizar esta operação o professor deverá adicionar o título, a descrição e o arquivo ou link do mesmo.

Figura 14 – Tela do *feed* de vagas

Fonte: Elaborado pelo autor

RF11: A Figura 14 apresenta a tela do *feed* de vagas. Esta é a tela padrão da aplicação, é onde as vagas de estágio podem ser visualizadas pelos alunos e onde podem fazer buscas por vagas específicas.

Figura 15 – Tela de bate-papo



Fonte: Elaborado pelo autor

RF12: A Figura 15 apresenta a tela de bate-papo. Esta tela apresenta a interface de bate-bate entre os usuários. As demais telas prototipadas podem ser encontradas no Apêndice B.

4 CONSIDERAÇÕES FINAIS

O presente trabalho abordou o uso da gamificação para engajamento dos alunos durante o processo de estágio supervisionado. A ferramenta dispõe de funcionalidades que possibilitam a comunicação entre orientandos e orientadores para que haja um ambiente integrado que facilite a realização das atividades do estágio, bem como, as suas demandas. Além disso, a ferramenta poderá ser o ponto de partida que facilitará o ingresso dos discentes no estágio supervisionado.

Desta forma, este trabalho poderá contribuir significativamente para o processo de estágio supervisionado através desta aplicação *web* que traz um ambiente que integra busca de vagas, comunicação entre orientandos e orientadores e uma maneira prática de manter a organização das atividades e metas de aprendizado para a realização das atividades do estágio supervisionado.

4.1 TRABALHOS FUTUROS

Como sugestão de trabalhos futuros, podem ser desenvolvidos alguns requisitos que não foram implementados nesta versão do sistema, bem como a criação de uma aplicação móvel que poderá proporcionar mais praticidade no dia-a-dia.

REFERÊNCIAS

- ALBUQUERQUE, L. S.; SILVA, E. M. da. **Pontos Positivos E Negativos Do Estágio Na Formação Profissional Dos Estudantes De Ciências Contábeis Da Cidade De Caruaru-Pe**. Salvador, BA: AnPAD, 2006. Disponível em: <<http://docplayer.com.br/9789826-Pontos-positivos-e-negativos-do-estagio-na-formacao-profissional-dos-estudantes-de-ciencias-contabeis-da-cidade-de-caruaru-pe.html>>.
- ALVES, G. O. **Recuperação de Arquitetura de Software com a Ferramenta ArRTool**. Porto Alegre, RS: Universidade Federal do Rio Grande do Sul, 2016. Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/151024/001009751.pdf>>.
- BUSCHMANN, F.; MEUNIER, H. R. R.; SOMMERLAD, P.; STAL, M. **Pattern-Oriented Software Architecture: A System of Patterns**. 1. ed. Chichester, UK: John Wiley Sons, 1996.
- CAMENISCH, J. **Make It a Game: Using Gamification to Build Your Business**. 2012. Disponível em: <<https://www.upwork.com/blog/2012/01/make-it-a-game-using-gamification-to-build-your-business/>>.
- DIAS, S. M. R. C.; PATRUS, R.; AES, Y. T. D. M. **Quem ensina um professor a ser orientador? Proposta de um modelo de orientação de monografias, dissertações e teses**. Rio de Janeiro: Cortez, 2011. v. 12.
- FORMANSKI, F. N. **Gamification 101: an introduction to the use of game dynamics to influence behavior**. 2016. Disponível em: <<http://btd.egc.ufsc.br/wp-content/uploads/2016/03/Francieli-Naspolini-Formanski.pdf>>.
- GACKENHEIMER, C. **Introduction to React**. 2015. Disponível em: <<https://pepa.holla.cz/wp-content/uploads/2016/12/Introduction-to-React.pdf>>.
- GARLAN, D.; SHAW, M. **An Introduction to Software Architecture**. Pittsburgh, PA, USA: Pearson, 1994.
- IFPB. **Estágio**. 2019. Disponível em: <<https://www.ifpb.edu.br/cajazeiras/assuntos/estagio/estagio>>.
- KAPP, K. M. **The gamification of learning and instruction**. 1. ed. San Francisco: Pfeiffer, 2012.
- LUCIANO, J.; ALVES, W. J. B. **PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER**. Centro Universitário UNIFAFIBE, Bebedouro, SP: Revista EPeQ Fafibe, 2011. Disponível em: <<https://unifafibe.com.br/revistasonline/arquivos/revistaeppeqfafibe/sumario/20/16112011142249.pdf>>.
- MARTIN, R. C. **Clean Architecture: A Craftsman's Guide to Software Structure and Design**. 1. ed. Londres: Pearson, 2017.
- MARTINS, S. P. **Estágio e Relação de Emprego**. 3. ed. São Paulo: Atlas, 2012.

MURR, C. E.; FERRARI, G. Entendendo e aplicando a gamificação: o que é, para que serve, potencialidades e desafios. Florianópolis, SC, p. 36, 2020.

NEXT. **O que é o Next.js?** 2022. Disponível em: <<https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>>.

OLIVEIRA, L. F. de. Entendendo a tríade model-view-controller (mvc) utilizando padrões de projeto de software orientado a objetos. Taubaté, SP, v. 1, 2013.

ORLAND, T. R. C.; DUQUE, C. G.; MORI, A. M. **Gamificação: uma Nova Abordagem Multimodal para a Educação**. Brasília, DF: Universidade de Brasília – UNB, 2018. Disponível em: <<http://www.scielo.org.pe/pdf/biblios/n70/a02n70.pdf>>.

POLZIN, F. R. **O ESTÁGIO OBRIGATÓRIO COMO INSTRUMENTO DE INSERÇÃO NO MERCADO DE TRABALHO**. Dissertação (Mestrado) — UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, Curitiba, PR, 2019.

REACT. **Tutorial: Introdução ao React**. 2021. Disponível em: <<https://pt-pt.reactjs.org/tutorial/tutorial.html>>.

REDHAT. **O que é Open Source?** 2019. Disponível em: <<https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>>.

VERLAG, B. **Model View Controller**. 2022. Disponível em: <<https://bmu-verlag.de/model-view-controller-mvc-softwareentwicklungsmuster/>>.

APÊNDICE A – ARQUITETURA DO BANCO DE DADOS DO SISTEMA

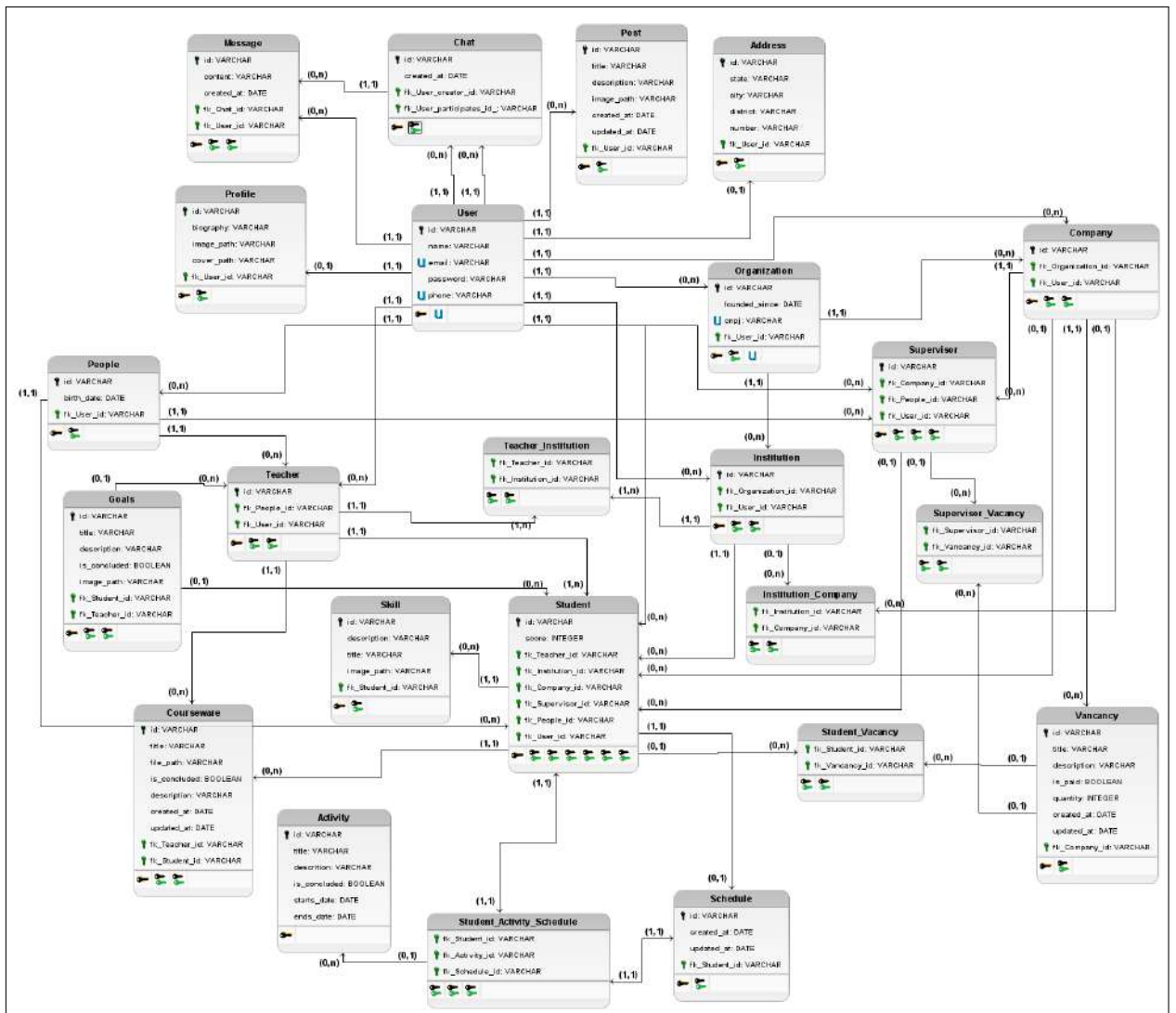
Descrição do Mini-Mundo

A base de dados irá armazenar as informações da plataforma Phase, onde é necessário armazenar as informações dos alunos, tais como: nome, e-mail, telefone e demais sendo opcionais. Também é preciso armazenar os seguintes dados dos professores: nome, e-mail, telefone e demais sendo opcionais. Para cada conteúdo postado na plataforma pelos professores é necessário guardar as seguintes informações: título, descrição, professor que postou o conteúdo, se a atividade já foi concluída. Para as atividades no cronograma, entretanto, é necessário guardar as seguintes informações: título, descrição, quando começa a atividade e quando termina, dessa forma, as atividades são relacionadas ao cronograma. As entidades restantes estão descritas no modelo lógico do banco de dados abaixo.

Modelo Lógico

A representação lógica do banco de dados irá compor o armazenamento dos dados da aplicação está representada na Figura 16 e consiste na representação de todas as tabelas a serem criadas na base de dados e seus atributos que serão gerados a partir de relacionamentos entre entidades.

Figura 16 – Modelo lógico



Fonte: Elaborado pelo autor

Modelo Físico

Esta seção apresenta o modelo físico da base de dados gerado a partir do modelo lógico presente na seção anterior. O código gerado segue os comandos e regras utilizados pelo PostgreSQL, dado que é a ferramenta utilizada na aplicação para a permanência dos dados.

CREATE TABLE User (


```

        id VARCHAR PRIMARY KEY,
        name VARCHAR,
        email VARCHAR,
        password VARCHAR,
        phone VARCHAR,
        UNIQUE (email, phone)
    );

CREATE TABLE Student (
    id VARCHAR PRIMARY KEY,
    score INTEGER,
    fk_Teacher_id VARCHAR,
    fk_Institution_id VARCHAR,
    fk_Company_id VARCHAR,
    fk_Supervisor_id VARCHAR,
    fk_People_id VARCHAR,
    fk_User_id VARCHAR
);

CREATE TABLE Teacher (
    id VARCHAR PRIMARY KEY,
    fk_People_id VARCHAR,
    fk_User_id VARCHAR
);

CREATE TABLE Institution (
    id VARCHAR PRIMARY KEY,
    fk_Organization_id VARCHAR,
    fk_User_id VARCHAR
);

CREATE TABLE Company (
    id VARCHAR PRIMARY KEY,
    fk_Organization_id VARCHAR,
    fk_User_id VARCHAR
);

CREATE TABLE Skill (
    id VARCHAR PRIMARY KEY,

```

```

        description VARCHAR,
        title VARCHAR,
        image_path VARCHAR,
        fk_Student_id VARCHAR
    );

CREATE TABLE Vancancy (
    id VARCHAR PRIMARY KEY,
    title VARCHAR,
    description VARCHAR,
    is_paid BOOLEAN,
    quantity INTEGER,
    created_at DATE,
    updated_at DATE,
    fk_Company_id VARCHAR
);

CREATE TABLE Courseware (
    id VARCHAR PRIMARY KEY,
    title VARCHAR,
    file_path VARCHAR,
    is_concluded BOOLEAN,
    description VARCHAR,
    created_at DATE,
    updated_at DATE,
    fk_Teacher_id VARCHAR,
    fk_Student_id VARCHAR
);

CREATE TABLE Chat (
    id VARCHAR PRIMARY KEY,
    created_at DATE,
    fk_User_creator_id VARCHAR,
    fk_User_participates_id_ VARCHAR
);

CREATE TABLE Schedule (
    id VARCHAR PRIMARY KEY,
    created_at DATE,

```

```
        updated_at DATE,  
        fk_Student_id VARCHAR  
    );  
  
CREATE TABLE Activity (  
    id VARCHAR PRIMARY KEY,  
    title VARCHAR,  
    description VARCHAR,  
    is_concluded BOOLEAN,  
    starts_date DATE,  
    ends_date DATE  
);  
  
CREATE TABLE Profile (  
    id VARCHAR PRIMARY KEY,  
    biography VARCHAR,  
    image_path VARCHAR,  
    cover_path VARCHAR,  
    fk_User_id VARCHAR  
);  
  
CREATE TABLE Supervisor (  
    id VARCHAR PRIMARY KEY,  
    fk_Company_id VARCHAR,  
    fk_People_id VARCHAR,  
    fk_User_id VARCHAR  
);  
  
CREATE TABLE People (  
    id VARCHAR PRIMARY KEY,  
    birth_date DATE,  
    fk_User_id VARCHAR  
);  
  
CREATE TABLE Organization (  
    id VARCHAR PRIMARY KEY,  
    founded_since DATE,  
    cnpj VARCHAR UNIQUE,  
    fk_User_id VARCHAR
```

```
);
```

```
CREATE TABLE Message (  
    id VARCHAR PRIMARY KEY,  
    content VARCHAR,  
    created_at DATE,  
    fk_Chat_id VARCHAR,  
    fk_User_id VARCHAR
```

```
);
```

```
CREATE TABLE Student_Vacancy (  
    fk_Student_id VARCHAR,  
    fk_Vancancy_id VARCHAR
```

```
);
```

```
CREATE TABLE Teacher_Institution (  
    fk_Teacher_id VARCHAR,  
    fk_Institution_id VARCHAR
```

```
);
```

```
CREATE TABLE Student_Activity_Schedule (  
    fk_Student_id VARCHAR,  
    fk_Activity_id VARCHAR,  
    fk_Schedule_id VARCHAR
```

```
);
```

```
CREATE TABLE Institution_Company (  
    fk_Institution_id VARCHAR,  
    fk_Company_id VARCHAR
```

```
);
```

```
CREATE TABLE Supervisor_Vacancy (  
    fk_Supervisor_id VARCHAR,  
    fk_Vancancy_id VARCHAR
```

```
);
```

```
CREATE TABLE Post (  
    id VARCHAR PRIMARY KEY,  
    title VARCHAR,
```

```
description VARCHAR,  
image_path VARCHAR,  
created_at DATE,  
updated_at DATE,  
fk_User_id VARCHAR  
);
```

```
CREATE TABLE Address (  
id VARCHAR PRIMARY KEY,  
state VARCHAR,  
city VARCHAR,  
district VARCHAR,  
number VARCHAR,  
fk_User_id VARCHAR  
);
```

```
CREATE TABLE Goals (  
id VARCHAR PRIMARY KEY,  
title VARCHAR,  
description VARCHAR,  
is_concluded BOOLEAN,  
image_path VARCHAR,  
fk_Student_id VARCHAR,  
fk_Teacher_id VARCHAR  
);
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_2  
FOREIGN KEY (fk_Teacher_id)  
REFERENCES Teacher (id)  
ON DELETE RESTRICT;
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_3  
FOREIGN KEY (fk_Institution_id)  
REFERENCES Institution (id)  
ON DELETE CASCADE;
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_4  
FOREIGN KEY (fk_Company_id)  
REFERENCES Company (id)
```

```
ON DELETE SET NULL;
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_5  
    FOREIGN KEY (fk_Supervisor_id)  
    REFERENCES Supervisor (id)  
    ON DELETE SET NULL;
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_6  
    FOREIGN KEY (fk_People_id)  
    REFERENCES People (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Student ADD CONSTRAINT FK_Student_7  
    FOREIGN KEY (fk_User_id)  
    REFERENCES User (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Teacher ADD CONSTRAINT FK_Teacher_2  
    FOREIGN KEY (fk_People_id)  
    REFERENCES People (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Teacher ADD CONSTRAINT FK_Teacher_3  
    FOREIGN KEY (fk_User_id)  
    REFERENCES User (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Institution ADD CONSTRAINT FK_Institution_2  
    FOREIGN KEY (fk_Organization_id)  
    REFERENCES Organization (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Institution ADD CONSTRAINT FK_Institution_3  
    FOREIGN KEY (fk_User_id)  
    REFERENCES User (id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE Company ADD CONSTRAINT FK_Company_2  
    FOREIGN KEY (fk_Organization_id)
```

```
REFERENCES Organization (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Company ADD CONSTRAINT FK_Company_3
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Skill ADD CONSTRAINT FK_Skill_2
FOREIGN KEY (fk_Student_id)
REFERENCES Student (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Vancancy ADD CONSTRAINT FK_Vancancy_2
FOREIGN KEY (fk_Company_id)
REFERENCES Company (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Courseware ADD CONSTRAINT FK_Courseware_2
FOREIGN KEY (fk_Teacher_id)
REFERENCES Teacher (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Courseware ADD CONSTRAINT FK_Courseware_3
FOREIGN KEY (fk_Student_id)
REFERENCES Student (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Chat ADD CONSTRAINT FK_Chat_2
FOREIGN KEY (fk_User_creator_id, fk_User_participates_id_)
REFERENCES User (id, id)
ON DELETE CASCADE;
```

```
ALTER TABLE Schedule ADD CONSTRAINT FK_Schedule_2
FOREIGN KEY (fk_Student_id)
REFERENCES Student (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Profile ADD CONSTRAINT FK_Profile_2
```

```
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Supervisor ADD CONSTRAINT FK_Supervisor_2
FOREIGN KEY (fk_Company_id)
REFERENCES Company (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Supervisor ADD CONSTRAINT FK_Supervisor_3
FOREIGN KEY (fk_People_id)
REFERENCES People (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Supervisor ADD CONSTRAINT FK_Supervisor_4
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE People ADD CONSTRAINT FK_People_2
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Organization ADD CONSTRAINT FK_Organization_2
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Message ADD CONSTRAINT FK_Message_2
FOREIGN KEY (fk_Chat_id)
REFERENCES Chat (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Message ADD CONSTRAINT FK_Message_3
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```



```
ALTER TABLE Student_Vacancy
ADD CONSTRAINT FK_Student_Vacancy_1
    FOREIGN KEY (fk_Student_id)
    REFERENCES Student (id)
    ON DELETE SET NULL;
```

```
ALTER TABLE Student_Vacancy
ADD CONSTRAINT FK_Student_Vacancy_2
    FOREIGN KEY (fk_Vancancy_id)
    REFERENCES Vancancy (id)
    ON DELETE SET NULL;
```

```
ALTER TABLE Teacher_Institution
ADD CONSTRAINT FK_Teacher_Institution_1
    FOREIGN KEY (fk_Teacher_id)
    REFERENCES Teacher (id)
    ON DELETE RESTRICT;
```

```
ALTER TABLE Teacher_Institution
ADD CONSTRAINT FK_Teacher_Institution_2
    FOREIGN KEY (fk_Institution_id)
    REFERENCES Institution (id)
    ON DELETE RESTRICT;
```

```
ALTER TABLE Student_Activity_Schedule
ADD CONSTRAINT FK_Student_Activity_Schedule_1
    FOREIGN KEY (fk_Student_id)
    REFERENCES Student (id)
    ON DELETE RESTRICT;
```

```
ALTER TABLE Student_Activity_Schedule
ADD CONSTRAINT FK_Student_Activity_Schedule_2
    FOREIGN KEY (fk_Activity_id)
    REFERENCES Activity (id)
    ON DELETE NO ACTION;
```

```
ALTER TABLE Student_Activity_Schedule
ADD CONSTRAINT FK_Student_Activity_Schedule_3
    FOREIGN KEY (fk_Schedule_id)
```

```
REFERENCES Schedule (id)
ON DELETE RESTRICT;
```

```
ALTER TABLE Institution_Company
ADD CONSTRAINT FK_Institution_Company_1
FOREIGN KEY (fk_Institution_id)
REFERENCES Institution (id)
ON DELETE SET NULL;
```

```
ALTER TABLE Institution_Company
ADD CONSTRAINT FK_Institution_Company_2
FOREIGN KEY (fk_Company_id)
REFERENCES Company (id)
ON DELETE SET NULL;
```

```
ALTER TABLE Supervisor_Vacancy
ADD CONSTRAINT FK_Supervisor_Vacancy_1
FOREIGN KEY (fk_Supervisor_id)
REFERENCES Supervisor (id)
ON DELETE SET NULL;
```

```
ALTER TABLE Supervisor_Vacancy
ADD CONSTRAINT FK_Supervisor_Vacancy_2
FOREIGN KEY (fk_Vacancy_id)
REFERENCES Vacancy (id)
ON DELETE SET NULL;
```

```
ALTER TABLE Post ADD CONSTRAINT FK_Post_2
FOREIGN KEY (fk_User_id)
REFERENCES User (id)
ON DELETE CASCADE;
```

```
ALTER TABLE Address ADD CONSTRAINT FK_Address_2
FOREIGN KEY (fk_User_id)
REFERENCES User (id);
```

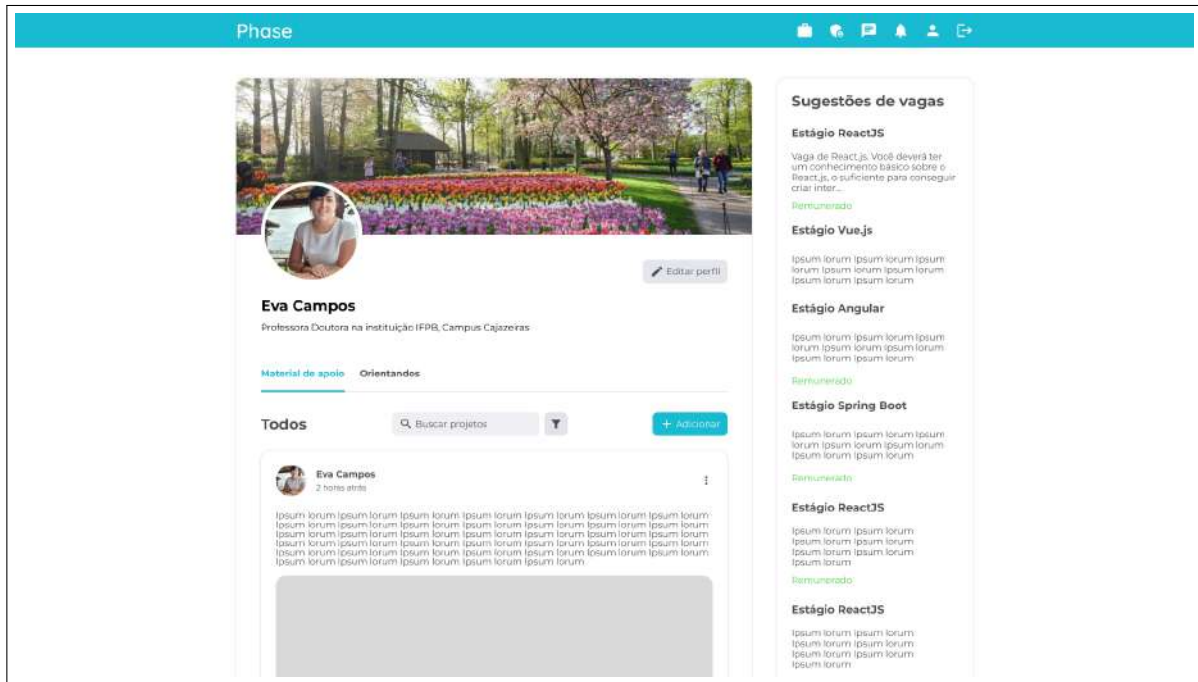
```
ALTER TABLE Goals ADD CONSTRAINT FK_Goals_2
FOREIGN KEY (fk_Student_id)
REFERENCES Student (id);
```

```
ALTER TABLE Goals ADD CONSTRAINT FK_Goals_3
FOREIGN KEY (fk_Teacher_id)
REFERENCES Teacher (id);
```

APÊNDICE B – PROTÓTIPOS

A seguir, serão apresentadas as demais telas que foram prototipadas.

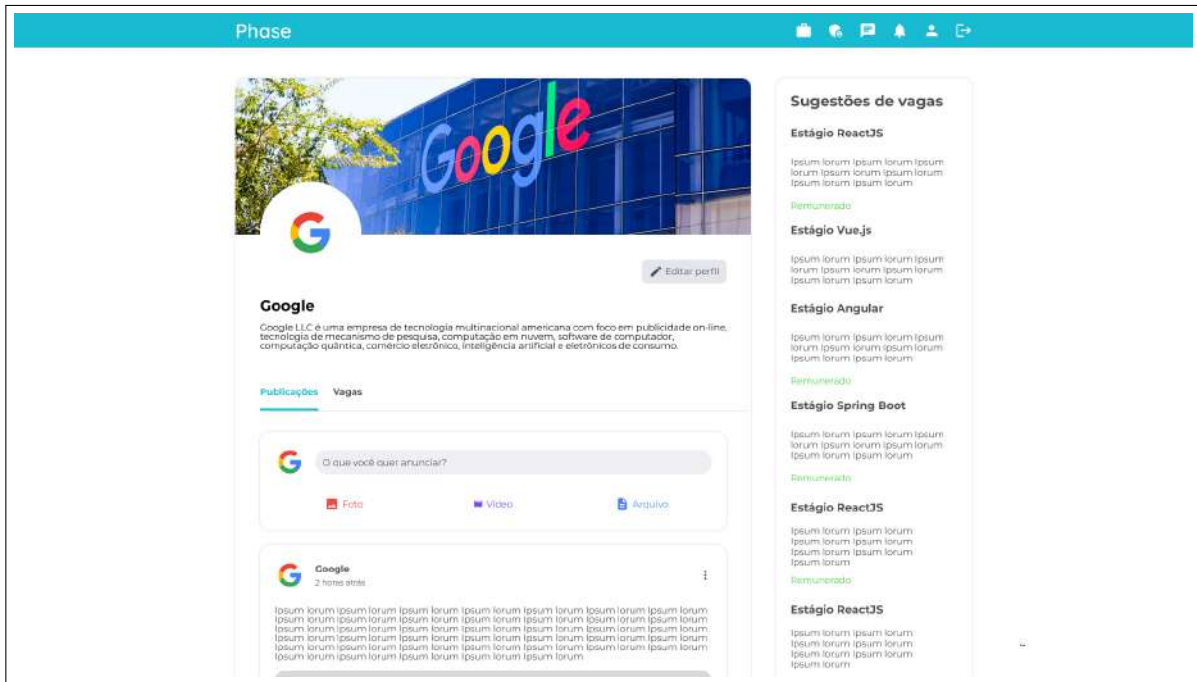
Figura 17 – Tela do perfil do professor orientador



Fonte: Elaborado pelo autor

A figura 17 apresenta uma tela de perfil de um professor orientador contendo suas informações tais como: nome, biografia, postagens de materiais de apoio e alunos.

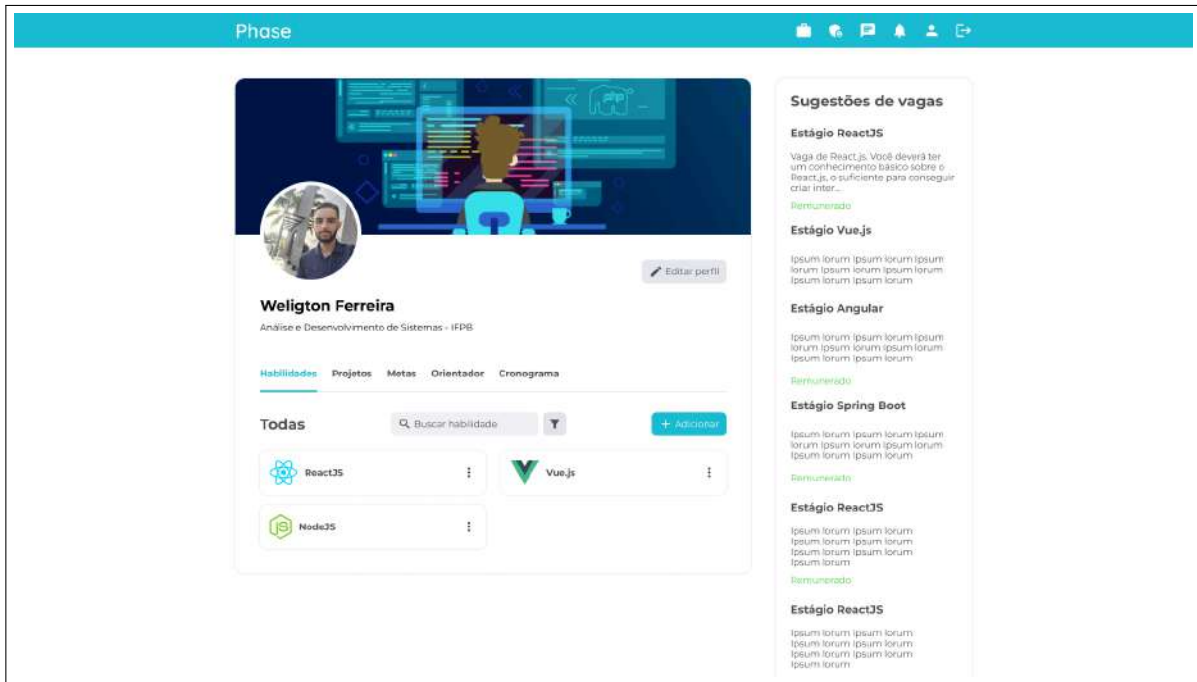
Figura 18 – Tela do perfil de uma empresa



Fonte: Elaborado pelo autor

A figura 18 apresenta a tela de perfil de uma empresa contendo sua informações tais como: nome, biografia, publicações e vagas.

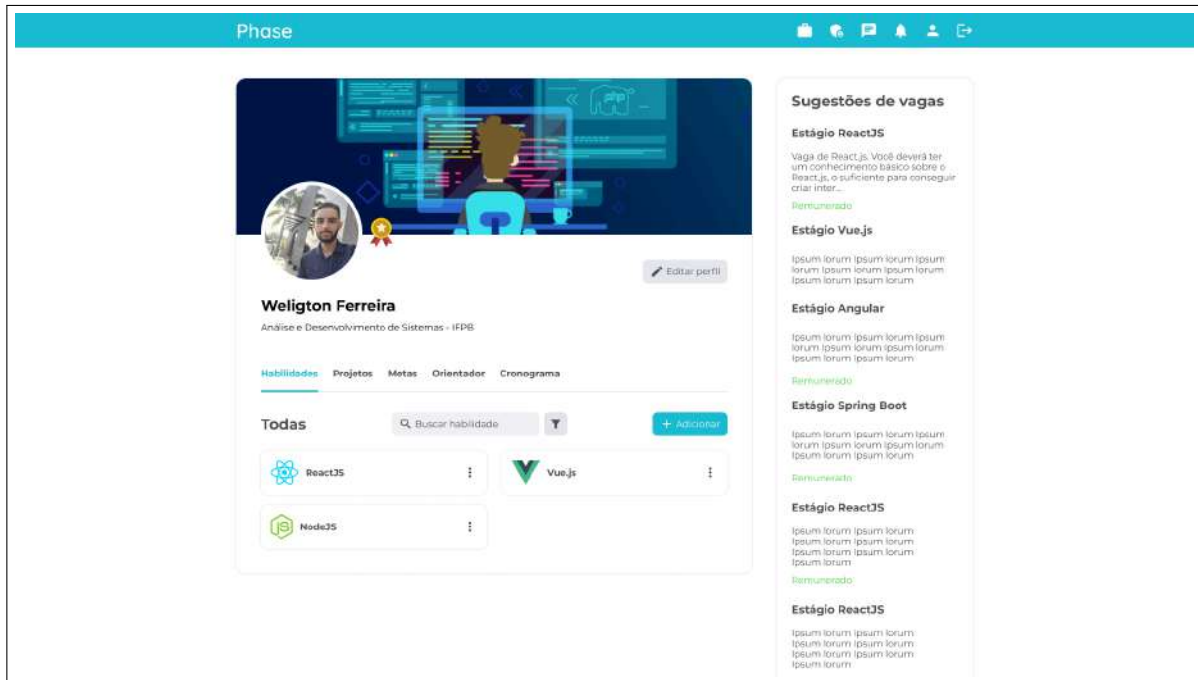
Figura 19 – Tela do perfil do discente



Fonte: Elaborado pelo autor

A figura 19 apresenta a tela de perfil de um discente contendo suas informações tais como: nome, biografia, habilidades, projetos, metas e cronograma. O discente que ainda não realizou nenhuma atividade não possui uma medalha. Portanto, a medalha surge e progride conforme o progresso do discente.

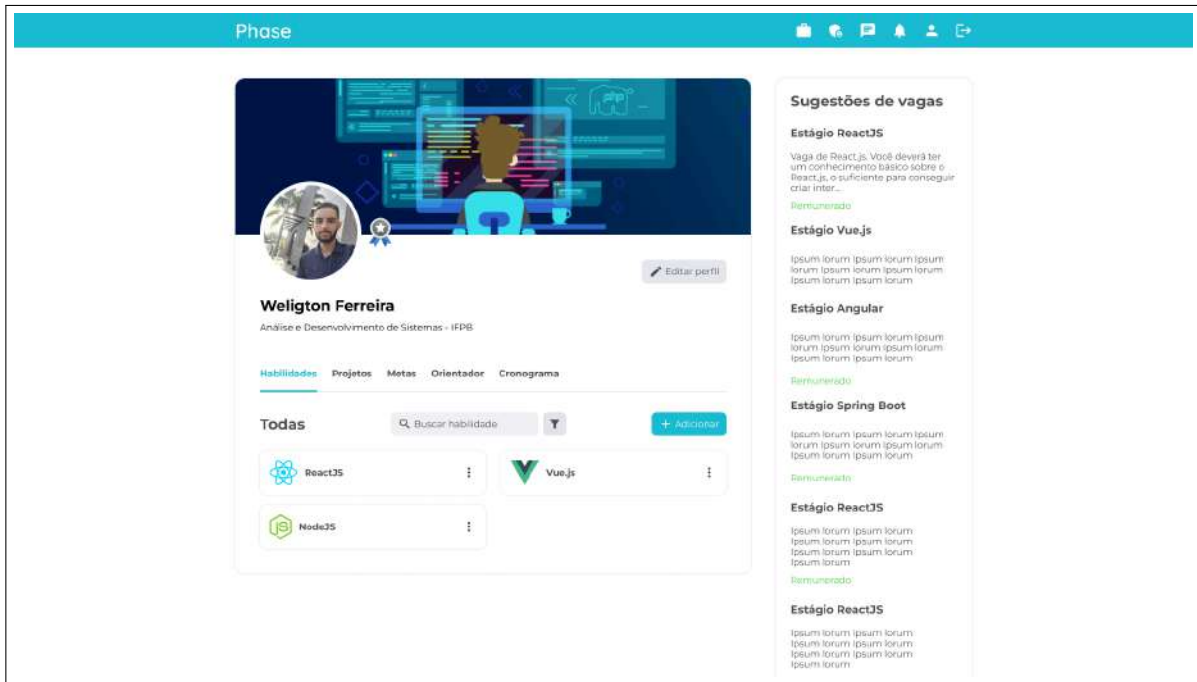
Figura 20 – Tela do perfil do discente com medalha de ouro



Fonte: Elaborado pelo autor

A figura 20 apresenta a tela de perfil de um discente que possui medalha de ouro. A medalha de ouro representa o estágio final de progresso onde o discente já realizou grande parte de suas atividades ou as concluiu. O progresso para a medalha de ouro pode ser acelerado caso o discente cumpra suas atividades com mais rapidez e demonstre bons resultados frente ao estágio supervisionado segundo for avaliado pelo supervisor.

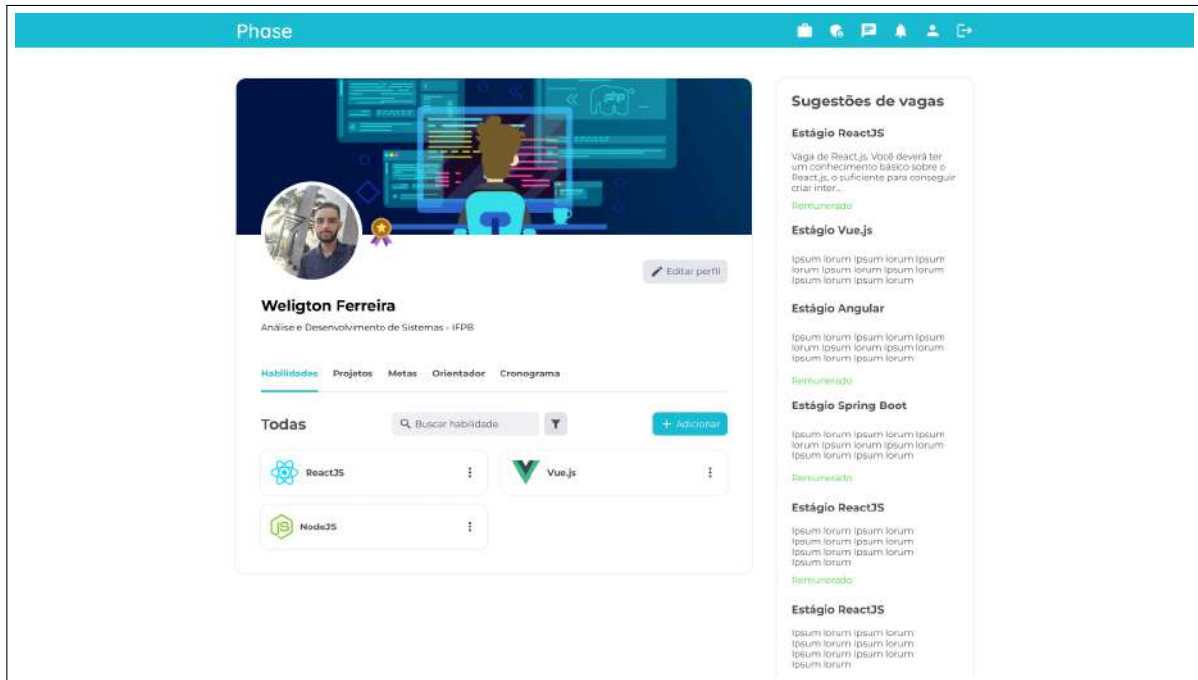
Figura 21 – Tela do perfil do discente com medalha de prata



Fonte: Elaborado pelo autor

A figura 21 apresenta a tela de perfil de um discente que possui medalha de prata. A medalha de prata representa o estágio intermediário de progresso onde o discente já realizou uma quantidade considerável de atividades.

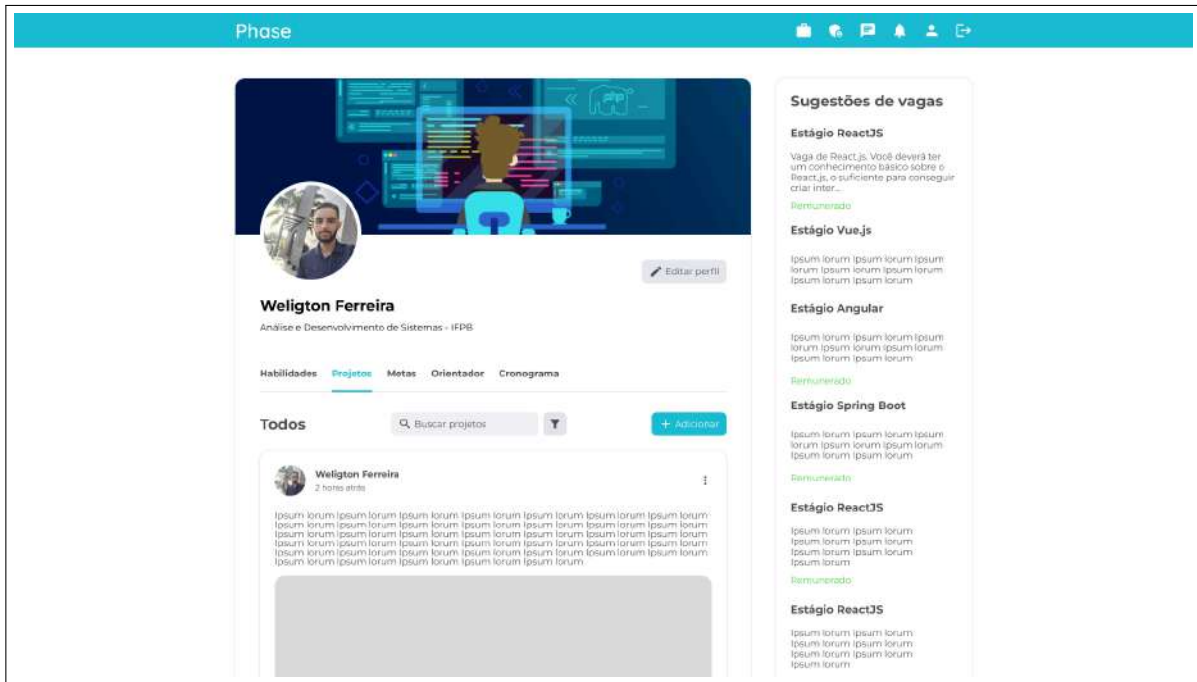
Figura 22 – Tela do perfil do discente com medalha de bronze



Fonte: Elaborado pelo autor

A figura 22 apresenta a tela de perfil de um discente que possui medalha de bronze. A medalha de bronze representa o estágio inicial de progresso onde o discente já realizou suas primeiras atividades.

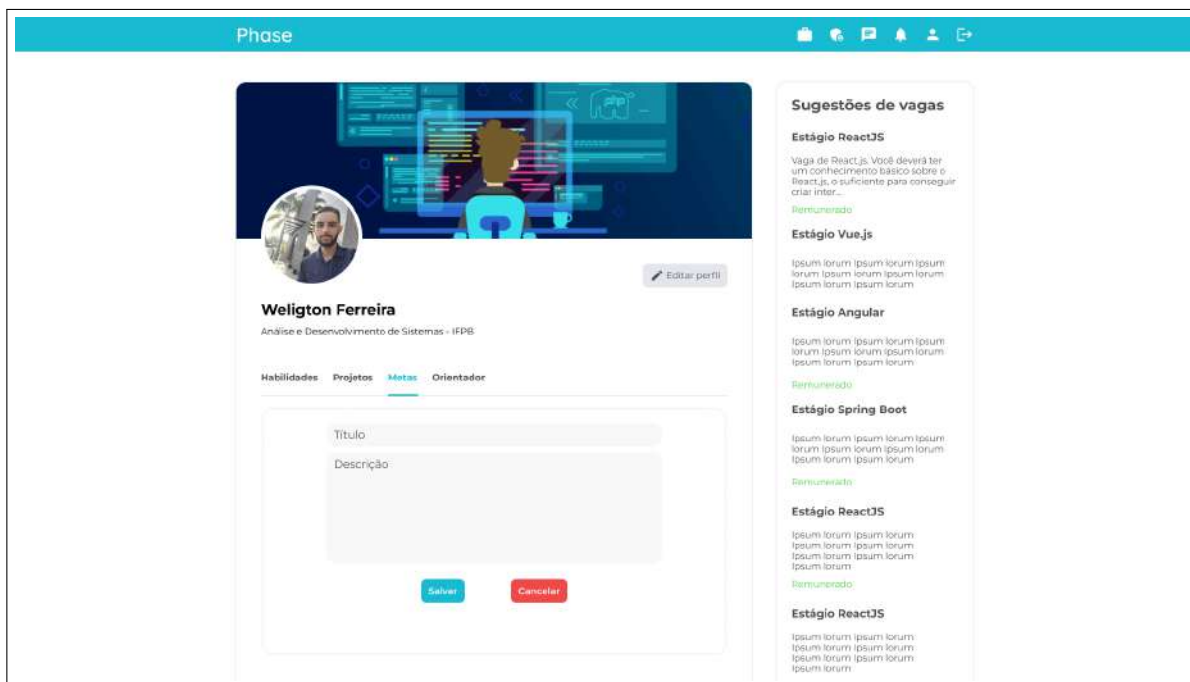
Figura 23 – Tela de projetos do discente



Fonte: Elaborado pelo autor

A figura 23 apresenta a tela de projetos do discente onde poderá exibir os projetos e atividades realizadas durante o curso ou durante o estágio supervisionado através de publicações.

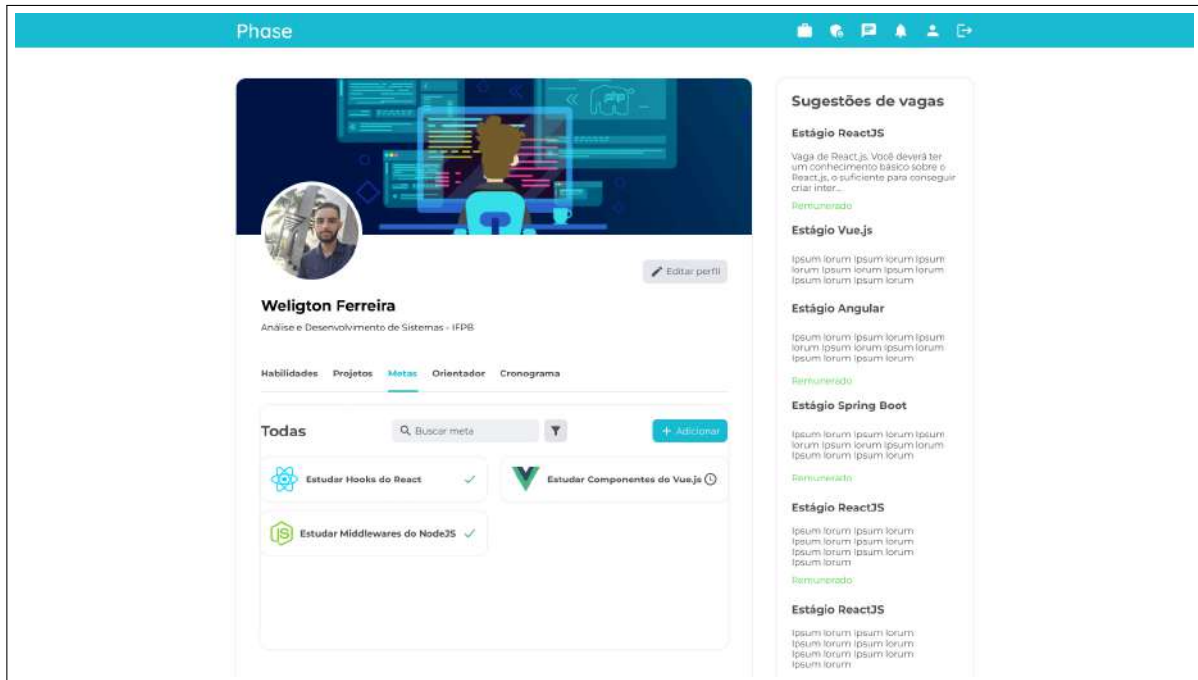
Figura 24 – Tela de cadastro de metas



Fonte: Elaborado pelo autor

A figura 24 apresenta a tela de cadastro de metas realizado por um discente. Para realizar esta operação será necessário fornecer o título e a descrição da meta.

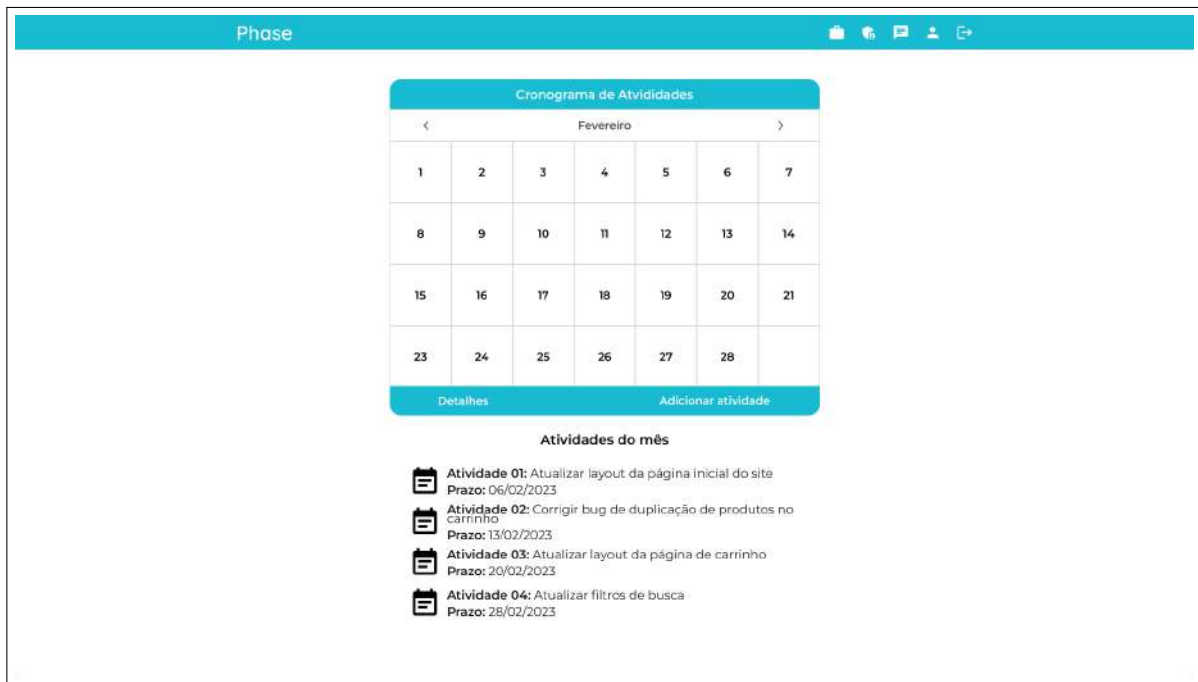
Figura 25 – Tela de listagem de metas



Fonte: Elaborado pelo autor

A figura 26 apresenta a tela de listagem de metas onde o discente poderá acompanhar as metas finalizadas e pendentes.

Figura 26 – Tela do cronograma de atividades



Fonte: Elaborado pelo autor

A figura 25 apresenta a tela do cronograma de atividades realizadas pelo discente ao longo do estágio supervisionado.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

Campus Cajazeiras

Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)

CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Restrito

TCC Corrigido contendo a ata

Assunto: TCC Corrigido contendo a ata
Assinado por: Weligton Ferreira
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Direito Autoral (Art. 24, III, da Lei no 9.610/1998)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

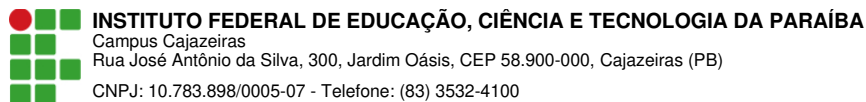
- **Weligton Ferreira de Araujo, ALUNO (201912010026) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 19/04/2023 22:42:47.

Este documento foi armazenado no SUAP em 19/04/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 811463

Código de Autenticação: 3564d52e38





Documento Digitalizado Restrito

Entrega do Trabalho de Conclusão de Curso

Assunto: Entrega do Trabalho de Conclusão de Curso
Assinado por: Weligton Ferreira
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Direito Autoral (Art. 24, III, da Lei no 9.610/1998)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Weligton Ferreira de Araujo, ALUNO (201912010026) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 25/04/2023 19:49:52.

Este documento foi armazenado no SUAP em 25/04/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 815414
Código de Autenticação: 7583ef4e72

