

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

**TCC EASIER: APLICAÇÃO VOLTADA PARA AUXILIAR NO
ACOMPANHAMENTO DE TRABALHOS DE CONCLUSÃO DE CURSO
(TCCS) E NA BUSCA POR ORIENTADORES.**

JOSÉ FARLEY SOUSA DA SILVA

**Cajazeiras - PB
Fevereiro, 2024**

JOSÉ FARLEY SOUSA DA SILVA

**TCC EASIER: APLICAÇÃO VOLTADA PARA AUXILIAR NO ACOMPANHAMENTO
DE TRABALHOS DE CONCLUSÃO DE CURSO (TCCS) E NA BUSCA POR
ORIENTADORES.**

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.

Orientador

Prof. MSc. Michel da Silva.

**Cajazeiras - PB
Fevereiro, 2024**

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

S586t Silva, José Farley Sousa da.
TCC easier : aplicação voltada para auxiliar no acompanhamento de trabalhos de conclusão de curso (TCCs) e na busca por orientadores / José Farley Sousa da Silva. – 2024.

31f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.

Orientador(a): Prof. Me. Michel da Silva.

1. Desenvolvimento de sistemas. 2. TCC *easier*. 3. Aplicação *web*. 4. Orientação TCC. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU:004.4 (043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

JOSÉ FARLEY SOUSA DA SILVA

TCC EASIER: APLICAÇÃO VOLTADA PARA AUXILIAR NO ACOMPANHAMENTO DE TRABALHOS DE CONCLUSÃO DE CURSO (TCCS) E NA BUSCA POR ORIENTADORES

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.

Orientador

Prof. Me. Michel da Silva

Aprovada em: **02 de Agosto de 2024.**

Prof. Me. Michel da Silva - Orientador

Prof. Dr. Hudson Geovane de Medeiros - Avaliador

IFPB - Campus Cajazeiras

Prof. Dr. Leandro Luttiane da Silva Linhares - Avaliador

Documento assinado digitalmente



LEANDRO LUTTIANE DA SILVA LINHARES

Data: 08/08/2024 15:45:28-0300

Verifique em <https://validar.iti.gov.br>

Documento assinado eletronicamente por:

- **Michel da Silva, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 06/08/2024 10:04:11.
- **Hudson Geovane de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 06/08/2024 11:04:16.

Este documento foi emitido pelo SUAP em 06/08/2024. 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 587501
Verificador: 54b3d3a42f
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

RESUMO

O objetivo deste trabalho foi desenvolver o TCC Easier, uma aplicação web destinada a auxiliar os alunos do curso de Análise e Desenvolvimento de Sistemas (ADS) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *Campus Cajazeiras*, no processo de busca por orientadores para seus Trabalhos de Conclusão de Curso (TCC). A aplicação também serve como uma ferramenta valiosa para os professores orientadores no acompanhamento dos trabalhos de seus orientandos. Com funcionalidades de gerenciamento das orientações, o TCC Easier possui potencial para contribuir com a melhoria dos TCCs do curso de ADS, oferecendo uma abordagem inovadora. A aplicação foi desenvolvida com foco na escalabilidade e facilidade de manutenção, visando sua expansão para outros cursos no futuro. Foram utilizadas tecnologias modernas e consolidadas, como *React*, *Node*, *Prisma ORM*, *PostgreSQL* e *TypeScript*, com o objetivo de criar um sistema robusto e de alto desempenho. O resultado deste trabalho é uma aplicação que prioriza desempenho, escalabilidade e facilidade de manutenção, oferecendo uma interface gráfica intuitiva e fluida para uma experiência de usuário satisfatória.

Palavras-chave: Trabalho de Conclusão de Curso. Artigo. *Aplicação Web*. TCC Easier

ABSTRACT

The objective of this work was to develop TCC Easier, a web application designed to assist students in the Analysis and Systems Development (ADS) program at the Federal Institute of Education, Science and Technology of Paraíba (IFPB), Cajazeiras Campus, in their search for advisors for their Final Projects (TCC). The application also serves as a valuable tool for faculty advisors in monitoring their advisees' projects. With features for managing advisement, TCC Easier has the potential to contribute to the improvement of the TCCs in the ADS program, offering an innovative approach. The application was developed with a focus on scalability and ease of maintenance, aiming for future expansion to other programs. Modern and established technologies were used, such as React, Node, Prisma ORM, PostgreSQL, and TypeScript, with the goal of creating a robust and high-performance system. The result of this work is an application that prioritizes performance, scalability, and ease of maintenance, providing an intuitive and smooth graphical interface for a satisfying user experience.

Keywords: Final Year Project (TCC) . Paper. Web application. TCC Easier.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Padrão de Arquitetura MVC | 23 |
| Figura 2 – Diagrama de caso de uso | 27 |
| Figura 3 – Modelo Entidade-Relacionamento | 28 |
| Figura 4 – Tela de login | 29 |
| Figura 5 – Tela de inicio do aluno | 29 |
| Figura 6 – Tela de orientadores do aluno | 30 |
| Figura 7 – Tela de reuniões do aluno | 31 |
| Figura 8 – Tela de tarefas do aluno | 31 |
| Figura 9 – Tela de inicio do professor | 32 |
| Figura 10 – Tela de tarefas do professor | 33 |
| Figura 11 – Tela de reuniões do professor | 33 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|--|
| ADS | Análise e Desenvolvimento de Sistemas |
| API | <i>Application Programming Interface</i> |
| IFPB | Instituto Federal de Educação, Ciência e Tecnologia da Paraíba |
| MER | Modelo Entidade-Relacionamento |
| MVC | <i>Model-View-Controller</i> |
| ORM | <i>Object-Relational Mapping</i> |
| RF | Requisitos Funcionais |
| RNF | Requisitos Não Funcionais |
| TCC | Trabalho de Conclusão de Curso |
| SQL | <i>Structured Query Language</i> |

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 15 |
| 1.1 | Objetivos | 15 |
| 1.1.1 | Objetivo Geral | 15 |
| 1.1.2 | Objetivos Específicos | 16 |
| 1.2 | METODOLOGIA | 16 |
| 1.3 | Organização do documento | 16 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 18 |
| 2.1 | Trabalho de conclusão de curso | 18 |
| 2.2 | Conceitos e trabalhos relacionados | 18 |
| 2.3 | Tecnologias e arquitetura | 21 |
| 2.3.1 | <i>Front-end</i> | 21 |
| 2.3.2 | <i>Back-end</i> | 21 |
| 2.3.3 | <i>ReactJS</i> | 21 |
| 2.3.4 | <i>NodeJS</i> | 22 |
| 2.3.5 | <i>PostgreSQL</i> | 22 |
| 2.3.6 | <i>Typescript</i> | 22 |
| 2.3.7 | <i>Prisma</i> | 23 |
| 2.3.8 | Arquitetura | 23 |
| 3 | <i>MODELAGEM TCC EASIER</i> | 24 |
| 3.1 | Requisitos funcionais | 24 |
| 3.2 | Requisitos não funcionais | 26 |
| 3.3 | Diagrama de caso de uso | 26 |
| 3.4 | Modelo Entidade-Relacionamento | 27 |
| 3.5 | Telas | 28 |
| 3.5.1 | Tela de <i>login</i> do aluno e professor | 28 |
| 3.5.2 | Tela de inicio do aluno | 29 |
| 3.5.3 | Tela de orientadores do aluno | 30 |
| 3.5.4 | Tela de reuniões do aluno | 30 |
| 3.5.5 | Tela de tarefas do aluno | 31 |
| 3.5.6 | Tela de inicio do professor | 32 |
| 3.5.7 | Tela de tarefas do professor | 32 |
| 3.5.8 | Tela de reuniões do professor | 33 |
| 4 | CONSIDERAÇÕES FINAIS | 34 |

| | |
|--------------------------|-----------|
| REFERÊNCIAS | 35 |
|--------------------------|-----------|

1 INTRODUÇÃO

Considerando a demanda existente no atual sistema acadêmico brasileiro, onde a elaboração e defesa de um Trabalho de Conclusão de Curso (TCC) é uma etapa essencial para a conclusão dos cursos superiores, reconheceu-se a necessidade de ferramentas que possam auxiliar os alunos nesse processo. Nesse contexto, foi proposta a criação do TCC *Easier*, uma aplicação *web* projetada para ser utilizada inicialmente no Curso de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), *Campus Cajazeiras*.

Um dos principais desafios enfrentados pelos estudantes do curso de ADS durante o processo de elaboração do TCC está relacionado à busca por um orientador. Até então, a instituição não possuía uma ferramenta específica que facilitasse os alunos a encontrar um docente disponível para orientação. Em vez disso, o professor responsável pela disciplina de TCC elaborava manualmente uma lista dos professores disponíveis para orientação, o que demandava tempo e atualizações manuais frequentes.

Outro aspecto que requeria atenção era o acompanhamento do desenvolvimento do TCC por parte do orientador. Como não havia uma plataforma dedicada na instituição para esse propósito, todo o acompanhamento ficava sob responsabilidade do docente, o que podia sobrecarregá-lo e resultar em uma qualidade de orientação insatisfatória, impactando negativamente na qualidade dos projetos.

Diante desses problemas e inspirado por projetos com propostas semelhantes, surgiu o TCC *Easier*. Trata-se de uma aplicação *web* que conecta alunos e orientadores, proporcionando um ambiente adequado para o acompanhamento do desenvolvimento dos TCCs. Os trabalhos relacionados a este projeto serão apresentados no Capítulo 2.

O desenvolvimento dessa aplicação foi realizado utilizando as tecnologias *React*, *Node*, *PostgreSQL* e *Prisma ORM*.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste trabalho foi desenvolver uma aplicação *web* que conectasse alunos e professores do IFPB, *Campus Cajazeiras*, e também auxiliasse no processo de acompanhamento dos trabalhos de conclusão de curso por parte dos professores, visando aprimorar a qualidade desses projetos.

1.1.2 Objetivos Específicos

- Identificar necessidades dos alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFPB, *Campus Cajazeiras*, em relação ao Trabalho de Conclusão de Curso;
- Efetuar o levantamento de requisitos da aplicação, assim como o desenvolvimento de seu diagrama de casos de uso;
- Elaborar a arquitetura de software, definindo as tecnologias a serem utilizadas em cada camada;
- Projetar e implementar banco de dados eficiente para o armazenamento de informações;
- Desenvolver interface de usuário que proporcione ao usuário uma experiência intuitiva e agradável.

1.2 METODOLOGIA

O desenvolvimento deste trabalho foi dividido em duas etapas. A primeira consistiu na elaboração de sua proposta, que envolveu, resumidamente, a avaliação de um determinado problema e a definição de uma solução, análise e projeto da aplicação, prototipação de telas e a escrita deste documento, o qual passou por um processo de avaliação inicial por meio de uma banca examinadora. Esse foi o escopo da disciplina de Trabalho de Conclusão de Curso I do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFPB - *Campus Cajazeiras*.

Na segunda etapa, efetuaram-se ajustes na proposta de acordo com as sugestões apresentadas pela banca examinadora e situações ainda não observadas durante a etapa inicial do trabalho. Além disso, foram implementados o *back-end* e o *front-end* da aplicação TCC *Easier*, tornando-a funcional, bem como a elaboração do documento final do Trabalho de Conclusão de Curso.

1.3 ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado em quatro capítulos. No primeiro capítulo, são apresentados a motivação para o desenvolvimento do trabalho e os objetivos que deram origem à concepção do projeto. O Capítulo 2 inicia-se trazendo algumas informações adicionais sobre os Trabalhos de Conclusão de Curso que contribuíram para a definição da proposta deste trabalho. Logo após são listados alguns conceitos importantes e trabalhos relacionados. Finalizando o segundo capítulo, são efetuadas

breves descrições sobre as tecnologias adotadas para o desenvolvimento do *TCC Easier*. O terceiro capítulo fornece informações relevantes de análise e projeto, abrangendo os requisitos funcionais e não funcionais, diagrama de caso de uso, o modelo entidade-relacionamento (MER) e os protótipos de tela da aplicação. Por fim, o Capítulo 4 realiza as considerações finais referentes à etapa inicial deste Trabalho de Conclusão de Curso e traz uma perspectiva de atividades a serem cumpridas para a sua conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

É cada vez mais evidente o constante surgimento de novas ferramentas web para a prestação de serviços. Isso ocorre principalmente devido à crescente democratização do acesso à internet. Segundo a Pesquisa Nacional por Amostra de Domicílios, em 2021, 90,0% dos lares brasileiros tinham acesso à internet. Além da questão do acesso à internet, o baixo custo de *hardware* para acesso a essas ferramentas torna-as um ótimo meio para a disseminação de serviços, pesquisas e materiais acadêmicos. Tendo isso em vista, o TCC *Easier* se insere nesse contexto com o objetivo de diminuir as barreiras que poderiam dificultar seu acesso.

2.1 TRABALHO DE CONCLUSÃO DE CURSO

Segundo PEREIRA (2012), “para a grande maioria dos discentes, o Trabalho de Conclusão de Curso (TCC) representa a primeira e única produção científica construída ao longo dos anos na experiência da Graduação. Essa construção segue um rigor epistemológico, estrutural e metodológico, alinhado às normas de trabalho acadêmico adotadas pelas instituições de ensino superior”. Apesar de o formato do TCC poder variar de acordo com a instituição, seu propósito central é aplicar o pensamento crítico do aluno, demonstrar os conhecimentos adquiridos ao longo do curso e evidenciar a capacidade de se aprofundar em um tema de interesse, produzindo um material de relevância acadêmica ou algum produto com impacto positivo na sociedade.

O processo de elaboração de um Trabalho de Conclusão de Curso geralmente inicia com a busca do aluno por um orientador. O orientador assume a responsabilidade de auxiliar o aluno na definição de um tema para o trabalho, caso ele ainda não o tenha, e orientá-lo no processo de pesquisa, definição de cronograma e elaboração do projeto. Em muitos casos, a instituição de ensino media esse processo por meio de uma plataforma *online* própria ou terceirizada. No entanto, nem todas as instituições possuem esse tipo de ferramenta, como é o caso do IFPB. Isso ocorre devido a diversos motivos, como falta de recursos financeiros ou incompatibilidade de interesses. O fato é que muitos alunos buscam orientadores que possuam experiência ou conhecimento na área de pesquisa do tema de TCC. Portanto, atuar de modo a facilitar e mediar esse processo pode ter um impacto direto na qualidade dos projetos de conclusão de curso da instituição.

2.2 CONCEITOS E TRABALHOS RELACIONADOS

Para atender aos objetivos propostos, é necessário abordar alguns conceitos no desenvolvimento da aplicação. Entre eles, destacam-se os princípios de código limpo

descritos por Martin (2019) em seu livro *Código Limpo: Habilidades Práticas do Agile Software*. No livro, o autor discute técnicas que facilitam a legibilidade e a manutenção do código. Essas técnicas incluem o uso de nomes significativos, a atribuição de uma única responsabilidade para cada função, funções curtas, uma estruturação e formatação adequadas, bem como a eliminação da repetição de código.

A escolha dos padrões de projeto a serem utilizados no desenvolvimento é crucial para garantir a qualidade, organização e manutenibilidade do sistema. Segundo Gamma et al. (2000), no livro 'Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos', um padrão de projeto é uma solução geral para um problema recorrente no desenvolvimento de software. Ele descreve um esquema de organização de componentes de software que proporciona uma solução reutilizável para um problema específico.

Os padrões de projeto são categorizados em três grupos principais: padrões criacionais, estruturais e comportamentais. Os padrões criacionais lidam com a criação de objetos de uma forma que seja independente do sistema, melhorando a flexibilidade e reutilização do código. Os padrões estruturais estão relacionados com a composição de classes e objetos para formar estruturas maiores, facilitando a compreensão e manutenção do código. Já os padrões comportamentais se concentram nos algoritmos e responsabilidades entre objetos, promovendo a comunicação e o gerenciamento de comportamentos entre eles.

Dentro de cada grupo, há um determinado número de padrões que atendem a casos de uso recorrentes. A ideia é que, durante o desenvolvimento da aplicação proposta por este trabalho, esses padrões sejam aplicados para tornar o código extensível e bem estruturado.

Além dos conceitos previamente mencionados, diversos trabalhos com propostas semelhantes desempenharam um papel fundamental no desenvolvimento do TCC *Easier*. Entre eles, destaca-se o trabalho de Arruda (2010), intitulado "Sistema de Gestão de Trabalhos de Final de Curso". Em seu estudo, Arruda abrange todo o fluxo de entrega e gerenciamento dos trabalhos de conclusão de curso. A ferramenta proposta por ele tem como principais funcionalidades o armazenamento de revisões do texto, o envio do trabalho para professores, bibliotecários e revisores, o registro de comentários sobre a revisão, a exibição de prazos de entrega e a apresentação da evolução do trabalho de acordo com as etapas do processo. Em seu trabalho, Arruda não especifica as tecnologias utilizadas; ele se atém mais ao processo de levantamento de requisitos e à pesquisa de ferramentas e projetos correlatos.

Outro trabalho relevante é o de Cabral (2017), intitulado "TCC *online*: Sistema de gerenciamento de trabalhos de conclusão de curso", que visa especificamente

propor uma ferramenta que facilite a interação entre aluno e orientador. Em seu trabalho, Cabral propõe uma ferramenta cujas principais funcionalidades são permitir que o aluno encontre um orientador, a criação e revisão de projetos, e a possibilidade de o orientador designar atividades para os alunos. Além disso, o trabalho de Cabral utiliza *PHP* (uma linguagem de programação), *Javascript* (uma linguagem de programação), *MySQL* (um sistema de gerenciamento de banco de dados), *Bootstrap* (um framework front-end), *HTML5* (uma linguagem de marcação) e *CSS3* (uma linguagem de estilo).

Tendo em vista esses dois trabalhos, que tiveram importância ímpar para o desenvolvimento do TCC *Easier*, podemos destacar algumas diferenças principais, especialmente em relação ao trabalho de Cabral, com o qual o TCC *Easier* mais se assemelha no que diz respeito às funcionalidades oferecidas. Ambos os sistemas utilizam *JavaScript*, mas de maneiras distintas. O trabalho de Cabral (2017) utiliza *JavaScript* no *front-end* para a criação da interface do usuário e *PHP* no *back-end*, que lida com a lógica do servidor e o gerenciamento de banco de dados.

Por outro lado, o TCC *Easier* adota uma abordagem mais moderna, utilizando *JavaScript* no *front-end* por meio da biblioteca *React* e *JavaScript* no *back-end* através do *Node.js*. Além disso, o TCC *Easier* é um sistema de **Single Page Application (SPA)**, enquanto a ferramenta de Cabral é um exemplo de **Server Side Rendering (SSR)**.

SPA (Single Page Application) é uma arquitetura em que toda a aplicação é carregada em uma única página da *web*, e as interações subsequentes são gerenciadas de forma dinâmica sem a necessidade de recarregar a página inteira. Isso proporciona uma experiência de usuário mais fluida e rápida, com transições mais suaves entre diferentes partes da aplicação e melhor desempenho em termos de resposta e carregamento.

SSR (Server Side Rendering), por outro lado, é uma abordagem em que cada solicitação ao servidor resulta no envio de uma nova página *HTML* completa para o cliente. Isso pode ser vantajoso para a otimização para motores de busca e para os tempos iniciais de carregamento, pois o conteúdo é gerado e enviado pelo servidor antes de chegar ao navegador do usuário.

O TCC *Easier*, como uma aplicação *SPA*, oferece várias vantagens, incluindo uma navegação mais rápida e uma experiência mais interativa para o usuário, pois reduz a quantidade de dados que precisam ser carregados a cada interação e evita o recarregamento completo da página. Isso resulta em uma melhor performance e uma experiência mais fluida para os usuários.

Além das vantagens tecnológicas, o TCC *Easier* inclui funcionalidades adicionais que não estão presentes na ferramenta de Cabral. Entre essas funcionalidades,

destacam-se a capacidade de o professor marcar reuniões com o aluno diretamente pela plataforma, o que facilita a comunicação e o acompanhamento do progresso, e a opção para que o professor defina se possui vagas disponíveis para orientação, permitindo uma gestão mais eficiente das orientações e evitando sobrecarga de trabalho.

2.3 TECNOLOGIAS E ARQUITETURA

Com o objetivo de fornecer as funcionalidades propostas, bem como garantir a escalabilidade e facilidade de manutenção, é proposta a utilização das seguintes tecnologias e arquitetura para o desenvolvimento do *TCC Easier*.

2.3.1 *Front-end*

O *front-end* refere-se à parte de uma aplicação *web* ou *software* que permite a interação direta do usuário. Em outras palavras, o *front-end* representa a camada visual da aplicação, incluindo as interfaces gráficas utilizadas pelos usuários. Essa é a parte do sistema que apresenta as informações e permite que os usuários realizem ações por meio de elementos visuais, como botões, menus, formulários e outros componentes interativos.

2.3.2 *Back-end*

O *back-end* é a parte de um *software* ou aplicação *web* responsável pelo processamento e persistência de dados, bem como a implementação da lógica de negócio. Ao contrário do *front-end*, o *back-end* não é acessado diretamente pelo usuário.

Enquanto o *front-end* lida com a interface visual e a interação do usuário, o *back-end* trabalha por baixo dos panos, lidando com a segurança dos dados, gerenciamento de requisições e manipulação e armazenamento das informações. É no *back-end* que a lógica de negócio é desenvolvida e onde ocorrem operações que garantem o devido funcionamento da aplicação.

2.3.3 *ReactJS*

O *React* é uma biblioteca de código aberto em *JavaScript* para o desenvolvimento de interfaces de usuário no *front-end*. Foi criada pelo *Facebook* com o objetivo de facilitar a construção de interfaces *web* altamente intuitivas. Sua abordagem baseada em componentes contribui significativamente para a modularização do código. Além disso, o *React* se destaca pelo seu desempenho excepcional e curva de aprendizado acessível.

2.3.4 *NodeJS*

Segundo a documentação oficial do Node.js (Node.js Foundation, s.d.), o Node.js é um ambiente de execução de código aberto projetado para a linguagem JavaScript, permitindo que o código JavaScript seja executado fora do navegador. Ele utiliza o motor V8 do Google Chrome para essa finalidade. Essa capacidade de execução fora do navegador torna possível o uso do JavaScript em servidores. Com o Node.js, é viável criar aplicativos back-end escaláveis e de alta performance, aproveitando as características do JavaScript e a eficiência do ambiente de execução proporcionada pelo Node.js.

2.3.5 *PostgreSQL*

O *PostgreSQL* é um sistema de gerenciamento de banco de dados (SGBD) de código aberto. SGBD é um *software* que permite a criação, manipulação e administração de banco de dados, que são coleções organizadas de dados inter-relacionados. Ele fornece uma interface para os usuários e aplicativos interagirem com os dados armazenados de forma eficiente. Amplamente utilizado no mercado, o *PostgreSQL* é reconhecido por sua robustez e confiabilidade.

2.3.6 *Typescript*

De acordo com a documentação oficial do *TypeScript* (Microsoft Corporation, 2024), podemos definir o *TypeScript* como uma linguagem de programação de código aberto desenvolvida pela *Microsoft*. Ela se apresenta como uma extensão do *JavaScript*, adicionando tipagem estática opcional e outros recursos avançados. Com o *TypeScript*, torna-se possível escrever código *JavaScript* de forma mais segura e escalável, detectando erros de tipo durante o desenvolvimento e oferecendo recursos avançados, como interfaces, enumerações, tipos genéricos e suporte para classes e módulos.

O principal diferencial do *TypeScript* é sua capacidade de aumentar a produtividade do desenvolvedor, fornecendo um sistema de tipagem robusto que ajuda a detectar e prevenir erros comuns durante o desenvolvimento. Além disso, o *TypeScript* é transcompilado para *JavaScript* padrão, o que significa que o código *TypeScript* pode ser executado em qualquer navegador ou ambiente de suporte *JavaScript*, como o *NodeJS*, por exemplo.

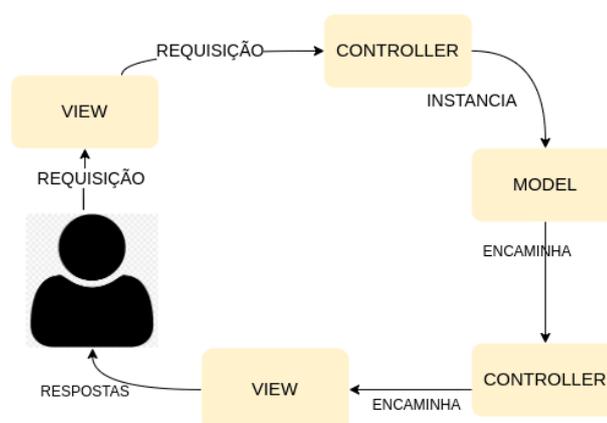
2.3.7 Prisma

O *Prisma ORM (Object-Relational Mapping)* é uma ferramenta de mapeamento objeto-relacional cuja principal função é abstrair a escrita de código *SQL* e facilitar a comunicação entre o back-end e o banco de dados. Basicamente, um mapeador objeto-relacional fornece funções para executar consultas, inserções, atualizações e exclusões de registros, sem que seja necessário escrever código *SQL* diretamente.

2.3.8 Arquitetura

A arquitetura adotada para o desenvolvimento do TCC *Easier* é a MVC (*Model-View-Controller*), elaborada por Trygve Reenskaug na década de 1970 e descrita de maneira detalhada por Steve Burbeck (1992) em seu trabalho intitulado "*Applications Programming in Smalltalk-80: How to use Model-View-Controller*". Nessa arquitetura, a aplicação é dividida em três camadas distintas. A primeira camada é o *Modelo*, que contém as classes responsáveis por representar os dados e implementar a lógica de negócio do sistema. Em seguida, temos a camada de *Visualização*, composta pelas interfaces gráficas. Essa camada pode ser entendida como o *front-end* da aplicação. Por fim, temos a camada de *Controle*, que assume a responsabilidade de receber as requisições da camada de *Visualização* e interagir com a camada de *Modelo*.

Figura 1 – Padrão de Arquitetura MVC



Fonte: Silva (2023)

3 MODELAGEM TCC EASIER

Neste capítulo, o aplicativo proposto é detalhado. Serão apresentados os requisitos funcionais e não funcionais identificados, bem como o Diagrama de Casos de Uso e o modelo Entidade-Relacionamento do banco de dados. Para garantir a funcionalidade da camada de Modelo do sistema, foi projetado e criado um banco de dados adequado utilizando PostgreSQL e realizando interações por meio do Prisma ORM. O principal objetivo deste capítulo é fornecer uma visão detalhada da camada de modelo da aplicação, além de apresentar as telas com as quais o usuário interage com o sistema e visualiza os dados fornecidos pela camada de modelo.

3.1 REQUISITOS FUNCIONAIS

RF-01 - Criar conta (Aluno): O sistema deve persistir os seguintes dados do aluno: nome, *e-mail*, curso, matrícula, senha e data de nascimento.

RF-02 - Fazer *Login* (Aluno): O sistema deve permitir que o aluno faça login, caso o *e-mail* e senha fornecidos sejam correspondentes aos de um aluno já cadastrado.

RF-03 - Listagem de orientadores (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir a exibição de uma lista de orientadores disponíveis.

RF-04 - Requisitar orientação (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir que o aluno realize um pedido de orientação, caso ele ainda não possua um orientador.

RF-05 - Visualizar reunião (Aluno): Uma vez que o aluno esteja logado e possua um orientador e uma reunião marcada, o sistema deve exibir os dados da reunião, como local, horário e data em que a reunião acontecerá.

RF-06 - Cadastro de tema do TCC (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir que o aluno cadastre um tema de TCC. Para isso ele deve informar um título e uma descrição do trabalho.

RF-07 - Alterar tema do TCC (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir que o aluno altere o tema do seu TCC. Para isso ele deve informar um título e/ou descrição.

RF-08 - Realizar *log out* do sistema (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir que o aluno faça *log out*.

RF-09 - Alterar senha (Aluno): Uma vez que o aluno esteja logado, o sistema deve permitir que o aluno altere sua senha.

RF-10 - Visualizar atividades (Aluno): Uma vez que o aluno esteja logado, possua um orientador e pelo menos uma atividade atribuída, o sistema deve permitir que o aluno visualize as atividades que lhe foram atribuídas pelo orientador.

RF-11 - Marcar atividade como concluída (Aluno): Uma vez que o aluno esteja logado, possua um orientador e pelo menos uma atividade atribuída. O sistema deve permitir que o aluno marque uma atividade como concluída.

RF-12 - Criar conta (Orientador): O sistema deve persistir os seguintes dados do orientador: nome, *e-mail*, descrição, senha e data de nascimento.

RF-13 - Fazer login (Orientador): O sistema deve permitir que o orientador faça *login*, caso o *e-mail* e senha fornecidos sejam correspondentes aos de um orientador já cadastrado.

RF-14 - Listar requisições de orientação (Orientador): Uma vez que o orientador esteja logado. O sistema deve listar as requisições de orientação recebidas. Nessas requisições o orientador deve visualizar o tema de TCC do aluno, caso ele possua um cadastrado.

RF-15 - Aceitar requisições de orientação (Orientador): Uma vez que o orientador esteja logado. O sistema deve permitir que o orientador aceite requisições de orientação.

RF-16 - Recusar requisições de orientação (Orientador): Uma vez que o orientador esteja logado. O sistema deve permitir que o orientador recuse requisições de orientação.

RF-17 - Listar orientandos (Orientador): Uma vez que o orientador esteja logado. O sistema deve listar todos os seus orientandos, caso ele possua algum.

RF-18 - Marcar reunião (Orientador): Uma vez que o orientador esteja logado e possua pelo menos um orientando. O sistema deve permitir que ele marque uma reunião. Para isso ele deve informar o aluno que irá participar da reunião, o horário e o local.

RF-19 - Editar reunião (Orientador): Uma vez que o orientador esteja logado e possua pelo menos um orientando e uma reunião cadastrada. O sistema deve permitir que ele altere os dados da reunião, como local, horário e data.

RF-20 - Excluir reunião (Orientador): Uma vez que o orientador esteja logado e possua pelo menos um orientando e uma reunião cadastrada. O sistema deve permitir que ele remova reuniões.

RF-21 - Atribuir atividades (Orientador): Uma vez que o orientador esteja

logado e possua pelo menos um orientando. O sistema deve permitir que ele atribua atividades aos seus orientandos. Para isso ele deve passar o nome da tarefa, descrição e data de entrega.

RF-22 - Listar atividades (Orientador): Uma vez que o orientador esteja logado e possua pelo menos um orientando e uma atividade atribuída. O sistema deve permitir que ele visualize uma lista de atividades atribuídas, bem como, o progresso de conclusão de cada aluno.

RF-23 - Excluir atividades (Orientador): Uma vez que o orientador esteja logado e possua pelo menos um orientando e uma atividade cadastrada. O sistema deve permitir que ele remova as atividades.

3.2 REQUISITOS NÃO FUNCIONAIS

RNF - 01 - Qualidade de código: O sistema deve possuir um código legível, de fácil manutenção e que siga os princípios de código limpo.

RNF - 02 - Disponibilidade: O sistema deve estar em funcionamento 24 horas por dia, 7 dias por semana, exceto em casos em que seja necessário interromper o sistema para integração de determinada funcionalidade ou para realizar manutenção programada. Nestes casos, os usuários devem ser informados previamente.

RNF - 03 - Segurança: Todas as senhas de usuário devem ser armazenadas no banco de dados de forma criptografada. Além disso, todas as rotas devem solicitar um token de *login* válido.

RNF - 04 - Usabilidade: O sistema deve possuir interfaces intuitivas e funcionais para proporcionar uma experiência de usuário aprimorada.

RNF - 05 - Desempenho: As respostas do sistema devem ser entregues aos usuários em um tempo máximo de 1 segundo, garantindo uma resposta rápida.

3.3 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso da Figura 2 apresenta visualmente as diferentes ações que os usuários do sistema podem realizar. Ele tem como objetivo fornecer uma visão geral da interação dos usuários com o sistema e das principais funcionalidades disponíveis para cada um deles.

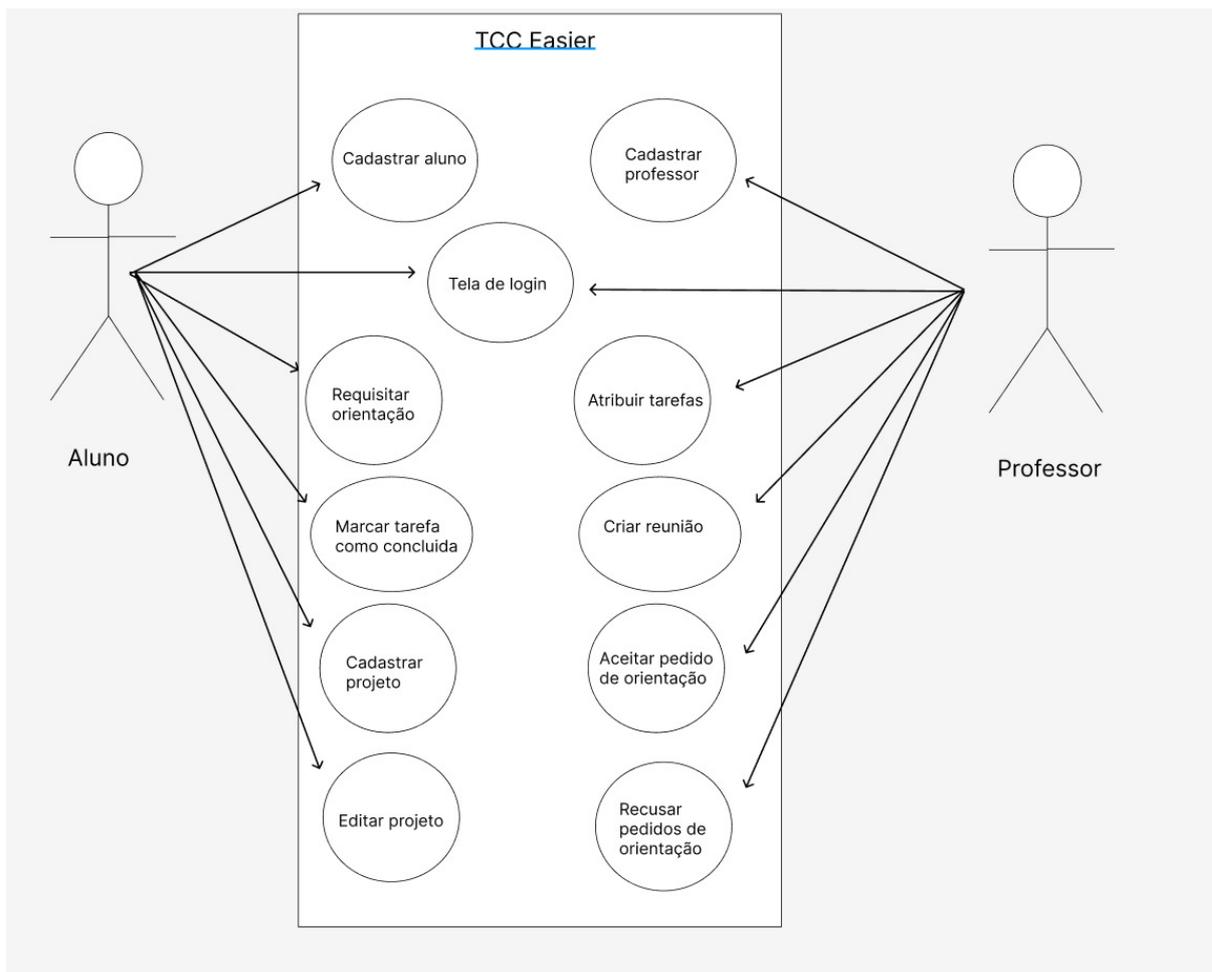


Figura 2 – Diagrama de caso de uso

3.4 MODELO ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento da Figura 3 tem como objetivo demonstrar de maneira visual os atributos e os relacionamentos entre as entidades que compõem o banco de dados do sistema.

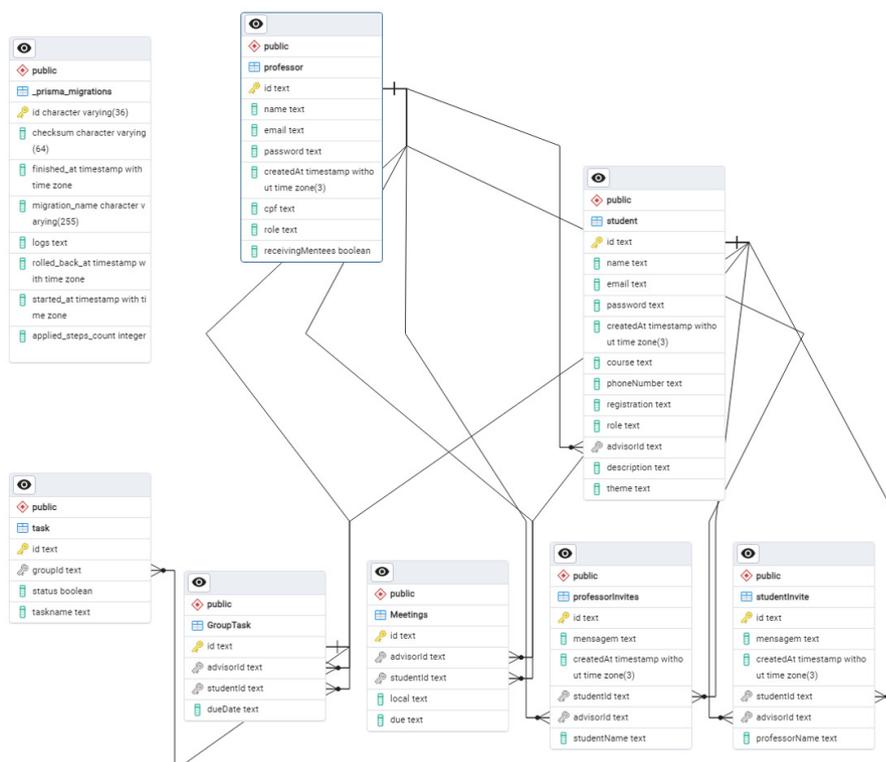


Figura 3 – Modelo Entidade-Relacionamento

3.5 TELAS

A seguir, serão exibidas as telas com as quais o usuário interage com o sistema, acompanhadas de uma breve descrição de suas principais funcionalidades. As telas do sistema foram projetadas para serem intuitivas e simples, de modo que o usuário não encontre barreiras ao tentar identificar uma funcionalidade.

3.5.1 Tela de *login* do aluno e professor

A tela apresentada na Figura 4 será comum tanto para alunos quanto para professores. Nela, estes atores poderão fazer *login* no sistema utilizando o *e-mail* e a senha previamente cadastrados.

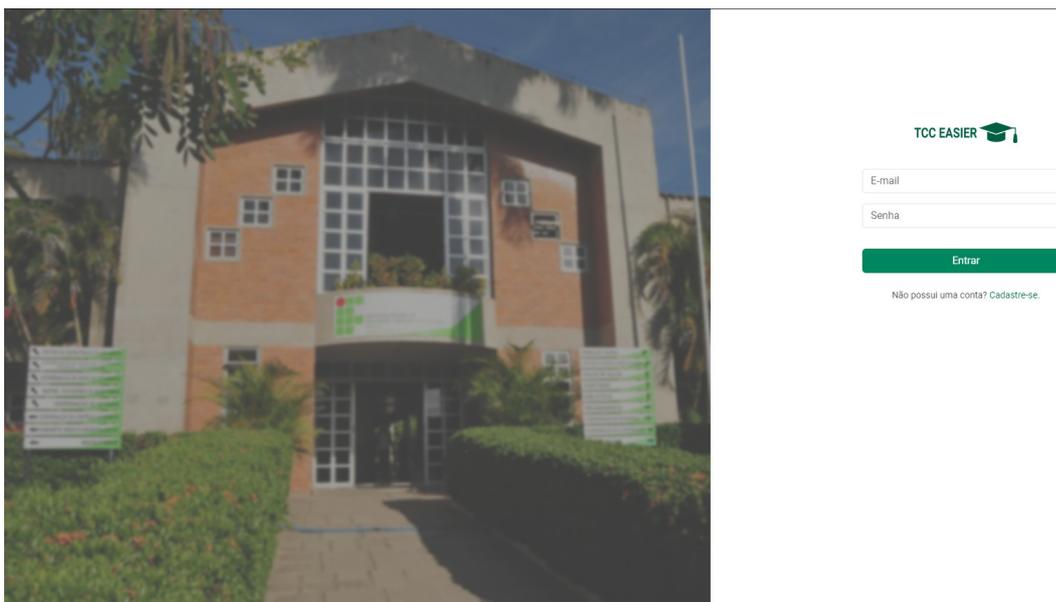


Figura 4 – Tela de login

3.5.2 Tela de início do aluno

Na tela inicial do aluno, Figura 5, serão exibidas as principais informações relevantes para ele, como os dados do seu projeto cadastrado, informações sobre a próxima reunião de orientação e a lista de tarefas atribuídas pelo orientador. Essa tela fornecerá um panorama das atividades e atualizações importantes relacionadas ao TCC do aluno.

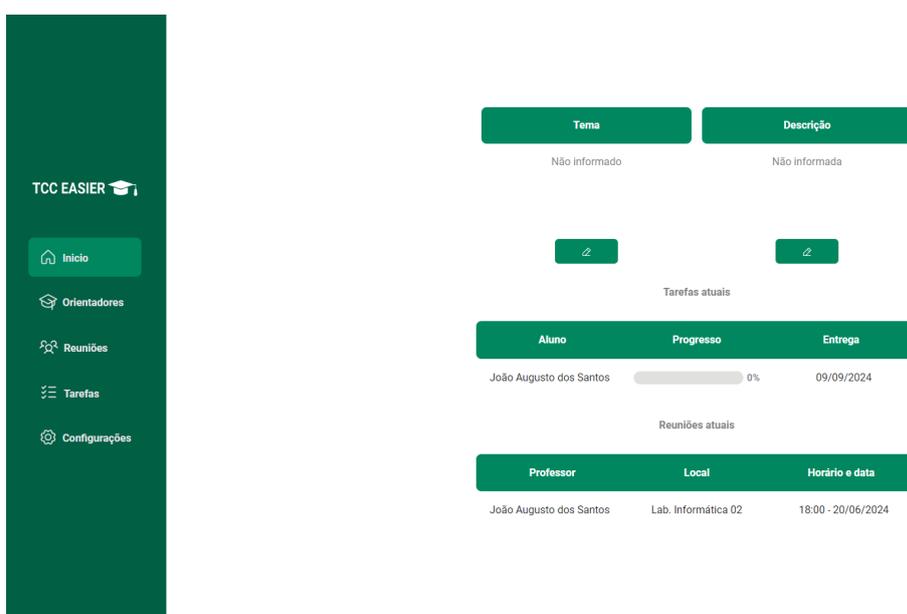


Figura 5 – Tela de início do aluno

3.5.3 Tela de orientadores do aluno

Na tela de orientadores, Figura 6, o aluno poderá buscar por orientadores e solicitar orientação. Caso já tenha um orientador, as informações de contato do orientador serão exibidas.



Figura 6 – Tela de orientadores do aluno

3.5.4 Tela de reuniões do aluno

Na tela de reuniões do aluno, Figura 7, será exibido o histórico completo das reuniões. Nessa tela, o aluno terá a visualização das reuniões que já ocorreram, assim como as reuniões agendadas para o futuro. Isso permitirá ao aluno ter um registro completo das suas interações com o orientador e facilitará o planejamento e o acompanhamento das próximas reuniões.

| Reuniões atuais | | |
|---------------------------|---------------------|--------------------|
| Professor | Local | Horário e data |
| João Augusto dos Santos | Lab. Informática 02 | 18:00 - 20/06/2024 |
| Reuniões passadas | | |
| Aluno | Local | Horário e data |
| Marcos augusto dos santos | Biblioteca | 17:20 - 27/02/2024 |

Figura 7 – Tela de reuniões do aluno

3.5.5 Tela de tarefas do aluno

Na tela de tarefas do aluno, ilustrada na Figura 8, será exibido o histórico das tarefas. Nessa tela, o aluno poderá acompanhar o seu progresso, verificar a data de entrega de cada tarefa e ler a descrição das atividades em detalhes. Essa funcionalidade permitirá ao aluno manter o controle e a organização das suas tarefas, facilitando o acompanhamento do desenvolvimento do seu TCC.

| Tarefas atuais | | |
|-------------------------|-----------------------------------|------------|
| Aluno | Progresso | Entrega |
| João Augusto dos Santos | <div style="width: 0%;"></div> 0% | 09/09/2024 |
| Tarefas anteriores | | |
| Aluno | Progresso | Entrega |
| João Augusto dos Santos | <div style="width: 0%;"></div> 0% | 08/02/2024 |

Figura 8 – Tela de tarefas do aluno

3.5.6 Tela de início do professor

Na tela inicial do professor, Figura 9, serão exibidas as principais informações de interesse, como o progresso das atividades de cada aluno, a lista de reuniões marcadas, a lista de orientandos e as requisições de orientação. Essa tela fornecerá uma visão geral das atividades e interações relacionadas ao acompanhamento dos alunos e suas respectivas demandas, facilitando a organização e o gerenciamento por parte do professor.

Convites
Você possui 1 pedido(s) de orientação.
[Visualizar](#)

Tarefas atuais

| Aluno | Progresso | Entrega |
|---------------------------|-----------|------------|
| Marcos augusto dos santos | 0% | 09/09/2024 |

Reuniões atuais

| Aluno | Local | Horário e data |
|---------------------------|---------------------|--------------------|
| Marcos augusto dos santos | Lab. Informática 02 | 18:00 - 20/06/2024 |

Lista de orientandos

| Nome | E-mail | Telefone | Ação |
|---------------------------|-----------------|------------------|-------------------------------------|
| Marcos augusto dos santos | aluno@gmail.com | (88) 9 3923-2343 | Visualizar detalhes |

Figura 9 – Tela de início do professor

3.5.7 Tela de tarefas do professor

Na tela de tarefas do professor, Figura 10, será exibido um histórico das tarefas atribuídas. Além disso, será mostrado o progresso de conclusão de cada aluno, incluindo aquelas que não foram entregues na data estipulada. O professor também terá a capacidade de atribuir novas tarefas diretamente nessa tela, facilitando o gerenciamento e acompanhamento das atividades dos alunos. Isso proporcionará ao professor uma visão abrangente do progresso e desempenho de cada aluno em relação às tarefas atribuídas.

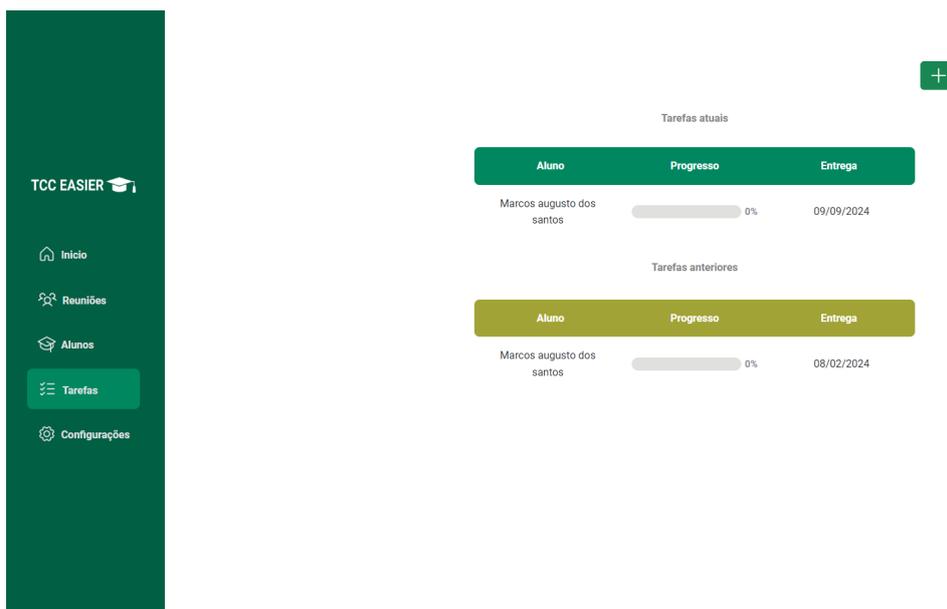


Figura 10 – Tela de tarefas do professor

3.5.8 Tela de reuniões do professor

Na tela de reuniões do professor, ilustrada na Figura 11, será exibido o histórico completo de reuniões. O professor poderá visualizar tanto as reuniões que já ocorreram quanto as que estão agendadas para o futuro. Além disso, o professor terá a opção de marcar novas reuniões diretamente nessa tela, facilitando o agendamento e organização de encontros com os alunos. Essa funcionalidade proporcionará ao professor uma visão abrangente de suas interações e compromissos de orientação.



Figura 11 – Tela de reuniões do professor

4 CONSIDERAÇÕES FINAIS

Concluindo este estudo, fica evidente a importância do projeto TCC *Easier* como uma ferramenta vital para auxiliar os alunos do IFPB, *Campus* Cajazeiras, no processo de desenvolvimento de seus Trabalhos de Conclusão de Curso. A concepção do sistema, considerando princípios de escalabilidade e manutenibilidade, reflete não apenas a necessidade imediata de suporte aos alunos do curso de Análise e Desenvolvimento de Sistemas, mas também uma visão de expansão para futuros cursos.

O direcionamento inicial para o curso de Tecnologia em Análise e Desenvolvimento de Sistemas demonstra um compromisso com a qualidade e o atendimento aos requisitos específicos da aplicação. As tecnologias escolhidas para o desenvolvimento, com sua curva de aprendizado reduzida, contribuem para a longevidade do sistema, garantindo sua utilidade e relevância ao longo do tempo.

Caso haja a incorporação do TCC *Easier* pela instituição, espera-se que proporcione um crescimento na qualidade dos Trabalhos de Conclusão de Curso. A contribuição potencial deste sistema para a comunidade acadêmica é significativa, oferecendo suporte essencial aos alunos e promovendo uma experiência mais eficiente e satisfatória durante todo o processo de desenvolvimento do TCC.

Em resumo, o TCC *Easier* representa não apenas uma solução prática para os desafios enfrentados pelos alunos, mas também um exemplo de como os conhecimentos teóricos e práticos adquiridos durante todo o curso podem ser utilizados em benefício da comunidade acadêmica local.

REFERÊNCIAS

MARTIN, Robert C. Código limpo: habilidades práticas do Agile Software. Porto Alegre: Bookman, 2008.

SILVA, E. M. Principais conceitos do padrão MVC (Model-View- Controller). 2023. Disponível em: <<https://dev.to/elisangelamsilva/principais-conceitos-do-padrao-mvc-model-view-controller-1o5a>>.

GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John. Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos. Porto Alegre: Bookman, 2000.

ARRUDA, Lucas Brito. Sistema de gestão de trabalhos de final de curso. 2010. Monografia - Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, Belo Horizonte.

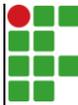
CABRAL, Leonardo Lima. TCC online: Sistema de gerenciamento de trabalhos de conclusão de curso. 2017. Monografia – Universidade Federal Fluminense, Niterói.

Burbeck, S. (1992). Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC). Palo Alto Research Center.

PEREIRA, A. A. C.; SILVA, M. L. R. O trabalho de conclusão de curso: constructo epistemológico no currículo formação, valor e importância. Laboratório de Pesquisa Multimeios, v.1, n.1, 2012.

TypeScript (Microsoft). (s.d.). Handbook Recuperado em 04 de junho de 2024, de <<https://www.typescriptlang.org/docs/handbook/intro.html>>

Node.js Foundation. (s.d.). About Node.js. Recuperado em 04 de junho de 2024, de <<https://nodejs.org/en/about>>

| | |
|---|---|
|  | INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA |
| | Campus Cajazeiras - Código INEP: 25008978 |
| | 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 Ostensivo (Público)

Entrega de trabalho de conclusão do curso

| | |
|-----------------------------|---|
| Assunto: | Entrega de trabalho de conclusão do curso |
| Assinado por: | José Silva |
| Tipo do Documento: | Anexo |
| Situação: | Finalizado |
| Nível de Acesso: | Ostensivo (Público) |
| Tipo do Conferência: | Cópia Simples |

Documento assinado eletronicamente por:

- **José Farley Sousa da Silva, ALUNO (202012010019) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 13/08/2024 15:28:04.

Este documento foi armazenado no SUAP em 13/08/2024. 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: 1217737

Código de Autenticação: 352943542c

