



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LEONARDO VEIGA DE MEDEIROS PORFÍRIO

**TENHO QUE CONCLUIR O CURSO:
SISTEMA DE GESTÃO DE TEMAS DE TCC**

CAMPINA GRANDE - PB

2023

LEONARDO VEIGA DE MEDEIROS PORFÍRIO

**TENHO QUE CONCLUIR O CURSO:
SISTEMA DE GESTÃO DE TEMAS DE TCC**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

Orientador : Fábio Jorge Almeida Morais

CAMPINA GRANDE - PB

2023

LEONARDO VEIGA DE MEDEIROS PORFÍRIO

**TENHO QUE CONCLUIR O CURSO:
SISTEMA DE GESTÃO DE TEMAS DE TCC**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.**

BANCA EXAMINADORA:

Fábio Jorge Almeida Morais

Orientador – UASC/CEEI/UFCG

Jorge Cesar Abrantes de Figueiredo

Examinador – UASC/CEEI/UFCG

Francisco Vilar Brasileiro

Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 17 de NOVEMBRO de 2023.

CAMPINA GRANDE - PB

RESUMO

O Trabalho de Conclusão de Curso (TCC) é um trabalho acadêmico de caráter obrigatório e atualmente instrumento de avaliação final do curso de Ciência da Computação na Universidade Federal de Campina Grande. A escolha do tema é algo que muitas vezes intimida o aluno, pois não sabe ao certo sobre o que vai fazer, o que gosta e qual professor irá aceitar ser orientador do mesmo.

Não obstante os professores também podem propor algum tema e procurar alunos que se interessem em realizar esse trabalho. Dessa forma é importante que haja uma solução que facilite esse processo, tanto para o aluno quanto para o professor. Este trabalho tem como objetivo o desenvolvimento de uma aplicação web para expor temas ou ideias de TCC e conectar alunos e professores com interesses mútuos, visando melhorar a experiência do aluno na escolha do tema e do professor orientador. Os resultados obtidos demonstram que os usuários ficaram satisfeitos com a usabilidade do sistema, mas ainda há pontos de melhoria.

TENHO QUE CONCLUIR O CURSO: THESIS TOPIC MANAGEMENT SYSTEM

ABSTRACT

The Course Conclusion Paper (TCC) is a compulsory piece of academic work and currently the final assessment instrument for the Computer Science course at the Federal University of Campina Grande. The choice of topic is something that often intimidates students, as they don't know exactly what they're going to do, what they like and which teacher will agree to advise them. However, teachers can also propose a topic and look for students who are interested in doing the work. It is therefore important that there is a solution that facilitates this process for both the student and the teacher. The aim of this work is to develop a web application to showcase thesis topics or ideas and connect students and teachers with mutual interests, with the aim of improving the experience of the student when choosing a topic and of the supervising teacher. The results obtained show that users were satisfied with the usability of the system, but there are still areas for improvement.

Tenho que Concluir o Curso: Sistema de gestão de temas de TCC

Leonardo Veiga de Medeiros Porfírio
Universidade Federal de Campina Grande
Campina Grande, Paraíba
leonardo.porfirio@ccc.ufcg.edu.br

Fabio Jorge Almeida Morais
Universidade Federal de Campina Grande
Campina Grande, Paraíba
fabio@computacao.ufcg.edu.br

RESUMO

O Trabalho de Conclusão de Curso (TCC) é um trabalho acadêmico de caráter obrigatório e atualmente instrumento de avaliação final do curso de Ciência da Computação na Universidade Federal de Campina Grande. A escolha do tema é algo que muitas vezes intimida o aluno, pois não sabe ao certo sobre o que vai fazer, o que gosta e qual professor irá aceitar ser orientador do mesmo. Não obstante os professores também podem propor algum tema e procurar alunos que se interessem em realizar esse trabalho. Dessa forma é importante que haja uma solução que facilite esse processo, tanto para o aluno quanto para o professor. Este trabalho tem como objetivo o desenvolvimento de uma aplicação web para expor temas ou ideias de TCC e conectar alunos e professores com interesses mútuos, visando melhorar a experiência do aluno na escolha do tema e do professor orientador. Os resultados obtidos demonstram que os usuários ficaram satisfeitos com a usabilidade do sistema, mas ainda há pontos de melhoria.

Palavras-Chave

TCC, Trabalho de Conclusão de Curso, *website*, código aberto

Respositórios

<https://github.com/leleoveiga/tcc-frontend>

<https://github.com/leleoveiga/tcc-backend>

1 INTRODUÇÃO

O desenvolvimento de um TCC é um passo importante para a formação de um aluno, nesse momento ele pode contribuir com algo valioso, explorar e até resolver problemas reais e relevantes. É muito importante que o aluno faça algo sobre o que ele se interesse e goste, e melhor ainda, com um professor que se interesse também pelo tema, resultando em um trabalho relevante tanto para o aluno, o professor e para o mundo acadêmico.

O aluno que está prestes a fazer o seu TCC quer demonstrar sua capacidade de pesquisa, desenvolvimento e conhecimento adquirido ao longo do curso. No entanto, há alguns desafios à sua frente como a escolha do tema e do professor orientador. A indecisão e a busca por um tema que se alinhe com seus interesses e paixões muitas vezes se transformam em uma ansiedade complexa, devido ao prazo, pressão familiar e ao objetivo de se formar. Isso resulta muitas vezes em escolhas de

tópicos e professores que não são compatíveis com o interesse do aluno. Dessa forma há uma procura entre estudantes e orientadores que possuam perfis com interesses e habilidades que se alinhem.

Sabendo disso, como opção para os problemas descritos anteriormente, foi desenvolvido o **Tenho que Concluir o Curso**, uma aplicação web que tem como objetivo viabilizar a procura entre alunos e professores que correspondam baseado em interesses e habilidades, exponham suas ideias de temas para o TCC e tenham um espaço para dialogar e discutir sobre o mesmo. A vantagem desta aplicação é que não é comum encontrar uma solução bem estabelecida para este problema, mostrando que há uma necessidade para a criação de uma no contexto no qual foi pensado, podendo ajudar muitos estudantes e impactando positivamente no seu futuro acadêmico e profissional. Além disso, a aplicação possui código aberto, possibilitando que instituições modifiquem a aplicação para seu uso próprio, além de ser possível evoluir a própria aplicação de acordo com o interesse dos alunos e professores do curso de Ciência da Computação da Universidade Federal de Campina Grande (UFCG), contexto no qual o projeto foi pensado e desenvolvido.

Para validar a aplicação, conduziu-se um teste de usabilidade, no qual os usuários preencheram um questionário após completarem uma tarefa proposta para o uso de funcionalidades do Tenho que Concluir o Curso. Utilizou-se o CSUQ (*The Computer System Usability Questionnaire*) [2] para o formato do questionário, comumente utilizado para avaliar a satisfação dos usuários ao usarem softwares e produtos ao final de seu desenvolvimento. Em sua maioria, os resultados apontaram para a facilidade de uso da aplicação, embora tenha sido observado que algumas funcionalidades do sistema receberam avaliações mistas, com respostas tanto positivas quanto negativas, sugerindo a necessidade de melhorias em áreas específicas.

2 DOMÍNIO DO PROBLEMA

Quando o aluno está prestes a fazer o seu TCC, é muito comum não ter um tema em mente, nem o conhecimento de professores que tenham os mesmos interesses do aluno para chamar alguns deles para o orientar. Outro problema é o aluno ter um tema em mente e não saber qual professor que tenha interesse e que esteja disponível para o orientar.

É comum ter casos de alunos que não sabem sobre o que vão escrever e pedem ajuda para os professores, muitas vezes mandando e-mails para eles. Ele tem que procurar pelos professores no qual tem interesse, coletar o e-mail deles e mandar

para cada um uma mensagem solicitando sugestões de temas. Também é uma tarefa exaustiva para o professor, que muitas vezes recebe e-mails desse tipo e tem que responder de um em um se está disponível para orientar o aluno ou se tem algum tema em mente para o aluno, e ter a sorte de que o aluno se interesse pelo tema dele. Nota-se que não existe um método padronizado para esse processo de busca por temas e professores, e por causa disso é muitas vezes ineficiente.

Quando o aluno não escolhe um tema no qual ele tenha interesse, ou com um professor que não se interessa muito pelo assunto, pode resultar numa frustração para o aluno, tornando a experiência de desenvolver seu TCC ruim. Quando na verdade deveria ser um momento muito importante para a sua carreira acadêmica.

3 SOLUÇÃO

3.1 Descrição

Neste trabalho é apresentado o Tenho que Concluir o Curso¹, que se trata de uma aplicação web, podendo ser usada tanto em desktops quanto em dispositivos móveis. A aplicação tem como objetivo ajudar no processo de encontrar um tema e um orientador, fornecendo uma plataforma no qual possibilita a união de alunos e professores com áreas de interesse em comum, a publicação de ideias e temas de TCC, e um ambiente destinado ao diálogo e debate sobre os mesmos.

O projeto, inicialmente, tem como público-alvo o curso de Ciência da Computação da UFCG. No entanto, em virtude de sua natureza como aplicação de código aberto, é possível que outros cursos e instituições a utilizem como base para criarem suas próprias versões. Ademais, por conta disso, pretende-se, com o tempo, engajar a comunidade do curso de Ciência da Computação a contribuir com o desenvolvimento de novas funcionalidades e melhorias no sistema de modo que o sistema atenda aos seus interesses.

3.2 Funcionalidades

O sistema e todas as suas funcionalidades são utilizadas através do *website* da aplicação, mas para isso é necessário que o usuário se autentique utilizando seu e-mail acadêmico, que é utilizado para identificar se o usuário é um aluno ou professor. O sistema de autenticação escolhido foi o Firebase Authentication [6], que será descrito em detalhes na seção 4. Esta tecnologia torna o processo de autenticação seguro e possibilita que a aplicação saiba identificar se o usuário é um aluno ou professor e, em seguida, autorizá-lo a acessar o sistema. Este é um passo crucial pois as funcionalidades se adaptam ao tipo de usuário logado.

A aplicação possui uma barra lateral, podendo ser vista na Figura 1 que permite o acesso a todas as funcionalidades da aplicação, são esses a listagem e busca de temas, o cadastro dos mesmos, visualização do próprio perfil, a busca de usuários e o logout do sistema. O acesso para essa barra lateral só é liberado após o usuário se autenticar no sistema.

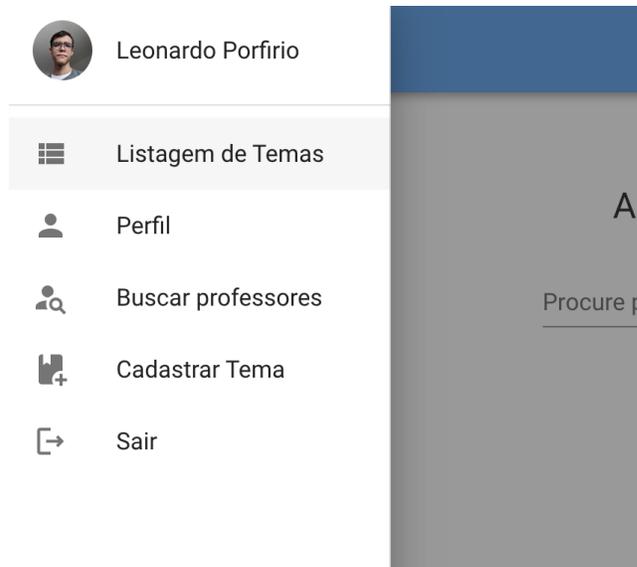


Figura 1: Menu lateral, mostrando algumas das funcionalidades do sistema

A página inicial da aplicação, Figura 2, apresenta um breve resumo sobre sua utilidade e um botão para o usuário realizar a autenticação. Após realizá-la, ele é redirecionado para a listagem de temas de TCC.



Figura 2: Tela inicial da aplicação

Na listagem de temas de TCC, Figura 3, o usuário não pode ver temas publicados pelo mesmo tipo de usuário. Caso seja um aluno, o servidor irá detectar através da *request* e irá retornar apenas os que foram criados por professores, e vice-versa. Além disso, há uma barra de pesquisa onde é possível filtrar os temas por palavras-chave e um botão para realizar o cadastro dos mesmos. O item da listagem consiste no título e no início do corpo do mesmo.

¹ <https://tcc-ufcg.live/>

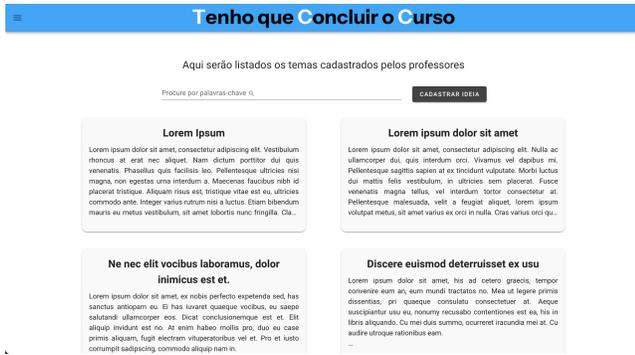


Figura 3: Página inicial do sistema, listando os temas, e possibilitando a busca e o cadastro dos mesmos

É possível visualizar os detalhes do tema clicando no item. A página no qual o usuário é redirecionado, Figura 4, contém as seguintes funcionalidades:

- Detalhes sobre o autor do tema, como a foto, nome e e-mail;
- O título e o corpo do tema;
- Um botão de excluir o tema, caso o usuário seja o autor do mesmo;
- Uma seção de comentários, onde é possível interagir de forma privada com o autor através de mensagens, enquanto na visão do autor na mesma seção é possível ver e responder individualmente cada usuário que interagiu;
- Além disso, cada mensagem é um botão no qual redireciona o usuário para a página de visualização do usuário.



Figura 4: Página de exibição do tema, na parte inferior também é possível visualizar a seção de comentários

A página do usuário, Figura 5, é composta por uma seção que expõe informações como foto, nome, e-mail e interesses em formato de tags do usuário, e uma listagem de temas postados por ele. Caso o usuário tenha acessado seu próprio perfil

ele pode selecionar tags que correspondem com seus interesses e habilidades.



Figura 5: Página de exibição do usuário, exibindo as tags ativadas por ele e a listagem de temas previamente criados.

Através da tela inicial ou do menu lateral é possível acessar a funcionalidade de cadastrar temas, como pode ser visto na Figura 6, nela é necessário apenas preencher o título e o corpo.



Figura 6: Página de cadastro do tema

Por fim, a funcionalidade de buscar usuários, Figura 7, que é semelhante à listagem de temas, o aluno só consegue buscar por professores e vice-versa. A página é apresentada com uma listagem dos usuários, podendo ser filtrada buscando por nomes através da barra de pesquisa, ou pelo filtro de tags. Clicando em algum usuário da listagem redireciona para a página de visualização de usuário, detalhada anteriormente.

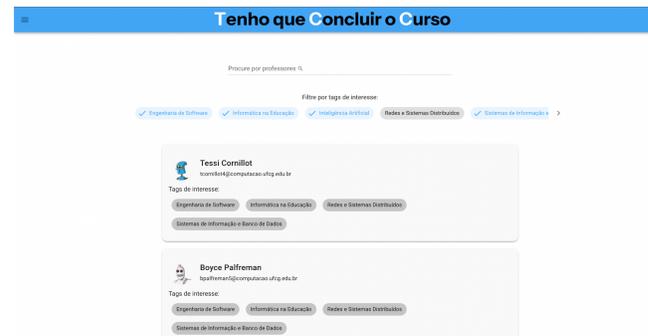


Figura 7: Página de listagem de usuários, mostrando alguns filtros já ativados para mostrar apenas usuários com aquelas tags

4 ARQUITETURA

O Tenho que Concluir o Curso tem duas partes principais: o *front-end* (cliente) e o *back-end* (servidor). O cliente é a parte que gera a interface gráfica para o usuário, e através dele o usuário interage com o sistema, recebendo e enviando dados para o servidor. Já o servidor é responsável por armazenar os dados em um banco de dados e também pela lógica de autenticação, autorização dos usuários e de como os dados serão armazenados e retornados para o cliente. Além disso, os dois se comunicam com outro serviço que é o Firebase Authentication, um serviço fornecido pelo Google Firebase [1], que oferece recursos de autenticação segura para aplicativos móveis e web. Ele simplifica a implementação de sistemas de autenticação, oferecendo métodos seguros de autenticação como a de redes sociais (por exemplo, Google e Facebook).

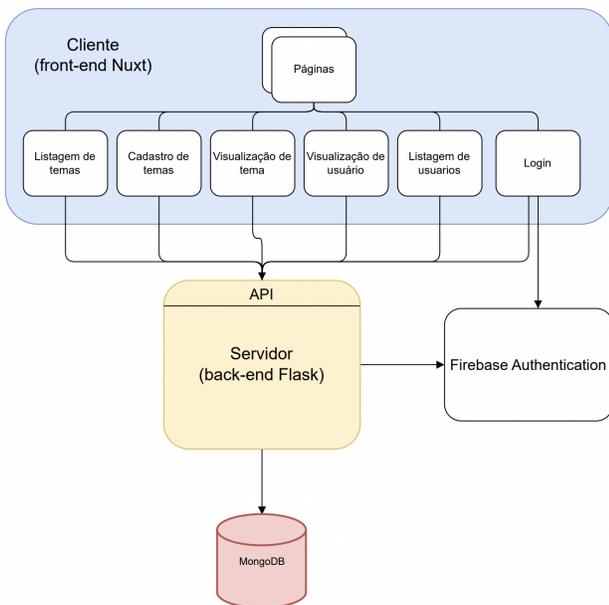


Figura 8: Diagrama de comunicação do cliente, servidor, banco de dados e Firebase Authentication

O cliente é um projeto NPM, que é um gerenciador de pacotes para a linguagem de programação JavaScript. Através dele é possível inferir versões específicas de bibliotecas utilizadas para garantir o funcionamento do sistema. Estas bibliotecas ficam disponíveis apenas naquele ambiente de desenvolvimento, fazendo com que seja necessário utilizar o NPM, também, para criar *scripts* que executam o projeto.

O servidor foi desenvolvido em Python, utilizando Poetry para a criação do ambiente virtual utilizado para gerenciar as dependências, facilitando o desenvolvimento e distribuição do software caso haja a necessidade de outros desenvolvedores participarem do projeto. Além disso, foi utilizado Docker [5] para reproduzir com facilidade o ambiente necessário para executar a aplicação, além de ter facilitado o processo de configuração do banco de dados MongoDB [3], pois é possível isolá-lo em um ambiente virtual no qual ele é instalado e executado

automaticamente, baseado nas configurações do arquivo “docker-compose.yml”.

4.1 Tecnologias do front-end

O cliente foi desenvolvido utilizando os *frameworks* Nuxt.js [11] e Vuetify.js [10]. Essas tecnologias simplificam bastante o desenvolvimento de interfaces de usuário, oferecendo diversas ferramentas, desempenho, escalabilidade e garantindo as melhores práticas.

A escolha do Nuxt.js se baseou em sua capacidade de simplificar o desenvolvimento de aplicações Vue.js [12], outra *framework* utilizada principalmente para construir interfaces de usuário interativas e reativas. Dentre essas simplificações, no contexto em que foi desenvolvido destacam-se: a criação automática de rotas, baseada no nome dos arquivos que ficam no diretório “pages”, e a integração pré-configurada com as tecnologias comumente utilizadas neste tipo de projeto, como Vuex para gerenciamento de estados e Axios para fazer requisições HTTP.

A fim de tornar o design das páginas mais coeso e economizar esforços, foi adotada a biblioteca de componentes Vuetify.js, que tem como premissa a criação rápida de aplicações Vue acessíveis seguindo as diretrizes de design de material do Google. O Vuetify.js possui uma gama de componentes, como botões, inputs e tabelas, que têm sua lógica de funcionamento integrada, não sendo necessário que o programador os construa desde o início utilizando tags HTML [8], código CSS [7] e JavaScript [9]. Além disso, a biblioteca oferece múltiplas opções de estilização que seguem o padrão de design proposto pela mesma, permitindo a flexibilização do estilo do componente base, como é possível ver na Figura 9. Também foi necessário utilizar CSS para a configuração de alguns estilos gerais da aplicação em que não era possível realizar com o Chakra UI, como a cor do plano de fundo e a fonte de texto desejada.

Para a autenticação, como mencionado anteriormente, foi utilizado o Firebase Authentication, no qual se encaixa muito bem neste contexto pois o e-mail acadêmico dos alunos e professores do curso de Ciência da Computação da UFCG faz parte de uma conta Google. Portanto, utilizar o Google como provedor de autenticação facilitou a identificação do usuário como aluno ou professor. Isso facilita a criação da lógica de autenticação da aplicação, além de ser altamente seguro.

```
1 <template>
2 <v-btn
3   v-show="shouldShowDeleteButton"
4   class="mt-10"
5   color="error"
6   fab
7   small
8   absolute
9   right
10  top
11  @click="deleteIdea()"
12 >
13 <v-icon>mdi-delete</v-icon>
14 </v-btn>
15 </template>
```

Figura 9: Exemplo de código de um componente Vue.js, utilizando a biblioteca Vuetify.js

4.2 Tecnologias do back-end

Para o servidor foi utilizado a combinação do *framework* Flask [13] e do banco de dados MongoDB executado em um container Docker. Os motivos foram semelhantes aos do cliente, essas tecnologias oferecem proporcionam desempenho adequado, simplicidade e flexibilidade na criação de APIs e rotas.

Flask proporciona simplicidade e flexibilidade no desenvolvimento de APIs e rotas, além de um desempenho apropriado para o contexto. Essa parte é o meio termo entre o cliente e o banco de dados, onde seus dados serão armazenados. É através dele que o cliente se comunicará, através de requisições HTTP. Após isso, caso necessário, o servidor se comunica com o banco de dados através do MongoEngine, uma biblioteca Python de mapeamento objeto-documento (ODM) projetada para trabalhar com o MongoDB.

Por sua vez, o MongoDB é um banco de dados NoSQL orientado a documentos, oferecendo flexibilidade no armazenamento e na consulta dos dados. O banco de dados é executado em um ambiente Docker, ou seja, não está sendo instalado diretamente no sistema operacional do servidor, mas sim em um container Docker. Essa abordagem oferece várias vantagens, incluindo a capacidade de criar ambientes de desenvolvimento, testes isolados e altamente reproduzíveis, e facilmente escaláveis.

4.3 Estrutura do front-end

Muitas vezes os projetos *front-end* não possuem uma regra bem definida sobre como estruturar o projeto. Porém, nesse ponto o Nuxt.js se destaca, pois obriga os desenvolvedores a seguirem uma estrutura pré-definida. Essa estrutura organizada no Nuxt.js contribui para a modularidade, a manutenção e a escalabilidade. A organização é dividida em diretórios, que podem ser vistos na Figura 10, e os principais serão detalhados abaixo.

4.3.1 Diretório Pages

No contexto do Nuxt.js, a estrutura de páginas é simplificada por meio do diretório "pages". Aqui, os componentes Vue.js são automaticamente mapeados para rotas da web, simplificando a adição e a modificação de páginas no aplicativo. Esta abordagem baseada em arquivos promove a expansibilidade da estrutura.

4.3.2 Layouts Personalizados

A criação de layouts personalizados é incentivada no Nuxt.js, e foi usado para a definição da estrutura geral das páginas, incluindo cabeçalhos, rodapés e barras de navegação. O diretório "layouts" abriga esses layouts, garantindo a consistência na aparência das páginas.

4.3.3 Componentes Reutilizáveis

Para maximizar a reutilização de código, o Nuxt.js promove o uso de componentes Vue.js modulares. O diretório "components" contém uma variedade de componentes personalizáveis, desde botões e formulários até cartões e outros elementos de interface de usuário.

4.3.4 Middleware

O Nuxt.js permite a definição de middleware para executar tarefas antes da renderização de páginas ou do acesso a rotas. Essa funcionalidade foi usada para implementar a lógica de autenticação e autorização.

4.3.5 Store

O diretório "store" é usado para implementar o padrão de gerenciamento de estados do Vuex, que é uma biblioteca de gerenciamento de estado para aplicativos Vue.js. O Vuex é altamente valioso quando se trata de gerenciar estados complexos em aplicativos front-end, como estados de autenticação do usuário, dados dinâmicos ou configurações globais.

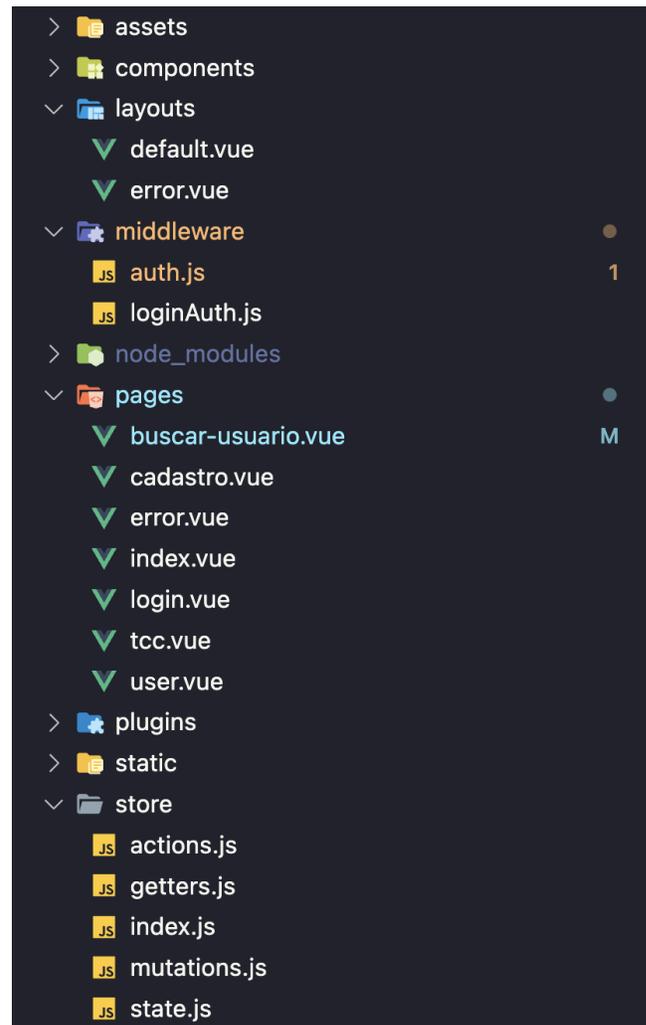


Figura 10: Organização da estrutura dos diretórios do projeto do cliente

4.4 Estrutura do back-end

No caso do *back-end*, não há regras específicas para a estruturação do projeto. Portanto, foi decidido separar os arquivos de acordo com as funções que exercem no sistema, tornando a leitura dos diretórios mais semântica, como é possível observar na Figura 11. Apenas o arquivo "main.py" foi mantido na raiz, que é o arquivo executado para iniciar o sistema.

Primeiramente a pasta "app", na qual guarda arquivos que remetem diretamente ao uso do sistema. Nele há o diretório "controllers", que armazena arquivos responsáveis diretamente

pelas rotas do *back-end*, eles possuem a lógica principal por trás de cada *endpoint*, além de proteger as rotas através da lógica de autenticação e autorização que foi implementada. Em seguida, o diretório “*models*” é responsável por armazenar as classes utilizadas para mapear objetos Python para documentos BSON (Binary JSON), que é o tipo de arquivo utilizado pelo MongoDB para armazenar dados, utilizando a biblioteca MongoEngine para realizar esse mapeamento, em vez de lidar diretamente com consultas ao banco de dados e documentos BSON.

Por último o diretório “*config*”, no qual armazena arquivos responsáveis por configurar o projeto. Neste caso, ele foi utilizado para configurar a conexão com o banco de dados.

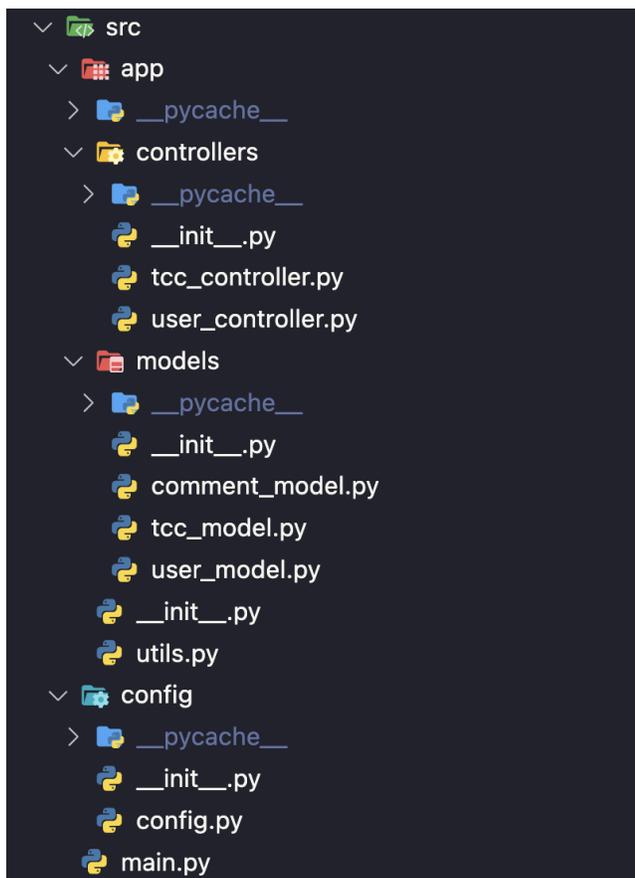


Figura 11: Organização da estrutura de diretórios do projeto do servidor

5 AVALIAÇÃO DA FERRAMENTA

Com o intuito de avaliar a satisfação dos usuários ao utilizar a plataforma Tenho que Concluir o Curso, foi conduzido um estudo utilizando o questionário CSUQ [2]. Este formato de questionário foi originalmente desenvolvido pela IBM (*International Business Machines Corporation*) em 1988 como parte de um projeto interno focado em métricas de usabilidade.

Entre várias opções de questionários padronizados para medir a satisfação do usuário, como o QUIS (*Questionnaire for User Interaction Satisfaction*), o SUMI (*Software Usability*

Measurement Inventory) e o SUS (*System Usability Scale*), optamos pelo CSUQ, foi escolhido o CSUQ devido à sua concisão, contendo menos perguntas do que alguns outros testes, enquanto ainda avalia eficientemente diversos aspectos importantes de usabilidade. Ele mede a eficácia do sistema em realizar suas funções propostas, a satisfação do usuário ao usá-lo e a gestão de erros pelo sistema.

O questionário² consiste em 19 questões, e os participantes respondem em uma escala de 7 pontos, variando de 1 a 7. Onde 1 indica "Discordo fortemente" e 7 indica "Concordo fortemente". Portanto, pontuações mais altas indicam uma melhor avaliação. Além disso, os participantes têm a opção de fornecer justificativas para suas respostas, se desejarem, e são solicitados a listar pontos positivos e negativos que encontraram no sistema.

Foram selecionados 7 usuários, que participaram voluntariamente, com o perfil de concluintes e recém-graduados do curso de Ciência da Computação na UFCG. O resultado do CSUQ é encontrado calculando-se a média dos valores das respostas para cada questão. No geral, a média de todas as questões foi de 6.37, indicando que a aplicação apresentou bons resultados de usabilidade, uma vez que este valor foi próximo de 7, sendo considerado alto.

5.1 Resultados

Na avaliação feita pelos usuários foram encontrados alguns pontos nos quais foram corrigidos e implementados após o *feedback*, tais como melhorias na interface, para a deixar mais intuitiva, adicionando uma página inicial explicando sobre como funciona o sistema e sua utilidade.

Alguns outros pontos foram levantados, como a simplicidade na interface, podendo colocar mais informações em alguns cenários como o *preview* na listagem de temas, na página inicial da aplicação.

Porém, isso não vai contra o que os usuários acharam do quão amigável a interface está, tendo pontuações positivas nas questões 16 e 17, ambas relacionadas à interface. Percebe-se que o minimalismo foi um ponto positivo, apesar de alguns terem achado minimalista demais em alguns casos. Eles aprovaram o visual da aplicação, a velocidade no qual o conteúdo carrega.

6 CONCLUSÕES

O planejamento de desenvolvimento envolveu a escolha de tecnologias com base no conhecimento prévio do desenvolvedor, consolidação de mercado, contexto do projeto e simplicidade de uso. A metodologia ágil Scrum foi adaptada para um único desenvolvedor. O desenvolvimento foi dividido em Sprints de 7 dias, totalizando 8 ciclos. A principal dificuldade foi criar um layout responsivo para dispositivos Desktop e Mobile, mantendo a simplicidade e intuitividade, usando as ferramentas que a biblioteca Vuetify.js oferece. O desenvolvedor enfrentou desafios no *back-end*, uso Docker para gerenciar o banco de dados e Firebase Authentication para autenticação de usuários. O projeto ensinou valiosas lições sobre avaliação de esforço e uso de bibliotecas consolidadas no mercado, apesar de enfrentar problemas com o Nuxt.js.

² <https://forms.gle/L2wgVkrNNExwcQdi7>

No desfecho, o projeto obteve resultados positivos, uma vez que foi concluído no prazo que foi dado, atendendo tanto as funcionalidades esperadas quanto aquelas que foram incorporadas com base nos *feedbacks* recebidos. Agora ela está pronta para auxiliar os estudantes a lidar de forma mais eficaz com a pressão e o estresse associado à busca por um tema de TCC e um orientador; que é o resultado esperado no final, resolver este problema.

7 TRABALHOS FUTUROS

Como trabalhos futuros, é possível abordar os problemas identificados na seção 5.1, referentes à interface do sistema. Adicionalmente, é possível evoluir o sistema para permitir:: notificações por e-mail quando novas mensagens chegarem; mensagens diretas entre professores e alunos, sem precisar entrar na página do tema para isso; tema escuro, que foi bastante pedido no questionário de avaliação.

Outra abordagem de evolução é o alinhamento de interesses com a coordenação do curso de Ciência da Computação da UFCG para encorajar o uso do sistema entre os alunos e professores, além de discutir e implementar sugestões propostas por ela. Por ser um projeto de código aberto, alunos e professores podem sugerir e participar do desenvolvimento de novas funcionalidades e melhorias no sistema, além de ser algo no qual os alunos são encorajados a realizar durante sua jornada no curso.

REFERÊNCIAS

- [1] Firebase. Firebase. Retrieved from <https://firebase.google.com/>
- [2] James R. Lewis. 1995. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-computer Interaction* 7, 1: 57–78. <https://doi.org/10.1080/10447319509526110>
- [3] MongoDB. MongoDB: a plataforma de aplicação de dados. MongoDB. Retrieved from <https://www.mongodb.com/pt-br>
- [4] Equipe Totvs. 2022. Scrum: guia completo. TOTVS. Retrieved from <https://www.totvs.com/blog/negocios/scrum/>
- [5] 2023. Docker: Accelerated Container Application Development. Docker. Retrieved from <https://www.docker.com/>
- [6] Firebase Authentication. Firebase. Retrieved from <https://firebase.google.com/docs/auth>
- [7] CSS | MDN. Retrieved from <https://developer.mozilla.org/pt-BR/docs/Web/CSS>
- [8] HTML: Linguagem de Marcação de Hipertexto | MDN. Retrieved from <https://developer.mozilla.org/pt-BR/docs/Web/HTML>
- [9] JavaScript | MDN. Retrieved from <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- [10] Vuetify — A Vue Component Framework. Vuetify. Retrieved from <https://vuetifyjs.com/en/>
- [11] Nuxt: The Intuitive Vue Framework. Nuxt. Retrieved from <https://nuxt.com/>
- [12] Vue.js - The Progressive JavaScript Framework | Vue.js. Retrieved from <https://vuejs.org/>
- [13] Welcome to Flask — Flask Documentation (3.0.X). Retrieved from <https://flask.palletsprojects.com/en/3.0.x/>