



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

**Matheus Vinicius Benevides Lima**

**BETTER TOGETHER:**

**APLICAÇÃO WEB PARA CONECTAR PROGRAMADORES**

**CAMPINA GRANDE - PB**

**2023**

**Matheus Vinicius Benevides Lima**

**BETTER TOGETHER:**

**APLICAÇÃO WEB PARA CONECTAR PROGRAMADORES**

**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 : Cláudio Elízio Calazans Campelo**

**CAMPINA GRANDE - PB**

**2023**

**Matheus Vinicius Benevides Lima**

## **BETTER TOGETHER:**

### **APLICAÇÃO WEB PARA CONECTAR PROGRAMADORES**

**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:**

**Cláudio Elízio Calazans Campelo**  
**Orientador – UASC/CEEI/UFCG**

**Patrícia Duarte de Lima Machado**  
**Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro**  
**Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 28 de Junho de 2023.**

**CAMPINA GRANDE - PB**

## RESUMO

Criar projetos de programação que necessitam de mais de um programador é uma prática comum na comunidade de desenvolvimento de software. Isso ocorre porque muitos projetos exigem uma equipe de programadores com diferentes habilidades e conhecimentos para serem concluídos de maneira eficiente e eficaz. É comum ver programadores buscando outros profissionais para colaborar em seus projetos e, assim, alcançar um resultado de alta qualidade e escalabilidade.

Infelizmente, muitas vezes esse projeto não é levado adiante devido à falta de pessoas capacitadas e dispostas a trabalhar neles. O problema é a dificuldade em encontrar programadores qualificados e interessados em trabalhar em um projeto específico, o que pode levar a uma perda de tempo e recursos para aqueles que estão procurando por colaboradores. Nesse contexto, o presente projeto propõe uma solução: uma aplicação web que organiza e conecta programadores para realizar projetos em conjunto.

O usuário que se cadastrar no site irá disponibilizar informações sobre suas habilidades e conhecimentos na área de programação, permitindo que seja conectado a projetos disponíveis de acordo com seu perfil. Além disso, o usuário também poderá criar seu próprio projeto, informando as tecnologias que serão utilizadas. Dessa forma, espera-se que, ao final deste projeto, os programadores que desejam criar algum projeto ou empreender consigam encontrar facilmente outros programadores capacitados e dispostos a trabalhar com eles, superando a dificuldade encontrada atualmente na procura por colaboradores.

## **BETTER TOGETHER:**

### **APLICAÇÃO WEB PARA CONECTAR PROGRAMADORES**

#### **ABSTRACT**

Creating programming projects that require more than one programmer is a common practice in the software development community. This is because many projects require a team of programmers with different skills and knowledge to be completed efficiently and effectively. It is common to see programmers seeking other professionals to collaborate on their projects and thus achieve high-quality and scalable results.

Unfortunately, often these projects are not pursued due to the lack of qualified and willing individuals to work on them. The problem lies in the difficulty of finding skilled programmers interested in working on a specific project, which can lead to wasted time and resources for those seeking collaborators. In this context, this project proposes a solution: a web application that organizes and connects programmers to work on projects together.

Users who register on the website will provide information about their skills and knowledge in the programming field, allowing them to be matched with available projects according to their profiles. Additionally, users will also be able to create their own projects, specifying the technologies to be used. Thus, it is expected that by the end of this project, programmers who wish to create a project or collaborate will be able to easily find other skilled programmers willing to work with them, overcoming the current difficulty in finding collaborators.

# Better Together: Aplicação web para conectar programadores

Trabalho de Conclusão de Curso

Matheus Vinicius Benevides Lima (Aluno), Cláudio Campelo (Orientador)

Departamento de Sistemas e Computação  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba - Brasil

## RESUMO

Criar projetos de programação que necessitam de mais de um programador é uma prática comum na comunidade de desenvolvimento de software. Isso ocorre porque muitos projetos exigem uma equipe de programadores com diferentes habilidades e conhecimentos para serem concluídos de maneira eficiente e eficaz. É comum ver programadores buscando outros profissionais para colaborar em seus projetos e, assim, alcançar um resultado de alta qualidade e escalabilidade.

Infelizmente, muitas vezes esse projeto não é levado adiante devido à falta de pessoas capacitadas e dispostas a trabalhar neles. O problema é a dificuldade em encontrar programadores qualificados e interessados em trabalhar em um projeto específico, o que pode levar a uma perda de tempo e recursos para aqueles que estão procurando por colaboradores. Nesse contexto, o presente projeto propõe uma solução: uma aplicação web que organiza e conecta programadores para realizar projetos em conjunto.

O usuário que se cadastrar no site irá disponibilizar informações sobre suas habilidades e conhecimentos na área de programação, permitindo que seja conectado a projetos disponíveis de acordo com seu perfil. Além disso, o usuário também poderá criar seu próprio projeto, informando as tecnologias que serão utilizadas. Dessa forma, espera-se que, ao final deste projeto, os programadores que desejam criar algum projeto ou empreender consigam encontrar facilmente outros programadores capacitados e dispostos a trabalhar com eles, superando a dificuldade encontrada atualmente na procura por colaboradores.

## PALAVRAS-CHAVE

Aplicação web, *Better Together*, *backend*, *frontend*

## 1 INTRODUÇÃO

A computação é uma área em constante evolução que vem influenciando significativamente todas as outras áreas da sociedade, desde a tecnologia da informação, finanças e saúde até a educação. A programação é a base da computação, pois é através dela que se escrevem os programas que serão utilizados pelos computadores.

Nos dias atuais, a programação é amplamente utilizada e é uma das profissões mais valorizadas e procuradas no mercado. Com o aumento da popularidade da computação e da facilidade de acesso a

conteúdos, é cada vez mais comum encontrarmos jovens estudantes interessados em seguir carreira na área.

Várias linguagens de programação, como Python, Java e JavaScript, estão entre as mais populares e utilizadas, e existem muitos recursos disponíveis na internet para ajudar a aprender, como cursos online, comunidades e fóruns.

No entanto, a programação é uma área muito ampla e é necessário ter conhecimento e compreender bem diversas linguagens, bibliotecas, frameworks, entre outros recursos, para se tornar um programador eficiente. Isso pode ser um desafio para muitos iniciantes na área, pois requer muito tempo e dedicação.

Muitos programadores optam por se especializar em uma determinada área da programação, como mobile, web, desktop, ciência de dados, entre outras. Isso pode ser benéfico em termos de carreira, mas pode limitar o acesso a oportunidades de empreendedorismo e trabalho em equipe, já que é comum que projetos de programação requeiram a colaboração de programadores com diferentes habilidades e especializações.

Por isso, este projeto tem como objetivo facilitar a conexão entre programadores de diferentes especializações, para que possam trabalhar juntos em projetos de programação e alcançar sucesso empreendedor. Além disso, o projeto oferece uma oportunidade para os programadores compartilharem conhecimentos e se desenvolverem profissionalmente ao trabalhar com colegas altamente capacitados. Em resumo, este projeto ajuda a resolver o problema da especialização e incentiva o empreendedorismo, o trabalho em equipe e o compartilhamento de conhecimento na programação.

## 2 ARQUITETURA

A arquitetura desse projeto envolve a divisão de responsabilidades entre o *backend* e o *frontend* [2]. Cada uma dessas partes desempenha um papel fundamental na construção de um sistema robusto e funcional.

O *backend*, também conhecido como camada de servidor, é responsável pelo processamento e gerenciamento dos dados, bem como pela lógica de negócio do sistema. Ele lida com a lógica computacional, a comunicação com bancos de dados e outras integrações externas, além de fornecer APIs (Interfaces de Programação de Aplicativos) para permitir a comunicação entre o *frontend* e o servidor. Em suma, o *backend* atua como o "cérebro" do sistema, processando solicitações, executando operações e fornecendo dados para o *frontend*.

O *frontend*, por sua vez, refere-se à interface com a qual os usuários interagem diretamente. É responsável por apresentar informações ao usuário de forma visualmente atraente e interativa. Isso envolve o desenvolvimento de interfaces de usuário (UI) e a implementação da lógica de interação, como captura de entrada do

Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

usuário e exibição de resultados. O *frontend* é construído usando tecnologias web, como *HTML*, *CSS* e *JavaScript*, e é executado no navegador do usuário. Ele se comunica com o *backend* para buscar dados e enviar comandos, garantindo uma experiência de usuário fluida.

A separação entre o *backend* e o *frontend* permite uma maior modularidade e escalabilidade do sistema como é visto na Figura 1. Cada camada pode ser desenvolvida e mantida independentemente, facilitando a manutenção, a depuração e a implementação de melhorias. Além disso, essa divisão também permite que equipes diferentes trabalhem em paralelo, especializando-se em suas respectivas áreas de expertise.

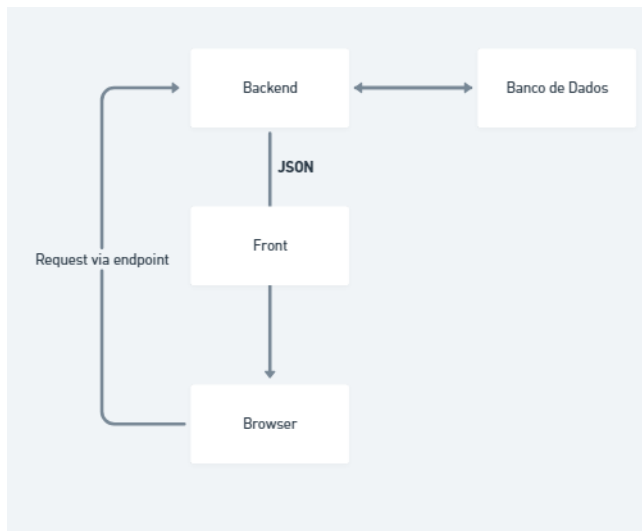


Figura 1: Arquitetura geral de um projeto web

## 2.1 Ambiente de Programação

Uma IDE (Integrated Development Environment) [4] é um software que fornece um conjunto integrado de ferramentas para desenvolvimento de software. Ela oferece recursos que ajudam os programadores a escrever, depurar e testar seu código de maneira eficiente. Uma IDE normalmente inclui um editor de código, um compilador ou interpretador, depurador, ferramentas de construção e automação, além de recursos de gerenciamento de projetos.

O Visual Studio Code (também conhecido como VS Code) [17] é um exemplo popular de IDE. Ele é desenvolvido pela Microsoft e é um editor de código-fonte leve e altamente personalizável. Embora seja chamado de "Visual Studio", o VS Code é uma versão simplificada e mais leve do ambiente de desenvolvimento completo chamado Visual Studio.

O VS Code foi projetado para ser uma ferramenta flexível e de alto desempenho que pode ser usada para desenvolver uma ampla gama de aplicativos e linguagens de programação. Ele possui suporte integrado para várias linguagens, incluindo *JavaScript*, *Python*, *C++*, *Java*, *HTML*, *CSS* e muitas outras. Além disso, o VS Code é altamente extensível por meio de sua arquitetura de plugin, permitindo aos desenvolvedores adicionar funcionalidades extras e personalizar a IDE de acordo com suas necessidades.

No contexto do *Better Together*, tanto o *backend* quanto o *frontend* foram desenvolvidos no vs code.

## 2.2 backend

O *backend* do projeto foi implementado utilizando o framework Laravel [5] em conjunto com o banco de dados MySQL [6]. A escolha dessas tecnologias traz benefícios significativos para o desenvolvimento e funcionamento do sistema.

O Laravel é um framework de desenvolvimento web em PHP [10] conhecido por sua elegância, facilidade de uso e recursos poderosos. Ele segue os princípios de programação orientada a objetos, utiliza uma sintaxe clara e expressiva e oferece uma ampla gama de funcionalidades pré-construídas, o que acelera o desenvolvimento e torna o código mais legível e organizado. O Laravel também oferece recursos avançados de segurança, gerenciamento de sessões, autenticação e autorização, além de uma arquitetura MVC (Model-View-Controller) bem definida, que separa as responsabilidades e facilita a manutenção e a escalabilidade do sistema.

O banco de dados MySQL é uma escolha popular para projetos web devido à sua confiabilidade, desempenho e suporte robusto da comunidade. Ele é um sistema de gerenciamento de banco de dados relacional (RDBMS)[3] amplamente utilizado e compatível com várias plataformas. O MySQL possui recursos avançados de indexação, consultas otimizadas e capacidade de lidar com grandes volumes de dados. Além disso, é fácil de aprender e possui uma ampla documentação disponível, o que facilita o desenvolvimento e a manutenção do banco de dados.

A estrutura do *backend* foi desenvolvida utilizando o framework Laravel, que segue uma arquitetura MVC (Model-View-Controller) e promove a separação de preocupações e a modularidade do código.

No contexto do Laravel, a estrutura do *backend* tem como base os seguintes elementos: Model, Controller, Service e DAO. Alguns desses elementos estão presentes no *backend* e estão representados na Figura 2.

### 2.2.1 Model.

A camada de modelo, no contexto do desenvolvimento de software, é responsável por representar a estrutura dos dados e a lógica de negócio da aplicação. Em termos gerais, os modelos são classes ou entidades que definem os atributos e comportamentos dos objetos manipulados pelo sistema. Eles servem como uma abstração das entidades do mundo real ou dos conceitos relevantes para a aplicação.

### 2.2.2 Controller.

O controller desempenha um papel fundamental na estrutura MVC (Model-View-Controller) do framework. O controller é responsável por receber as requisições HTTP, processá-las e retornar as respostas adequadas.

O principal objetivo de um controller é controlar o fluxo de uma aplicação, recebendo as requisições vindas das rotas e coordenando as ações necessárias. Ele atua como um intermediário entre as rotas e os modelos (models) da aplicação.

### 2.2.3 Service.

O service (ou serviço) é uma camada responsável pela implementação da lógica de negócio da aplicação. Ele atua como uma ponte

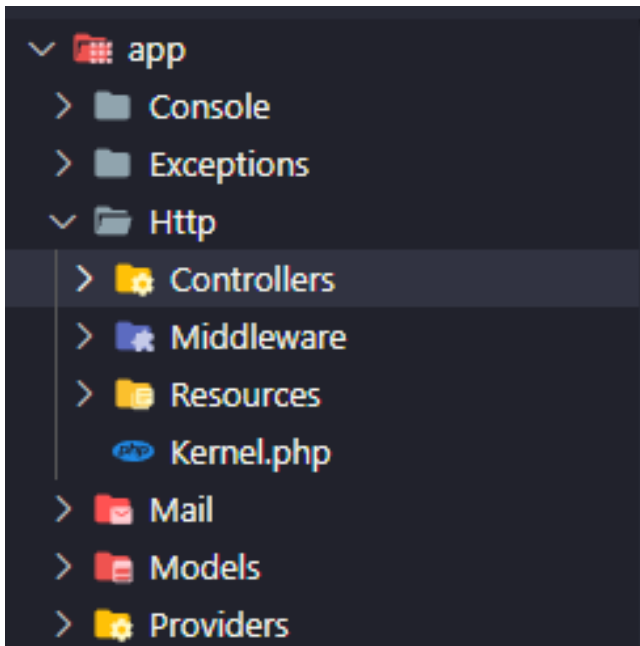


Figura 2: Parte da estrutura do backend

entre o controller e o modelo (model), concentrando-se na execução de tarefas mais complexas e na manipulação dos dados.

O service desempenha um papel importante na organização e estruturação do código, ajudando a manter a responsabilidade do controller focada na coordenação das ações e a manter os modelos mais leves e focados nas operações de banco de dados.

#### 2.2.4 DAO.

Camada de acesso a dados (DAO) é gerenciado por meio das classes de modelo, que utilizam a ORM do framework para interagir com o banco de dados. No *Better Together*, os modelos "Usuário", "Experiência" e "Projeto" são responsáveis por realizar operações de consulta, inserção, atualização e exclusão de dados no banco de dados MySQL. O Laravel fornece recursos convenientes para realizar essas operações de forma simplificada e segura.

### 2.3 frontend

O *frontend* do projeto foi implementado utilizando as tecnologias Next.js, *react* e *SCSS* para estilização. Essas escolhas proporcionam benefícios significativos no desenvolvimento e na experiência do usuário.

O Next.js é um framework de desenvolvimento web em JavaScript que utiliza o conceito de renderização do lado do servidor (SSR) e pré-renderização estática para criar aplicações web rápidas e eficientes. Ele oferece uma experiência de desenvolvimento moderna e escalável, permitindo a criação de páginas dinâmicas e interativas. O Next.js possui uma arquitetura baseada em componentes reutilizáveis, facilitando a organização e a manutenção do código. Além disso, ele oferece recursos avançados, como roteamento automático, suporte a API integrada e otimização de desempenho.

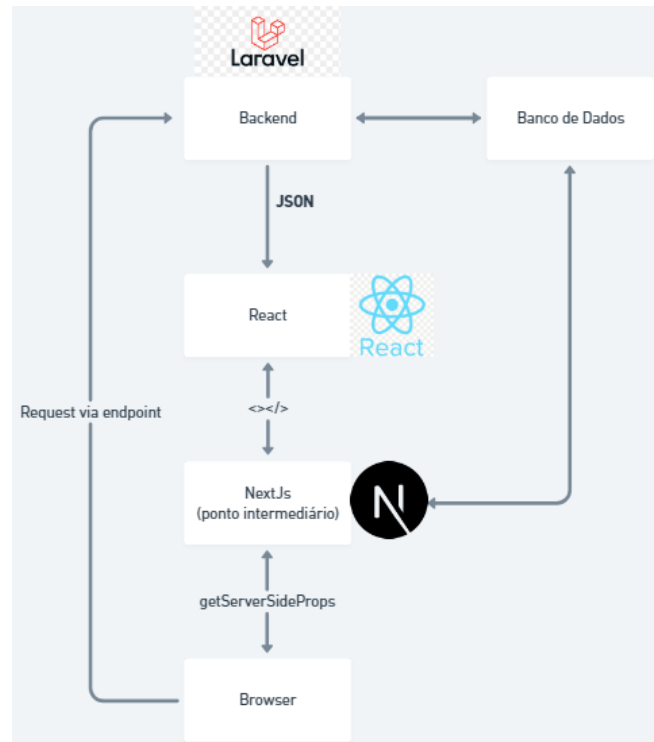


Figura 3: Arquitetura geral de um projeto web com o next

O *react* [16] é uma linguagem de programação que adiciona recursos de tipagem estática ao JavaScript. Ele fornece um sistema de tipos robusto, permitindo a detecção de erros durante a fase de desenvolvimento e melhorando a manutenção e a legibilidade do código. O uso do *react* traz mais confiabilidade ao *frontend*, garantindo a integridade dos dados e facilitando a colaboração entre os membros da equipe.

O *SCSS (Sass)* [13] é uma extensão do *CSS* que oferece recursos adicionais, como variáveis, aninhamento de seletores, mixins e importação de arquivos. Ele permite a criação de estilos reutilizáveis, facilitando a estilização consistente e a manutenção do design da aplicação. O *SCSS* oferece uma sintaxe mais poderosa e expressiva em comparação ao *CSS* tradicional, resultando em um código mais limpo e legível.

A combinação dessas tecnologias proporciona um ambiente de desenvolvimento moderno, escalável e eficiente para o *frontend* do projeto, como pode ser visto na Figura 3 que representa essa combinação. O Next.js oferece recursos avançados de renderização, roteamento e desempenho, enquanto o *react* traz mais segurança e legibilidade ao código. Além disso, o *SCSS* facilita a estilização e a manutenção do design da aplicação. Essas escolhas tecnológicas contribuem para uma experiência agradável e de alta qualidade para os usuários do sistema.

A aplicação também utiliza os seguintes pacotes no funcionamento do seu front end:

- *Axios* [1]: Responsável por lidar com requisições HTTP para o back end da aplicação.



- *React Router* [12]: Controla o redirecionamento do usuário para diferentes telas a partir da URL acessada.
- *nookies* [7]: Possibilita o uso de cookies[18], arquivos que são armazenados localmente no computador do usuário com informações como o token da sessão do usuário[15].
- *phospor-react* [9]: O projeto Phosphor é uma biblioteca de ícones de código aberto que oferece uma ampla variedade de ícones para uso em aplicativos web e interfaces de usuário.
- *react-spinners*: React Spinners é uma biblioteca de componentes pré-construídos para criar indicadores de carregamento (spinners) em aplicações React[11].

Para lidar com tais dependências do front end, foi utilizado o gerenciador de pacotes npm [8].

### 3 TESTES

Os testes em uma API são usados para verificar se ela está funcionando corretamente e se atende aos requisitos e especificações definidos. Eles ajudam a garantir que a API esteja respondendo adequadamente às solicitações, retornando os resultados esperados e lidando corretamente com possíveis cenários de erro.

Nesse projeto, foi usado o Thunder Client [14] como ferramenta de testes além de um grupo selecionado de pessoas para compartilhar suas experiências com o programa.

Além de compartilharem suas experiências como usuários da interface, o grupo de pessoas selecionado também foi solicitado a avaliar a facilidade e rapidez de uso do programa, pois essas questões são problemas recorrentes enfrentados por muitos programadores em ecossistemas semelhantes, levando muitos a desistirem de utilizá-los. A inclusão dessa avaliação permitiu obter informações adicionais sobre a usabilidade do programa e identificar possíveis obstáculos ou pontos de melhoria. Esses aspectos são fundamentais para garantir que o programa seja intuitivo e eficiente, proporcionando uma experiência satisfatória aos usuários.

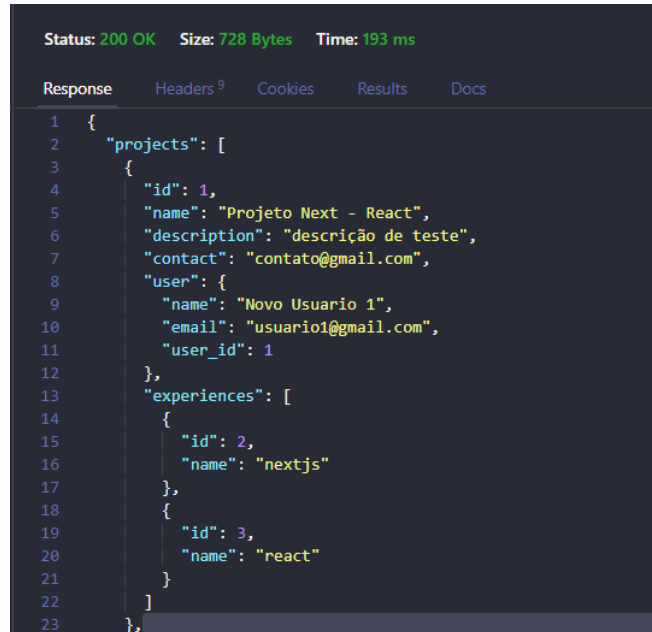
#### 3.1 Thunder Client

O Thunder Client é uma extensão do Visual Studio Code que permite executar e testar solicitações HTTP diretamente no ambiente de desenvolvimento. Ele simplifica o processo de teste de APIs e serviços da web. Na Figura 4, é possível ver bem um exemplo de requisição.

Além de simplificar o processo de teste de APIs e serviços da web, o Thunder Client apresenta diversos pontos positivos. Em primeiro lugar, sua integração direta com o Visual Studio Code proporciona uma experiência de desenvolvimento contínua, permitindo que os desenvolvedores testem suas solicitações HTTP sem sair do ambiente de programação.

Outro ponto positivo é a facilidade de uso do Thunder Client. Sua interface intuitiva e amigável torna simples a definição e execução de solicitações HTTP. Com apenas alguns cliques, é possível configurar as solicitações, adicionar parâmetros, cabeçalhos e payloads, e enviar as requisições para testar os endpoints.

Além disso, o Thunder Client oferece recursos avançados, como a capacidade de criar coleções de solicitações para organizar e reutilizar testes, o suporte a variáveis de ambiente para tornar os testes mais flexíveis e dinâmicos, e a possibilidade de automatizar os testes utilizando scripts.



```

Status: 200 OK   Size: 728 Bytes   Time: 193 ms

Response  Headers  Cookies  Results  Docs

1  {
2    "projects": [
3      {
4        "id": 1,
5        "name": "Projeto Next - React",
6        "description": "descrição de teste",
7        "contact": "contato@gmail.com",
8        "user": {
9          "name": "Novo Usuario 1",
10         "email": "usuario1@gmail.com",
11         "user_id": 1
12       },
13       "experiences": [
14         {
15           "id": 2,
16           "name": "nextjs"
17         },
18         {
19           "id": 3,
20           "name": "react"
21         }
22       ]
23     }
  
```

Figura 4: Request feito com o thunder client

Outra vantagem do Thunder Client é a capacidade de visualizar as respostas em tempo real. Ele exibe as informações detalhadas da resposta HTTP, incluindo códigos de status, cabeçalhos e corpo da resposta, facilitando a análise e depuração dos resultados dos testes.

Além disso, o Thunder Client oferece recursos de validação de resposta, permitindo que os desenvolvedores verifiquem se a resposta está de acordo com as expectativas. Isso é especialmente útil ao lidar com APIs que seguem esquemas JSON, pois o Thunder Client pode validar a estrutura da resposta em relação a um esquema definido.

## 4 CRONOLOGIA

O projeto foi dividido em quatro etapas: preparação de ambiente, *backend*, *frontend* e testes, cada uma com um tempo definido de acordo com sua dificuldade.

### 4.1 Preparação de Ambiente

Com a escolha do VS Code como ide para o projeto, foi alocado um tempo de 1 semana para instalar todas as dependências necessários para o projeto, vistas na Figura 5.

### 4.2 *backend*

O tempo alocado para o desenvolvimento do *backend* foi de 1 mês e meio por causa da complexidade do framework Laravel e sua curva de aprendizado.

Apesar de ser uma ferramenta poderosa e altamente versátil, sua abrangência exigiu um esforço adicional para compreender e utilizar efetivamente suas funcionalidades. Foram necessárias pesquisas extensivas, estudos aprofundados e experimentações para dominar os conceitos e as melhores práticas do Laravel.

```

"dependencies": {
  "@phosphor-icons/react": "^2.0.9",
  "@types/node": "20.1.1",
  "@types/react": "18.2.6",
  "@types/react-dom": "18.2.4",
  "axios": "^1.4.0",
  "eslint": "8.40.0",
  "eslint-config-next": "13.4.1",
  "jwt-decode": "^3.1.2",
  "next": "13.4.1",
  "nookies": "^2.5.2",
  "phosphor-icons": "^1.4.2",
  "phosphor-react": "^1.4.1",
  "react": "18.2.0",
  "react-dom": "18.2.0",
  "react-spinners": "^0.13.8",
  "sass": "^1.62.1",
  "typescript": "5.0.4"
}

```

Figura 5: Dependências utilizadas no projeto

### 4.3 frontend

O tempo alocado para o desenvolvimento do frontend foi de 1 mês e meio por causa da adaptação ao framework Next.js, em conjunto com o uso do *react*.

Embora o Next.js seja conhecido por sua capacidade de construir aplicativos *react* poderosos, a incorporação do *react* trouxe desafios adicionais devido à sua tipagem estática. Foi necessário adquirir conhecimento em *react* e ajustar-se à sua sintaxe específica para garantir a correta tipagem das variáveis, componentes e fluxos de dados.

### 4.4 Testes

Para a realização dos testes, foi reservado um período de 3 semanas para utilizar o Thunder Client e solicitar que algumas pessoas compartilhassem suas experiências como usuários da interface.

## 5 USABILIDADE

O usuário, ao se deparar com a tela inicial, exemplificada na Figura 6, tem uma breve apresentação do projeto, bem como as opções de *login* e cadastro.

O cadastro requer apenas nome - que será utilizado como nome do usuário -, email e senha. O *login* requer apenas email e senha.

Ao fazer *login* e ser redirecionado para o painel, o usuário se depara com o menu de navegação contendo todas as opções disponíveis. É recomendado que, ao fazer *login*, o usuário adicione suas áreas de



Figura 6: Página inicial do *Better Together*

experiência para ser compatibilizado com os projetos disponíveis de acordo com seu perfil.

Além disso, o usuário também pode criar seu próprio projeto, indicando as tecnologias necessárias para o mesmo.

Dessa forma, tanto aqueles que procuram projetos quanto aqueles que procuram programadores terão acesso aos perfis das pessoas ou dos projetos, com informações de contato visíveis na tela para que possam se conectar com o candidato ou proprietário do projeto em questão.

O projeto também oferece todas as funcionalidades padrões de um sistema, como a capacidade de alterar a senha, o nome de usuário e a opção de *login* automático caso tenha feito *login* recentemente. Todas as opções de navegação são vistas na Figura 7.

## 6 DIFICULDADES E APRENDIZADO

Durante a realização do Trabalho de Conclusão de Curso, foram obtidos diversos aprendizados valiosos. No entanto, também foram enfrentadas algumas dificuldades, especialmente ao lidar com a gestão do tempo em meio a outras responsabilidades da faculdade e do trabalho, além de se adaptar às tecnologias Next.js e Laravel.

Uma das principais dificuldades encontradas foi a conciliação entre as demandas do TCC, as aulas regulares e as obrigações profissionais. Gerenciar efetivamente o tempo para equilibrar todas essas responsabilidades foi um desafio significativo. Foi necessário um planejamento cuidadoso, estabelecendo prioridades claras e definindo metas realistas para garantir que todas as tarefas pudessem ser, na medida do possível, concluídas dentro dos prazos estabelecidos.

Além disso, a adaptação ao Next.js e ao Laravel também apresentou seus desafios. Ambos os frameworks possuem curvas de aprendizado distintas, com suas próprias peculiaridades e convenções. Foi necessário investir tempo em estudos, pesquisas e experimentações para se familiarizar com as melhores práticas, estruturas de projeto e recursos específicos de cada tecnologia. A capacidade de adaptar-se rapidamente e adquirir novos conhecimentos foi fundamental para superar essas dificuldades e avançar no desenvolvimento do projeto.

No entanto, essas dificuldades também proporcionaram aprendizados valiosos. A gestão eficaz do tempo e o equilíbrio entre as

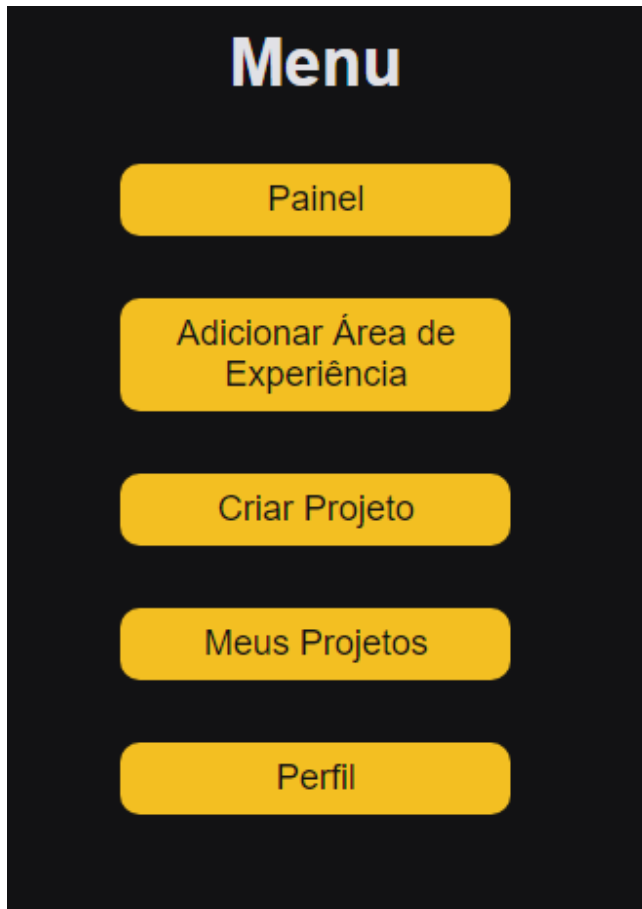


Figura 7: Menu de navegação

responsabilidades foram habilidades essenciais adquiridas durante o processo do TCC. A experiência de se adaptar a novas tecnologias fortaleceu a capacidade de aprendizado contínuo e a habilidade de enfrentar desafios técnicos com confiança.

## 7 TRABALHOS FUTUROS

Atualmente, o projeto apresenta algumas oportunidades de melhorias em futuras implementações, incluindo:

- **Notificações em tempo real:** Adicionar um sistema de notificações em tempo real para manter os usuários atualizados sobre novos projetos, atualizações de status ou mensagens recebidas.
- **Sistema de avaliação e feedback:** Incorporar um sistema de avaliação e feedback entre os membros do projeto, permitindo que eles classifiquem e deixem comentários sobre a colaboração de outros membros.
- **Integração com redes sociais:** Implementar recursos de compartilhamento em redes sociais, permitindo que os usuários divulguem projetos interessantes ou suas próprias experiências em outras plataformas.
- **Melhorias na interface do usuário:** Realizar estudos de usabilidade e feedback dos usuários para aprimorar a interface

do usuário, tornando-a mais intuitiva, amigável e responsiva em diferentes dispositivos.

- **Recomendação de projetos:** Desenvolver um sistema de recomendação de projetos com base nas áreas de experiência dos usuários, histórico de projetos anteriores e preferências individuais, proporcionando uma experiência de busca personalizada.

## AGRADECIMENTOS

Gostaria de expressar minha gratidão a Deus, minha família, amigos e ao meu orientador, Cláudio Campelo. Com o apoio divino, amor familiar, amizade sincera e orientação dedicada, pude concluir este trabalho com sucesso. Sou imensamente grato por todo o suporte, incentivo e sabedoria compartilhados ao longo dessa jornada acadêmica. Obrigado a todos por fazerem parte dessa conquista significativa em minha vida.

## REFERÊNCIAS

- [1] axios [n. d.]. axios. Retrieved June 14, 2023 from <https://www.npmjs.com/package/axios>
- [2] back e front [n. d.]. Back e Front. Retrieved June 14, 2023 from <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>
- [3] banco relacional [n. d.]. Banco Relacional. Retrieved June 14, 2023 from <https://www.oracle.com/br/database/what-is-a-relational-database/>
- [4] ide [n. d.]. IDE. Retrieved June 14, 2023 from <https://aws.amazon.com/pt/what-is/ide/>
- [5] laravel [n. d.]. Laravel. Retrieved June 14, 2023 from <https://laravel.com/>
- [6] mysql [n. d.]. MySQL. Retrieved June 14, 2023 from <https://www.mysql.com/>
- [7] nookies [n. d.]. Nookies. Retrieved June 14, 2023 from <https://www.npmjs.com/package/nookies>
- [8] npm [n. d.]. npm. Retrieved June 14, 2023 from <https://www.npmjs.com/>
- [9] phosphor-react [n. d.]. Phosphor-react. Retrieved June 14, 2023 from <https://www.npmjs.com/package/phosphor-react?activeTab=readme>
- [10] php [n. d.]. PHP. Retrieved June 14, 2023 from <https://www.php.net/>
- [11] react [n. d.]. react. Retrieved June 14, 2023 from <https://react.dev/>
- [12] router [n. d.]. React Router. Retrieved June 14, 2023 from <https://reactrouter.com/en/main>
- [13] scss [n. d.]. Scss. Retrieved June 14, 2023 from <https://sass-lang.com/>
- [14] thunderclient [n. d.]. Thunder Client. Retrieved June 14, 2023 from <https://www.thunderclient.com/>
- [15] token [n. d.]. Session tokens overview. Retrieved June 14, 2023 from <https://shopify.dev/apps/auth/oauth/session-tokens>
- [16] TypeScript [n. d.]. typescript. Retrieved June 14, 2023 from <https://www.typescriptlang.org/>
- [17] vscode [n. d.]. VS Code. Retrieved June 14, 2023 from <https://code.visualstudio.com/>
- [18] whatcookies [n. d.]. Cookies - Informações que os sites armazenam no seu computador. Retrieved June 14, 2023 from <https://support.mozilla.org/pt-BR/kb/cookies-informacoes-sites-armazenam-no-computador>