



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

**LUCIO NATHAN TRIGUEIRO MATIAS**

**POMOSYNC: UMA APLICAÇÃO PARA POMODORO EM GRUPO**

**CAMPINA GRANDE - PB**

**2023**

**LUCIO NATHAN TRIGUEIRO MATIAS**

**POMOSYNC: UMA APLICAÇÃO PARA POMODORO EM GRUPO**

**Trabalho de Conclusão de 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: Everton Leandro Galdino Alves**

**CAMPINA GRANDE - PB**

**2023**

**LUCIO NATHAN TRIGUEIRO MATIAS**

**POMOSYNC: UMA APLICAÇÃO PARA POMODORO EM GRUPO**

**Trabalho de Conclusão de 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:**

**Everton Leandro Galdino Alves**

**Orientador – UASC/CEEI/UFPG**

**Robert Kalley Cavalcanti de Menezes**

**Examinador – UASC/CEEI/UFPG**

**Francisco Vilar Brasileiro**

**Professor da Disciplina TCC – UASC/CEEI/UFPG**

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

**CAMPINA GRANDE - PB**

## RESUMO

Em ambientes de trabalho e estudo cada vez mais digitais e colaborativos, manter a produtividade individual e de equipe pode ser um desafio significativo. A técnica Pomodoro, desenvolvida nos anos 80, é uma metodologia de gerenciamento de tempo amplamente usada para aumentar a produtividade, mas sua implementação em um contexto colaborativo pode ser difícil. Portanto, a necessidade de uma solução digital eficaz para facilitar a implementação colaborativa da técnica Pomodoro se torna clara. O PomoSync surge como uma aplicação web que visa abordar esse problema, permitindo aos usuários organizar e participar de sessões Pomodoro de forma colaborativa. O objetivo deste trabalho é detalhar o processo de desenvolvimento desta aplicação, analisando seus aspectos técnicos, metodológicos e seu impacto no aumento da produtividade em ambientes colaborativos. Os resultados obtidos sugerem um bom grau de satisfação do usuário com a usabilidade e eficácia do PomoSync, validando a necessidade de tal solução e, a partir das críticas feitas ao sistema, são sugeridas futuras melhorias para a aplicação.

# **POMOSYNC: AN APPLICATION FOR GROUP POMODORO**

## **ABSTRACT**

In increasingly digital and collaborative work and study environments, maintaining individual and team productivity can be a significant challenge. The Pomodoro Technique, developed in the 1980s, is a time management methodology widely used to increase productivity, but its implementation in a collaborative context can be difficult. Therefore, the need for an effective digital solution to facilitate the collaborative implementation of the Pomodoro technique becomes clear. PomoSync emerges as a web application that aims to address this problem, allowing users to organize and participate in Pomodoro sessions collaboratively. The objective of this work is to detail the development process of this application, analyzing its technical and methodological aspects and its impact on increasing productivity in collaborative environments. The results obtained suggest a good degree of user satisfaction with the usability and effectiveness of PomoSync, validating the need for such a solution and, based on the criticisms made to the system, future improvements are suggested for the application.

# PomoSync: Uma aplicação para pomodoro em grupo

Lucio Nathan Trigueiro Matias

Unidade Acadêmica de Sistemas e Computação

Universidade Federal de Campina Grande

Campina Grande - PB, Brasil

lucio.matias@ccc.ufcg.edu.br

Everton Leandro Galdino Alves

Unidade Acadêmica de Sistemas e Computação

Universidade Federal de Campina Grande

Campina Grande - PB, Brasil

everton@computacao.ufcg.edu.br

## RESUMO

Em ambientes de trabalho e estudo cada vez mais digitais e colaborativos, manter a produtividade individual e de equipe pode ser um desafio significativo. A técnica Pomodoro, desenvolvida nos anos 80, é uma metodologia de gerenciamento de tempo amplamente usada para aumentar a produtividade, mas sua implementação em um contexto colaborativo pode ser difícil. Portanto, a necessidade de uma solução digital eficaz para facilitar a implementação colaborativa da técnica Pomodoro se torna clara. O PomoSync surge como uma aplicação web que visa abordar esse problema, permitindo aos usuários organizar e participar de sessões Pomodoro de forma colaborativa. O objetivo deste trabalho é detalhar o processo de desenvolvimento desta aplicação, analisando seus aspectos técnicos, metodológicos e seu impacto no aumento da produtividade em ambientes colaborativos. Os resultados obtidos sugerem um bom grau de satisfação do usuário com a usabilidade e eficácia do PomoSync, validando a necessidade de tal solução e, a partir das críticas feitas ao sistema, são sugeridas futuras melhorias para a aplicação.

**Repositório:** <https://github.com/lucionathan/pomodoro>

**Palavras-chave:** Técnica Pomodoro, *WebSocket*, *website*

## 1. INTRODUÇÃO

A crescente demanda por produtividade e eficiência levou ao desenvolvimento de uma variedade de técnicas e ferramentas de gerenciamento de tempo. A técnica Pomodoro, criada por Francesco Cirillo na década de 1980 [10], tornou-se um dos métodos mais populares, com uma vasta gama de aplicativos e plataformas que oferecem suporte a esta técnica. O Pomodoro envolve a divisão do tempo de trabalho em intervalos focados e regulares, geralmente de 25 minutos, seguidos por breves intervalos de descanso, promovendo assim a eficiência e reduzindo a fadiga mental e estresse [24].

Apesar de sua eficácia reconhecida, a implementação desta técnica pode ser desafiadora sem a ajuda de ferramentas adequadas. Por exemplo, organizar sessões de Pomodoro entre equipes pode ser complexo devido a questões de sincronização e gerenciamento de tempo.

A partir de pesquisas em ferramentas de busca na *internet* são encontradas diversas ferramentas para aplicação de uso pessoal da técnica. No entanto, a maioria dessas aplicações não oferece a capacidade de realização de sessões de Pomodoro de maneira

síncrona com outras pessoas, o que pode ser particularmente útil em ambientes de trabalho e estudo colaborativo.

Diante disso, este trabalho visa relatar o desenvolvimento de uma aplicação *web* chamada PomoSync. Essa ferramenta é projetada para apoiar e otimizar a técnica de Pomodoro, proporcionando uma plataforma onde os usuários podem realizar sessões de Pomodoro de maneira síncrona com outros usuários *online*. O PomoSync busca diferenciar-se por oferecer uma experiência interativa e colaborativa, permitindo aos usuários sincronizar suas sessões de trabalho e intervalos, participar de sessões públicas ou privadas e visualizar o progresso dos outros participantes em tempo real. Além disso, a aplicação apresenta um design intuitivo e amigável ao usuário, proporcionando um ambiente de trabalho atrativo e de fácil uso.

Os usuários do PomoSync terão a capacidade de criar e participar de sessões Pomodoro públicas ou privadas, rastrear e visualizar o progresso das suas sessões, receber notificações sobre o início de novas sessões, e até mesmo se comunicarem durante o Pomodoro. O objetivo é facilitar o uso da técnica Pomodoro em ambientes colaborativos e melhorar a produtividade dos usuários, permitindo que eles se concentrem mais efetivamente em suas tarefas e reduzam a fadiga mental e estresse de forma conjunta.

Além disso, este artigo também irá explorar a arquitetura do *software*, os métodos usados para testar e avaliar a aplicação, bem como os desafios encontrados durante o processo de desenvolvimento. A intenção é oferecer uma visão abrangente do processo de desenvolvimento, desde a concepção até a realização do projeto, além de fornecer percepções sobre possíveis melhorias para o futuro do sistema.

Para a validação da aplicação PomoSync, foi realizado um teste de usabilidade, preenchido por usuários após executarem uma série de tarefas designadas que exploram as funcionalidades da ferramenta. Utilizou-se o questionário CSUQ (Computer System Usability Questionnaire) [9], amplamente reconhecido como um eficaz instrumento de medição da satisfação do usuário ao interagir com softwares e produtos. Os resultados, em sua maioria, sugeriram uma facilidade de uso da aplicação. Contudo, algumas características do sistema receberam avaliações mistas, indicando a necessidade de aprimoramento em determinados aspectos da funcionalidade.

## 2. DOMÍNIO DO PROBLEMA

### 2.1 Sobre a Técnica Pomodoro

A técnica Pomodoro é uma estratégia de gerenciamento de tempo criada por Francesco Cirillo [10] na década de 1980. A ideia

central é dividir o trabalho em períodos de 25 minutos, separados por curtos intervalos de 5 minutos. Esses intervalos são conhecidos como "pomodoros". A técnica é especialmente útil para promover a produtividade e o foco, ao encorajar os usuários a se concentrarem em uma única tarefa durante um período definido e, em seguida, descansarem para evitar a fadiga.

Além dos benefícios mencionados, o uso da técnica Pomodoro pode ajudar a melhorar a gestão do tempo, evitar a procrastinação e aumentar a motivação. Por outro lado, requer uma adaptação e compromisso do usuário com a estrutura de tempo proposta.

A ideia surgiu a partir do uso de um temporizador de cozinha no formato de um tomate ("pomodoro" em italiano) para rastrear esses intervalos, ao qual foi atribuído o nome da técnica. No entanto, com a evolução da tecnologia, a prática tem se adaptado a diferentes ferramentas e plataformas, incluindo aplicativos digitais e *websites*.

## 2.2 Soluções Conhecidas

Com a popularização da técnica Pomodoro, várias soluções digitais foram criadas para facilitar o seu uso. Alguns exemplos populares, facilmente encontrados ao pesquisar pelo termo Pomodoro em ferramentas de busca na internet, são o PomoDoneApp [15], o Pomomo [16] e o Pomofocus.io [17].

O PomoDoneApp é um aplicativo de gerenciamento de tempo que integra a técnica Pomodoro com outras ferramentas de produtividade como Trello, Asana, Todoist, Jira, entre outras. Ele permite que os usuários acompanhem a quantidade de tempo gasto em tarefas específicas e também fornece recursos para relatórios de tempo e análise de produtividade.

O Pomomo tem uma abordagem diferente do PomoDoneApp, sendo um *bot* para o Discord, que se trata de uma plataforma de comunicação popular entre comunidades *online*, que possibilita o uso de *bots* em *chats* em grupo. O *bot* Pomomo permite que usuários em uma chamada de voz no Discord usem a técnica Pomodoro de forma síncrona.

Já o Pomofocus.io é uma aplicação *web* de Pomodoro online que inclui também uma ferramenta de lista de tarefas. O site oferece um design minimalista e permite que os usuários personalizem a duração dos períodos de trabalho e de descanso. Além disso, os usuários podem acompanhar as tarefas que concluíram durante cada período.

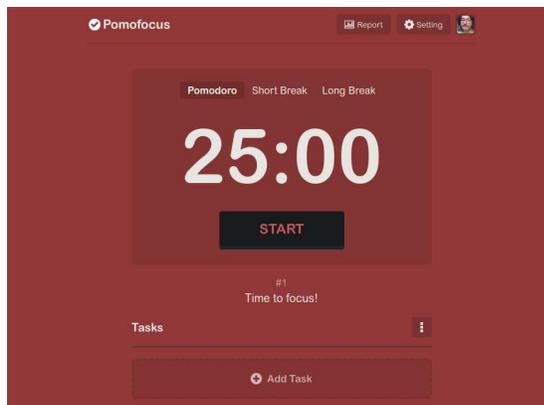


Figura 1: Interface do Pomofocus.io

## 3. SOLUÇÃO

### 3.1 Descrição

PomoSync é uma ferramenta de gerenciamento de tempo que permite que usuários realizem sessões de Pomodoro conjuntamente. A interface de usuário é acessível via uma aplicação *web* utilizando um navegador, que facilita a interação entre usuários que utilizam diferentes sistemas operacionais e se encontram em localizações físicas diferentes uns dos outros. O principal propósito da aplicação é fornecer um meio de realizar sessões de Pomodoro em grupo de forma sincronizada, permitindo que os usuários interajam entre si durante as pausas.

A ferramenta é acionada por meio de um processo de *login*. Uma vez autenticado, o usuário tem acesso a uma série de funcionalidades, incluindo a criação de novas sessões, acesso a sessões existentes por meio de um identificador único de sala, e a visualização de uma lista de sessões públicas disponíveis para ingresso.

O PomoSync se diferencia das soluções anteriormente apresentadas por sua capacidade de permitir sessões de Pomodoro em grupo de maneira sincronizada, facilitando a interação e colaboração entre os participantes. Além disso, o PomoSync destaca-se pelo seu foco específico na aplicação da técnica Pomodoro, priorizando a simplicidade e eficiência na execução desta metodologia, evitando distrações desnecessárias e proporcionando um ambiente propício para o aumento da produtividade. O PomoSync também incorpora um sistema de chat integrado, permitindo que os participantes interajam durante a sessão. Além disso, oferece uma atividade de descanso na forma de um vídeo sincronizado, ou seja, todos os usuários em uma sessão assistirão ao mesmo vídeo durante o período de descanso caso assim desejem.

### 3.2 Funcionalidades

#### 3.2.1 Gerenciamento de Usuários

O gerenciamento de usuários é essencial para o funcionamento do PomoSync, pois os usuários precisam estar logados na aplicação para utilizá-la. Nesta parte, os usuários podem se cadastrar, fazer *login* e gerenciar suas informações pessoais. Os casos de uso relacionados ao gerenciamento de usuários são:

1. Na interface *web*, é possível criar um novo usuário.
2. Na interface *web*, o usuário pode efetuar login usando suas credenciais.
3. Na interface *web*, o usuário pode se desconectar da sua conta.

As telas referentes a esses casos podem ser visualizadas nas Figuras 2 e 3.

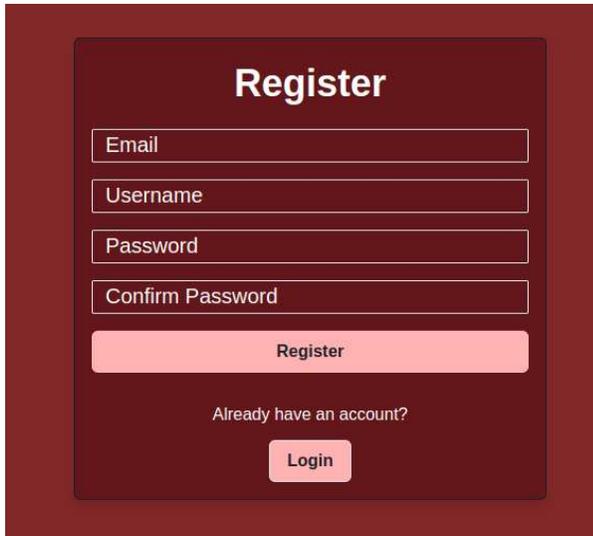


Figura 2: Tela de cadastro de novo usuário

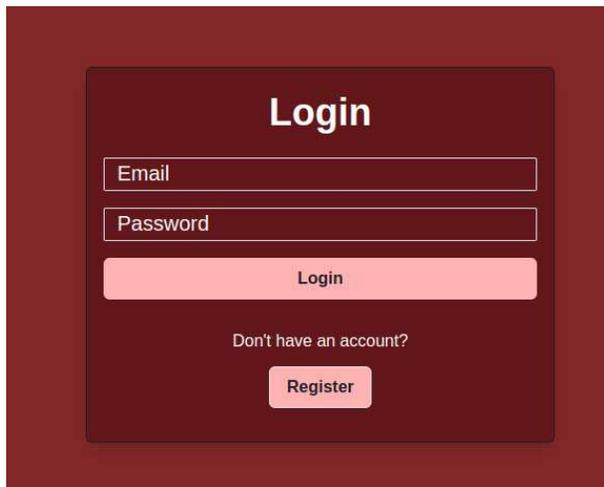


Figura 3: Tela de login de usuário

### 3.2.2 Home Page

A *Home Page* é a página que os usuários devem usar para escolher como participar de uma sessão de Pomodoro. Nesta parte, os usuários podem criar novas sessões de Pomodoro que podem ser públicas ou privadas. As sessões públicas são acessíveis para todos os usuários da plataforma, enquanto as sessões privadas são acessíveis apenas por meio de convites. Os usuários também podem entrar em sessões existentes e visualizar a lista de sessões públicas disponíveis. Os casos de uso relacionados à *Home Page* são:

1. Na interface *web*, é possível criar uma nova sessão de Pomodoro, onde é possível escolher se uma sessão é pública ou privada e se uma sessão tem um modo de descanso com entretenimento ou sem entretenimento.
2. Na interface *web*, o usuário pode visualizar a lista de sessões públicas disponíveis.

3. Na interface *web*, o usuário pode entrar em uma sessão existente usando o identificador único da sala.

As telas referentes a esses casos podem ser visualizadas nas Figuras 4 e 5.



Figura 4: Tela da Criação de Sessão

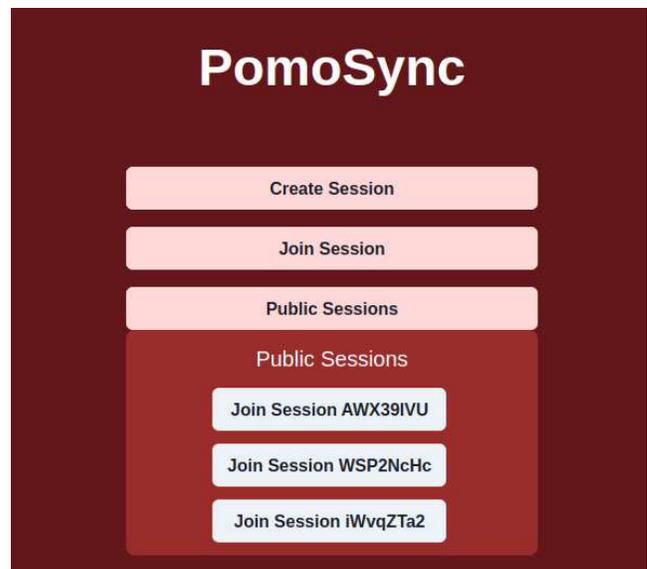


Figura 5: Tela da Listagem de Sessões Públicas

### 3.2.3 Sessão de Pomodoro

A Sessão de Pomodoro é o ambiente onde os usuários interagem durante a execução do Pomodoro de maneira sincronizada. Na Sessão de Pomodoro, os usuários podem iniciar ou pausar o timer do Pomodoro de forma simultânea, interagir com os outros participantes em tempo real por meio de um chat integrado,

visualizar a lista de participantes atualizada e enviar convites para se juntarem a uma sessão via *e-mail*. Os casos de uso relacionados à Sessão de Pomodoro são:

1. Em uma sessão, o usuário pode iniciar ou pausar o timer de Pomodoro, e essa ação será replicada para todos os usuários na sessão.
2. Em uma sessão, o usuário pode interagir com os outros participantes através do *chat* integrado em tempo real.
3. Em uma sessão, o usuário pode visualizar os outros participantes da sessão de maneira atualizada.
4. Em uma sessão, o usuário pode enviar convites para participar em uma sessão por *e-mail*.
5. Em uma sessão, o usuário pode configurar o tempo de estudo e de descanso.
6. Em uma sessão, caso o usuário tenha escolhido o modo entretenimento, será disponibilizado um vídeo a todos os usuários na sessão de forma síncrona.

As telas referentes a esses casos podem ser visualizadas nas Figuras 6, 7, 8 e 9.

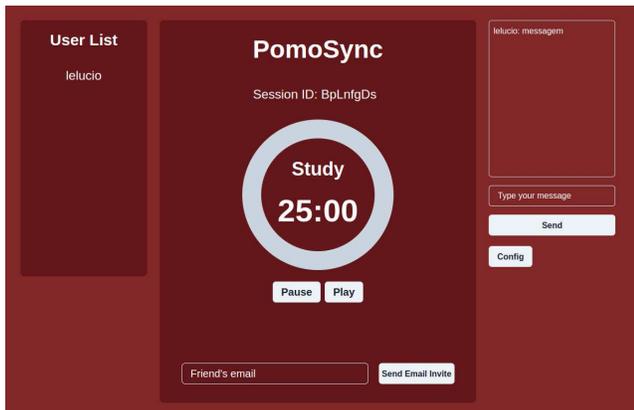


Figura 6: Tela de sessão de Pomodoro em modo estudo

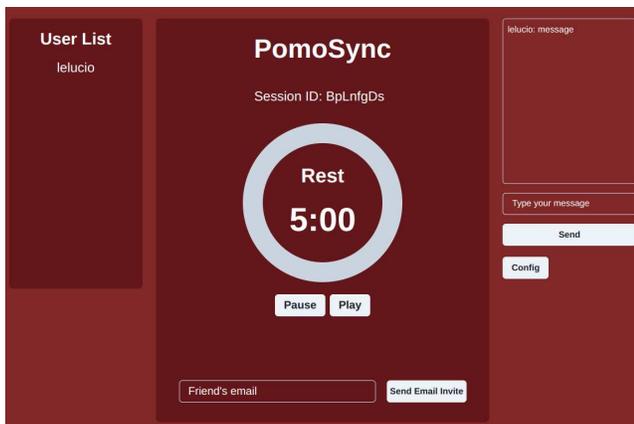


Figura 7: Tela de sessão de Pomodoro em modo de descanso sem modo entretenimento

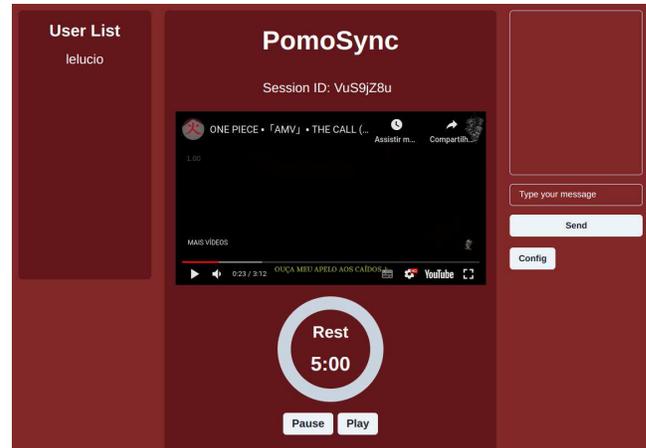


Figura 8: Tela de sessão de Pomodoro em modo de descanso com modo entretenimento

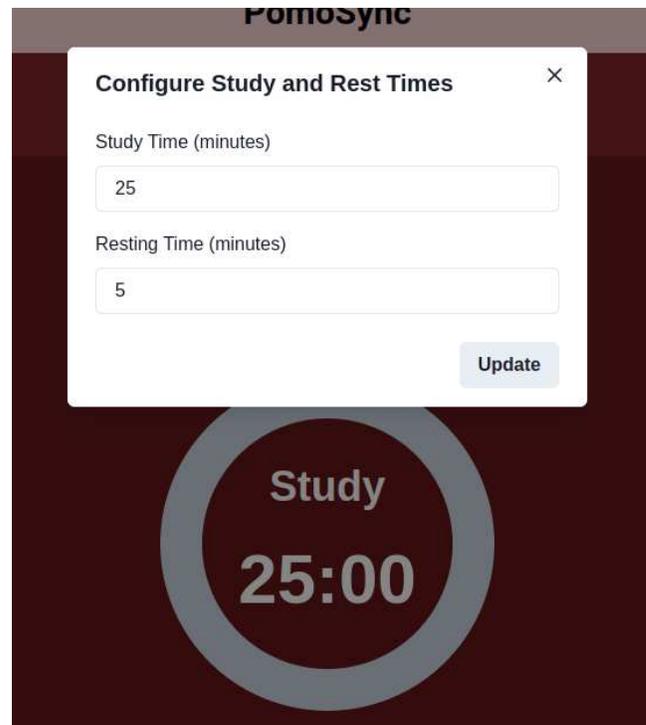


Figura 9: Configuração de tempo de pomodoro

#### 4. ARQUITETURA

O PomoSync é composto por duas partes fundamentais como visto na figura 8, o cliente (*front-end*) e o servidor (*back-end* Firebase), alinhando-se com a arquitetura cliente-servidor, amplamente adotada no desenvolvimento *web* moderno [12]. O cliente é encarregado de interpretar e executar as ações dos usuários relacionadas às sessões de Pomodoro, transmitindo essas informações para o servidor, o cliente também é responsável por apresentar, de forma organizada e intuitiva, os dados relevantes sobre as sessões de Pomodoro.

Já o servidor, por sua vez, além de gerenciar a comunicação entre os clientes e manter em memória os dados pertinentes a sessões e

usuários, também é responsável pela autenticação, utilizando o Firebase Authentication [6], que é um serviço de autenticação.

Essa interação entre cliente e servidor é estabelecida por meio de protocolos específicos - *Hypertext Transfer Protocol* (HTTP) [18] e *WebSocket* [1] - que permitem uma comunicação eficiente e em tempo real entre as diferentes partes do sistema. O servidor recebe os dados enviados pelo cliente, processa esses dados e envia a resposta para o cliente com os dados que serão necessários para o funcionamento do *front-end*, permitindo que os clientes sejam atualizados segundo as respostas recebidas. Para a troca de mensagens entre cliente e servidor, optou-se pelo uso do formato de dados *JavaScript Object Notation* (JSON) [21], muito utilizado por aplicações *web* em geral.

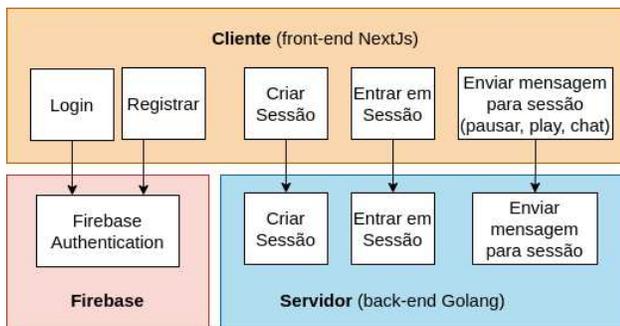


Figura 10: Arquitetura da aplicação

## 4.1 Servidor

O servidor desta aplicação desempenha um papel crucial ao administrar a autenticação, a comunicação e interações entre os clientes. Uma das suas principais responsabilidades é a manipulação de mensagens, que inclui a transmissão de mensagens para todos os clientes em uma sessão. Esta funcionalidade é implementada utilizando a tecnologia *WebSockets*, que se trata de uma interface que permite uma comunicação bidirecional em tempo real entre o cliente e o servidor.

Além da transmissão de mensagens, o *back-end* é também encarregado do processamento dessas mensagens. Isto inclui tarefas como a interpretação das ações do cliente (por exemplo, "pausar", "reproduzir", "chat"), as quais são encaminhadas e processadas de acordo. Outro exemplo de processamento é a formatação de tempo, essencial para manter a sincronização correta durante uma sessão. Por exemplo, quando uma sessão é pausada ou reproduzida, o tempo de início e o tempo decorrido da sessão são calculados para garantir a precisão.

Outra parte vital do servidor é a implementação do Firebase Authentication, um serviço do Google que fornece uma solução completa para autenticação. Com o Firebase Authentication foi integrado o método de autenticação utilizando *e-mail* e senha. A ferramenta também provê recursos como verificação de *e-mail* e recuperação de senha. O uso do Firebase oferece uma abstração robusta para autenticação, fazendo com que o código do *back-end* foque na lógica de processamento de mensagens e gerenciamento de sessões, possibilitando um processo desenvolvimento mais rápido do servidor.

Na construção do *back-end*, foi utilizada a linguagem de programação Golang [13], caracterizada por sua simplicidade, robustez, eficiência e, principalmente, facilidade de lidar com

paralelismo. Esta escolha foi complementada pelo uso do Gorilla API Toolkit [19], uma biblioteca em Golang que facilita a criação de *Application Programming Interface* (APIs) *web* sofisticadas e seguras. O Gorilla API Toolkit disponibiliza uma variedade de funcionalidades, incluindo gerenciamento de rotas e sessões, manipulação de *cookies* e suporte a *WebSockets*.

Por se tratar de um *back-end* em Golang, essa aplicação segue uma estrutura modularizada, como pode ser observado na figura 11. Essa estrutura modularizada facilita a organização do código e a manutenção da aplicação, permitindo que cada pasta e arquivo seja responsável por uma parte específica do projeto. As pastas estão organizadas da seguinte forma:

- Pasta config: Aqui, temos as configurações relacionadas ao Firestore, que se trata de um banco de dados NoSQL provido pelo Firebase. O arquivo `firestore.go` contém as definições e funcionalidades específicas para a configuração e interação com o Firestore.
- Pasta web: Nesta pasta, estão as configurações e arquivos relacionados à parte *web* da aplicação.
  - Pasta client: Aqui, encontramos o arquivo `client.go`, que contém as definições e funcionalidades específicas para o gerenciamento de usuários da aplicação *web*.
  - Arquivo `main.go`: Este arquivo é o ponto de entrada principal da aplicação. É aqui que a execução do programa é iniciada. Ele contém a lógica para configurar e inicializar o servidor *web*.
  - Pasta session: Aqui, estão as definições e funcionalidades específicas para a gestão de sessões de Pomodoro na aplicação.
  - Pasta websocket: Nesta pasta, encontramos as definições e funcionalidades específicas para a comunicação via *WebSockets* na aplicação.

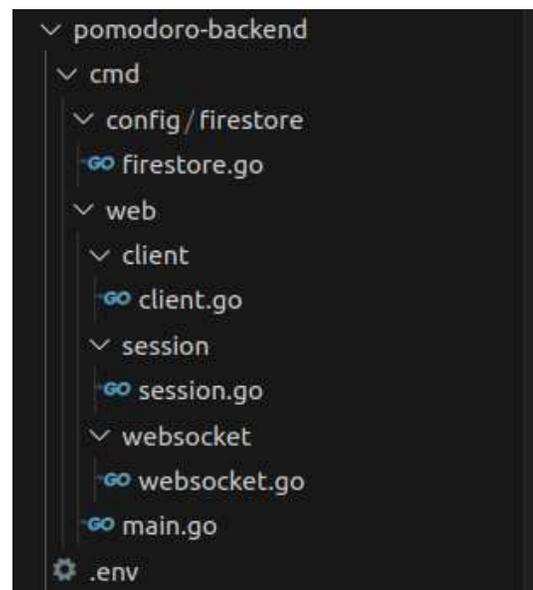


Figura 11: Estrutura de módulo do Back-end

## 4.2 Cliente

No desenvolvimento do cliente, foi utilizado o *framework* NextJs [11], que se trata de um *framework* baseado na biblioteca React [25], que oferece uma estrutura robusta e otimizações automáticas para construir o front-end de aplicações *web* rápidas, dinâmicas e interativas, oferecendo configurações automáticas das ferramentas. Também foi utilizado o TypeScript [14] como a linguagem de programação principal. O TypeScript foi escolhido devido à sua capacidade de facilitar o encapsulamento de componentes *web*, além de promover uma melhor leitura e compreensão do código por terceiros. Para a concepção visual dos componentes, a biblioteca ChakraUI [3] foi utilizada. Essa biblioteca fornece componentes React já estilizados, e permite uma integração eficaz entre *Cascading Style Sheets* (CSS) [4] e TypeScript, proporcionando maior versatilidade na criação de componentes dinâmicos.

No contexto de comunicação do *front-end* com o *back-end*, foram utilizados *WebSockets* nativos do TypeScript para estabelecer conexões persistentes. Isso possibilita uma comunicação bidirecional em tempo real entre cliente e servidor. Já em relação às chamadas HTTP, a biblioteca Axios [20] foi a escolhida. Trata-se de uma solução popular baseada em Promises<sup>1</sup> para efetuar requisições HTTP, destacando-se pela facilidade de uso e ampla compatibilidade com os padrões modernos da *web*.

Como *frameworks* de *front-end* não possuem uma regra restrita de arquitetura, foi escolhida uma arquitetura similar à modular para as pastas, como é possível observar na figura 12. Em “config” ficam os arquivos de configuração, como, por exemplo, o arquivo para conectar no Firebase Authentication. Já em “public” ficam os arquivos menos relacionados com código, como os arquivos de áudio que tocam na sessão de Pomodoro. Em “src” fica a parte central do código, e esta está dividida em quatro partes: “api”, onde fica toda lógica de conexões HTTP com o back-end; “components”, onde ficam todos os componentes visuais reutilizáveis da aplicação; “pages”, onde ficam todas as páginas da aplicação; e por fim “styles”, onde ficam todos os arquivos responsáveis pela estilização global da aplicação.

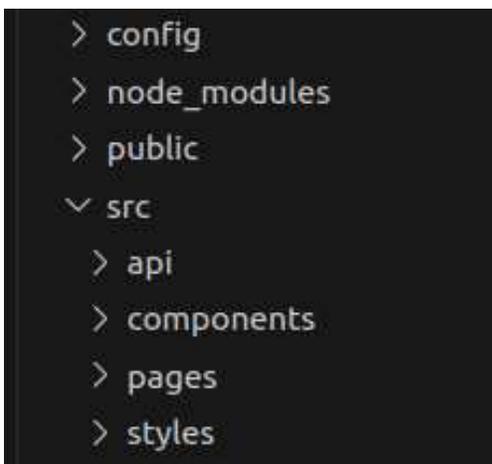


Figura 12: Estrutura de módulo do Front-end

## 5. SISTEMA EM USO

### 5.1 Metodologia

A ferramenta de criação de túneis seguros, ngrok [23], foi empregada para permitir a disponibilização de aplicações na internet de forma segura. Esta foi utilizada para possibilitar que a aplicação, operando em uma porta específica do computador do desenvolvedor, se tornasse acessível para outros usuários mediante um *link* gerado pelo ngrok.

O estudo contou com a participação de nove indivíduos, todos com perfil de estudante. Os usuários foram orientados a seguir um roteiro que englobava a criação de uma conta, o login na aplicação, a criação de uma sessão de Pomodoro e o envio de um convite a um terceiro para participar de uma sessão de Pomodoro. Esse roteiro foi desenvolvido com o intuito de permitir que os usuários pudessem explorar as principais funcionalidades da aplicação, mantendo a liberdade desses estudantes para descobrirem outros aspectos do sistema.

A avaliação da usabilidade e satisfação dos usuários foi conduzida através do questionário CSUQ (*Computer System Usability Questionnaire*), composto por 19 perguntas, cujas respostas variam em uma escala de 1 a 7, em que 1 é a pior avaliação e 7 é a melhor avaliação. As médias das respostas para cada pergunta compõem a pontuação final do CSUQ, refletindo a qualidade de usabilidade da aplicação.

O Google Forms [22] foi a ferramenta selecionada para coletar as respostas ao CSUQ de forma organizada e anônima. A análise das respostas revelou que a questão 1, "No geral, estou satisfeita ou satisfeito com quão fácil é utilizar o PomoSync?", obteve a menor média, como é possível visualizar na figura 13, indicando possíveis dificuldades enfrentadas pelos usuários na utilização da aplicação.

Algumas das dificuldades relatadas incluíram a falta de conhecimento sobre a técnica de Pomodoro, sugerindo a necessidade de inclusão de uma breve explicação sobre a técnica na página inicial da aplicação ou numa página "Sobre". Além disso, outros aspectos foram destacados para melhorias futuras, como aprimoramentos na estilização da página e na eficiência da aplicação.

Apesar da primeira pergunta ter registrado a menor pontuação, é importante salientar que, no contexto geral, a avaliação demonstrou um alto nível de satisfação dos usuários. A média obtida a partir de todas as respostas foi de 6,4 numa escala que vai de 1 até 7, onde 7 é a melhor resposta possível, indicando uma percepção positiva da usabilidade da aplicação. Tal constatação evidencia que, mesmo considerando as melhorias que podem ser implementadas, os usuários se sentem de modo geral satisfeitos com a experiência proporcionada pela aplicação.

1

[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Referen ce/Global\\_Objects/Promise](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Referen ce/Global_Objects/Promise)

Média das respostas do teste de usabilidade

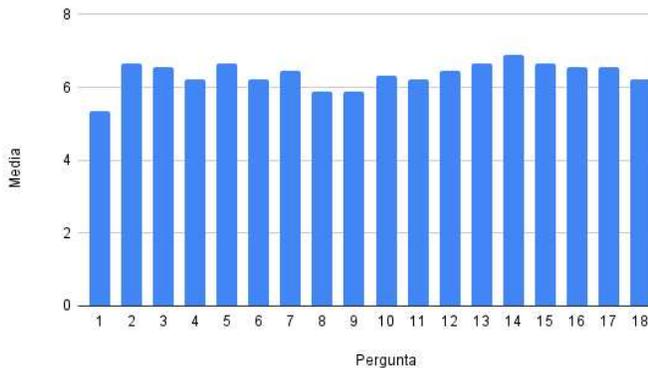


Figura 13: Médias dos valores das respostas do teste CSUQ realizado

## 6. EXPERIÊNCIAS E LIÇÕES APRENDIDAS

### 6.1 Processo de desenvolvimento

Para o desenvolvimento desta aplicação, foi essencial a aplicação de conhecimento diversificado, englobando o que foi aprendido durante o estudo de várias disciplinas que abrangem a concepção de software durante a graduação no curso de Ciência da Computação. O processo começou com a idealização teórica, onde se formularam as ideias iniciais e se delinearão os objetivos e funcionalidades principais do projeto, já explicados na seção 3.2. Em seguida, foram elaborados os casos de uso. Após isso, houve a escolha de tecnologias para a construção do sistema, pensando em paradigmas de programação que mais se encaixavam na solução desejada, e considerando as tecnologias mais populares utilizadas pelo mercado atualmente.

Finalmente, a fase de desenvolvimento do produto foi iniciada, adotando-se a metodologia Scrum [2]. Scrum é uma metodologia ágil para gestão e planejamento de projetos que promove a colaboração contínua, a adaptabilidade e a entrega de valor em incrementos contínuos. Nesta abordagem, o desenvolvimento foi dividido em sub tarefas, executadas em ciclos chamados 'sprints'. Foram planejados um total de cinco sprints, com uma duração de duas semanas cada para garantir um fluxo de trabalho contínuo e controlado. Ao final de cada sprint, era realizada uma reunião de revisão com o orientador, que assumiu o papel de *Product Owner*, pessoa responsável por gerenciar o produto conforme o Scrum. Nessas reuniões, avaliou-se o progresso do projeto e as atividades realizadas, redefinindo, se necessário, a trajetória para os próximos passos.

### 6.2 Desafios

O desafio mais significativo ao desenvolver esta aplicação foi assegurar uma comunicação em tempo real coerente e eficaz entre os usuários. Além disso, teve-se que administrar o paralelismo no servidor, ou seja, gerenciar requisições simultâneas de múltiplos clientes sem causar nenhum tipo de inconsistência nas respostas enviadas. Para solucionar esses problemas, foi crucial aplicar conhecimentos obtidos nas disciplinas de Redes de Computadores e Programação Concorrente.

Adicionalmente, outro desafio enfrentado foi a falta de familiaridade com certas tecnologias, incluindo Golang, dado que esta foi uma das primeiras experiências com o uso desta linguagem pelo desenvolvedor do projeto em questão. Mais ainda, para a construção da interface visual da aplicação, tornou-se necessário não apenas aprender brevemente sobre design de aplicativos, como também dominar o uso da biblioteca Chakra UI, a fim de auxiliar na criação da interface de usuário do sistema. Outro desafio envolveu o primeiro contato com tecnologias como o *WebSocket*.

## 7. CONSIDERAÇÕES FINAIS

### 7.1 Limitações e Trabalhos futuros

O feedback coletado dos usuários por meio do questionário CSUQ na seção 5.1 e a análise da experiência do usuário durante o estudo apontam para várias possíveis melhorias e recursos adicionais para trabalhos futuros. Estas melhorias visam potencializar a usabilidade e a experiência do usuário na plataforma. A implementação dessas mudanças e adições podem ser exploradas em pesquisas futuras.

As sugestões de melhorias e recursos adicionais incluem:

- Página informativa sobre a técnica do Pomodoro: a inclusão de uma explicação breve sobre a técnica de Pomodoro na página inicial ou em uma página "Sobre" para auxiliar usuários menos familiarizados com o método.
- Recurso de conexão social: um recurso que permita aos usuários adicionarem amigos e convidá-los para sessões diretamente através da plataforma, facilitando a interação e cooperação entre os usuários.
- Personalização do perfil do usuário: permitir ao usuário personalizar seu perfil, incluindo a opção de *upload* de uma foto pessoal, para uma experiência mais individualizada.
- Coleta e armazenamento de informações das sessões de Pomodoro: permitir análises e estatísticas de produtividade para oferecer percepções úteis para aprimoramento do desempenho dos usuários.
- Sistema de conquistas: introduzir um sistema de conquistas para incentivar a gamificação e o uso contínuo da plataforma.
- Classificação de sessões por tópicos: facilitar a conexão entre usuários com interesses de estudo semelhantes, potencializando o engajamento e a colaboração.

## 8. AGRADECIMENTOS

Quero agradecer à minha mãe, que sempre me apoiou durante a graduação. Também sou grato a Renato, que me incentivou a começar a programar e me motivou a entrar no curso. Minha namorada Mariana está sempre me apoiando, e, por isso, agradeço a ela. Queria também agradecer a Everton por ter me orientado. Por fim, quero agradecer a todos os meus amigos que fiz durante a graduação e sempre me apoiaram.

## REFERÊNCIAS

- [1] IETF (Internet Engineering Task Force). The WebSocket Protocol. RFC 6455. 2011. Retrieved May 16, 2023 from <https://datatracker.ietf.org/doc/html/rfc6455>.

- [2] [n.d.]. Scrum. Retrieved May 19, 2023 from <https://www.desenvolvimentoagil.com.br/scrum/>.
- [3] [n. d.]. Chakra UI. Retrieved May 16, 2023 from <https://chakra-ui.com/>.
- [4] [n. d.]. CSS. Retrieved May 16, 2023 from <https://developer.mozilla.org/pt-BR/docs/Web/CSS>.
- [5] [n. d.]. Figma: the collaborative interface design tool. Retrieved May 16, 2023 from <https://www.figma.com/>.
- [6] [n. d.]. Firebase Authentication. Retrieved May 08, 2023 from <https://firebase.google.com/docs/auth>.
- [7] [n. d.]. Firebase helps you build and run successful apps. Retrieved May 08, 2023 from <https://firebase.google.com>.
- [8] [n. d.]. GitHub: Where the world builds software. Retrieved May 16, 2023 from <https://github.com/>.
- [9] James R Lewis. 1995. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7, 1 (1995), 57-78.
- [10] Cirillo, Francesco. *The Pomodoro technique: The acclaimed time-management system that has transformed how we work*. Currency, 2018.
- [11] [n. d.]. NextJs. Next.js by vercel - the react framework. Retrieved from <https://nextjs.org/>.
- [12] TERRA, J. What is client-server architecture? everything you should know. Retrieved from <https://www.simplilearn.com/what-is-client-server-architecture-re-article>.
- [13] [n. d.]. Go Programming Language. Retrieved June 07, 2023 from <https://go.dev/>.
- [14] [n. d.]. TypeScript: JavaScript that scales. Retrieved June 07, 2023 from <https://www.typescriptlang.org/>.
- [15] [n. d.]. PomoDoneApp: Work with time, not against it. Retrieved June 07, 2023 from <https://pomodoneapp.com/>.
- [16] [n. d.]. Pomomo: Pomodoro Timer for Focus. Retrieved June 07, 2023 from <https://pomomo.us/>.
- [17] [n. d.]. PomoFocus: Simple Pomodoro timer to boost your productivity. Retrieved June 07, 2023 from <https://pomofocus.io/>.
- [18] IETF (Internet Engineering Task Force). Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616. 1999. Retrieved June 07, 2023 from <https://datatracker.ietf.org/doc/html/rfc2616>.
- [19] [n. d.]. Gorilla Web Toolkit. Retrieved June 07, 2023 from <https://www.gorillatoolkit.org/>.
- [20] [n. d.]. Axios: Promise based HTTP client for the browser and node.js. Retrieved June 07, 2023 from <https://axios-http.com/ptbr/docs/intro>.
- [21] [n. d.]. JSON: A Light-Weight, Language Independent, Data Interchange Format. Retrieved June 07, 2023 from <https://www.json.org/json-pt.html>.
- [22] [n. d.]. Google Forms utilizado para coleta de respostas do CSUQ. Retrieved June 07, 2023 from <https://forms.gle/1rXLRquKbSiUkZjZA>.
- [23] [n. d.]. ngrok: Secure Tunneling to localhost. Retrieved June 07, 2023 from <https://ngrok.com/>.
- [24] BetterUp. The Pomodoro Technique: How a break can improve productivity and well-being. Retrieved June 07, 2023 from <https://www.betterup.com/blog/pomodoro-technique>.
- [25] [n. d.]. React: A JavaScript library for building user interfaces. Retrieved June 07, 2023 from <https://react.dev/>.