FocusTrack: Ferramenta de Autogestão da Atenção Baseada em Indicadores Faciais

Reinan de Souza Santos

Antonio Carlos dos Santos Souza

Orientador

Instituto Federal da Bahia - IFBA
Curso de Análise e Desenvolvimento de Sistemas
Campus Salvador

Salvador, Bahia, Brasil

Agosto 2025

Sumário

1 Visão geral **1**
1.1 Declaração do Problema . . . 1
1.2 Proposta de Solução de Software . . . 2
1.3 Tecnologias Adotadas . . . 2
1.3.1 Back-end . . . 3
1.3.1.1 Python . . . 3
1.3.1.2 FastAPI . . . 3
1.3.2 Banco de Dados . . . 3
1.3.2.1 PostgreSQL . . . 3
1.3.3 Front-end . . . 3
1.3.3.1 TypeScript . . . 3
1.3.3.2 React . . . 4
1.3.3.3 Vite . . . 4
1.3.3.4 Chakra UI . . . 4
1.3.4 Versionamento . . . 4
1.3.4.1 Git . . . 4
1.3.4.2 GitHub . . . 4
1.3.5 Testes . . . 5
1.3.5.1 Pytest . . . 5
1.4 Trabalhos Relacionados . . . 5
1.4.1 Smart Eye . . . 5
1.4.2 Seeing Machines . . . 5
1.4.3 Headspace . . . 6
1.4.4 Pomofocus . . . 6
1.4.5 Tabela Comparativa . . . 7

2 Requisitos **7**
2.1 Requisitos Funcionais . . . 7
2.2 Requisitos Não-Funcionais . . . 7

3 Design **8**
3.1 Projeto UML . . . 8
3.1.1 Diagrama de Classe . . . 8
3.1.2 Diagrama de Caso de Uso . . . 8
3.1.3 Diagrama de Sequência . . . 9
3.2 Visão Arquitetural . . . 11
3.3 Modelo de Banco de Dados . . . 12
3.3.1 Modelo de Dados Lógico . . . 12

4 Qualidade **12**
4.1 Projeto de Testes . . . 12

5 Implantação **13**
5.1 Projeto de Implantação . . . 13

6 Manual do Usuário **15**

1 Visão geral

1.1 Declaração do Problema

A capacidade de manter a atenção em atividades cognitivas complexas é uma habilidade fundamental para o desempenho acadêmico e profissional, especialmente em um cenário cada vez mais mediado por tecnologias digitais. Com a popularização da internet e dos dispositivos conectados, surgiram novas formas de acesso ao conhecimento e à informação, mas também novos desafios relacionados à atenção e ao foco. A exposição constante a estímulos digitais fragmenta o tempo de concentração e compromete a produtividade, gerando consequências significativas para a saúde mental, o aprendizado e a eficiência no trabalho.

Estudos conduzidos pela professora Gloria Mark, da Universidade da Califórnia, revelam uma tendência alarmante: o tempo médio que uma pessoa consegue manter a atenção em uma única tarefa caiu de aproximadamente 150 segundos em 2004 para cerca de 47 segundos entre 2016 e 2021[1, 2]. Além disso, interrupções frequentes podem exigir até 25 minutos para que o cérebro recupere plenamente o estado de concentração anterior[2]. Esse fenômeno, conhecido como fragmentação cognitiva, afeta particularmente estudantes e profissionais que desempenham atividades em ambientes domésticos ou sem supervisão direta, onde os limites entre foco e distração são menos definidos. A pandemia de COVID-19 agravou esse cenário ao acelerar a transição para o ensino remoto e o trabalho em home office. Segundo um estudo com 195 estudantes de graduação nos Estados Unidos, 89% relataram dificuldades de concentração relacionadas ao excesso de estímulos, distúrbios do sono e aumento do estresse. Outros 86% apontaram piora no padrão de sono, o que impacta diretamente na capacidade atencional [3].

O avanço tecnológico e a popularização de dispositivos conectados à internet transformaram profundamente a maneira como pessoas estudam, trabalham e se comunicam. Se, por um lado, o ambiente digital oferece inúmeras ferramentas para o aprendizado, por outro, também introduz uma série de estímulos que competem pela atenção do indivíduo, contribuindo para a fragmentação cognitiva. A capacidade de manter o foco tornou-se um dos principais desafios contemporâneos, sobretudo entre estudantes e profissionais que desempenham tarefas intelectualmente exigentes em ambientes domésticos ou sem supervisão direta.

Diante desse contexto, torna-se evidente a necessidade de estratégias e ferramentas capazes de auxiliar na autorregulação da atenção, especialmente em ambientes digitais. A atenção sustentada pode ser inferida por meio de parâmetros fisiológicos e comportamentais, como a frequência de piscadas, a direção do olhar, a postura da cabeça e a medida PERCLOS (Percentage of Eye Closure), que têm sido utilizados com sucesso em estudos de fadiga e distração no trânsito ou em exames supervisionados [4][5]. Entretanto, ainda são escassas as soluções que utilizam esses recursos de maneira acessível e em tempo real, aplicando-os em cenários cotidianos de estudo e trabalho remoto.

Dessa forma, é possível observar uma lacuna significativa na disponibilidade de sistemas que promovam o monitoramento passivo da concentração, de forma contínua, personalizada

e não invasiva. A maior parte das ferramentas atuais se limita a abordagens comportamentais, como timers ou bloqueadores de sites, desconsiderando sinais fisiológicos que poderiam enriquecer a análise da atenção. Frente a esse cenário, este trabalho propõe o desenvolvimento de um sistema de monitoramento de concentração baseado em visão computacional, com o objetivo de detectar sinais de distração e alertar o usuário em tempo real. A proposta visa ampliar o acesso a soluções de autorregulação da atenção, contribuindo para a melhoria do desempenho cognitivo, a redução do estresse e a promoção do bem-estar em contextos de estudo e trabalho remoto.

1.2 Proposta de Solução de Software

Diante das dificuldades enfrentadas por estudantes e profissionais para manter a concentração em ambientes digitais, o FocusTrack surge como uma solução tecnológica voltada ao monitoramento da atenção durante atividades de estudo ou trabalho. Seu principal objetivo é oferecer suporte à autorregulação da concentração por meio de análise contínua e não invasiva de sinais comportamentais, utilizando técnicas de visão computacional.

O sistema busca identificar indícios de distração com base em parâmetros como direção do olhar, frequência de piscadas e postura da cabeça, a partir da captação da webcam do dispositivo do usuário. A interface é pensada para ser simples, discreta e funcional, permitindo que o monitoramento ocorra em segundo plano, sem interferir no fluxo de trabalho. Ao detectar sinais consistentes de desatenção, o FocusTrack emite alertas visuais ou sonoros, com o intuito de estimular o retorno do foco de maneira leve e imediata.

Além disso, o sistema armazenará os registros de atenção em um banco de dados, possibilitando a geração de relatórios com métricas de concentração ao longo do tempo. Essa funcionalidade visa auxiliar o usuário a compreender seus padrões de atenção e a identificar fatores que impactam negativamente sua produtividade. Com isso, o FocusTrack não apenas atua em tempo real, mas também fornece subsídios para que o próprio usuário adote estratégias mais eficazes de organização e estudo.

O propósito central do sistema é contribuir para a criação de ambientes digitais mais produtivos e saudáveis, por meio de uma abordagem acessível, personalizada e baseada em evidências. Dessa forma, o FocusTrack pretende ser um aliado no cotidiano de quem busca melhorar seu desempenho cognitivo e manter o foco diante das múltiplas distrações que caracterizam o contexto atual.

1.3 Tecnologias Adotadas

O desenvolvimento do FocusTrack contou com um conjunto de tecnologias escolhidas com base nas necessidades específicas da aplicação. Como o projeto envolve a detecção de sinais de atenção por meio de vídeo, além de uma interface web para interação com o usuário, foi essencial adotar ferramentas que fossem eficientes, bem documentadas e com boa aceitação na comunidade. As decisões tecnológicas também levaram em conta a familiaridade do autor com as ferramentas, buscando um equilíbrio entre desempenho, facilidade de uso e manutenção do sistema.

1.3.1 Back-end

1.3.1.1 Python

Python [6] é uma linguagem de programação de alto nível, interpretada e de tipagem dinâmica, amplamente utilizada no desenvolvimento de aplicações web, automação de tarefas, ciência de dados e inteligência artificial. Sua sintaxe simples e legível facilita a manutenção do código e acelera o desenvolvimento, especialmente em projetos de pequeno a médio porte. No contexto do FocusTrack, Python foi adotado pela sua vasta comunidade, ampla disponibilidade de bibliotecas e por oferecer suporte robusto ao desenvolvimento de APIs.

1.3.1.2 FastAPI

FastAPI [7] é um framework moderno e de alto desempenho para a construção de APIs com Python. Baseado no padrão ASGI e com suporte nativo a operações assíncronas, permite a criação de aplicações leves, rápidas e seguras. O framework também oferece geração automática de documentação interativa por meio do OpenAPI (Swagger), facilitando a visualização e o teste das rotas durante o desenvolvimento. No FocusTrack, o FastAPI foi utilizado para estruturar a camada de back-end, oferecendo os serviços necessários à interface web e ao processamento dos dados faciais recebidos.

1.3.2 Banco de Dados

1.3.2.1 PostgreSQL

PostgreSQL [8] é um sistema gerenciador de banco de dados relacional (SGBD) de código aberto, reconhecido pela sua robustez, conformidade com o padrão SQL e extensibilidade. Possui suporte a tipos de dados avançados, operações complexas e controle de concorrência, sendo uma escolha confiável tanto para aplicações simples quanto para sistemas empresariais. No FocusTrack, o PostgreSQL foi utilizado para armazenar e organizar os dados estruturados da aplicação, como usuários, sessões e registros de avaliação. Sua estabilidade e maturidade foram determinantes para garantir integridade nas transações e facilidade na manutenção da base de dados.

1.3.3 Front-end

1.3.3.1 TypeScript

TypeScript [9] é uma linguagem que estende o JavaScript com recursos de tipagem estática e orientação a objetos. Sua principal vantagem está na detecção precoce de erros durante o desenvolvimento, o que contribui para a criação de aplicações mais confiáveis e fáceis de manter. No FocusTrack, o uso do TypeScript proporcionou maior segurança no desenvolvimento da interface, além de melhorar a integração com bibliotecas e editores de código, especialmente em um projeto com múltiplos componentes reutilizáveis.

1.3.3.2 React

React [10] é uma biblioteca JavaScript para construção de interfaces de usuário baseadas em componentes. Ela permite a criação de interfaces reativas e modulares, com atualização eficiente da visualização em resposta a mudanças de estado. No FocusTrack, React foi escolhido por sua flexibilidade e pela vasta comunidade de suporte, o que facilitou o desenvolvimento de uma interface interativa, leve e com boa experiência para o usuário final.

1.3.3.3 Vite

Vite [11] é uma ferramenta de build moderna que oferece inicialização rápida, recarregamento instantâneo e compilação eficiente para projetos front-end. Ao contrário de ferramentas tradicionais como Webpack, o Vite utiliza o poder dos módulos ES nativos e otimizações inteligentes para melhorar a performance durante o desenvolvimento e a construção final da aplicação. No projeto FocusTrack, foi utilizado como ambiente de desenvolvimento e empacotamento da interface em React, contribuindo para maior agilidade no ciclo de desenvolvimento.

1.3.3.4 Chakra UI

Chakra UI [12] é uma biblioteca de componentes acessíveis e estilizados para React, voltada à criação rápida de interfaces modernas e responsivas. Ela fornece um conjunto de componentes pré-construídos e altamente personalizáveis, além de seguir boas práticas de acessibilidade. No FocusTrack, o Chakra UI facilitou a construção de uma interface limpa e consistente, reduzindo o tempo necessário para estilização manual e promovendo uma melhor experiência para o usuário.

1.3.4 Versionamento

1.3.4.1 Git

Git [13] é um sistema de controle de versão distribuído amplamente utilizado no desenvolvimento de software. Ele permite o rastreamento de alterações no código, o gerenciamento de diferentes versões e a colaboração segura em projetos. No desenvolvimento do FocusTrack, o Git foi essencial para manter um histórico organizado do projeto, possibilitando testes e alterações sem comprometer a base principal do código.

1.3.4.2 GitHub

GitHub [14] é uma plataforma online para hospedagem de repositórios Git, oferecendo recursos adicionais como controle de issues, revisão de código, integração com pipelines de automação e colaboração remota. Durante o desenvolvimento do FocusTrack, o GitHub foi utilizado para armazenar o código-fonte da aplicação e organizar o fluxo de trabalho, facilitando o gerenciamento das versões e o backup seguro do projeto.

1.3.5 Testes

1.3.5.1 Pytest

Pytest [15] é um framework de testes para Python que permite a criação de testes unitários, funcionais e de integração de forma simples e eficiente. Sua sintaxe enxuta e o suporte a fixtures facilitam a organização e reutilização do código de teste. No FocusTrack, o pytest foi adotado para garantir a qualidade do código do back-end, assegurando que as funcionalidades implementadas se comportem conforme o esperado e possibilitando a detecção precoce de eventuais falhas durante o desenvolvimento.

1.4 Trabalhos Relacionados

O FocusTrack é uma ferramenta que busca auxiliar o usuário no desenvolvimento da autogestão da atenção a partir da análise de expressões faciais captadas por webcam. Embora essa proposta apresente um foco específico, outras soluções já utilizam técnicas similares, especialmente em áreas como segurança no trânsito e produtividade pessoal. Esta seção apresenta um panorama de sistemas relacionados que também empregam indicadores faciais ou abordagens comportamentais para monitoramento de atenção, comparando suas funcionalidades e objetivos com os do FocusTrack.

1.4.1 Smart Eye

Smart Eye [16] é uma tecnologia embarcada em veículos que realiza o rastreamento ocular e facial do motorista em tempo real. Utilizada por montadoras e na indústria automotiva, a ferramenta detecta sinais de fadiga, distração e comportamento de risco, contribuindo para a prevenção de acidentes.

Vantagens:

- Alta precisão na detecção de atenção e fadiga;
- Utilização consolidada em aplicações de segurança veicular;
- Funciona em tempo real com análise de múltiplos vetores faciais.

Desvantagens:

- Exige hardware especializado;
- Custo elevado e foco exclusivo no setor automotivo.

1.4.2 Seeing Machines

Seeing Machines [17] desenvolve sistemas de monitoramento de atenção baseados em visão computacional para frotas comerciais, ônibus, aviões e veículos de passeio. Seus algoritmos detectam sonolência, distração e ausência visual da estrada.

Vantagens:

- Foco em segurança preventiva no transporte;
- Análise contínua e precisa em ambientes críticos;
- Reconhecimento facial robusto sob diferentes condições de iluminação.

Desvantagens:

- Requer sensores e câmeras embarcadas;
- Aplicabilidade restrita a contextos industriais ou automotivos.

1.4.3 Headspace

Headspace [18] é um aplicativo amplamente utilizado para meditação e atenção plena. Ele oferece sessões guiadas para foco, produtividade, sono e redução do estresse, com base em práticas de mindfulness acessíveis e adaptáveis ao cotidiano.

Vantagens:

- Conteúdo voltado à concentração, com trilhas específicas para foco;
- Interface amigável e visualmente atraente;
- Disponível em várias plataformas e com ampla aceitação global.

Desvantagens:

- Não utiliza sensores ou análise de indicadores faciais;
- Requer assinatura para acesso completo ao conteúdo.

1.4.4 Pomofocus

Pomofocus [19] é uma aplicação web baseada na técnica Pomodoro, que visa melhorar a concentração e o controle do tempo. Ela divide o trabalho em ciclos de foco e pausa, incentivando a produtividade por meio de gestão consciente do tempo.

Vantagens:

- Simples, gratuita e acessível por navegador;
- Ajuda a criar hábitos de concentração com reforço visual e sonoro;
- Popular entre estudantes e trabalhadores remotos.

Desvantagens:

- Não realiza nenhuma forma de monitoramento de atenção;
- Baseado apenas na organização do tempo, sem análise comportamental.

1.4.5 Tabela Comparativa

Características	Smart Eye	Seeing Machines	Headspace	Pomofocus	FocusTrack
Monitoramento Facial	Sim	Sim	Não	Não	Sim
Feedback em Tempo Real	Sim	Sim	Não	Não	Sim
Requer Hardware Especial	Sim	Sim	Não	Não	Não
Interface Amigável	Não	Parcialmente	Sim	Sim	Sim
Custo	Alto	Alto	Pago (com versão gratuita)	Pago (com versão gratuita)	Gratuito/Open Source
Uso Principal	Segurança veicular	Transporte comercial	Autocuidado e foco	Gestão de tempo	Autogestão da atenção

Tabela 1: Comparação entre ferramentas e aplicações relacionadas ao FocusTrack.

2 Requisitos

2.1 Requisitos Funcionais

RF1 Como usuário, desejo iniciar e encerrar sessões de monitoramento de atenção para acompanhar meu nível de foco durante atividades.

RF2 Como usuário, desejo receber alertas visuais sempre que o sistema detectar sinais de distração ou sonolência, para que eu possa retomar o foco.

RF3 Como usuário, desejo visualizar estatísticas ao final da sessão, como tempo total focado, distrações detectadas e gráficos de desempenho.

RF4 Como usuário, desejo configurar os limites de tolerância para distração e sonolência, para que o sistema se adapte ao meu perfil de uso.

RF5 Como usuário, desejo acessar um histórico das sessões realizadas, com seus respectivos resumos e métricas.

2.2 Requisitos Não-Funcionais

RNF1 O sistema deve ser acessado diretamente pelo navegador (plataforma web), sem necessidade de instalação, e ser compatível com os principais navegadores modernos.

RNF2 O sistema deve utilizar a câmera do usuário apenas durante as sessões ativas, mediante autorização explícita, garantindo privacidade.

RNF3 O sistema deve apresentar interface simples, responsiva e usável em dispositivos desktop e mobile.

RNF4 O sistema deve oferecer feedback em tempo real, com baixo tempo de resposta na detecção de distração ou sonolência.

RNF5 O sistema deve ser desenvolvido utilizando tecnologias de código aberto, promovendo transparência e reprodutibilidade.

RNF6 O sistema deve estar disponível gratuitamente para usuários finais.

3 Design

3.1 Projeto UML

3.1.1 Diagrama de Classe

Figura 1: Diagrama de Classe do sistema. Fonte: Elaborado pelo autor com auxílio da ferramenta Mermaid (2025).

3.1.2 Diagrama de Caso de Uso

O Diagrama de Caso de Uso mostra, de forma visual, as principais funcionalidades do sistema e como os usuários interagem com elas. Ele ajuda a entender o que o sistema deve fazer e quais são os papéis dos usuários nessas interações.

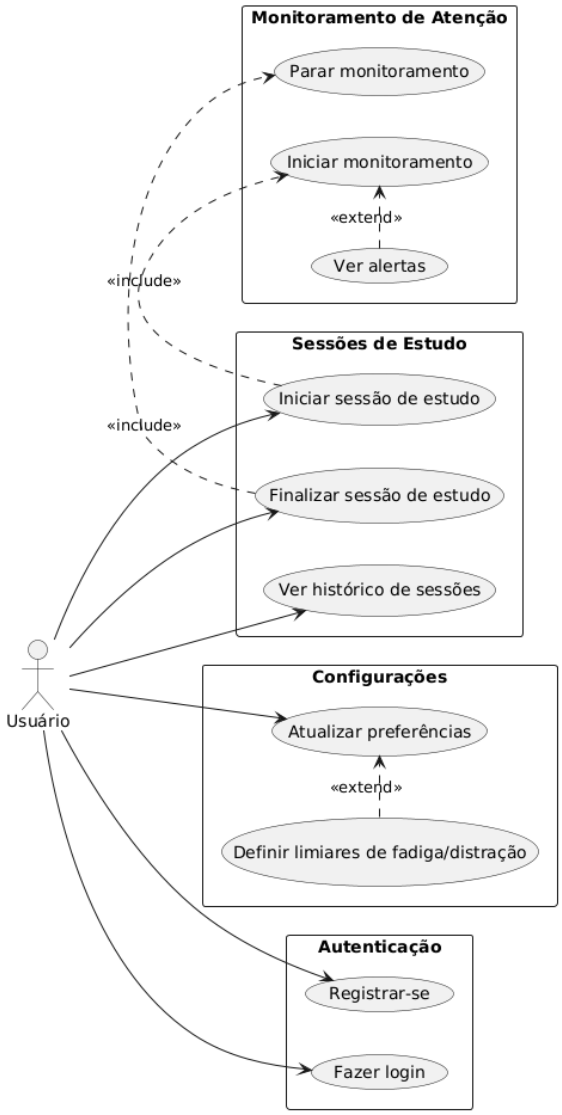

Figura 2: Diagrama de Caso de Uso do sistema. Fonte: Elaborado pelo autor com auxílio da ferramenta PlantText (2025).

Nota: Todos os casos de uso (exceto ”Registrar-se”e ”Fazer login”) pressupõem que o usuário já esteja autenticado no sistema.

3.1.3 Diagrama de Sequência

Os diagramas de sequência constituem ferramentas visuais fundamentais para representar as interações dinâmicas entre os componentes de um sistema ao longo do tempo. Este diagrama em particular descreve o fluxo completo do processo de monitoramento de atenção e fadiga, documentando a sequência de operações desde o estabelecimento da conexão WebSocket até o processamento de cada frame de vídeo e a finalização adequada da sessão.

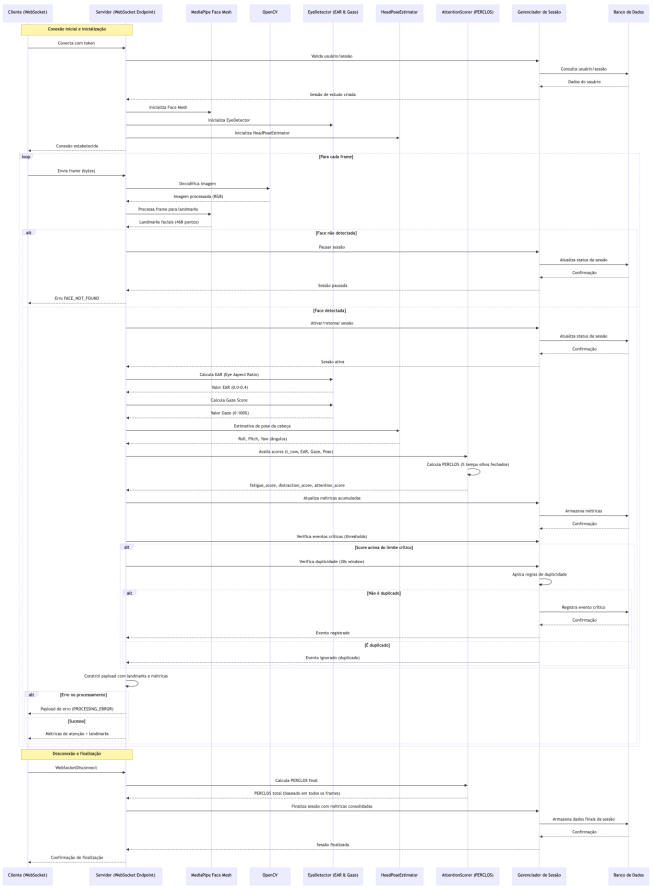

Figura 3: Diagrama de Sequência do processo de monitoramento de atenção e fadiga. Fonte: Elaborado pelo autor com auxílio da ferramenta Mermaid (2025).

O diagrama demonstra como:

- A conexão WebSocket é estabelecida e os componentes são inicializados;
- Cada frame é processado sequencialmente para detecção facial e cálculo de métricas;
- Os algoritmos de EAR (Eye Aspect Ratio) e PERCLOS são aplicados para avaliação da atenção;
- Eventos críticos são detectados e registrados quando os limiares são excedidos;
- A sessão é finalizada adequadamente com o cálculo das métricas finais.

Nota: O diagrama evidencia a integração em tempo real entre os componentes de visão computacional (MediaPipe, OpenCV) e os algoritmos especializados de detecção de fadiga (EAR, PERCLOS), destacando a arquitetura em pipeline do sistema.

3.2 Visão Arquitetural

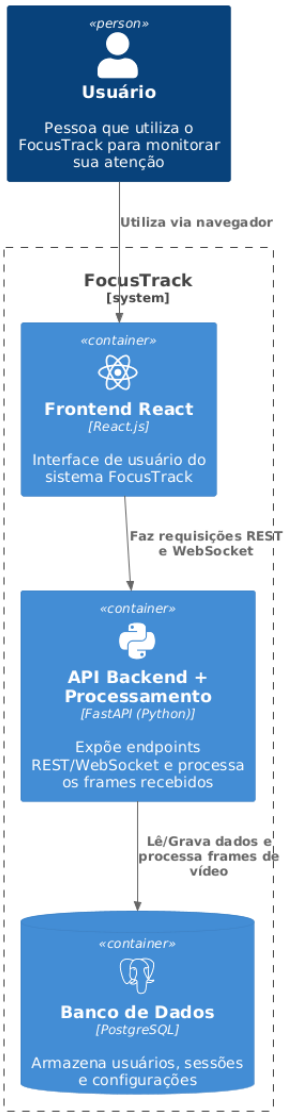

Figura 4: Diagrama de Container — Modelo C4. Fonte: Elaborado pelo autor com auxílio da ferramenta PlantText (2025).

3.3 Modelo de Banco de Dados

3.3.1 Modelo de Dados Lógico

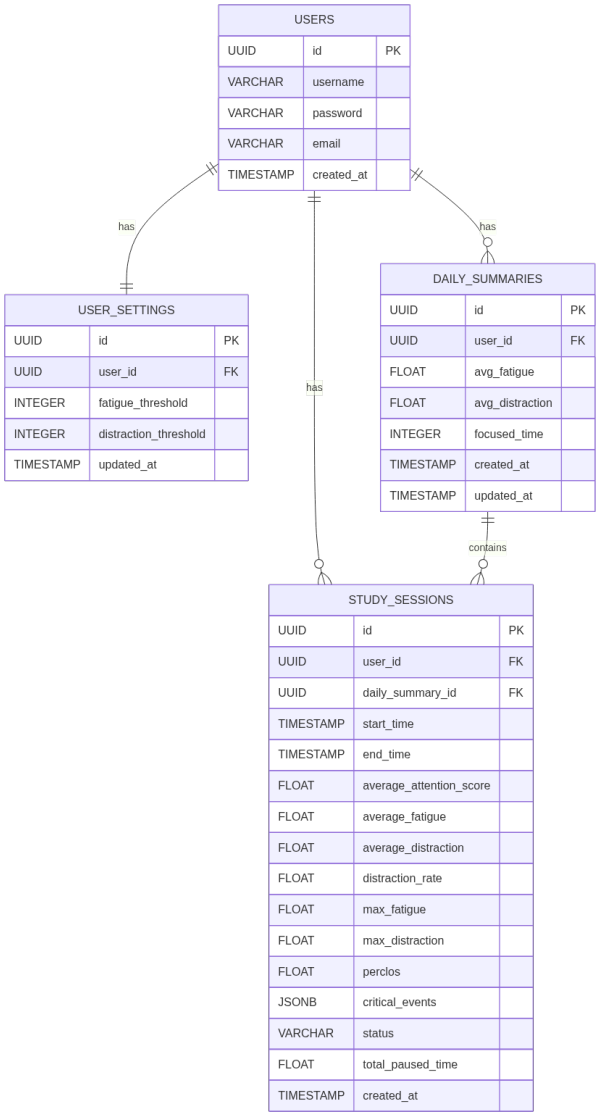

Figura 5: Diagrama Entidade-Relacionamento. Fonte: Elaborado pelo autor com auxílio da ferramenta Mermaid (2025).

4 Qualidade

4.1 Projeto de Testes

Para garantir que os principais componentes do *FocusTrack* estivessem funcionando corretamente de forma independente, desenvolvi testes unitários utilizando o framework *Pytest*, uma das ferramentas mais populares da comunidade Python.

Executar os testes é simples, basta rodar o seguinte comando na raiz do projeto:

```
task test
```

Além disso, configurei a automação dos testes com integração contínua por meio do *GitHub Actions*. Dessa forma, a cada nova alteração no código, os testes são executados automaticamente, ajudando a manter a estabilidade do sistema ao longo do desenvolvimento.

5 Implantação

5.1 Projeto de Implantação

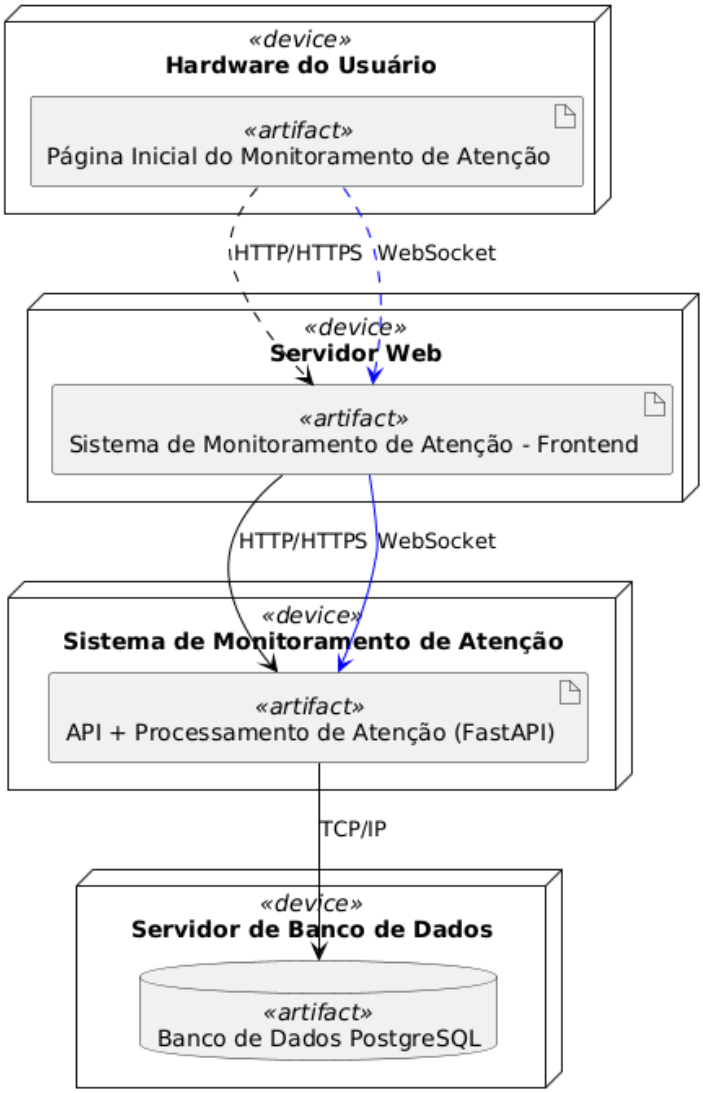

Figura 6: Diagrama de implantação. Fonte: Elaborado pelo autor com auxílio da ferramenta PlantText (2025).

O projeto *FocusTrack* foi desenvolvido para ser acessado por notebooks e computadores de mesa, por meio de navegadores web. Essa escolha se deve à necessidade do uso da webcam para realizar as análises de concentração.

A aplicação segue o fluxo descrito na Figura 6, dividido nas seguintes etapas:

1. **Acesso ao sistema**: O usuário entra na aplicação por meio de um navegador web que tenha suporte à webcam, requisito essencial para o funcionamento da ferramenta.
2. **Renderização da interface**: A interface gráfica da aplicação é carregada no navegador, permitindo que o usuário visualize e interaja com as funcionalidades disponíveis.
3. **Comunicação com o servidor**: Sempre que o usuário executa alguma ação, como iniciar ou encerrar uma análise, uma requisição é enviada ao servidor (back-end), que valida as informações e executa a lógica necessária.
4. **Retorno das informações**: Com a requisição processada, o sistema realiza as consultas no banco de dados (se necessário) e envia uma resposta de volta ao frontend, atualizando a interface conforme o resultado.

A arquitetura do sistema foi dividida em duas partes principais: **frontend** e **backend**. Cada parte foi implantada separadamente em seu respectivo ambiente de hospedagem, utilizando o banco de dados *PostgreSQL* da plataforma *Supabase*.

Código-fonte:

- FocusTrack API
- FocusTrack Frontend

Ambiente de Produção:

- FocusTrack Frontend
- FocusTrack Backend

Por se tratar de um projeto pessoal e acadêmico, foram utilizados planos gratuitos nas plataformas de hospedagem. As soluções adotadas incluem:

- **Supabase**: plano gratuito com requisições ilimitadas, 500MB de banco de dados e 1GB para armazenamento de arquivos.
- **Fly.io**: modelo *pay-as-you-go*, no qual é possível utilizar gratuitamente uma máquina de pequeno porte, ideal para hospedar a API do projeto.
- **Vercel**: plano *Hobby*, que oferece até 4 horas de uso ativo da CPU e análise de eventos para os primeiros 50 mil registros, suficiente para hospedar o frontend de forma eficiente.

6 Manual do Usuário

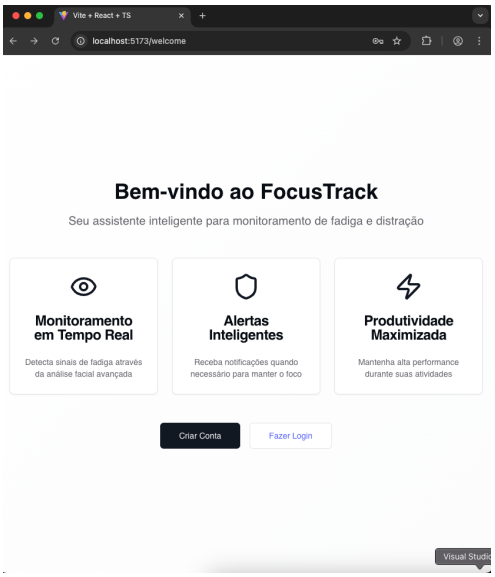

Figura 7: Tela inicial do FocusTrack

- A tela inicial apresenta duas opções principais: criar uma nova conta ou fazer login na plataforma.

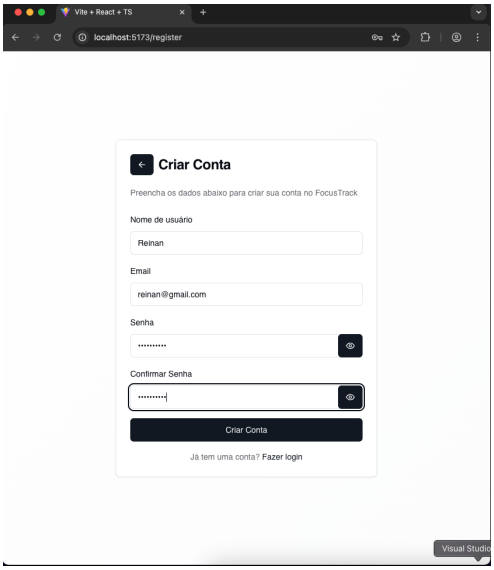

Figura 8: Tela de cadastro do FocusTrack

- Ao acessar a área de cadastro, o usuário encontra os campos necessários para criar sua conta e começar a utilizar o sistema.

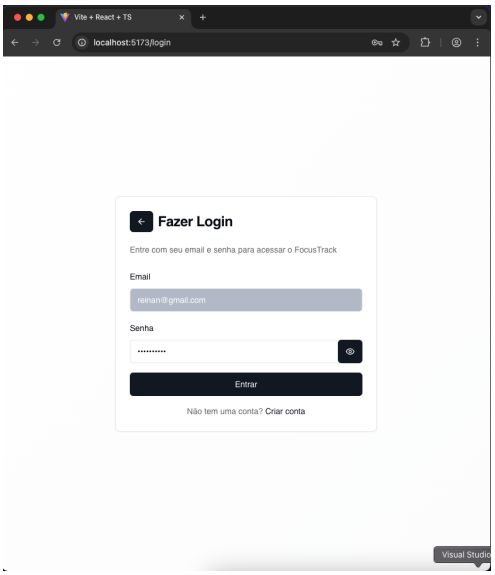

Figura 9: Tela de login do FocusTrack

- A tela de login permite que o usuário informe seu e-mail e senha para acessar o sistema. Também há um botão para confirmar a entrada.

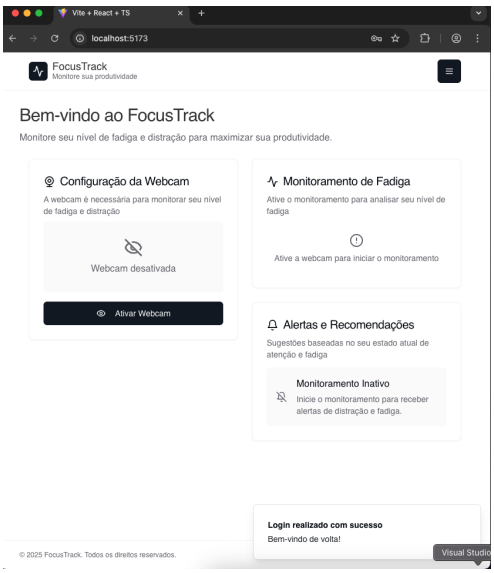

Figura 10: Dashboard do FocusTrack

- A interface do dashboard é dividida em três seções principais: configuração da webcam, monitoramento de fadiga e alertas.

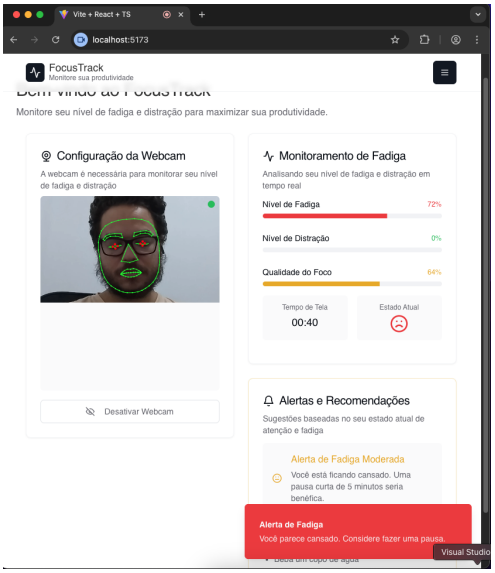

Figura 11: Dashboard com aviso de fadiga

- Quando o sistema detecta sinais de fadiga acima de determinado nível, o usuário é alertado visualmente dentro do dashboard.

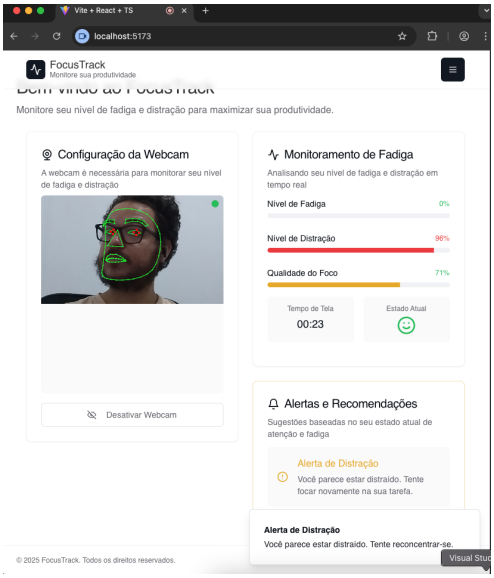

Figura 12: Dashboard com aviso de distração

- De forma semelhante, quando o nível de distração atinge um limite predefinido, o sistema emite um aviso para o usuário.

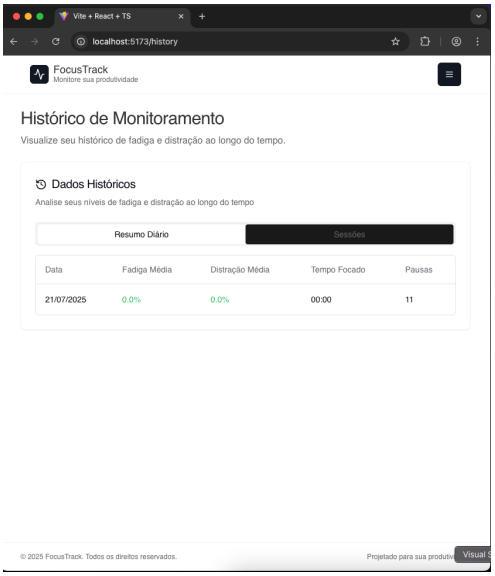

Figura 13: Histórico de monitoramento — resumo diário

- O resumo diário apresenta uma visão agrupada das sessões de monitoramento realizadas em cada dia.

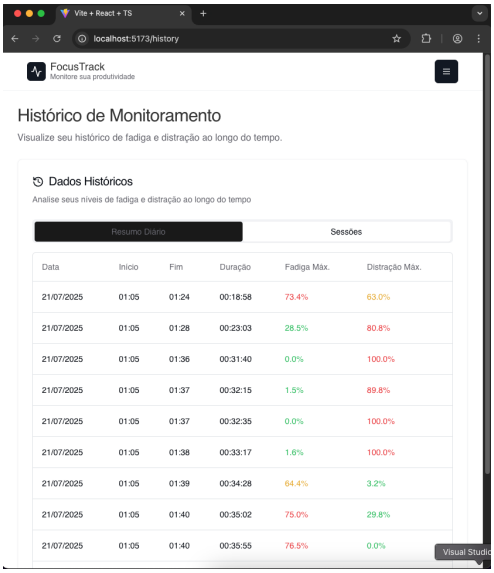

Figura 14: Histórico de monitoramento — sessões

- Na aba de sessões é exibida uma tabela com todos os registros detalhados, contendo colunas com as informações coletadas em cada análise.

Figura 15: Página de configurações

- A área de configurações permite que o usuário personalize os níveis de alerta, escolhendo quando deseja ser notificado sobre fadiga ou distração.

Agradecimentos

Agradeço, em primeiro lugar, a Deus por nunca ter me deixado desamparado e por colocar pessoas incríveis ao meu lado ao longo dessa jornada.

Sou profundamente grato à minha esposa, Ingred, por sua presença constante, apoio incondicional e por me incentivar a continuar mesmo nos momentos mais difíceis. Ao meu orientador, deixo meu sincero agradecimento pela paciência, orientação e disponibilidade sempre que precisei.

Também agradeço aos amigos que fiz durante esse período e que levarei comigo para a vida toda.

Meu muito obrigado a todos que, de alguma forma, fizeram parte da minha trajetória no IFBA e contribuíram para o meu crescimento pessoal e acadêmico.

Referências

1 MARK, G.; IQBAL, S. T.; CZERWINSKI, M. The cost of interrupted work: More speed and stress. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. [S.l.]: ACM, 2018.

2 MARK, G. **Attention Span: A Groundbreaking Way to Restore Balance, Happiness**

and Productivity. New York: Hanover Square Press, 2023.

3 SON, C. et al. Effects of COVID-19 on College Students' Mental Health in the United States: Interview Survey Study. **Journal of Medical Internet Research**, v. 22, n. 9, p. e21279, 2020. Disponível em: ⟨https://www.jmir.org/2020/9/e21279⟩.

4 JOHNS, M. W. A sleep physiologist's view of the drowsy driver. **Transportation Research Part F: Traffic Psychology and Behaviour**, v. 3, n. 4, p. 241–249, 2000.

5 VICENTE, F. et al. Driver monitoring: A review of vision-based techniques for fatigue and distraction detection. **Journal of Sensors**, v. 2016, p. 1–16, 2016.

6 Python Software Foundation. **Python**. 2025. Disponível em: ⟨https://www.python.org⟩.

7 Tiangolo. **FastAPI**. 2025. Disponível em: ⟨https://fastapi.tiangolo.com⟩.

8 PostgreSQL Global Development Group. **PostgreSQL**. 2025. Disponível em: ⟨https://www.postgresql.org⟩.

9 Microsoft. **TypeScript**. 2025. Disponível em: ⟨https://www.typescriptlang.org⟩.

10 Meta Platforms, Inc. **React**. 2025. Disponível em: ⟨https://react.dev⟩.

11 Evan You. **Vite**. 2025. Disponível em: ⟨https://vite.dev⟩.

12 Chakra UI Team. **Chakra UI**. 2025. Disponível em: ⟨https://chakra-ui.com⟩.

13 Software Freedom Conservancy. **Git**. 2025. Disponível em: ⟨https://git-scm.com⟩.

14 GitHub, Inc. **GitHub**. 2025. Disponível em: ⟨https://github.com⟩.

15 pytest-dev. **pytest Documentation**. 2025. Disponível em: ⟨https://docs.pytest.org/en/stable/⟩.

16 Smart Eye. **Smart Eye**. 2025. Disponível em: ⟨https://www.smarteye.se⟩.

17 Seeing Machines. **Seeing Machines — Technology to get everyone home safe**. 2025. Disponível em: ⟨https://seeingmachines.com⟩.

18 Headspace. **Headspace: Meditation and Sleep Made Simple**. 2025. Disponível em: ⟨https://www.headspace.com⟩.

19 Pomofocus. **Time to focus!** 2025. Disponível em: ⟨https://pomofocus.io⟩.