

TRANQUIL MIND: Chat de atendimento psicológico

Pedro Henrique Mendes Oliverio
Graduado em Sistemas de Informação – Uni-FACEF
pedrooliverio4@gmail.com

Davi Luiz de Alcântara
Graduado em Engenharia de Software – Uni-FACEF
dxvialcantara@gmail.com

Leandro Borges
Docente do Uni-FACEF
leandro.borges@facef.br

Resumo

A solução proposta para uma plataforma de atendimento psicológico envolve a criação de um ambiente online seguro e privado que oferece terapia virtual. Essa plataforma inclui terapeutas qualificados, selecionados, avaliação inicial para direcionar os pacientes, acompanhamento através de chat, recursos de suporte, acompanhamento do progresso, conformidade legal e ética, feedback dos pacientes e avaliações terapêuticas. Essa abordagem visa atender às necessidades de indivíduos buscando apoio psicológico, mantendo a confidencialidade e garantindo a qualidade e a segurança dos serviços prestados. O objetivo principal da plataforma de atendimento psicológico proposta é oferecer um ambiente online seguro e confidencial para atender às necessidades de indivíduos que buscam apoio psicológico. Isso inclui proporcionar terapia virtual por meio de terapeutas qualificados, garantir a conformidade legal e ética, oferecer suporte personalizado e acompanhar o progresso terapêutico dos pacientes.

Palavras-chave: Chat. Terapia. Virtual

Abstract

The proposed solution for a psychological support platform involves creating a secure and private online environment that offers virtual therapy. This platform includes qualified and carefully selected therapists, an initial assessment to direct patients, chat-based follow-up, support resources, progress tracking, legal and ethical compliance, patient feedback, and therapeutic evaluations. This approach aims to address the needs of individuals seeking psychological support while maintaining confidentiality and ensuring the quality and safety of the services provided. The main goal of the proposed psychological support platform is to provide a secure and confidential online environment to meet the needs of individuals seeking psychological support. This includes providing virtual therapy through qualified therapists, ensuring legal and ethical compliance, offering personalized support, and tracking patients' therapeutic progress..

Keywords: Chat. Therapy. Virtual.

1. Introdução

Nos últimos anos, a saúde mental tem ganhado cada vez mais destaque na sociedade. A conscientização sobre a importância do cuidado emocional e psicológico tem levado mais pessoas a buscar suporte profissional para enfrentar os desafios da vida moderna. No entanto, muitas barreiras ainda dificultam o acesso a serviços de saúde mental, como a falta de profissionais disponíveis, longas listas de espera e restrições geográficas. Diante desses desafios, as plataformas de atendimento psicológico surgem como uma alternativa promissora, oferecendo uma maneira mais acessível, conveniente e flexível de receber suporte emocional. Essas plataformas oferecem um ambiente virtual onde indivíduos podem se conectar com profissionais de saúde mental qualificados, por meio de sessões de terapia online e chats.

Neste artigo, vamos apresentar uma aplicação desenvolvida para facilitar a interação entre psicólogo e paciente e mostrar a maneira como as pessoas acessam cuidados de saúde mental. Abordaremos também as vantagens e desafios associados a essas plataformas, bem como o impacto que elas têm tido na democratização do acesso à terapia.

A metodologia adotada para a plataforma de apoio psicológico proposta envolve vários componentes-chave, os terapeutas são selecionados com cuidado e devem atender a qualificações e certificações específicas para garantir a mais alta qualidade de atendimento. A plataforma fornece um ambiente online seguro no qual os pacientes podem receber sessões de terapia virtual com os terapeutas, isso pode envolver sessões de chat ou outros métodos seguros de comunicação, os pacientes têm a opção de se comunicar com seus terapeutas por meio de chat seguro, permitindo suporte contínuo, a plataforma adere estritamente às diretrizes legais e éticas para proteger a confidencialidade e a privacidade dos pacientes, ao mesmo tempo que fornece atendimento psicológico profissional e ético. Por fim os pacientes são incentivados a fornecer feedback sobre suas experiências, permitindo melhorias contínuas na qualidade dos serviços e no desempenho dos terapeutas.

No geral, a metodologia tem como foco oferecer suporte psicológico acessível, confidencial e de alta qualidade por meio de uma plataforma online segura, com forte ênfase no atendimento centrado no paciente e na melhoria contínua.

Segundo Ceroni (2017), há um grande número de pessoas se comunicando virtualmente e participando de atividades em ambientes virtuais. Porém no Brasil, nota-se que um número expressivo de pessoas não possui acesso a esse tipo de tecnologia, que conforme Ang (2020), são 61,4 milhões de pessoas em uma população que ultrapassa os 210 milhões de pessoas.

Por fim, destacamos algumas evidências que apoiam a eficácia das plataformas de atendimento psicológico e seu papel complementar aos serviços tradicionais de saúde mental. Veremos como esse tipo de plataformas podem contribuir para a ampliação do acesso aos cuidados emocionais, especialmente em comunidades remotas, além de desempenhar um papel importante na desestigmatização da busca por ajuda psicológica.

À medida que a tecnologia continua a avançar e a demanda por serviços de saúde mental cresce, é essencial explorar as oportunidades e desafios oferecidos pelas plataformas de atendimento psicológico. Ao compreender melhor esse campo em constante evolução, podemos promover a melhoria da saúde mental em escala

global e proporcionar suporte necessário às pessoas em momentos de vulnerabilidade emocional.

2. Referencial Teórico

2.1 Atendimento psicológico

A resolução CFP nº 11/2018, do Conselho Federal de Psicologia, regulamenta o atendimento psicológico online e estabelece os requisitos técnicos e éticos para sua realização. O atendimento psicológico online é uma modalidade de atendimento psicológico que utiliza a tecnologia para oferecer suporte emocional, aconselhamento e terapia para pessoas em qualquer lugar do mundo. O uso da tecnologia para a realização do atendimento psicológico online tem crescido nos últimos anos, especialmente devido à pandemia de *COVID-19*, que limitou o acesso a serviços presenciais. O atendimento psicológico online pode ser realizado por meio de plataformas de vídeo, áudio, texto e outras tecnologias de comunicação.

É importante que as plataformas de atendimento psicológico sejam seguras e confiáveis, para garantir a privacidade e a confidencialidade dos pacientes e psicólogos. Os psicólogos que realizam atendimento psicológico online devem seguir o Código de Ética Profissional do Psicólogo e a resolução CFP nº 11/2018, e garantir que o atendimento seja realizado de forma ética, responsável e efetiva.

O atendimento psicológico é uma prática que busca promover a saúde mental e o bem-estar psicológico dos indivíduos. Segundo o Conselho Federal de Psicologia (CFP), a psicoterapia é uma intervenção técnica que se utiliza de recursos psicológicos para promover o desenvolvimento emocional e psíquico do paciente. Dessa forma, o atendimento psicológico é uma prática que tem como objetivo compreender as questões emocionais e psicológicas do paciente, de modo a auxiliá-lo a lidar com suas dificuldades, compreender seus sentimentos e desenvolver novas habilidades para enfrentar seus desafios. A análise de conteúdo é uma técnica que pode ser utilizada no atendimento psicológico para analisar e compreender as informações obtidas a partir das sessões terapêuticas.

A terapia virtual é uma forma de atendimento que faz uso de recursos tecnológicos para conduzir sessões de aconselhamento. De acordo com Figueiredo (2007), a tecnologia pode proporcionar apoio, acolhimento e oportunidades de alteração da vivência emocional.

2.2. Interação com psicólogo

A interação entre psicólogo(a) e paciente é uma parte fundamental do processo terapêutico. A relação terapêutica é um espaço seguro e confidencial em que o paciente pode se expressar livremente, sem julgamentos ou críticas. A empatia é uma habilidade importante do psicólogo (a) no estabelecimento da relação terapêutica. De acordo com Rogers (1957), a empatia é a capacidade do terapeuta de se colocar no lugar do paciente e compreender seus sentimentos e experiências, o autor pondera que a empatia é essencial para a construção de um relacionamento saudável e estável entre terapeuta e paciente. O psicólogo (a) deve ser honesto e transparente em relação aos seus sentimentos e pensamentos, e ser capaz de se expressar de forma clara e direta. Já a aceitação incondicional refere-se à aceitação do paciente em sua totalidade, sem julgamentos ou tentativas de mudá-lo.

Ainda de acordo com Rogers (1957), a aceitação incondicional é essencial para que o paciente se sinta seguro e confiante no ambiente terapêutico. O artigo de Ulkovski, Silva e Ribeiro (2017) tem como objetivo apresentar argumentos e desafios que essa modalidade de atendimento apresenta. As autoras destacam que, apesar das possibilidades e vantagens oferecidas pelo atendimento psicológico online, é preciso atentar-se para as especificidades dessa modalidade de atendimento e para o impacto que ela pode ter na relação entre o psicólogo e o paciente. No entanto, no contexto do atendimento psicológico online, essa relação pode ser afetada por diversos fatores, como a falta de contato físico, a comunicação mediada pela tecnologia e a ausência de sinais não-verbais.

Almeida e Rodrigues (2003) indagam se a ausência de contato físico pode representar um dos principais obstáculos para estabelecer a relação terapêutica no contexto da terapia virtual ou se isso depende mais da abordagem individual de cada profissional, ou seja, de como cada um opta por conduzir suas sessões, no entanto, eles destacam que essa forma de atendimento não deve ser descartada.

A comunicação mediada pela tecnologia pode apresentar limitações, como a falta de clareza e interpretação errônea das mensagens, que podem prejudicar a compreensão mútua entre o psicólogo e o paciente. Por isso, é importante que o psicólogo desenvolva habilidades específicas para o atendimento psicológico online, como a capacidade de interpretar e responder às mensagens de forma clara.

De acordo com Ulkovski, Silva e Ribeiro (2017) ressaltam que, apesar dessas limitações, o atendimento psicológico online pode ser uma opção válida e eficaz para muitos pacientes, desde que o psicólogo esteja preparado para lidar com as particularidades dessa modalidade de atendimento. É fundamental que o profissional esteja familiarizado com as tecnologias utilizadas e possua habilidades específicas para a construção da relação terapêutica no contexto online. Além disso, é importante que haja uma supervisão adequada e uma regulamentação clara do atendimento psicológico online, visando garantir a qualidade e segurança do serviço prestado.

2.3. Interação com paciente

A interação com o paciente é um dos principais componentes do processo terapêutico, independentemente do formato de atendimento utilizado, seja ele presencial ou online. A literatura científica tem abordado cada vez mais o atendimento psicológico online e seus efeitos na relação terapêutica. De acordo com Suler (2000), existem duas abordagens para conduzir o atendimento psicológico de forma remota: a abordagem síncrona e a abordagem assíncrona. O atendimento síncrono ocorre de maneira instantânea, na qual o terapeuta e o paciente se conectam virtualmente ao mesmo tempo, embora não estejam fisicamente presentes. Isso implica em sessões com tempo limitado ao encontro virtual imediato, por outro lado, na abordagem assíncrona, terapeuta e paciente desfrutam de maior flexibilidade em relação ao tempo. As interações não ocorrem simultaneamente, permitindo que o paciente compartilhe seus pensamentos e sentimentos quando se sentir mais à vontade. Enquanto isso, o terapeuta tem a vantagem de analisar de forma mais aprofundada os relatos escritos pelo paciente, possibilitando uma reflexão mais minuciosa e detalhada.

Com base em estudos conduzidos por Pieta e Gomes (2014) sobre a relação terapêutica nas intervenções psicoterápicas, foi constatado que as terapias online não apresentam diferenças significativas em comparação com as terapias

tradicionais. Esse estudo ajuda a reforçar que o atendimento virtual pode sim ser eficaz tanto como o presencial. Alguns psicólogos observaram que os pacientes apresentam um certo grau de confiança no ambiente da internet e criam essa conexão mais rapidamente do que em terapias presenciais. Os pacientes possuem uma tendência a revelar seus problemas com mais facilidade, com uma postura mais aberta e sincera em comparação com as terapias convencionais, o que leva a uma exposição mais rápida de suas dificuldades. Porém é importante salientar que cada tipo de atendimento vai de acordo com as necessidades e recursos de cada pessoa, e somente o paciente e psicólogo vão poder dizer com exatidão qual a forma preferem interagir entre si.

3. Estrutura da plataforma

Para a criação de uma plataforma de atendimento psicológico eficiente, foi necessário o uso de tecnologia e *softwares* específicos. A plataforma precisou garantir a privacidade e segurança das informações dos pacientes e dos profissionais envolvidos no atendimento.

A tecnologia utilizada para desenvolver a parte de front/back-end foi o *framework Next.js*. De acordo com Felipe Nascimento (2022) o *Next.js* é uma estrutura *JavaScript* de código aberto, amplamente utilizada para desenvolver aplicativos web modernos e escaláveis. Embora o *Next.js* não seja uma linguagem em si, ele é uma ferramenta que pode ser usada com diversas linguagens de programação, como o *TypeScript*. No desenvolvimento da plataforma de atendimento psicológico, consideramos as necessidades específicas do projeto e escolhemos essa tecnologia por sua capacidade de oferecer uma fácil manipulação e manutenção.

Para desenvolvimento do código a *IDE (Integrated Development Environment)* utilizada foi o *Visual Studio Code*. O *Visual Studio Code* é um editor de código-fonte desenvolvido pela *Microsoft* para *Windows, Linux e macOS*. Ele inclui suporte para depuração, controle de versionamento *Git* incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código.

Também optamos pelo *MongoDB* como sistema de gerenciamento de banco de dados. O *MongoDB* foi escolhido devido à sua flexibilidade na estruturação de dados, o que permitiu que os dados fossem organizados de forma adaptável. Isso foi especialmente útil para lidar com dados de diferentes tipos e formatos, com a possibilidade de adicionar novos campos e tipos de dados sem alterar a estrutura do banco de dados como um todo. Segundo Anderson Franciscato (2023), o *MongoDB* é altamente escalável, o que é crucial para aplicações de atendimento psicológico que podem experimentar um aumento significativo na demanda. Ele oferece um desempenho rápido e eficiente, fundamental para responder rapidamente às solicitações dos usuários, e recursos de indexação e busca para facilitar o acesso aos dados. Devido ao uso de um banco de dados não relacional, não foi necessário realizar o diagrama de relacionamento dos dados.

No desenvolvimento da plataforma, adotamos um modelo de *API (Application Programming Interface) REST (Representational State Transfer)* para o controle das rotas e endpoints do sistema. Segundo Cairo Noletto (2022) as *APIs REST* são uma abordagem popular para o desenvolvimento de interfaces de programação de aplicativos que permitem que diferentes sistemas se comuniquem de forma eficiente e segura. Em uma plataforma de atendimento psicológico, as *APIs REST* oferecem vantagens como integração com diferentes sistemas, compatibilidade com diversas linguagens de programação, padronização de interfaces e acesso

seguro aos dados. Elas também facilitam o teste e desenvolvimento das funcionalidades da plataforma.

O *Postman* foi escolhido para realização de controle das rotas da aplicação e para a aplicação dos testes. É uma ferramenta amplamente usada por desenvolvedores para testar *APIs*, especialmente *APIs REST*. Dado o contexto de desenvolvimento de uma plataforma de atendimento psicológico que utiliza uma arquitetura de *API REST*, o uso do *Postman* foi extremamente importante e benéfico. Ele ajuda a garantir a funcionalidade, segurança e desempenho da *API*, além de facilitar a colaboração entre a equipe de desenvolvimento. O uso desta ferramenta foi fundamental para garantir que a *API* funcione de maneira confiável e eficaz, atendendo as necessidades dos usuários e dos profissionais envolvidos na plataforma.

Ao longo do desenvolvimento do projeto, utilizamos o versionamento *Git* (sistema de controle de versão) e hospedamos o código na plataforma *Github*. Segundo Cairo Noletto (2021) o *Git* é uma ferramenta essencial para o desenvolvimento de projetos de software. Permite o acompanhamento do histórico de todas as alterações realizadas no código-fonte, facilitando a restauração de versões anteriores e a comparação das mudanças ao longo do tempo. O *Git* também suporta a colaboração entre membros da equipe de desenvolvimento, o que é crucial em projetos complexos como uma plataforma de atendimento psicológico.

A plataforma foi desenvolvida levando em consideração a regulamentação do Conselho Federal de Psicologia (CFP) que regulamenta o atendimento psicológico online no Brasil. Essa regulamentação define as condições necessárias para que o atendimento seja realizado de forma ética e segura, incluindo a necessidade de registro ativo no CFP e o seguimento das diretrizes éticas do conselho (CONSELHO FEDERAL DE PSICOLOGIA, 2018, p. 11).

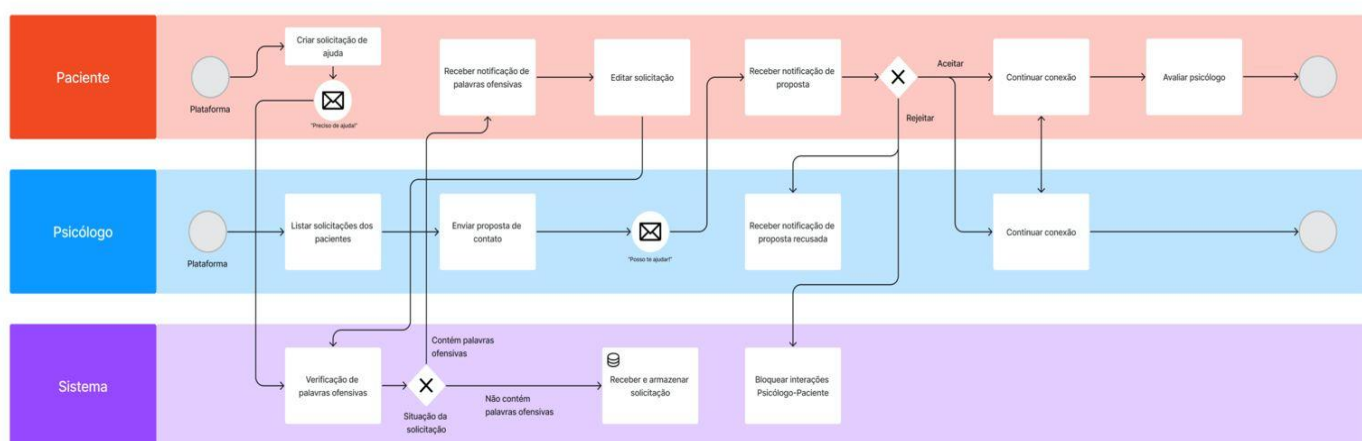
A estrutura da plataforma de atendimento psicológico foi planejada para oferecer um ambiente acolhedor e seguro para os pacientes. Garantimos que o ambiente virtual fosse confortável, com cores adequadas, uma decoração que transmitisse calma e tranquilidade.

O processo de atendimento psicológico online foi semelhante ao atendimento presencial, com uma avaliação inicial realizada pelo psicólogo para entender as necessidades do paciente e definir um plano de tratamento personalizado.

Para armazenamento de imagens e mensagens foi utilizado o *Supabase* que é uma solução que oferece armazenamentos focados em aplicações web. O *Supabase* possui um sistema robusto de armazenamento e de simples implementação, isso foi de suma importância visto que visamos pela fácil manipulação e manutenção no projeto. A ferramenta também oferece um banco de dados em tempo real, muito comum em aplicações que possuem a funcionalidade de *chat*. De forma resumida, a solução funciona através de canais e eventos que são disparados e lidos conforme a demanda.

Na figura 1 temos um *BPMN* (*business Process Model and Notation*), onde é exibido todo o fluxo da plataforma:

Figura 1 - BPMN do sistema



Fonte: Os autores

4. Solução proposta

Desenvolver uma plataforma online segura para proteger a privacidade dos pacientes e terapeutas. Garantir que as informações pessoais e as sessões de terapia sejam mantidas confidenciais. Estabelecer um processo de seleção para terapeutas licenciados e qualificados. Isso pode incluir verificação de credenciais, experiência e histórico profissional.

Oferecer uma avaliação inicial online para ajudar a entender as necessidades do paciente e oferecê-lo um tratamento adequado. Oferecer a terapia através do chat em tempo real, para acomodar as preferências e necessidades dos pacientes. Permitir que os pacientes escolham horários de terapia que se adequem às suas agendas. Isso pode incluir horários noturnos e fins de semana, além das sessões de terapia.


Facilitar o acompanhamento do progresso dos pacientes ao longo do tempo. Isso pode incluir avaliações regulares e a capacidade de ajustar o plano de tratamento conforme necessário. Certificar-se de que a plataforma esteja em conformidade com as leis locais e regulamentações de prática psicológica online. Isso pode incluir o cumprimento das leis de privacidade de dados e regulamentações específicas para a prática de psicologia. Permitir que os pacientes forneçam feedback sobre suas experiências e avaliem os terapeutas. Isso ajuda a melhorar a qualidade dos serviços e auxilia os futuros pacientes na escolha de um terapeuta.

4.1 Protótipo do sistema

A interface foi desenvolvida pensando em uma abordagem amigável e confortável para os usuários, cores neutras foram escolhidas propositalmente. Os designs e os componentes foram construídos a partir de um esquema do figma que foi construído por um designer profissional, seguindo os padrões modernos de UI/UX, o design é reutilizável, adaptável, e foi implementado e validado por diversas empresas.

De início foi desenvolvido a tela de login, em um formato simples, contendo os campos necessários para acesso ao sistema, nela está presente o nome da aplicação e a sua logo, que também foi pensada para ser algo minimalista, visando assim que os usuários da plataforma, principalmente os pacientes, sintam-se confortáveis e acolhidos ao acessarem a plataforma para fazer seus tratamentos. A figura 2 exibe o design implementado.

Figura 2 - Tela de Login do sistema

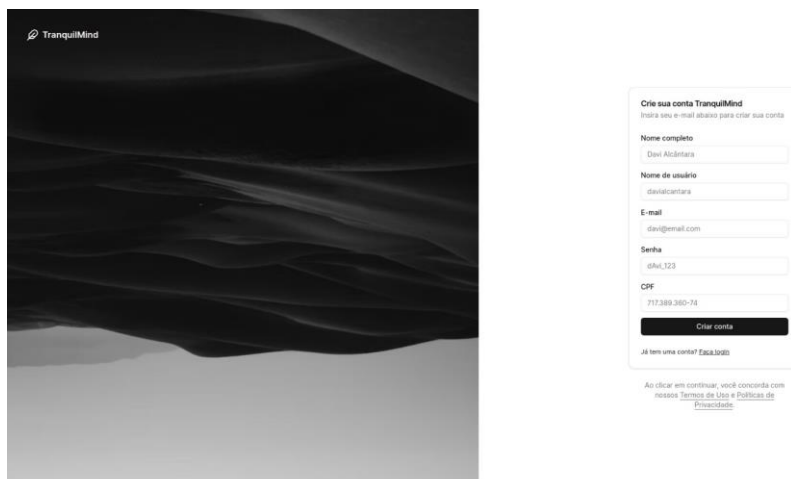


A imagem mostra a tela de login do sistema TranquilMind. No topo, há o logo da TranquilMind e o título "Entrar na plataforma". Abaixo, há uma instrução: "Preencha os campos com suas credenciais para continuar". O formulário contém dois campos de entrada: "E-mail" com o valor "seu@email.com" e "Senha" com o valor "Sua senha". Abaixo dos campos, há um botão "Continuar" com uma seta para a direita. Na base da tela, há um link "Não tem uma conta? Registre-se".

Fonte: Os autores

Conforme a figura 3, foi desenvolvido uma tela de cadastro exclusiva para os pacientes, contendo informações básicas para o cadastro, e após clicar em "criar conta" será enviado uma confirmação no e-mail informado pelo usuário

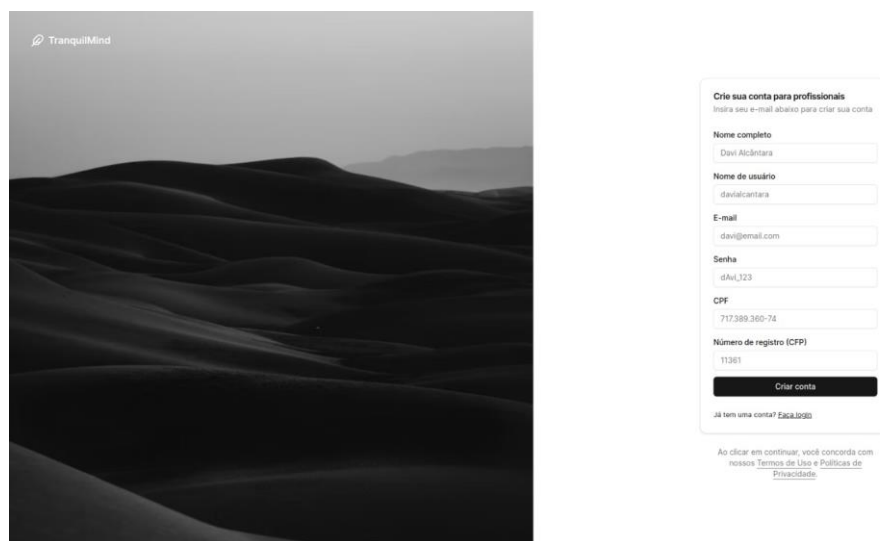
Figura 3 - Cadastro do Paciente



Fonte: os autores

A figura 4 ilustra que para psicólogos o cadastro é praticamente o mesmo, o diferencial é inserir o número de registro do cadastro no CFP, para confirmação da identidade do profissional. O processo de confirmação de e-mail e identidade também é realizado nessa etapa, com acréscimo apenas da confirmação da CFP

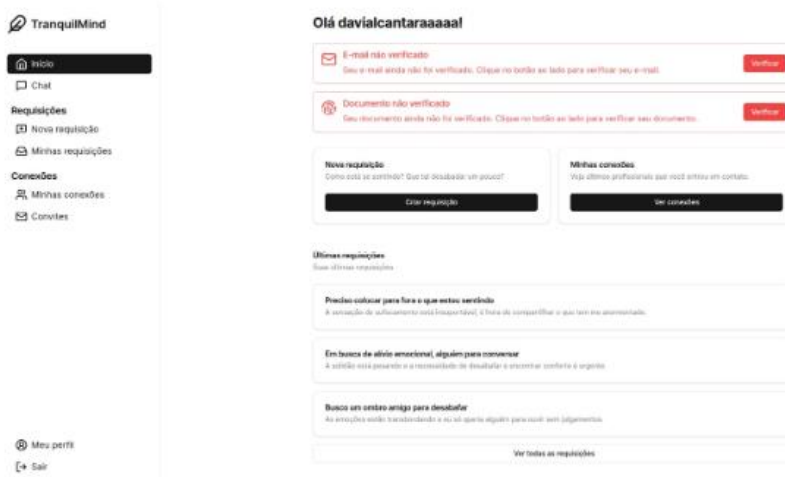
Figura 4 - Cadastro do psicólogo



Fonte: Os autores

A tela inicial do sistema trará um breve histórico das últimas requisições do paciente (se tiver) além de alertas de informações pendentes do usuário, nessa tela também é possível iniciar uma nova requisição, conforme a figura 5:

Figura 5 - Tela inicial



Fonte: os autores

Para iniciar um atendimento, o paciente deve indicar um assunto principal do problema e descrever brevemente o que está ocorrendo e logo após o chat será iniciado, conforme a figura 6.

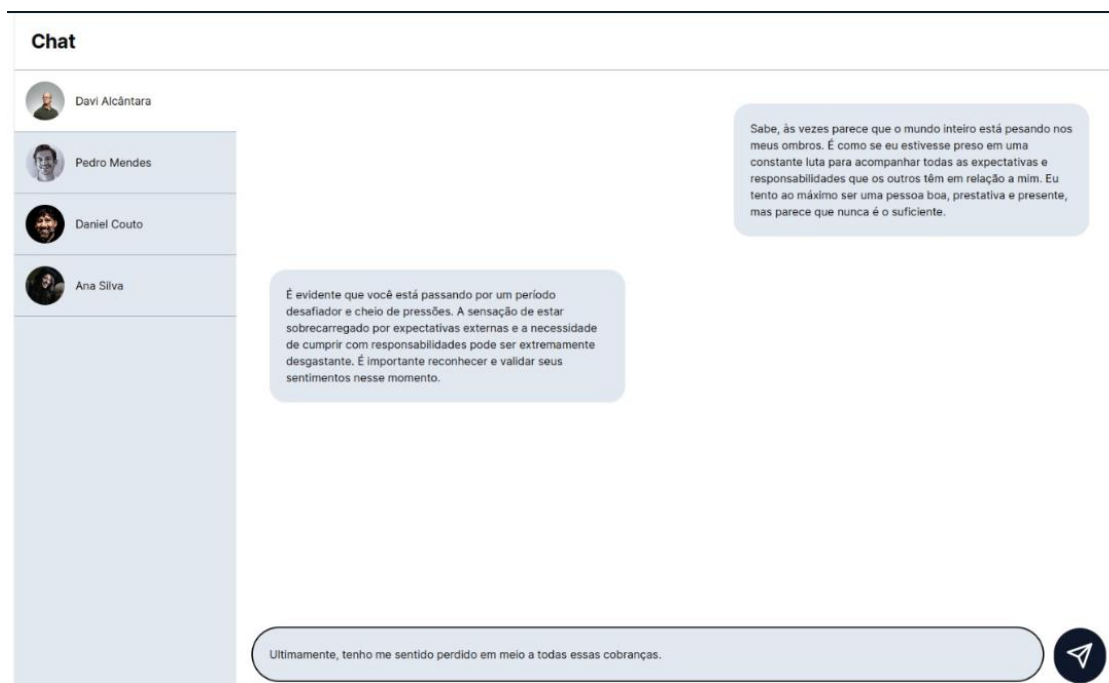
Figura 6 - Descrição do problema

The form for describing a problem consists of two main input fields. The first is labeled 'Assunto do problema' and contains the text 'Estou com problemas na escola...'. The second is labeled 'Descreva seu problema' and contains the text 'Conte-nos um pouco mais sobre seu problema'. Below the second field is a note: 'Não meça palavras. Conte tudo que está sentindo.' At the bottom of the form is a black button labeled 'Criar requisição'.

Fonte: Os autores

A figura 7 mostra como o chat foi feito com uma interface simples, porém de fácil entendimento para ambos que estiverem envolvidos no atendimento, é importante manter uma interface intuitiva, porém ao mesmo tempo agradável. No canto esquerdo, estarão presentes as pessoas com quem já trocou mensagens.

Figura 7 - Chat de atendimento



Fonte: os autores

4.2 Desenvolvimento

Essa seção contém partes do código, porém não representa sua totalidade, para acesso ao projeto completo, basta acessar o link do repositório contendo todo o código: <https://github.com/davialcantaraa/tranquilmind>

As rotas de API foram criadas no Next.js, praticamente todas as rotas da aplicação seguem o mesmo padrão de fluxo, na figura 8 temos a criação de um usuário, primeiramente é feita uma validação dos dados vindos do front-end. Caso os dados estejam inválidos ou incorretos, uma exceção será lançada e devolvida para o usuário em tela. Também é feita uma verificação para caso o usuário já esteja cadastrado. Passadas as validações, geramos um hash da senha para segurança do usuário e é gerado uma foto de perfil aleatória, em seguida é feito o registro no banco de dados MongoDB, que por sua vez tem uma conexão usando biblioteca prisma - *ORM (Object-Relational Mapping)*. Em seguida é gerado um trabalho na fila para disparar um e-mail para criação de usuário, por fim retornamos uma mensagem que o usuário foi criado com sucesso.

Figura 8 - Rota de criação de usuário

```

10 res: NextApiResponse
11 } {
12   if (req.method === "POST") return res.status(405).json(errorMessages[405])
13
14   const validation = createUserSchema.safeParse(req.body)
15
16   if (!validation.success)
17     return res.status(404).json({ message: "Dados inválidos" })
18
19   const { email, password, username, user_type, document_id, full_name } =
20     req.body
21
22   if (!user_type) return res.status(400).json({ message: "Dados inválidos" })
23
24   const userExists = await prisma.user.findFirst({
25     where: { OR: [{ email }, { username }, { document_id }] },
26   })
27
28   if (userExists)
29     return res.status(409).json({ message: "Usuário já cadastrado" })
30
31   const hashedPassword = await generateHash(password)
32   const randomAvatar = `${process.env.DICEBEAR_URL}?seed=${username}`
33
34   const data = {
35     email,
36     password: hashedPassword,
37     username,
38     user_type,
39     avatar: randomAvatar,
40     document_id,
41     full_name,
42   }
43
44   const emailOptions = {
45     from: "tranquilmind@example.com",
46     to: user.email,
47     subject: "TranquilMind: Confirme seu endereço de e-mail.",
48   }
49
50   await sendEmailConfirmation.enqueue({
51     options: emailOptions,
52     user_id: user.id,
53   })
54
55   return res.status(201).json({ message: "User created successfully!" })
56
57

```

Fonte: os criadores

Após a criação do usuário novos registros podem ser visualizados no MongoDB Atlas (painel administrativo do MongoDB), conforme figura 9.

Figura 9 - Painel administrativo do MongoDB

```

_id: ObjectId('6412260b0c07ac64554de983')
password: "$2b$12$4Dx72jpe0f1WXb6G58k0meUDvV20AJX08Gb0aUg1vmH/c4aZ4kEqG"
email: "davi_patient@gmail.com"
username: "patient_davialcantara"
full_name: "Davi Luiz de Alcântara"
avatar: "https://rqkzqjkincxzdwxgdfrrf.supabase.co/storage/v1/object/public/imag..."
document_id: "27752187844"
user_type: "patient"
updated_at: 2023-03-18T02:39:07.033+00:00
email_verified: "not_verified"
document_verified: "error"
description: "New description"

```

Fonte: Os autores

Após o registro do usuário e autenticação na plataforma ele será redirecionado para a tela de confirmação de identidade, no código abaixo é feito a

confirmação da identidade do usuário, essa validação é feita através da API de leituras de imagem do google (google vision). Dependendo do resultado da confirmação uma alteração é feita no campo "document_verified" no registro do banco de dados do usuário. O usuário terá o retorno do status da confirmação na própria plataforma, para os psicólogos também é feito uma confirmação de CFP (Conselho Federal de Psicologia), para garantir a eficácia e a segurança dos tratamentos e sessões que serão realizados, conforme figura 10:

Figura 10 - Confirmação de identidade

```
export default Queue<IConfirmIdQueue>({
  "api/queues/confirm-id",
  async ({ user_id, file_path }) => {
    logger.info("Confirming ID")
    const photo = fs.readFileSync(file_path)
    const user = await prisma.user.findFirst({
      where: { id: user_id },
    })
    if (!user) {
      logger.error("Error trying to confirm ID: User not found")
      throw new Error("User not found")
    }

    const { result } = await ocr.textDetection(photo)
    const text = result.fullTextAnnotation?.text
    if (!text) {
      logger.error("Error trying to confirm ID: Invalid content")
      throw new Error("Invalid content")
    }

    const textWithNoSpecial = removeSpecial(text)
    const hasName = text.includes(user.full_name.toUpperCase())
    const hasDocument = textWithNoSpecial.includes(user.document_id)

    if (hasName && hasDocument) {
      const newUser = await prisma.user.update({
        where: { id: user_id },
        data: {
          document_verified: "verified",
        },
      })
      logger.info("Document verified")
      cache.set(cacheUserId(newUser.id), cacheUser(newUser))
      await fs.unlinkSync(file_path)
      return
    } else {
      logger.info("Document invalid")
      const newUser = await prisma.user.update({
        where: { id: user_id },
        data: {
          document_verified: "error",
        },
      })
      await fs.unlinkSync(file_path)
      cache.set(cacheUserId(newUser.id), cacheUser(newUser))
    }
  }
})
```

Fonte: Os autores

Após isso é feito um processo de confirmação de e-mail nos padrões que já são conhecidos. O código presente na figura 11 ilustra como é realizado uma das principais funcionalidades da plataforma, a criação de uma solicitação por um paciente. Verificamos se o usuário que fez a solicitação é devidamente um paciente e em seguida inserimos uma solicitação no banco de dados.

Figura 11 - Criação de solicitação


```
export default async function handler(  
  req: NextApiRequest,  
  res: NextApiResponse  
) {  
  if (req.method !== "POST") return res.status(405).end()  
  const session = await getServerSession(req, res, authOptions)  
  if (!session) return res.status(401).end()  
  
  if (session.user.user_type !== "patient")  
    return res  
      .status(401)  
      .json({ message: "Você não tem permissão para criar uma solicitação" })  
  
  const { description, title, urgency } = patientRequestSchema.parse(req.body)  
  const patient_id = session.user.id  
  
  const request = await prisma.request.create({  
    data: {  
      description,  
      title,  
      urgency,  
      patient_id,  
    },  
  })  
}
```

Fonte: Os autores

Após isso as solicitações criadas serão disponibilizadas nos acessos dos psicólogos, e eles poderão responder a essas solicitações, as respostas serão retornadas para os pacientes, que por sua vez podem aceitar ou rejeitar a resposta do psicólogo.

Na figura 12 exhibe como é realizado o processo de aceitar ou rejeitar a resposta do psicólogo, o processo envolve criar um chat entre o psicólogo e o paciente com as mensagens da solicitação e a resposta do psicólogo é criado, em seguida verifica se a uma conexão existente entre o psicólogo e o paciente, caso não haja, um convite será enviado ao paciente para que ele possa aceitar a conexão, então o processo de tratamento é iniciado:

Figura 12 - Resposta de solicitação e criação de conexão

```

await prisma.$transaction(async (ctx) => {
  const newMessage = await supabase.from("messages").insert({
    chat_id,
    user_id: patient_id,
    content: message,
  })

  const doesConnectionExists = await ctx.connection.findFirst({
    where: {
      patient_id,
      professional_id,
    },
  })

  if (!doesConnectionExists) {
    const invitation = await prisma.invitations.findFirst({
      where: {
        AND: [{ receiver_id: patient_id, sender_id: professional_id }],
      },
    })

    if (!invitation) {
      return res.status(500).json({ message: "Erro do servidor" })
    }
    ctx.connection.create({
      data: {
        patient_id: isPatient ? patient_id : professional_id,
        professional_id: isPatient ? professional_id : patient_id,
      },
    })
    prisma.invitations.delete({
      where: { id: invitation.id },
    })
  }

  return newMessage
})

res
  .status(200)
  .json({ message: "Resposta de solicitação retornada com sucesso" })
}

```

Fonte: Os autores

A figura 13 exibe como é feita a usabilidade do chat, usando a ferramenta de *Realtime* do *Supabase*.

Figura 13 - Usabilidade do chat

```

export default async function handle(
  req: NextApiRequest,
  res: NextApiResponse
) {
  if (req.method !== "POST") return res.status(405).end()
  const session = await getServerSession(req, res, authOptions)
  if (!session) return res.status(401).end()

  const { chat_id, payload, media, request_answer_id } =
    sendMessageSchema.parse(req.body)
  const user_id = session.user.id

  const message = {
    chat_id,
    user_id,
    payload,
    media,
    request_answer_id,
  }

  await supabase.from("messages").insert(message)

  res.status(201).json({ message: "Message sent successfully" })
}

```

Fonte: Os autores

Após a sessão de tratamento através do chat o paciente poderá criar um depoimento sobre o psicólogo. Os depoimentos aparecerão no perfil do psicólogo, a figura 14 exibe como é realizado esse processo.

Figura 14 - Feedback sobre o psicólogo

```

if (req.method === "POST") return res.status(405).end()
const session = await getSession(req, res, authOptions)
if (!session) return res.status(401).end()

if (session.user.user_type === "patient")
  return res
    .status(401)
    .json({ message: "User have does not have the required permissions" })

const { description, title, professional_id } =
  professionalTestimonialSchema.parse(req.body)
const patient_id = session.user.id

await prisma.testimonial.create({
  data: {
    description,
    title,
    patient_id,
    professional_id,
  },
})

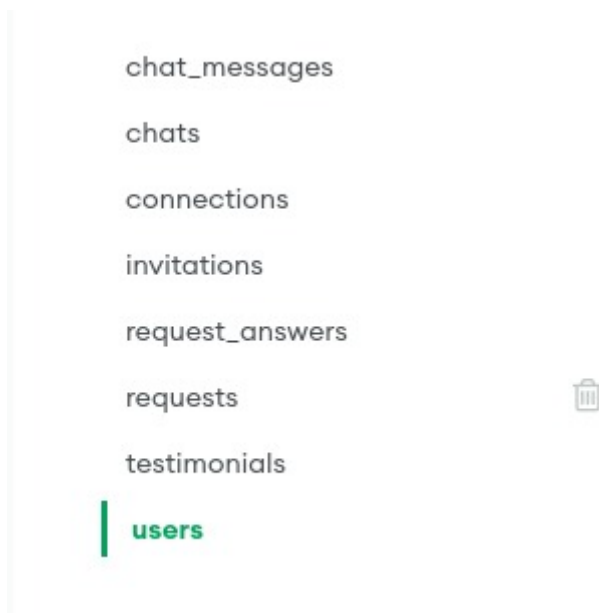
res.status(201).json({ message: "Testimonial created successfully" })

```

Fonte: Os autores

Para o funcionamento da aplicação, foram meticulosamente projetadas collections no banco de dados, cada qual desempenhando um papel crucial na funcionalidade e no desempenho geral da aplicação. Essas collections funcionam como blocos de construção vitais, permitindo que os dados sejam armazenados de forma organizada e acessíveis quando necessário, garantindo assim a eficiência e a confiabilidade do sistema. A figura 15 exhibe as collections criadas, destacando a importância da estrutura de dados sólida para o sucesso da aplicação. Cada collection representa um elo essencial na cadeia de operações que permite aos usuários interagir, comprar, avaliar produtos e personalizar suas preferências na plataforma, criando uma experiência abrangente e satisfatória.

Figura 15 - Collection Banco de dados



Fonte: Os autores

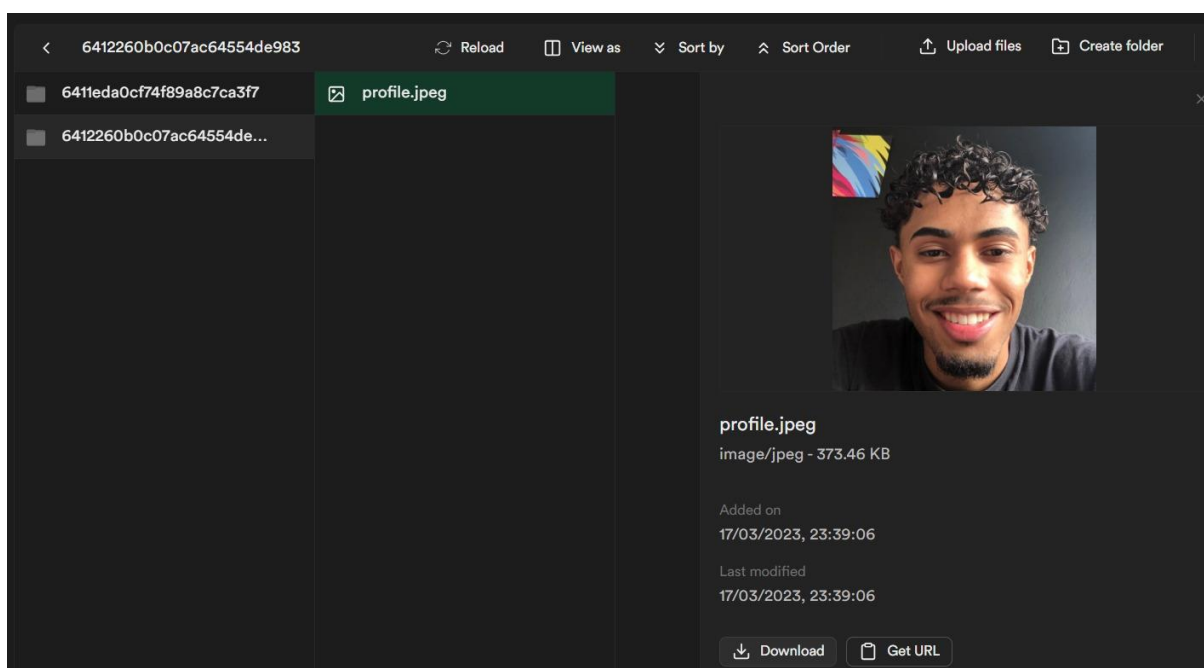
Para cada collection foram criadas rotas de CRUD (*Create, Read, Update, Delete*) e cada uma delas tem sua devida validação baseado no tipo de usuário (Psicólogo e paciente).

O armazenamento das imagens e mensagens foram realizadas no *Supabase Storage*, que como já citado, oferece uma solução de armazenamento de

fácil manutenção e aplicabilidade, uma escolha que se revelou altamente vantajosa devido à sua facilidade de manutenção e ampla aplicabilidade. Optar por armazenar os dados em nuvem é uma prática comum e fundamental para aplicações web modernas, uma vez que oferece benefícios como escalabilidade e um desempenho confiável. Além disso, a utilização de armazenamento em nuvem também contribui para garantir que as imagens sejam carregadas de forma rápida e eficiente, ao mesmo tempo em que reduz o espaço de armazenamento necessário.

Uma consideração importante ao lidar com imagens em aplicações web é a otimização. É fundamental reduzir o tamanho dos arquivos de imagem sem comprometer significativamente a qualidade visual. Isso não apenas economiza espaço de armazenamento, mas também melhora a experiência do usuário, uma vez que as imagens são carregadas mais rapidamente. Portanto, além do armazenamento em nuvem, é crucial implementar técnicas de compressão e otimização de imagens para garantir um desempenho eficiente e uma experiência de usuário aprimorada. As figuras 16 e 17 mostram a interface e a maneira como ficam organizadas e armazenadas as informações.

Figura 16 - Storage das imagens



Fonte: Os autores

Figura 17 - Storage das mensagens

id	uid	created_at	payload	user_id	media	chat_id	request_answer
962c9962-752d-464d-9865-adf376ea697		2023-03-23 00:46:41.92075+00	primeira mensagem	6412260b0c07ac64554de983	NULL	64154e419ef9c389deab1685	NULL
7fff8365-8279-40e9-ae10-ecd634e1db3b		2023-03-23 00:51:14.622094+00	segunda mensagem	641226370c07ac64554de984	NULL	64154e419ef9c389deab1685	NULL
1886b8ff-3fd3-48d2-8cc0-4590477e071		2023-03-23 00:51:22.367817+00	terceira mensagem	641226370c07ac64554de984	NULL	64154e419ef9c389deab1685	NULL
75a8c910-c3d4-43cb-8cca-27e9cdf5ac55		2023-03-23 00:51:23.129085+00	quarta mensagem	6412260b0c07ac64554de983	NULL	64154e419ef9c389deab1685	NULL

Fonte: Os autores

4.3 Segurança

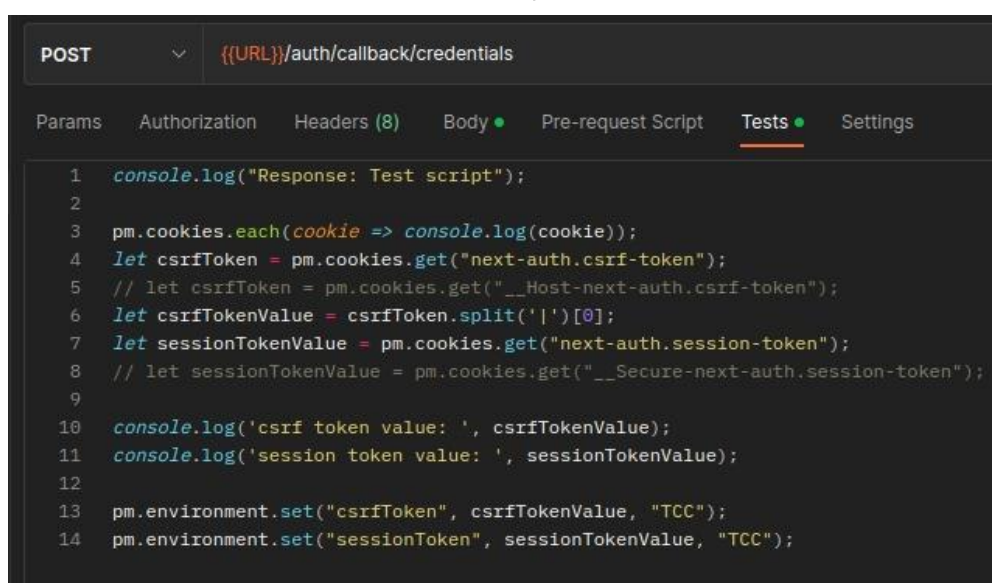
Por se tratar de uma aplicação que lida diretamente com dados extremamente sigilosos e sensíveis, tais como: documentos de identidade, confissões e problemas pessoais, foi realizado um trabalho dedicado à segurança. O gerenciamento da autenticação dos usuários foi desenvolvido de forma que as senhas dos mesmos são criptografadas quando registradas no Banco de Dados. As sessões duram exatamente 1 hora, sendo necessário um novo login após isso. Durante as sessões de atendimento, as mensagens que são enviadas no chat são criptografadas com criptografia ponta-a-ponta, tanto por parte do paciente quanto por parte do psicólogo, assim garantindo a segurança da integridade dos usuários. Para ter acesso a plataforma também é necessário confirmação de identidade por parte dos usuários; não é possível usufruir das funcionalidades da plataforma sem que haja essa confirmação.

4.4 Testes

Para assegurar a qualidade da aplicação foram realizados testes em todo o fluxo da aplicação e em ambos os tipos de usuário. Começamos no cadastro dos usuários e sua autenticação na plataforma, confirmação de identidade e e-mail, criação de solicitações para os pacientes, listagem e resposta das solicitações para os psicólogos, processo de conexão entre psicólogo e paciente e por fim a sessão de tratamento através do chat, finalizando com o depoimento do paciente.

A segurança dos usuários também foi testada, o usuário não terá acesso às funcionalidades da plataforma caso não tenha confirmado sua identidade e e-mail, as senhas são devidamente criptografadas e as sessões são de pouca duração para manter a integridade dos usuários, fazendo-se necessário um novo login após 1 hora de uso na plataforma. Para manter a integridade dos pacientes, os mesmos terão controle de suas conexões e interações com os psicólogos. Foram realizados testes de api utilizando a plataforma Postman. Todas as rotas e todas as funcionalidades foram testadas. A collection foi separada por entidades e cada entidade com seus casos de uso, também foram criados scripts de testes para facilitar a criação e alteração de ambiente, conforme figura 18:

Figura 18 - Script de validação de retorno das rotas



```
POST {{URL}}/auth/callback/credentials

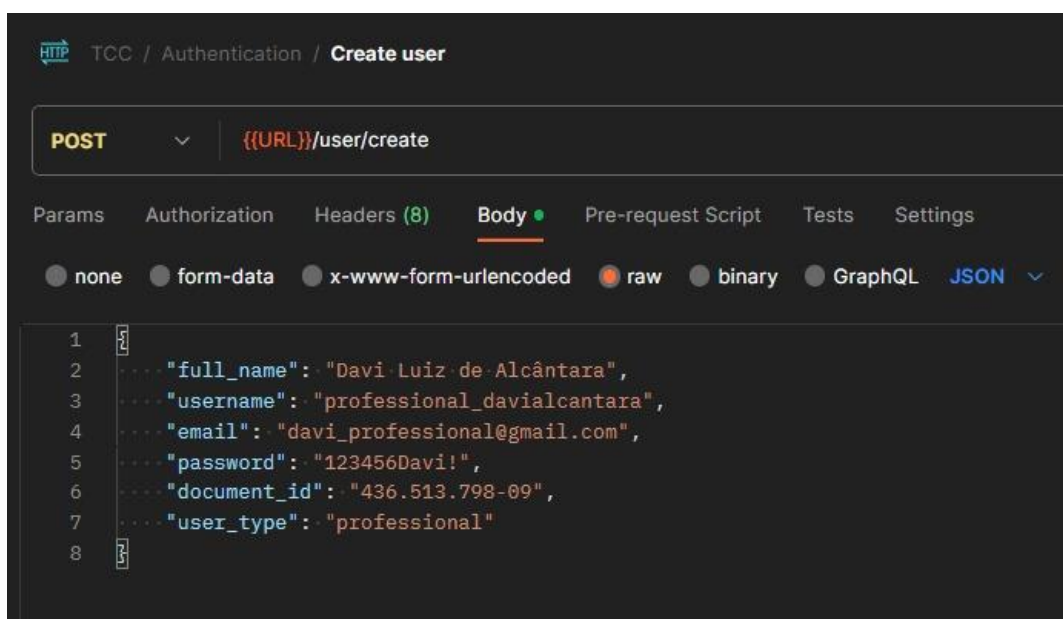
1 console.log("Response: Test script");
2
3 pm.cookies.each(cookie => console.log(cookie));
4 let csrfToken = pm.cookies.get("next-auth.csrf-token");
5 // let csrfToken = pm.cookies.get("__Host-next-auth.csrf-token");
6 let csrfTokenValue = csrfToken.split('|')[0];
7 let sessionTokenValue = pm.cookies.get("next-auth.session-token");
8 // let sessionTokenValue = pm.cookies.get("__Secure-next-auth.session-token");
9
10 console.log('csrf token value: ', csrfTokenValue);
11 console.log('session token value: ', sessionTokenValue);
12
13 pm.environment.set("csrfToken", csrfTokenValue, "TCC");
14 pm.environment.set("sessionToken", sessionTokenValue, "TCC");
```

Fonte: Os autores

No Postman, você pode realizar um teste de criação de um novo usuário na API usando o método HTTP (*Hypertext Transfer Protocol*) "POST", para isso é necessário abrir o Postman e criar uma nova solicitação. Definir o método HTTP como "POST" para indicar que você deseja criar um novo registro. Inserir a URL da API onde os registros de usuário são processados.

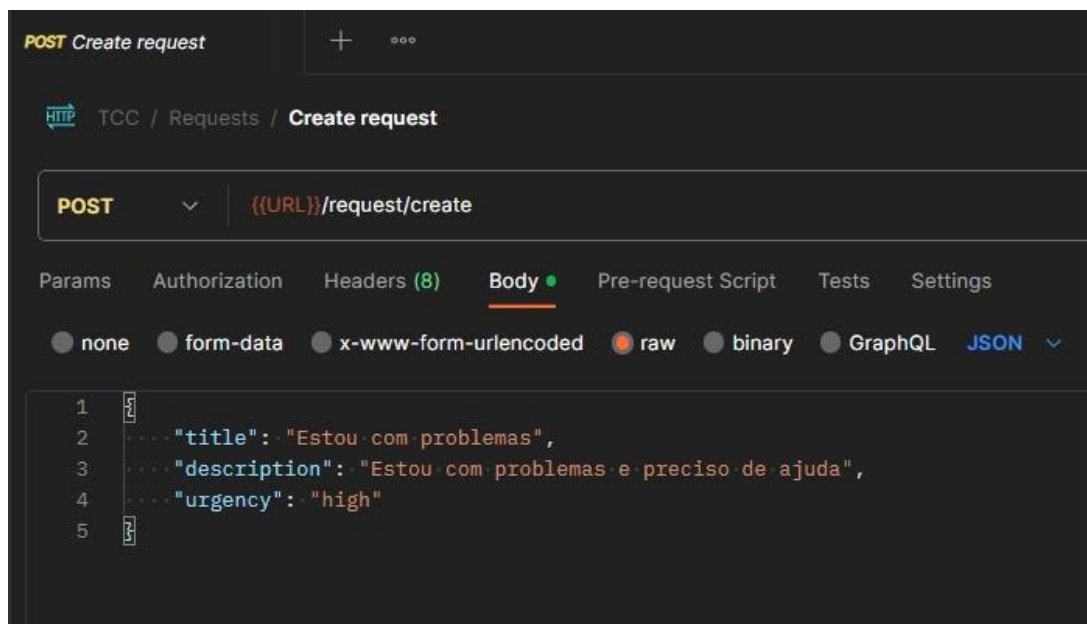
No corpo da solicitação, devem ser inseridos os dados necessários para criar um novo registro. Isso pode incluir informações como nome, email, senha e outros campos necessários. Deve-se certificar de incluir quaisquer cabeçalhos necessários, como tokens de autenticação ou cabeçalhos de autenticação básica, conforme especificado pela API. Clicando em "send" o Postman irá processar a solicitação e retornar a resposta da API, após isso deve -se examinar a resposta da API para garantir que a criação do novo registro tenha sido bem-sucedida. Isso pode incluir informações sobre o novo usuário criado, um código de status HTTP 200 (OK) e outras informações relevantes. As figuras 19 e 20 exemplificam esse processo:

Figura 19 - Rota de criação de usuário



Fonte: Os autores

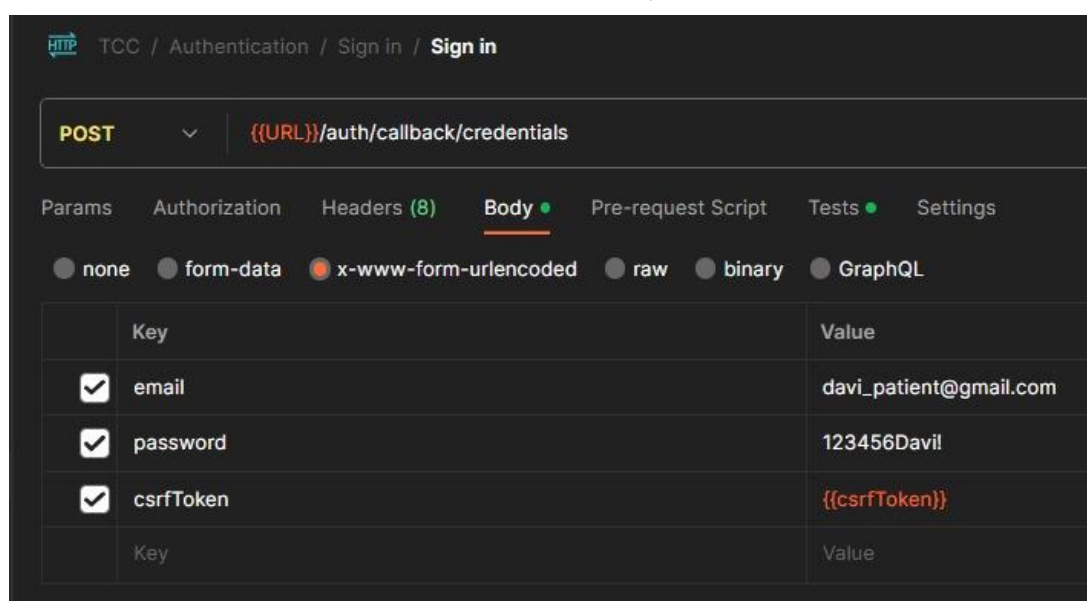
Figura 20 - Rota de criação de requisição



Fonte: Os autores

Foram utilizadas também, variáveis de ambiente, para tornar as solicitações mais dinâmicas e flexíveis. As variáveis de ambiente permitem que você armazene valores que podem ser reutilizados em várias solicitações, tornando mais fácil a configuração e a manutenção de seus testes e coleções, a figura 21 ilustra bem isso.

Figura 21 - Rota de autenticação dos usuários



Fonte: Os autores

4.4.1 Testes Unitários

Para garantir que a lógica e as regras de negócio funcionem adequadamente, foram realizados testes de unidade nas funções exercidas das rotas da *API*, eles são uma parte essencial da prática de desenvolvimento de software, independentemente da linguagem de programação, incluindo *JavaScript*. Os testes unitários são responsáveis por verificar se partes individuais (unidades) do seu código, como funções ou métodos, funcionam corretamente. As principais funcionalidades foram testadas, tais como: criação de usuário, login, criação de solicitação, criação de conexão, criação de avaliação, comunicação entre paciente e psicólogo etc.

A estrutura do teste segue um padrão bem definido. Primeiro, é criado um *mockup*, que é uma representação visual do produto ou sistema em desenvolvimento. Este *mockup* é essencial para fornecer uma referência visual clara do que está sendo testado. Além disso, uma resposta esperada também é definida. Esta resposta é uma expectativa do que o sistema deve retornar quando a funcionalidade em teste é acionada. A resposta esperada serve como um ponto de comparação para verificar se o sistema está se comportando conforme o esperado, em seguida, é realizada uma chamada para a rota específica do sistema que está sendo testada. Isso simula a interação do usuário ou de outro sistema com a funcionalidade em questão.

Após a chamada ser feita, é realizada a verificação da resposta retornada pelo sistema. Essa verificação envolve comparar a resposta obtida durante o teste com a resposta esperada previamente definida. Se as respostas coincidirem, o teste é considerado bem-sucedido, indicando que a funcionalidade está funcionando corretamente. Caso contrário, se houver diferenças entre a resposta obtida e a resposta esperada, isso pode indicar um problema ou bug no sistema que precisa ser corrigido.

Esse processo de teste ajuda a garantir que as funcionalidades do sistema se comportem conforme o esperado e que não ocorram regressões ou erros não detectados durante o desenvolvimento. Além disso, fornece uma base sólida para a melhoria contínua do sistema, à medida que novas funcionalidades são adicionadas ou atualizações são feitas, as figuras 20 e 21 exibem como foi feito esse processo.

Figura 22 - Teste de criação de usuário

```
test("API Route - User Creation", async () => {
  const req = mockR({
    method: "POST",
    body: {
      ...mockedUser,
    },
  })
  const res = createRes()
  const hashedPassword = "hashed-password"
  const mockUser = {
    ...mockedUser,
  }
  mockedGenerateHash.mockResolvedValue(hashedPassword)
  mockedPrisma.user.create.mockResolvedValue(mockUser)
  mockedSendEmailConfirmation.enqueue.mockResolvedValue()
  await handle(req, res)
  res.on("end", () => {
    const data = JSON.parse(res._getData())
    assert.equal(res._getStatusCode(), 201)
    assert.equal(data.message, "User created successfully!")
    assert.equal(mockedGenerateHash.mock.calls.length, 1)
    assert.equal(mockedPrisma.user.create.mock.calls.length, 1)
    assert.equal(mockedSendEmailConfirmation.enqueue.mock.calls.length, 1)
  })
})
```

Fonte: Os autores

Figura 23 - Teste de envio de solicitação

```
test("API Route - Create Patient Request", async () => {
  const req = createReq({
    method: "POST",
    body: {
      ...mockedData
    },
  })

  const res = createRes()

  mockedGetServerSession.mockResolvedValue(mockSession)
  mockedPrisma.request.create.mockResolvedValue({})

  await handler(req, res)

  assert.equal(res._getStatusCode(), 201)
  assert.equal(mockedPrisma.request.create.mock.calls.length, 1)
  assert.equal(res._getData(), "{}")
})
```

Fonte: Os autores

5. Considerações finais

Nos últimos anos, a saúde mental tem se destacado como uma questão de extrema importância em diversas esferas da sociedade. Reconhecer a necessidade de compreender os principais processos envolvidos no atendimento psicológico é fundamental para garantir que as pessoas tenham acesso a apoio e cuidados adequados. Nesse contexto, desenvolvemos uma aplicação dedicada a oferecer uma opção segura e eficaz para aqueles que buscam construir uma rede de apoio para cuidar de sua saúde mental. Acreditamos que a tecnologia pode desempenhar um papel crucial na promoção do bem-estar mental, desde que seja projetada e implementada de forma sensível e responsável.

Um dos princípios centrais que orientaram o desenvolvimento da nossa aplicação foi a criação de uma interface simples e intuitiva. Entendemos que muitas pessoas que buscam apoio psicológico podem estar enfrentando momentos difíceis e, portanto, é essencial que a utilização da aplicação seja acessível e livre de complicações. Além disso, não menos importante, é a garantia de que os dados dos pacientes e dos profissionais de saúde sejam tratados com o mais alto nível de segurança e confidencialidade. A privacidade é um direito fundamental que deve ser preservado em todos os aspectos do atendimento psicológico online.

Olhando para o futuro, temos planos para aprimorar ainda mais nossa aplicação, reconhecemos a importância de acompanhar as mudanças nas necessidades das pessoas que buscam apoio psicológico. Uma das nossas próximas metas é a implementação de videochamadas, o que proporcionará uma experiência de atendimento ainda mais abrangente. No entanto, reforçamos que a segurança das imagens e a proteção da privacidade de todos os envolvidos continuarão sendo nossa prioridade máxima.

Em resumo, nossa aplicação é uma resposta aos desafios atuais de saúde mental, fornecendo uma plataforma segura, intuitiva e ética para o apoio psicológico.

Referências:

ALMEIDA, L. P.; RODRIGUES, J. T. Narrativa e internet: possibilidades e limites do atendimento psicoterápico mediado pelo computador. 2003. *Psicologia, Ciência e Profissão*, v. 23, n. 3, p. 10-17.

ANG, Carmen. Where Will the Next Billion Internet Users Come From?. 2020. Disponível em: <https://bityli.cc/oyS>. Acesso em 3 de março de 2023

CERONI, M. L. V. Atendimento Psicoterápico Online e seus desdobramentos na relação Mente e Corpo. 2017. *Revista Latino-Americana De Psicologia Corporal*, 6(1), 99–110.

CFP. Conselho Federal de Psicologia. Resolução de nº 012, de 18 de agosto. Regulamenta o atendimento psicoterapêutico e outros serviços psicológicos mediados por computador e revoga a Resolução CFP Nº 003/2000. 2005. Brasília.

_____. Resolução de nº 011, de 21 de junho. Regulamenta os serviços psicológicos realizados por meios tecnológicos de comunicação a distância, o atendimento psicoterapêutico em caráter experimental e revoga a Resolução CFP N.º 12/2005. 2012. Brasília.

_____. Resolução de nº11 de 11 de maio. Regulamenta a prestação de serviços psicológicos realizados por meios de tecnologias da informação e da comunicação e revoga a Resolução CFP Nº 11/2012. 2018. Brasília

_____. Resolução de nº4, de 26 de março. Dispõe sobre regulamentação de serviços psicológicos prestados por meio de Tecnologia da Informação e da Comunicação durante a pandemia do COVID-19. 2020. Brasília

FRANCISCATO, Anderson. Qual a vantagem do MongoDB sobre os outros bancos de dados?". 2023. Disponível em: <https://bityli.cc/hmr>. Acesso em 22 de janeiro de 2023.

PIETA, M.A.M.; GOMES, W. B. Psicoterapia pela internet: viável ou inviável? 2014. Psicologia: Ciência E Profissão, v. 34, n. 1, p. 18-31.

RIBEIRO, A. B., SILVA, L. P. D. & ULKOVSKI, E. P. Atendimento Psicológico Online: Perspectivas E Desafios Atuais Da Psicoterapia. 2021. Psicologia: A Ciência do Bem-Estar (Atena Edit, pp. 56–64).

ROGERS, C. R. The necessary and sufficient conditions of therapeutic personality change. 1957. Journal of Consulting Psychology, 21(2), 95–103.

SCHINARO, Pietro. Beautifully designed components built with RadixUI and TailwindCSS.[S.I.]. 2023. Disponível em:<https://bityli.cc/nVb>. Acesso em 12 de junho de 2023.

NOLETO, Cairo. API REST: o que é e como montar uma API sem complicação? 2022. Disponível em:<https://bityli.cc/zns>. Acesso em 4 de janeiro de 2023

NOLETO, Cairo. Git e GitHub: o que são, quais as diferenças e como usar na prática? 2020. Disponível em: <https://bityli.cc/rXE>. Acesso em 18 de abril de 2023.

SULER, J. Psychology of cyberspace. PsyBC program. 2000. Disponível em: <http://users.rider.edu/~suler/psyber>. Acesso em 13 de agosto de 2023.