INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE
TECNOLOGIA EM SISTEMAS PARA INTERNET

JOÃO VITOR OTHON SILVA
KAYKE DE PÁDUA DA SILVA SOUZA
THALES ADRIEL SOARES DE ARAÚJO
WELSON ROSENDO RODRIGUES

Documento de Requisitos: Plataforma de Apoio à Saúde Mental
Versão 0.1

CURRAIS NOVOS / RN
2024

JOÃO VITOR OTHON SILVA
KAYKE DE PÁDUA DA SILVA SOUZA
THALES ADRIEL SOARES DE ARAÚJO
WELSON ROSENDO RODRIGUES

Documento de Requisitos: Plataforma de Apoio à Saúde Mental
Versão 0.1

Documento de Requisitos: Desenvolvimento de uma plataforma web para suporte à saúde mental, com foco em informações sobre ansiedade, depressão e o movimento Setembro Amarelo.

CURRAIS NOVOS / RN
2024

Histórico de Alterações

Data	Versão	Descrição	Modificado
19/09/2024	0.0	Criação do documento de requisitos, identificação dos requisitos, descrição dos requisitos funcionais	Welson Rosendo Rodrigues
20/09/2024	0.1	Descrição de todos os requisitos necessários e implementação dos diagramas	Welson Rosendo Rodrigues

1. Prefácio

Este documento descreve os requisitos, arquitetura e modelos do sistema **Safe Mind**, uma plataforma de apoio à saúde mental. Desenvolvido como parte de um projeto acadêmico, ele visa atender pacientes e profissionais de saúde mental, fornecendo recursos como agendamentos de consultas, conteúdo de autoajuda e a campanha **Setembro Amarelo.**

2. Índice

3. Introdução

Durante os últimos anos, onde a facilidade de acesso às tecnologias e a informação se tornou cada vez mais fácil e evidente, trouxe consigo uma certa dependência entre pessoas e suas tecnologias, que se tornou mais fácil de se obter, e graças ao incidente pandêmico entre os anos de 2019 e 2021, o isolamento social envolvendo toda a população, para fins de evitar um agravamento de mortes pelo mundo, acabou trazendo consigo um mal perigoso, casos de pessoas com ansiedade e depressão se agravou de forma agressiva.

Este projeto tem como objetivo a criação de uma plataforma de auto ajuda para pessoas que se sentem afetadas por esses sintomas, suportando clínicas psicológicas e terapêuticas, de forma online e acessível, destacando a integração estratégica de tecnologias-chave como **HTML**, **CSS**, **Javascript**, **Node.js**, **PostgresSQL** e **MongoDB**.

A combinação de **HTML** e **CSS**, usados como base de construção de uma interface visualmente confortável, atrativa e intuitiva. Utilizando o backend com o auxílio do Node.js, que proporciona estabilidade na construção das camadas de funcionalidades mais técnicas, utilizando juntamente da linguagem de programação Javascript, no qual permite realizar configurações complexas que ajudarão com a funcionalidade das principais funções da plataforma.

Além disso, o sistema utilizará duas estruturas de banco de dados distintas para o armazenamento de informações. O **PostgreSQL**, um banco de dados relacional, será responsável pela gestão dos dados estruturados, como o cadastro de usuários e perfis. Já o **MongoDB**, um banco de dados não relacional, será empregado para armazenar logs de atividades e outros dados menos estruturados.

4. Glossário

A seguir, são listados os principais termos técnicos e siglas utilizadas ao longo do projeto, com suas respectivas definições.

- **API (Application Programming Interface)**: Interface de programação que permite a comunicação entre diferentes softwares, facilitando o uso de funcionalidades de um sistema por outro.
- **Backend**: Parte do sistema responsável pelo processamento de dados e lógica de negócios, que não é diretamente visível ao usuário.
- **Frontend**: Parte do sistema com a qual o usuário interage diretamente, geralmente composta por interfaces gráficas e elementos visuais.
- **HTML (HyperText Markup Language)**: Linguagem de marcação utilizada para estruturar conteúdo na web.
- **JavaScript**: Linguagem de programação utilizada para criar funcionalidades dinâmicas em páginas web.
- **MongoDB**: Banco de dados NoSQL utilizado para armazenar dados de forma flexível, baseado em documentos no formato JSON.
- **Node.js**: Plataforma que permite a execução de JavaScript no servidor, possibilitando o desenvolvimento de aplicações web back-end.
- **PostgreSQL**: Sistema de gerenciamento de banco de dados relacional (RDBMS) que utiliza SQL como linguagem de consulta.
- **Setembro Amarelo**: Campanha de conscientização sobre a prevenção ao suicídio, promovida anualmente com foco em saúde mental.

5. Definição de Requisitos de Usuários

Os principais requisitos de usuário são **Pacientes** e **Profissionais.**

5.1. Requisitos Funcionais para Pacientes

- **Cadastro de Pacientes:** O sistema deve permitir que novos pacientes se cadastrem fornecendo informações básicas como nome, e-mail, data de nascimento, gênero e uma senha.
- **Login de Pacientes:** Pacientes devem conseguir acessar o sistema por meio de autenticação com e-mail e senha.
- **Visualização de Perfil:** Após o login, o paciente poderá acessar uma página de perfil onde suas informações pessoais são exibidas, incluindo um campo para editar seus dados.
- **Acesso a Conteúdos de Autoajuda:** O paciente poderá acessar uma seção de vídeos e artigos relacionados a saúde mental, incluindo conteúdos específicos sobre a campanha "Setembro Amarelo".
- **Consulta a Profissionais:** O paciente poderá visualizar os perfis dos profissionais disponíveis no sistema, com suas respectivas biografias e especialidades.
- **Agendamento de Consulta:** O paciente poderá agendar consultas com profissionais cadastrados, definindo data e hora disponíveis.

5.2. Requisitos Funcionais para Profissionais

- **Cadastro de Profissionais**: Profissionais de saúde mental poderão se cadastrar no sistema, fornecendo suas credenciais e áreas de atuação.
- **Login de Profissionais**: O sistema deve permitir que profissionais façam login usando e-mail e senha.
- **Gerenciamento de Perfil**: Profissionais devem ter acesso à sua página de perfil, onde poderão editar suas informações, incluindo biografia, foto de perfil e áreas de especialização.
- **Publicação de Conteúdo**: Profissionais poderão criar e publicar conteúdos de autoajuda (artigos e vídeos), relacionados a saúde mental, diretamente no sistema.

5.3. Requisitos Funcionais Geral

- **Home Page Personalizada:** Após o login, tanto pacientes quanto profissionais serão direcionados a uma home page personalizada, com atalhos para suas principais funcionalidades (autoajuda, perfil, profissionais disponíveis).
- **Página Setembro Amarelo:** Uma seção dedicada à campanha de prevenção ao suicídio, onde serão disponibilizados conteúdos educacionais e de conscientização para todos os usuários do sistema.
- **Autenticação e Segurança:** Todo o fluxo de cadastro e login deve ser protegido por criptografia, garantindo a privacidade e segurança dos dados dos usuários.
- **Suporte a Dispositivos Móveis:** O sistema deve ser responsivo, garantindo uma boa experiência de uso em smartphones e tablets.

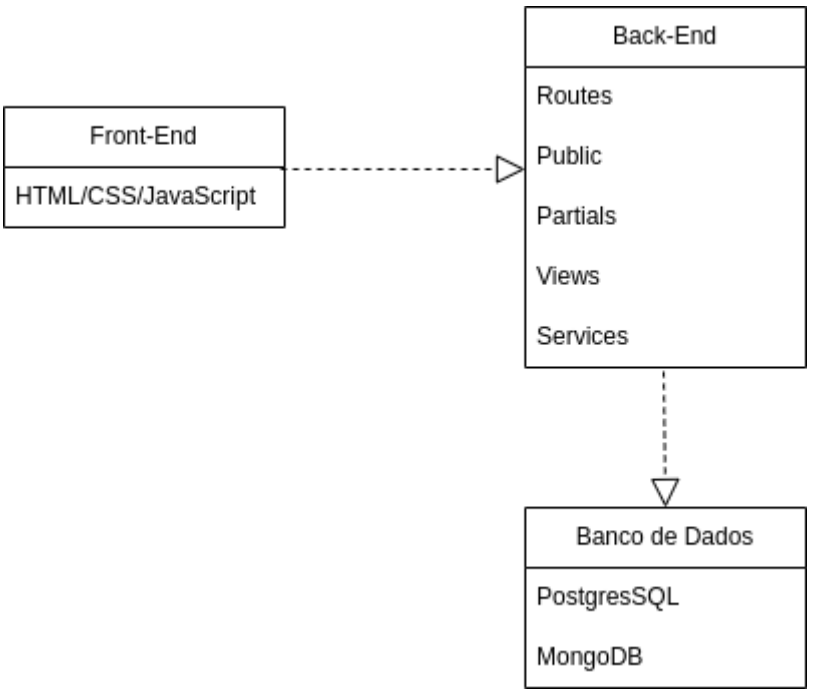
6. Arquitetura de Sistema

O sistema é dividido entre frontend e backend. O frontend foi desenvolvido utilizando HTML, CSS e JavaScript para proporcionar uma interface amigável ao usuário. O backend utiliza Node.js para processar as requisições dos clientes, PostgreSQL como banco de dados principal para armazenar as informações dos usuários e MongoDB para armazenar os logs.

6.1 Diagrama de Pacotes:

Pacotes do projeto:

- **Frontend (UI):**
    - Pacote HTML/CSS/JavaScript
- **Backend (Node.js):**
    - Pacote Routes
    - Pacote Services
    - Pacote Public
    - Pacote Partials
    - Pacote Views
- **Banco de Dados:**
    - Pacote PostgreSQL (relacional)
    - Pacote MongoDB (não relacional)

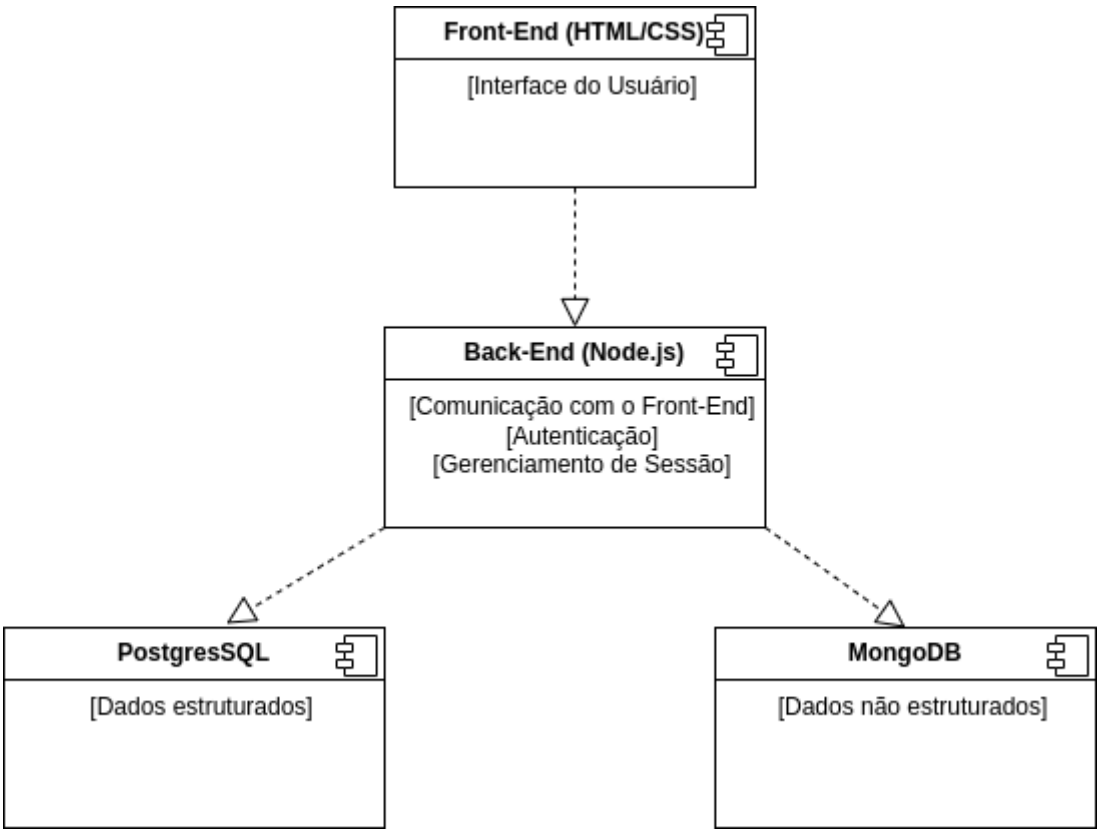


[Diagrama de Pacotes]

6.2 Diagrama de Componentes:

Componentes principais:

- **Frontend (HTML/CSS/JavaScript):**
    - Interações do usuário.
- **Backend (Node.js):**
    - Comunicação com o frontend
    - Autenticação de usuários.
    - Gerenciamento de Sessão
- **Banco de Dados:**
    - Componente PostgreSQL: Gerenciamento de dados dos usuários.
    - Componente MongoDB: Armazenamento de dados não estruturados (conteúdos de autoajuda).

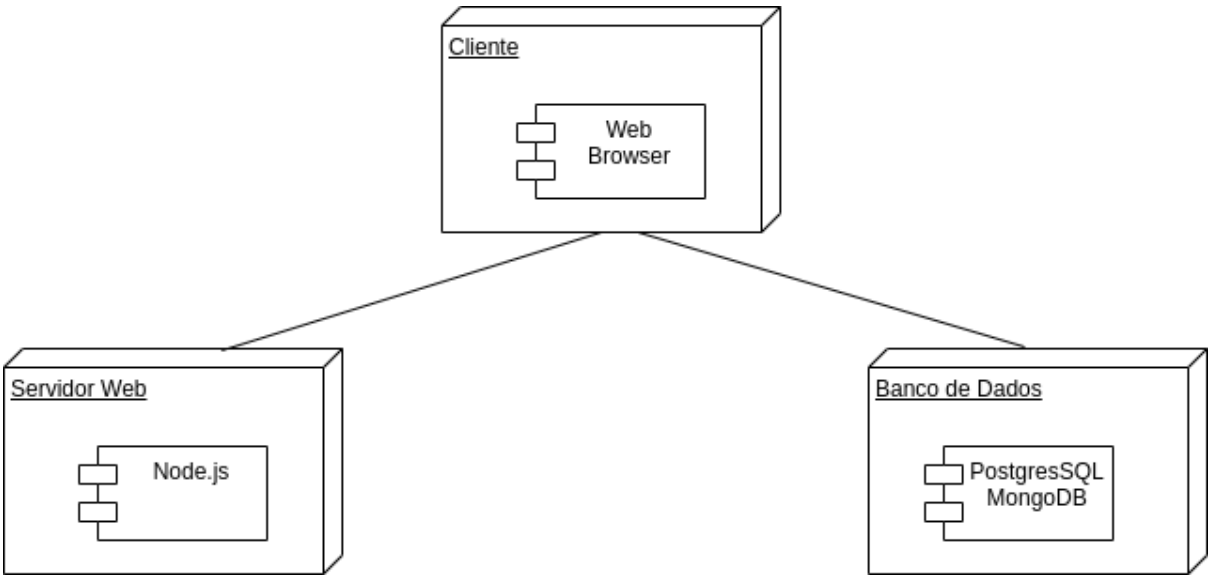


[Diagrama de Componentes]

6.3 Diagrama de Implantação:

Elementos do diagrama de implementação:

- **Cliente (Usuário final):**
    - Dispositivos (computadores, smartphones) acessando a plataforma via navegador.
- **Servidor Web:**
    - Node.js Server (servidor responsável pela lógica da aplicação).
- **Banco de Dados:**
    - PostgreSQL: Gerencia dados dos usuários.
    - MongoDB: Armazena conteúdos de autoajuda (artigos, vídeos, etc.).



[Diagrama de Implantação]

7. Especificação de Requisitos do Sistema

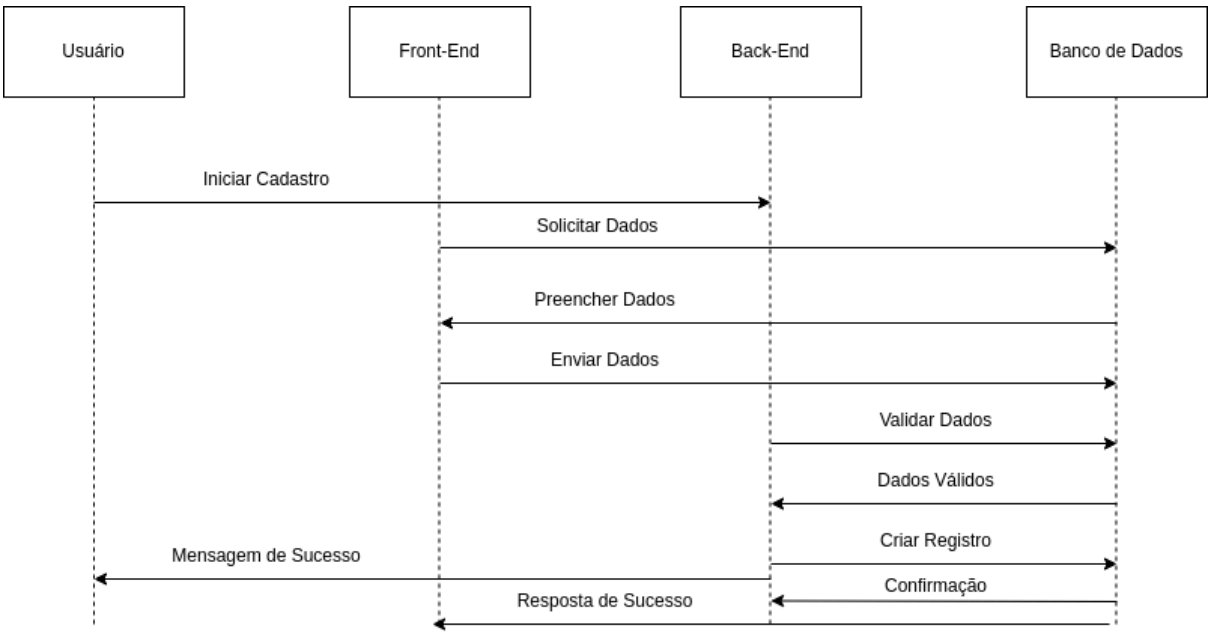
7.1 Diagrama de Sequência:

Pacientes:

1. Usuário
2. Frontend (Interface)
3. Backend (Node.js)
4. Banco de Dados (PostgreSQL)

Sequência de Alterações:

1. Usuário inicia o cadastro na interface.
2. Frontend coleta os dados do usuário (nome, email, senha).
3. Frontend envia os dados para o Backend.
4. Backend valida os dados recebidos.
5. Se os dados forem válidos
    - **Backend** envia uma solicitação ao **Banco de Dados** para criar um novo registro de usuário.
    - **Banco de Dados** confirma a criação do usuário.
    - **Backend** envia uma resposta de sucesso ao **Frontend**.
    - **Frontend** exibe uma mensagem de sucesso ao **Usuário**.
6. Se os dados forem inválidos:
    - **Backend** envia uma mensagem de erro ao **Frontend**.
    - **Frontend** exibe a mensagem de erro ao **Usuário**.



[Diagrama de Sequência]

8. Modelo de Sistema

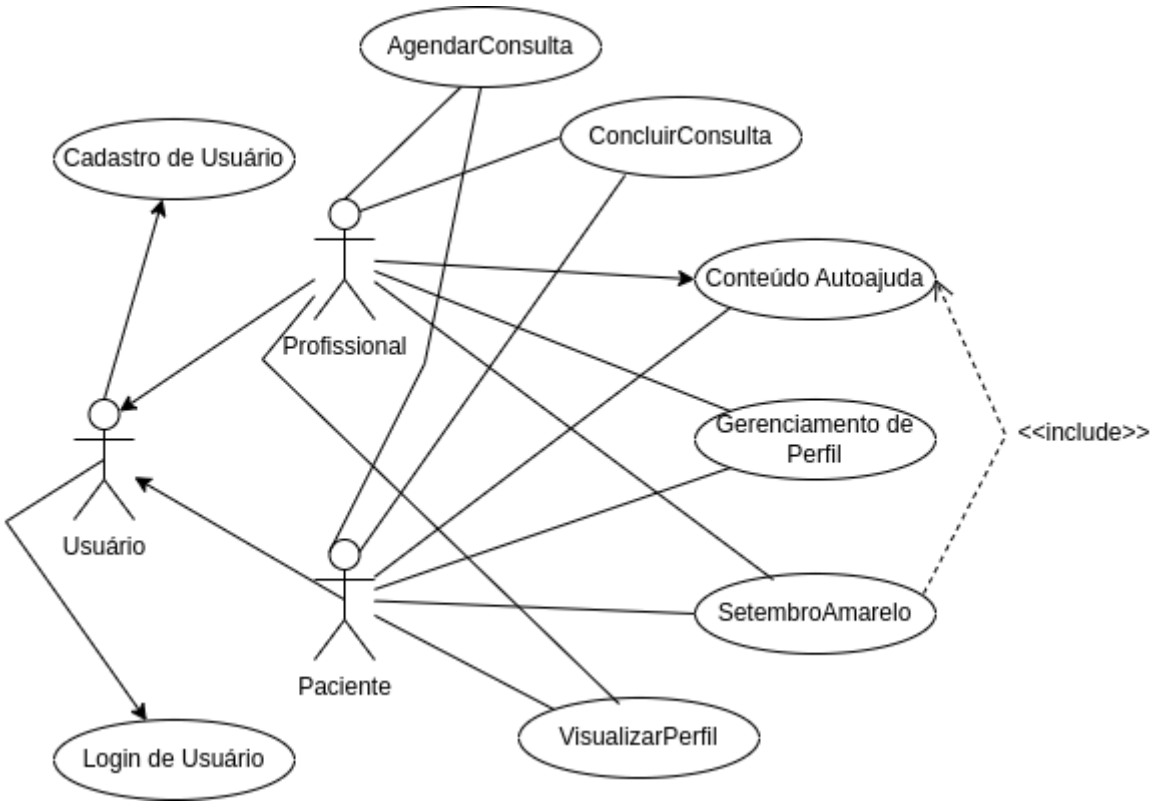
8.1 Diagrama de Caso de Uso:

O **Diagrama de Casos de Uso** demonstra as interações entre os diferentes atores do sistema e as principais funcionalidades oferecidas. Nele, identificamos dois atores principais: o **Paciente** e o **Profissional**, que são representações das duas categorias de usuários.

- O **Paciente** pode realizar ações como visualizar o conteúdo de autoajuda, gerenciar seu perfil e participar de campanhas do **Setembro Amarelo**.
- O **Profissional**, além de criar e gerenciar o conteúdo de autoajuda, pode agendar e concluir consultas com os pacientes.

Além disso, o **Usuário** é responsável pelo processo de cadastro e login no sistema. As funcionalidades como **Agendar Consulta** e **Concluir Consulta** são relacionadas diretamente ao profissional, enquanto o paciente interage com as funcionalidades de visualização e autoajuda.

Por fim, o diagrama utiliza uma relação de inclusão (<<*include*>>) para representar que a funcionalidade do **Setembro Amarelo** está diretamente relacionada à seção de autoajuda, mas com foco específico na campanha de prevenção ao suicídio.



[Diagrama de Caso de Uso]

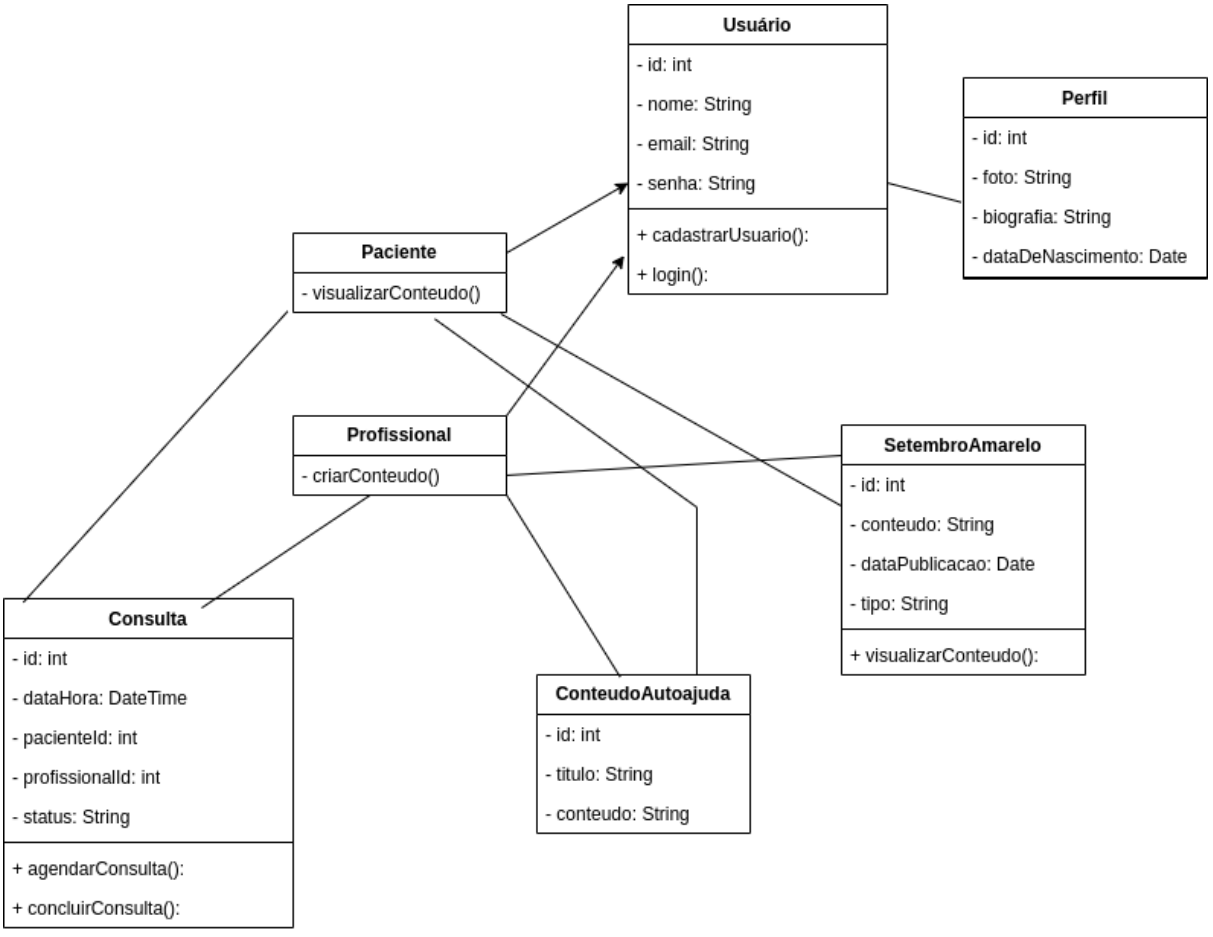
8.2 Diagrama de Classes:

O **Diagrama de Classes** define a estrutura estática do sistema, apresentando as classes principais e seus atributos e métodos. As principais classes modeladas são:

- **Usuário**: Representa tanto pacientes quanto profissionais, contendo informações como nome, email e senha. A classe também possui métodos para cadastro e login de usuários.
- **Paciente** e **Profissional**: Estas classes herdam da classe **Usuário**, adicionando funcionalidades específicas, como o método visualizarConteudo() para o paciente e criarConteudo() para o profissional.
- **ConteudoAutoajuda**: Modela os conteúdos criados pelos profissionais e acessados pelos pacientes. Contém atributos como título e conteúdo, além do método visualizarConteudo() para que o paciente acesse o material.
- **Consulta**: Representa as consultas entre pacientes e profissionais, com atributos como data e hora, status da consulta, além dos métodos agendarConsulta() e concluirConsulta() para controle da interação.

- **SetembroAmarelo**: Classe que modela os conteúdos relacionados à campanha de conscientização, com atributos como data de publicação e tipo de conteúdo (vídeo, artigo). Também contém o método visualizarConteudo() que permite a interação dos pacientes com os materiais da campanha

Este diagrama é essencial para entender como as entidades do sistema estão organizadas e se relacionam, além de evidenciar os métodos e atributos principais que sustentam a lógica de funcionamento do sistema. Ele mostra as interações entre as classes de maneira clara, permitindo uma visão geral da estrutura de software.



[Diagrama de Classes]

9. Evolução do Sistema:

- **Funcionalidades básicas:** Cadastro de usuários, login, e preenchimento de questionários.

- **Interface Simples:** Foco em uma interface intuitiva e amigável para facilitar a navegação.
- **Banco de Dados:** Implementação do PostgreSQL para dados estruturados e MongoDB para logs.

9.1 Fase de expansão:

Novas Funcionalidades:

- Adição de um chat em tempo real entre pacientes e profissionais.
- Seção de autoajuda com vídeos e artigos.
- Personalização do Usuário: Permitir que os usuários personalizem seus perfis e preferências de conteúdo.
- Melhoria na Segurança: Implementação de autenticação em duas etapas e criptografia de dados sensíveis.

9.2 Fase de integração:

Integração com Ferramentas Externas:

- Conexão com APIs de serviços de saúde para acesso a recursos adicionais.
- Integração com plataformas de telemedicina.
- **Feedback e Avaliação:** Sistema de feedback onde usuários podem avaliar profissionais e conteúdos.

9.3 Fase de análise e melhoria contínua:

- **Análise de Dados:** Uso de ferramentas de análise para entender o comportamento dos usuários e melhorar a experiência.
- **Personalização Baseada em Dados:** Algoritmos que sugerem conteúdos e profissionais com base nas interações anteriores.
- **Relatórios de Saúde:** Geração de relatórios que ajudam os usuários a acompanhar seu progresso ao longo do tempo.

10. Apêndices

As imagens dos diagramas utilizados neste projeto, bem como a documentação completa, estão disponíveis no repositório do GitHub. Para acessar o projeto, executar e visualizar todos os detalhes, incluindo os diagramas de sequência, pacotes e componentes, utilize o seguinte link: https://github.com/welson-rodrigues/documento-de-requisitos.git

Este repositório contém todos os arquivos relevantes e informações necessárias para compreender a implementação e funcionamento do sistema.

Referências

PGADMIN. *PostgreSQL Tools*. Disponível em: https://www.pgadmin.org/. Acesso em: 07 set. 2024.

MONGODB INC. *MongoDB: The Developer Data Platform*. Disponível em: https://www.mongodb.com/. Acesso em: 11 set. 2024.

MICROSOFT. *Visual Studio Code: Code Editing. Redefined*. Disponível em: https://code.visualstudio.com/. Acesso em: 31 ago. 2024.

OPENJS FOUNDATION. *Node.js: JavaScript runtime built on Chrome's V8 JavaScript engine*. Disponível em: https://nodejs.org/. Acesso em: 31 ago. 2024.

OPENJS FOUNDATION. *Express: Fast, unopinionated, minimalist web framework for Node.js*. Disponível em: https://expressjs.com/. Acesso em: 31 ago. 2024.

Draw.io. (n.d.). Diagrams made easy. Retrieved. Disponível em: https://app.diagrams.net/ Acesso em: 20 set. 2024.