

Projeto Prático: Sistema Sustentável Orientado a Objetos

UNIVERSIDADE PRESBITERIANA MACKENZIE
Desenvolvimento de Software
Thiago Donizetti dos Santos



Breno Soares Borges	RA:10417553
Bruno Rabelo Torchio de Oliveira	RA:10239373
Caio Teixeira Torres	RA:10417478
Guilherme Dias Ferreira Pereira	RA:10417684
João Victor de Paula Silva	RA:10418556

Sumário:

Tema:	3
Justificativa:	3
Casos de Uso:	4
Caso de uso 1: Registrar Usuário.	4
Caso de Uso 2: Conversar com o Chat	5
Caso de uso 3: Realizar chamada de Suicídio.	6
Caso de uso 4: Agendar consulta.	7
Casos de Uso 5: Perguntar sobre sintomas e doenças.	8
Requisitos Iniciais:	9
1. Consulta sobre sintomas	9
2. Informações sobre doenças	9
3. Orientação para buscar ajuda médica	10
4. Conversas sobre bem estar mental	10
5. Registro de conversas antigas	10
1. Usabilidade	10
2. Segurança da informação	10
3. Escalabilidade	10
Diagramas:	11
Diagrama de caso de uso:	11
Diagrama de Classes:	12
Diagrama de Atividades:	13
Diagrama de Estados:	14
Diagramas de Sequência:	15
Diagramas de Comunicação:	19
Documentação Detalhada:	23
Descrição	23
Arquitetura do Projeto	23
Requisitos	23
Funcionalidades	23
Página Inicial	23
Tela de Registro	23
Tela de Login	23
Interface de Chat	23
Página Sobre	24
Lógica de Negócios	24
Cadastro de Usuários	24
Regras de Resposta do Chatbot	24
Gerenciamento de Estado	24
Flash Messages	24
Detalhamento das Funções e Métodos	24
Função home()	24
Função register()	24
Função login()	24

Função chat()	25
Função sobre()	25
Função resposta_Chat(user_input)	25
Instruções para Usuário:	25
Reflexões Finais:	25
Links:	26

Tema:

O ConverSaudezinha é um chatbot por voz ou texto, inovador projetado para proporcionar um acompanhamento grandioso da saúde. Nosso objetivo é ajudar os usuários a entender melhor suas condições de saúde, pesquisar informações sobre doenças, entender como se cuidar e receber apoio emocional, funcionando como um médico virtual sempre à disposição.

Destinado a todos que enfrentam dificuldades para agendar consultas no SUS ou que não têm condições financeiras para uma consulta particular, o ConverSaudezinha oferece uma alternativa acessível e eficiente. Através de uma interface amigável, os usuários poderão interagir com o chatbot para esclarecer dúvidas sobre sintomas, aprender sobre diversas condições de saúde e acessar recursos sobre bem-estar mental.

É importante ressaltar que, embora o ConverSaudezinha forneça informações úteis e suporte, ele não substitui a consulta médica. Em situações que demandam atenção especializada, o chatbot recomendará que o usuário procure um médico profissional. O agendamento será redirecionado a um atendente que poderá ajudar o usuário com os específicos.

Com o ConverSaudezinha, buscamos democratizar o acesso à informação e ao apoio em saúde, promovendo o bem-estar e a qualidade de vida de nossos usuários de forma prática e acessível, ajudando no objetivo de desenvolvimento sustentável número 3 - saúde e bem estar.

Justificativa:

Acreditamos que a saúde é um tema essencial e complexo, especialmente no Brasil, onde o acesso a hospitais de qualidade pode ser bastante complicado. Além de ser um assunto de extrema importância que é comumente ignorado ou debochado por certas pessoas. Quanto mais difícil for o acesso à saúde, piores se tornam os índices relacionados a ela. Com o nosso chatbot, queremos ajudar as pessoas a se cuidarem melhor, entenderem quando devem procurar um médico e reconhecerem a importância de seguir corretamente os tratamentos.

É fundamental que a sociedade compreenda que um tratamento adequado é essencial para o sucesso na luta contra qualquer doença. Por isso, com o ConverSaudezinha, vamos fornecer informações valiosas e acessíveis, ajudando diretamente as pessoas a tomarem decisões mais informadas sobre sua saúde.

Casos de Uso:

Caso de uso 1: Registrar Usuário.

Resumo: Neste caso de uso, o usuário decide se registrar no site/app para acessar o chatbot e seus serviços.

Ator Principal: Usuário

Pré-condições: O sistema estar em pleno funcionamento.

Pós-condições: O usuário se torna um usuário registrado e apto a acessar o sistema.

Fluxo Principal

1. O usuário abre o site/app e escolhe a opção de registro.
2. O sistema solicita os dados iniciais para o registro: nome completo, CPF, idade e endereço.
3. O usuário fornece os dados solicitados.
4. O sistema solicita que o usuário escolha uma senha com no mínimo 8 caracteres, incluindo ao menos dois caracteres especiais, e que a forneça duas vezes para confirmação.
5. O sistema verifica que a senha atende aos critérios estabelecidos e que ambas as entradas coincidem.
6. O sistema registra o usuário como Usuário.
7. O sistema informa ao usuário que ele está registrado e pronto para acessar o sistema.

Fluxos Alternativos

Fluxo Alternativo 1 - Cadastro Simplificado:

- **Descrição:** Caso o usuário prefira preencher apenas dados básicos (nome e idade) para iniciar, ele pode prosseguir com um cadastro simplificado.
- **Passo 2 do Fluxo Principal:** O sistema oferece a opção de um registro básico, limitando certos serviços até que o cadastro completo seja feito.
- **Retorno:** O caso de uso retorna ao passo 6 do Fluxo Principal.

Fluxo Alternativo 2 - Verificação de CPF via API:

- **Descrição:** O sistema utiliza uma API para validar o CPF automaticamente ao ser preenchido pelo usuário.
- **Passo 3 do Fluxo Principal:** O sistema verifica o CPF em tempo real, minimizando o erro e facilitando o preenchimento.
- **Retorno:** O caso de uso retorna ao passo 4 do Fluxo Principal.

Fluxos de Exceção

- **Passo 3 (CPF/endereço inválido):**
 - O sistema detecta CPF e/ou endereço inválido. O caso de uso retorna ao passo 2 do fluxo principal.
- **Passo 4 (senha inválida):**
 - O sistema verifica que a senha não atende aos critérios estabelecidos e solicita que o usuário escolha outra senha. O caso de uso retorna ao passo 4 do fluxo principal.
- **Passo 5 (as duas entradas da senha não coincidem):**
 - O sistema detecta que as entradas de senha não conferem e solicita que o usuário preencha novamente os campos. O caso de uso retorna ao passo 4 do fluxo principal.

Caso de Uso 2: Conversar com o Chat

Resumo: Este caso de uso descreve a interação geral do usuário com o chat, possibilitando que ele faça perguntas, busque informações sobre saúde e receba apoio emocional.

Ator Principal: Usuário

Pré-condições: O sistema está funcionando corretamente e o usuário consegue acessar o chat.

Pós-condições: O usuário recebe as informações ou o apoio emocional desejado.

Fluxo Principal:

1. O usuário inicia uma conversa com o chat.
2. O sistema responde com uma saudação acolhedora, estabelecendo um ambiente amigável para a interação.
3. O usuário prossegue com perguntas ou solicitações.
4. O sistema analisa as solicitações e decide qual ação realizar com base no tipo de interação solicitado pelo usuário.

Extends:

- **Prevenção ao Suicídio:** Em situações de alta tensão emocional, o sistema reconhece a necessidade de apoio e ativa o caso de uso de prevenção ao suicídio.
- **Agendar Consulta:** O usuário solicita ajuda para agendar uma consulta médica, ativando o caso de uso de agendamento.
- **Sintomas e Doenças:** O usuário busca informações sobre sintomas ou doenças, e o sistema ativa o caso de uso correspondente para fornecer informações detalhadas.

Caso de uso 3: Realizar chamada de Suicídio.

Extendido de: Conversar com o Chat

Resumo: O chat ao perceber certo nível de tensão e/ou alguns tipos de perguntas e falas irá aconselhar e direcionar o usuário para a chamada de prevenção ao suicídio cedida pelo governo (número 188).

Ator principal: Usuário

Pré-condições: Nível de tensão do usuário e frases como: “Preferia estar morto”, “Vão ser mais felizes sem mim” entre outras escritas ou ditas pelo usuário.

Pós-condições: Usuário será redirecionado para o chat ou ligação por voz de prevenção ao suicídio.

Fluxo principal:

1. Chat iniciado com uma amigável saudação.
2. O chat monitora continuamente o conteúdo das mensagens do usuário. Se detectar frases associadas com grandes níveis de tensão, ou expressões que indicam desespero ou falta de esperança, o sistema ativa uma resposta de suporte.
3. Quando as frases são detectadas, o sistema classifica a situação como uma possível crise emocional. A resposta do sistema deve ser atenciosa, mostrando empatia e preocupação.
4. O chat oferece uma orientação, recomendando o número de prevenção ao suicídio 188 ou o contato com profissionais de ajuda emocional.
5. O sistema oferece para continuar a conversa reforçando a ideia de que a pessoa é importante e eles são “amigos” e deve procurar ajuda.
6. Ao final da interação, o sistema reforça uma mensagem positiva para o usuário incentivando-o a continuar na luta e a buscar ajuda. Exemplo: “Você é forte, continue firme, as coisas sempre melhoram, estou aqui sempre que precisar”.

Fluxo de exceção:

Passo 4 (Usuário recusa ajuda):

- Usuário recusa ajuda direta, o sistema respeita a decisão do usuário mas continua oferecendo apoio e alternativas menos diretas.

Passo 5 (Sem retorno):

- O sistema detecta que o usuário não respondeu, ele tenta novamente após um tempo de espera com uma abordagem mais cuidadosa.

Fluxo Alternativo:

1. Após intervenção do sistema o usuário demonstra que está começando a se sentir melhor ou que a conversa está ajudando. O sistema continua oferecendo apoio, sem forçar mais intervenções, podendo perguntar sobre outros assuntos para distrair o usuário.

Caso de uso 4: Agendar consulta.

Extendido de: Conversar com o Chat

Resumo: O sistema oferece ajuda para agendar uma consulta para o usuário, oferecendo números de telefones, sites entre outros de hospitais ou clínicas próximas ao endereço do usuário.

Ator principal: Usuário

Pré-condições: O sistema saber endereço entre outras informações do usuário.

Pós-condições: O usuário conclui o agendamento da consulta.

Fluxo principal:

1. O usuário solicita ajuda para agendar uma consulta com médico ou psicólogo.
2. O sistema exibe uma mensagem perguntando qual tipo de profissional e especialização o usuário estaria buscando.
3. O usuário seleciona o tipo de profissional desejado
4. O sistema pergunta qual tipo de modalidade o usuário quer, presencial ou online.
5. Se for escolhido presencial, o sistema confirma o endereço do usuário e se for online apenas passa números e emails de clínicas ou médicos que atendem.
6. Após confirmar o endereço, o sistema usa o cep fornecido para buscar clínicas, hospitais ou médicos na região.
7. O sistema apresenta uma lista de clínicas ou médicos próximos ao usuário com várias informações, tais como: endereço, nome da clínica, número de telefone e site.

8. Após o redirecionamento, o sistema pergunta se o usuário necessita de mais alguma ajuda e se a resposta for não, encerra o chat com uma mensagem amigável.

Fluxo de exceção:

Passo 3 (Especialização não especificada):

- O usuário não sabe qual profissional precisa. O sistema faz perguntas e o ajuda a orientar, com base nos sintomas, qual seria o profissional mais adequado.

Passo 6 (Não existem clínicas disponíveis):

- O sistema não localiza clínicas disponíveis próximas ao endereço do usuário. Após isso, oferece opções alternativas, como consultas online ou clínicas em cidades próximas.

Fluxo Alternativo:

1. O usuário solicita opções para um futuro. O sistema oferece opções para enviar informações por email ou armazenar a busca para futura referência.

Casos de Uso 5: Perguntar sobre sintomas e doenças.

Extendido de: Conversar com o Chat

Resumo: O usuário deseja saber mais sobre tratamentos, sintomas ou doenças no geral. Ao perguntar ao chat, o próprio buscará informações sobre a opção desejada na internet, oferecendo essas informações de maneira rápida e eficiente ao usuário e aconselhando a procurar um médico em casos mais sérios.

Ator principal: Usuário

Pré-condições: O sistema ter a capacidade de buscar informações na internet.

Pós-condições: O usuário estará bem aconselhado e com informações úteis sobre doenças e/ou sintomas

Fluxo principal:

1. O usuário inicia o chat perguntando sobre doenças ou sintomas.
2. O sistema responde com uma mensagem acolhedora, perguntando se possui mais alguma coisa que o usuário queira saber sobre determinado sintoma ou doença.
3. O usuário descreve sintomas, doenças ou tratamentos dos quais ele deseja mais informações.
4. O sistema busca informações atualizadas na internet, como em sites de saúde, OMS, Albert Einstein, SUS e etc.

5. O sistema após a pesquisa oferece as informações imprimindo uma mensagem do por que é importante seguir um tratamento correto. Como você pode se cuidar e também claro, os sintomas e informações sobre a doença.
6. Dependendo da gravidade da doença o sistema oferece ajuda para marcar uma consulta (Casos de uso 3)
7. O sistema encerra a conversa de forma aberta, perguntando ao usuário se ele deseja mais alguma informação e sempre lembrando de seguir o tratamento correto, procurar ajuda e não confiar só nas informações do chat.

Fluxo de exceção:

Passo 4 (Sem acesso a informações):

- O sistema não foi capaz de achar informações confiáveis pois os sintomas são muito complexos. O sistema pede desculpas e sugere que o usuário encontre um médico e se oferece para ajudar na procura.

Passo 4 (Sem conexão):

- Conexão ou erro de busca. O sistema não consegue se conectar com a internet. Pede desculpas ao usuário e pede para tentar novamente mais tarde.

Fluxos Alternativos:

1. O usuário pede informações complexas sobre tratamento. O sistema oferece essas informações mas nunca dando nomes de remédio e sempre aconselhando a procurar um médico pois o tratamento ideal depende de cada caso.
2. O usuário pergunta de uma maneira mais geral sobre doenças ou sintomas. O sistema oferece uma explicação mais rasa e de fácil entendimento.

Requisitos Iniciais:

Como Requisitos funcionais podemos citar:

1. Consulta sobre sintomas

- O sistema deve ser capaz de receber sintomas e dizer possíveis casos de saúde relacionados assim como possíveis tratamentos. E aconselhar que o usuário procure um médico em casos mais sérios.

2. Informações sobre doenças

- O sistema deve permitir que usuário pergunte sobre doenças e ele mostra informações pertinentes, como causas, sintomas e como é importante seguir o tratamento sobre tal doença.

3. Orientação para buscar ajuda médica

- O chatbot deve ser capaz de identificar casos que necessitam de atendimento médico.
- O chatbot deve ter a capacidade de checar e entrar em contato com centros de atendimento.

4. Conversas sobre bem estar mental

- O sistema deve ser capaz de dar dicas/direções para a pessoa poder se ajudar de forma saudável.

5. Registro de conversas antigas

- O sistema deve guardar conversas antigas com o usuário para que o usuário possa consultar informações previamente ditas e estabelecidas. Por exemplo, recomendar não comer tal alimento por conta de diabetes.

Como Requisitos NÃO funcionais que podemos citar:

1. Usabilidade

- O chatbot deve ter uma interface de fácil entendimento e amigável, também deve oferecer respostas rápidas.

2. Segurança da informação

- O chatbot deve ter altos padrões de segurança para que as informações de todos os usuários estejam seguras

3. Escalabilidade

- O sistema deve ser capaz de suportar grandes números de usuários.

Diagramas:

Diagrama de caso de uso:

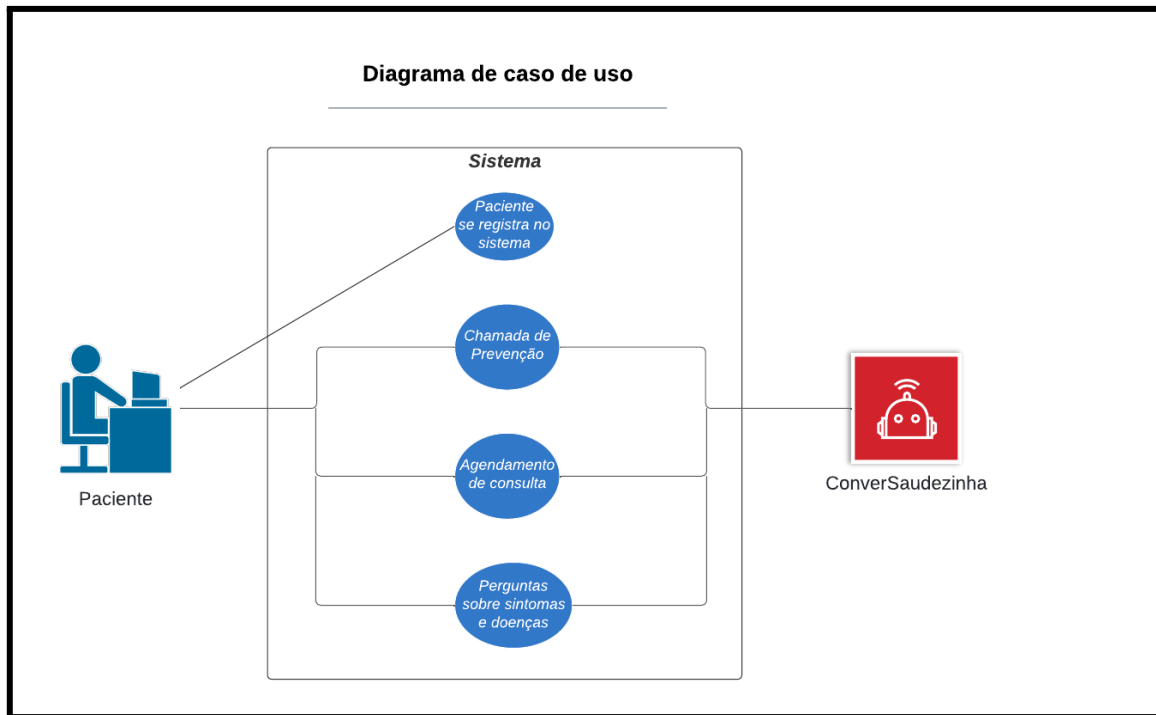


Diagrama de Classes:

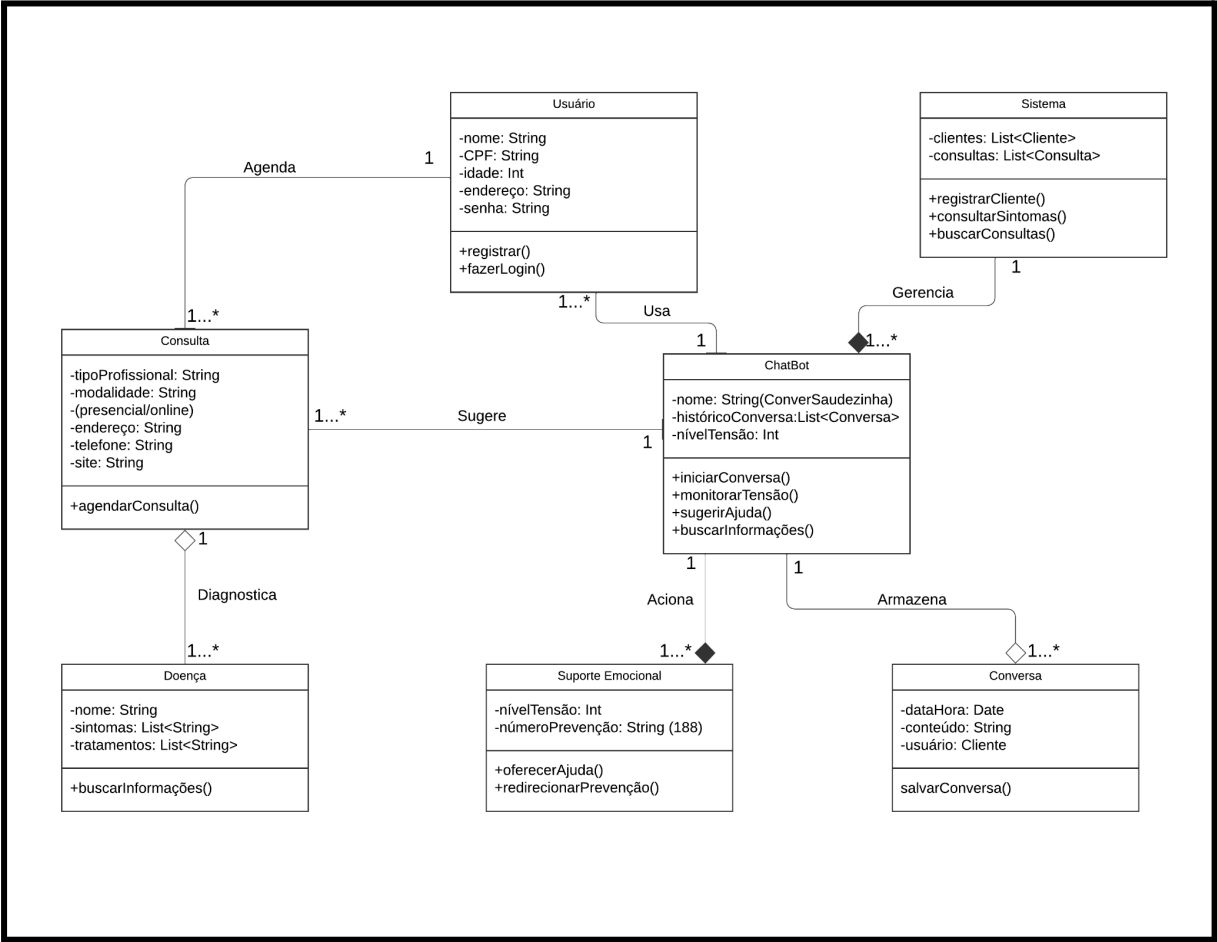


Diagrama de Atividades:

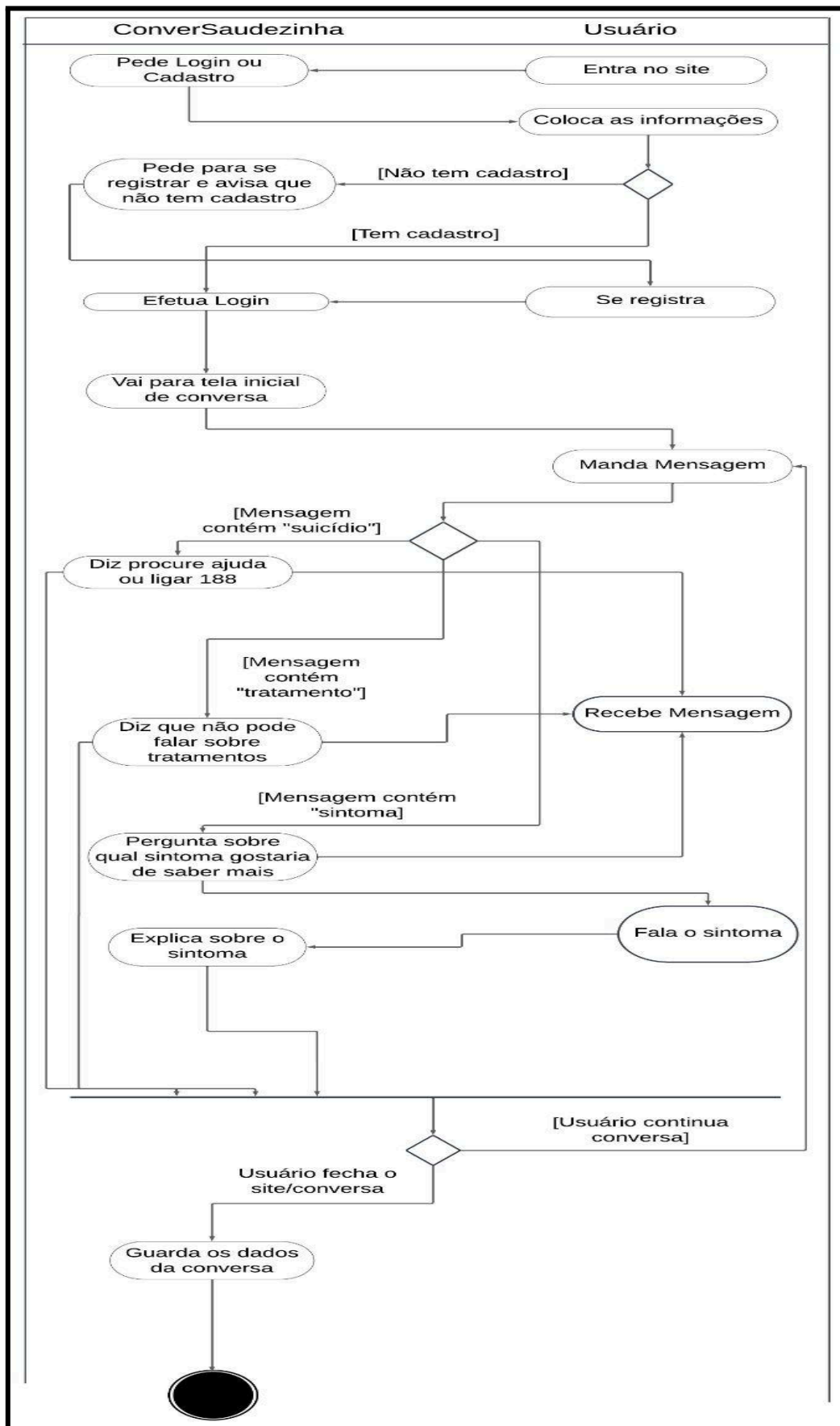
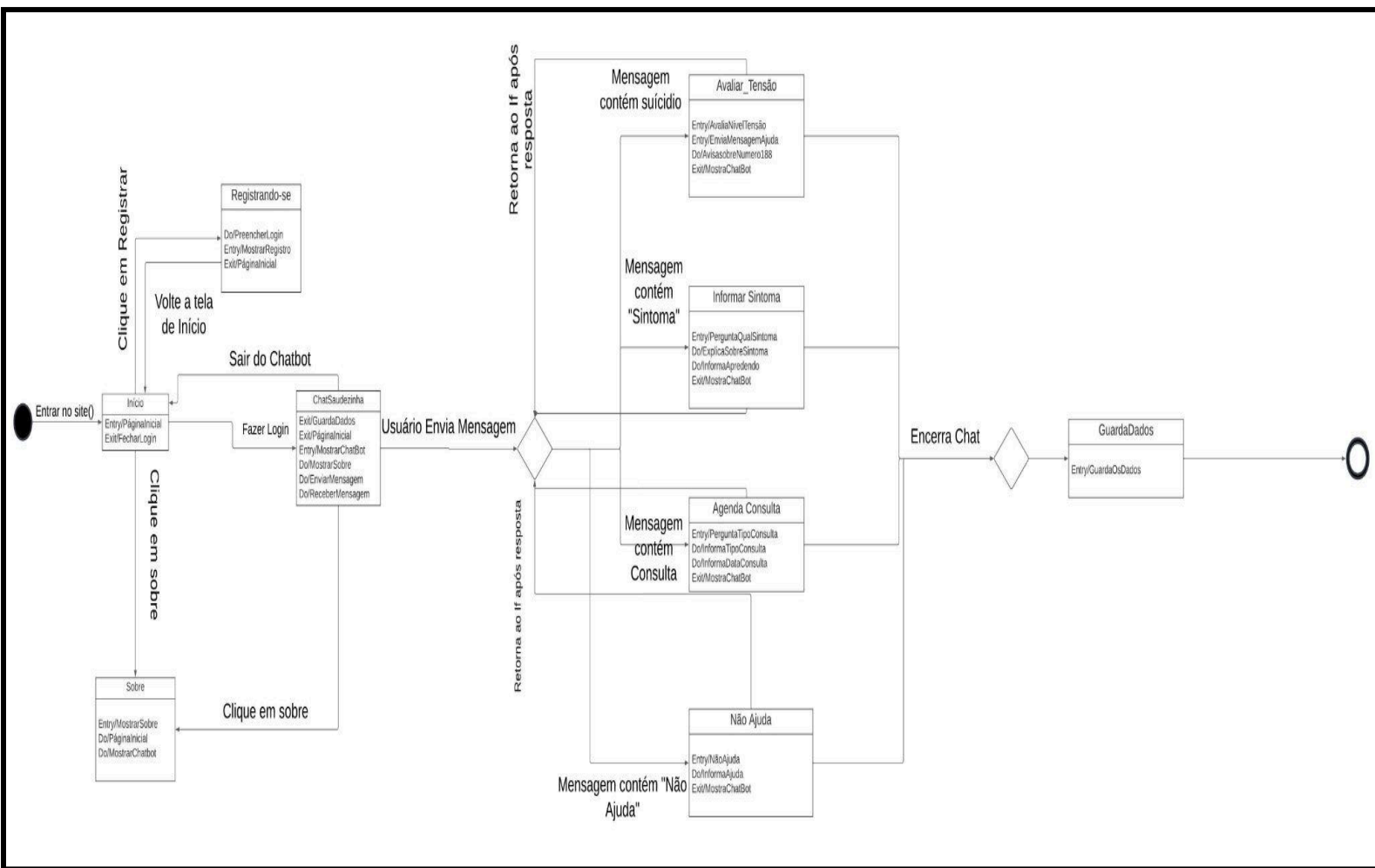
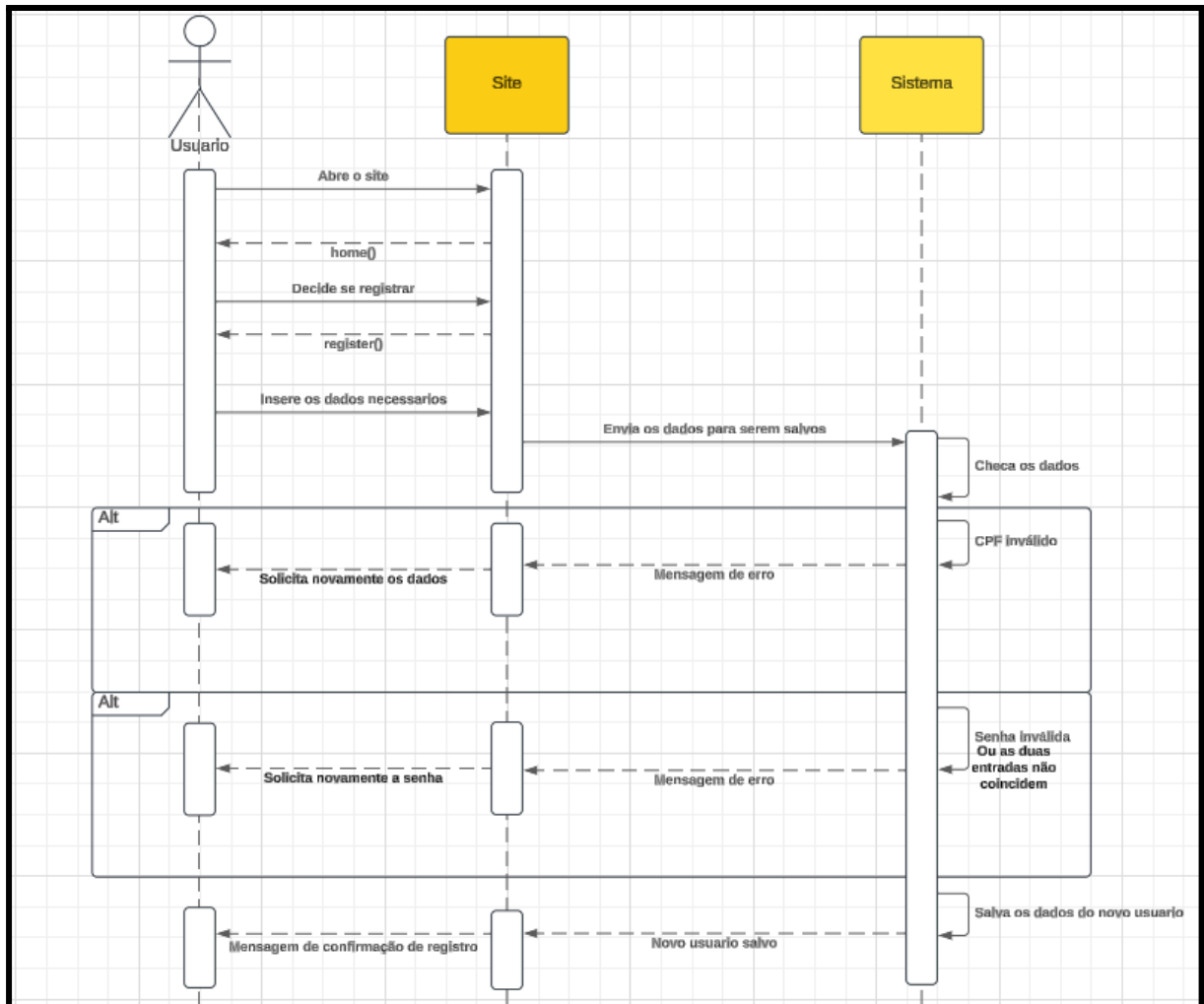


Diagrama de Estados:

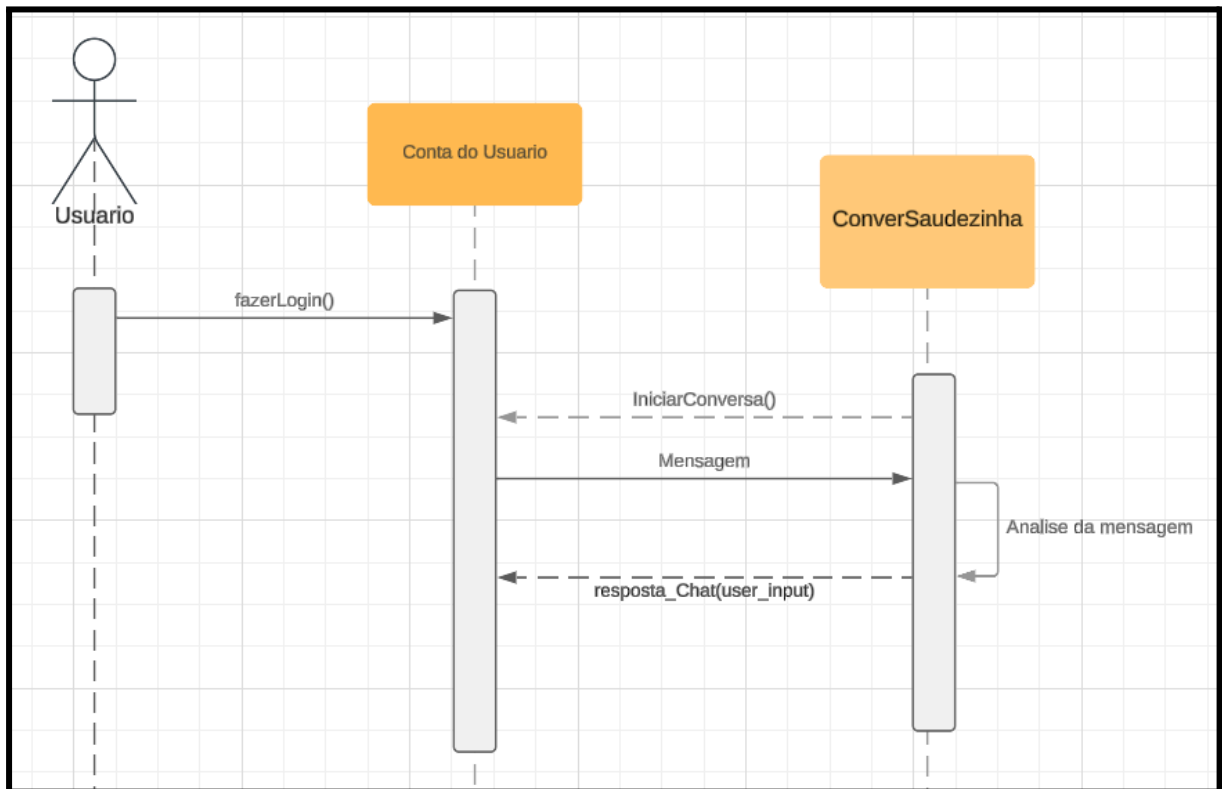


Diagramas de Sequência:

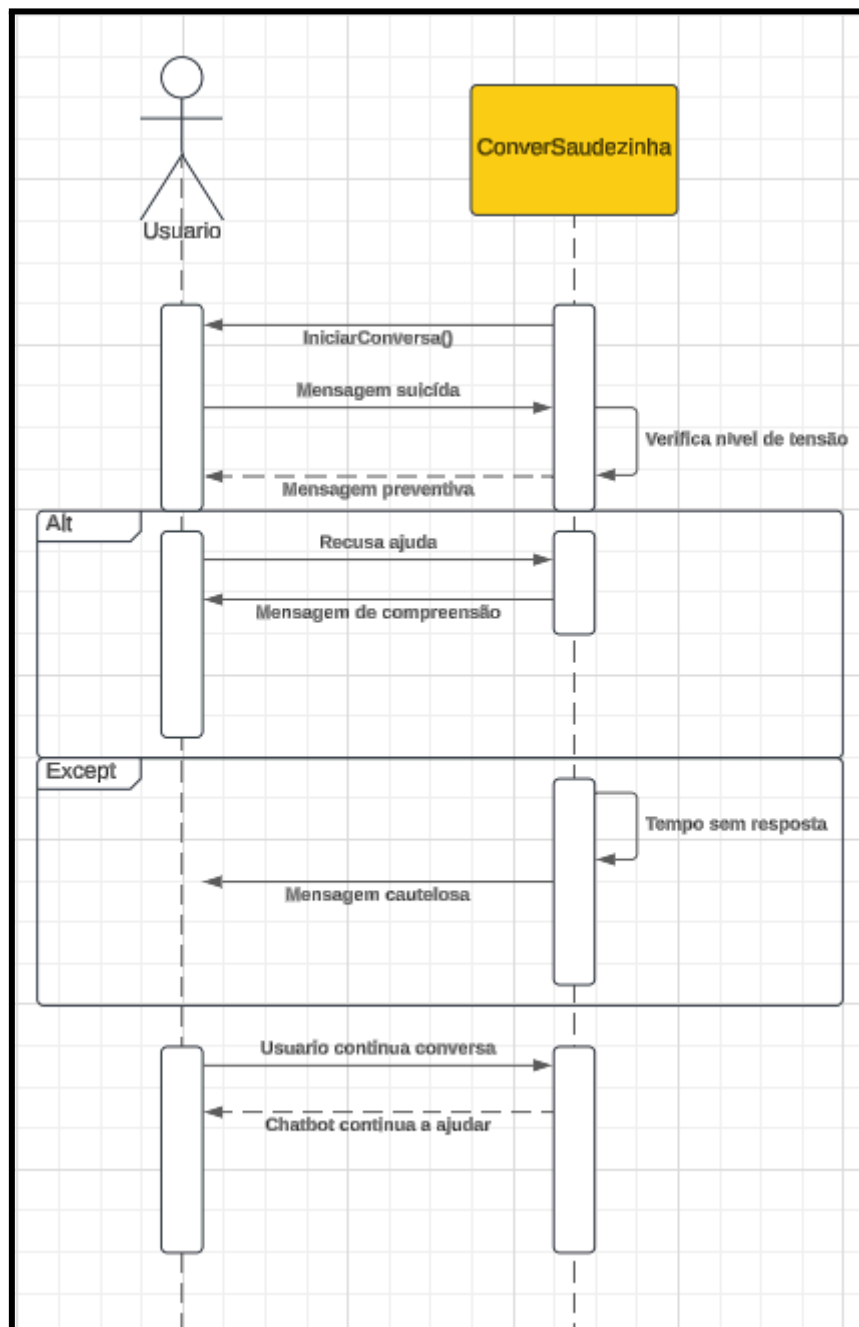
Caso de uso 1: Registrar usuário.



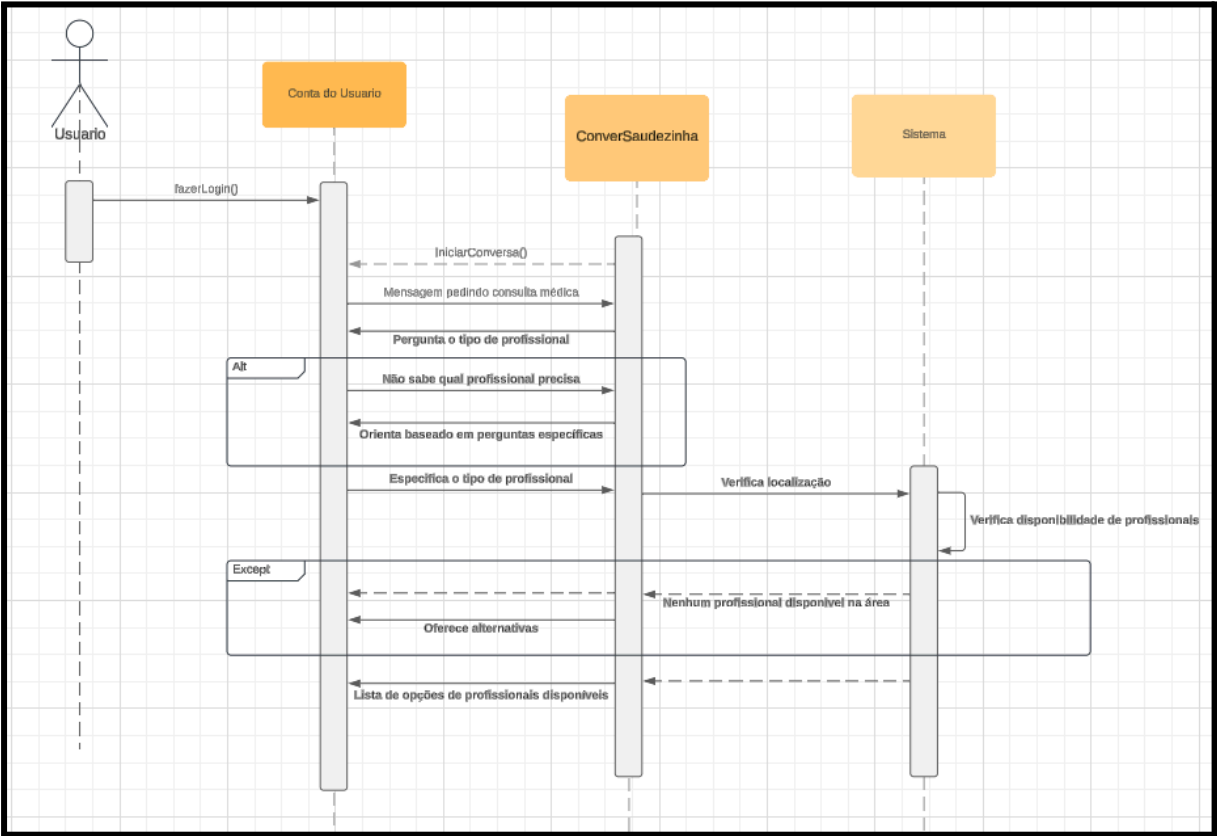
Caso de Uso 2: Conversar com o Chat



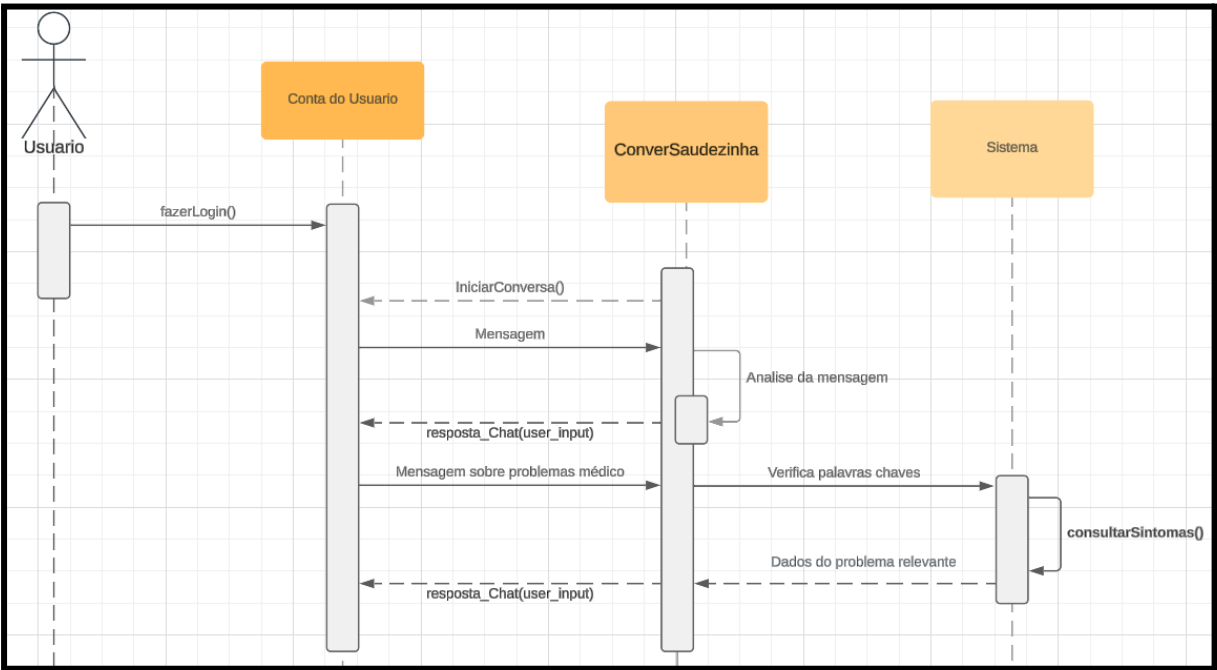
Caso de uso 3: Realizar chamada de Suicídio.



Caso de uso 4: Agendar consulta.

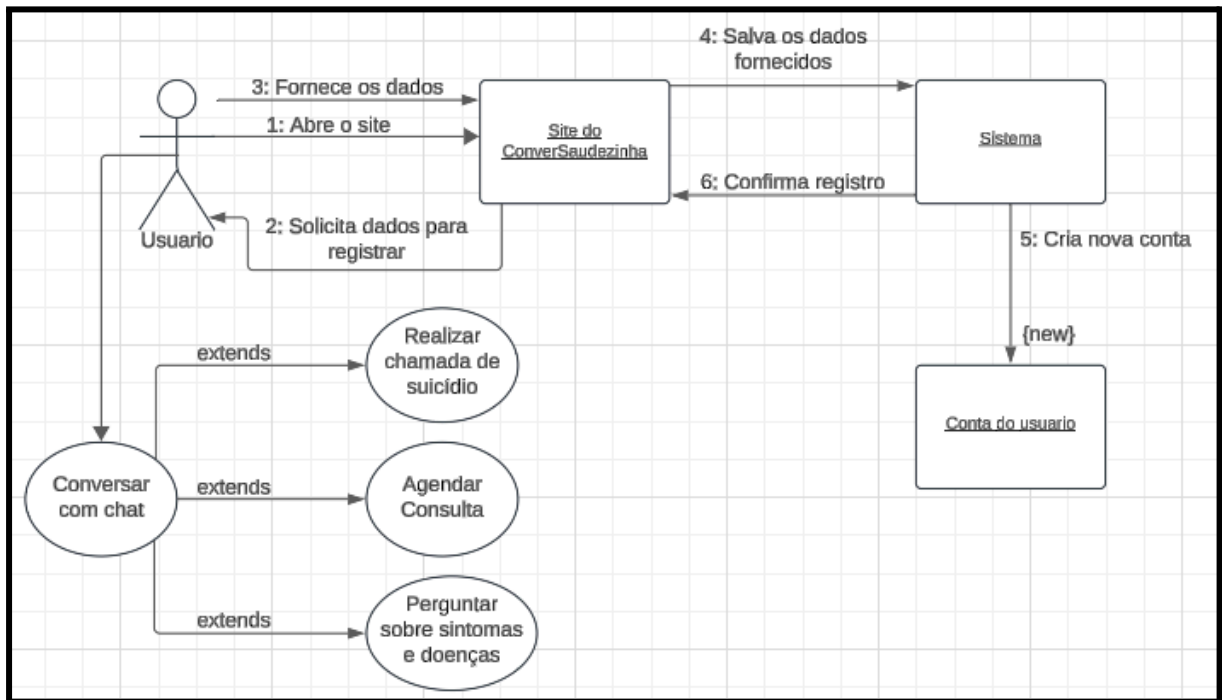


Casos de Uso 5: Perguntar sobre sintomas e doenças.

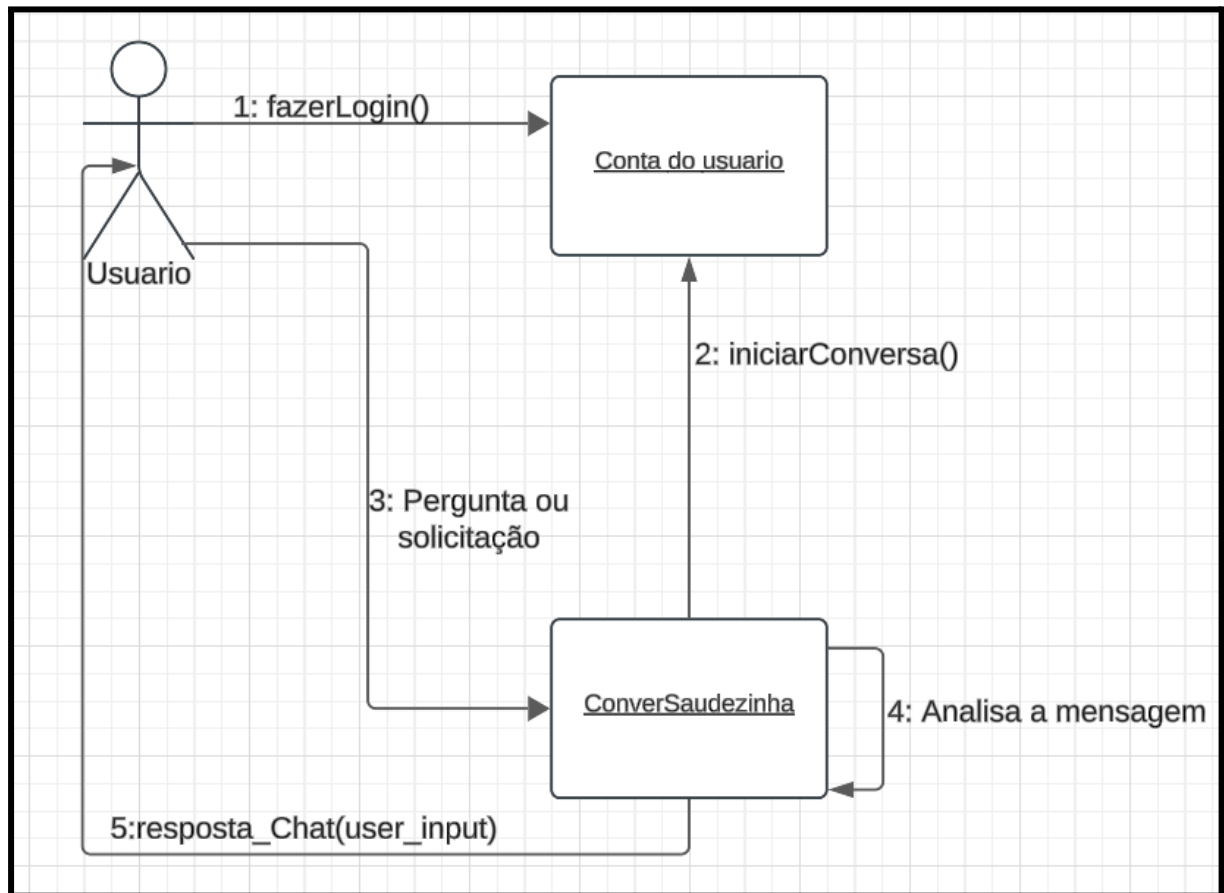


Diagramas de Comunicação:

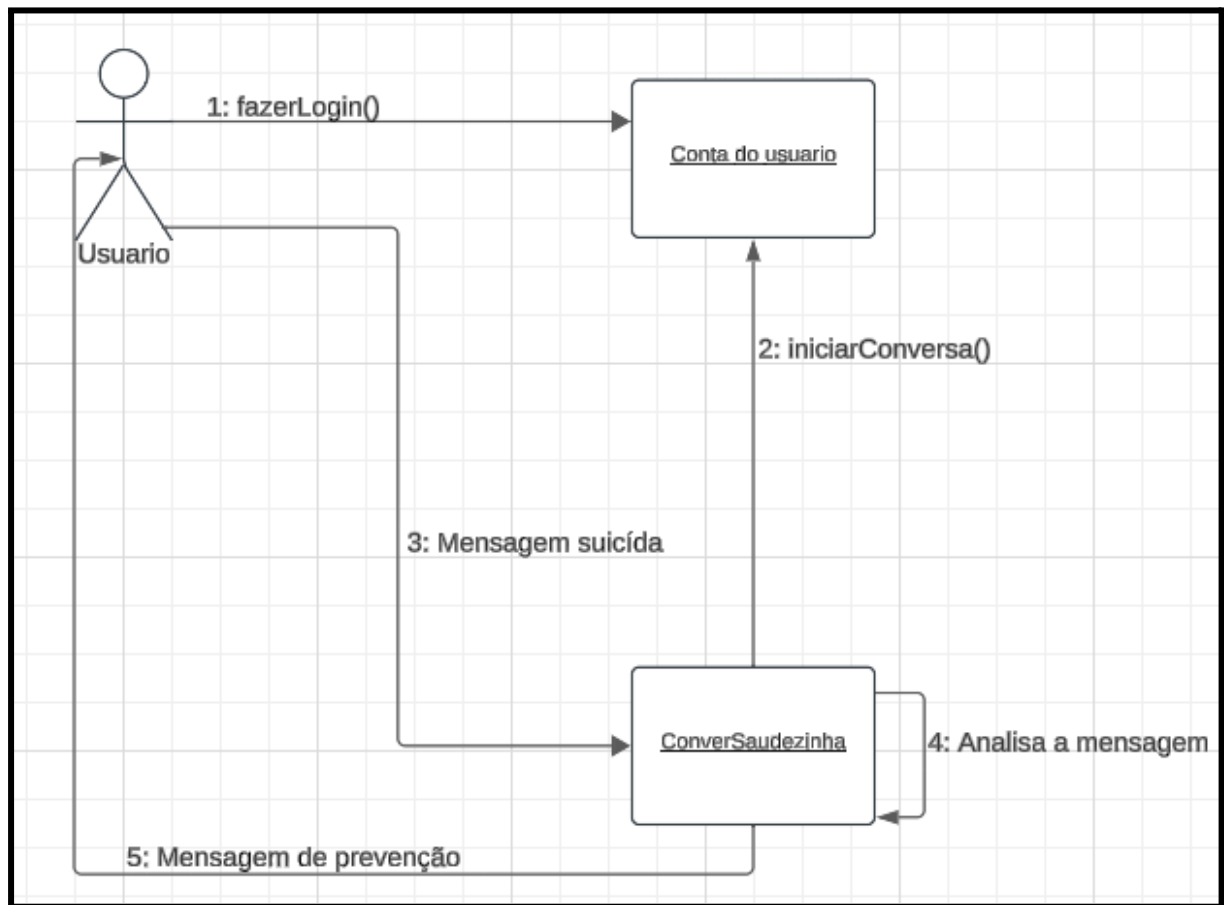
Caso de uso 1: Registrar Usuário



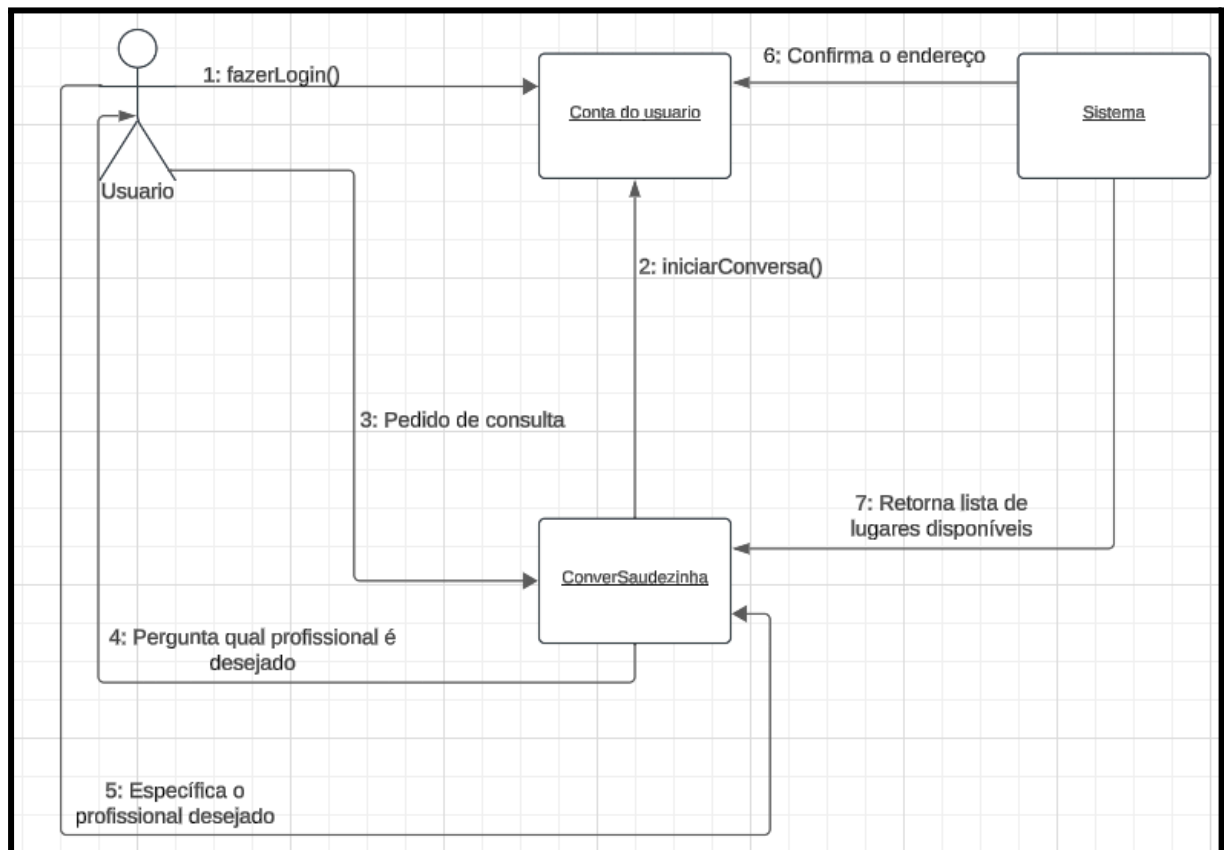
Caso de Uso 2: Conversar com o Chat



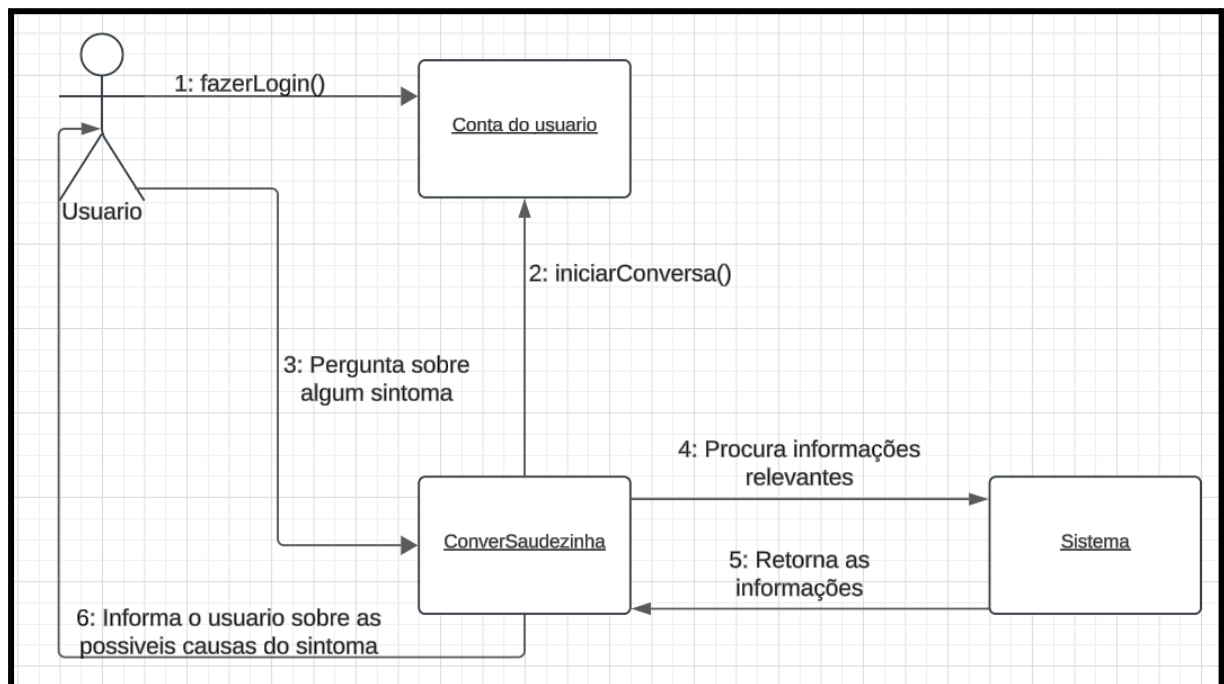
Caso de uso 3: Realizar chamada de Suicídio.



Caso de uso 4: Agendar consulta.



Casos de Uso 5: Perguntar sobre sintomas e doenças.



Documentação Detalhada:

Descrição

O ConverSaudezinha é um chatbot desenvolvido em Flask para fornecer suporte em saúde mental. O foco principal é a prevenção ao suicídio e o apoio psicológico. Além disso, o chatbot pode auxiliar na marcação de consultas presenciais e online, e oferece informações sobre sintomas e doenças, com uma abordagem acolhedora e empática.

Arquitetura do Projeto

Flask Application: O código é estruturado em torno de um único aplicativo Flask. A aplicação gerencia as rotas para o login, registro de usuários, interações no chat, e exibição de informações sobre o chatbot.

Sessions: O sistema utiliza session para armazenar dados temporários dos usuários, como o nome do usuário, durante a interação no chat.

Requisitos

- Python 3.x: Linguagem utilizada para o desenvolvimento.
- Flask: Framework web para Python, utilizado para criar o servidor e gerenciar as rotas.
- Jinja2: Motor de templates utilizado pelo Flask para gerar HTML dinâmico.
- Flash: Função do Flask para exibir mensagens de feedback temporárias.
- Session: Para manter o estado do usuário durante a navegação no sistema.

Funcionalidades

Página Inicial

- Rota: /
- A página inicial apresenta um formulário de login.

Tela de Registro

- Rota: /registrar
- Permite o cadastro de novos usuários. O formulário exige o nome, CPF, idade, endereço, senha e confirmação de senha.
- Caso as senhas não coincidam ou o CPF já tenha sido registrado, o sistema exibe uma mensagem de erro.

Tela de Login

- Rota: /login
- Realiza a validação do login com base no nome e senha fornecidos.
- Se as credenciais forem válidas, o usuário é redirecionado para a página de chat. Caso contrário, uma mensagem de erro é exibida.

Interface de Chat

- Rota: /chat
- A página de chat permite que o usuário interaja com o chatbot.

- O usuário envia mensagens e o chatbot responde de acordo com regras específicas, como apoio em crises de suicídio, informações sobre sintomas e doenças, e agendamento de consultas.

Página Sobre

- Rota: /sobre
- A página apresenta informações sobre o projeto, sua finalidade e os objetivos.

Lógica de Negócios

Cadastro de Usuários

- Armazenamento: Dados do usuário (nome, CPF, idade, endereço e senha) são armazenados temporariamente em um dicionário users.
- Validação: Se o CPF já existir, o sistema não permite o cadastro de um novo usuário com o mesmo CPF. Também há validação das senhas para garantir que coincidam.

Regras de Resposta do Chatbot

- Suporte a Crise: Respostas empáticas e de apoio para situações de risco de suicídio.
- Consultas: Informações sobre como agendar consultas presenciais e online.
- Sintomas e Doenças: O chatbot fornece informações sobre sintomas e doenças e auxilia o usuário a identificar condições de saúde. Dependendo do sintoma relatado, o chatbot sugere acompanhamento médico.
- Tratamento: O chatbot orienta o usuário a procurar ajuda especializada para tratamentos médicos, sem fornecer diagnósticos ou tratamentos diretamente.

Gerenciamento de Estado

- Session: O estado do usuário, como o nome e o histórico de conversa, é mantido na sessão (session), o que permite que o usuário continue a conversa sem perder o contexto.

Flash Messages

- O Flash é utilizado para mostrar mensagens de feedback temporárias ao usuário, como alertas de erro, confirmações de ações e dicas.

Detalhamento das Funções e Métodos

Função home()

- Descrição: Rota inicial que renderiza o template de login.
- Método HTTP: GET

Função register()

- Descrição: Rota de registro de novos usuários. Valida o formulário de registro e, se válido, armazena os dados na variável users.
- Método HTTP: GET (exibe o formulário) e POST (processa os dados do formulário).

Função login()

- Descrição: Rota de login. Verifica as credenciais fornecidas e, se válidas, redireciona para a página de chat.

- Método HTTP: POST

Função chat()

- Descrição: Rota de chat, onde o usuário interage com o chatbot. O histórico de mensagens é mantido e exibido.
- Método HTTP: GET (exibe a página de chat) e POST (processa a entrada do usuário e gera a resposta do chatbot).

Função sobre()

- Descrição: Rota que exibe a página "Sobre" do projeto.
- Método HTTP: GET

Função resposta_Chat(user_input)

- Descrição: Função central que processa a entrada do usuário e gera uma resposta do chatbot com base nas regras predefinidas.
- Lógica: Respostas são baseadas em palavras-chave, como "suicídio", "consulta", "sintoma", entre outras. A função também gerencia o nível de gravidade da situação (ex: nível de tensão do usuário) e oferece respostas adequadas.

Instruções para Usuário:

Tela de Login: Ao acessar o protótipo, você será inicialmente direcionado para a tela de login. Caso ainda não tenha uma conta, será necessário realizar o registro;

Registro de Usuário: Clique no botão "Registre-se" para acessar o formulário de cadastro. Preencha as informações solicitadas, como nome, CPF, idade, endereço e senha. Após preencher o formulário, clique em "Registrar";

Login: Após o registro, você será redirecionado automaticamente de volta para a tela de login. Insira o nome de usuário e senha cadastrados para acessar sua conta;

Acesso ao Chat: Após realizar o login com sucesso, você será redirecionado para a página de chat. Nela, você poderá digitar suas dúvidas ou necessidades na caixa de texto. O chatbot responderá automaticamente, oferecendo suporte em saúde mental, agendamento de consultas e informações sobre sintomas e doenças;

Interação: O chatbot responderá de forma empática e acolhedora, oferecendo assistência conforme suas interações;

Reflexões Finais:

Do ponto de vista técnico, a criação do **ConverSaudezinha** destacou a importância de ferramentas como os diagramas de classe, de estados e de atividades. Esses elementos não apenas nos ajudaram a planejar melhor o sistema, mas também garantiram um código mais organizado, coeso e eficiente. Entendemos que esses

recursos são fundamentais para nortear o desenvolvimento e, a partir desta experiência, pretendemos utilizá-los continuamente em futuros projetos. Outro aprendizado importante foi a importância de uma documentação sólida e bem estruturada. Definir claramente o nosso objetivo é o caminho para alcançá-lo, registrando decisões cruciais, alterações no código e a função de cada parte, nos proporcionou uma visão mais clara e organizada do projeto. Isso facilitou a execução e também garantiu que as informações estivessem acessíveis e compreensíveis para todos, tornando o processo de construção mais eficiente e permitindo que concretizássemos o projeto de forma mais fácil.

Links:

<https://youtu.be/FKWgKqf8M0w>

[GitHub - ConverSaudezinha](#)