**Documento de Requisitos de Sistema**

**Instituição:** Universidade Cidade de São Paulo - UNICID

**Nome do projeto:** UNIChat

**Orientador:** Professor Rodolfo Riyoei Goya

**Local e Data:** SÃO PAULO - SP - 04/10/2022

|  |
| --- |
| Autores |
| Enzo Leone Santana da Silva |
| Victor Sanches Rodrigues Mitsubayashi |
| Diego Louzeiro Pinheiro |
| Thiago Ramos dos Santos |
| Yago Matos dos Santos |
| Weverton Viana Lima |
| Nahuan de Abreu Silva |
| Vitor Alexandre Fernandes da Silva |
| Matheus Akihito Masuda |
| Maycon Cavalcante De Souza |
| Renan Barbosa Rodrigues |
| Lucas Maranhão Braga de Oliveira |
| Iuri de Jesus Martins |

# Prefácio

Os principais leitores são os Analistas, Desenvolvedores e Orientadores.

|  |
| --- |
| CONTROLE DE VERSÕES |
| Versão 01 – Criação deste documento e preenchimento do tópico 01 - Prefácio |
| Versão 02 – Preenchendo o tópico 02 - Introdução |
| Versão 03 – Desenvolvimento dos Diagramas |
|  |
|  |

|  |  |
| --- | --- |
| Nº Versões | Justificativa das versões |
| Versão 01 | Estamos relatando os possíveis leitores e criando este controle de versões para ter controle de todas as mudanças do projeto. |
| Versão 02 | Descrevemos a ideia e as funções desenvolvidas no projeto do CHAT. |
| Versão 03 | Desenvolvimento dos diagramas: Diagrama de Classes, Diagrama de Casos de Uso, Diagrama de Sequência e Diagrama de Atividades |
|  |  |
|  |  |

# Introdução

O projeto tem a ideia de ajuda os alunos (inicialmente da área de TI) que precisa de um serviço fácil, prático e rápido para se comunicar com os colegas dos cursos relacionados a TI. O acesso irá ser via GitHub para se identificar na plataforma, fazendo com que o projeto crie conexões entre os alunos e ajude no decorrer do curso.

O projeto será um sistema de chat em tempo real e de fácil acesso por meio do GitHub, com uma única sessão como se fosse um grupo onde a partir do acesso do aluno, ele será redirecionado a sessão de chat, que estará os outros alunos que estiverem logados na plataforma, para que os usuários possam se comunicar uns aos outros onde quer que estejam.

# Glossário

Define os termos técnicos usados no documento. Não se deve fazer suposições sobre a experiência ou o conhecimento do leitor.

# Definição de Requisitos de Usuário

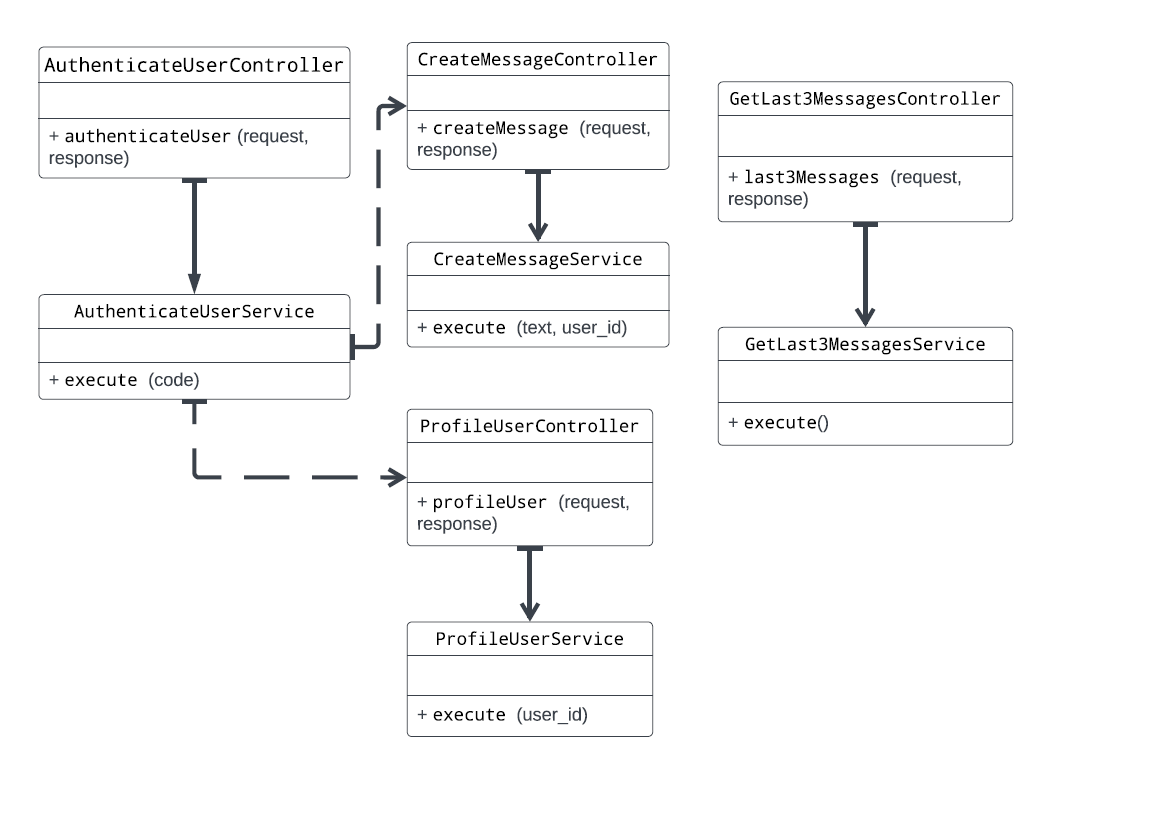
Descreve os serviços fornecidos ao usuário. Os requisitos não funcionais de sistema também devem ser descritos nessa seção. Essa descrição pode usar a linguagem natural, diagramas ou outras notações compreensíveis para os clientes. Normas de produto e processos que devem ser seguidos devem ser especificados.

# Arquitetura do Sistema

**DIAGRAMA DE CLASSES:**

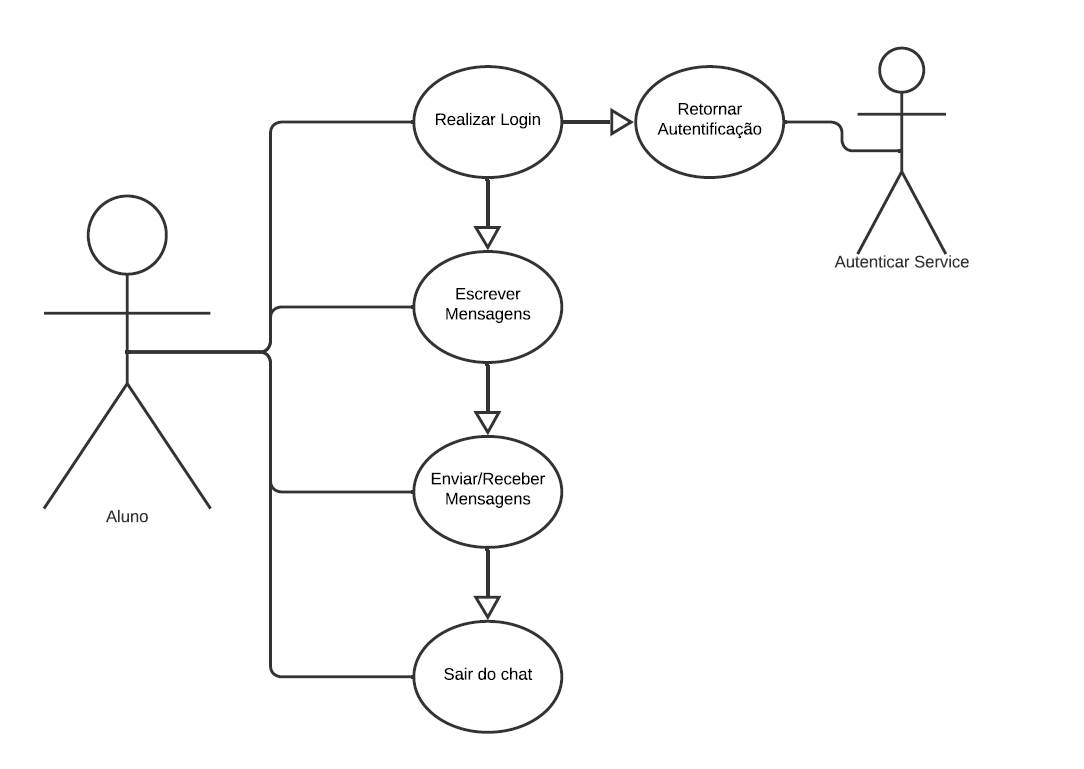
No diagrama de classes abaixo, temos as respectivas classes que serão criadas para desenvolver o projeto.

1. AuthenticateUserController tem o método público authenticateUser que vai receber uma requisição, pegar os dados e passar ao AuthenticateUserService.
2. AuthenticateUserService tem o método publico execute que vai receber os dados do AuthenticateUserController para fazer as validações e autenticações do usuário.
3. CreateMessageController tem o método público createMessage que vai receber os dados da mensagem e quem está enviando e passar para a classe CreateMessageService.
4. CreateMessageService tem o método público execute que irá tratar os dados recebidos do CreateMessageController e aplicar a lógica de criar a mensagem, enviar, validar e autenticar.
5. ProfileUserController tem o método público profileUser que recebe os dados do usuário e repassa para o ProfileUserService do formato que ele precisa.
6. ProfileUserService tem o método público execute que vai receber os dados formatados da maneira correta, verificar se ele é um usuário autenticado e retornar os dados parar apresentar na tela com por exemplo foto do usuário e nome.
7. GetLast3MessagesController tem o método público last3Messages está recebendo os valores numérico da requisição que é a quantidade de mensagens em ordem decrescente para repassar no GetLast3MessagesService.
8. GetLast3MessagesService tem o método público execute que acessa o banco e retorna as últimas mensagens informadas.



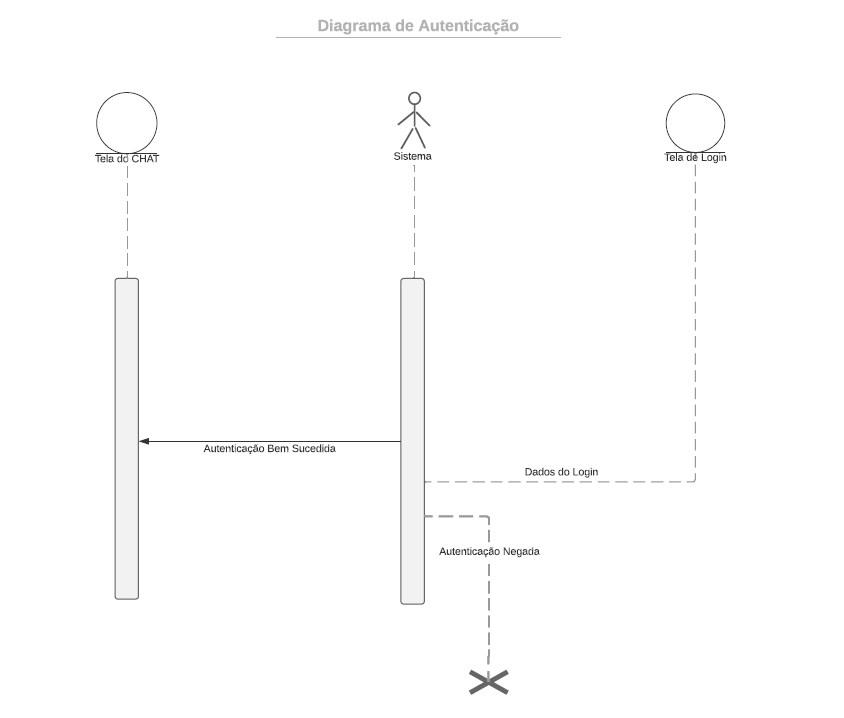
**DIAGRAMA DE CASOS DE USO:**

O usuário aluno ao acessar o chat deverá **realizar o login**, em seguida o sistema irá retornar à **autenticação do acesso** a conta, caso não seja autenticado o usuário não terá acesso ao chat, na plataforma ele poderá **escrever mensagens**. Após, ele terá que **enviar,** para que os outros usuários possam **receber**. E por fim depois que utilizar o sistema, ele irá se **desligar do chat**.

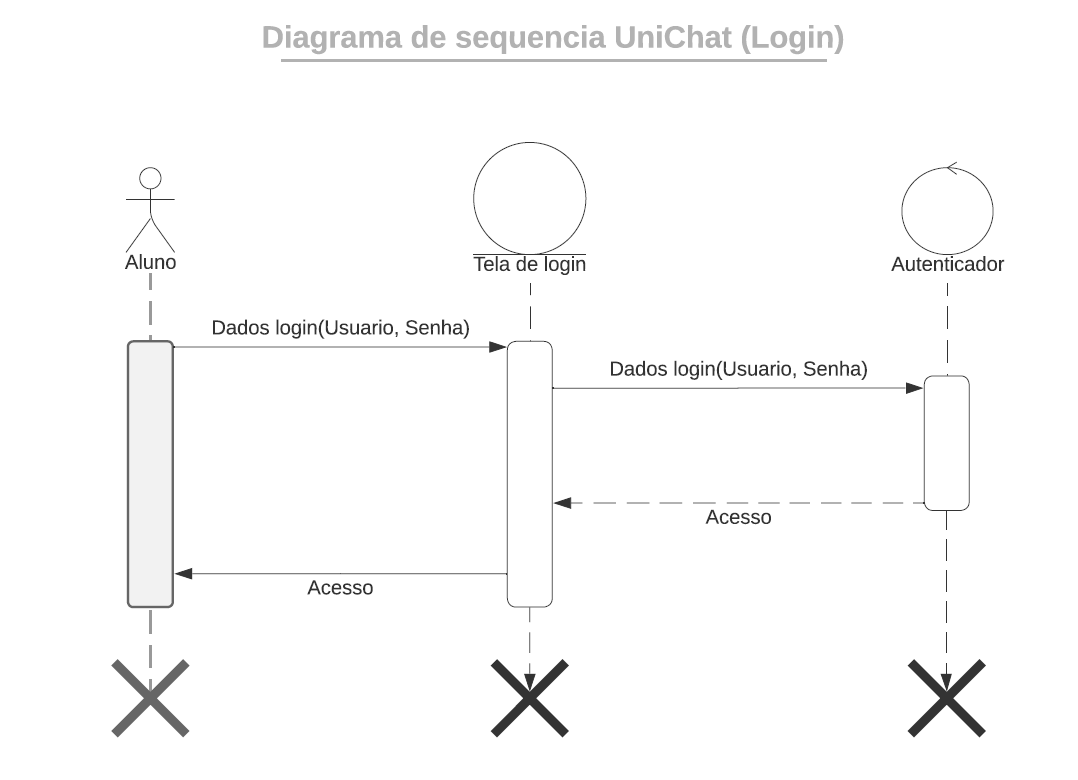
****

**DIAGRAMA DE SEQUÊNCIA:**

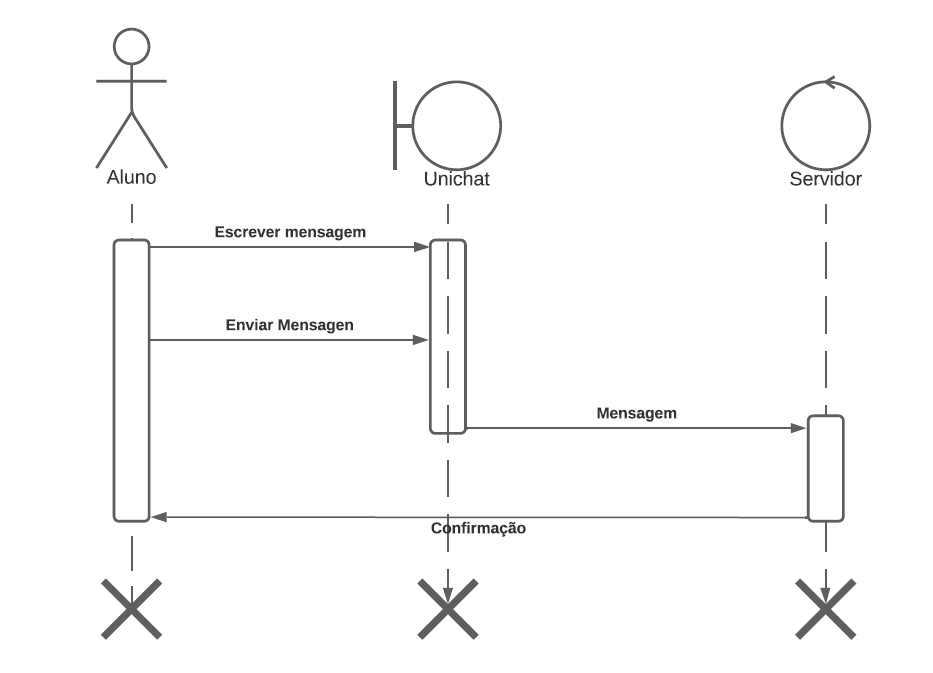
1. Nesse primeiro diagrama de sequência, ele relata a sequência a ser seguida na requisição de autenticar o usuário para acesso ao chat

****

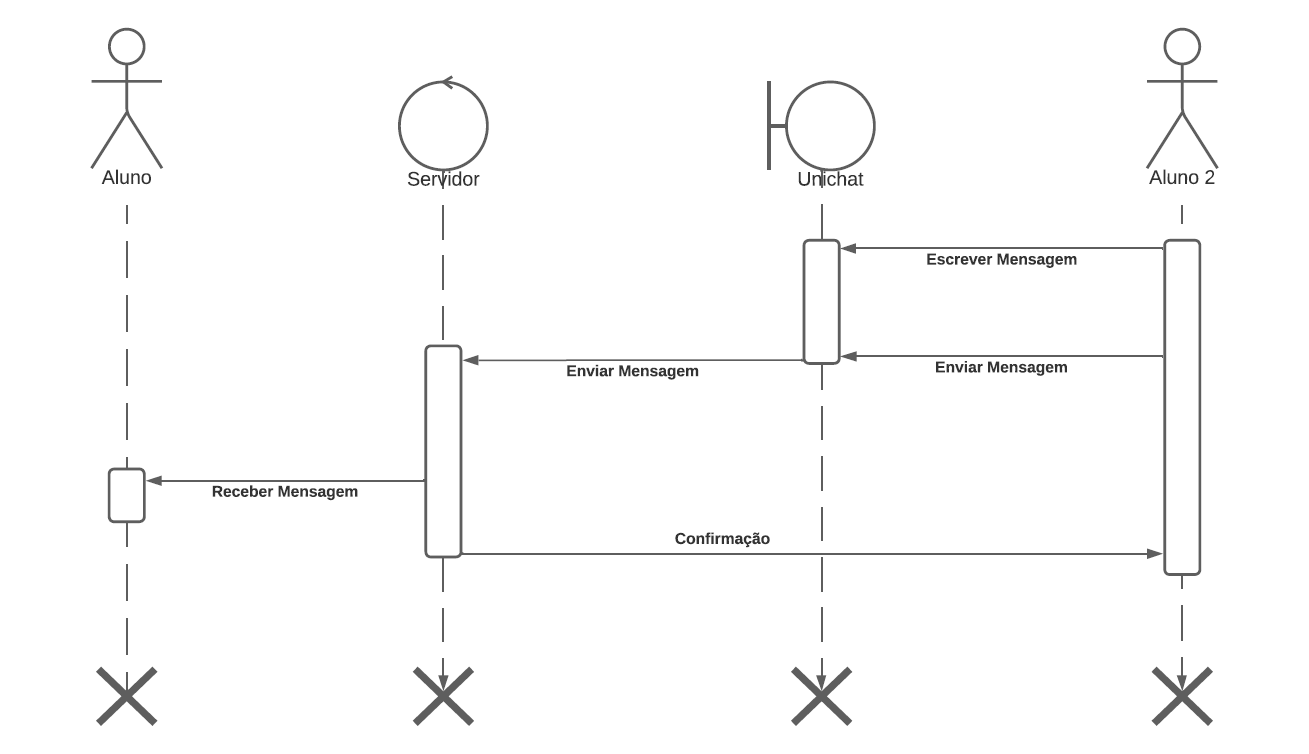
1. No segundo diagrama de sequência, ele apresentar o seguimento de requisição do Login para o usuário acessar o chat.

****

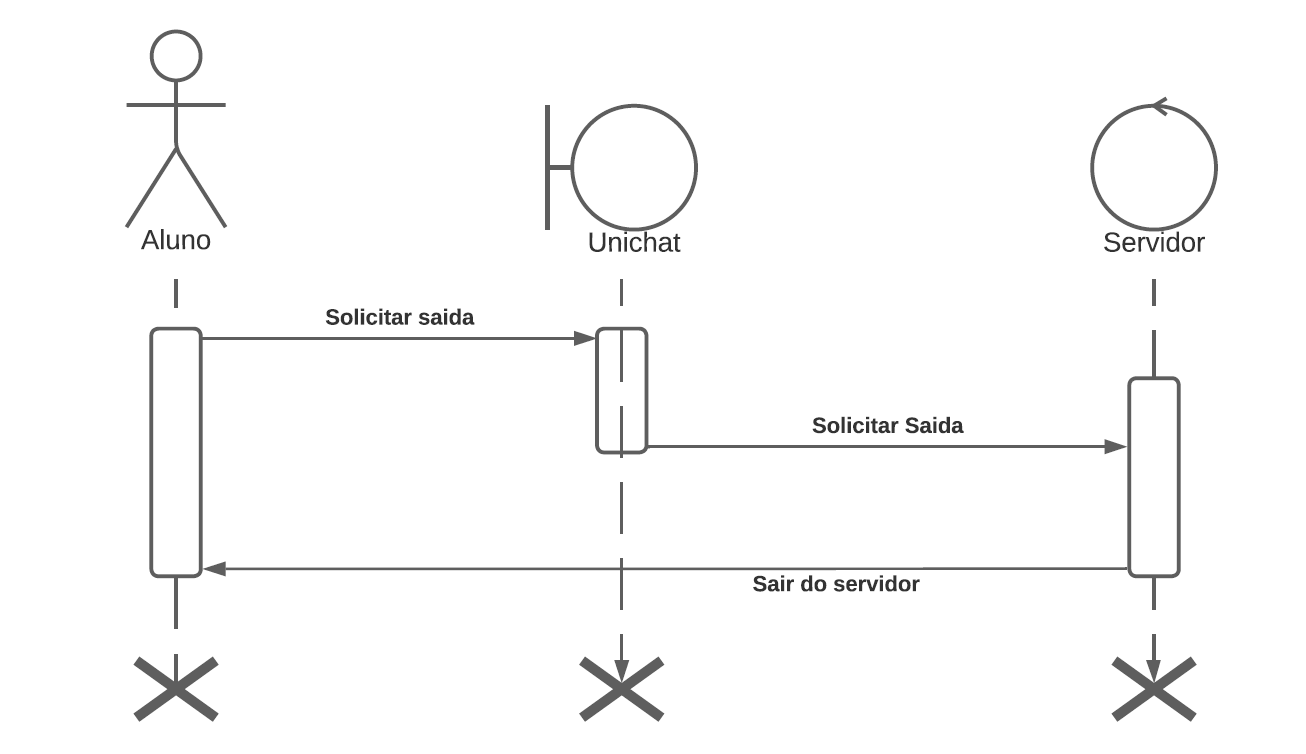
1. No terceiro diagrama de sequência, ele descreve o processo de envio de mensagens, que passa pelo servidor e retorna ao destinatário.

****

1. No quarto diagrama de sequência, ele representa o processo de retorno de mensagens, após passar pelo servidor e retorna ao destinatário.

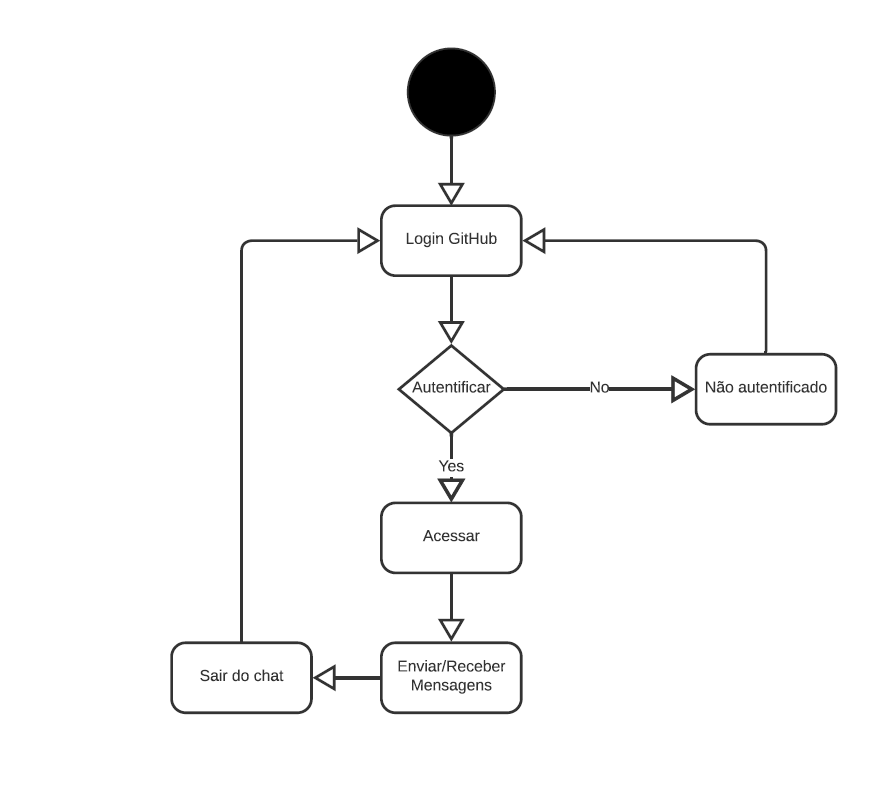
****

1. No quinto e último diagrama de sequência, ele demonstra o processo de saída do chat, que se inicia após o usuário solicitar.

****

**DIAGRAMA DE ATIVIDADES:**

Este diagrama mostra o processo de um sistema de chat. Como podemos ver no diagrama, o processo começa pela tela de Login, onde você terá que acessar com a sua Conta do GitHub. para ter um acesso mais fácil e rápido, em vez de preencher com dados. Após realizar o acesso, você já terá permissão para enviar e receber mensagens dos colegas.

****

# Especificação de requisitos do sistema

Descreve em detalhes os requisitos funcionais e não funcionais. Se necessário, também podem ser adicionados mais detalhes aos requisitos não funcionais. Interfaces com outros sistemas podem ser definidas.

# Modelos do Sistema

Modelos gráficos do sistema que mostram os relacionamentos entre os componentes do sistema, o sistema e seu ambiente. Possíveis modelos são: modelos de objetos, modelos de fluxo de dados ou modelos semânticos de dados.

# Evolução do Sistema

Descreve os pressupostos fundamentais em que o sistema se baseia, bem como quaisquer mudanças previstas, em decorrência da evolução de hardware, de mudanças nas necessidades do usuário etc. Essa seção é útil para projetistas de sistema, pois pode ajudá-los a evitar decisões capazes de restringir possíveis mudanças futuras no sistema.

# Apêndices

Fornece informações detalhadas e específicas relacionadas à aplicação em desenvolvimento, além de descrições de hardware e banco de dados, entre outros. Os requisitos de hardware definem as configurações mínimas e ideais para o sistema. Requisitos de banco de dados definem a organização lógica dos dados usados pelo sistema e os relacionamentos entre esses dados.

# Índice

Vários índices podem ser incluídos no documento. Pode haver, além de um índice alfabético normal, um índice de diagramas, de funções, de tabelas, entre outros pertinentes.