**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 |
| Versão 04 – Atualizações dos Tópicos 6, 9 e 10 |
| Versão 05 –Finalização do documento |

|  |  |
| --- | --- |
| 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 |
| Versão 04 | Incremento dos tópicos 6 Especificação de requisitos do sistema, 9 Apêndices e 10 Índices |
| Versão 05 | Finalizamos os tópicos 3, 4, 7 e 8. Assim concluímos o documento de requisitos |

# 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

* AuthenticateUser: Autenticar usuário;
* AuthenticateUserController: Controle de autenticação de usuário;
* AuthenticateUserService: Serviço de autenticação de usuário;
* BackEnd: Estrutura que possibilita a operação do sistema;
* Banco de Dados: É uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador;
* CreateMessage: Criar mensagem;
* CreateMessageController: Criar controlador de mensagens;
* CreateMessageService: Criar serviço de mensagem;
* Diagrama de classes: Os diagramas de classe são as cópias do sistema ou subsistema. Você pode utilizar os diagramas de classe para modelar os objetos que compõem o sistema, para exibir os relacionamentos entre os objetos e para descrever o que esses objetos fazem e os serviços que eles fornecem;
* Framework: É um termo inglês que, em sua tradução direta, significa estrutura. De maneira geral, essa estrutura é feita para resolver um problema específico
* FrontEnd: É onde se desenvolve a aplicação com a qual o usuário irá interagir diretamente, seja em softwares, sites, aplicativos, etc.;
* GetLast3MessagesController: Obter as últimas 3 mensagens do controlador;
* GitHub: É uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo
* Last3Messages: As ultima 5 mensagens
* ProfileUser: Perfil de usuário:
* ProfileUserController: Controle de perfil de usuário;
* ProfileUserService: Serviço de perfil de usuário;
* Requisitos Funcionais: São todos os problemas e necessidades que devem ser atendidos e resolvidos pelo software por meio de funções ou serviços

* Requisitos Não Funcionais: São os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas
* Servidor: É um software ou computador, com sistema de computação centralizada que fornece serviços a uma rede de computadores, chamada de cliente. Esses serviços podem ser de naturezas distintas, como por exemplo, arquivos e correio eletrônico.

# Definição de Requisitos de Usuário

**Requisitos de Usuário Funcionais:**

RF01 - Ter facilidade de acesso;

RF02 - Ter rapidez no envio e recebimento de mensagens;

RF03 - Ser seguro com as informações que passam pelo sistema;

RF04 - Ter o login efetuado via GitHub para facilitar as conexões entre alunos de T.I;

**Requisitos de Usuário Não Funcionais:**

RNF01 - Ter uma interface didática;

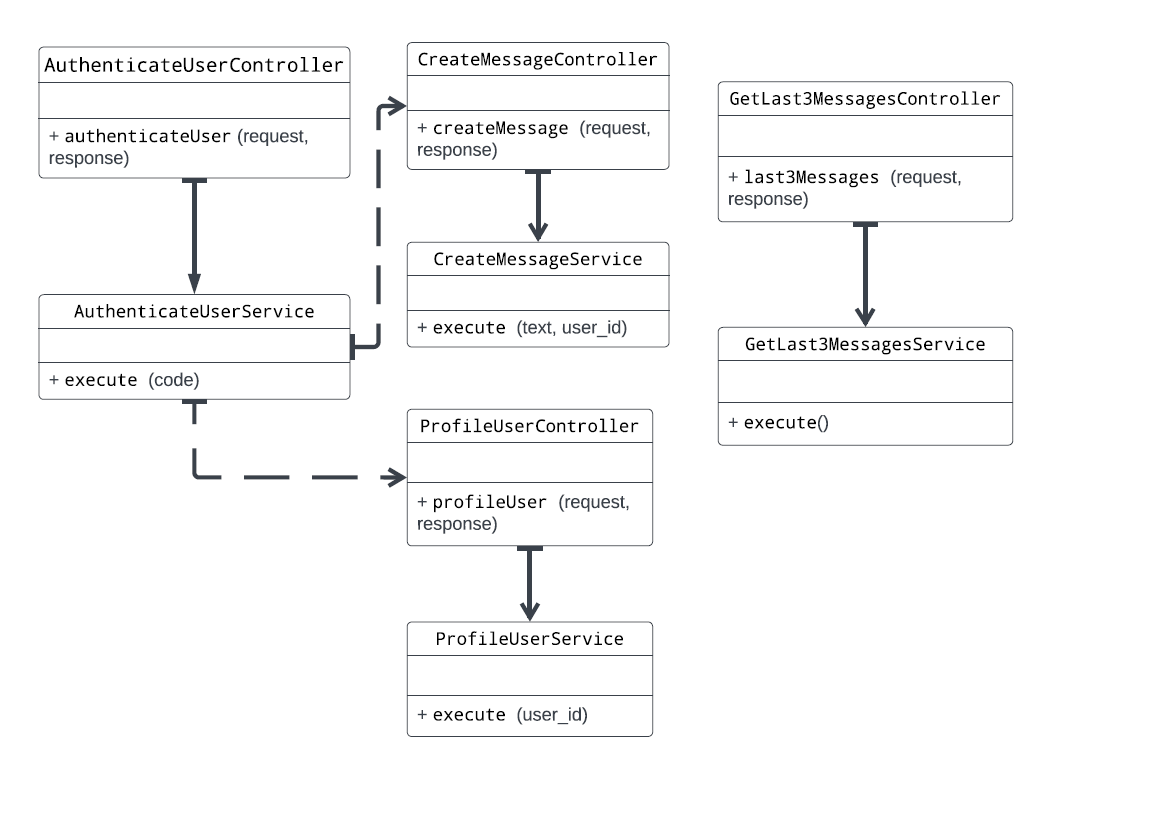
RNF02 - Ser intuitivo para quem usa;

# Arquitetura do Sistema

**DIAGRAMA DE CLASSES:**

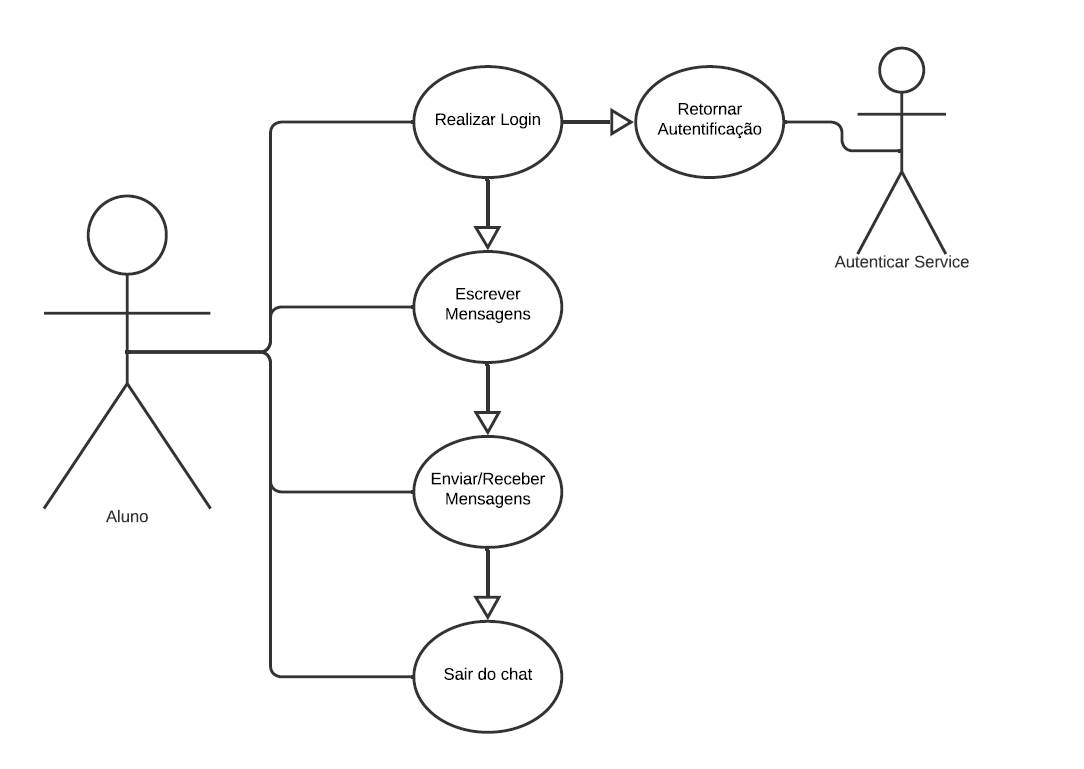
No diagrama de classes abaixo, temos as respectivas classes que serão criadas para desenvolver o projeto.

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.



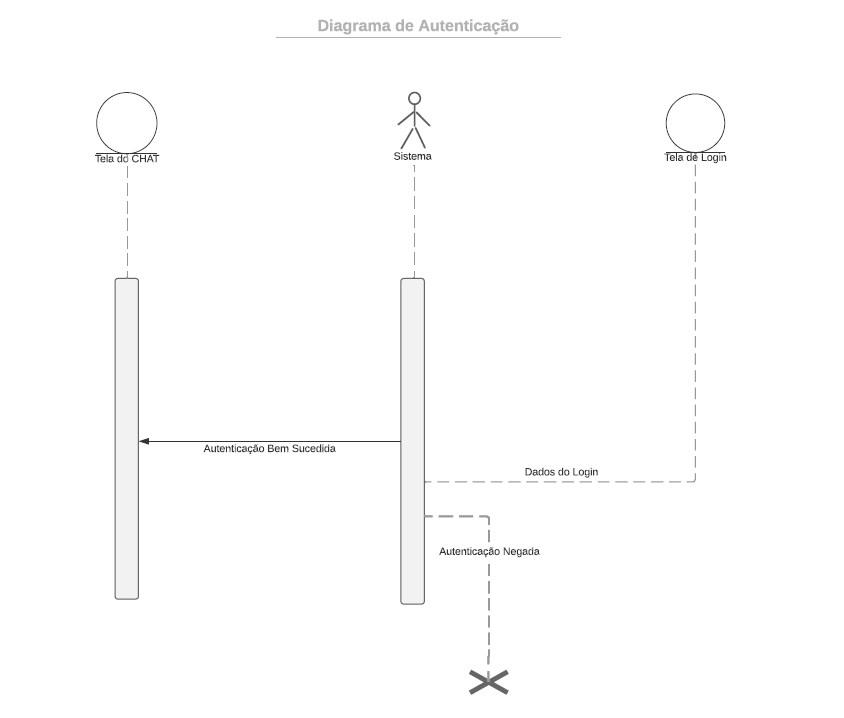
**DIAGRAMA DE CASOS DE USO:**

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**.

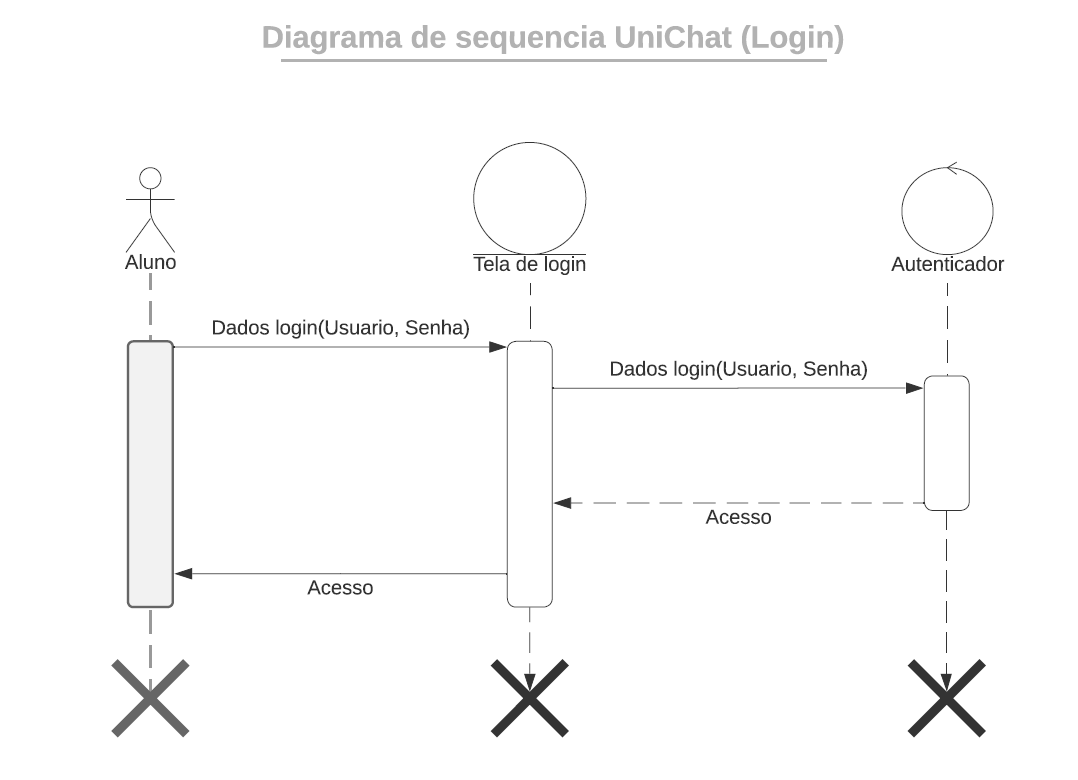
****

**DIAGRAMA DE SEQUÊNCIA:**

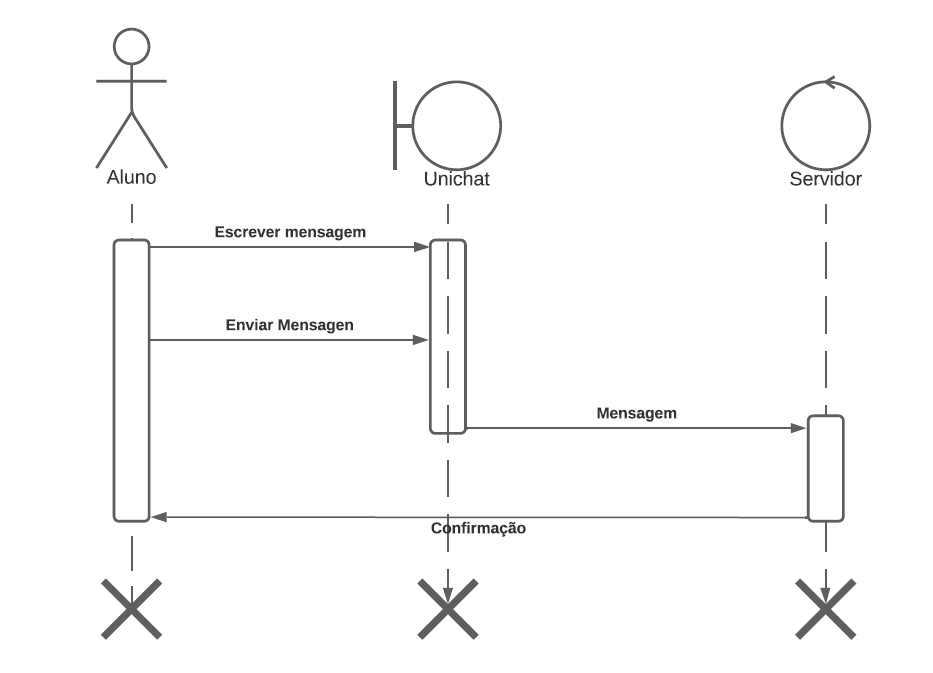
1. Nesse primeiro diagrama de sequência, ele relata a sequência a ser seguida na requisição de autenticar o usuário para acesso ao chat

****

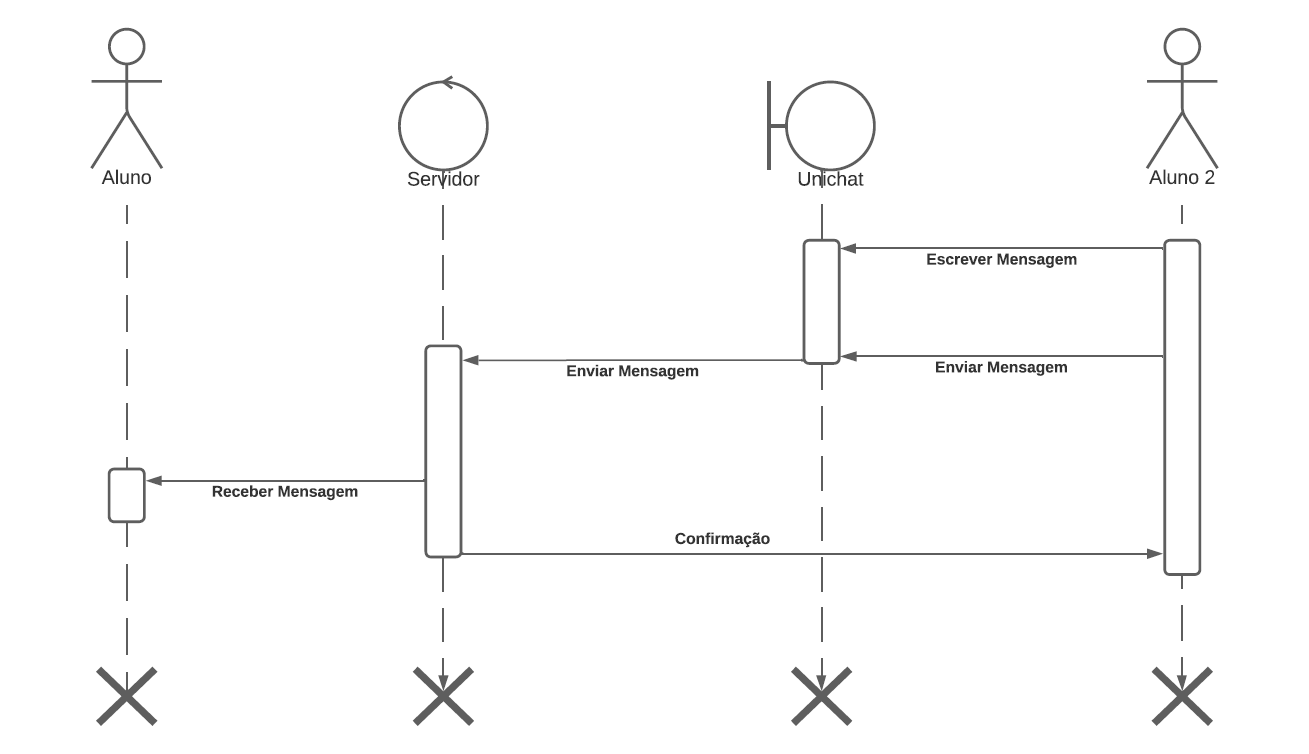
1. No segundo diagrama de sequência, ele apresentar o seguimento de requisição do Login para o usuário acessar o chat.

****

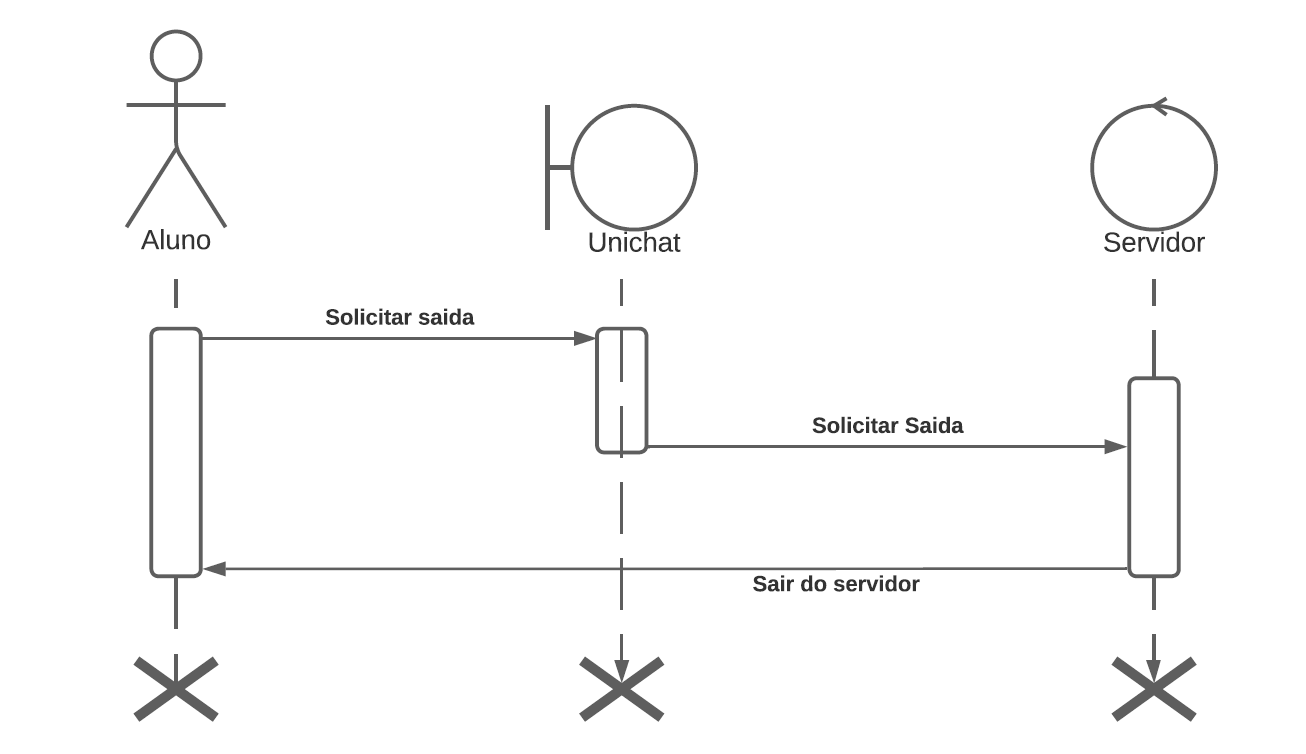
1. No terceiro diagrama de sequência, ele descreve o processo de envio de mensagens, que passa pelo servidor e retorna ao destinatário.

****

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

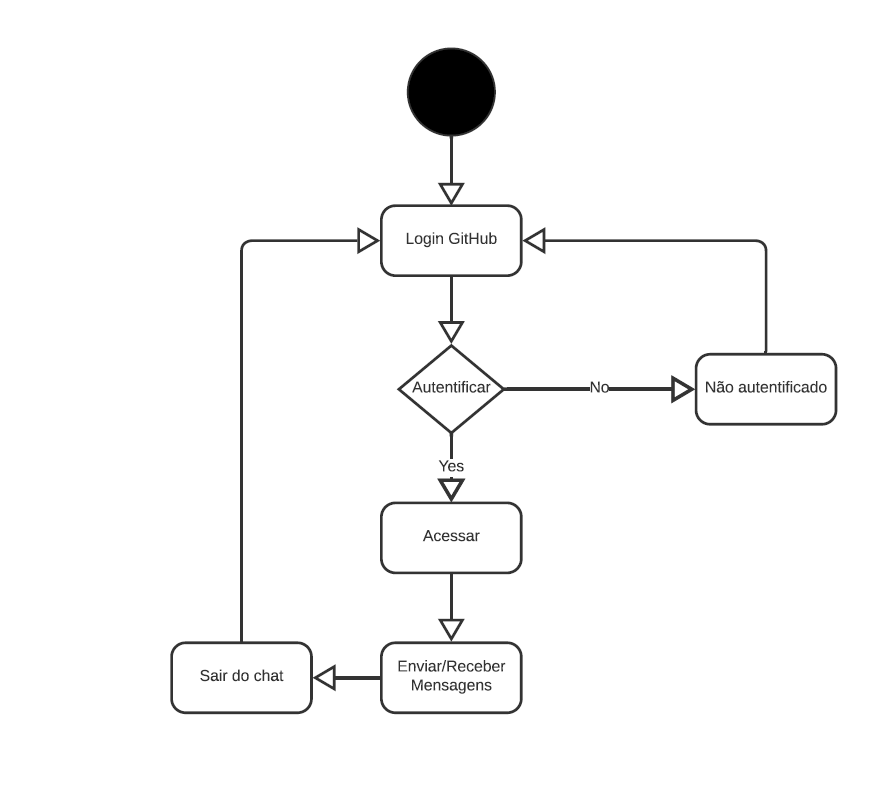
****

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

**Requisitos Funcionais:**

RF01-Realizar Login;

RF02-Realizar Logout do Chat;

RF03-Enviar mensagens;

RF04-Receber mensagens;

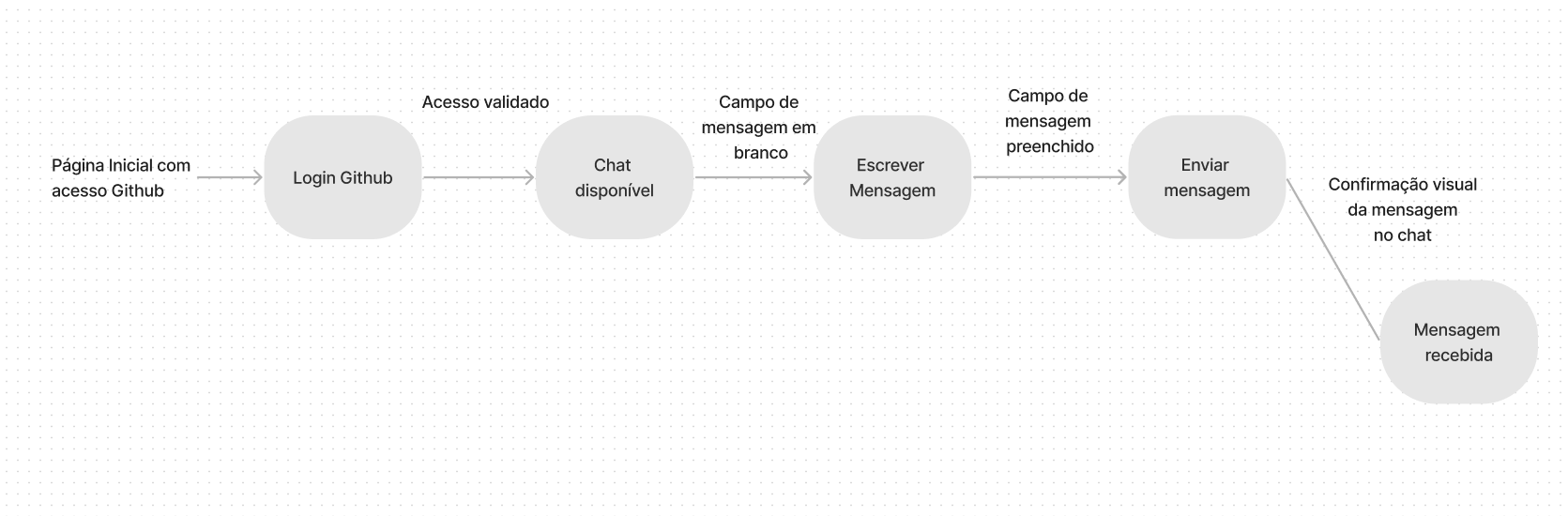
**Requisitos Não Funcionais:**

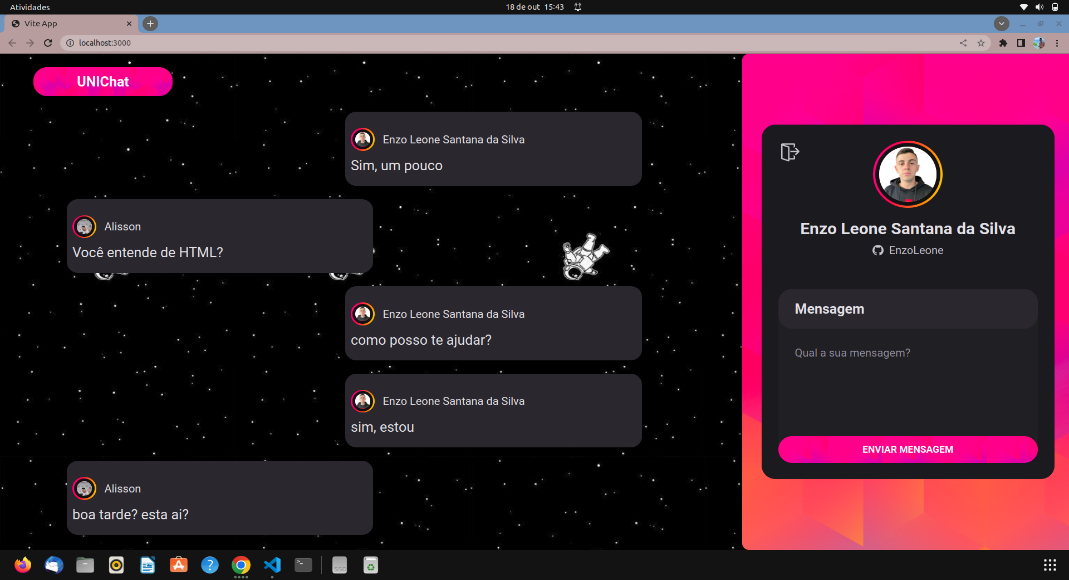
RNF01-Entrega instantânea de mensagens;

RFN02-Não onera armazenamento interno do computador;

RFN03-Acessível de qualquer lugar;

# Modelos do Sistema





# Evolução do Sistema

As principais evoluções que prevemos para esse projeto é:

* Adição de uma hud para separar cada uma das conversas;
* Um banco de dados mais robusto para que possamos aguentar um fluxo maior de transação de mensagens;
* Chamadas e vídeo conferências são uma das funcionalidades que estarão em versões mais futuras;
* Transferência de arquivos entre os alunos como materiais de estudo, fotos, vídeos entre muitos outros;
* Avisos de trabalhos/calendário na hud do app;

Além dessas ao decorrer do projeto serão adicionadas mais funções para a aplicação, entre outros planejamentos para o futuro.

# Apêndices

**O projeto será desenvolvido nas linguagens:**

* BackEnd: TypeScript com apoio da biblioteca SocketIO;
* FrontEnd: ReactJS;
* Banco de Dados: SQLite;
* Framework: ExpressJS;

**Requisitos de hardware:**

* Conexão com internet;
* Computador compatível com navegadores atuais e atualizados (Google Chrome, Microsoft Edge, Opera, etc...);

**Requisitos de banco de dados:**

* Não necessita de instalação ou configuração;
* Guarda o banco de dados em um único arquivo;
* É estável, multiplataforma e compatível com versões anteriores;

# Índice

1.Prefácio....................................................................................................................................02

2.Introdução................................................................................................................................02

3.Glossário...................................................................................................................................03

4. Definição de Requisitos de Usuário.........................................................................................04

5. Arquitetura do Sistema...........................................................................................................05

6. Especificação de requisitos do sistema...................................................................................10

7.Modelos do Sistema.................................................................................................................10

8.Evolução do Sistema.................................................................................................................11

9.Apêndices.................................................................................................................................11

10.Índices....................................................................................................................................12

Índice de Diagramas

1. Diagrama de Classes................................................................................................................05

2. Diagrama de Casos de Uso......................................................................................................06

3. Diagrama de Sequência (Autenticação)..................................................................................07

3.1. Diagrama de Sequência (Login)............................................................................................07

3.2. Diagrama de Sequência (Envio de mensagens) ...................................................................08

3.3. Diagrama de Sequência (Recebimento de mensagens) ......................................................08

3.4. Diagrama de Sequência (Logout).........................................................................................09

4. Diagrama de Atividades..........................................................................................................09