Lucas Toshio de Castro Uyekita - F07089-1

Gabriel Batista Matos - T96615-9

Lucas Bernardo da Silva - N51432-9

Kevin de Moura Cardoso - N5327F-8

Matheus de Souza Almeida - N51058-7

Lucas Oliveira Prado - F1171F-8

DESENVOLVIMENTO DE SISTEMA DE COMUNICAÇÃO EM MASSA VIA GRUPOS E REUNIÕES

**INTRODUÇÃO**

Este trabalho tem como a finalidade reunir vários usuários em um determinado grupo e os permitir que se reúnam e conversem entre si.

**Desenvolvimento de ideias**: Nossa ideia inicial foi criar um modelo em UML (Unified Modeling Language) e com isso teríamos uma base inicial do trabalho, em sequência delegar para cada um uma função principal, ou seja, programador Front-End, Back-End e Banco de dados.

**Desenvolvimento do Modelo**: Decidido o modelo e dados que seriam necessários para a aplicação, o início da criação do molde e decisão das linguagens que seriam utilizadas em sua composição, que em questão são vue.js (Front-End), PHP (Back-End) e como banco de dados foi utilizado MySql.

**Desenvolvimento do projeto**: Uma vez decidida a linguagem que cada um usaria, foi dado início a produção do software em questão, que se utiliza de uma estrutura que permite armazenamento de conversas, usuários e reuniões.

**A divisão da aplicação é dada pelos seguintes módulos:**

**Autonomia do Usuário:** O usuário tem algumas funções dentro do software que permite com que ele crie reuniões, bate papo entre outros usuários, também o gerenciamento de cara reunião criada pelo próprio usuário.

**Reunião:** podendo ser criada para dois ou mais usuários, que proporciona uma conversa tendo uma previsão para seu início, onde cada contato do usuário que for chamado será marcado em uma lista que será criada, e na data prevista será criada uma “sala” particular para a conversa do grupo.

**Conversas:** o elemento principal que faz a ligação entre as mensagens que os usuários enviam um para o outro.

Figura 1

Diagrama, Esquemático

Descrição gerada automaticamente

Fonte: Criação própria

As conversas serão enviadas do usuário para o Back-End, que validará o dado e o direciona para mensagem de dois lugares, um vai ser o banco de dados e o outro será para o usuário alvo da mensagem.

Figura 2

Diagrama

Descrição gerada automaticamente

Fonte: Criação própria

**Modelagem Banco de Dados:** Criamos uma modelagem de banco de dados mais simples que, porém, no decorrer da criação percebemos que eram necessários mais campos funções.

Diagrama

Descrição gerada automaticamenteFigura 3

Fonte: Criação própria

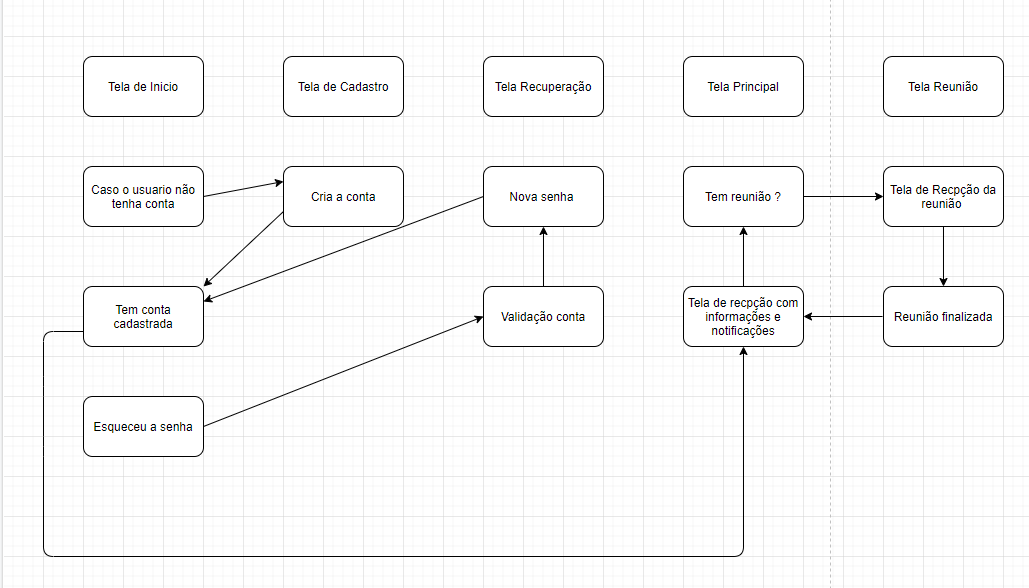
No plano de desenvolvimento do sistema backend está sendo desenvolvidos e composto pelo Gabriel Batista e Lucas Toshio, utilizando visual Studio como editor e o PHP como linguagem principal de programação e trabalhando no mesmo sempre dias úteis na semana.

No desenvolvimento da mamografias e parte escrita está sendo desenvolvida e também composta apenas por Lucas Oliveira com padrão de norma ABNT seguindo sempre os critérios estabelecido pelo mesmo, editando sempre aos fins de semana.

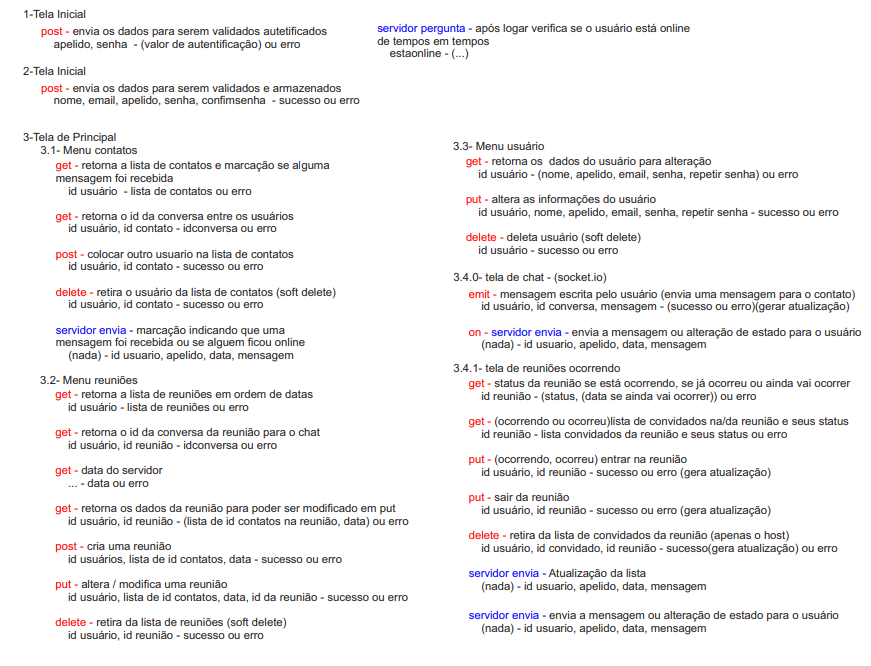
No desenvolvimento da sistema de frontend está sendo desenvolvidos e composto por Lucas Bernardo utilizando VScode (vue.js) como framework principal de programação trabalhando em dias úteis da semana.

No desenvolvimento do sistema e modelagem do banco de dados está sendo desenvolvido pelo aluno \*\*\*\*\*\*\* my SQL workbench para gerenciamento de dados e My SQL como linguagem principal de programação sendo desenvolvido juntamente com o backend, sendo assim tendo aplicações técnicas e interações rápidas em ambos os desenvolvimentos tendo os dias úteis semanais para tais desenvolvimentos e interações entre os alunos.

**ESTRUTURA DO USUÁRIO AO PROGRAMA**



**Mapa de requisições e inserções**



**FUNDAMENTOS DA COMUNICAÇÃO EM REDE**

Banco de Dados

Nossa escolha para banco de dados foi MySql, pela sua facilidade e disponibilidade de tecnologias, o MySql utiliza protocolo TCP sendo assim mais seguro para os dados. A porta de conexão escolhida é padrão 3306.

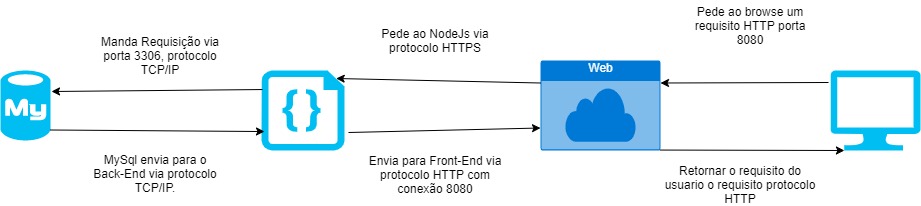
Front-End

Utilizando-se de uma framework voltado para componentes, o Front-End foi desenvolvido totalmente e vue.js que faz uma união organizada entre CSS, HTML e JavaScript, permitindo assim a utilização de estruturas pré estabelecidas, que por sua vez permitem a criação de diversas janelas e temas. A conexão entre o Front-End e o Back-End se estabelece via Axios (REST) permitindo assim a conversação entre essas duas etapas do programa.

Back-End

O projeto por ser em Js optamos para o NodeJs e facilitar a transferência de dados do banco de dados para o Front-End sem precisar fazer qualquer conversão, sendo assim o Front-End pede ao Back-End via protocolo HTTP na porta 8080 uma requisição do banco de dados, o back-end se conecta ao banco via protocolo TCP com a conexão pela porta 3306 o banco de dados retorna para o Back-End ele trata o dados e envia para o Front-End o pelo protocolo HTTP na porta 8080.

Figura 1



Fonte: Proprietária via Draw.IO.

**CONSIDERAÇÕES FINAIS**

A respeito das técnicas escolhidas, pode-se observar não apenas variações de tempo, como de filosofia a qual foi criada a idea de como realizar o processo de ordenação de valores, durante os processos de pesquisa sobre as ordenações, as vezes houveram diferenças discrepantes entre alguns resustados, mesmo que estivessem sendo realizados em um mesmo ambiente de testes, com exatamente os mesmos valores.

Os testes demonstraram que existem vantagens em utilizar certos tipos de ordenações para a realização de diferentes cenários, um exemplo é como o bubble sort se comporta ao ordenar um array cem por cento desordenado, e como pode ser previsto seu resultado foi inferior ao insertion sort e ao quick sort, o que em um array com níveis maiores de ordenação pode-se constatar que a diferença se tornou menor porem ainda expressiva devido a ideia que foi utilizada na criação desse tipo de ordenação que apesar de simples de realizar e compreender, torna-se lenta em comparação com as demais. O insertion sort pode-se dizer que alem da facilidade de entendimento é extremamente funcional em arrays já com altos níveis de ordenação onde os valores por mais que estejam desordenados estejam próximos aos valores que se aproximam um do outro, tornam do o rápido. Porem apesar do insertion sort ser rápido, não pode superar o quick sort que foi mais rápido em todos os cenários estudados, o que mostrou que apesar de ser complexo de se compreender a filosofia por traz de sua criação se mostrou extremamente funcional.

Durante os testes para que pudessem ser realizados, o grupo optou por utilizar a linguagem python a qual a maior parte do grupo já está familiarizado, e tem uma ferramenta que ajudou em uma melhor obtenção de dados que no caso é o numba que permitiu uma obtenção de performance na ordenação dos resultados

-----------------------------------------------------------------------------------------------------------------