**UNIVERSIDADE CIDADE DE SÃO PAULO**

DERIEL VITOR LOPES – 26748843

FELIPE SOUZA SALLES – 26711176

GIOVANI SÃO B. MASCARENHAS – 26651921

MATHEUS RODRIGUES MARTINS - 26080176

RODRIGO CLARO RIEKMANN LAISE - 26640805

VINÍCIUS MORAES MINOWA – 25952226

**Documento de Requisitos de Sistema**

**PROJETO GOOD TALKS**

Orientador: Prof º Rodolfo Goya

SÃO PAULO

2022

# Prefácio

Além dos nossos clientes, que são para quem o nosso projeto está sendo desenvolvido, também definimos como possíveis leitores os Desenvolvedores e os Arquitetos de "softwares", pois, esses profissionais estarão responsáveis por algumas partes do desenvolvimento do programa. Para mais, investidores que possam se tornar possíveis clientes ou sócios, também são considerados prováveis leitores.

# Introdução

O nosso programa será desenvolvido em Java Script, com o auxílio da biblioteca React JS. A principal função do nosso sistema é a troca de mensagens entre usuários, para utilizar o programa os usuários poderão realizar o ‘login’ com sua conta de e-mail.

# Glossário

**CSS** - Linguagem de estilo que define a aparência de documento web HTML. Exemplo: manipulação da posição dos elementos, cores, fontes, fundos, etc.

**GitHub** - O GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git.

**Google Authenticator** - O Google Authenticator gera códigos da verificação em duas etapas.

**Java Script** - Linguagem que viabiliza a criação de páginas interativas, ou seja, que são modificáveis de acordo com a interação de quem usa e que conectam o Front-End ao Back-End.

**React JS** - Biblioteca JavaScript para criar o Front-End de aplicações.

**Servidor Web** - Solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes (é um programa web de Back-End, um tipo de API que precisa de uma rede para se comunicar).

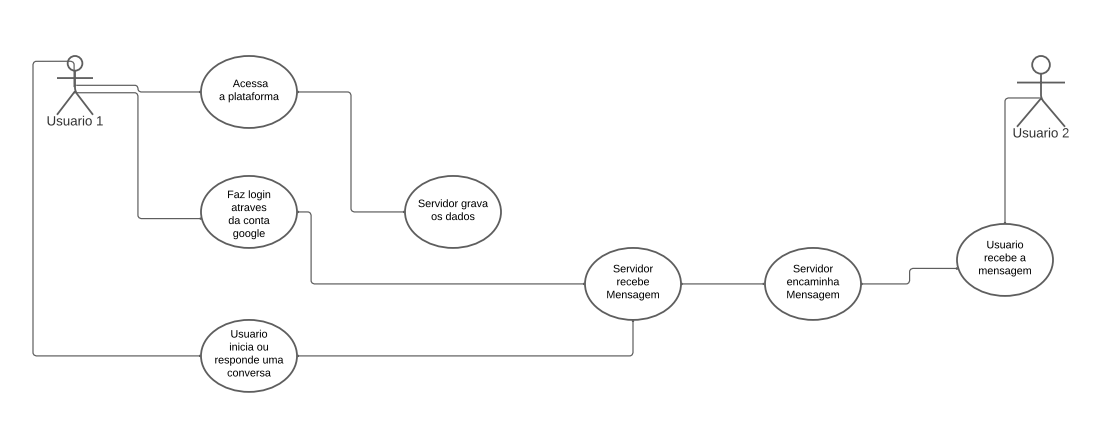
# Definição de Requisitos de Usuário

O sistema deve ser desenvolvido em ‘Java Script’, com o auxílio da biblioteca ‘React JS’, além também de permitir ser acessado através do navegador.

No nosso sistema os clientes se comunicarão através de um bate-papo, o Usuário 1 fará ‘login’ através da conta google, quando ele enviar alguma mensagem para o Usuário 2 o servidor receberá essa mensagem e encaminhará para o segundo usuário.

O servidor receberá tanto as mensagens que iniciaram a conversa, quanto as que foram enviadas como resposta do Usuário 2.

Vale ressaltar que o servidor gravará os dados dos clientes após eles acessarem a plataforma. Logo abaixo temos um Diagrama de Caso de Uso para melhor entendimento.

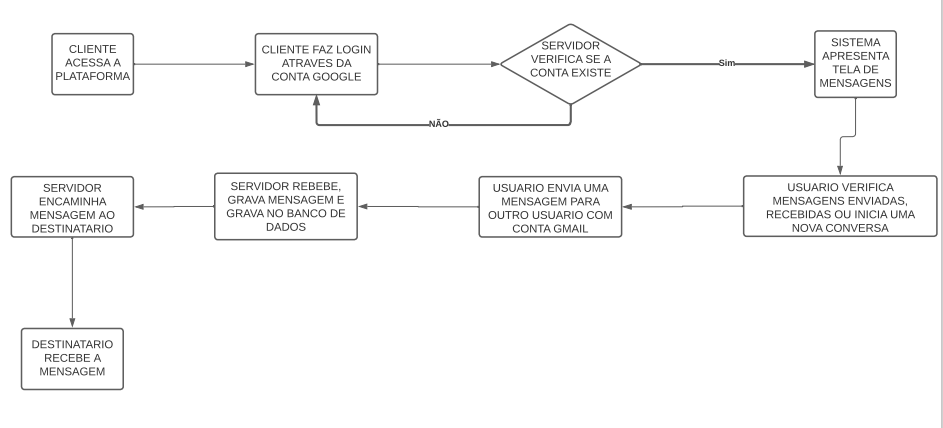


# Arquitetura do Sistema

O sistema deve trabalhar da seguinte forma: no 1° momento o usuário “A” deve acessar a página inicial do ‘software’ onde será possível fazer o ‘login’ através de uma conta Gmail existente, após realizar o ‘login’ ele visualizará a tela do mensageiro, na qual caso o indivíduo já seja um usuário da plataforma, constara os registro de conversas dele com outros usuários, caso contrário, será possível iniciar uma nova conversa desde que o indivíduo “B” possua uma conta Gmail existente, também será possível mandar mensagens de texto e ‘emojis’.

A aplicação será desenvolvida utilizando em sua maior parte os componentes ‘Firebase’, ‘React, JavaScript’, CSS, HTML. Logo abaixo temos um Diagrama de Atividades para melhor entendimento.

**Diagrama de Atividades**

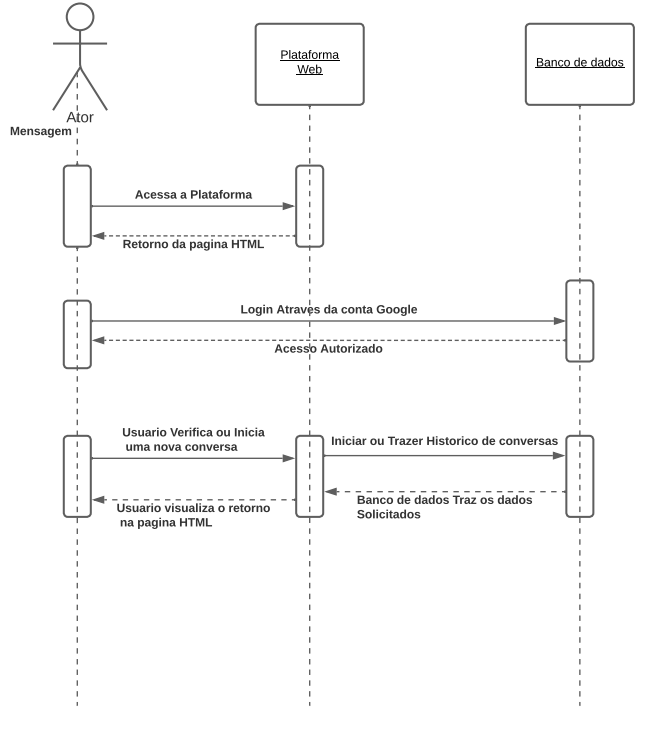


# Especificação de requisitos do sistema

O sistema deve permitir que os usuários se comuniquem através de mensagens de texto.

O software funcionará da seguinte forma: o Cliente acessa a plataforma e faz o seu ‘login’, após isso o servidor verifica se o ‘login’ de usuário existe, caso não exista, a plataforma volta para tela de login. Realizando um login existente, o sistema apresentará a tela de mensagens, na qual o usuário verifica as mensagens enviadas e as quais recebeu, podendo também iniciar uma nova conversa. Caso o cliente deseje iniciar uma nova conversa, ele deve enviar a mensagem para um outro usuário que também tenha conta Gmail, após envia-la o servidor recebe esse texto e grava no banco de dados, depois ele encaminha a mensagem para o destinatário, logo em seguida, ocorrendo tudo certo, o destinatário recebe a mensagem. Logo abaixo temos um Diagrama de Sequência para melhor entendimento.

**Diagrama de Sequência**



# 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.