PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA POLITÉCNICA E DE ARTES

GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

  
  
  
  
  
  
  
  
  
  
AED - 1ª parte

 Aplicação Com Comunicação Cliente-Servidor

Sistema de Chat em Tempo Real com Laravel WebSockets

 FERNANDO CARLOS BRANDÃO FILHO

FERNANDO COSTA OLIVEIRA

HIAGO LAURENÇO DONHA

MANOELLA CRISTYNA BORGES MATIAS

GOIÂNIA,

2023

FERNANDO CARLOS BRANDÃO FILHO

FERNANDO COSTA OLIVEIRA

HIAGO LAURENÇO DONHA

MANOELLA CRISTYNA BORGES MATIAS

 AED - 1ª PARTE

 APLICAÇÃO COM COMUNICAÇÃO CLIENTE-SERVIDOR

SISTEMA DE CHAT EM TEMPO REAL COM LARAVEL WEBSOCKETS

Atividade Extra Disciplinar apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção de nota N1 do período 2023/2 a matéria de Sistemas Distribuídos (CMP1190), turma C01.

GOIÂNIA,

2023

**Introdução**

Este documento apresenta uma visão abrangente do nosso projeto de software, uma aplicação de chat em tempo real construída utilizando as tecnologias Vue.js e PHP com o framework Laravel. A aplicação oferece aos usuários a capacidade de se comunicar instantaneamente através de salas de chat, com suporte ao protocolo WebSocket para troca eficiente de mensagens em tempo real.

Nossa escolha de tecnologias inclui o Vue.js como framework front-end e o Laravel como plataforma de back-end. O Vue.js simplifica o desenvolvimento de interfaces de usuário interativas e responsivas, enquanto o Laravel é conhecido por sua robustez e facilidade de uso, permitindo-nos desenvolver a lógica do servidor de forma eficiente e confiável.

Este documento abordará todos os aspectos essenciais do projeto, desde a arquitetura e implementação até os testes e instruções de uso. Embora nossa aplicação seja desenvolvida para um contexto específico, a compreensão dos conceitos aqui apresentados tem potencial para ser aplicada em projetos de desenvolvimento web do mundo real que envolvem recursos em tempo real.

1. **O que a aplicação irá fazer**

Apresentamos nossa aplicação de chat em tempo real, uma plataforma de comunicação que oferece uma experiência de interação instantânea e colaborativa. Desenvolvida com as tecnologias Vue.js e PHP com o framework Laravel, esta aplicação possibilita que os usuários participem de conversas dinâmicas em salas de chat virtuais.

Os principais recursos desta aplicação incluem a capacidade de criar e ingressar em salas de chat temáticas, trocar mensagens em tempo real por meio do protocolo WebSocket, receber notificações em tempo real quando novas mensagens são enviadas, e a integração de mídia que permite compartilhar imagens, vídeos e outros tipos de conteúdo nas conversas. Além disso, os usuários podem criar contas personalizadas, editar seus perfis e interagir com outros participantes das conversas. Moderadores ou administradores têm a capacidade de gerenciar as salas de chat para manter um ambiente saudável.

Os benefícios gerais desta aplicação incluem uma comunicação eficaz e em tempo real, adequada para uma ampla variedade de casos de uso, desde discussões em grupo até colaborações de trabalho. A organização simplificada das conversas em salas temáticas torna mais fácil a localização de informações relevantes. Com uma interface intuitiva, a aplicação é acessível e amigável, mesmo para usuários iniciantes, estimulando a colaboração e o compartilhamento eficaz de informações.

Nossa aplicação de chat em tempo real é uma ferramenta versátil e poderosa que facilita a comunicação e a colaboração, tornando-se uma valiosa adição para diversos cenários de uso. Continuaremos a explorar detalhes da arquitetura do sistema para uma compreensão completa do projeto.

1. **Papel do cliente e do servidor**

O projeto faz uso do modelo cliente-servidor para criar a estrutura do chat. Nesse modelo, o cliente não compartilha nenhum recurso com o servidor, mas utiliza alguma função do servidor, sendo o cliente que inicia a comunicação enquanto o servidor aguarda a requisição.

Para o projeto o servidor servirá como unidade centralizadora de todas as conexões recebidas via socket e terá como responsabilidade o envio de uma mensagem (recebida de um cliente) para todos os demais conectados no servidor. Quando um cliente se conecta ao servidor, esta cria uma Thread para aquele cliente, ou seja, cada conexão terá sua respectiva Thread e o servidor fará a gestão dessas conexões.

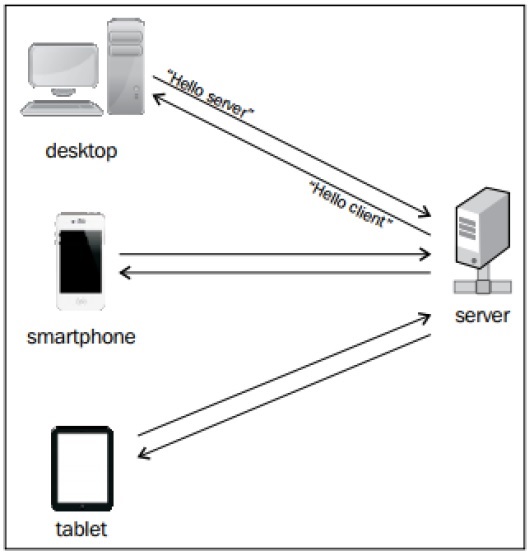
Cada usuário criará uma instância do cliente e fará uma conexão com o servidor socket. O cliente deverá informar o endereço do server socket e a respectiva porta.

1. **Qual o tipo de comunicação será utilizada entre o cliente e o servidor**

WebSocket é uma tecnologia avançada para criar uma ligação entre um cliente e um servidor (browser e servidor) e permitir a comunicação entre eles em tempo real. A principal diferença com o WebSocket é que permite receber dados sem ter de enviar um pedido separado, como, por exemplo, acontece no HTTP. Uma vez estabelecida a ligação, os dados virão por si mesmos sem necessidade de enviar o pedido. É a vantagem de utilizar o protocolo WebSocket em chats ou relatórios de stock, onde precisa de receber informação constantemente atualizada. O protocolo pode receber e enviar informação simultaneamente, permitindo uma comunicação bidireccional full-duplex, o que resulta numa troca de informação mais rápida.

A ligação entre cliente e servidor permanece aberta até ser terminada por uma das partes ou encerrada por um timeout. Eles executam um aperto de mão para estabelecer uma ligação entre o cliente e o servidor. Uma ligação estabelecida permanece aberta, e a comunicação é efectuada utilizando o mesmo canal até que a ligação seja terminada no lado do cliente ou do servidor. As mensagens são trocadas bidirecionalmente. O WebSocket permite encriptar os dados que são transmitidos. Para tal, é utilizado um suplemento sobre o protocolo WSS, que codifica os dados do lado do remetente e os decodifica do lado do destinatário. Para quaisquer intermediários, a informação permanece encriptada. Sem encriptação, os dados tornam-se um alvo de ameaças.

Quando devo utilizá-lo, websocket é negociável em casos como plataformas de troca como por exemplo: aplicações de jogos, chatbots, notificação de *“push*”, redes sociais, aplicação de *IOT (internet of things).* Considerá-lo é fundamental para fornecer atualizações em tempo real. É suportado por quase todos os navegadores, Chrome, Mozilla, Firefox. Um único servidor pode ter várias conexões ao mesmo cliente permitindo escalar a aplicação.



1. **Qual a linguagem para implementação**

* PHP: é uma linguagem interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no lado do servidor, capazes de gerar conteúdo dinâmico na World Wide Web;
* Laravel: é um framework PHP livre e open-source criado por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC;
* Javascript: é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.
* Vue: é um framework JavaScript de código-aberto, focado no desenvolvimento de interfaces de usuário e aplicativos de página única;
* Vuex: é uma biblioteca de gerenciamento de estado de código aberto para aplicativos Vue.js. Ele foi criado por Evan You, o desenvolvedor por trás do Vue.js, para fornecer um armazenamento centralizado para gerenciamento de estado em aplicativos Vue.js;

1. **Requisitos e especificações da aplicação**

**Requisitos Funcionais:**

1. Salas de Chat
   1. Os usuários devem conversar com outros usuários
2. Troca de Mensagens em Tempo Real
   1. Os usuários devem poder enviar e receber mensagens em tempo real nas salas de chat.
   2. As mensagens devem ser entregues instantaneamente.
3. Histórico de Conversa
   1. Os usuários podem acessar o histórico de conversa de uma sala de chat, incluindo mensagens antigas

**Requisitos Não Funcionais:**

1. WebSocket
   1. A aplicação deve utilizar o protocolo WebSocket para permitir a troca de mensagens em tempo real.
2. Segurança
   1. Deve ser implementada segurança adequada para proteger os dados dos usuários e evitar ataques.
3. Escalabilidade
   1. A aplicação deve ser capaz de lidar com um número significativo de usuários e mensagens simultâneas.
4. Desempenho
   1. A troca de mensagens em tempo real deve ser eficiente e responsiva.

1. **Descrição estrutural e comportamental da aplicação**

Descrição Estrutural:

* Participantes: São os usuários envolvidos na conversa, identificados por nomes de usuário, nomes reais ou outros identificadores.
* Mensagens em bolhas: As mensagens são exibidas em bolhas separadas para cada pessoa, com a cor ou identificação do remetente.
* Timestamps: Cada mensagem pode incluir a hora e a data em que foi enviada, permitindo acompanhar o histórico da conversa.
* Organização temporal: Ordem cronológica das mensagens para entender a sequência da conversa.
* Ícones de perfil: Pequenas imagens que representam os participantes da conversa.
* Identificação de remetente: A maneira pela qual a conversa identifica quem enviou cada mensagem, sendo os ícones de perfil.
* Caixa de texto para digitar mensagens: Um espaço onde os participantes podem inserir suas mensagens para envio.
* Emojis e adesivos: A capacidade de incluir emojis e adesivos para expressar emoções ou sentimentos.

Descrição Comportamental:

* Escolha do nome do chat;
* Conexão do chat em tempo real;
* Envio de mensagem: mínimo de 400 caracteres;
* Leitura das mensagens enviadas em tempo real;
* Histórico das mensagens;
* Exibição da lista de usuários;
* Desconectar do chat;

1. **Planejamento do mapa de telas**

.

Esse protótipo é a representação digital de uma conversa entre duas pessoas em tempo real, que nesse caso está sendo utilizado através de um aplicativo de mensagem.

O protótipo desse chat foi pensado na experiência do usuário, e a facilidade de uso é um critério fundamental, garantindo que os participantes possam se comunicar de forma intuitiva. Esse protótipo foi realizado na ferramenta Paint.

