



UNIVERSIDADE DO MINHO
DEPARTAMENTO DE ENGENHARIA E RECURSOS DO MAR

CURSO DE LICENCIATURA EM ENGENHARIA DE INFORMÁTICA E SISTEMAS COMPUTACIONAIS

RELATÓRIO DO TF - ICHAT
ANO LETIVO 2024/2025 - 3º ANO
Disciplina: Sistemas Distribuídos

Docente: Frederico Soares

Discente: Jorge Fortes

Mindelo, 2024

1. Introdução

O iChat representa um sistema de comunicação em tempo real desenvolvido no âmbito da disciplina de Sistemas Distribuídos, projetado para facilitar a troca de mensagens e arquivos entre múltiplos utilizadores através de uma interface web intuitiva. Este projeto incorpora os princípios fundamentais dos sistemas distribuídos, incluindo comunicação assíncrona, gestão de estados partilhados e tolerância a falhas, enquanto oferece uma experiência de utilizador moderna e responsiva.

Num contexto académico, o iChat serve como um estudo de caso prático para explorar desafios comuns em aplicações distribuídas, tais como sincronização de mensagens entre clientes heterogéneos, transmissão eficiente de ficheiros com garantia de integridade, notificações em tempo real de eventos de rede (conexão/desconexão).

A solução adota uma arquitetura cliente-servidor com comunicação bidirecional via WebSocket, complementada por APIs REST para operações assíncronas (e.g., upload de ficheiros). A escolha de tecnologias como Node.js (backend) e JavaScript vanilla (frontend) permitiu priorizar a simplicidade e portabilidade, sem comprometer funcionalidades essenciais.

Este relatório detalha os componentes do sistema, decisões de design, fluxos de trabalho críticos e lições aprendidas durante o desenvolvimento.

2. Objetivos

O desenvolvimento do iChat teve como propósito principal criar uma aplicação distribuída funcional que implementasse os conceitos teóricos estudados em sala de aula. Especificamente, buscou-se atingir três metas fundamentais: estabelecer comunicação bidirecional em tempo real através do protocolo WebSocket, implementar um sistema confiável de transferência de arquivos com feedback imediato aos usuários, e garantir a escalabilidade básica para suportar múltiplas conexões simultâneas. O projeto também visou demonstrar na prática os desafios de sincronização em sistemas distribuídos, como a manutenção de estado consistente entre todos os clientes conectados.

3. Tecnologias Utilizadas

A implementação do iChat adotou um conjunto de tecnologias modernas e amplamente utilizadas no mercado. Para o backend, optou-se por Node.js com o framework Express, combinado com a biblioteca ws para implementação do protocolo WebSocket, proporcionando alta eficiência no tratamento de conexões simultâneas. O processamento de arquivos foi resolvido com o middleware Multer, que gerencia eficientemente uploads no sistema de arquivos do servidor. No frontend, manteve-se uma abordagem minimalista com JavaScript vanilla para demonstrar a integração direta com APIs web modernas, sem a complexidade de frameworks adicionais. O sistema completo é containerizável e pode ser implantado em qualquer ambiente compatível com Node.js versão 16 ou superior.

4. Funcionalidades Implementadas

O sistema iChat oferece um conjunto abrangente de funcionalidades de comunicação. A troca de mensagens em tempo real ocorre através de conexões WebSocket persistentes, com mecanismos de reconexão automática em caso de falhas na rede. Para compartilhamento de arquivos, implementou-se um sistema completo de upload com feedback visual, limitando o tamanho máximo dos arquivos a 10MB para garantir a responsividade do sistema. Notificações inteligentes informam os usuários sobre eventos relevantes, como a entrada ou saída de participantes da sala de chat. A interface do usuário foi cuidadosamente projetada para oferecer experiência intuitiva, com histórico de conversas persistente durante a sessão e indicadores visuais de status de conexão.

5. Arquitetura do Sistema

O sistema é composto por:

- Servidor (server.js): recebe conexões dos clientes, gere os eventos de mensagens e de upload de ficheiros.
- Cliente (client/index.html, script.js, style.css): interface do utilizador que permite enviar mensagens e ficheiros.
- Pasta uploads/: armazena temporariamente os ficheiros enviados.

A comunicação é realizada via WebSockets (através do Socket.io), permitindo que todos os utilizadores conectados recebam mensagens e ficheiros em tempo real.

6. Execução

Para executar o iChat no teu próprio computador ou rede local:

- i. Certificar de que tens o Node.js instalado.
- ii. Abrir um terminal na pasta do projeto e executa:
 - `npm install`
 - `npm start`
- iii. Depois disso, basta abrir o navegador e aceder a:
 - `http://localhost:3000`

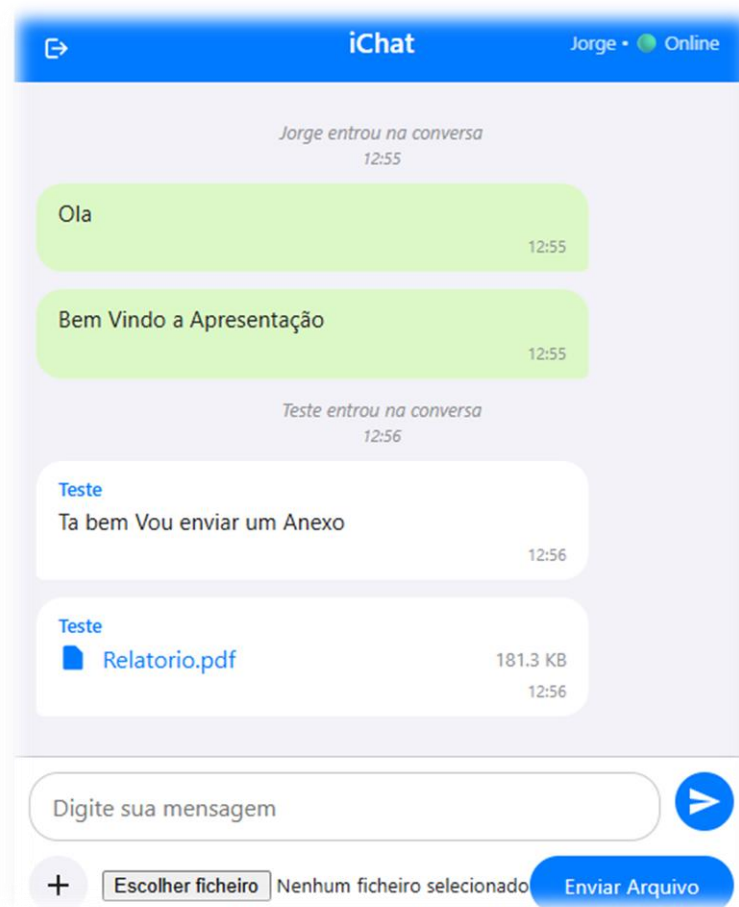
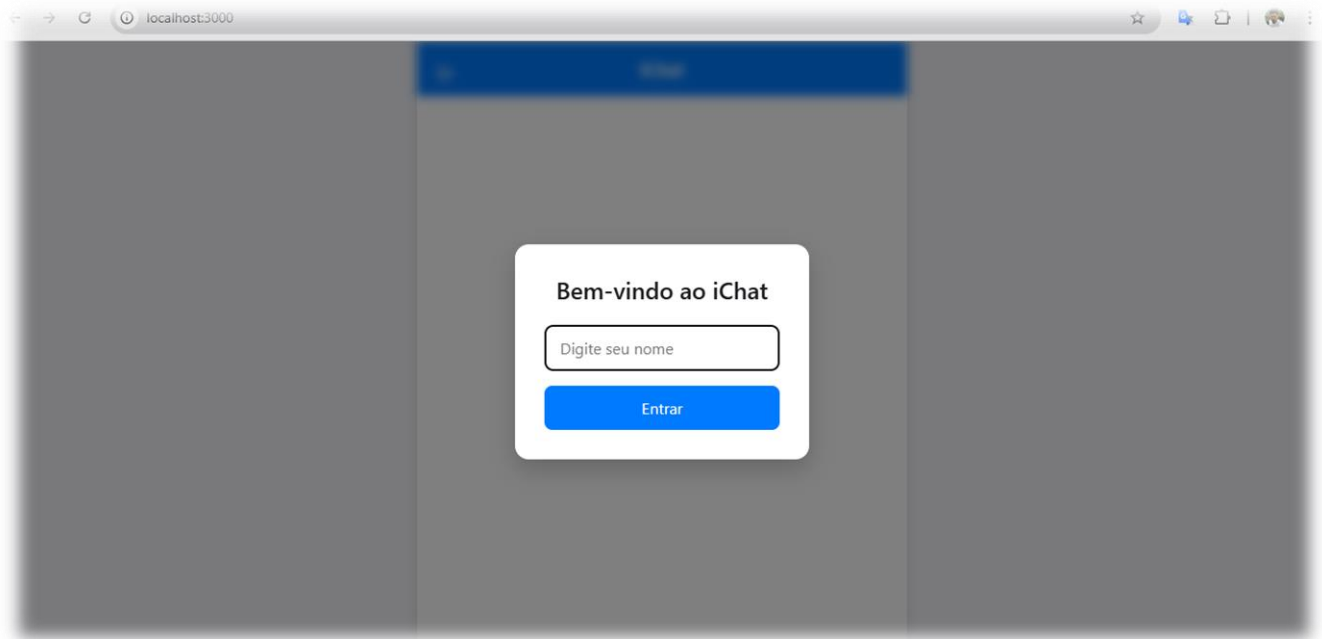
A aplicação estará funcional e pronta para uso. Pode abrir em vários dispositivos na mesma rede para testar o funcionamento em tempo real.

Online (versão pública):

O iChat encontra-se também disponível numa versão online pública, alojada na plataforma Render:

<https://ichat-oqv3.onrender.com/>

7. Anexos



8. Considerações Finais

Este projeto permitiu aplicar na prática os conhecimentos adquiridos na área de sistemas distribuídos, particularmente no que diz respeito à comunicação entre múltiplos nós clientes, gestão de eventos assíncronos, e transferência de ficheiros pela rede.

O iChat, apesar da sua simplicidade, representa uma base sólida para projetos mais avançados, como plataformas de atendimento em tempo real, salas de aula virtuais, ou aplicações colaborativas. A interface clara e a fluidez na comunicação demonstram o potencial das tecnologias utilizadas.

Ter disponibilizado o sistema numa plataforma de cloud como a Render foi um passo adicional para reforçar o entendimento de ambientes de produção e distribuição de software online, algo essencial para o mundo real da engenharia informática.

O projeto iChat está online e disponível para testes, sendo um exemplo prático e funcional de um sistema distribuído de troca de mensagens e ficheiros com interface gráfica.