

# Grupo Sarapatel - Chat

Alunos: Flavimar, Giancarlo, Luca e Lucas Souza.



# Proposta

O trabalho visa desenvolver um sistema que traga consigo um chat, permitindo que duas pessoas distintas possam conversar entre si (comunicação entre pares), além de existir a possibilidade de conversar com várias outras pessoas ao mesmo tempo (grupo/sala).



# Requisitos funcionais

- Cadastro de usuário por email;
- Conversa usuário 1x1;
- Conversa em grupo/sala



# Requisitos não funcionais

- O sistema será hospedado em nuvem (AWS), utilizando serviços para hospedagens de imagens docker;
- Será gerada uma imagem docker da aplicação, possibilitando o uso de containers;
- O sistema utilizará filas (serviço AWS SQS) para o envio de Email para cadastro de usuário;
- O sistema utilizará serverless (serviço lambda) para o envio de email para envio de cadastro de usuário;
- Haverá uma aplicação web ASP.NET MVC, frontend e backend na mesma solução, com Javascript, HTML e CSS no front;



- Haverá uma aplicação serverless responsável por receber o evento de cadastro e enviar o email;
- Será utilizado comunicação com o frontend por meio de websockets, para que as imagens sejam refletidas em tempo real;
- Será utilizado o banco de dados Postgres, por meio do serviço RDS da AWS

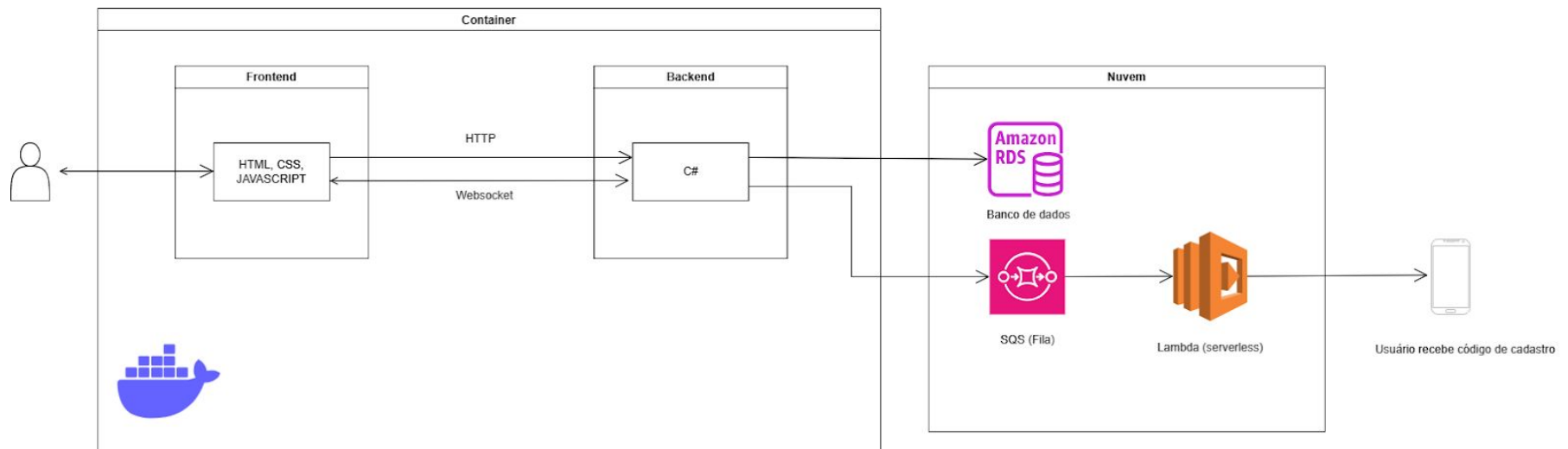


# Relações com a disciplina:

- Utilização de containers (Docker);
- Aplicação distribuída;
- Aplicação baseada em eventos;
- Uso de conceitos serverless;
- Uso de filas;
- Uso de websockets;
- Uso de semáforos, controles de threads e tasks;
- A aplicação tem uma natureza distribuída e concorrente;
- Backend será desenvolvido de acordo com os padrões API RestFUL;
- Deverão ser aplicadas lógicas inerentes à concorrência e distribuição;



# Arquitetura



# Cronograma

