

MIDDLEWARE ESCALÁVEL E RESILIENTE PARA COMUNICAÇÃO EM TEMPO REAL DE APLICAÇÕES WEB MÓVEIS EM AMBIENTES DE COMPUTAÇÃO DISTRIBUÍDA

Aluno(a): Ariel Adonai Souza

Orientador: Prof. Me. Francisco
Adell Péricas

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Requisitos
- Especificação
- Implementação
- Análise de resultados
- Conclusões

Introdução

- Aumento do uso de Smartphones
- Bridging Frameworks
- Escalabilidade sistemas distribuídos
- Sistemas em grandes empresas
- Simplificação comunicação

Objetivo Geral

- Apresentar um Middleware escalável e resiliente para comunicação cliente/servidor em tempo real entre aplicações web para dispositivos móveis e sistemas em um ambiente de computação distribuída

Objetivos Específicos

- Avaliar a escalabilidade do software usando AWS
- Avaliar a resiliência na entrega de mensagens
- Selecionar um protocolo padrão para comunicação de aplicações web
- Atender a um contexto de comunicação entre aplicações web e ambiente distribuído.

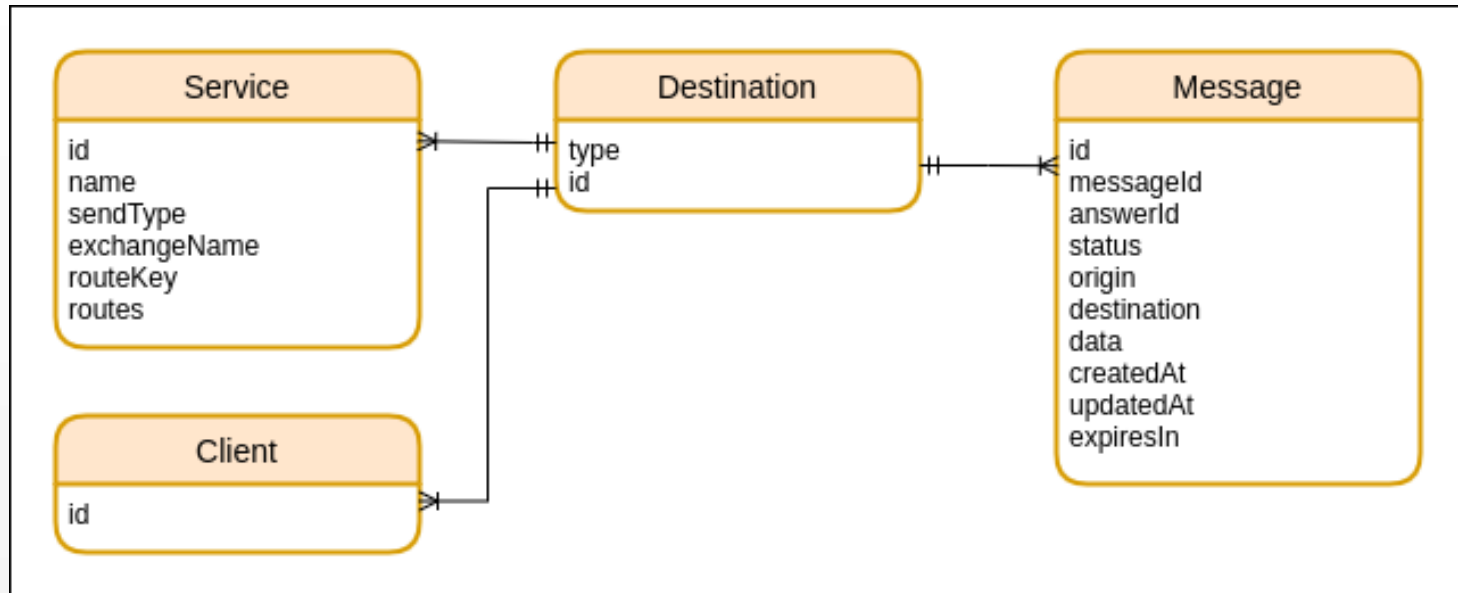
Fundamentação Teórica

- REST
- WebSocket
- Sistemas Distribuídos
- Middleware
- Backend For Frontend

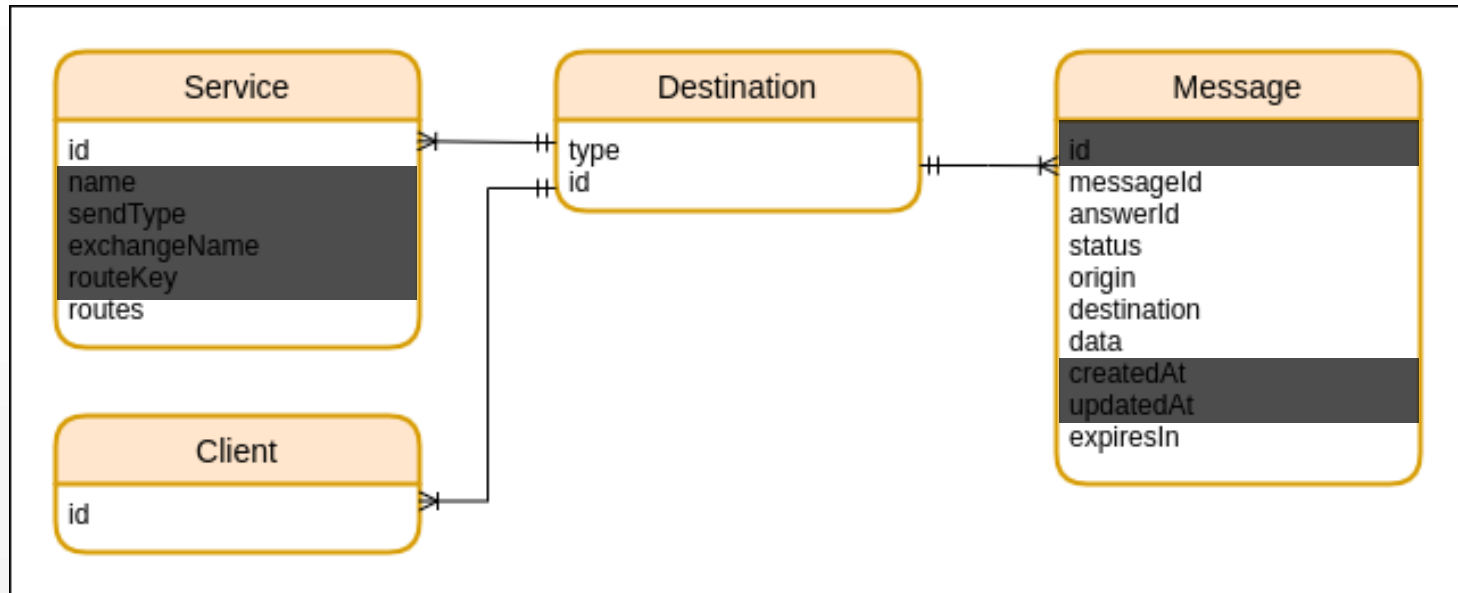
Requisitos

- Permitir a troca de mensagens entre a aplicação web e sistemas em um ambiente distribuído
- Entrega das mensagens precisa ser resiliente
- Cliente deve receber as mensagens pendentes armazenadas no middleware logo após a conexão
- Mensagens devem contar com um tempo de expiração
- Comunicação entre cliente e middleware será através do protocolo WebSocket
- Comunicação entre serviços e middleware será através de filas/mensageria
- Deve ser capaz de escalar horizontalmente

Especificação



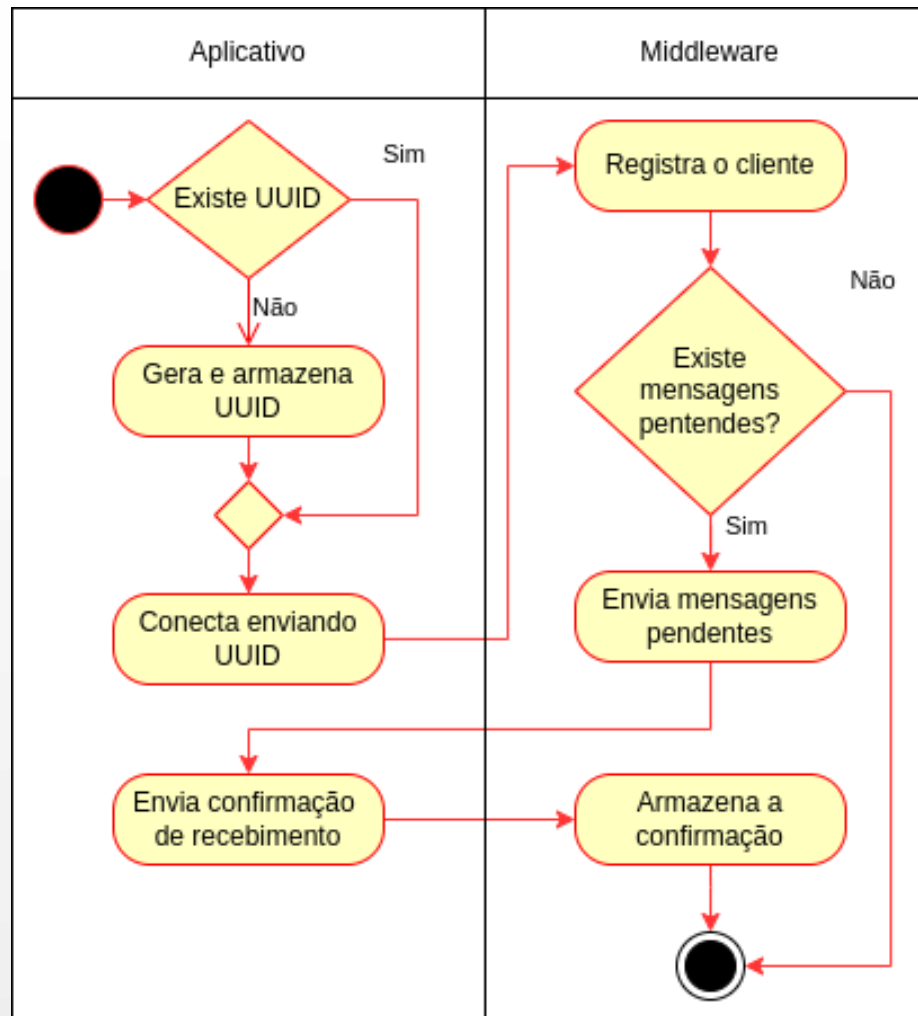
Especificação



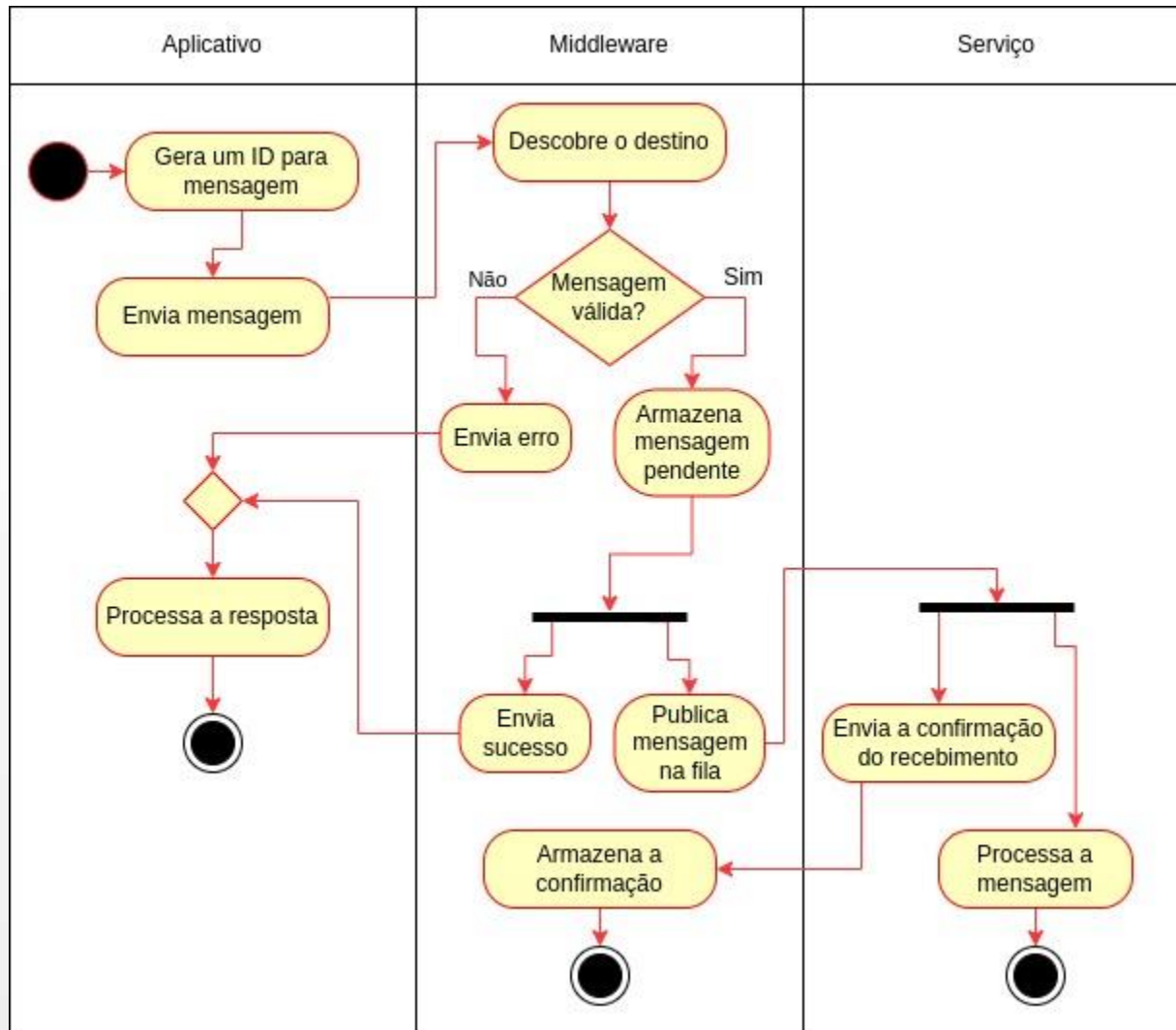
Especificação

```
01 {  
02     "id": "818e875f-2cdb-401d-ac81-6860ad588bb5",  
03     "route": "chat",  
04     "ttl": 1652494335000,  
05     "data": {  
06         "to": "fulano20",  
07         "message": "Olá. Tudo bem com vc?",  
08         "date": 1652494335000  
09     }  
10 }
```

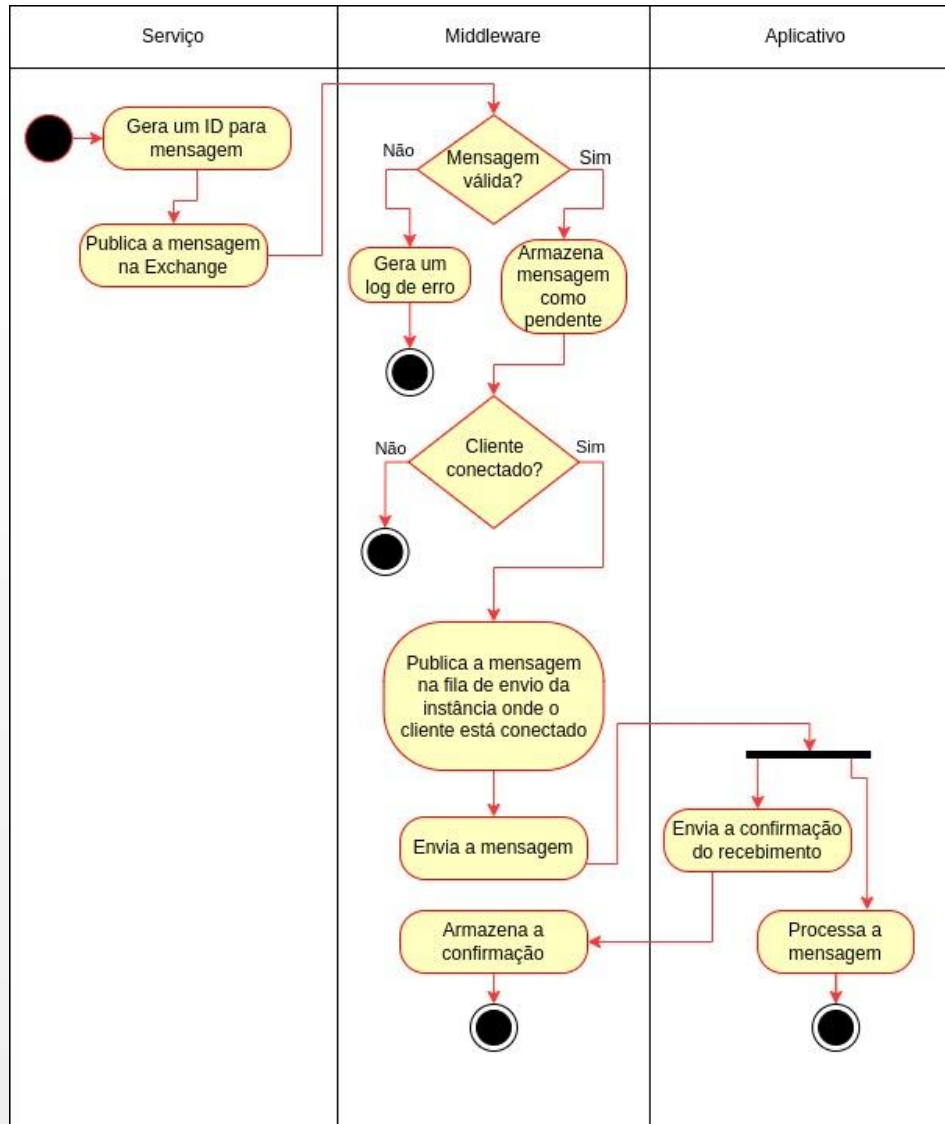
Especificação



Especificação



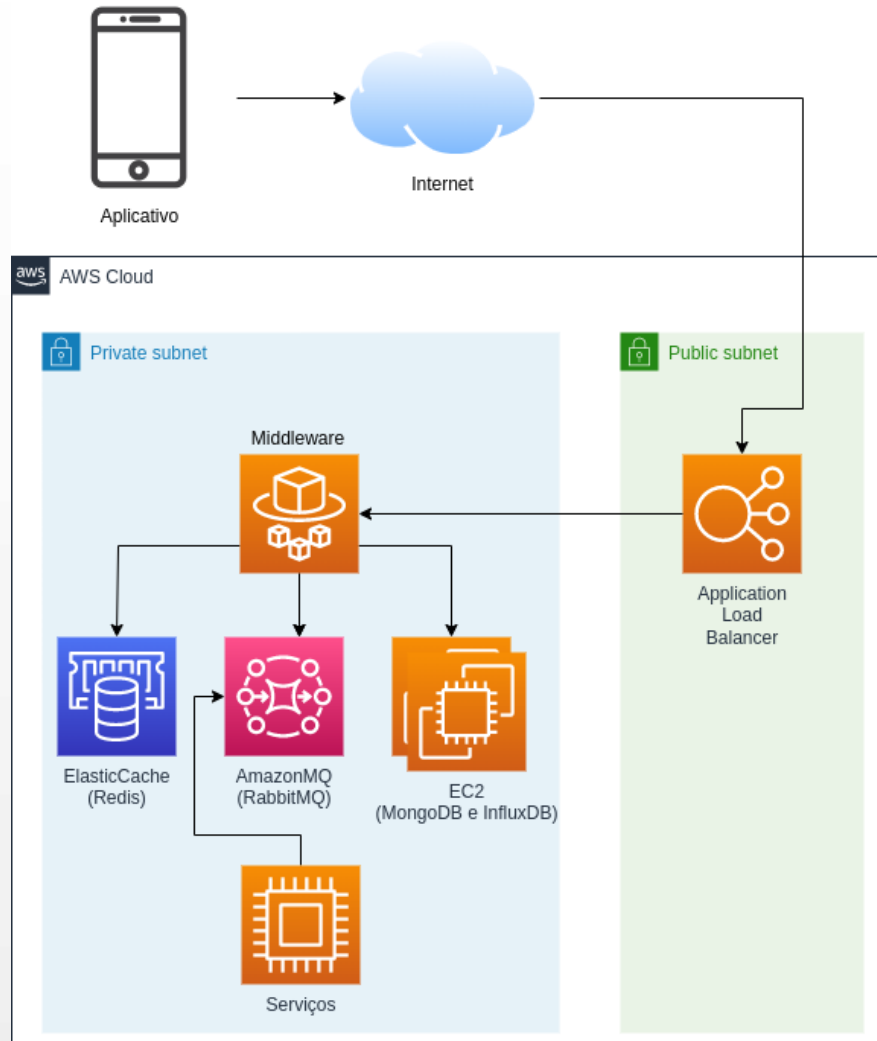
Especificação



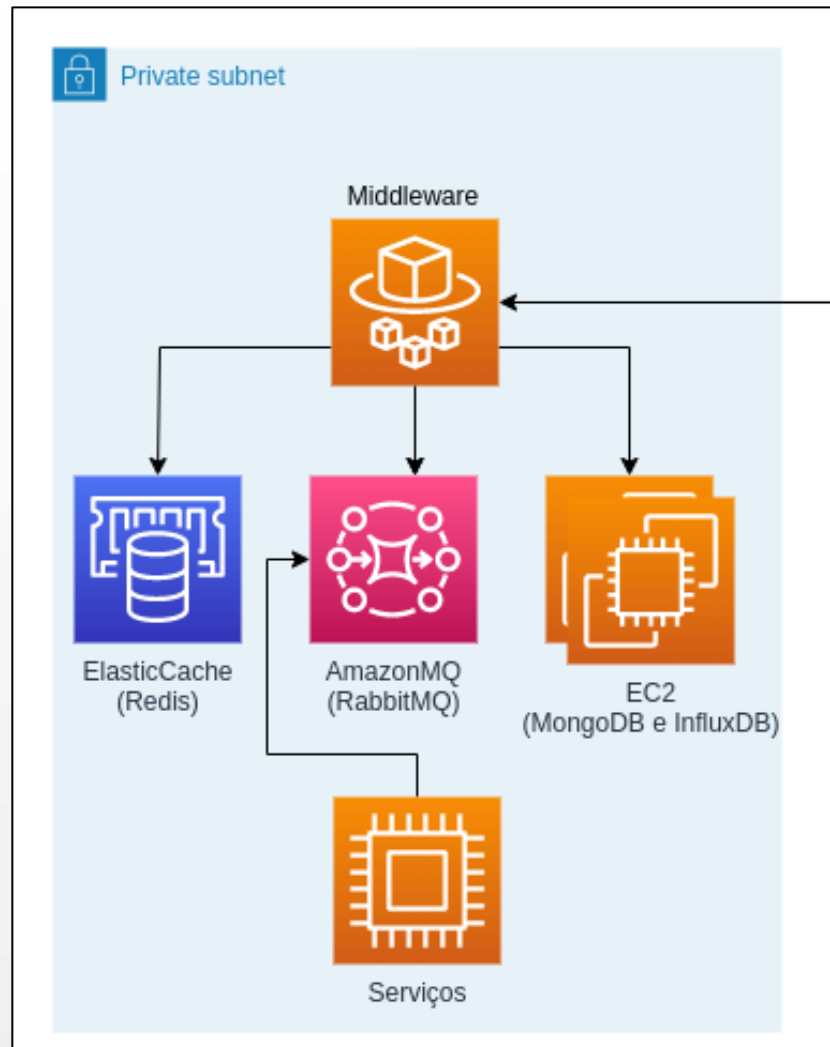
Implementação

- Quarkus
- Java 11
- RabbitMQ
- Redis
- MongoDB
- InfluxDB
- AWS

Implementação



Implementação

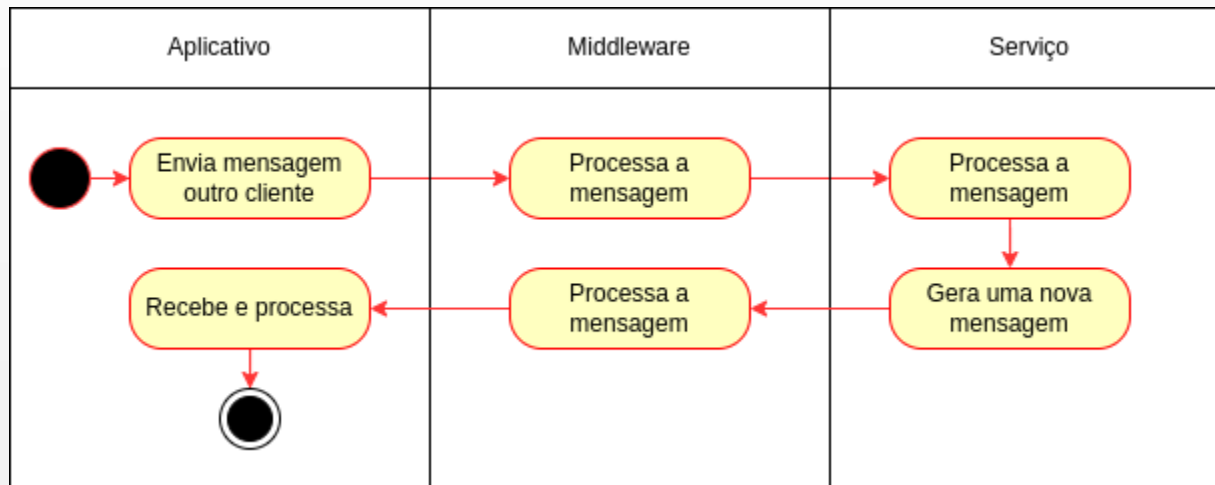


Implementação

- Registro de cliente
- Atualização por mensagem
- Atualização por Ping-Pong

Análise dos Resultados

- Sistema de chat
- Serviço chat
- Simulador de cliente de chat



Análise dos Resultados

- 4 instâncias serviço chat
- 4 instâncias simulador de cliente
- 1024 unidades de CPU
- 2GB de RAM
- ECS com 3 instâncias c6a.large (4vCPU e 8GB de RAM)
- Fargate 0,5CPU e 1GB de RAM

Análise dos Resultados

- Número de conexões constante
- Taxa de envio constante
- 10min cada teste

Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados

Cientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

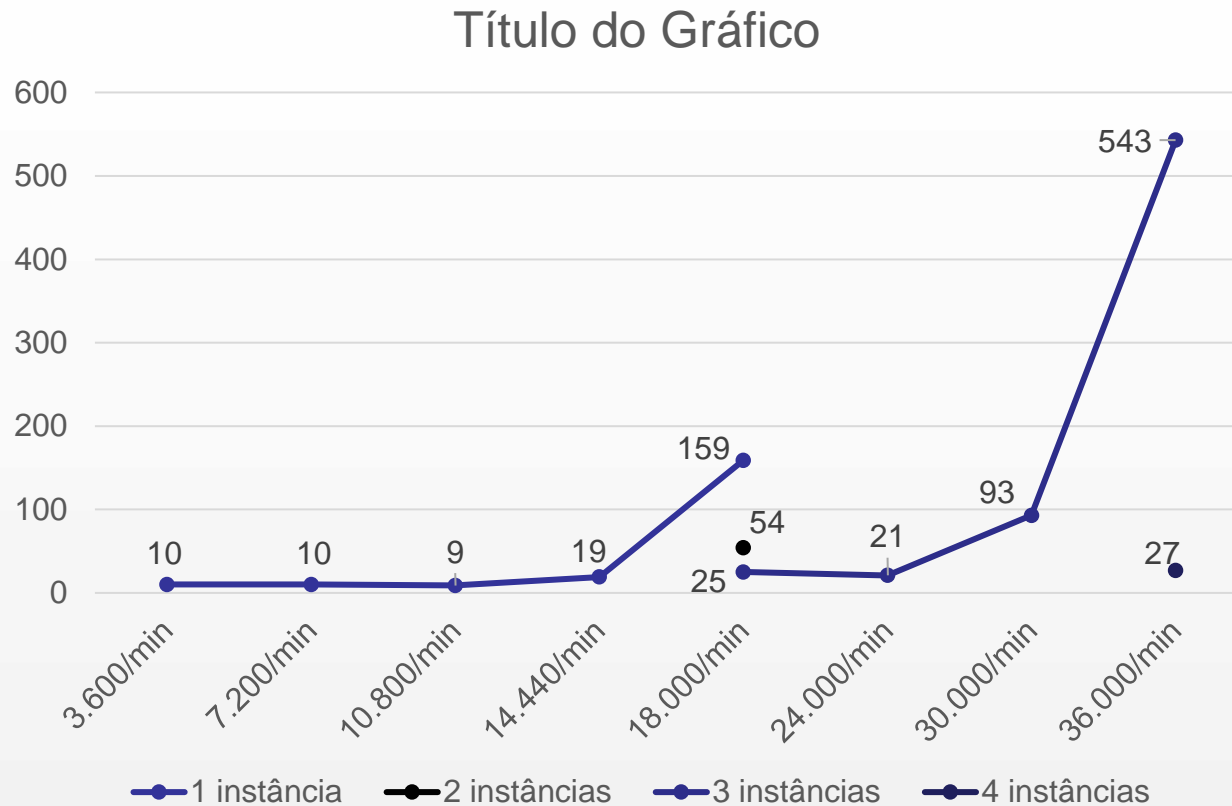
Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados

Clientes	Taxa de envio	Instâncias	CPU	RAM	Tempo resposta médio	Tempo resposta p(95)
60	3.600/min	1	26%	22%	6ms	10ms
120	7.200/min	1	45%	23%	7ms	10ms
180	10.800/min	1	66%	23%	8ms	9ms
240	14.440/min	1	84%	23%	9ms	19ms
300	18.000/min	1	98%	26%	47ms	159ms
300	18.000/min	2	69%	28%	29ms	54ms
300	18.000/min	3	47%	30%	16ms	25ms
400	24.000/min	3	59%	30%	15ms	21ms
500	30.000/min	3	79%	30%	31ms	93ms
600	36.000/min	3	95%	30%	81,6ms	543ms
600	36.000/min	4	72%	27%	20ms	27ms

Análise dos Resultados



Análise dos Resultados

- Resiliência realizada com testes manuais

Server URL (Required)

Enter the server URL:

```
wss://middleware.ariel-middleware.site/websocket?id=111111
```

Connection Status: **OPENED**

[Disconnect](#)

Send a Message

Enter a message and press Ctrl+Enter or click the Send button after connecting to a server:

```
{
  "id": "818e875f-2cdb-401d-ac81-6860ad588bb5",
  "route": "chat",
  "ttl": 1682494335000,
  "data": {
    "to": "222222",
    "message": "Mensagem para cliente desconectado."
  }
}
```

Análise dos Resultados

- Resiliência realizada com testes manuais

Server URL (Required)
Enter the server URL:

```
wss://middleware.ariel-middleware.site/websocket?id=222222
```

Connection Status: **OPENED**

[Disconnect](#)

Incoming Message

```
{
  "id": "2c958171-c073-414c-8c49-b5ed75017924",
  "from": "6296b74e888b7539953169fe",
  "data": {
    "message": "Mensagem para cliente desconectado."
    "from": "111111"
  }
}
```

Conclusões e Sugestões

- Atendeu a escalabilidade
- Atendeu a resiliência na entrega
- AWS contribuiu com a escalabilidade
- WebSocket nativo
- Simplificação na comunicação

Conclusões e Sugestões

- Gerar os identificadores pelo middleware
- Distribuir a aplicação por mais regiões do mundo
- Permitir comunicação com mais servidores do RabbitMQ
- Não possuir distinção de protocolo para servidor e cliente

Obrigado