

# Sistemas Distribuídos

## SISTEMA DE NOTIFICAÇÃO USANDO RABBITMQ + WEBSOCKET

**Discentes:**  
**Messias Acacy**  
**Raldnei Miguel**

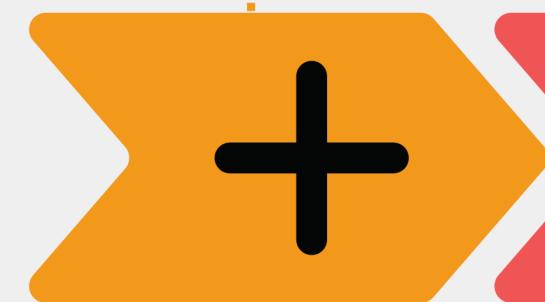
**Doscente:**  
**Luciano Cabral**



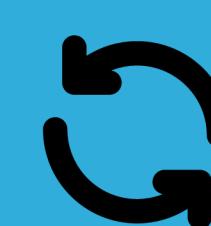


# OQUE O SISTEMA FAZ?

Cliente faz pedido



Cliente vê atualização em tempo real



Loja recebe e atualiza status

RabbitMQ faz a ponte



# IMPORTS

```
const express = require('express');
const amqp = require('amqplib');
const WebSocket = require('ws');
const path = require('path');
```



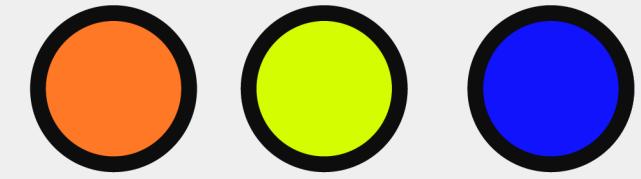


# ESTRUTURA

```
const appCliente = express();
const appLoja = express();

let pedidos = [];
let pedidoId = 1;
```





# CONEXÃO RABITMQ

```
connection = await amqp.connect('amqp://rabbitmq');
channel = await connection.createChannel();
await channel.assertExchange('pedidos_status', 'fanout');
```

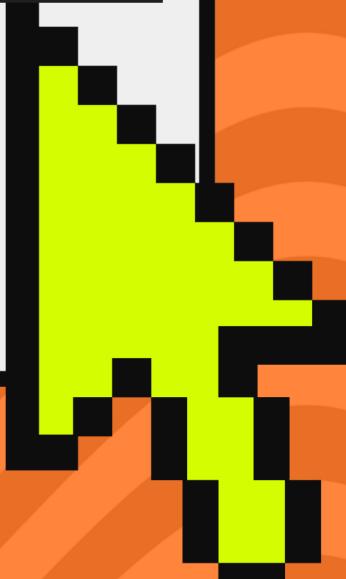




# CREATE & UPDATE

```
const pedido = { id: pedidoId++, descricao, quantidade: qtd, status: 'Pedido realizado', usuarioId };
pedidos.push(pedido);

// Publica no RabbitMQ para avisar outros serviços
try {
  channel.publish(EXCHANGE_NAME, '', Buffer.from(JSON.stringify({ type: 'novo_pedido', pedido })));
  setTimeout(() => {
    pedido.status = 'Em preparo';
    channel.publish(EXCHANGE_NAME, '', Buffer.from(JSON.stringify({ type: 'status_atualizado', pedido })));
  }, 1000);
  res.json(pedido);
}
```



# PROTOTIPO



Bem-vindo, **pastel** [Sair](#)

## Meus Pedidos

abacaxi

5  
▼

Fazer Pedido

# PROTOTIPO



Bem-vindo, **pastel** [Sair](#)

## Meus Pedidos

O que você quer pedir?

1

Fazer Pedido

abacaxi (x5)

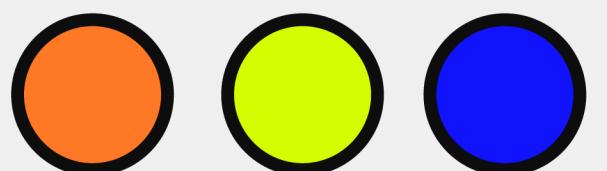
PEDIDO REALIZADO



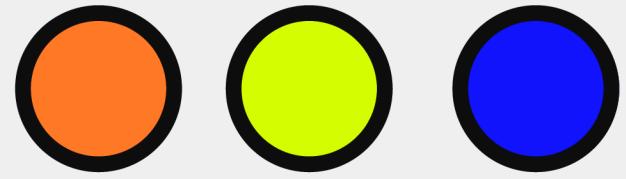
# PROTOTIPO

## Pedidos Recebidos

ID	Usuário	Descrição	Quantidade	Status	Alterar Status
1	1	4545	2	Em preparo	-- Alterar --
2	1	888	1	Saiu para entrega	-- Alterar --
3	pastel	abacaxi	5	Em preparo	-- Alterar --



**ENCERRANDO...**



Obrigado pela atenção

