

Fluxograma, Parte Funcional e Descrição da Arquitetura

1. Parte Funcional e Fluxo de Programação

O sistema Mava Baristro foi desenvolvido para gerenciar o atendimento de uma cafeteria, incluindo:

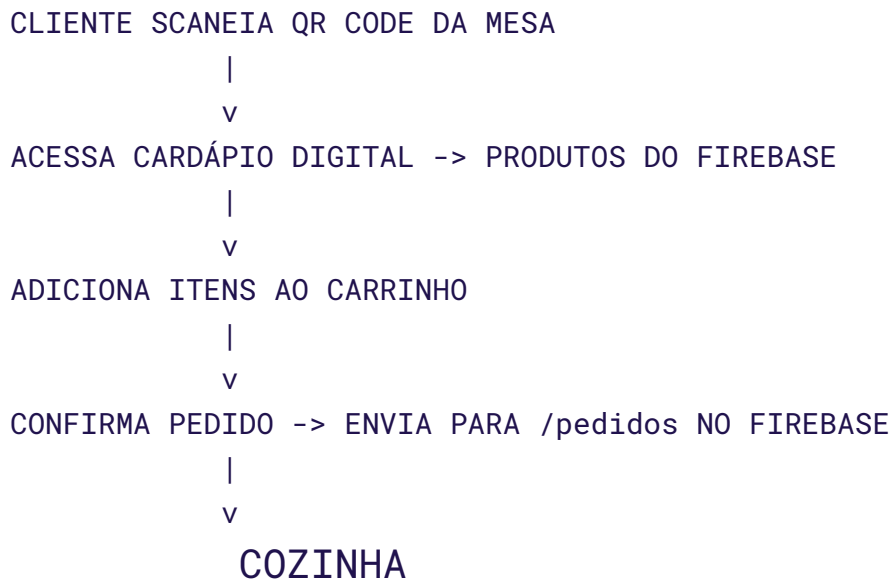
- controle de mesas
- pedidos feitos pelos clientes via QR Code
- acompanhamento dos pedidos na cozinha
- fechamento dos pedidos no caixa
- cadastro de produtos
- gestão de colaboradores
- painel administrativo (dashboard)

Todas as ações do sistema são registradas em tempo real no **Firestore Realtime Database**.

2. Fluxograma da Operabilidade do Sistema

Abaixo está o **fluxo principal do sistema**, do ponto de vista do cliente e da administração.

FLUXOGRAMA GERAL



```
RECEBE PEDIDO COM STATUS "novo"
|
v
COZINHEIRO MARCA "preparando"
|
v
QUANDO PRONTO -> MARCA "pronto"
|
v
```

CAIXA

```
CAIXA VÊ TODOS OS PEDIDOS PRONTOS POR MESA
|
v
SOMA TOTAL DA MESA
|
v
SELECIONA FORMA DE PAGAMENTO
|
v
FINALIZA PEDIDOS -> STATUS "finalizado"
|
v
LIBERA A MESA -> "livre"
```

3. Descrição da Arquitetura do Sistema

A arquitetura do Mava Baristro foi pensada para ser simples, funcional e baseada em comunicação em tempo real usando o Firebase.

1. Estrutura Geral

O sistema é dividido em **duas partes principais**:

Parte do Cliente (Cardápio via QR Code)

- Página HTML única ([cardapio.html](#))
- Cliente acessa produtos salvos no Firebase
- Pode adicionar produtos ao carrinho
- Envia o pedido para o banco em [/pedidos](#)
- Cada pedido inclui:
 - mesa

- nome do cliente
- itens
- total
- status

Parte Administrativa

Composta por várias páginas:

• Dashboard

- Mostra números em tempo real:
 - pedidos do dia
 - total vendido
 - mesas livres/ocupadas
 - total de produtos
 - total de colaboradores

• Produtos

- Cadastro, edição e exclusão de produtos
- Fotos enviadas em Base64
- Categorias: cafés, bebidas, doces, salgados

• Cozinha

- Tela para acompanhar os pedidos
- Ordenação automática: NOVOS → PREPARANDO → PRONTOS
- Alerta sonoro para novos pedidos
- Mudança de status pelo cozinheiro

• Caixa

- Mostra mesas com pedidos em aberto
- Lista de pedidos agrupados por mesa
- Escolha da forma de pagamento

- Finalização da mesa → libera mesa + fecha pedido

• Mesas

- Controle das 20 mesas do salão
- Cliente é vinculado à mesa
- QR Code gerado automaticamente
- Status: **livre** ou **ocupada**

• Colaboradores

- Cadastro de colaboradores com permissões
- Níveis de acesso:
 - cozinha
 - caixa
 - recepção
 - administrador

4. Como foram coletados os dados

Todos os dados são coletados diretamente de ações reais dentro do sistema. Exemplos:

- Quando um cliente envia um pedido → grava em [/pedidos](#)
- Quando o cozinheiro muda o status → atualiza o pedido
- Quando o caixa finaliza → salva pagamento e hora de fechamento
- Quando coloca nome na mesa → grava em [/mesas/](#)
- Quando um colaborador é cadastrado → salvo em [/colaboradores](#)
- Logs de atividades → gravados em [/logs_colaboradores](#)

Não há formulário específico para coleta de usuário, pois o sistema é operacional e alimentado pelas ações realizadas no dashboard e no cardápio.

5. Explicação da Arquitetura

O sistema usa:

- **HTML + CSS + JavaScript**
- **Firebase Authentication** (login de colaboradores)
- **Firebase Realtime Database** (salva tudo em tempo real)
- **API de QR Code** para gerar o código das mesas

Não usamos frameworks pesados (como React ou Angular), para manter o projeto simples e direto.

Os arquivos HTML funcionam como “telas independentes”, todas conectadas ao mesmo banco em tempo real.

6. Fluxos Específicos (funcionais)

1. Fluxo do Pedido

```
Cliente escolhe produto
      |
      v
Adiciona ao carrinho
      |
      v
Confirmar pedido
      |
      v
Firebase grava pedido (status = novo)
```

2. Fluxo da Cozinha

```
Pedido novo → Aparece com aviso sonoro
      |
cozinheiro marca PREPARANDO
      |
depois marca PRONTO
```

3. Fluxo do Caixa

```
Mesa com pedidos PRONTOS → aparece no caixa
      |
Somatória automática do total
      |
Escolhe forma de pagamento
      |
Finaliza → mesa fica LIVRE
```

4. Fluxo das Mesas

Administrador cadastra nomes na mesa

|

Mesa muda para OCUPADA

|

Quando finalizar → volta LIVRE

5. Fluxo dos Colaboradores

Admin cadastra colaborador

|

É criado login no Firebase Auth

|

Colaborador recebe senha

|

Registros de logs guardam as ações