Arquitetura: Clean Architecture

1. Apresentação (Interface Gráfica - View)

- Telas com informações dinâmicas (posição na fila, tempo estimado, pedidos, mesa)
- Botões de ação: “Sair da fila”, adicionar/remover item do pedido
- Exibe dados vindos da lógica de negócio (como status da fila e itens do cardápio)

2. Camada de Aplicação / Controller (ou ViewModel em MVVM)

- Coordena interações da interface com os casos de uso
- Exemplo:
    - Solicita dados de posição na fila
    - Atualiza status da mesa
    - Envia pedidos selecionados para a camada de domínio

3. Domínio (Casos de Uso)

- Regras como:
    - Controle de fila (entrada, saída, tempo estimado)
    - Controle de mesas disponíveis
    - Processamento de pedidos por mesa
- Aqui ficariam os **casos de uso**: EntrarNaFila, AtualizarStatusFila, RegistrarPedido, etc.

4. Camada de Dados (Repositories / Gateways)

- Comunicação com banco de dados ou API
- Exemplo:
    - FilaRepository, PedidoRepository, MesaRepository

Principais Componentes Arquiteturais

Componente	Função
FilaService	Gerencia a fila, posições e tempos estimados
MesaService	Libera e aloca mesas conforme a vez do cliente
PedidoService	Processa pedidos feitos a partir da mesa selecionada
UI Components	Telas modulares (fila, mesa, pedido)
Controller	Orquestra eventos da interface e regras de negócio
Repositories	Fazem acesso ao banco ou back-end
Models	Entidades como Cliente, Mesa, Pedido, Pizza, Fila

A estrutura do aplicativo está fortemente alinhada a **Clean Architecture** com inspiração em **MVC/MVVM**, separando claramente:

- Interface (UI)
- Regras de negócio (casos de uso)
- Acesso a dados

Essa arquitetura garante **escalabilidade**, **testabilidade** e **flexibilidade**, ideais para um sistema de restaurante automatizado com gestão de filas e pedidos.