

# Sistemas Distribuídos - GSI529

## Sistema de Gerenciamento de Pedidos de Restaurante

Nicolly Ribeiro Luz  
Matrícula: 32221BSI022

### 1 Descrição Detalhada do Tema do Micro-Mundo

O projeto consiste em um sistema de gerenciamento de pedidos para um restaurante, projetado como um sistema distribuído. O objetivo é simular o fluxo de pedidos desde a sua criação pelo cliente até a entrega final pelo garçom, passando pelas etapas de preparação na cozinha/bar. As entidades e componentes do sistema se comunicarão e coordenarão para garantir um fluxo eficiente e a atualização do status dos pedidos em tempo real.

#### Principais Interações Centrais a Simular

1. **Criação de Pedidos:** Clientes (via aplicativo ou web) fazem seus pedidos, que são enviados para o servidor central.
2. **Encaminhamento e Preparação:** O servidor de pedidos valida e encaminha os pedidos para as estações de trabalho apropriadas (Cozinha e Bar).
3. **Atualização de Status:** Cozinheiros e barmen atualizam o status dos pedidos (em preparação, pronto) em suas telas dedicadas.
4. **Entrega e Conclusão:** Garçons consultam os pedidos prontos e marcam-nos como entregues.
5. **Comunicação Contínua:** Todas as entidades envolvidas (clientes, servidor, estações de trabalho, garçons) se comunicam para manter o estado do sistema consistente e atualizado.

#### Entidades que Compõem o Sistema

1. **Cliente:** O usuário que faz o pedido. Contém ID e informações de contato.
2. **Pedido:** Representa o pedido em si. Contém ID, lista de itens, status (novo, preparando, pronto, entregue, cancelado), ID do cliente, e informações de mesa/entrega.
3. **Item de Menu:** Representa um item disponível no cardápio. Contém Nome, Descrição e Preço.
4. **Estação de Trabalho:** Representa as áreas de preparação (Cozinha, Bar). Contém uma fila de pedidos a processar.

## 2 Metas para meu Sistema Distribuído

No decorrer do projeto, as seguintes metas de sistemas distribuídos serão consideradas e, quando apropriado, implementadas:

- **Escalabilidade:** O sistema será projetado para permitir a adição de mais clientes, garçons e estações de trabalho (cozinha/bar) sem uma degradação significativa de desempenho. Isso pode envolver a replicação de serviços ou a distribuição de carga.
- **Disponibilidade:** Em caso de falha de um componente (ex: tela da cozinha), o objetivo é que o sistema continue funcionando, mesmo que com funcionalidade reduzida, para outras operações (ex: novos pedidos, tela do bar).
- **Tolerância a Falhas:** O sistema buscará mecanismos para lidar com erros ou falhas de componentes (ex: reconexão automática, retransmissão de mensagens), minimizando o impacto no fluxo de trabalho.
- **Compartilhamento de Recursos:** O estado dos pedidos (status, itens) e o menu do restaurante serão recursos compartilhados que precisarão ser acessados e atualizados por diferentes componentes do sistema de forma consistente.
- **Comunicação (Sockets, RPC, Mensagens):** A base da comunicação entre os diversos componentes será explorada através de diferentes paradigmas, garantindo a troca de informações e eventos.
- **Sincronização:** Será necessário garantir que o status de um pedido seja consistente em todos os componentes relevantes (servidor, tela da cozinha, terminal do garçom).

### 3 Diagrama da Arquitetura Inicial

Este diagrama de fluxo é um esboço e mostra como um sistema de gerenciamento de restaurante deve funcionar, tanto para os clientes, quanto para os funcionários que trabalham no restaurante.

É importante ressaltar que este é um modelo preliminar, podendo ser ajustado e expandido conforme a evolução do projeto e as necessidades identificadas durante o desenvolvimento. Sua estrutura serve como base para discussões e refinamentos futuros.

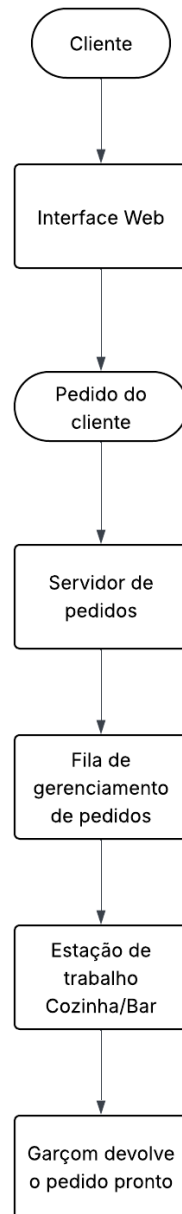


Figura 1: Diagrama de fluxo inicial do sistema de gerenciamento de um restaurante

## 4 Componentes do Sistema

### 1. Cliente de Pedido (App/Web):

- **Responsabilidades:** Interface para o cliente visualizar o menu, fazer novos pedidos e, opcionalmente, consultar o status de seus pedidos.
- **Tipo de Comunicação:** Cliente-Servidor (envia pedidos, recebe confirmações/atualizações de status).
- **Implementação:** Será um processo (aplicativo ou web service cliente) que interage com o Servidor de Pedidos.

### 2. Servidor de Pedidos (Principal):

- **Responsabilidades:** Recebe os pedidos dos clientes, valida-os, gerencia o estado geral dos pedidos, e os encaminha para as estações de trabalho apropriadas. Atua como um ponto central de coordenação.
- **Tipo de Comunicação:** Cliente-Servidor (com clientes e estações de trabalho), Ponto-a-Ponto (com Gerenciador de Fila de Pedidos e estações de trabalho para comunicação de status).
- **Implementação:** Será um processo principal, com múltiplos threads para lidar com as requisições de clientes (fazer pedido) e com as atualizações de status das estações de trabalho.

### 3. Gerenciador de Fila de Pedidos:

- **Responsabilidades:** Componente que distribui os pedidos para as estações de trabalho (Cozinha, Bar) com base no tipo de itens do pedido. Pode gerenciar prioridades.
- **Tipo de Comunicação:** Ponto-a-Ponto (com Servidor de Pedidos, enviando pedidos para estações de trabalho).
- **Implementação:** Pode ser um thread dentro do Servidor de Pedidos ou um processo separado para maior escalabilidade, dependendo da complexidade. Inicialmente, um thread no Servidor de Pedidos.

### 4. Estação de Trabalho (Cozinha/Bar):

- **Responsabilidades:** Processa os pedidos recebidos, indicando seu status (em preparação, pronto).
- **Tipo de Comunicação:** Cliente-Servidor (recebe pedidos do Servidor de Pedidos, envia atualizações de status de volta).
- **Implementação:** Será um processo separado para cada estação (Cozinha, Bar), rodando em terminais dedicados e se comunicando com o Servidor de Pedidos.

### 5. Tela da Cozinha/Bar:

- **Responsabilidades:** Interface visual para cozinheiros/baristas verem os pedidos a preparar e atualizarem o status (em preparação, pronto).
- **Tipo de Comunicação:** Cliente-Servidor (recebe atualizações do Servidor de Pedidos, envia ações de atualização).
- **Implementação:** Será um thread para manter a interface atualizada e outro para ouvir por novos pedidos do servidor. Implementado como parte do processo da Estação de Trabalho ou como um cliente separado que se conecta à Estação de Trabalho.

### 6. Terminal do Garçom:

- **Responsabilidades:** Interface para garçons verem pedidos prontos e marcarem-nos como entregues.
- **Tipo de Comunicação:** Cliente-Servidor (consulta status de pedidos prontos, envia atualização de status).
- **Implementação:** Será um processo separado, se comunicando com o Servidor de Pedidos.