

REINAN CARVALHO
LUCCA CINTRA
DIEGO PIMENTA
FERNANDO CAIRES
HÉLIO JÚNIOR

FILAS EM ESTRUTURA DE DADOS: IMPLEMENTAÇÃO E APLICAÇÕES EM
JAVA

Trabalho apresentado ao curso de Ciência da Computação, como requisito parcial para obtenção de nota na disciplina de Estrutura de Dados.

CIÊNCIA DA COMPUTAÇÃO
2025

FILAS EM ESTRUTURA DE DADOS: IMPLEMENTAÇÃO E APLICAÇÕES EM JAVA

REINAN CARVALHO, LUCCA CINTRA, DIEGO PIMENTA, FERNANDO
CAIRES, HÉLIO JÚNIOR

Ciência da Computação – 2025

RESUMO

Este artigo apresenta a estrutura de dados Fila, destacando sua importância teórica e aplicações práticas. Para exemplificar, foi desenvolvido um simulador de filas de atendimento em cinemas, usando a linguagem Java. O sistema simula múltiplos guichês, cada um com sua fila de clientes, permitindo redistribuição automática em caso de pausas, além de priorizar certos grupos (idosos, estudantes, etc.). A simulação mostra como a estrutura de fila organiza o atendimento sequencial e lida com situações reais, como sobrecarga e interrupções. O estudo destaca o papel das filas em sistemas concorrentes e sua aplicação em diversos contextos, como hospitais, bancos e sistemas operacionais.

Palavras-chave: Fila; Estrutura de dados; Java.

1. INTRODUÇÃO

Organizar tarefas de forma sequencial é essencial na computação. Muitos sistemas precisam atender solicitações por ordem de chegada, como em filas de impressão, servidores web ou atendimento ao público. Para isso, a estrutura de dados fila (do inglês queue) é amplamente utilizada, seguindo o princípio FIFO (First In, First Out).

Neste trabalho, utilizamos o ambiente de um cinema com vários guichês para simular o comportamento das filas. Cada guichê possui uma fila associada, e os clientes são distribuídos conforme regras de balanceamento e prioridade.

A aplicação foi desenvolvida em Java, usando estrutura de lista ligada e inclui recursos como visualização do estado das filas, redistribuição de clientes e priorização de atendimento.

Embora o cenário seja um cinema, o objetivo é mostrar como filas ajudam a resolver problemas de organização e desempenho em sistemas reais. Nos próximos tópicos, vamos explicar o funcionamento da estrutura, a lógica do simulador e os resultados obtidos.

2. ESTRUTURA DE DADOS FILA – CONCEITOS E VARIAÇÕES

A estrutura de dados do tipo fila é uma coleção ordenada de elementos em que as operações seguem uma lógica de entrada e saída em sequência. Esse comportamento é caracterizado como FIFO (First In, First Out), ou seja, o primeiro elemento a entrar na fila será também o primeiro a sair. Esse padrão é ideal para sistemas em que a ordem de chegada dos elementos deve ser respeitada, especialmente em contextos de atendimento, processamento e execução de tarefas.

2.1 FUNCIONAMENTO BÁSICO

As principais operações de uma fila são: Enqueue (inserir), adiciona um elemento no final da fila. Dequeue (remover), retira o elemento do início da fila. Peek, permite ver o primeiro elemento sem removê-lo. Essas operações são simples, mas muito poderosas quando usadas corretamente.

2.2 TIPOS DE FILAS

Diversas variações foram criadas para atender diferentes necessidades como Fila Linear, forma básica, ideal para atendimentos simples. Implementações com listas encadeadas evitam o custo de realocação. Fila Circular, reaproveita espaço em um vetor, útil em sistemas embarcados e buffers. Fila de Prioridade (Priority Queue), atende primeiro quem tem maior prioridade, mesmo que tenha chegado depois. Deque (Double-Ended Queue), permite inserções e remoções nas duas extremidades, oferecendo mais flexibilidade.

2.3 APLICAÇÕES

Filas são usadas em: Sistemas operacionais (escalonamento de processos), Redes (transmissão de dados), Jogos e interfaces (eventos), Atendimentos reais (bancos, hospitais, cinemas). Neste projeto, usaremos a fila para simular um ambiente de cinema com múltiplos guichês, refletindo como a estrutura pode ser aplicada em cenários reais com múltiplos atendimentos e eventos inesperados.

3. DESCRIÇÃO DO PROBLEMA E CENÁRIO DE APLICAÇÃO

Embora o foco do trabalho seja a estrutura de dados fila, o cenário escolhido para exemplificar sua aplicação foi um cinema com múltiplos guichês de atendimento. Esse ambiente simula uma situação comum onde é necessário organizar o atendimento de forma justa, rápida e adaptável.

3.1 O CENÁRIO

O sistema simula N guichês, cada um com sua própria fila (Fila<Cliente>), onde os clientes são alocados na fila mais curta disponível. O atendimento ocorre em ordem de chegada (FIFO), mas o sistema também pode aplicar prioridade para certos clientes, como idosos ou estudantes.

3.2 EVENTOS DINÂMICOS

Para tornar a simulação mais realista, o sistema considera situações de pausas ou fechamento de guichês, redistribuição automática de clientes para filas ativas, simulação de tempo de atendimento, atualização visual do estado de todas as filas em tempo real.

Esses eventos simulam o comportamento de um sistema real que precisa se adaptar a mudanças sem perder eficiência.

3.3 O PAPEL DAS FILAS

Mais do que simular um cinema, o objetivo é representar um problema algorítmico, como dividir tarefas (clientes) entre vários pontos de atendimento (guichês), mantendo ordem, eficiência e flexibilidade. As filas são a solução ideal, pois garantem ordem de atendimento, permitem redistribuição equilibrada, Suportam atendimento por prioridade, facilitam a expansão e adaptação do sistema.

A partir desse modelo, é possível entender como filas funcionam em sistemas maiores, como hospitais, call centers ou servidores.

4. SIMULAÇÃO E ANÁLISE DO FUNCIONAMENTO

A simulação foi desenvolvida para mostrar, na prática, como a estrutura de fila se comporta em um ambiente dinâmico com múltiplos guichês. O sistema acompanha o ciclo de entrada, atendimento e redistribuição de clientes, permitindo observar a eficiência da organização por filas.

4.1 CICLO DA SIMULAÇÃO

A execução da simulação segue quatro etapas principais: adição de clientes, atendimento, eventos dinâmicos e visualização do estado. Na adição de clientes, cada novo cliente é inserido na fila mais curta disponível. Caso possua prioridade, é encaminhado para uma PriorityQueue, respeitando sua categoria. Na etapa de atendimento, a cada “tick” de tempo, um cliente é atendido e removido da fila, sendo o tempo médio ajustável conforme necessário. Durante os eventos dinâmicos, se um guichê for pausado ou encerrado, os clientes daquela fila são redistribuídos automaticamente entre as demais filas ativas, garantindo o equilíbrio do sistema. Por fim, na visualização do estado, o sistema exibe em tempo real a situação de cada fila, incluindo os clientes em espera, seus tipos e o tempo estimado para atendimento.

4.2 EXEMPLO DE FUNCIONAMENTO (CENÁRIO ILUSTRATIVO)

Vamos considerar um cenário com 3 guichês: Inicialmente, todas as filas estão vazias. Clientes A, B, C e D chegam em sequência: A → Guichê 1 (todas vazias), B → Guichê 2 (tamanho 0), C → Guichê 3 (tamanho 0), D → Guichê 1 (tamanho 1, mais curta entre as opções restantes).

Guichê 2 entra em pausa, Clientes na fila de Guichê 2 (por exemplo, apenas B) são redistribuídos para Guichê 1 ou 3. O sistema verifica o tamanho atual de cada fila e escolhe a de menor carga. Após alguns ciclos, os clientes são atendidos sequencialmente (um por vez, por tick).

Clientes E e F entram: São adicionados conforme o novo balanço de carga, levando em conta os guichês ativos. Esse fluxo mostra claramente a utilização da estrutura de fila para garantir ordem, adaptar-se a eventos e manter o sistema equilibrado.

4.3 FILA COM OU SEM PRIORIDADE

Sem prioridade: todos seguem ordem de chegada. Com prioridade: clientes com necessidades especiais (idosos, estudantes) são atendidos primeiro, mesmo que tenham chegado depois.

5. DISCUSSÃO TÉCNICA: A FILA COMO SOLUÇÃO CENTRAL

Embora a simulação use um cinema como exemplo, o foco real do projeto é demonstrar como a estrutura de fila resolve problemas de organização em ambientes concorrentes.

5.1 POR QUE USAR FILA?

A fila foi escolhida por ser a melhor opção quando a ordem de chegada precisa ser respeitada. Seu uso garante previsibilidade no atendimento, evita clientes ignorados (starvation), permite redistribuir tarefas sem perder a ordem e dá suporte à priorização, quando necessário.

5.2 REDISTRIBUIÇÃO: O DESAFIO REAL

Quando um guichê é pausado, a fila associada precisa ser redistribuída. O algoritmo retira todos os clientes da fila pausada, identifica as filas ativas com menos clientes, redistribui mantendo a ordem e o equilíbrio. Esse processo mostra como a fila pode ser adaptável e resiliente a falhas ou mudanças no sistema.

5.3 POSSÍVEIS APLICAÇÕES

A lógica usada no simulador pode ser aplicada em diversos contextos: hospitais (fila para triagem ou consultas), bancos e serviços públicos, call centers e suporte técnico, sistemas operacionais, para controle de processos.

Também é possível trocar o tipo de fila conforme a necessidade: LinkedList → fila básica, PriorityQueue → atendimento com prioridade, ConcurrentLinkedQueue → para sistemas paralelos, Deque → inserções e remoções nas duas pontas.

5.4 LIMITAÇÕES E MELHORIAS FUTURAS

Apesar da boa performance, o sistema atual pode evoluir: Adicionar tempo real com relógio real (em vez de ticks), criar uma interface gráfica mais intuitiva, salvar dados da simulação para análise posterior, implementar algoritmos mais inteligentes de balanceamento, usando tempo de espera ao invés do tamanho da fila. Essas melhorias deixariam o sistema mais completo, sem perder a simplicidade da ideia central.

6. CONCLUSÃO

A estrutura de dados Fila é simples, mas extremamente útil para organizar tarefas de forma justa e eficiente. Neste trabalho, mostramos como ela pode ser aplicada em sistemas de atendimento, usando como exemplo um simulador de guichês de cinema.

Com a implementação em Java, foi possível organizar atendimentos em ordem de chegada (FIFO), redistribuir clientes em caso de falhas nos guichês, priorizar

certos tipos de clientes, como idosos e estudantes, coletar métricas que avaliam o desempenho do sistema.

Mais do que um exemplo didático, o projeto mostrou que filas são essenciais em ambientes com múltiplos atendimentos simultâneos — como hospitais, call centers, ou sistemas operacionais. O entendimento dessa estrutura permite criar sistemas mais justos, escaláveis e adaptáveis.

Possibilidades futuras incluem interfaces mais visuais, uso de dados reais para simulações, integração com bancos de dados, suporte a múltiplas threads e processamento paralelo.

O domínio de estruturas como a fila é um passo importante para qualquer profissional da área de computação, pois permite projetar soluções eficazes em diversas situações do mundo real.

REFERÊNCIAS

GONÇALVES, W. Estruturas de Dados em Java: algoritmos, pilhas, filas e listas encadeadas. São Paulo: Novatec Editora, 2018.

GOODRICH, M. T.; TAMASSIA, R.; GOLDWASSER, M. H. Estruturas de Dados e Algoritmos em Java. 6. ed. Rio de Janeiro: LTC, 2016.

LOUREIRO, A.; MALUCELLI, A. Estruturas de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++. São Paulo: Thomson Learning, 2004.