

NEBULOSA JAVA

EXPLORANDO O GARBAGE COLLECTOR

COMPREENDA COMO O JAVA MANTÉM O
SEU SISTEMA LIMPO E PERFORMÁTICO

JOÃO OLIVEIRA

JAVA GARBAGE COLLECTOR

Descomplicando a limpeza do seu projeto

O Garbage Collector é uma peça fundamental do ecossistema Java. Entender seu funcionamento é sem dúvidas indispensável para todo desenvolvedor Java.

Neste e-book, vamos nos aventurar nessa tecnologia e entender melhor seu funcionamento.



01

ESTRUTURA DA MEMÓRIA

Imagine a memória do Java como um grande prédio onde cada andar tem um propósito específico — e o zelador do prédio é o Garbage Collector (GC), removendo os objetos que não são mais usados.

ESTRUTURA DA MEMÓRIA

A Java Virtual Machine (JVM) gerencia a memória através dos segmentos de memória: Stack, Heap e Metaspace.

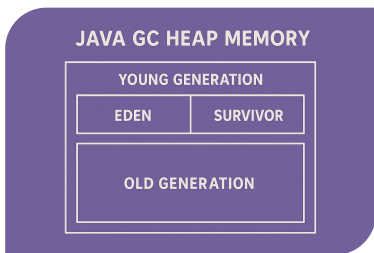
O principal segmento de memória usada pelo GC é a Heap (pilha de objetos) e é dividida em duas grandes áreas.



ESTRUTURA DA MEMÓRIA

Young Generation (Geração Jovem):
Onde os objetos recém-criados "nascem". Pense como o berçário do prédio. A maioria dos objetos vive pouco tempo e já são descartados aqui.

Old Generation (Geração Antiga):
Quando um objeto sobrevive tempo suficiente no berçário, ele "cresce" e é movido para cá. Este é o andar dos objetos duradouros.



02

YOUNG GENERATION

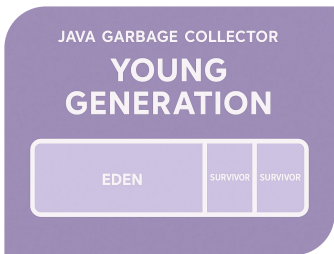
O Java GC gerencia a criação de novos objetos por meio do segmento Young Generation. Confira:

YOUNG GENERATION

A Young Generation (Geração Jovem) é o lugar onde os objetos são criados. Eden é onde eles "nascem".

Quando o Eden enche, o GC entra em ação e remove os objetos que não estão mais em uso. Os sobreviventes são movidos para os Survivor Spaces.

Após várias passagens por esses espaços, os objetos que continuam vivos são promovidos para a Old Generation.



03

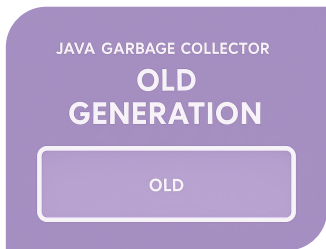
OLD GENERATION

O Java GC mantém controle sobre objetos que vivem por muito tempo separando-os em um segmento de memória dedicado.

OLD GENERATION

Objetos que vivem por muito tempo são movidos para a Old Generation (Geração Antiga). Eles são considerados estáveis e, por isso, o GC os verifica com menos frequência.

Mas, eventualmente, essa área também precisa de limpeza, e isso é mais custoso — é como reformar um apartamento antigo: dá mais trabalho.



04

QUANDO UM OBJETO PODE SER COLETADO?

Um objeto é considerado “lixo” quando não pode mais ser acessado por nenhuma parte do código.

QUANDO UM OBJETO PODE SER COLETADO?

Pense em uma casa abandonada, sem ninguém morando ou tendo as chaves. Para o GC, esse objeto pode ser removido com segurança.

Java usa o conceito de "reachability" (acessibilidade). Se não há caminhos a partir das variáveis em uso até o objeto, ele está pronto para ser coletado.



05

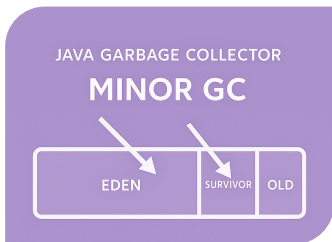
MINOR GC & MAJOR GC

O Garbage Collector executa dois processos de coleta de objetos, que podem impactar o desempenho do sistema. Confira abaixo:

MINOR GC

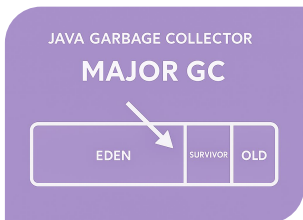
Minor GC é uma limpeza rápida feita na Young Generation. Ela é rápida e frequente, remove objetos de curta vida e afeta pouco o desempenho do sistema.

O processo é como esvaziar a lixeira da mesa de trabalho — simples, rápido, rotineiro.



MAJOR GC (FULL GC)

É a faxina completa, incluindo a Old Generation. O processo é mais lento e pesado, pode causar pausas na aplicação e é necessário de tempos em tempos.



06

TIPOS DE GARBAGE COLLECTOR

Java oferece diferentes coletores de lixo, cada um com seu estilo de trabalho. Veja a lista a seguir:

TIPOS DE GARBAGE COLLECTOR

Serial GC:

Usa um único thread para coleta. É Ideal para aplicações pequenas ou em ambientes com poucos recursos (ex: IoT). Apresenta pausas longas.

Parallel GC (ou Throughput GC):

Usa múltiplos threads para coletar. Foca em desempenho e throughput (processar mais em menos tempo) com pausas menores que no Serial.

CMS (Concurrent Mark-Sweep):

Coleta parte do lixo enquanto a aplicação ainda roda, com menores pausas, mas consome mais CPU.



TIPOS DE GARBAGE COLLECTOR

G1 GC (Garbage-First):

Divide o heap em regiões e otimiza coleta com menor impacto na performance. Balanceia pausas e desempenho.

ZGC (Z Garbage Collector):

Coleta quase sem pausas (pausas < 10ms), com suporte a heaps muito grandes (até terabytes). Usa muita tecnologia interna (barreiras de escrita/leitura).

Shenandoah GC:

Similar ao ZGC, mas focado em pausas curtas também. Ótimo para ambientes interativos e responsivos.



AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Esse ebook foi gerado por IA e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção e pode conter erros gerados por IA.



[Github](#)



[Linkedin](#)

