

diferença entre listas

resumo das diferenças:

Aa Característica	≡ Vector	≡ ArrayList	≡ LinkedList
<u>Estrutura</u>	Array dinâmico	Array dinâmico	Lista duplamente encadeada
<u>Crescimento</u>	Dobra a capacidade	Aumenta por 50%	Não há realocação (uso de nós)
<u>Thread-Safety</u>	Sincronizado	Não sincronizado	Não sincronizado
<u>Desempenho (Acesso).</u>	Lento (sincronização)	Rápido	Lento (acesso sequencial)
<u>Desempenho (Inserção/Remoção).</u>	Razoável (relocação necessária)	Razoável (relocação necessária)	Rápido (sem realocação, mas pode ser lento para acesso aleatório)
<u>Uso comum</u>	Obsoleto em novas versões	Amplamente utilizado	Usado quando inserção/remoção é frequente

Exemplos de quando usar cada um:

- **Vector** : Geralmente, você deve evitar o uso de **Vector** em novos projetos, exceto se for necessário garantir a sincronização (embora outras soluções, como **CopyOnWriteArrayList**, possam ser mais eficientes).
- **ArrayList** : Ideal quando você precisa de acesso rápido a elementos e não se preocupa com concorrência. É a escolha mais comum.
- **LinkedList** : Útil quando você precisa de inserções e remoções rápidas em qualquer parte da lista, mas não precisa de acesso aleatório frequente.