

### Associação Múltipla com Arrays

1. **A Turma (Implementação com Array):** Crie a classe Turma conforme visto em aula. Ela deve ter um atributo `private Aluno[] alunos;` e um `private int proximaPosicaoLivre;`. O construtor deve receber a capacidade máxima da turma e inicializar o array. Implemente os métodos `matricularAluno(Aluno aluno)` (com verificação de limite) e `listarAlunos()`.
2. **Buscando na Turma:** Adicione um método `public Aluno buscarAlunoPorRa(int ra)` à sua classe Turma. Este método deve percorrer o array de alunos (apenas as posições preenchidas) e retornar o objeto Aluno se encontrar um com o RA correspondente. Se não encontrar, deve retornar `null`.
3. **O Problema da Remoção (Conceitual):** Em um comentário no seu código, descreva com suas palavras os passos que você precisaria seguir para implementar um método `removerAluno(int ra)`. Quais são as principais dificuldades e riscos dessa implementação usando um array?
4. **A Playlist (com Array):** Crie uma classe Playlist que pode armazenar até 20 Musicas (use a classe Musica de exercícios anteriores). Implemente a mesma lógica da Turma: um construtor que inicializa o array, um atributo para controlar a próxima posição e um método `adicionarMusica(Musica musica)`.
5. **O Erro Oculto (Conceitual):** No método `listarAlunos` da sua classe Turma, por que o loop `for` deve ir até `proximaPosicaoLivre` e não até `alunos.length`? O que aconteceria se você usasse `alunos.length` para percorrer e tentasse imprimir `alunos[i].getNome()`?

### Simplificando Coleções com ArrayList

6. **O Carrinho de Compras:** Implemente a classe `CarrinhoDeCompras` do zero, conforme o desafio da aula. Ela deve usar um `private List<Livro> itens;`. Implemente os métodos:
  - `adicionarItem(Livro livro)`
  - `removerItem(Livro livro)`
  - `listarItens()`
  - `calcularTotal()`
7. **Refatorando a Turma:** Pegue a sua classe Turma do exercício 1, que usava um array. Refatore-a completamente para usar um `private List<Aluno> alunos;`. Remova o contador manual e simplifique os métodos `matricularAluno` e `listarAlunos` (usando `for-each`).
8. **Novos Poderes para a Turma:** Na sua Turma refatorada, adicione os seguintes métodos, que agora são muito mais fáceis de implementar:
  - `public int getNumeroDeAlunos():` Retorna o número de alunos na turma.
  - `public Aluno getAlunoNaPosicao(int indice):` Retorna o aluno em uma posição específica da lista.
  - `public void removerAluno(Aluno aluno):` Remove um aluno da lista.
9. **Busca Simplificada:** Reimplemente o método `buscarAlunoPorRa(int ra)` na sua Turma refatorada com `ArrayList`. Compare a simplicidade desta versão com a versão que usava array.
10. **Time de Futebol:** Crie as classes `Jogador` e `Time`. A classe `Time` deve ter um atributo `private List<Jogador> jogadores;`. Implemente métodos para `contratarJogador(Jogador jogador)` e `exibirEscalacao()`.