

Universidade Tuiuti do Paraná
Curso: Análise e Desenvolvimento de Sistemas
2º Estudo Dirigido de Estruturas de Dados - 16/05/25
Prof. Baroni

Simulador de Player de Músicas em C

É o ano de 2025 e a startup "MelodiaCode" está desenvolvendo uma nova plataforma de streaming musical. Para treinar sua equipe de desenvolvedores júnior, a empresa criou um desafio de programação: desenvolver um simulador de player de música em C.

Como novo contratado da MelodiaCode, você foi designado para criar este simulador como seu projeto de integração. O objetivo não é criar um player que realmente toque músicas, mas sim simular as funcionalidades e operações internas de um player, utilizando estruturas de dados fundamentais.

João Silva, líder técnico da MelodiaCode, explicou na sua primeira reunião:

"Este projeto vai ajudar você a entender como os dados são organizados e manipulados em nossos sistemas reais. Um player de música é um excelente exemplo prático para aprender sobre listas, pilhas e filas - estruturas que usamos em toda nossa plataforma. Queremos ver como você implementa essas estruturas e as utiliza para resolver problemas concretos."

Esta é sua oportunidade de demonstrar suas habilidades em programação em C e seu entendimento sobre estruturas de dados fundamentais.

Objetivo do Projeto

Desenvolver um simulador de player de músicas em C que utilize:

- Listas duplamente encadeadas;
- Pilhas;
- Filas;

O simulador deve demonstrar o uso adequado de:

- Ponteiros e alocação dinâmica de memória;
- Manipulação adequada de estruturas de dados;
- Modularização de código;
- Interface de usuário via terminal;

O trabalho pode ser feito em até 4 alunos;

Requisitos Funcionais

O simulador "MusicPlayer" deve incluir as seguintes funcionalidades:

1. **Gerenciamento de Músicas**
 - Cadastrar músicas (título, artista, duração);
 - Remover músicas da biblioteca;
 - Listar todas as músicas cadastradas;
 - Buscar músicas por título ou artista;
2. **Simulação de Playlist**
 - Criar playlists usando listas duplamente encadeadas;
 - Adicionar e remover músicas das playlists;
 - Navegar entre músicas (próxima, anterior);
 - Exibir detalhes da playlist atual;
3. **Histórico de Reprodução**
 - Manter histórico das últimas músicas "reproduzidas" usando pilha;
 - Permitir voltar às músicas recentes do histórico;
4. **Fila de Reprodução**
 - Adicionar músicas à fila de reprodução;
 - Visualizar próximas músicas na fila;
5. **Funcionalidades de Simulação**
 - "Reproduzir" uma música (simulação via console);
 - Pausar e retomar reprodução;
 - Avançar e retroceder na playlist;
 - Modo aleatório (shuffle);

Requisitos Técnicos

1. Estruturas de Dados:

As seguintes estruturas devem ser implementadas do zero, sem utilizar bibliotecas prontas:

Lista Duplamente Encadeada:

- Implementação completa com nós contendo ponteiros para anterior e próximo
- Operações: inserção (início, meio, fim), remoção, busca, navegação em ambas direções
- Utilização para armazenamento da biblioteca musical e playlists

Pilha:

- Implementação usando alocação dinâmica
- Operações: push, pop, peek, isEmpty
- Utilização para gerenciar o histórico de reprodução

Fila:

- Implementação usando alocação dinâmica
- Operações: enqueue, dequeue, peek, isEmpty
- Utilização para gerenciar a fila de reprodução

2. Gerenciamento de Memória

- Alocação dinâmica apropriada para todas as estruturas
- Liberação adequada de memória para evitar vazamentos
- Gerenciamento correto de ponteiros

3. Modularização

- Funções bem definidas com responsabilidades claras
- Comentários explicativos sobre o funcionamento de cada componente

4. Interface

- Menu interativo via console
- Feedback visual claro para cada operação
- Sistema de navegação intuitivo entre diferentes funcionalidades

Critérios de Avaliação

Critérios	Peso	Explicação
Funcionalidade	40%	O programa funciona corretamente com todos os requisitos funcionais?
Arquitetura	30%	O design é bem estruturado, modular e segue princípios de OO?
Implementação	20%	Uso adequado de estruturas de dados, padrões de design e multithreading;
Relatório técnico;	10%	Relatório completo, claro, bem formatado, com informações relevantes;
Criatividade (Bônus);	15%	Deixe sua criatividade fluir;

Deverão ser entregues:

- Pacote ZIP ou RAR contendo:
 - Todos os arquivos do programa;
 - Relatório claro e bem formatado contendo as explicações das escolhas tomadas e que deixe claro quem ficou responsável por qual parte do trabalho;

Entrega:

- Dia 27/06;
- Via Teams da disciplina;