

INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO I
SCC0221

SISTEMA DE GERENCIAMENTO DE MÚSICAS

1 Introdução

A ascensão de serviços de streaming trouxe para o pessoal da tecnologia (e até mesmo para os fora desse mundo) uma facilidade imensa de ouvir músicas sem qualquer tipo de pirataria. É perceptível que, hoje em dia, as pessoas estão muito mais antenadas no mundo da música do que nunca!

Juliana é uma ávida amante do mundo musical e, como muita gente, utiliza de serviços de streaming para poder ouvir suas músicas favoritas. Recentemente, em meio a tantas playlists com tantas músicas que ela possui, Juliana acabou percebendo que precisava de uma maneira mais prática de conseguir organizá-las, visto que o serviço de streaming que ela utiliza não proporciona a ela uma interface muito boa para isso.

Dessa forma, honrando seu estereótipo de *nerd*, a garota pretende utilizar de seus conhecimentos em linguagem C, adquiridos na disciplina de Introdução à Computação da faculdade, para construir um gerenciador de playlists em formato CLI (*Command Line Interface*). Como ainda é iniciante, Juliana decidiu pedir sua ajuda para elaborar o projeto do gerenciador.

Estrutura básica de uma playlist

De acordo com a garota, é proposto que o gerenciador consiga armazenar, para a playlist, **seu nome**, o **número de músicas** nela armazenada e, obviamente, o **conjunto de músicas** que a representam.

Como o projeto ainda está no começo, Juliana lhe avisou que ela não pretende usar mais do que uma playlist ao mesmo tempo. Dessa forma, não há necessidade de se preocupar com armazenar mais de uma playlist em memória para trabalhar. Além disso, por questões de limitação de memória, **uma playlist só poderá conter no máximo 15 músicas**.

A ideia é que, no começo, a playlist esteja vazia e o usuário vá adicionando novas músicas informando suas características.

Estrutura básica de uma música

Juliana lhe disse que pretende armazenar, para cada uma, o **nome da música**, seu respectivo **artista**, e o **tempo de duração, em segundos** que, para ela, são as características mais importantes a serem exibidas na tela. Como ela já lhe disse anteriormente, o conjunto de músicas deve estar, de forma intrínseca, relacionada à playlist.

Sobre o gerenciador e suas funcionalidades

Como qualquer gerenciador, ele deve permitir ao usuário... *gerenciar*! Por isso, Juliana pede a você que o gerenciador a ser desenvolvido possua um total de 5 funcionalidades principais:

- **Comando 1:** Deve ser possível **adicionar uma nova música no final da playlist**, informando logo após as características que a fazem parte (nome, artista e tempo de duração, nessa ordem).
- **Comando 2:** Deve ser possível **exibir na tela todas as músicas que estão contidas na playlist**, com uma formatação específica (vide seção 5). A música que está sendo tocada no momento também deve ser destacada.
- **Comando 3:** Deve ser possível **avançar para a próxima música** da playlist.
- **Comando 4:** Deve ser possível **retroceder para a música anterior** da playlist.
- **Comando 5:** Por fim, o usuário deve ter a opção de **finalizar o programa**.

Nesse programa, cada comando é representado apenas por um número dado de entrada.

Está pronto para essa aventura? Pois Juliana está muito entusiasmada e ansiosa para começar! Ela disse que, como recompensa, lhe pagará um cookie do Kamzu quando possível (foi a Juliana que disse, não o autor do exercício).

2 Entrada

No começo da execução, será dado de entrada o nome da playlist. Logo após, serão dados de entrada indeterminados números relativos aos comandos já citados (até o programa receber o comando 5 para parar seu funcionamento). Cada comando sucederá dos argumentos que necessita (por exemplo, se dado o comando 1, logo em seguida deverão ser lidas as características da nova música na ordem determinada anteriormente).

- O valor de duração de uma música é um número inteiro sem sinal.
- Todos os nomes, artistas e nome de playlist são cadeias de caracteres (*strings*). Todas elas possuem apenas caracteres presentes na tabela ASCII e podem conter mais de uma palavra.
- Nesse trabalho **não** há necessidade de tratamento de *strings* ou de valores de entrada. Essa garantia é assegurada pelas entradas que serão proporcionadas.
- Não há necessidade de tratar a inserção de músicas repetidas. Isso é assegurado pelas entradas que serão proporcionadas.
- Note que o comando 1 (adicionar nova música) é o único que necessita de argumentos. Cada argumento será dado em uma linha diferente após a entrada do comando.

3 Saída

A única saída efetiva para o programa é a lista de músicas que a playlist possui através do comando 2 (exibir todas as músicas na tela) com uma formatação específica (vide seção 5), destacando a música que está sendo tocada no momento, ou alguma das mensagens de orientação citadas abaixo:

- A cada nova música adicionada com sucesso, a mensagem `'Musica <nome da musica> de <nome do artista> adicionada com sucesso.\n'`, onde os termos entre `<>` são relativos aos dados de entrada, deve ser exibida na tela.
- Caso a playlist esteja cheia, a mensagem `'Playlist cheia!\n'` deverá ser exibida ao usuário ao tentar adicionar uma nova música (e consequentemente exceder o número máximo) e a música não deve ser adicionada. O programa **não** deverá ser finalizado nessa situação.
- O comando 2 pode ser utilizado mais de uma vez na mesma execução (isto é, deve ser possível exibir a playlist na tela mais de uma vez durante a execução do programa).

4 Orientações para confecção do código

- Utilize de **dados estruturados** (*structs*) na construção da playlist e das músicas. Levante o seguinte questionamento: qual a melhor maneira de organizar os dados que possuo para trabalhar? Lembre-se que é possível colocar uma struct dentro de outra!
- Utilize de **alocação dinâmica** para armazenamento das strings e dos dados estruturados na memória. Para os demais dados, não há necessidade de armazenamento dinâmico (visto que não há nenhuma vantagem prática). **Sugere-se ao aluno que seja reaproveitada a função de leitura dinâmica de strings feita no exercício passado. Não assuma valores estáticos para as strings.**
- É responsabilidade do programador desalocar toda a memória alocada dinamicamente no final da execução do programa. Atente-se para não perder os ponteiros de referência para essas regiões de memória ao longo da elaboração do código. A ferramenta `valgrind` pode lhe ser útil.
- **Não utilize** mais memória do que o necessário!
- Utilize um **ponteiro da struct de música** para controlar o fluxo da troca de músicas e representar a atual música que está sendo tocada. O ponteiro deve sempre ser inicializado na primeira música da lista (quando houver) e não deverá apontar para nenhum endereço caso não haja músicas na playlist.
- **Modularize** seu código através da utilização devida de funções, conforme houver necessidade.
- Junto com a modularização, **documente** as principais partes e funções do seu código. Lembre-se: o excesso de comentários também é prejudicial.
- Confira a próxima seção para exemplos de como a formatação da saída padrão deve ser feita.

5 Exemplos de entrada e saída

- Exemplo 1

Entrada

```
1 Amora e Manjericao
2 1
3 On Ira
4 Zaz
5 178
6 1
7 Comfortably Numb
8 Pink Floyd
9 382
10 3
11 1
12 Magic - Original Mix
13 Nhato
14 339
15 2
16 5
```

Saída

```
1 Musica On Ira de Zaz adicionada com sucesso.
2 Musica Comfortably Numb de Pink Floyd adicionada com sucesso.
3 Musica Magic - Original Mix de Nhato adicionada com sucesso.
4 ---- Playlist: Amora e Manjericao ----
5 Total de musicas: 3
6
7 (1). 'On Ira'
8 Artista: Zaz
9 Duracao: 178 segundos
10
11 === NOW PLAYING ===
12 (2). 'Comfortably Numb'
13 Artista: Pink Floyd
14 Duracao: 382 segundos
15
16 (3). 'Magic - Original Mix'
17 Artista: Nhato
18 Duracao: 339 segundos
```

- Exemplo 2

Entrada

```
1 Oui oui je parle francais
2 1
3 Paris
4 Camille
5 226
6 1
7 Roi
8 Videoclub
9 222
10 3
```

```
11 1
12 Meme robe qu'hier
13 Pomme
14 166
15 1
16 A nous souvenirs
17 Trois Cafes Gourmands
18 192
19 3
20 2
21 3
22 1
23 Balance ton quoi
24 Angele
25 189
26 1
27 Cool Colorado
28 La Femme
29 231
30 1
31 Packshot
32 La Femme
33 176
34 3
35 4
36 3
37 2
38 5
```

Saída

```
1 Musica Paris de Camille adicionada com sucesso.
2 Musica Roi de Videoclub adicionada com sucesso.
3 Musica Meme robe qu'hier de Pomme adicionada com sucesso.
4 Musica A nous souvenirs de Trois Cafes Gourmands adicionada com sucesso.
5 ---- Playlist: Oui oui je parle francais ----
6 Total de musicas: 4
7
8 (1). 'Paris'
9 Artista: Camille
10 Duracao: 226 segundos
11
12 (2). 'Roi'
13 Artista: Videoclub
14 Duracao: 222 segundos
15
16 === NOW PLAYING ===
17 (3). 'Meme robe qu'hier'
18 Artista: Pomme
19 Duracao: 166 segundos
20
21 (4). 'A nous souvenirs'
22 Artista: Trois Cafes Gourmands
23 Duracao: 192 segundos
24
25 Musica Balance ton quoi de Angele adicionada com sucesso.
26 Musica Cool Colorado de La Femme adicionada com sucesso.
```

```
27 Musica Packshot de La Femme adicionada com sucesso.
28 ---- Playlist: Oui oui je parle francais ----
29 Total de musicas: 7
30
31 (1). 'Paris'
32 Artista: Camille
33 Duracao: 226 segundos
34
35 (2). 'Roi'
36 Artista: Videoclub
37 Duracao: 222 segundos
38
39 (3). 'Meme robe qu'hier'
40 Artista: Pomme
41 Duracao: 166 segundos
42
43 (4). 'A nous souvenirs'
44 Artista: Trois Cafes Gourmands
45 Duracao: 192 segundos
46
47 === NOW PLAYING ===
48 (5). 'Balance ton quoi'
49 Artista: Angele
50 Duracao: 189 segundos
51
52 (6). 'Cool Colorado'
53 Artista: La Femme
54 Duracao: 231 segundos
55
56 (7). 'Packshot'
57 Artista: La Femme
58 Duracao: 176 segundos
```

Bom trabalho! :)