

Aluno: Pedro Henrique Nunes de Assis

Matrícula: 19.2.8008

Professor: LUIZ CARLOS BAMBIRRA TORRES

Disciplina: CSI419/CSI107 - LINGUAGENS DE PROGRAMACAO - TURMAS 11

1 - “Música”, “Playlist” e “Catálogo” são naturalmente escolhidas quando se pensa em um streaming de músicas. No momento, a grande porção dos usuários está habituado com um aplicativo desse modelo e dentre as funções primárias para o funcionamento adequado se procura um catálogo, onde se possa escolher músicas para criar uma playlist. Música representa uma entidade fundamental pois suporta o título, artista e duração, enquanto os outros dois são formas de manipular as músicas direta ou indiretamente.

2 - Utilizando uma lista para armazenar as músicas (Catálogo e playlists) e dicionários para representar cada música. Os dados das músicas serão manipulados diretamente, dificultando e flexibilizando o código e dessa forma exigindo a necessidade de grandes alterações para implementação de novas funcionalidades.

2 - Implementação: Código no arquivo ‘2_Implementação Imperativa.zip’

3 - No paradigma orientado a objetos a solução será construída utilizando classes e objetos que irão encapsular as informações e as implementações realizadas. Utilizando a classe Música, Playlist e catálogo é possível solucionar o problema e tornar a aplicação flexível e limitando a visualização do usuário somente o necessário. Cada funcionalidade estará contida em sua classe, dessa forma caso algum formato de armazenamento seja alterado apenas a classe em questão precisará ser alterada. Além disso, para a adição de novas funcionalidades é possível criar blocos de código inicialmente apartados que não irão afetar a integridade do código pré-existente.

3 - Implementação: Código no arquivo ‘3_Implementação Orientada a Objetos.zip’

4 -

4.1 - Implementação Orientada a Objetos devido à sua flexibilidade.

4.2 - Implementação Orientada a Objetos, por fatores como a abstração, encapsulamento e baixo acoplamento e alta coesão. Nessa forma de implementação será exibido somente o que é relevante para o uso e também isolando as operações dentro de cada classe que por sua vez dependem minimamente das outras classes presentes no sistema possibilitando alterações sem grandes impactos em outras partes do sistema, trazendo robustez à aplicação.

4.3 - Na Implementação Imperativa devido ao tamanho do código e a agrupação de todas informações em um arquivo, facilitando o entendimento do fluxo. Porém com o crescimento da aplicação a situação passa a se inverter devido à desorganização ocasionada pelo agrupamento de códigos que são quase totalmente dependentes, assim tendendo a tornar a orientação a objetos uma escolha mais sensata em grande parte dos casos.