

## Especificação do Trabalho Prático

O trabalho prático desta disciplina consiste em desenvolver sistema computacional para a solução do problema descrito abaixo na linguagem de programação apresentada durante o curso: C++.

### 1. Descrição do Problema

Propomos um sistema capaz de inventariar as “mídias” de entretenimento – considerando músicas e podcasts.

Além da relação do usuário com a mídia e o nome da mesma, deseja-se registrar as fichas técnicas de cada uma, a saber:

- Para músicas: nome, produtores (cujo registro é opcional), gênero, duração e data de publicação;
- Para podcasts: produtores, gênero, duração, número da temporada (caso seja do tipo podcast), álbum (caso seja do tipo música), número do episódio, nome da mídia e data de publicação;

O gênero é a categoria usada pelo mercado para classificação de músicas e podcasts, por exemplo: Rock, Pop, Indie, K-POP, etc. Quando mais de um gênero é aplicável (ex.: Indie e Rock), o sistema se limitará a registrar apenas um (considerado principal).

Para facilitar um usuário a ouvir suas mídias preferidas, o sistema conta com a funcionalidade de favoritar mídias. Ao favoritar uma mídia ela é enviada para a “Lista de Favoritos” do usuário.

Com tais informações, espera-se que o sistema seja capaz de gerar para o usuário alguns relatórios úteis como, por exemplo:

- Listar os favoritos de todos os assinantes: Listar o tipo de mídia (música ou podcast), seu código, gênero e duração.
- Mídias por produtores: Para cada nome (destas pessoas), indicar as mídias relacionadas a elas (ex.: músicas e podcasts que Pablo Vittar participa como produtor);
- Estatísticas gerais:
  - Horas Consumidas: Horas totais somadas dos favoritos de todos os usuários;
  - Gênero mais ouvido: Listar o gênero mais favoritado da plataforma e qual o tempo para ouvir todas as músicas desse gênero.
  - Mídias por gênero: Listar para cada gênero a quantidade de mídias que o mesmo possui;
  - Top 10 Músicas: Listar as dez músicas mais favoritadas pelos assinantes;
  - Top 10 Produtores: Listar os dez produtores mais favoritados pelos assinantes;
- Realizar o backup da plataforma: Imprimir todas as informações dos itens cadastrados.

As próximas seções detalham o diagrama de classe do sistema, os dados de entrada e relatórios de saída esperados.

## 2. Diagrama de Classe

Nesta seção é apresentado o diagrama de classe da aplicação.

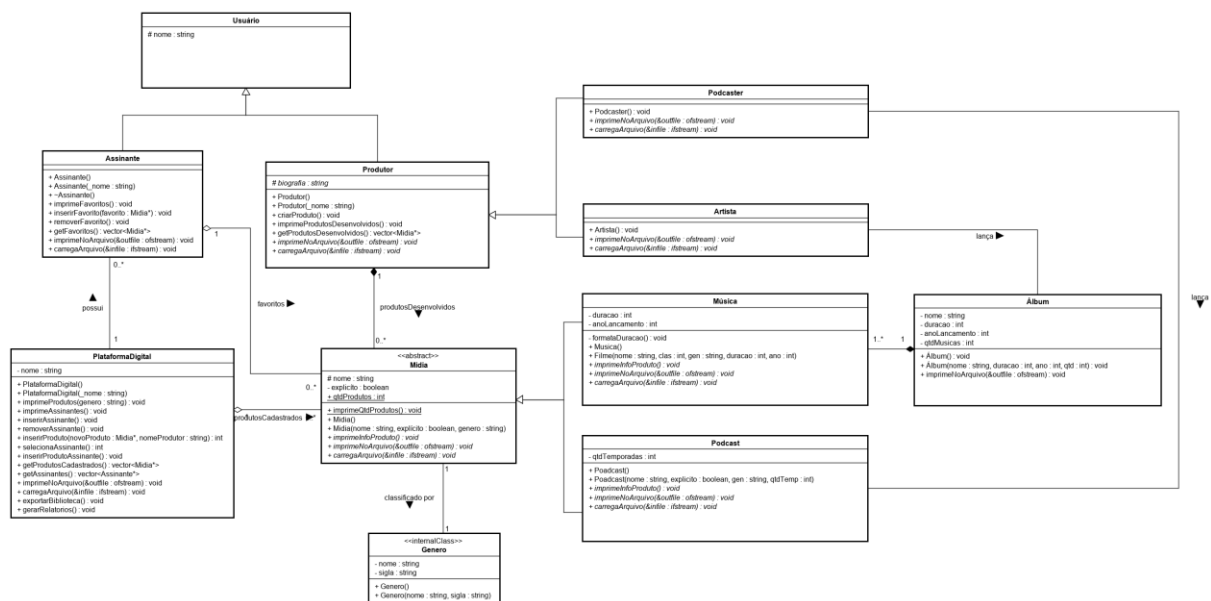


Diagrama de Classe – Spotify++ (link)

### 3. Formatos de entrada e saída

Os cadastros dos dados da plataforma são feitos em planilhas eletrônicas. Para o processamento destes dados e geração dos relatórios desejados, as planilhas serão exportadas para um formato de texto simples com valores separados por vírgulas, conhecidos como CSV (Comma Separated Values).

Para facilitar a leitura dos relatórios produzidos pelo programa, será feita a importação dos dados dos relatórios do formato CSV para planilha eletrônica. Portanto, seu programa deve ser capaz, além de ler dados neste formato, também gerar os relatórios em CSV.

Esta seção descreve os dados que estarão presentes em cada um dos arquivos de entrada e os dados que devem estar presentes em cada um dos arquivos de saída (relatórios). Para saber como estes dados serão formatados, verifique os arquivos de exemplo disponibilizados juntamente com esta descrição.

É muito importante que o programa siga os padrões de formatação prescritos, pois do contrário pode apresentar erro na leitura ou diferenças nos relatórios durante a correção automatizada dos trabalhos (vide Seção 4). Note que tanto os arquivos de entrada quanto os de saída possuem linhas cabeçalhos que devem ser levadas em consideração (ou seja, descartadas durante a leitura das planilhas de entrada e inseridas durante a escrita dos relatórios de saída).

## 3.1 Entrada de dados

São quatro arquivos de entrada de dados:

- Cadastro de gêneros;
- Cadastro de usuários;
- Inventário de mídias;
- Mídias favoritas.

Os nomes dos arquivos são especificados durante a execução do programa (vide Seção 3). Abaixo encontra-se especificada a ordem que os dados devem aparecer em cada um destes arquivos:

### Cadastro de Gêneros:

<sigla>;<nome>

Ambos podem ser lidos como texto (*string*)

### Cadastro de Usuários:

<código>;<tipo>;<nome>

O código é numérico (inteiro), tipo um caractere e nome pode ser lido como texto.

### Inventário de Mídias:

<código>;<nome>;<tipo>;<produtores>;<duração>;<gênero>;<temporada>;<álbum>;<data de publicação>

Cada campo é preenchido como se segue:

- **Código:** numérico (inteiro);
- **Nome:** texto;
- **Tipo:** um caractere – ‘M’ para músicas e ‘P’ para Podcasts;
- **Produtores:** códigos dos usuários (vide cadastro de usuários) que são produtores principais da música ou podcast. Os códigos são separados por vírgula;

- **Duração:** texto, indicando a duração da mídia (**hh:mm:ss**);
- **Gênero:** sigla (vide cadastro de gêneros) referente ao gênero principal da mídia;
- **Temporada:** número da temporada. No caso de músicas, ficará vazio;
- **Álbum:** texto;
- **Data de Publicação:** texto, indicando a data de publicação da mídia (**dd/mm/yyyy**);

### Mídias Favoritas:

<código do assinante>;<mídias>

Código do assinante se refere ao código no qual o assinante foi cadastrado (vide cadastro de usuários). As mídias são os códigos das mídias que estão presentes na lista de favoritos do usuário, caso exista mais de uma mídia, separar os códigos por ','

## 3.2 Processamento e Escrita dos Relatórios

Lidos todos os dados, o programa deve criar os objetos em memória representando as informações contidas nos arquivos de entrada. Tais objetos devem estar ligados adequadamente, conforme a associação entre as classes de objetos.

Os relatórios gerados devem ser escritos em arquivos com os seguintes nomes:

Relatório	Nome dos Arquivos
Estatísticas (não-CSV)	1-estatisticas.txt
Mídias por Produtores	2-produtores.csv
Lista de favoritos	3-favoritos.csv
Backup	4-backup.txt

Abaixo encontra-se especificada a ordem que os dados devem aparecer em cada um destes arquivos:

### Estatísticas (não-CSV)

Horas Consumidas: <HC> minutos

Gênero mais ouvido: <G> - <HG> minutos

Mídias por Gênero:

<G>:<QG>

Top 10 Mídias:

<M>:<G>:<NF>

Top 10 Produtores:

<P>:<NF>

Este relatório é o único que não possui o formato CSV, mas texto simples. Os códigos acima devem ser substituídos pelos seguintes dados:

- <HC>: Soma das durações das mídias presentes nas listas de favoritos dos assinantes;
- <HG>: Somas das durações das mídias pertencentes ao gênero mais ouvido;
- <G>: Nome do Gênero;
- <QG>: Quantidade de mídias por gênero;
- <M>: Nome da mídia;
- <NF>: Número de vezes que aparece nas listas de favoritos dos assinantes;
- <P>: Nome do produtor.

### Mídias Por Produtores:

<nome do produtor>;<nomes das mídias>

Este relatório deve ser ordenado por nome do produtor, em ordem crescente. Caso algum produtor tenha participação em mais de uma mídia, a coluna de nomes das mídias deve separar os nomes com vírgula e espaço (" , "), apresentando-os também em ordem alfabética.

### Lista de Favoritos:

<código do assinante>;<tipo>;<código da mídia>;<gênero>;<duração>

Este relatório deve ser ordenado primeiro pelo código do assinante (crescente), seguido pelo tipo de mídia (decrecente) e, em caso de empate, pelo código da mídia (crescente). Tipo e gênero devem ser escritos por extenso (e não seus códigos/siglas).

### Backup

Usuários:

<código>;<nome>

Mídias:

<nome>;<tipo>;<produtores>;<duração>;<gênero>;<temporada>;<álbum>;<data de publicação>

## 3.3 Tratamento de Exceção

Leitura de dados de arquivos, formatação, etc. São fontes comuns de erros e exceções. Seu programa deve tratar **apenas** os seguintes tipos de erros:

- Erros de entrada e saída de dados como, por exemplo, o arquivo especificado não existir ou o programa não ter permissão para ler ou escrever em um arquivo. Nestes casos, o programa deve exibir a mensagem “Erro de I/O” (sem as aspas);
- Erro de formatação dos dados nos arquivos, ou seja, um valor formatado de forma incorreta nos arquivos de entrada (ex.: encontrado caractere onde espera-se um número), causando erro de *parsing* dos dados. Nestes casos, o programa deve exibir a mensagem “Erro de formatação” (sem as aspas);
- Inconsistência nos dados de entrada: por exemplo, uso de Código de gênero ou usuário inexistente nos respectivos cadastros, etc. Nestes casos, o programa deve exibir a mensagem “Inconsistências na entrada” (sem as aspas).

As mensagens acima devem ser impressas na tela e o programa deve ser terminado em seguida. Quaisquer outras situações de erros possíveis devem ser ignoradas. Pode-se assumir que nos testes feitos durante a avaliação dos trabalhos outros tipos de erros diferentes dos listados acima nunca acontecerão.

## 4. Execução

Seu programa deve ser executado especificando os nomes dos arquivos de entrada como opções de linha de comando, especificadas a seguir:

- -g <arquivo>: cadastro de gêneros;
- -u <arquivo>: cadastro de usuários;
- -m <arquivo>: inventário de mídias;
- -f <arquivo>: lista de favoritos.

Para executar seu programa lendo os arquivos `generos.csv`, `usuarios.csv`, `midias.csv` e `favoritos.csv` como arquivos de entrada, o comando seria:

```
./trabalho -g generos.csv -u usuarios.csv -m midias.csv -f favoritos.csv
```

**Importante:** as opções de execução podem ser passadas em qualquer ordem. Portanto, o comando

```
./trabalho -f favoritos.csv -u usuarios.csv -m midias.csv -g generos.csv
```

É equivalente a:

```
./trabalho -g generos.csv -u usuarios.csv -m midias.csv -f favoritos.csv
```

## 5. Condições de Entrega

O trabalho deve ser feito obrigatoriamente em dupla. A data limite de entrega do trabalho é **26/10/2019**.

Dados que existem várias versões do compilador de C++, fica determinado o uso das versões instaladas nas máquinas do **Labgrad** com versões de referência para o trabalho prático. Seu trabalho deve compilar e executar nas máquinas deste laboratório. Além disso, os arquivos de código-fonte devem estar codificados com Unicode (UTF-8) para evitar erros de compilação.

## 6. Entrega do Trabalho

O Código-fonte e o arquivo **Makefile** de sua solução deverão ser compactados e enviados por e-mail (anexo do e-mail) para a professora ([jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)). Serão aceitos trabalhos entregues até as 23h59 da data limite. O assunto do e-mail deverá ser o seguinte:

*PROG3 – Trab1 – Nomes dos Alunos*

Substituindo *Nomes dos Alunos* pelos nomes dos alunos do grupo, separado por vírgula.

Dada a quantidade de trabalhos que devem ser avaliados, a correção dos trabalhos passará primeiro por um processo de testes automáticos e, em seguida, por uma entrevista. Para que os testes automáticos funcionem, o arquivo compactado por e-mail deve ser no formato *zip* com o nome *trabalho.zip* e conter o arquivo de **Makefile** e o código-fonte.

O arquivo *.zip* deverá também incluir dois conjuntos de arquivos de entrada (*generos.csv*, *usuarios.csv*, *midias.csv* e *favoritos.csv*) que atendam os seguintes critérios:

- Não conter trechos iguais a outros arquivos de testes disponíveis no site;
- Conter o cadastro de pelo menos 5 gêneros, 30 usuários, 20 mídias e 10 favoritos.

Os arquivos de teste enviados poderão, a critério do professor, ser disponibilizados aos demais alunos como parte do script de testes. Atualizações do script serão divulgadas em sala de aula.

O arquivo enviado não deve conter nenhuma classe compilada. Os testes automáticos serão executados no diretório onde encontra-se o arquivo *Makefile*. O código-fonte pode ser organizado da forma que a dupla achar melhor, desde que o *Makefile* esteja adequado a estrutura.

### 6.1 Preparação e execução do script de testes

Será disponibilizado aos alunos um script para execução de alguns testes automáticos, sendo possível, portanto, garantir que o trabalho passa nesses testes antes de submetê-lo ao professor.

O script de teste funcionará somente em ambiente Linux e será testado no Labgrad. Recomenda-se fortemente que os testes finais do seu trabalho sejam feitos no Labgrad, pois as versões das ferramentas instaladas nas máquinas do laboratório serão consideradas como versões de referência para correção do trabalho.

## 7. Critérios de avaliação

Os trabalhos serão avaliados em duas etapas:

- Avaliação objetiva (com testes automáticos), e
- Entrevista

Critérios utilizados na avaliação subjetiva incluem (mas não estão limitados a):

- Uso dos princípios básicos da orientação a objetos, como encapsulamento, abstração e modularização;
- Legibilidade (nomes de variáveis bem escolhidos, Código bem formatado, uso de comentários quando necessário, etc);
- Consistência (utilização de um mesmo padrão de código);
- Eficiência (sem exageros, tentar evitar grandes desperdícios de recursos);
- Uso eficaz das funcionalidades oferecidas pela linguagem (coleções, for each, etc.);

## 8. Observações Finais

Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, frequentando as aulas ou acompanhando as novidades na página da disciplina na internet.