

// Questao 7 Atividade Manual TADs

//Luiz Gustavo De Almeida Lopes Turma 10A

```
class DocumentoMusical{
```

```
private:
```

```
    string nomeMusica;
```

```
    string nomeCompositor;
```

```
    string estiloMusical;
```

```
    string tipoDocumento;
```

```
public:
```

```
    DocumentoMusical();
```

```
    void cadastrar(string nM, string nC, string eM, string tD); // todos os valores entre  
    parenteses serao inseridos pelo usuario
```

```
    void mostrarDadosDocumento();
```

```
};
```

```
DocumentoMusical::DocumentoMusical(){
```

```
    //inicilizar os valores de cada documento musical
```

```
}
```

```
void DocumentoMusical::cadastrar(string nM, string nC, string eM, string tD){
```

```
    //cadastrar cada documento musical com base no que o usuario digitar para preencher
```

```
}
```

```
void DocumentoMusical::mostrarDadosDocumento(){
```

```
    // mostrar os dados de cada documento que possui o mesmo estilo inserido pelo usuario  
    pela entrada padrao
```

```
}
```

```

class DadosBandas{
    private:
        string nomeBanda;
        string instrumentoTocado;
        string estiloBanda;

    public:
        DadosBandas();
        void cadastrarBandas(string nB, string iT, string eB);
        void mostrarDadosBanda();

};

DadosBandas::DadosBandas(){
    //inicializar os valores do nome da banda e etc
}

void DadosBandas::cadastrarBandas(string nB, string iT, string eB){
    // cadastrar os dados de cada banda baseado nos dados inseridos pelo usuario
}

void DadosBandas::mostrarDadosBanda(){
    //mostrar os dados da banda baseado no estilo que o usuario inserio pela entrada padrao
}

class AcervoDocumentos{
    private:
        DocumentoMusical *documento;
        int qntDocumentos;

```

public:

AcervoDocumentos(int qntD); // construtor

~AcervoDocumentos(); // destrutor

bool registrarDocumentos(string nM, string nC, string eM, string tD);

bool consultarDocumentos(string eM);

};

AcervoDocumentos::AcervoDocumentos(int qntD){

// inicializar o vetor que conterà todos os documentos cadastrados pelo usuario

}

AcervoDocumentos::~~AcervoDocumentos(){

// remover da memoria os dados armazenados pelo vetor documentos e apagar o vetor

}

bool AcervoDocumentos::registrarDocumentos(string nM, string nC, string eM, string tD){

//registrar todos os documentos inseridos pelo usuario em um vetor documentos

}

bool AcervoDocumentos::consultarDocumentos(string eM){

// consultar no vetor documentos se ha algum deles com o estilo musical inserido pelo usuario, se houver, sera imprimido na tela

}

class AcervoBandas{

private:

DadosBandas \*banda;

int qntBandas;

public:

AcervoBandas(int qntB); // construtor

~AcervoBandas(); // destrutor

bool registrarBandas(string nB, string iT, string eB);

bool consultarBandas(string eB);

};

AcervoBandas::AcervoBandas(int qntB){

// inicializar o vetor que conterà as informacoes de cada banda cadastrada pelo usuario

}

AcervoBandas::~~AcervoBandas(){

// remover da memoria os dados armazenados pelo vetor bandas e apagar o vetor

}

bool AcervoBandas::registrarBandas(string nB, string iT, string eB){

//registrar todos as bandas inseridos pelo usuario em um vetor bandas

}

bool AcervoBandas::consultarBandas(string eB){

// consultar no vetor bandas se ha algum deles com o estilo musical inserido pelo usuario, se houver, sera imprimido na tela

}