

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
SCC0215 - ORGANIZAÇÃO DE ARQUIVOS
PROFA. RESPONSÁVEL: CRISTINA DUTRA DE AGUIAR CIFERRI

ARMAZENAMENTO DE REGISTROS DE FILMES EM ARQUIVO BINÁRIO

BRUNO BACELAR ABE	9292858
KAUE LOPES DE MORAIS	9277576
LUCAS ALEXANDRE SOARES	9293265
RAFAEL AUGUSTO MONTEIRO	9293095

SÃO CARLOS
01/05/2016

Índice

[Índice](#)

- [1. Introdução](#)
- [2. Estrutura de Dados](#)
- [3. Interface e funcionamento geral](#)
- [4. Geração do arquivo binário de dados](#)
- [5. Arquivo textual “moviedb.txt”](#)
- [6. Listagem e busca em arquivo binário](#)
- [7. Exemplos da interface](#)
- [8. Teste](#)

1. Introdução

O projeto consiste na implementação de um programa que gerencie registros em um arquivo binário. O programa gerenciador, implementado na linguagem C, carrega informações de filmes escritas em um arquivo textual, e as armazena em um arquivo binário para realizar operações de busca e listagem. Esta documentação descreve o funcionamento do programa, assim como algumas decisões de projetos tomadas pelo grupo (como a escolha da estrutura de dados para armazenar o filme) e também acompanha imagens da interface em execução e exemplos com casos de teste.

O programa possui um Makefile com diretivas `make all` e `make clean` para facilitar a compilação. Mais informação sobre os comandos Make disponíveis estão incluídas no arquivo `README.txt` na raiz do projeto.

O programa foi compilado e testado em Linux (Ubuntu 15.10, Arch Linux), Windows (8 e 10, compilado com Code::Blocks) e Mac(OS X El Captain 11.3). As imagens que exemplificam o funcionamento da interface foram retiradas de execuções em Linux(Ubuntu 15.10), sendo semelhantes em outros sistemas operacionais.

O código fonte do projeto encontra-se disponível no GitHub, que foi utilizado para facilitar o desenvolvimento do projeto em grupo. O endereço do repositório é <https://github.com/RafaelMonteiro95/Arquivos>

O diretório base do projeto contém os diretórios “include” e “src”, que contém headers e códigos-fonte, respectivamente, utilizados no programa. Ainda, o diretório principal contém um arquivo Makefile, com diretivas de compilação e execução, um arquivo `README.md`, que descreve as operações disponíveis no Makefile, e um arquivo “`moviedb.txt`”, que será abordado em uma sessão específica. Quando compilado via Makefile, é criado um diretório “build”, que contém o executável do projeto, “`trab1_filmes`”. Durante a execução, é criado um arquivo “`dados.bin`” e é possível realizar a criação de um arquivo “`exemplo.txt`”. Ambos arquivos são abordados na sessão “4. Geração do arquivo binário de dados”.

Para compilar:

Make all

Para rodar:

Make run (execução simples)

Make run ARGS=exemplo (abordado na sessão 5)

Make run debug=1 (executa o programa com valgrind, verifica memory leaks)

2. Estrutura de Dados

O primeiro passo da implementação do projeto foi definir os campos e tipos de dados a serem utilizados na estrutura do registro. A Tabela X mostra os campos e seus respectivos tipos e tamanhos.

Campo	Tipo	Tamanho	Descrição
idFilme	inteiro	4 bytes	Chave primária de busca
anoLancamento	inteiro	4 bytes	Ano de lançamento do filme
duracaoFilme	inteiro	4 bytes	Duração em minutos do filme
generoFilmes	string	Variável	Gênero(s) do filme
tituloFilme	string	Variável	Título do filme
descFilme	string	Variável	Síнопse
producao	string	Variável	País(es) de produção

Tabela 1 - Campos do registro

Todos os campos de tamanho variável são inseridos depois dos campos de tamanho fixo para facilitar leitura/escrita. Utilizamos delimitadores de campo e de registro para determinar o final de cada campo e de registro. Os delimitadores utilizados foram, respectivamente, '|' e '#'. Apesar de utilizarmos como delimitadores caracteres existentes, não realizamos nenhum tipo de verificação nas strings pois assumimos que não existem esses dois caracteres em nenhum dos campos utilizados na aplicação.

A estrutura contém as funções básicas de inicialização e destruição, setters, getters e impressão. A estrutura segue a ideia de encapsulamento com o intuito de evitar memory leaks, pois tudo que é alocado dentro da estrutura é desalocado dentro da estrutura; e evitar problemas de segurança, impedindo acesso direto aos dados dentro da estrutura.

O tipo de cada campo foi decidido de forma simples: numeros são campos inteiros de tamanho fixo e textos são campos strings de tamanho variável. Os campos numéricos não possuem grandes variações, podendo ser armazenados em campos de tamanho fixo, enquanto os campos textuais poderiam variar bastante, então optou-se por utilizar campos de tamanho variável para tais.

Deve-se notar que apenas o campo idFilme é único, pois é utilizado como chave primária de busca. Os outros campos podem conter valores repetidos.

3. Interface e funcionamento geral

Definida a estrutura de dados básica para o projeto, definiu-se o funcionamento básica do programa: carrega-se uma lista de filmes previamente escrita em um arquivo textual “moviedb.txt” e, utilizando os dados lidos, preenche-se um arquivo binario “dados.bin”. Em seguida, o programa exibe um menu com opções para o usuário, que pode escolher entre as operações:

1. Gerar um novo arquivo de dados binario;
2. Exibir, um a um, os registros armazenados no arquivo binario;
3. Realizar uma busca por chave primária no arquivo binario.
0. Sair do programa

Caso nenhuma opção válida seja selecionada, o programa exibe uma mensagem informando o usuário que uma opção invalida foi escolhida e retorna ao menu de seleção de operação.

Caso a primeira opção seja selecionada, o programa informa o usuário que um novo arquivo binario está sendo gerado, gera um novo arquivo, exibe uma mensagem de confirmação e retorna ao menu de seleção de opções

Caso a segunda opção seja selecionada, o programa exibirá, um a um, os registros armazenados no arquivo binário de dados. Ainda, será exibida uma mensagem solicitando que o usuário pressione enter para continuar a execução do programa (“Pressione Enter para o proximo registro” ou “Pressione enter para voltar ao menu inicial”, caso aquele seja o ultimo registro a ser exibido).

Caso a terceira opção seja selecionada, o programa solicitará ao usuário um ID para a busca. Em seguida, será feita uma busca sequencial pelo ID informado, que exibirá o registro encontrado, caso ele exista, ou exibirá uma mensagem informando o usuario que o registro não foi encontrado, caso ele não exista.

Caso a opção zero seja selecionada, o programa libera a memória utilizada, fecha os arquivos e finaliza a execução.

A escolha de uma opção específica para finalizar a execução foi feita para facilitar a interação do usuário com o programa, evitando que a execução seja finalizada por uma entrada não intencional pelo usuário (como pressionar enter sem querer, por exemplo).

4. Geração do arquivo binário de dados

Para a criação dos registros e inserção no arquivo binário foi decidido o uso de um arquivo texto pré-definido. O arquivo contém uma lista com os dados de 100 filmes, que são transformados em registros e inseridos um a um no arquivo binário de dados sempre que é necessário gerar um novo arquivo.

O arquivo de texto foi escolhido para que não haja a necessidade de inserir os registros a cada execução do programa de forma manual, mas que ainda exista facilidade em manipular os dados fora da execução do programa, tornando mais simples o entendimento do projeto.

Ao iniciar a execução do programa, um arquivo binário é gerado automaticamente com base no arquivo texto. Isso ocorre pois é necessário ter um arquivo preparado antes de utilizar as funcionalidades de exibição e busca. Ainda, é possível solicitar durante a execução do programa que o arquivo binário seja gerado novamente, atendendo aos requisitos do projeto.

A geração do arquivo segue a ordem::

1. Abrem-se os arquivos necessários no início da execução do programa (“moviedb.txt”, arquivo de texto para referência da geração, e “dados.bin”, arquivo binário utilizado em busca e exibição de registros)
2. Gera-se uma lista com 100 valores aleatórios de 1 a 100 (Estes valores serão os IDs dos registros carregados)
3. Escolhe-se, de forma aleatória, um registro no arquivo de texto que ainda não foi inserido no arquivo binário. Associa-se um dos valores aleatórios gerados anteriormente à esse registro (o valor escolhido será o ID do registro)
4. Repete-se o processo acima (passo 3) até que todos 100 os registros tenham sido inseridos no arquivo binário.

O procedimento acima, em conjunto com a utilização da data e hora atual do computador como semente de geração de numeros aleatórios, garantem a aleatoriedade do arquivo binário sempre que ele é gerado novamente (tanto em execuções diferentes quanto numa mesma execução).

Ainda, para facilitar a visualização dos dados armazenados no arquivo de texto, é possível executar o programa com um argumento especial “exemplo”, que escreve de forma textual, em um arquivo chamado “exemplo.txt”, todos os dados escritos no arquivo binário. Isso permite visualizar melhor como os dados são armazenados, facilitando o entendimento do arquivo binario.

Por exemplo, em uma execução do programa com o comando:

“make run ARGS=exemplo”

O seguinte arquivo foi gerado:

```
rafael@bebequingue:~/grad/arqs/t1/Arquivos/trab1_filmes$ make run ARGS=exemplo
./build/trab1_filmes exemplo
Armazenamento de registros de filmes
Abrindo os arquivos necessarios...Done!
Um arquivo "exemplo.txt" sera gerado nessa execucao
Inserindo os registros de moviedb.txt para dados.bin...Done!

Escolha uma das opções abaixo:
1 - Gerar 100 registros aleatórios de filmes
2 - Exibir todos os dados, um a um
3 - Recuperar um registro pelo ID(Chave Primária)
0 - Sair do programa
0
rafael@bebequingue:~/grad/arqs/t1/Arquivos/trab1_filmes$ l
build/ dados.bin exemplo.txt include/ Makefile moviedb.txt README.md src/
```

Imagem 1: A execução do programa, com o comando “make run ARGS=exemplo” gerou um arquivo chamado “exemplo.txt”

Abrindo o arquivo “exemplo.txt”, temos os registros escritos da seguinte maneira:

```
831979117Alien|Terror|Um grupo de astronautas se ve preso com um Alien|EUA, Reino Unido|#7
supervisionar os Vingadores, o governo cria uma lei|EUA|#182006151The Departed|Policia|A
organizado em Boston. Billy Costigan (Leonardo DiCaprio), um jovem policial, recebe a miss
grupo comandado por Frank Costello (Jack Nicholson). |EUA, Hong Kong|#43199299Reservoir Dr
historia de um mafioso prestes a morrer|EUA|#242014129Kingsman|Aventura|Uma organizacao se
Warl and the Dying Girl|Drama|Um garoto eh obrigado a acompanhar uma mocca com cancer|EUA|#
```

Imagem 2: Recorte dos dados no arquivo “exemplo.txt”. A imagem retrata apenas um recorte, pois o arquivo armazena um grande volume de dados.

Examinando o primeiro registro escrito, podemos notar o formato utilizado para a escrita do arquivo binário de dados

idFilme	anoLancamento	duracaoFilme	tituloFilme	generoFilme	descFilme	producao
83	1979	117	Alien	Terror	Um grupo...	EUA,Reino Unido

Tabela 2: Exemplo da distribuição dos campos dos registros no arquivo binário.

Entre os campos de tamanho variável (tituloFilme, generoFilme, descFilme, producao) podemos notar os delimitadores de campo ‘|’. Após o ultimo delimitador ‘|’, podemos notar o delimitador de registro ‘#’.

5. Arquivo textual “moviedb.txt”

Para facilitar a implementação do projeto, foi utilizado um arquivo textual como fonte de dados dos registros que seriam armazenados no arquivo binário. O arquivo “moviedb.txt” deve estar presente na execução do projeto (no mesmo diretório que o arquivo Makefile, caso o comando “make run” seja usado para executar o programa, ou no diretório “/build”, caso o programa seja executado diretamente). Caso não esteja, o programa será finalizado mostrando a seguinte mensagem de erro:

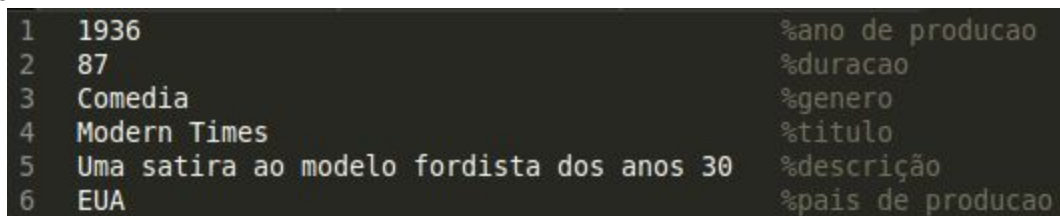
*Abrindo os arquivos necessarios...Erro!
Problema na abertura do arquivo "moviedb.txt"*

Esta verificação é feita para evitar que o programa seja inicializado sem o arquivo que fornece os dados dos filmes, o que impossibilitaria a geração do arquivo binário e comprometeria todas as funcionalidades do sistema.

O arquivo “moviedb.txt” contém todos os dados utilizados na estrutura de dados Filme, excluindo o campo Id, e cada campo é delimitado por uma quebra de linha (caractere \n). O arquivo deve conter 100 registros de filmes, ou ocorrerá um loop infinito durante a geração do arquivo binário. Não existe tratamento para casos onde o arquivo possui menos de 100 registros pois assume-se que não devem haver alterações nesse arquivo por parte do usuário.

Caso existam mais de 100 registros no arquivo, apenas os 100 primeiros serão utilizados. Isso ocorre pois a geração aleatória dos registros leva em consideração que apenas 100 registros devem ser utilizados e não é feito o mapeamento do arquivo para mais de 100 registros durante a geração do arquivo binário. Assim como no paragrafo acima, não devem haver alterações no arquivo “moviedb.txt,” então não seria necessário tratar casos onde o arquivo tenha mais de 100 registros.

Observando um trecho do arquivo utilizado no projeto, podemos notar a forma de separação dos campos por quebras de linha:



```
1 1936                                %ano de producao
2 87                                  %duracao
3 Comedia                            %genero
4 Modern Times                       %titulo
5 Uma satira ao modelo fordista dos anos 30 %descricao
6 EUA                                %pais de producao
```

Imagem 3: Arquivo “moviedb.txt”. A imagem acima exhibe o primeiro registro guardado no arquivo moviedb.txt, exemplificando a distribuição dos campos no arquivo. Note que os trechos iniciados por “%” são apenas comentários e não fazem parte do arquivo original.

6. Listagem e busca em arquivo binário

Após a geração de um arquivo binário devidamente preenchido, é possível realizar a listagem dos registros armazenados (Opção 2 da interface) ou buscar um registro por ID(Opção 3 da interface). As operações de busca e listagem são semelhantes, pois ambas percorrem o arquivo binário de forma sequencial.

A operação de listagem ocorre da seguinte maneira:

1. Um registro é carregado do arquivo binário para a memória principal
2. Esse registro é exibido ao usuário
3. É liberada a memória alocada para receber o registro lido

Este algoritmo se repete até que 100 registros tenham sido exibidos. A alocação e liberação de memória é feita em todas as etapas para garantir que haverá espaço disponível para armazenar um novo registro e para simplificar a implementação do programa.

A operação de busca ocorre da seguinte maneira:

1. Um ID é fornecido pelo usuário
2. Carrega-se um registro do arquivo binário
3. É realizada a comparação entre o ID fornecido pelo usuário e o ID do registro carregado
4. Caso sejam iguais, retorna-se o registro encontrado, que é exibido ao usuário
5. Caso sejam diferentes, libera-se a memória alocada e retorna ao passo 2

Este algoritmo também se repete até que 100 registros tenham sido lidos, ou caso o ID buscado tenha sido encontrado. Caso nenhum registro seja encontrado, exibe-se uma mensagem ao usuário que o ID solicitado não está no arquivo de dados. Novamente, o uso do carregamento e liberação de memória foi feito para simplificar a implementação do programa, já que não existe impacto significativo no desempenho final.

7. Exemplos da interface

```
Armazenamento de registros de filmes
Abrindo os arquivos necessarios...Done!
Inserindo os registros de moviedb.txt para dados.bin...Done!

Escolha uma das opções abaixo:
1 - Gerar 100 registros aleatórios de filmes
2 - Exibir todos os dados, um a um
3 - Recuperar um registro pelo ID(Chave Primária)
0 - Sair do programa
█
```

Imagem 4: Exemplo da interface inicial do programa

```
Escolha uma das opções abaixo:
1 - Gerar 100 registros aleatórios de filmes
2 - Exibir todos os dados, um a um
3 - Recuperar um registro pelo ID(Chave Primária)
0 - Sair do programa
1

Inserindo os registros de moviedb.txt para dados.bin...Done!
```

Imagem 5: Exemplo da interface quando a opção 1 é selecionada

```
Escolha uma das opções abaixo:
1 - Gerar 100 registros aleatórios de filmes
2 - Exibir todos os dados, um a um
3 - Recuperar um registro pelo ID(Chave Primária)
0 - Sair do programa
2

ID: 68
Lancamento: 2008
Duracao: 114min
Genero: Terror
Titulo: Deixa ela entrar
Descricao: Um garoto apaixona-se por uma menina, no minimo, estranha
Producao: Suecia

Pressione enter para o proximo filme
█
```

Imagem 6: Exemplo da interface quando a opção 2 é selecionada. A impressão dos registros exibe a mensagem “Pressione enter para o proximo filme” enquanto ainda existe um próximo registro a ser exibido

```
ID: 61
Lancamento: 2001
Duracao: 125min
Genero: Animação
Titulo: A Viagem de Chihiro
Descricao: Uma menina e sua família mudam-se de cidade, porém, coisas estranhas
acontecem no caminho
Producao: Japão

Pressione enter para voltar ao menu
```

Imagem 7: Exemplo da interface quando a opção 2 é selecionada. A impressão do último registro exibe a mensagem “Pressione enter para voltar ao menu”

```
Escolha uma das opções abaixo:
1 - Gerar 100 registros aleatórios de filmes
2 - Exibir todos os dados, um a um
3 - Recuperar um registro pelo ID(Chave Primária)
0 - Sair do programa
3

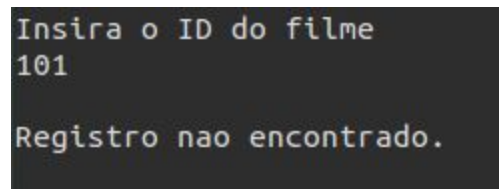
Insira o ID do filme
█
```

Imagem 8: Exemplo da interface quando a opção 3 é selecionada. O programa solicita um ID para realizar a busca

```
Insira o ID do filme
61

Registro encontrado!
ID: 61
Lancamento: 2001
Duracao: 125min
Genero: Animação
Titulo: A Viagem de Chihiro
Descricao: Uma menina e sua família mudam-se de cidade, porém, coisas estranhas
acontecem no caminho
Producao: Japão
```

Imagem 9: Exemplo da interface quando a opção 3 é selecionada. O programa realiza a busca sequencial no arquivo binário, encontra o registro com o ID informado e imprime os dados relativos ao registro encontrado.

A terminal window with a dark background and light-colored text. The text is as follows:

```
Insira o ID do filme  
101  
  
Registro nao encontrado.
```

Imagem 10: Exemplo da interface quando a opção 3 é selecionada. O programa realiza a busca por um ID que não existe no arquivo binário(ID 101). Uma mensagem informa o usuário que o ID buscado não foi encontrado.

8. Teste

As tabelas a seguir exemplificam o funcionamento do programa por meio de alguns casos de teste. As entradas e saídas estão definidas nas colunas da esquerda e da direita, respectivamente. Os valores entre “***” não são entradas e saídas reais, mas sim alertam o início e o final da execução do programa. Devido à aleatoriedade da geração do arquivo binário, é provável que ocorram variações nos dados apresentados.

Entrada	Saída
Execução iniciada	Armazenamento de registros de filmes Abrindo os arquivos necessarios...Done! Inserindo os registros de moviedb.txt para dados.bin...Done! Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa
3	Insira o ID do filme
10	Registro encontrado! ID: 10 Lancamento: 1989 Duracao: 128min Genero: Drama Titulo: Dead Poets Society Descricao: Um professor cria um grupo de poesias Producao: EUA Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa
0	***execução finalizada***

Tabela 3: Teste da interface. Início a execução, recupero o registro de ID 10 e finalizo a execução

Entrada	Saída
Execução iniciada	<p>Armazenamento de registros de filmes Abrindo os arquivos necessarios...Done! Inserindo os registros de moviedb.txt para dados.bin...Done!</p> <p>Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa</p>
2	<p>ID: 63 Lancamento: 1902 Duracao: 14min Genero: Aventura Titulo: Le Voyage Dans La Lune Descricao: Uma expedicao de corajosos homens vai a Lua Producao: Franca</p> <p>Pressione enter para o proximo filme</p>
enter	<p>ID: 75 Lancamento: 2001 Duracao: 101min Genero: Terror Titulo: The Others Descricao: Durante a 2 Guerra Mundial, Grace (Nicole Kidman) decide por se mudar, juntamente com seus dois filhos, para uma mansao isolada na ilha de Jersey, a fim de esperar que seu marido retorne da guerra. Producao: Espanha, EUA, Franca</p> <p>Pressione enter para o proximo filme</p>
enter	<p>ID: 33 Lancamento: 1989 Duracao: 128min Genero: Drama Titulo: Dead Poets Society Descricao: Um professor cria um grupo de poesias Producao: EUA</p> <p>Pressione enter para o proximo filme</p>

Tabela 4: Teste da interface. Inicio a execução, exibo os registros de forma sequencial.

As saídas seguintes serão semelhantes, variando apenas os dados dos registros. Após pressionar enter 96 vezes:

Entrada	Saída
enter	ID: 61 Lancamento: 2005 Duracao: 132min Genero: Acao Titulo: V for Vendetta Descricao: Uma inglaterra distopica tem um vigilante anarquista Producao: Alemanha, EUA, Inglaterra Pressione enter para voltar ao menu
enter	Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa
0	***execução finalizada***

Tabela 5: Teste de interface. Continuação do teste descrito na Tabela 3, exibo o ultimo registro e finalizo a execução do programa.

Entrada	Saída
Execução iniciada	Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa
3	Insira o ID do filme:
25	Registro encontrado! ID: 25 Lancamento: 2014 Duracao: 129min Genero: Aventura Titulo: Kingsman Descricao: Uma organizacao secreta, chamada Kingsman. Producao: Inglaterra

Tabela 6: Teste de interface. Exemplo de busca. O registro de ID 25 foi encontrado.

Entrada	Saída
Execução Iniciada com ARGS=exemplo	<p>Armazenamento de registros de filmes Abrindo os arquivos necessarios...Done! Um arquivo "exemplo.txt" sera gerado nessa execucao Inserindo os registros de moviedb.txt para dados.bin...Done!</p> <p>Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa</p>
3	Insira o ID do filme:
101	<p>Registro nao encontrado.</p> <p>Escolha uma das opções abaixo: 1 - Gerar 100 registros aleatórios de filmes 2 - Exibir todos os dados, um a um 3 - Recuperar um registro pelo ID(Chave Primária) 0 - Sair do programa</p>
0	***Execução finalizada***

Tabela 7: Teste de interface. Exemplo de busca por um ID não existente. Note também que foi utilizada a opção ARGS=exemplo, gerando um arquivo exemplificando o arquivo binário. Nessa ocasião, uma mensagem informa o usuário que um arquivo de exemplo será gerado.