



MAPA – Material de Avaliação Prática da Aprendizagem

Nome: Filipe Lino de Souza Bacof	R.A 1917836-5
Curso: Engenharia de Software	
Disciplina: Programação de Sistemas 1	
Valor da atividade: 3,50	Prazo: 24/11/2024

Instruções para Realização da Atividade

1. Revise seu arquivo antes do envio. Certifique-se de que é o arquivo correto, formato correto, se contempla todas as demandas da atividade, etc.
2. Após o envio não serão permitidas alterações.
3. Durante a disciplina, procure sanar suas dúvidas pontuais em relação ao conteúdo relacionado à atividade. Porém, não são permitidas correções parciais, ou seja, enviar para que o professor possa fazer uma avaliação prévia e retornar para que o aluno possa ajustar e enviar novamente. Isso não é permitido, pois descaracteriza o processo de avaliação.
4. Ao utilizar quaisquer materiais de pesquisa referencie conforme as normas da ABNT.

Em caso de dúvidas, entre em contato com seu Professor Mediador.

Bons estudos!

IMPORTANTE:

1. Acesse o link com um vídeo tutorial para ajudá-lo nesse processo de criação e desenvolvimento. O acesso deverá ser realizado através do fórum interativo: "Links das Aulas ao Vivo".
2. Disserte a respeito do tema, seguindo, como roteiro, os tópicos elencados.



3. A entrega deve ser feita exclusivamente usando o template de entrega da atividade MAPA, disponível no material da disciplina.
4. Antes de enviar a sua atividade, certifique-se de que respondeu a todas as perguntas e realize uma cuidadosa correção ortográfica.
5. Após o envio, não são permitidas alterações ou modificações. Logo, você tem apenas uma chance de enviar o arquivo corretamente. Revise bem antes de enviar!
6. Lembre-se de que evidências de cópias de materiais, incluindo de outros acadêmicos, sem as devidas referências, serão inquestionavelmente zeradas. As citações e as referências, mesmo que do livro da disciplina, devem ser realizadas de acordo com as normas da Instituição de Ensino.
7. Não são permitidas correções parciais no decorrer do módulo, ou seja, o famoso: “professor, veja se minha atividade está certa?”. Isso invalida o seu processo avaliativo. Lembre-se de que a interpretação da atividade também faz parte da avaliação.
8. Procure sanar as suas dúvidas junto à mediação em tempo hábil sobre o conteúdo exigido na atividade, de modo que consiga realizar a sua participação.
9. Atenção ao prazo de entrega. Evite o envio da atividade muito próximo do prazo. Você pode ter algum problema com a internet, o computador, o software etc., e os prazos não serão flexibilizados, mesmo em caso de comprovação.

Bons estudos!

Em caso de dúvidas, encaminhe mensagem ao seu professor mediador.

Resposta atividade MAPA:



Objetivo

O objetivo deste projeto é desenvolver um programa em Java que permita manipular um arquivo CSV contendo registros de doações de sangue. O programa permite:

1. **Leitura** de registros do arquivo CSV.
2. **Inserção** de novas doações no arquivo CSV.
3. **Remoção** de doações, com base no identificador da doação.

O usuário interage com o programa via um menu simples que oferece as três funcionalidades acima.

Antes de usar o Algoritmo

Crie dentro de **C:** uma pasta chamada **CSV** e dentro um arquivo **doacoes.csv** com o seguinte conteúdo:

```
1,Marcio dos Santos,521523159-57,2023-10-01,O+,150
2,Jose Eduardo,457896549-85,2023-12-01,O-,300
3,Adriana Jardim,123845678-12,2023-09-15,AB+,280
4,Carlos Roberto,341524981-35,2023-05-25,A+,380
5,Vinicius Cosatto,531765566-47,2023-04-14,A-,450
6,Felipe Neves,665977825-87,2023-10-17,O+,600
7,Alicia Flores,654895159-48,2023-07-22,AB+,650
8,Natalia Costa,977455987-12,2023-11-12,O-,720
9,Gertrudes Menezes,377551298-44,2023-09-11,O+,460
10,Henrique Martins,259886411-58,2023-18-10,O-,350
```

```

1  import java.io.BufferedReader;
2  import java.io.BufferedWriter;
3  import java.io.FileReader;
4  import java.io.FileWriter;
5  import java.io.IOException;
6  import java.util.ArrayList;
7  import java.util.List;
8  import java.util.Scanner;
9
10 /**
11  * Programa para manipulação de registros de doações de sangue em um arquivo
12  * CSV.
13  * O programa permite visualizar, inserir e remover registros.
14  *
15  * Funcionalidades:
16  * 1. Ler o conteúdo de um arquivo CSV.
17  * 2. Inserir uma nova doação no arquivo.
18  * 3. Remover uma doação pelo código.
19  *
20  * O programa oferece um menu para a seleção das funcionalidades.
21  *
22  * @author Filipe Bacof
23  */
24 public class ManipuladorCSV {
25
26     public static void main(String[] args) {
27         Scanner scanner = new Scanner(System.in);
28         String caminhoArquivo = null;
29         boolean arquivoValido = false;
30
31         do {
32             System.out.println("Informe o caminho do arquivo CSV (ex: C:/CSV/doacoes.csv): ");
33             caminhoArquivo = scanner.nextLine();
34             if (verificarArquivo(caminhoArquivo)) {
35                 System.out.println("Arquivo lido com sucesso!");
36                 arquivoValido = true;
37             } else {
38                 System.out.println("Falha em ler o arquivo, digite o caminho novamente.");
39             }
40         } while (!arquivoValido);
41
42         int opcao;
43         do {
44             System.out.println("\nMenu:");
45             System.out.println("1 - Visualizar registros");
46             System.out.println("2 - Inserir nova doação");
47             System.out.println("3 - Remover doação");
48             System.out.println("0 - Sair");
49             System.out.print("Selecione uma opção: ");
50             opcao = scanner.nextInt();
51             scanner.nextLine();
52
53             switch (opcao) {
54                 case 1:
55                     lerCSV(caminhoArquivo);
56                     break;
57                 case 2:
58                     inserirDoacao(caminhoArquivo, scanner);
59                     break;
60                 case 3:
61                     System.out.print("Informe o código da doação a ser removida: ");
62                     int codigoRemover = scanner.nextInt();
63                     removerDoacao(caminhoArquivo, codigoRemover);
64                     break;
65                 case 0:
66                     System.out.println("Saindo...");
67                     break;
68                 default:
69                     System.out.println("Opção inválida.");
70             }
71         } while (opcao != 0);
72     }
73
74     private boolean verificarArquivo(String caminho) {
75         try {
76             new FileReader(caminho);
77             return true;
78         } catch (IOException e) {
79             return false;
80         }
81     }
82
83     private void lerCSV(String caminho) {
84         try {
85             List<Doacao> doacoes = new ArrayList<>();
86             FileReader fr = new FileReader(caminho);
87             BufferedReader br = new BufferedReader(fr);
88             String linha;
89             while ((linha = br.readLine()) != null) {
90                 String[] dados = linha.split(",");
91                 Doacao doacao = new Doacao(dados[0], dados[1], dados[2], dados[3]);
92                 doacoes.add(doacao);
93             }
94             br.close();
95             System.out.println("Conteúdo do arquivo CSV:");
96             for (Doacao doacao : doacoes) {
97                 System.out.println(doacao);
98             }
99         } catch (IOException e) {
100             System.out.println("Erro ao ler o arquivo: " + e.getMessage());
101         }
102     }
103
104     private void inserirDoacao(String caminho, Scanner scanner) {
105         try {
106             FileWriter fw = new FileWriter(caminho, true);
107             BufferedWriter bw = new BufferedWriter(fw);
108             System.out.print("Informe o código da doação: ");
109             int codigo = scanner.nextInt();
110             scanner.nextLine();
111             System.out.print("Informe o nome do doador: ");
112             String nome = scanner.nextLine();
113             System.out.print("Informe o tipo de doação: ");
114             String tipo = scanner.nextLine();
115             System.out.print("Informe o valor da doação: ");
116             double valor = scanner.nextDouble();
117             scanner.nextLine();
118             Doacao doacao = new Doacao(codigo, nome, tipo, valor);
119             bw.write(doacao.toCSV());
120             bw.newLine();
121             bw.close();
122             System.out.println("Doação inserida com sucesso!");
123         } catch (IOException e) {
124             System.out.println("Erro ao inserir doação: " + e.getMessage());
125         }
126     }
127
128     private void removerDoacao(String caminho, int codigo) {
129         try {
130             List<Doacao> doacoes = new ArrayList<>();
131             FileReader fr = new FileReader(caminho);
132             BufferedReader br = new BufferedReader(fr);
133             String linha;
134             while ((linha = br.readLine()) != null) {
135                 String[] dados = linha.split(",");
136                 Doacao doacao = new Doacao(dados[0], dados[1], dados[2], dados[3]);
137                 if (doacao.getCodigo() != codigo) {
138                     doacoes.add(doacao);
139                 }
140             }
141             br.close();
142             FileWriter fw = new FileWriter(caminho, false);
143             BufferedWriter bw = new BufferedWriter(fw);
144             for (Doacao doacao : doacoes) {
145                 bw.write(doacao.toCSV());
146                 bw.newLine();
147             }
148             bw.close();
149             System.out.println("Doação removida com sucesso!");
150         } catch (IOException e) {
151             System.out.println("Erro ao remover doação: " + e.getMessage());
152         }
153     }
154 }

```

```
70     }
71     } while (opcao != 0);
72
73     scanner.close();
74 }
75
76 /**
77  * Verifica se o arquivo pode ser lido corretamente.
78  *
79  * @param caminhoArquivo Caminho do arquivo CSV.
80  * @return true se o arquivo for lido com sucesso, false caso contrário.
81  */
82 public static boolean verificarArquivo(String caminhoArquivo) {
83     try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
84         // Verifica se consegue ler a primeira linha do CSV pelo menos
85         if (br.readLine() != null) {
86             return true;
87         }
88     } catch (IOException e) {
89         return false;
90     }
91     return false;
92 }
93
94 /**
95  * Lê o conteúdo do arquivo CSV e exibe cada linha no console.
96  *
97  * @param caminhoArquivo Caminho do arquivo CSV.
98  */
99 public static void lerCSV(String caminhoArquivo) {
100     try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
101         String linha;
102         while ((linha = br.readLine()) != null) {
103             System.out.println(linha);
104         }
105     } catch (IOException e) {
106         System.err.println("Erro ao ler o arquivo: " + e.getMessage());
107     }
108 }
109
110 /**
111  * Insere uma nova doação no final do arquivo CSV.
112  *
113  * @param caminhoArquivo Caminho do arquivo CSV.
114  * @param scanner Scanner para entrada de dados do usuário.
115  */
116 public static void inserirDoacao(String caminhoArquivo, Scanner scanner) {
117     try {
118         boolean precisaDeNovaLinha = true;
119         try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
120             String ultimaLinha = null, linhaAtual;
121             while ((linhaAtual = br.readLine()) != null) {
122                 ultimaLinha = linhaAtual;
123             }
124             if (ultimaLinha != null && !ultimaLinha.isEmpty()) {
125                 precisaDeNovaLinha = false;
126             }
127         } catch (IOException e) {
128             System.err.println("Erro ao verificar o arquivo: " + e.getMessage());
129         }
130
131         try (BufferedWriter bw = new BufferedWriter(new FileWriter(caminhoArquivo, true))) {
132             if (!precisaDeNovaLinha) {
133                 bw.newLine();
134             }
135
136             System.out.println("Informe os dados da nova doação:");
137             System.out.print("Código: ");
138             int codigo = scanner.nextInt();
```

```
139 scanner.nextLine();
140
141 System.out.print("Nome: ");
142 String nome = scanner.nextLine();
143
144 System.out.print("CPF (111222333-44): ");
145 String cpf = scanner.nextLine();
146
147 System.out.print("Data de nascimento (AAAA-MM-DD): ");
148 String dataNascimento = scanner.nextLine();
149
150 System.out.print("Tipo sanguíneo (exemplo: A+): ");
151 String tipoSanguineo = scanner.nextLine();
152
153 System.out.print("Quantidade de sangue (ml): ");
154 int quantidade = scanner.nextInt();
155
156 String novaLinha = String.format("%d,%s,%s,%s,%s,%d", codigo, nome, cpf, dataNascimento, tipoSanguineo,
157 quantidade);
158 bw.write(novaLinha);
159 bw.newLine();
160 System.out.println("Doação inserida com sucesso!");
161 }
162 } catch (IOException e) {
163     System.err.println("Erro ao inserir a doação: " + e.getMessage());
164 }
165 }
166
167 /**
168  * Remove uma doação do arquivo CSV com base no código informado.
169  *
170  * @param caminhoArquivo Caminho do arquivo CSV.
171  * @param codigo Código da doação a ser removida.
172  */
173 public static void removerDoacao(String caminhoArquivo, int codigo) {
174     List<String> linhas = new ArrayList<>();
175     boolean removido = false;
176
177     try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
178         String linha;
179         while ((linha = br.readLine()) != null) {
180             if (!linha.startsWith(String.valueOf(codigo) + ",")) {
181                 linhas.add(linha);
182             } else {
183                 removido = true;
184             }
185         }
186     } catch (IOException e) {
187         System.err.println("Erro ao ler o arquivo: " + e.getMessage());
188         return;
189     }
190
191     // Reescreve o arquivo porém sem a linha que foi solicitada a remoção
192     if (removido) {
193         try (BufferedWriter bw = new BufferedWriter(new FileWriter(caminhoArquivo))) {
194             for (String l : linhas) {
195                 bw.write(l);
196                 bw.newLine();
197             }
198             System.out.println("Doação removida com sucesso!");
199         } catch (IOException e) {
200             System.err.println("Erro ao escrever no arquivo: " + e.getMessage());
201         }
202     } else {
203         System.out.println("Código não encontrado.");
204     }
205 }
206 }
207 }
```



Compilação e Execução

Para compilar e executar o projeto, siga os passos abaixo:

```
PS C:\Users\filip\OneDrive\Desktop\Mapa-Programacao-De-Sistemas-1> javac ManipuladorCSV.java
PS C:\Users\filip\OneDrive\Desktop\Mapa-Programacao-De-Sistemas-1> java ManipuladorCSV
Informe o caminho do arquivo CSV (ex: C:/CSV/doacoes.csv):
C:/CSV/doacoes2.csv
Falha em ler o arquivo, digite o caminho novamente.
Informe o caminho do arquivo CSV (ex: C:/CSV/doacoes.csv):
C:/CSV/doacoes.csv
Arquivo lido com sucesso!

Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: |
```

Comando para Compilação:

```
javac ManipuladorCSV.java
```

Comando para execução:

```
java ManipuladorCSV
```

- Certifique-se de ter o Java corretamente instalado e configurado no seu ambiente.
- O programa solicitará o caminho para o arquivo CSV. Se o caminho informado estiver incorreto, você poderá inseri-lo novamente.



Leitura dos Dados Iniciais

Ao iniciar o programa e informar o caminho correto do CSV, você pode escolher a opção 1 para ler os dados e exibi-los no terminal.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 1
1,Marcio dos Santos,521523159-57,2023-10-01,O+,150
2,Jose Eduardo,457896549-85,2023-12-01,O-,300
3,Adriana Jardim,123845678-12,2023-09-15,AB+,280
4,Carlos Roberto,341524981-35,2023-05-25,A+,380
5,Vinicius Cosatto,531765566-47,2023-04-14,A-,450
6,Felipe Neves,665977825-87,2023-10-17,O+,600
7,Alicia Flores,654895159-48,2023-07-22,AB+,650
8,Natalia Costa,977455987-12,2023-11-12,O-,720
9,Gertrudes Menezes,377551298-44,2023-09-11,O+,460
10,Henrique Martins,259886411-58,2023-18-10,O-,350

Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: |
```




Inserção de um Novo Registro

Para inserir uma nova doação, selecione a opção 2 no menu. O programa solicitará os dados necessários para inserir um novo registro no final do arquivo CSV.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 2
Informe os dados da nova doação:
Código: 11
Nome: Filipe Bacof
CPF (111222333-44): 521132650-44
Data de nascimento (AAAA-MM-DD): 1998-01-05
Tipo sanguíneo (exemplo: A+): O+
Quantidade de sangue (mL): 360
Doação inserida com sucesso!
```

Verificando o Arquivo Modificado

Após inserir o registro, você pode utilizar novamente a opção 1 para verificar a inclusão da nova linha no CSV.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 1
1,Marcio dos Santos,521523159-57,2023-10-01,O+,150
2,Jose Eduardo,457896549-85,2023-12-01,O-,300
3,Adriana Jardim,123845678-12,2023-09-15,AB+,280
4,Carlos Roberto,341524981-35,2023-05-25,A+,380
5,Vinicius Cosatto,531765566-47,2023-04-14,A-,450
6,Felipe Neves,665977825-87,2023-10-17,O+,600
7,Alicia Flores,654895159-48,2023-07-22,AB+,650
8,Natalia Costa,977455987-12,2023-11-12,O-,720
9,Gertrudes Menezes,377551298-44,2023-09-11,O+,460
10,Henrique Martins,259886411-58,2023-18-10,O-,350
11,Filipe Bacof,521132650-44,1998-01-05,O+,360
```



Remoção de um Registro

Para remover um registro, utilize a opção 3 e informe o identificador da doação. No exemplo abaixo, o primeiro registro foi removido.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 3
Informe o código da doação a ser removida: 1
Doação removida com sucesso!

Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 3
Informe o código da doação a ser removida: 36
Código não encontrado.
```

Verificando o Arquivo Após Remoção

Você pode utilizar a opção 1 novamente para verificar que o registro foi removido com sucesso.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 1
2,Jose Eduardo,457896549-85,2023-12-01,O-,300
3,Adriana Jardim,123845678-12,2023-09-15,AB+,280
4,Carlos Roberto,341524981-35,2023-05-25,A+,380
5,Vinicius Cosatto,531765566-47,2023-04-14,A-,450
6,Felipe Neves,665977825-87,2023-10-17,O+,600
7,Alicia Flores,654895159-48,2023-07-22,AB+,650
8,Natalia Costa,977455987-12,2023-11-12,O-,720
9,Gertrudes Menezes,377551298-44,2023-09-11,O+,460
10,Henrique Martins,259886411-58,2023-18-10,O-,350
11,Filipe Bacof,521132650-44,1998-01-05,O+,360
```



Opção Inválida e Saída do Programa

Caso o usuário insira uma opção inválida no menu, o programa exibirá uma mensagem de erro e solicitará uma nova escolha.

Para sair do programa, basta selecionar a opção 0.

```
Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 6
Opção inválida.

Menu:
1 - Visualizar registros
2 - Inserir nova doação
3 - Remover doação
0 - Sair
Selecione uma opção: 0
Saindo...
PS C:\Users\filip\OneDrive\Desktop\Mapa-Programacao-De-Sistemas-1> |
```

Considerações Finais

- Gravação em Tempo Real: Todas as alterações (inserções e remoções) são salvas imediatamente no arquivo CSV. Não há necessidade de salvar manualmente ao encerrar o programa.
- Tratamento de Exceções: Foram implementados tratamentos de exceções para garantir que o programa não seja interrompido em caso de:
 - Caminho inválido do CSV.
 - Remoção de um registro inexistente.
 - Leitura de dados corrompidos ou inexistentes.



Caso Necessário vou deixar abaixo o código para o arquivo

ManipuladorCSV.java

```
```java
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * Programa para manipulação de registros de doações de sangue em um arquivo
 * CSV.
 * O programa permite visualizar, inserir e remover registros.
 *
 * Funcionalidades:
 * 1. Ler o conteúdo de um arquivo CSV.
 * 2. Inserir uma nova doação no arquivo.
 * 3. Remover uma doação pelo código.
 *
 * O programa oferece um menu para a seleção das funcionalidades.
 *
 * @author Filipe Bacof
 */
public class ManipuladorCSV {

 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
 String caminhoArquivo = null;
 boolean arquivoValido = false;

 do {
 System.out.println("Informe o caminho do arquivo CSV (ex: C:/CSV/doacoes.csv): ");
 caminhoArquivo = scanner.nextLine();
 if (verificarArquivo(caminhoArquivo)) {
 System.out.println("Arquivo lido com sucesso!");
 arquivoValido = true;
 } else {
 System.out.println("Falha em ler o arquivo, digite o caminho novamente.");
 }
 } while (!arquivoValido);

 int opcao;
```



```
do {
 System.out.println("\nMenu:");
 System.out.println("1 - Visualizar registros");
 System.out.println("2 - Inserir nova doação");
 System.out.println("3 - Remover doação");
 System.out.println("0 - Sair");
 System.out.print("Selecione uma opção: ");
 opcao = scanner.nextInt();
 scanner.nextLine();

 switch (opcao) {
 case 1:
 lerCSV(caminhoArquivo);
 break;
 case 2:
 inserirDoacao(caminhoArquivo, scanner);
 break;
 case 3:
 System.out.print("Informe o código da doação a ser removida: ");
 int codigoRemover = scanner.nextInt();
 removerDoacao(caminhoArquivo, codigoRemover);
 break;
 case 0:
 System.out.println("Saindo...");
 break;
 default:
 System.out.println("Opção inválida.");
 }
} while (opcao != 0);

scanner.close();
}

/**
 * Verifica se o arquivo pode ser lido corretamente.
 *
 * @param caminhoArquivo Caminho do arquivo CSV.
 * @return true se o arquivo for lido com sucesso, false caso contrário.
 */
public static boolean verificarArquivo(String caminhoArquivo) {
 try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
 // Verifica se consegue ler a primeira linha do CSV pelo menos
 if (br.readLine() != null) {
 return true;
 }
 }
 } catch (IOException e) {
```



```
 return false;
 }
 return false;
}

/**
 * Lê o conteúdo do arquivo CSV e exibe cada linha no console.
 *
 * @param caminhoArquivo Caminho do arquivo CSV.
 */
public static void lerCSV(String caminhoArquivo) {
 try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
 String linha;
 while ((linha = br.readLine()) != null) {
 System.out.println(linha);
 }
 } catch (IOException e) {
 System.err.println("Erro ao ler o arquivo: " + e.getMessage());
 }
}

/**
 * Insere uma nova doação no final do arquivo CSV.
 *
 * @param caminhoArquivo Caminho do arquivo CSV.
 * @param scanner Scanner para entrada de dados do usuário.
 */
public static void inserirDoacao(String caminhoArquivo, Scanner scanner) {
 try {
 boolean precisaDeNovaLinha = true;
 try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
 String ultimaLinha = null, linhaAtual;
 while ((linhaAtual = br.readLine()) != null) {
 ultimaLinha = linhaAtual;
 }
 if (ultimaLinha != null && !ultimaLinha.isEmpty()) {
 precisaDeNovaLinha = false;
 }
 } catch (IOException e) {
 System.err.println("Erro ao verificar o arquivo: " + e.getMessage());
 }

 try (BufferedWriter bw = new BufferedWriter(new FileWriter(caminhoArquivo, true))) {
 if (!precisaDeNovaLinha) {
 bw.newLine();
 }
 }
 }
}
```



```
System.out.println("Informe os dados da nova doação:");
System.out.print("Código: ");
int codigo = scanner.nextInt();
scanner.nextLine();

System.out.print("Nome: ");
String nome = scanner.nextLine();

System.out.print("CPF (111222333-44): ");
String cpf = scanner.nextLine();

System.out.print("Data de nascimento (AAAA-MM-DD): ");
String dataNascimento = scanner.nextLine();

System.out.print("Tipo sanguíneo (exemplo: A+): ");
String tipoSanguineo = scanner.nextLine();


System.out.print("Quantidade de sangue (ml): ");
int quantidade = scanner.nextInt();

String novaLinha = String.format("%d,%s,%s,%s,%s,%d", codigo, nome, cpf,
dataNascimento, tipoSanguineo,
 quantidade);
bw.write(novaLinha);
bw.newLine();
System.out.println("Doação inserida com sucesso!");
}
} catch (IOException e) {
 System.err.println("Erro ao inserir a doação: " + e.getMessage());
}
}

/**
 * Remove uma doação do arquivo CSV com base no código informado.
 *
 * @param caminhoArquivo Caminho do arquivo CSV.
 * @param codigo Código da doação a ser removida.
 */
public static void removerDoacao(String caminhoArquivo, int codigo) {
 List<String> linhas = new ArrayList<>();
 boolean removido = false;

 try (BufferedReader br = new BufferedReader(new FileReader(caminhoArquivo))) {
 String linha;
 while ((linha = br.readLine()) != null) {
```





```
 if (!linha.startsWith(String.valueOf(codigo) + ",")) {
 linhas.add(linha);
 } else {
 removido = true;
 }
 }
} catch (IOException e) {
 System.err.println("Erro ao ler o arquivo: " + e.getMessage());
 return;
}

// Reescreve o arquivo porém sem a linha que foi solicitada a remoção
if (removido) {
 try (BufferedWriter bw = new BufferedWriter(new FileWriter(caminhoArquivo))) {
 for (String l : linhas) {
 bw.write(l);
 bw.newLine();
 }
 System.out.println("Doação removida com sucesso!");
 } catch (IOException e) {
 System.err.println("Erro ao escrever no arquivo: " + e.getMessage());
 }
} else {
 System.out.println("Código não encontrado.");
}
}
}
```