

Nome: Michel Angelo da Silva Portugal .

Matrícula: 202208826174

Campus: Duque de Caxias

Disciplina: RPG0016 - BackEnd sem banco não tem/ Sem 2/2023.

Objetivos da prática

Implementar persistência com base no middleware JDBC.

Utilizar o padrão DAO (Data Access Object) no manuseio de dados.

Implementar o mapeamento objeto-relacional em sistemas Java.

Criar sistemas cadastrais com persistência em banco relacional.

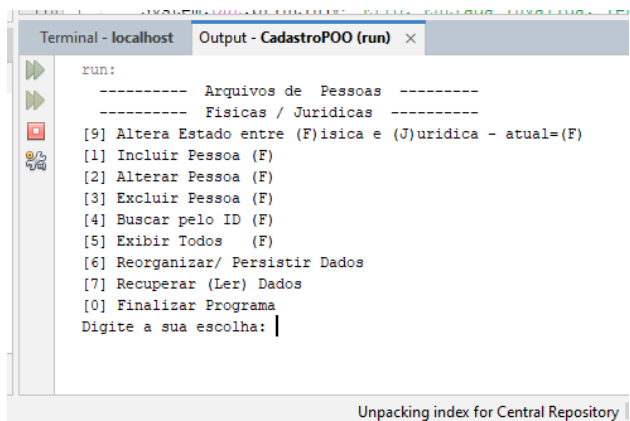
Link do Repositório:

<https://github.com/miaspe/Nivel-3---Back-end-Sem-Banco-N-o-Tem>

Código na Prática:

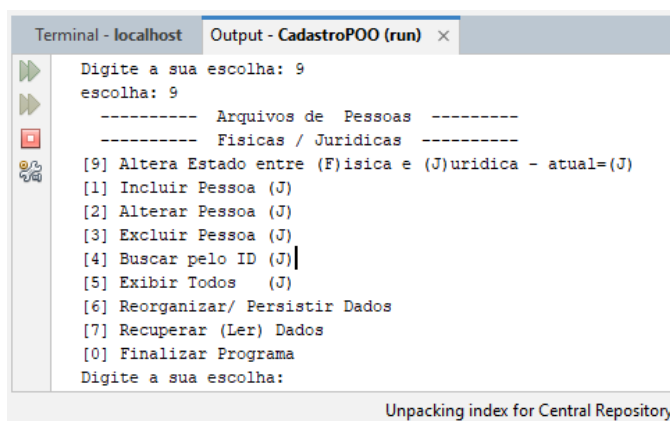
Abaixo apresentamos os códigos pertinentes às funcionalidades de um simples menu de: Cadastro, Alteração, Reinclusão e Listagem de CPFs e CNPJs em cadastro array fazendo persistência em um arquivo (.bin):

Começando pelo menu composto das opções pertinentes as funcionalidades do referido programa:



```
run:
----- Arquivos de Pessoas -----
----- Fisicas / Juridicas -----
[9] Altera Estado entre (F)isica e (J)uridica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: |
```

Fig1:



```
Terminal - localhost Output - CadastroPOO (run) x
Digite a sua escolha: 9
escolha: 9
----- Arquivos de Pessoas -----
----- Fisicas / Juridicas -----
[9] Altera Estado entre (F)isica e (J)uridica - atual=(J)
[1] Incluir Pessoa (J)
[2] Alterar Pessoa (J)
[3] Excluir Pessoa (J)
[4] Buscar pelo ID (J)|
[5] Exibir Todos (J)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha:
```

Fig:2

Como pode ser notado nas figuras de execução do programa, temos na figura1 o start inicial com a opção de cadastrar pessoas fisicas ou seja CPFs e Nomes ; E na fig 2 com a opção (9) selecionada o programa cambeia para pessoa juridica onde não mais cadastrará CPFs e sim CNPJs.

Nas figuras 1.1 e 2.1 podemos verificar a continuação da execução tanto do módulo pessoa física quanto jurídica:

```
Terminal - localhost  Output - CadastroPOO (run) x
ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: wwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

ID: 3
Nome: eeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44

Digite o nome da pessoa (ou 'sair' para encerrar): |
Cadast
```

Fig 1.1(pessoa física)

```
Terminal - localhost  Output - CadastroPOO (run) x
Nome: xxxxxxxxxxxxxxxxxxxx
CNPJ: 12345678933333

ID: 3
Nome: jjjjjjjjjjjjjjjjjjjjjj
CNPJ: 1234123412341234

ID: 4
Nome: vvvvvvvvvvvvvvvvvvvv
CNPJ: 12345678901234

ID: 5
Nome: bbbbbbbbbbbbbbbbbb
CNPJ: 98765432177755

ID: 6
Nome: nnnnnnnnnnnnnnnnnnnn
CNPJ: 987654987654987654
```

Fig 2.1(pessoa jurídica)

Agora vamos fazer a operação de (2)alterar pessoa física,repere que digitamos a opção e o programa nos avisa que existem 4 registros no total e o registro 2 que foi o registro escolhido pra alteração está disponível:

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Físicas / Jurídicas -----
[9] Altera Estado entre (F)ísica e (J)urídica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 2
escolha: 2
Digite o ID da pessoa que deseja ler: 2
maior ID :4
Pessoa encontrada:

ID: 2
Nome: wwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

Digite o novo Nome: kkkkkkkkkkkk
Digite o novo CPF: 66677788822
Digite a nova Idade: 22
Cadastre
```

Fig 1.3

Foi digitado o novo registro 2:

Como: Nome :kkkkk

CPF:66677788822

Idade:22

E agora vamos relistar os registros pra verificar se foi tudo bem mesmo na fig 1.4 e 1.5 abaixo:

```
ID: 2
Nome: wwwwwwwwwwwwwwwwwwwww
CPF: 12345678922
Idade: 22

Digite o novo Nome: kkkkkkkkkkkk
Digite o novo CPF: 66677788822
Digite a nova Idade: 22
Atualizado com sucesso!
```

Fig 1.4

```
Terminal - localhost  Output - CadastroPOO (run) x
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: kkkkkkkkkkkkkkkkkk
CPF: 66677788822
Idade: 22

ID: 3
Nome: eeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44

----- Arquivos de Pessoas -----
```

Fig 1.5

Na figura ao lado vemos a função listar (5) Exibir todos funcionando perfeitamente verificamos que o registro (wwwwww) foi atualizado pra (kkkkkkk) conforme o desejado, com um adendo de que no nosso programa a persistência ocorre a cada alteração não sendo necessária que seja pedida pelo usuário mas mesmo assim está disponível na opção (6) que reúne também a função reorganizar, vamos explica-la melhor abaixo:

Função (6) – Reorganizar / Persistir Dados:

Conforme já foi dito em nosso programa a persistência ocorre a cada alteração não sendo necessária que seja pedida pelo usuário. Assim o risco da necessidade de redigitação por erro do usuário após a inclusão de vários dados sucessivos pra somente no fim pedir ou esquecer de fazer a persistência fica afastado.

Agora pra função Reorganizar demonstraremos a função (3) Excluir, primeiro... conforme abaixo:

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Fisicas / Juridicas -----
[9] Altera Estado entre (F)isica e (J)uridica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 3
escolha: 3
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): 2
Pessoa excluída com sucesso!
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): sair
Arquivo salvo com sucesso!
```

Na listagem abaixo vemos que o arquivo já não mais apresenta o ID2 pois este foi excluído conforme figura acima:

```
Terminal - localhost  Output - CadastroPOO (run) x
Digite o ID da pessoa para excluir (ou 'sair' para encerrar): sair
Arquivo salvo com sucesso!
----- Arquivos de Pessoas -----
----- Fisicas / Juridicas -----
[9] Altera Estado entre (F)isica e (J)uridica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44
```

Agora vamos reincluir o ID2 com a opção (2) veja ao lado

```
Terminal - localhost  Output - CadastroPOO (run) x
----- Arquivos de Pessoas -----
----- Fisicas / Juridicas -----
[9] Altera Estado entre (F)isica e (J)uridica - atual=(F)
[1] Incluir Pessoa (F)
[2] Alterar Pessoa (F)
[3] Excluir Pessoa (F)
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 2
escolha: 2
Digite o ID da pessoa que deseja ler: 2
maior ID :4
Nenhum arquivo encontrado com o ID informado.
Deseja registrar arquivo pra posição 2 ? (S/N) s
Digite o novo Nome: ggggggggggg
Digite o novo CPF: 33344455522
Digite a nova Idade: 21
Atualizado com sucesso!
```

```
Terminal - localhost  Output - CadastroPOO (r
[4] Buscar pelo ID (F)
[5] Exibir Todos (F)
[6] Reorganizar/ Persistir Dados
[7] Recuperar (Ler) Dados
[0] Finalizar Programa
Digite a sua escolha: 5
escolha: 5
Pessoas cadastradas:

ID: 1
Nome: qqqqqqqqqqqqqqqqqqqqq
CPF: 12345678900
Idade: 11

ID: 2
Nome: ggggggggggg
CPF: 33344455522
Idade: 21

ID: 3
Nome: eeeeeeeeeeeeeeeeeeeeeee
CPF: 12345678933
Idade: 33

ID: 4
Nome: rrrrrrrrrrrrrrrrrrrrrrrr
CPF: 12345678944
Idade: 44
```

Agora listamos todos os registros com a opção 5 e verificamos o sucesso da operação: notando que o registro ID2 agora passou a ser (gggggggg , 33344455522, 21)

Opção (7) Recuperar (Ler) Dados:

Essa função não foi implementada pelo simples fato de que, como o salvamento é automático no arquivo (.bin), foi verificado que a função (5) Exibir Todos , acaba por desempenhar a mesma função.

Opção (0) Finalizar Programa. É auto explicativa.

Link do Repositório:

<https://github.com/miaspe/Nivel-3---Back-end-Sem-Banco-N-o-Tem>

Perguntas a serem respondidas:
Análise e Conclusão:

P: Quais as diferenças entre a persistência em arquivo e a persistência em banco de dados?

R: Em arquivo somente teremos o banco de dados(informações) localmente(aonde está a máquina) ao passo que em um sistema SGBD podemos hospedar os nossos DADOS e acessar remotamente se necessário, isso sem mencionar a chance reduzida de perda pela redundância do SGBD coisa que localmente temos infinitamente mais risco ,como, queima de máquina por evento externo qualquer erro de escrita e leitura por hardware, vírus e invasão, que podem culminar em corrupção do banco por falha ou negligência operacional tudo isso fica reduzido quando em nuvem.

P:Como o uso de operador lambda simplificou a impressão dos valores contidos nas entidades, nas versões mais recentes do Java?

R: simplifica a manipulação de dados, facilitando a obtenção de resultados de consultas.

P:Por que métodos acionados diretamente pelo método main, sem o uso de um objeto, precisam ser marcados como static?

R:Na verdade essa convenção do método foi posta pra facilitar a JVM não precisar instanciar uma classe ou objetificação o que consumiria recursos e dificultaria a versatilidade de chamada do método.