

Programa IT Academy – Processo Seletivo – Edição #15

Bruno Simm Alves

Sistemas de Informação

PUCRS

Etapa 2 – Explicação da solução

- Repositório da aplicação: <https://github.com/BrunoSimm/desafio-dell>
- Requisitos para executar o software:
 - [Java 17](#)
 - [Maven](#)
- Comando para compilar a aplicação:
 - Na raiz do projeto execute:
 - mvn clean
 - mvn clean install
 - mvn package
- Comando para executar:
 - Na raiz do projeto execute o SNAPSHOT gerado na compilação:
 - java -jar .\target\etapa2-1.0-SNAPSHOT.jar

1) Explicação da Solução

1.1) Decisões Arquiteturais

A aplicação foi organizada seguindo o padrão de camadas (Layers), utilizando-se de um padrão conhecido como Arquitetura Limpa (Clean Architecture de Robert Martin). A motivação central para o uso desta arquitetura é propiciar o isolamento do core de negócio da aplicação através do uso de camadas e separação de responsabilidades, evitando o acoplamento de camadas inferiores (que tendem a mudar com menos frequência) com camadas superiores.

Representação do modelo de arquitetura utilizado:

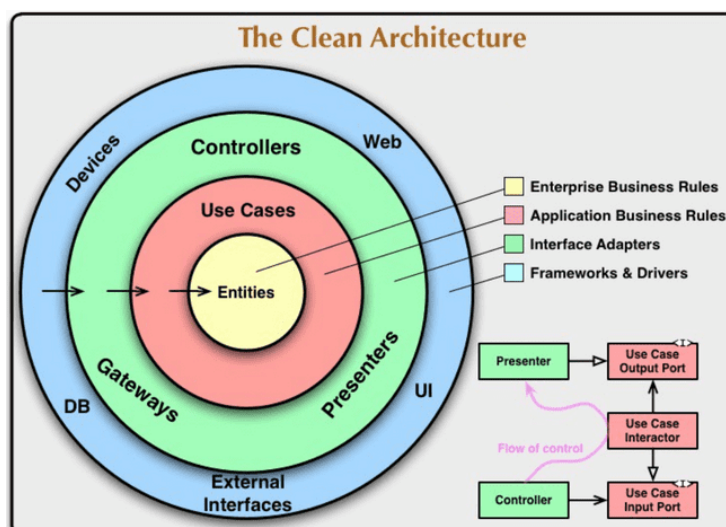


Figura 1 - Exemplo de uso da arquitetura limpa.

Neste projeto separamos a aplicação em 3 “grandes” camadas sendo elas de negócio, de aplicação e de adaptadores.

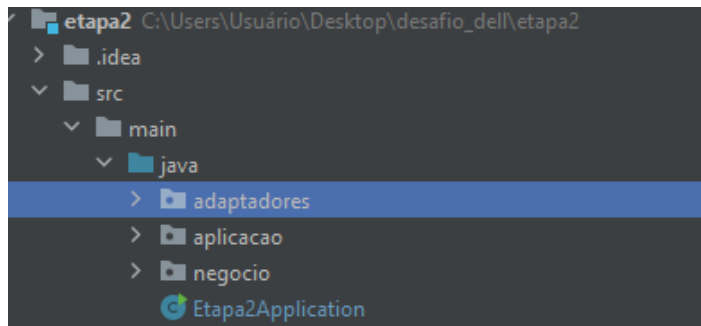


Figura 2 - Camadas do Projeto.

1.1.2) Camada de “Negócio”

É a camada que menos tende a mudar, onde as entidades e as regras de negócio se encontram. Esta camada não deve depender de nenhuma outra para seu funcionamento para evitar acoplamento com camadas superiores.

No projeto essa camada foi dividida em três pastas: entidades, repositórios e serviços. “Entidades” refere-se às classes de negócio Bolsa e Bolsista, onde foram definidos seus atributos e operações. “Repositórios” possuem as interfaces de acesso a dados para bolsas e bolsistas, estas interfaces permitem a implementação (na camada de adaptadores) através do “contrato” definido por elas. “Serviços” possuem funcionalidades comuns do sistema e fortemente associados aos casos de uso dos usuários, como por exemplo na classe “BolsasServico” no método “consultarMediaAnual”, onde é realizado o cálculo da média dos valores das bolsas de um determinado ano.

É importante ressaltar que classes desta camada *não devem utilizar classes de camadas superiores*. Dessa forma, as classes de serviço poderão apenas utilizar as interfaces de repositório definidas também nesta camada. Esta prática permite o baixo acoplamento e facilita, se for necessário, a mudança de implementação da classe concreta de repositório que será implementada na camada de “adaptadores”. Além disso, facilita a mudança da origem dos dados para um banco de dados ou outro tipo de arquivo, por exemplo.

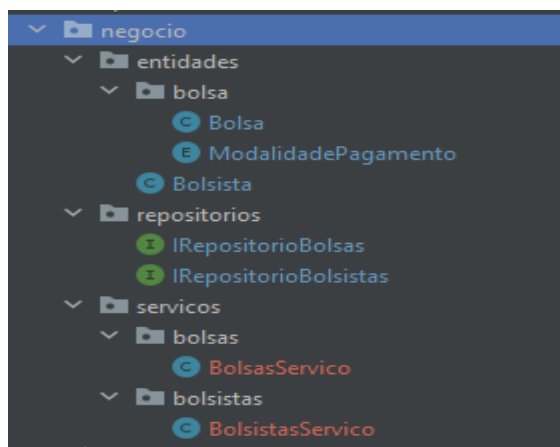


Figura 3 - Camada de Negócio.

1.1.3) Camada de “Aplicação”

A camada de aplicação possui a responsabilidade de fazer a comunicação entre a camada de negócio e a camada de adaptadores. Neste projeto, define os casos de uso apresentados nos requisitos do desafio.

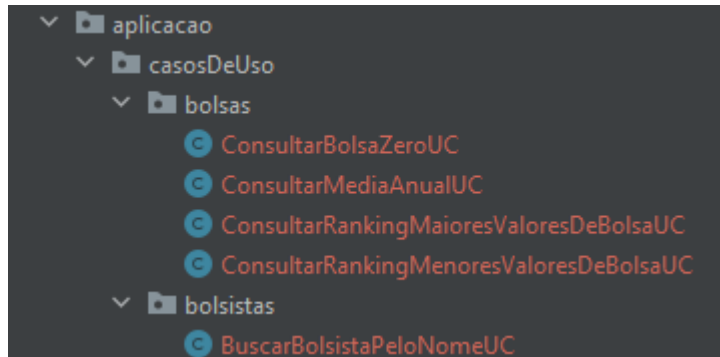


Figura 4 - Camada de Aplicação.

1.1.4) Camada de “Adaptadores”

A camada de adaptadores tem a missão de realizar a comunicação com o mundo exterior, no caso deste projeto com o arquivo “.csv”, que serviu como base de dados, e com o usuário, através do console. Esta camada é a que mais tende a mudar, pois não faz parte do “core” de negócio.

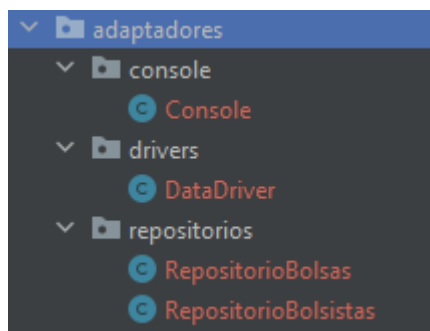


Figura 5 - Camada de Adaptadores.

Nesta implementação, a classe “Console” tem a responsabilidade de tratar toda interação com o usuário, exibindo menu, erros, inputs e resultados de consultas solicitadas. Neste contexto e por estar em uma camada superior, caso fosse necessário modificar a interface de interação do usuário, bastaria modificar esta classe e todo o restante da aplicação não necessitaria de grandes alterações.

As classes de “repositórios” possuem a finalidade de implementar a interface correspondente na camada de negócio (Ex: RepositorioBolsas implementa iRepositorioBolsas) e realizar o acesso e retorno dos dados da aplicação. Como estamos lendo um arquivo e armazenando em memória, cada repositório possui em seu contexto os objetos criados pela classe DataDriver. Entretanto, se alterarmos a forma de acesso a dados, como por exemplo para banco de dados, seria necessário modificar apenas a implementação dos métodos descritos na interface da camada de negócio e o restante da aplicação que utiliza estes repositórios continuaria funcionando conforme esperado.

A classe “*DataDriver*” possui a responsabilidade de ler o arquivo “.csv” e armazenar os dados coletados nos repositórios, também implementados nesta camada. Dessa forma, ela tem a responsabilidade de mapear uma linha do arquivo CSV em suas respectivas entidades da camada de negócio (“Bolsista” e “Bolsa”).

2) Demonstração da execução da aplicação:

2.1) Início da solução e apresentação das opções de menu disponíveis.

A aplicação possui interface gráfica para interação com usuário através do console de forma textual.

```
CAPES - Indicadores de Bolsas, Auxílios e Projetos da DED/CAPES 2013 a 2016,Bolsistas da Universidade Aberta do Brasil.

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
|
```

Figura 6 - Menu de opções do programa.

Neste console tive o cuidado de realizar verificações dos dados inseridos para evitar comportamentos inesperados ou falhas críticas que poderiam levar a interrupção da execução do programa. Além do menu de opções, busquei utilizar de programação defensiva durante todo software, visando o tratamento de erros esperados para melhorar a fluidez na interação com o console dos usuários.

```
CAPES - Indicadores de Bolsas, Auxílios e Projetos da DED/CAPES 2013 a 2016,Bolsistas da Universidade Aberta do Brasil.

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
opção invalida
Opção inválida, tente novamente.

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
|
```

Figura 7 - Erro de input de usuário tratado e notificado (em vermelho).

2.2) Consultar bolsa zero/Ano

Esta funcionalidade permite que o usuário digite um ano de sua escolha e então a aplicação irá buscar pelo bolsista zero (o primeiro bolsista do ano escolhido). Este bolsista

é escolhido comparando o mês de referência de cada bolsa.

```
CAPEs - Indicadores de Bolsas, Auxílios e Projetos da DED/CAPEs 2013 a 2016, Bolsistas da Universidade Aberta do Brasil.

ESCOLHA UMA DAS SEGUINTEs OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
|
DIGITE UM ANO PARA BUSCAR: 2013

BOLSISTA ZERO DE ANO 2013:
Nome: JOSE QUERGINALDO BEZERRA
CPF: ***.283.554-**
Entidade de Ensino: UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
Valor da Bolsa: R$ 1400,00

ESCOLHA UMA DAS SEGUINTEs OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
```

Figura 8 - Resultado de busca do Bolsista Zero do ano de 2013.

Caso o usuário digite um ano que não possua dados ele será notificado e poderá escolher uma nova opção no menu inicial.

```
CAPEs - Indicadores de Bolsas, Auxílios e Projetos da DED/CAPEs 2013 a 2016, Bolsistas da Universidade Aberta do Brasil.

ESCOLHA UMA DAS SEGUINTEs OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
|
DIGITE UM ANO PARA BUSCAR: 2023
Nenhuma bolsa encontrada neste ano!

ESCOLHA UMA DAS SEGUINTEs OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa
|
```

Figura 9 - Ano escolhido sem dados cadastrados.

2.3) Consultar bolsista por nome (Codificar nomes)

Esta funcionalidade permite ao usuário buscar por um bolsista através do seu nome (completo ou parcial). Deve-se então apresentar o nome codificado, ano, entidade de ensino e valor da bolsa deste bolsista.

*Uma vez que um bolsista pode possuir mais de uma bolsa, é importante ressaltar que se fez necessária uma tomada de decisão, de certa forma “arbitrária”, durante a escolha da bolsa que será apresentada ao usuário depois da busca. **Em vista disso, optei por escolher a bolsa mais recente (ano mais recente) de cada bolsista para exibição nesta funcionalidade.***

```
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa

2
Digite o nome de um bolsista: Marly Sacramento
BOLSISTA ENCONTRADO(A):
Nome: NHSBNBD PUOFNBSDBT ZMSBP
Ano: 2014
Entidade de Ensino: UNIVERSIDADE DE BRASILIA
Valor da Bolsa: R$ 765,00

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
```

Figura 10 - Busca pelo nome, resultando na bolsa mais atual e o nome do bolsista codificado.

A **codificação do nome do bolsista** foi realizada seguindo as orientações dispostas na especificação do desafio. Dessa forma, segui as seguintes regras para realizar a codificação:

Passo 1: Inverter primeiro e último caractere do nome (Exemplo: BRUNO -> ORUNB).

Passo 2: Realizar a troca de posições entre caracteres do nome. Por exemplo, troca-se o 1 elemento com o último, o 2 com o penúltimo, o 3 com o antepenúltimo e assim sucessivamente (Exemplo: ORUNB -> BNURO). Se o tamanho do nome for ímpar então o *caractere central* não será trocado com nenhum outro.

* Caso o tamanho do nome for menor que 4 caracteres, então este passo não será executado, pois resultaria no nome original do bolsista.

Passo 3: Em cada letra deve-se "mover" a letra para o próximo caractere do alfabeto. Para implementar esta etapa utilize a tabela ASCII, onde bastaria somar 1 em cada caractere do nome para obter o resultado desejado, como por exemplo A (65) + 1 = 66 (B). Caso a letra for "Z" (90) então transformo o caractere em "A", visto que o "Z" é a última letra da tabela ASCII.

* Em caso de espaços não realizo a substituição do caractere, como pode ser visto na imagem acima.

A implementação desta codificação se encontra na classe "BolsistasServico" no método "codificarNome()".

2.4) Consultar média anual

A partir de um ano escolhido pelo usuário o programa irá exibir a média dos valores das bolsas deste ano.

```
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa

3
DIGITE UM ANO PARA BUSCAR: 2014

MÉDIA ANUAL DAS BOLSAS DE 2014: R$ 936,24
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
```

Figura 11 - Média Anual dos Valores das Bolsas do ano de 2014.

Caso não existam bolsas no ano informado o usuário é notificado e retorna ao menu principal.

```
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa

3
DIGITE UM ANO PARA BUSCAR: 2010

NÃO FORAM ENCONTRADAS BOLSAS NESTE ANO!

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
```

Figura 12 - Consulta de Média Anual de ano sem dados.

2.5) Ranking de valores de bolsa

Esta funcionalidade irá listar os 3 alunos com maiores bolsas e os 3 alunos com menores bolsas.

```
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa

TRÊS ALUNOS COM AS MAIORES BOLSAS:
1 = Nome: LIAMARA SCORTEGAGNA | Ano: 2016 | Entidade de Ensino: UNIVERSIDADE FEDERAL DE JUIZ DE FORA | Valor da Bolsa: R$ 1500,00
2 = Nome: CESAR ALFREDO CARDOSO | Ano: 2016 | Entidade de Ensino: UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ | Valor da Bolsa: R$ 1500,00
3 = Nome: IZABEL PATRICIA MEISTER | Ano: 2016 | Entidade de Ensino: UNIVERSIDADE FEDERAL DE SÃO PAULO | Valor da Bolsa: R$ 1500,00

TRÊS ALUNOS COM AS MENORES BOLSAS:
1 = Nome: ROSILENE DE LIMA | Ano: 2013 | Entidade de Ensino: UNIVERSIDADE ESTADUAL DE MARINGÁ | Valor da Bolsa: R$ 765,00
2 = Nome: GLAUCIA ANK GUARINO | Ano: 2013 | Entidade de Ensino: UNIVERSIDADE FEDERAL DO RIO DE JANEIRO | Valor da Bolsa: R$ 765,00
3 = Nome: NATÁLIA NOGUEIRA SARAIVA | Ano: 2013 | Entidade de Ensino: UNIVERSIDADE ESTADUAL DO CEARÁ | Valor da Bolsa: R$ 765,00

ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
```

Figura 13 - Consulta do Ranking de Valores de Bolsa.

2.6) Terminar Programa

Esta funcionalidade encerra a execução do programa.

```
ESCOLHA UMA DAS SEGUINTE OPÇÕES DE CONSULTA:
1 -> Consultar Bolsa Zero
2 -> Consultar Bolsista por Nome
3 -> Consultar Média Anual dos Valores das Bolsas
4 -> Consultar Ranking de Valores de Bolsa
5 -> Terminar Programa

5
Encerrando programa...
Obrigado!
```

Figura 14 - Terminar programa.

3. Auto avaliação

De forma geral estou satisfeito com meu desempenho pois consegui atender a todos os requisitos do projeto dentro do prazo estipulado e busquei apresentar uma solução seguindo boas práticas de desenvolvimento. Além disso, foi muito enriquecedor para meu conhecimento a oportunidade de praticar diversos aspectos recorrentes no desenvolvimento de software como o planejamento da estratégia para desenvolver o software, a definição de uma arquitetura, trabalhar com leitura de arquivos, orientação a objetos e diversos outros aspectos. Acredito que pontos que poderiam ser melhorados na solução que apresentei poderiam ser uma interface de interação com usuário mais rica e com mais elementos visuais e também uma maior cobertura de testes automatizados para a aplicação.

- *Pontos de destaque:*
 - Reservei algumas horas para pensar na arquitetura do projeto de forma a mantê-la coesa e visando a redução ao máximo do acoplamento entre as camadas do projeto. Acredito ter atingido este objetivo e caso esta solução continuasse a evoluir não

seria difícil de alterar pontos como: interface de comunicação com usuário e forma de acesso a dados/fonte de dados.

- Segui atentamente os requisitos do projeto.
 - Consegui cumprir as implementações solicitadas de funcionalidades (casos de uso) por completo.
 - Busquei utilizar o máximo possível a utilização dos princípios SOLID durante a construção da solução.
 - Realizei tratamentos nos “inputs” do usuário, visando garantir uma boa experiência de uso da aplicação.
 - Documentei os requisitos e como executar o projeto em arquivo README.md.
- *Pontos de dificuldade:*
 - Infelizmente devido ao tempo disponível para construção da solução não foi possível realizar uma grande cobertura de testes automatizados para aplicação. Este seria o próximo passo para aumentar a qualidade da solução e conferir segurança e manutenibilidade futura na aplicação.