

Tópicos Avançados em Tecnologia da Informação (TSI35A)

Tecnologia em **Sistemas** para **Internet**

Aula 2

Introdução ao Apache Maven

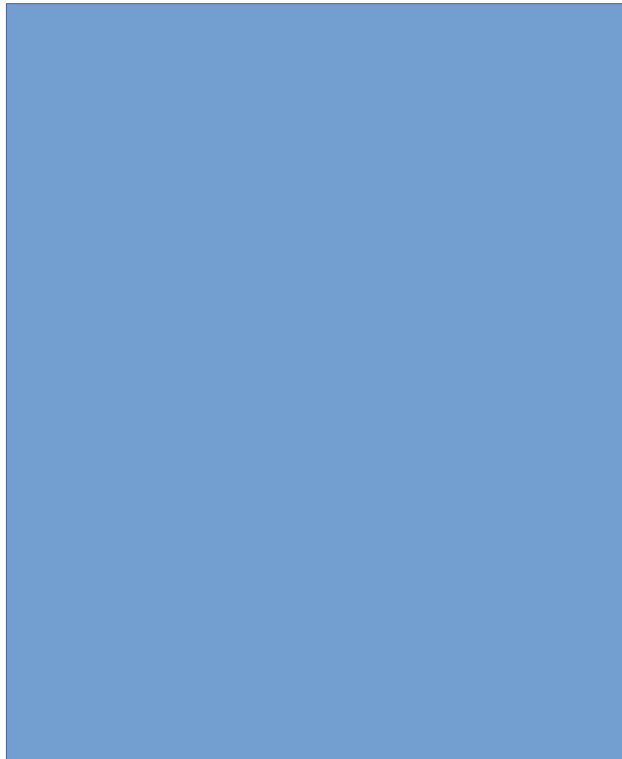
Introdução ao Apache Maven

Tópicos:

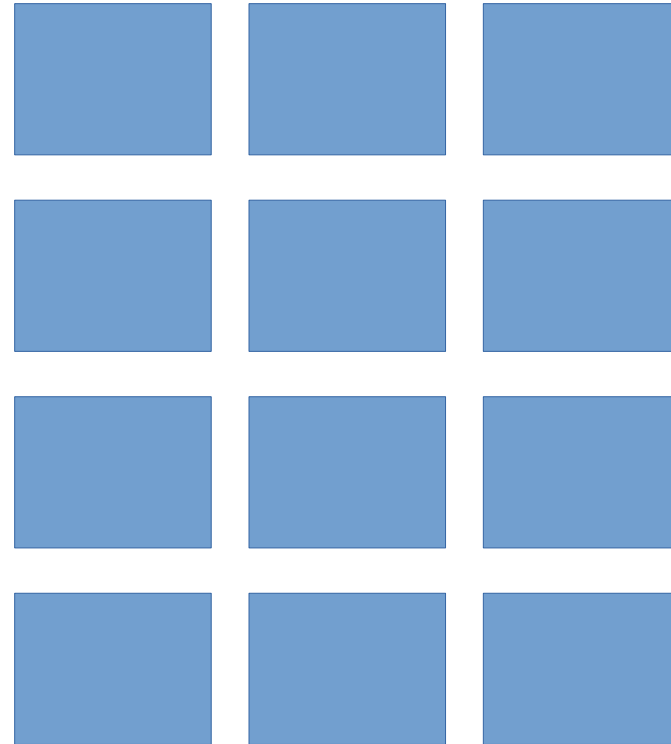
- Desenvolvimento de sistemas modulares
- Controle de dependências
- Apache Maven

Desenvolvimento de sistemas modulares

Não modular

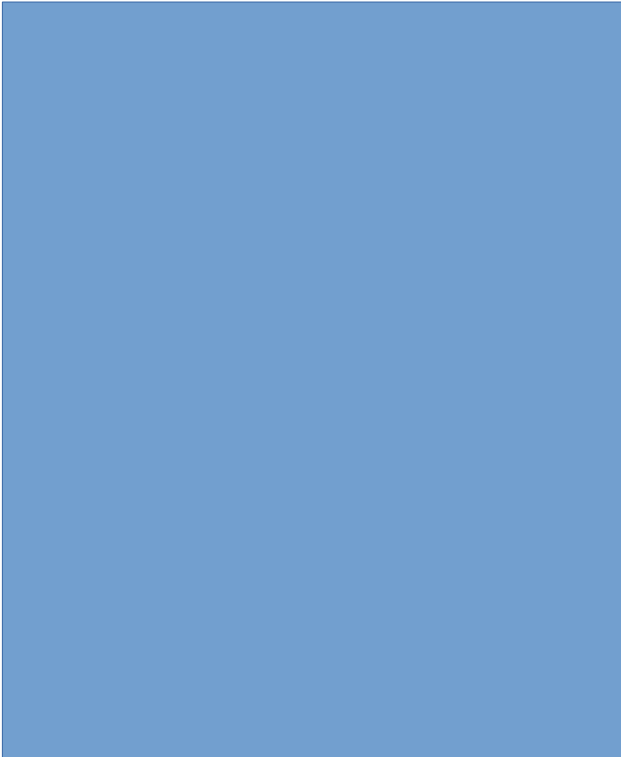


Modular



Desenvolvimento de sistemas modulares

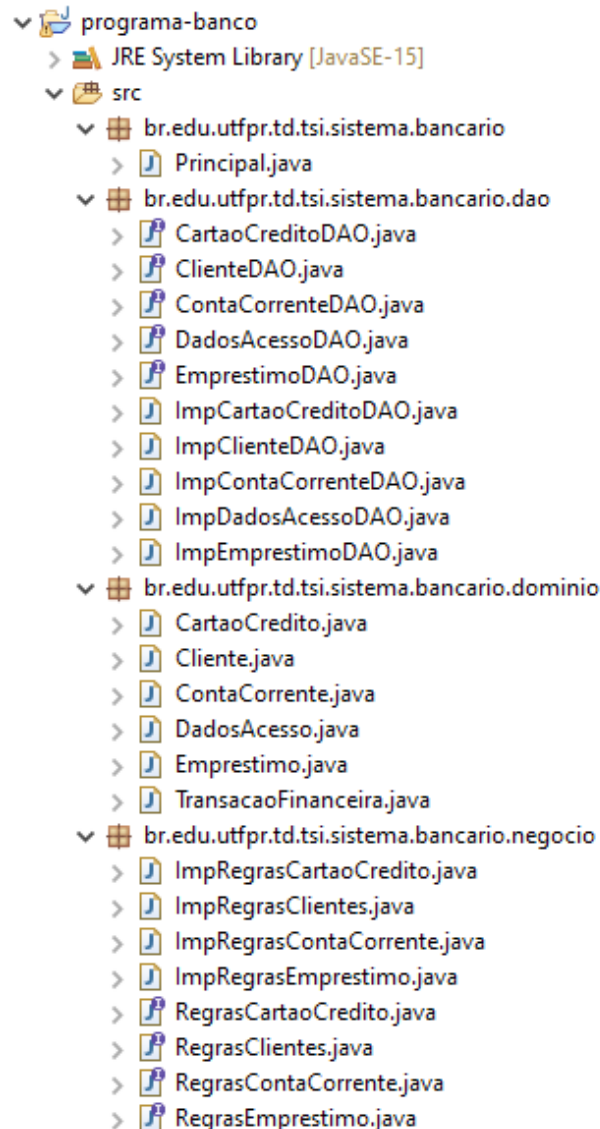
Não modular



Características:

- Todas as funcionalidades em um único artefato (projeto);
- Maior concorrência de acesso.
- Fraca definição de fronteiras entre funcionalidades
- Alta entropia
- Baixa coesão

Desenvolvimento de sistemas modulares



Características:

- Todas as funcionalidades em um único artefato (projeto);
- Maior concorrência de acesso.
- Fraca definição de fronteiras entre funcionalidades
- Alta entropia
- Baixa coesão

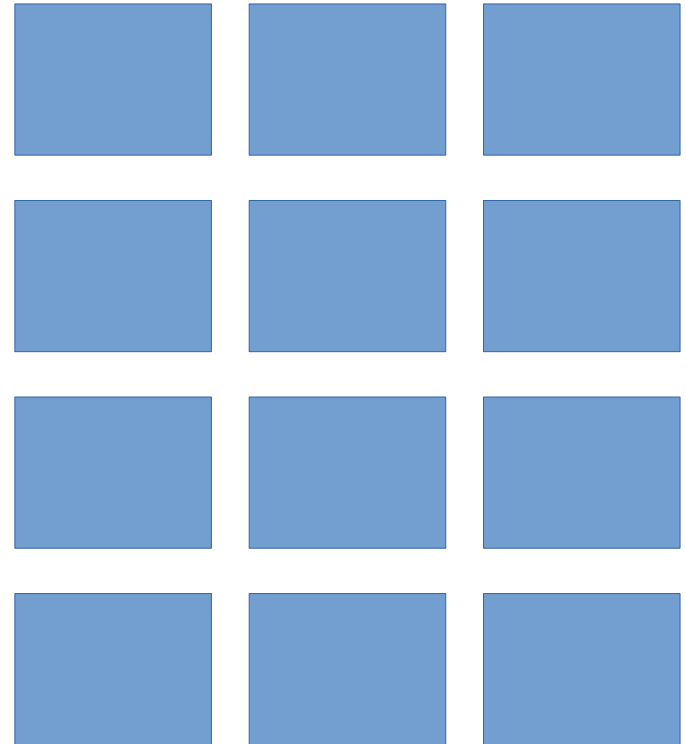
Não modular

Desenvolvimento de sistemas modulares

Características:






- Funcionalidades dispersas em vários artefatos (projetos);
- Menor concorrência de acesso.
- Forte definição de fronteiras entre funcionalidades
- Alta coesão
- Facilidade depuração e testes

Modular



Desenvolvimento de sistemas modulares

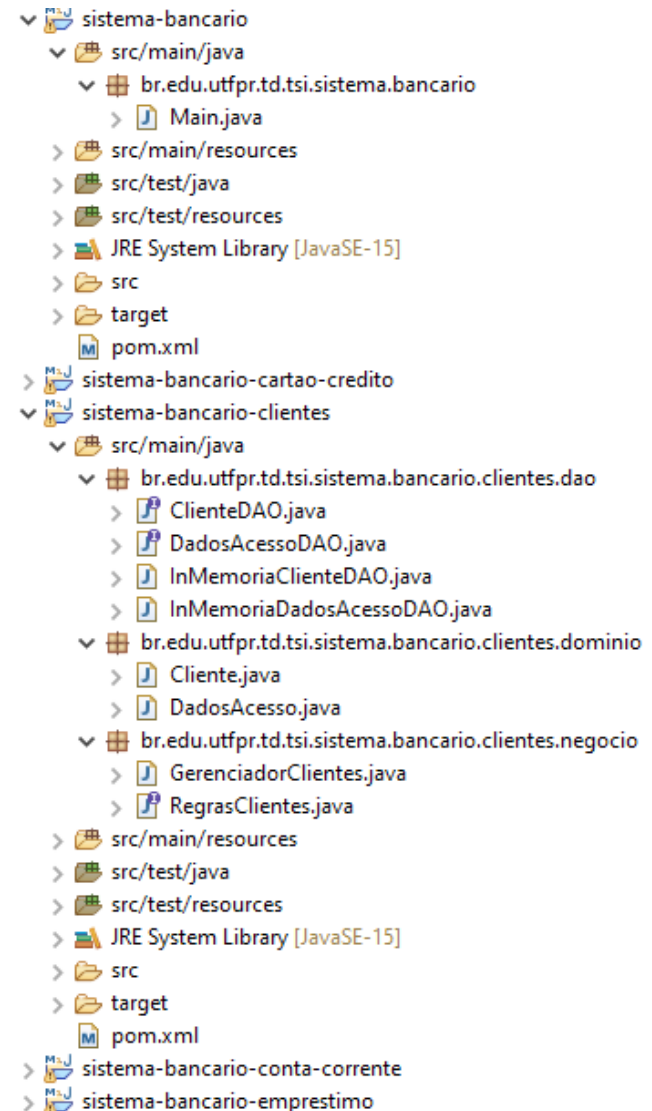
Características:

- Funcionalidades dispersas em vários artefatos (projetos);
 - Menor concorrência de acesso.
 - Forte definição de fronteiras entre funcionalidades
 - Alta coesão
 - Facilidade depuração e testes
- >  sistema-bancario
 - >  sistema-bancario-cartao-credito
 - >  sistema-bancario-clientes
 - >  sistema-bancario-conta-corrente
 - >  sistema-bancario-emprestimo

Desenvolvimento de sistemas modulares

Características:

- Funcionalidades dispersas em vários artefatos (projetos);
- Menor concorrência de acesso.
- Forte definição de fronteiras entre funcionalidades
- Alta coesão
- Facilidade depuração e testes



Motivos para criar módulos:

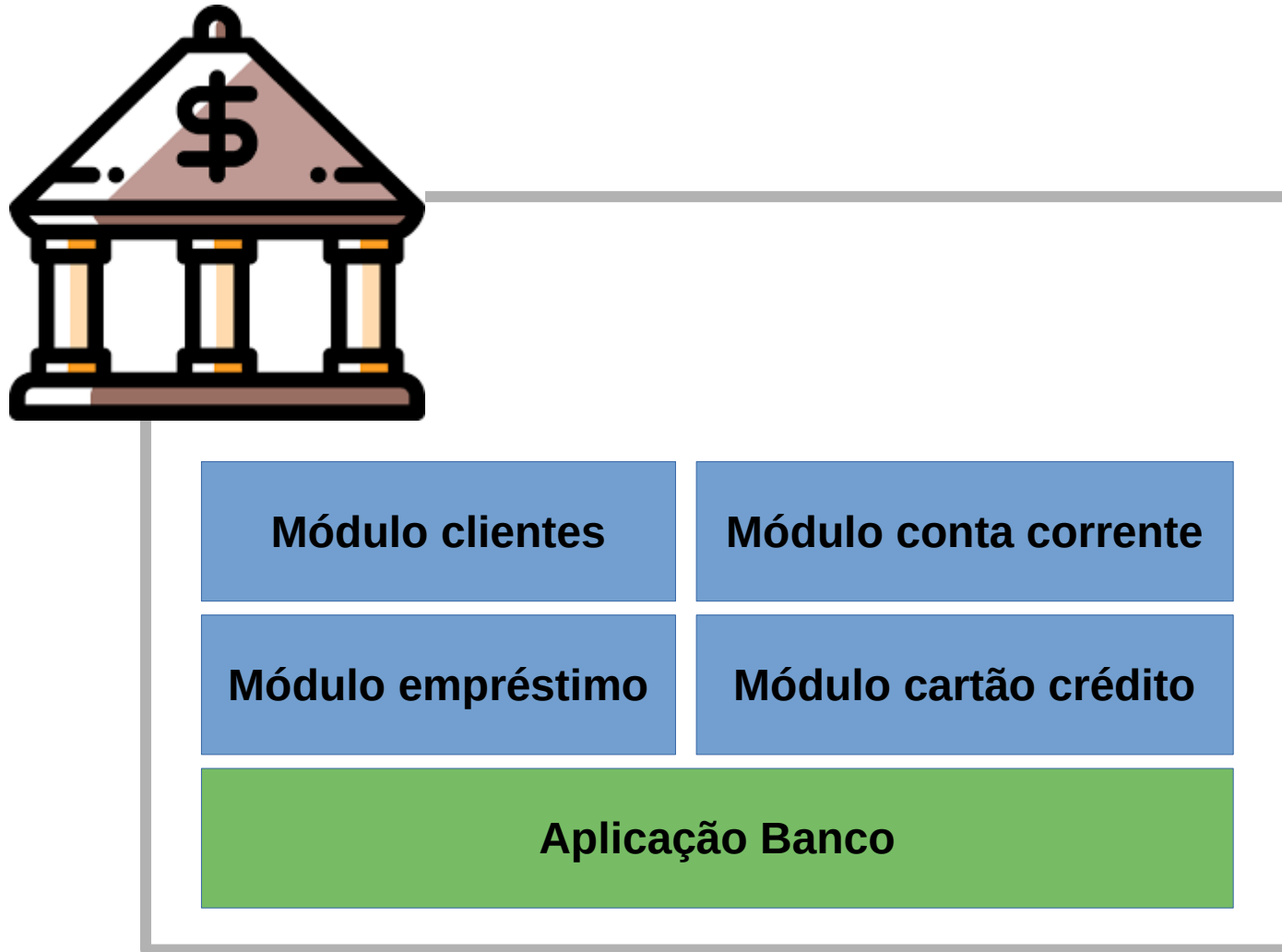
- Definição explícita de fronteiras de software
- Organização da estrutura de sistema (bibliotecas utilitárias)
- Testabilidade (identificar pontos de melhoria e correções)
- Organização de equipes
- Simplicidade para manter a base de código
- Menos configuração de ambiente de desenvolvimento
- Menor interferência entre elementos internos de um sistema

Desenvolvimento de sistemas modulares

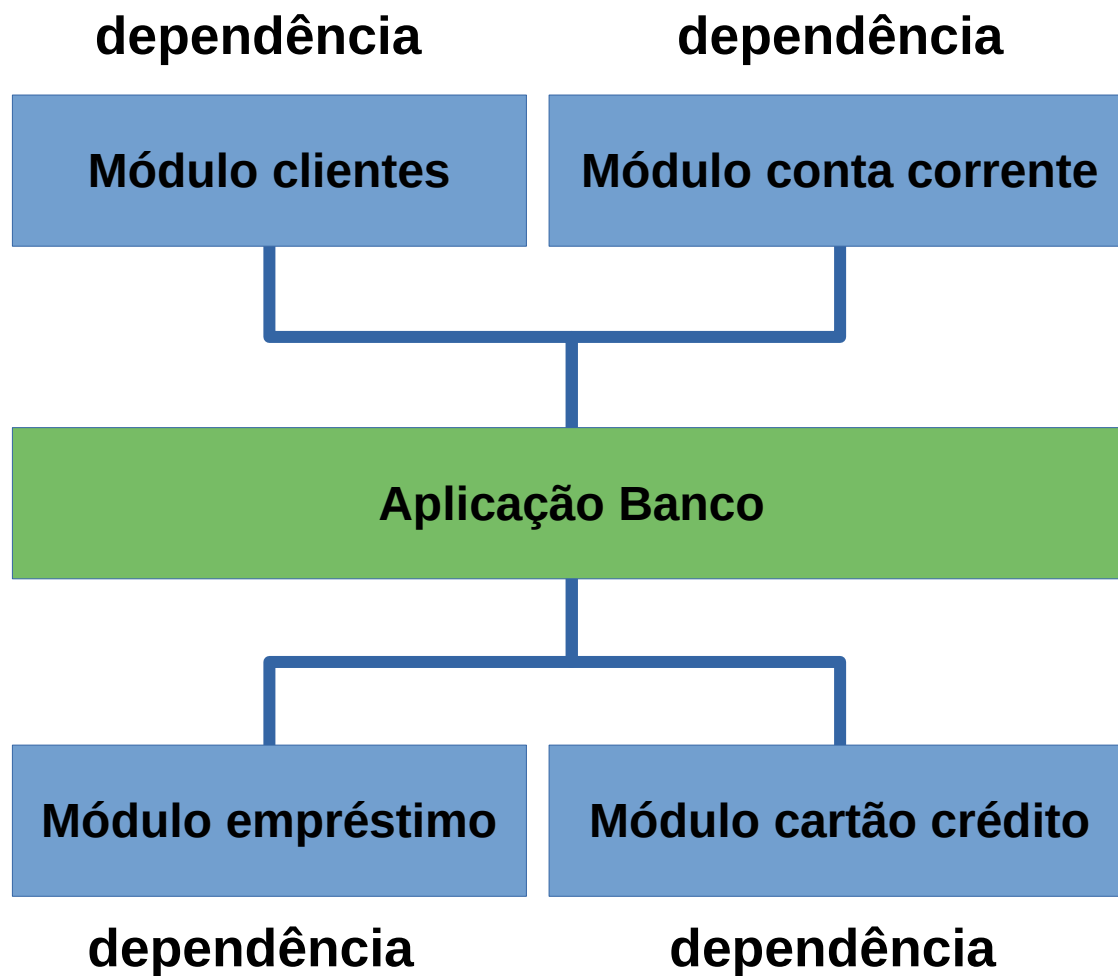


- Gerenciar clientes
- Gerenciar conta corrente
- Gerenciar cartão de crédito
- Gerenciar empréstimo

Desenvolvimento de sistemas modulares

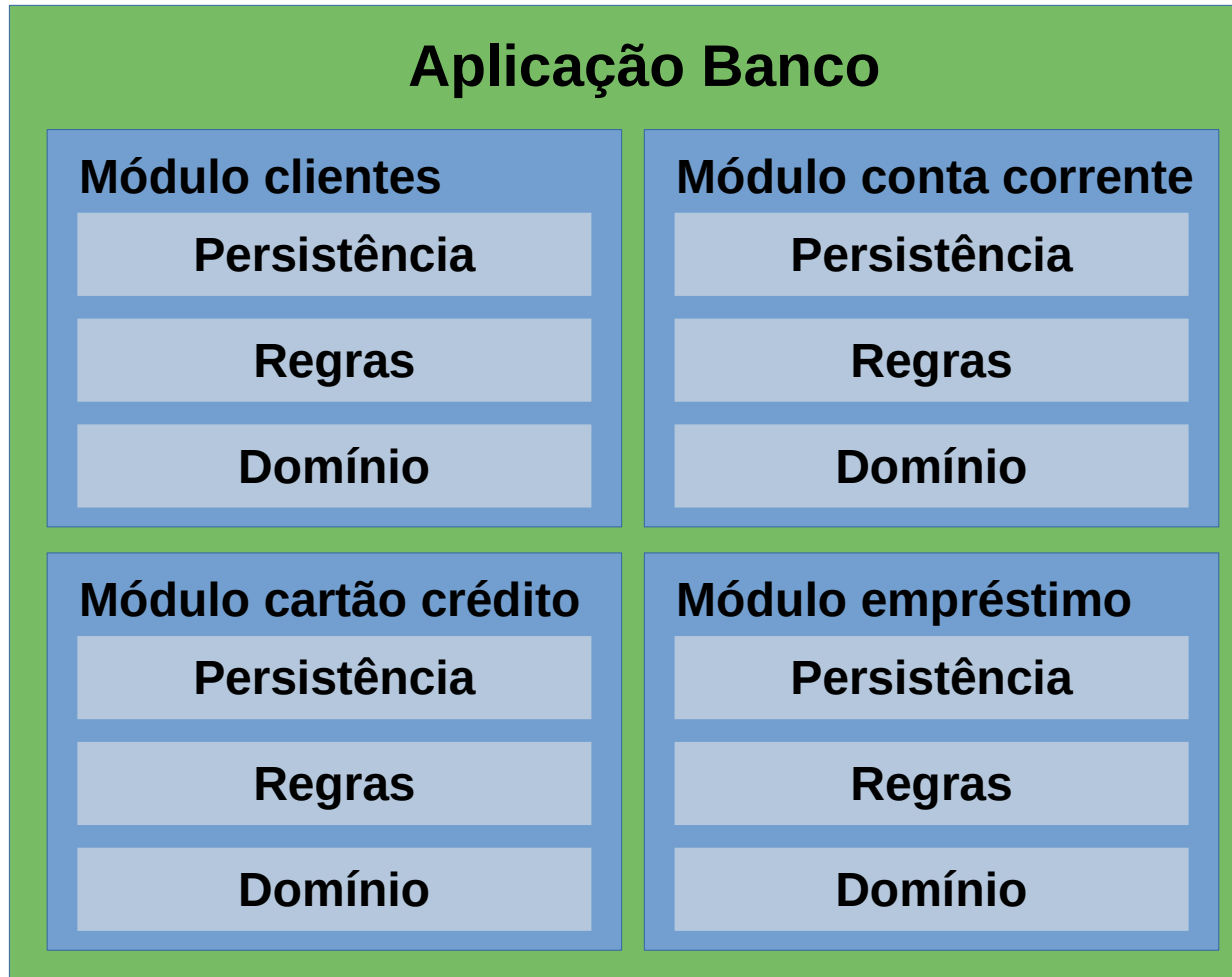


Desenvolvimento de sistemas modulares



Desenvolvimento de sistemas modulares

Software executável



Desenvolvimento de sistemas modulares

Múltiplos repositórios

 **sistema-bancario**

 **Main.java**

 **src/main/test**

 **Teste1.java**

 **Teste2.java**

 **Teste3.java**

 **sistema-bancario-conta-corrente**

 **ImpRegras.java**

 **src/main/test**

 **Teste1.java**

 **Teste2.java**

 **Teste3.java**

 **sistema-bancario-cartao-credito**

 **ImpRegras.java**

 **src/main/test**

 **Teste1.java**

 **Teste2.java**

 **Teste3.java**

 **sistema-bancario-emprestimo**

 **ImpRegras.java**

 **src/main/test**















 **Teste1.java**

 **Teste2.java**















 **Teste3.java**

Exemplo de Projeto Modular (Java)

Projeto modular















- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo

Projeto modular

- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo















Classes do projeto

Projeto modular

- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo















Arquivos de configuração

Projeto modular

- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo















Cenários de teste

Projeto modular

- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo

Arquivos de configuração
(Cenários de teste)

Projeto modular

- ▼  sistema-bancario
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - >  target
 -  pom.xml
- >  sistema-bancario-cartao-credito
- >  sistema-bancario-clientes
- >  sistema-bancario-conta-corrente
- >  sistema-bancario-emprestimo

Configuração Maven
do Projeto

Projeto modular

```
sistema-bancario/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>br.edu.utfpr.td.tsi</groupId>
6   <artifactId>sistema-bancario</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8
9   <dependencies>
10    <dependency>
11      <groupId>br.edu.utfpr.td.tsi</groupId>
12      <artifactId>sistema-bancario-conta-corrente</artifactId>
13      <version>0.0.1-SNAPSHOT</version>
14    </dependency>
15    <dependency>
16      <groupId>br.edu.utfpr.td.tsi</groupId>
17      <artifactId>sistema-bancario-cartao-credito</artifactId>
18      <version>0.0.1-SNAPSHOT</version>
19    </dependency>
20    <dependency>
21      <groupId>br.edu.utfpr.td.tsi</groupId>
22      <artifactId>sistema-bancario-clientes</artifactId>
23      <version>0.0.1-SNAPSHOT</version>
24    </dependency>
25    <dependency>
26      <groupId>br.edu.utfpr.td.tsi</groupId>
27      <artifactId>sistema-bancario-emprestimo</artifactId>
28      <version>0.0.1-SNAPSHOT</version>
29    </dependency>
30  </dependencies>
31
32 </project>
```

Projeto modular

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>br.edu.utfpr.td.tsi</groupId>
6   <artifactId>sistema-bancario</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8
9   <dependencies>
10    <dependency>
11      <groupId>br.edu.utfpr.td.tsi</groupId>
12      <artifactId>sistema-bancario-conta-corrente</artifactId>
13      <version>0.0.1-SNAPSHOT</version>
14    </dependency>
15    <dependency>
16      <groupId>br.edu.utfpr.td.tsi</groupId>
17      <artifactId>sistema-bancario-cartao-credito</artifactId>
18      <version>0.0.1-SNAPSHOT</version>
19    </dependency>
20    <dependency>
21      <groupId>br.edu.utfpr.td.tsi</groupId>
22      <artifactId>sistema-bancario-clientes</artifactId>
23      <version>0.0.1-SNAPSHOT</version>
24    </dependency>
25    <dependency>
26      <groupId>br.edu.utfpr.td.tsi</groupId>
27      <artifactId>sistema-bancario-emprestimo</artifactId>
28      <version>0.0.1-SNAPSHOT</version>
29    </dependency>
30  </dependencies>
31
32 </project>
```



0.0.1-SNAPSHOT
?

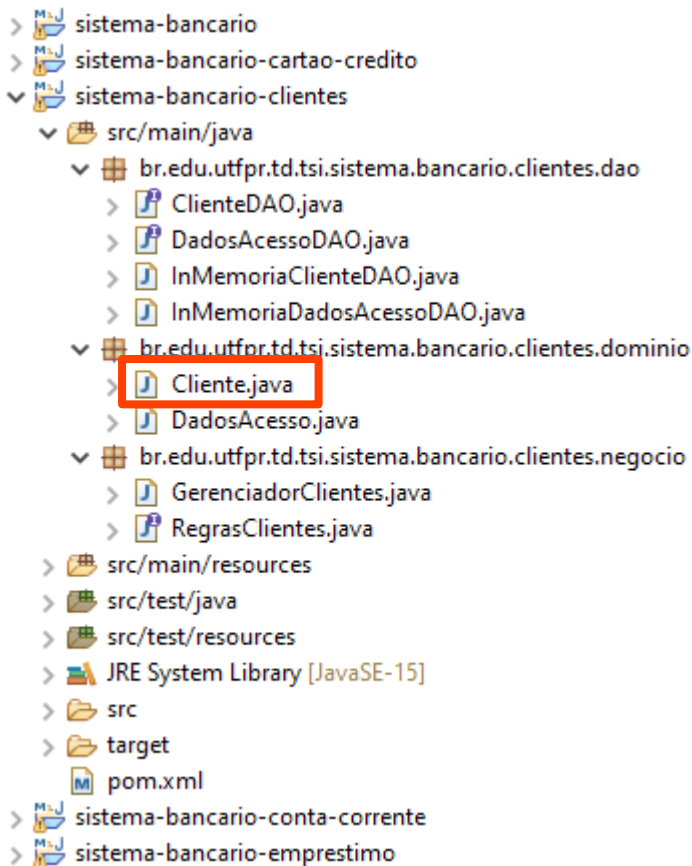
Projeto modular

```
sistema-bancario/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>br.edu.utfpr.td.tsi</groupId>
6   <artifactId>sistema-bancario</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8
9   <dependencies>
10     <dependency>
11       <groupId>br.edu.utfpr.td.tsi</groupId>
12       <artifactId>sistema-bancario-conta-corrente</artifactId>
13       <version>0.0.1-SNAPSHOT</version>
14     </dependency>
15     <dependency>
16       <groupId>br.edu.utfpr.td.tsi</groupId>
17       <artifactId>sistema-bancario-cartao-credit</artifactId>
18       <version>0.0.1-SNAPSHOT</version>
19     </dependency>
20     <dependency>
21       <groupId>br.edu.utfpr.td.tsi</groupId>
22       <artifactId>sistema-bancario-clientes</artifactId>
23       <version>0.0.1-SNAPSHOT</version>
24     </dependency>
25     <dependency>
26       <groupId>br.edu.utfpr.td.tsi</groupId>
27       <artifactId>sistema-bancario-emprestimo</artifactId>
28       <version>0.0.1-SNAPSHOT</version>
29     </dependency>
30   </dependencies>
31
32 </project>
```

Dependências
do Projeto

Exemplo de Implementação

Exemplo de implementação



```
Cliente.java
1 package br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio;
2
3 public class Cliente {
4
5     private String id;
6     private String nome;
7     private String cpf;
8     private String email;
9     private String dataNascimento;
10    private double rendaAnual;
11
12    public String getId() {..}
13
14
15
16    public void setId(String id) {..}
17
18
19
20    public String getNome() {..}
21
22
23
24    public void setNome(String nome) {..}
25
26
27
28    public String getCpf() {..}
29
30
31
32    public void setCpf(String cpf) {..}
33
34
35
36    public String getEmail() {..}
37
38
39
40    public void setEmail(String email) {..}
41
42
43
44    public String getDataNascimento() {..}
45
46
47
48    public void setDataNascimento(String dataNascimento) {..}
49
50
51
52    public double getRendaAnual() {..}
53
54
55
56    public void setRendaAnual(double rendaAnual) {..}
57
58
59
60 }
```

Exemplo de implementação

- > sistema-bancario
 - > sistema-bancario-cartao-credito
 - > sistema-bancario-clientes
 - src/main/java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao
 - ClienteDAO.java
 - DadosAcessoDAO.java
 - InMemoriaClienteDAO.java
 - InMemoriaDadosAcessoDAO.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio
 - Cliente.java
 - DadosAcesso.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio
 - GerenciadorClientes.java
 - RegrasClientes.java**
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-15]
 - src
 - target
 - pom.xml
 - > sistema-bancario-conta-corrente
 - > sistema-bancario-emprestimo

```
RegrasClientes.java x
1 package br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio;
2
3 import java.util.List;
4
5
6
7
8 public interface RegrasClientes {
9
10     public void cadastrar(Cliente cliente);
11     public void remover(String cpf);
12     public void alterar(Cliente cliente);
13     public Cliente ler(String cpf);
14     public List<Cliente> listarTodosClientes();
15
16     public void recuperarSenha(String cpf);
17     public void alterarDadosAcesso(DadosAcesso dadosAcesso);
18 }
```

Exemplo de implementação

- > sistema-bancario
 - > sistema-bancario-cartao-credito
 - > sistema-bancario-clientes
 - src/main/java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao
 - ClienteDAO.java
 - DadosAcessoDAO.java
 - InMemoriaClienteDAO.java
 - InMemoriaDadosAcessoDAO.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio
 - Cliente.java
 - DadosAcesso.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio
 - GerenciadorClientes.java**
 - RegrasClientes.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-15]
 - src
 - target
 - pom.xml
 - > sistema-bancario-conta-corrente
 - > sistema-bancario-emprestimo

```
GerenciadorClientes.java
1 package br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio;
2
3 import java.util.List;
4
5
6
7
8
9 public class GerenciadorClientes implements RegrasClientes{
10
11     InMemoriaClienteDAO clienteDAO = new InMemoriaClienteDAO();
12
13     @Override
14     public void cadastrar(Cliente cliente) {
15         System.out.println("Verificando dados do cliente");
16         if(cliente.getRendaAnual() < 50000) {
17             throw new RuntimeException("Renda anual insuficiente");
18         }
19         clienteDAO.gravar(cliente);
20         System.out.println("Cliente cadastrado");
21     }
22
23
24 public void remover(String cpf) {}
25
26
27
28 public void alterar(Cliente cliente) {}
29
30
31
32
33
34 public Cliente ler(String cpf) {}
35
36
37
38
39
40 public List<Cliente> listarTodosClientes() {}
41
42
43
44
45
46
47
48 public void recuperarSenha(String cpf) {}
49
50
51
52
53
54 public void alterarDadosAcesso(DadosAcesso dadosAcesso) {}
55
56
57
58
59 }
```

Exemplo de implementação

- > sistema-bancario
- > sistema-bancario-cartao-credito
- ▼ sistema-bancario-clientes
 - ▼ src/main/java
 - ▼ br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao
 - ▶ **ClienteDAO.java**
 - > DadosAcessoDAO.java
 - > InMemoriaClienteDAO.java
 - > InMemoriaDadosAcessoDAO.java
 - ▼ br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio
 - > Cliente.java
 - > DadosAcesso.java
 - ▼ br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio
 - > GerenciadorClientes.java
 - > RegrasClientes.java
 - > src/main/resources
 - > src/test/java
 - > src/test/resources
 - > JRE System Library [JavaSE-15]
 - > src
 - > target
 - > pom.xml- > sistema-bancario-conta-corrente
- > sistema-bancario-emprestimo

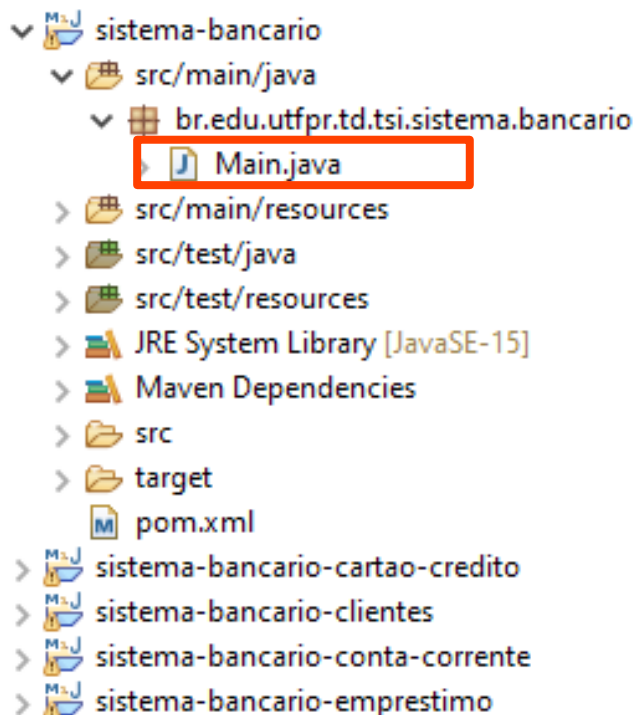
```
ClienteDAO.java ✕
1 package br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao;
2
3 import java.util.List;
4
5
6
7 public interface ClienteDAO {
8
9     public void gravar(Cliente cliente);
10    public Cliente ler(String cpf);
11    public void remover(String cpf);
12    public List<Cliente> lerTodos();
13 }
14
```

Exemplo de implementação

- > sistema-bancario
 - > sistema-bancario-cartao-credito
 - > sistema-bancario-clientes
 - src/main/java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao
 - ClienteDAO.java
 - DadosAcessoDAO.java
 - InMemoriaClienteDAO.java
 - InMemoriaDadosAcessoDAO.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio
 - Cliente.java
 - DadosAcesso.java
 - br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio
 - GerenciadorClientes.java
 - RegrasClientes.java
 - src/main/resources
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-15]
 - src
 - target
 - pom.xml
 - > sistema-bancario-conta-corrente
 - > sistema-bancario-emprestimo

```
InMemoriaClienteDAO.java
1 package br.edu.utfpr.td.tsi.sistema.bancario.clientes.dao;
2
3 import java.util.ArrayList;
4
5
6
7 public class InMemoriaClienteDAO implements ClienteDAO{
8
9
10     List<Cliente> clientesCadastrados = new ArrayList<Cliente>();
11
12     @Override
13     public void gravar(Cliente cliente) {
14         this.clientesCadastrados.add(cliente);
15         System.out.println("Cliente persistido em memoria.");
16     }
17
18     @Override
19     public Cliente ler(String cpf) {
20         // TODO Auto-generated method stub
21         return null;
22     }
23
24     @Override
25     public void remover(String cpf) {
26         // TODO Auto-generated method stub
27     }
28
29
30     @Override
31     public List<Cliente> lerTodos() {
32         // TODO Auto-generated method stub
33         return null;
34     }
35 }
```

Exemplo de implementação



```
Main.java
1 package br.edu.utfpr.td.tsi.sistema.bancario;
2
3 import br.edu.utfpr.td.tsi.sistema.bancario.clientes.dominio.Cliente;
4 import br.edu.utfpr.td.tsi.sistema.bancario.clientes.negocio.GerenciadorClientes;
5
6 public class Main {
7
8     public static void main(String[] args) {
9
10         Cliente cliente = new Cliente();
11         cliente.setCpf("1234");
12         cliente.setDataNascimento("10/10/2000");
13         cliente.setNome("joao silva");
14         cliente.setEmail("joao@silva.com");
15         cliente.setRendaAnual(50000);
16
17         GerenciadorClientes gerenciadorClientes = new GerenciadorClientes();
18         gerenciadorClientes.cadastrar(cliente);
19     }
20 }
```

Problems @ Javadoc Declaration Console Progress

<terminated> Main (7) [Java Application] C:\Program Files\Java\bin\javaw.exe (15 de ago. de 2022 14:54:35 – 14:54:35)

Verificando dados do cliente
Cliente persistido em memoria.
Cliente cadastrado


















Mvn build

```
sistema-bancario/pom.xml
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>br.edu.utfpr.td.tsi</groupId>
6   <artifactId>sistema-bancario</artifactId>
7   <version>0.0.1-SNAPSHOT</version>
8
9   <properties>
10     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
11     <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
12     <java.version>15</java.version>
13     <maven.compiler.source>15</maven.compiler.source>
14     <maven.compiler.target>15</maven.compiler.target>
15   </properties>
16
17   <dependencies>
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44   <build>
45     <plugins>
46       <plugin>
47         <groupId>org.springframework.boot</groupId>
48         <artifactId>spring-boot-maven-plugin</artifactId>
49         <executions>
50           <execution>
51             <goals>
52               <goal>repackage</goal>
53             </goals>
54           </execution>
55         </executions>
56       </plugin>
57     </plugins>
58   </build>
59
60 </project>
```


















Mvn build

```
Overview Dependencies Dependency Hierarchy Effective POM pom.xml
Problems @ Javadoc Declaration Console Progress
<terminated> sistema-bancario [Maven Build] C:\Program Files\Java\bin\javaw.exe (15 de ago. de 2022 15:03:37 - 15:03:41)
[INFO] -----< br.edu.utfpr.td.tsi:sistema-bancario >-----
[INFO] Building sistema-bancario 0.0.1-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ sistema-bancario ---
[INFO] Deleting D:\Dev\eclipse-workspace\sistema-bancario\target
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ sistema-bancario ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ sistema-bancario ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\Dev\eclipse-workspace\sistema-bancario\target\classes
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ sistema-bancario ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ sistema-bancario ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ sistema-bancario ---
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ sistema-bancario ---
[INFO] Building jar: D:\Dev\eclipse-workspace\sistema-bancario\target\sistema-bancario-0.0.1-SNAPSHOT.jar
[INFO] --- spring-boot-maven-plugin:2.7.2:repackage (default) @ sistema-bancario ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.362 s
[INFO] Finished at: 2022-08-15T15:03:41-03:00
[INFO] -----
```

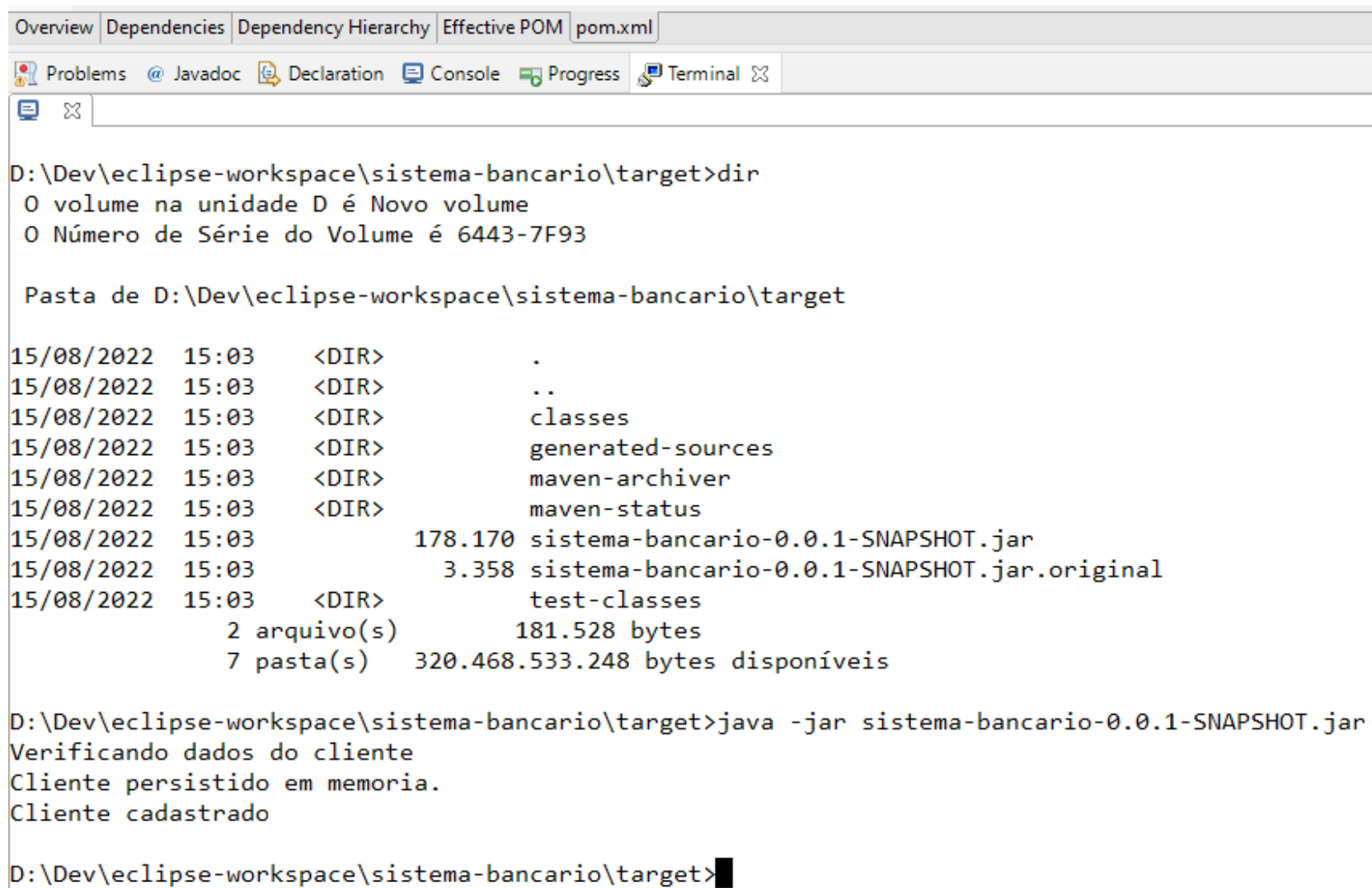
Exemplo de implementação

- ▼  sistema-bancario
 - ▼  src/main/java
 - ▼  br.edu.utfpr.td.tsi.sistema.bancario
 - >  Main.java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - ▼  target
 - >  generated-sources
 - >  maven-archiver
 - >  maven-status
 - >  sistema-bancario-0.0.1-SNAPSHOT.jar
 - >  sistema-bancario-0.0.1-SNAPSHOT.jar.original
 - >  pom.xml

Exemplo de implementação

- ▼  sistema-bancario
 - ▼  src/main/java
 - ▼  br.edu.utfpr.td.tsi.sistema.bancario
 - >  Main.java
 - >  src/main/resources
 - >  src/test/java
 - >  src/test/resources
 - >  JRE System Library [JavaSE-15]
 - >  Maven Dependencies
 - >  src
 - ▼  target
 - >  generated-sources
 - >  maven-archiver
 - >  maven-status
 - >  sistema-bancario-0.0.1-SNAPSHOT.jar
 - >  sistema-bancario-0.0.1-SNAPSHOT.jar.original
-  pom.xml

Exemplo de implementação



The screenshot shows an IDE interface with tabs for Overview, Dependencies, Dependency Hierarchy, Effective POM, and pom.xml. Below the tabs are icons for Problems, Javadoc, Declaration, Console, Progress, and Terminal. The main area displays the output of a Maven build and a Java execution.

```
D:\Dev\eclipse-workspace\sistema-bancario\target>dir
O volume na unidade D é Novo volume
O Número de Série do Volume é 6443-7F93

Pasta de D:\Dev\eclipse-workspace\sistema-bancario\target

15/08/2022  15:03    <DIR>        .
15/08/2022  15:03    <DIR>        ..
15/08/2022  15:03    <DIR>        classes
15/08/2022  15:03    <DIR>        generated-sources
15/08/2022  15:03    <DIR>        maven-archiver
15/08/2022  15:03    <DIR>        maven-status
15/08/2022  15:03           178.170 sistema-bancario-0.0.1-SNAPSHOT.jar
15/08/2022  15:03           3.358 sistema-bancario-0.0.1-SNAPSHOT.jar.original
15/08/2022  15:03    <DIR>        test-classes
                2 arquivo(s)        181.528 bytes
                7 pasta(s)    320.468.533.248 bytes disponíveis

D:\Dev\eclipse-workspace\sistema-bancario\target>java -jar sistema-bancario-0.0.1-SNAPSHOT.jar
Verificando dados do cliente
Cliente persistido em memoria.
Cliente cadastrado

D:\Dev\eclipse-workspace\sistema-bancario\target>
```

Questões para pensar:

- 1)Qual a relação entre aplicações modulares e teste de software?
- 2)Como resolver o problema de modelo anêmico?
- 3)Qual relação entre aplicações modulares, dependências e versões?