

Universidade Presbiteriana Mackenzie  
Faculdade de Computação e Informática

# Lab. 05 - Classes de acesso aos dados da base

Linguagem de Programação II  
2º semestre de 2015  
Versão 1.0

# Objetivos

- Criar tabelas em base de dados para persistir informações da aplicação.
- Desenvolver classes responsáveis pelo acesso aos dados

# CRIAÇÃO DO BANCO DE DADOS



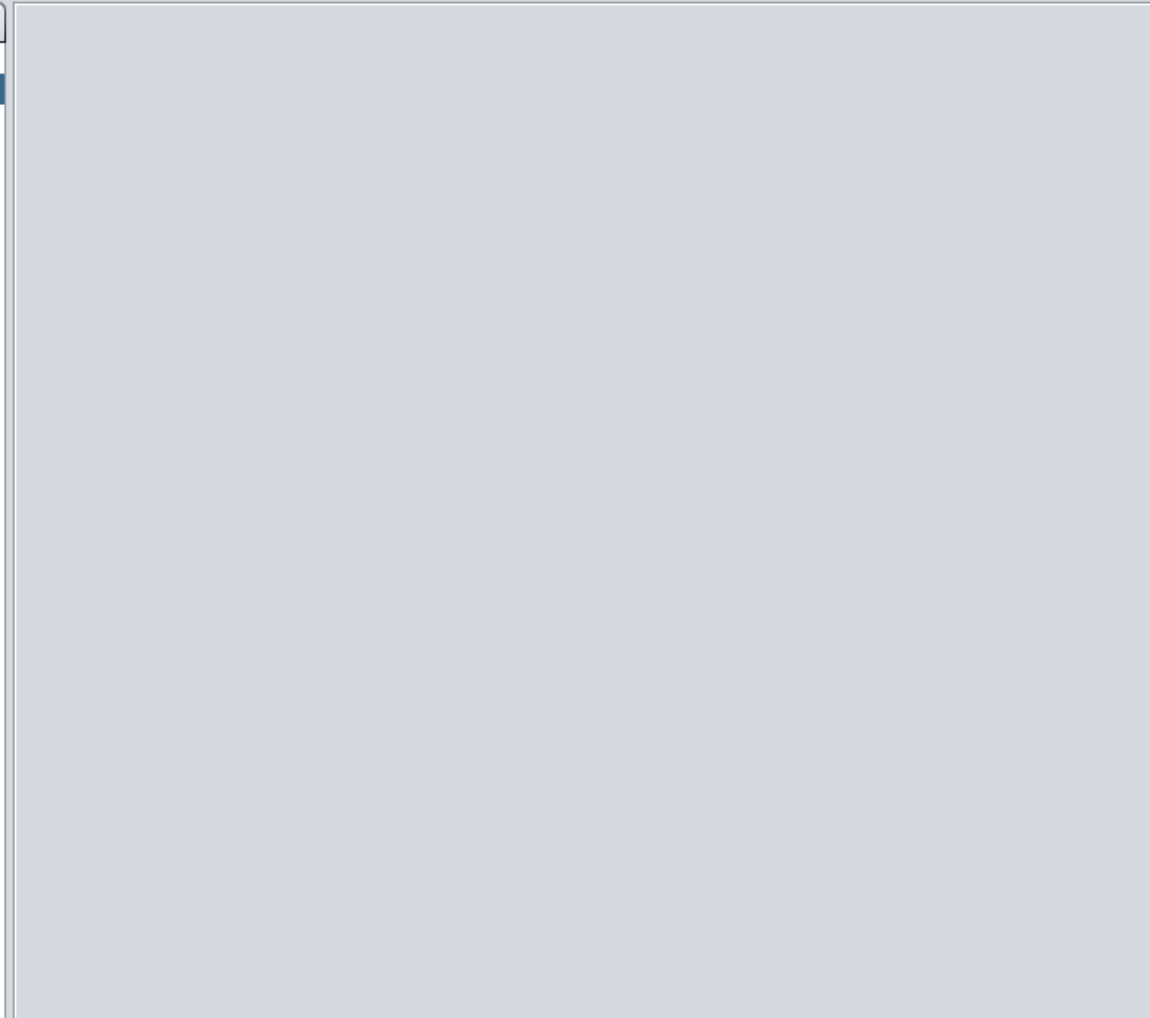
Projetos x Serviços Arquivos

&lt;Nenhum Projeto Aberto&gt;

Notificações Resultados do Teste Saída Resultados da Pesquisa

INS

- Projetos Serviços x Arquivos
- Bancos de Dados
    - Java
      - Driver
        - Iniciar Servidor
        - Interromper Servidor
        - Criar Banco de Dados...
        - Propriedades...
      - jdbc [app em APP]
      - Servidor
      - Repositório
      - Nuvem
      - Builders do Hudson
      - Repositórios de Tarefas



Projetos Serviços x Arquivos

- ▼ Bancos de Dados
  - ▶ Java DB
    - ▶ Drivers
    - ▶ jdbc:derby://localhost:1527/banco [app em APP]
- ▶ Servidores
- ▶ Repositórios Maven
- ▶ Nuvem
- ▶ Builders do Hudson
- ▶ Repositórios de Código

**Criar Banco de Dados Java DB**

Nome do Banco de Dados:

Nome do Usuário:

Senha:

Confirmar Senha:

Localização do Banco de Dados:  [Propriedades...](#)

Projetos Serviços x Arquivos

Bancos de Dados

- Java DB
  - Drivers
  - jdbc:derby://localhost:1527/banco [app em APP]
- Servidores
- Repositórios Maven
- Nuvem
- Builders do Huc
- Repositórios de

Nome do Ba

Nome do Us

Senha:

Confirmar S

Localização

## Propriedades do Java DB

Especifique a pasta onde o Java BD está instalado e a pasta onde serão mantidos seus bancos de dados. A pasta da localização do banco de dados será utilizada como valor da propriedade derby.system.home.

Instalação do Java BD: C:\Program Files\GlassFish\_Server\javadb

Procurar...

Localização do Banco de Dados: chive\2015\MACK-LP2-lab04-jdbc\BancoDados

Procurar...

Se você estiver utilizando o NetBeans 8.0.2, utilize o Java DB que está no diretório do GlassFish (conforme mostrado no campo destacado)

OK

Cancelar

Cancelar

Notificações Resultados do Teste Saída Resultados da Pesquisa

INS

Projetos Serviços x Arquivos

Bancos de Dados

- Java DB
  - Drivers
  - jdbc:derby://localhost:1527/banco [app em APP]
- Servidores
- Repositórios Maven
- Nuvem
- Builders do Huc
- Repositórios de

Nome do Ba

Nome do Us

Senha:

Confirmar S

Localização

## Propriedades do Java DB

Especifique a pasta onde o Java BD está instalado e a pasta onde serão mantidos seus bancos de dados. A pasta da localização do banco de dados será utilizada como valor da propriedade derby.system.home.

Instalação do Java BD: C:\Program Files\GlassFish Server\javadb

Procurar...

Localização do Banco de Dados: chive\2015\MACK-LP2-lab04-jdbc\BancoDados

Procurar...

OK

Cancelar

Proprieda

Cancelar

Notificações Resultados do Teste Saída Resultados da Pesquisa

INS



Projetos Serviços x Arquivos

Bancos de Dados

Java DB

Drivers

jdbc:derby://localhost:1527/banco

Servidores

Repositórios Maven

Nuvem

Builders do Huc

Repositórios de

Nome do Ba

Nome do Us

Senha:

Confirmar S

Localização



Abrir



Pesquisar em: MACK-LP2-lab04-jdbc



BancoDados

img

js

netbeans

selecione a pasta onde voce deseja  
gravar os dados da sua base

Nome do Arquivo: archive\2015\MACK-LP2-lab04-jdbc\BancoDados

Abrir

Arquivos do Tipo: Todos os Arquivos

Cancelar

Proprieda

Cancelar



Notificações



Resultados do Teste



Saída



Resultados da Pesquisa

INS



- Projetos Serviços x Arquivos
- Bancos de Dados
    - Java DB
      - Drivers
      - jdbc:derby://localhost:1527/banco [app em APP]
  - Servidores
  - Repositórios Maven
  - Nuvem
  - Builders do Huc
  - Repositórios de

Nome do Ba

Nome do Us

Senha:

Confirmar S

Localização

## Propriedades do Java DB

Especifique a pasta onde o Java BD está instalado e a pasta onde serão mantidos seus bancos de dados. A pasta da localização do banco de dados será utilizada como valor da propriedade derby.system.home.

Instalação do Java BD: C:\Program Files\GlassFish\_Server\javadb

Procurar...

Localização do Banco de Dados: chive\2015\MACK-LP2-lab04-jdbc\BancoDados

Procurar...

OK

Cancelar

Proprieda

Cancelar

- Projetos Serviços x Arquivos
- ▼ Bancos de Dados
    - ▶ Java DB
      - ▶ Drivers
      - ▶ jdbc:derby://localhost:1527/banco [app em APP]
  - ▶ Servidores
  - ▶ Repositórios Maven
  - ▶ Nuvem
  - ▶ Builders do Hudson
  - ▶ Repositórios de Código

**Criar Banco de Dados Java DB** [X]

Nome do Banco de Dados:

Nome do Usuário:

Senha:

Confirmar Senha:

Localização do Banco de Dados:  [Propriedades...](#)

Projetos Serviços x Arquivos

- ▼ Bancos de Dados
  - ▶ Java DB
  - ▶ Drivers
  - ▶ jdbc:derby://localhost:1527/sistema\_bancario\_fan

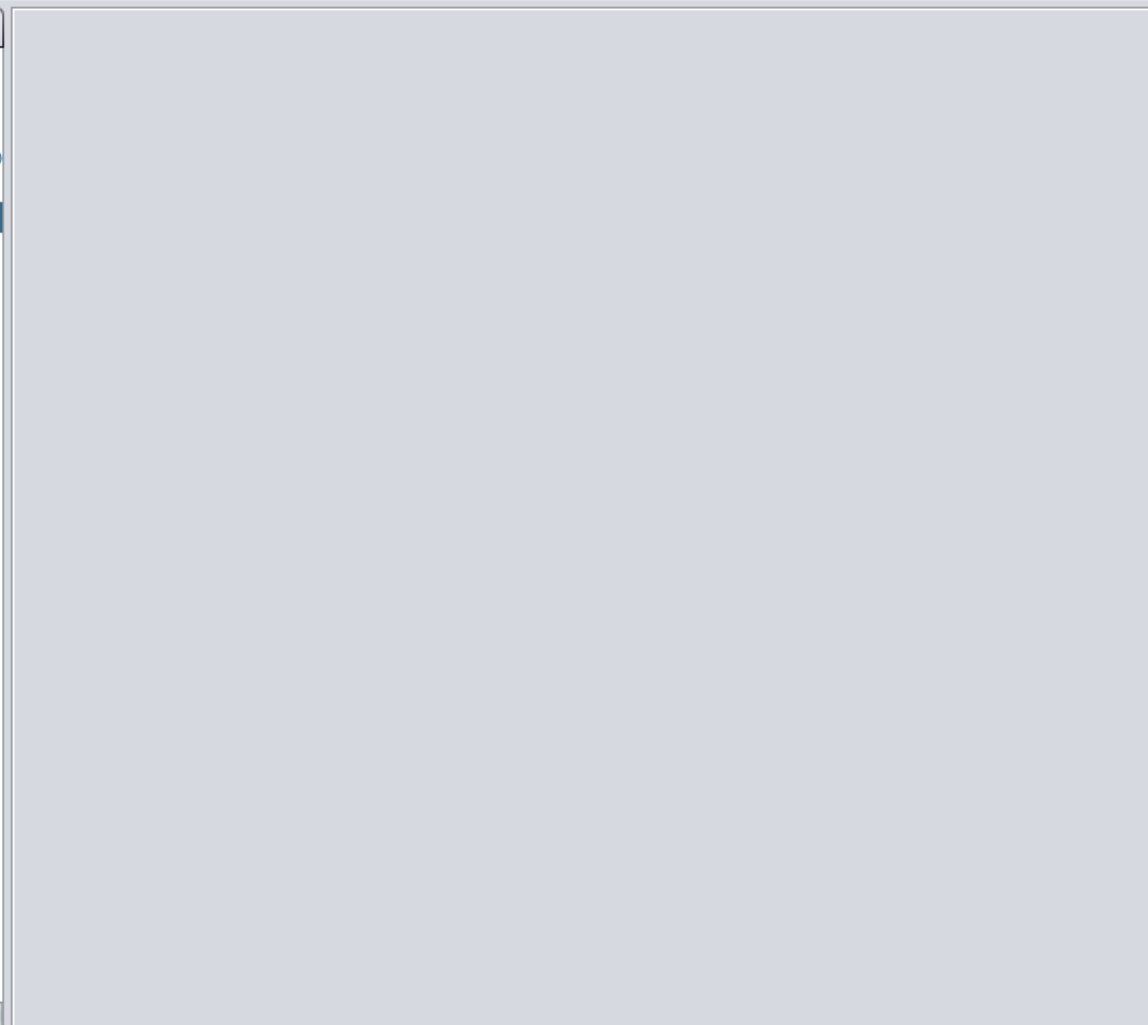
- ▶ Servidores
- ▶ Repositórios Maven
- ▶ Nuvem
- ▶ Builders do Hudson
- ▶ Repositórios de Tarefas

- Conectar...
- Desconectar
- Executar Comando...
- Atualizar
- Deletar Deletar
- Renomear...
- Propriedades

Notificações Resultados do Teste Saída Resultados da Pesquisa

INS

- Projetos Serviços x Arquivos
- ▼ Bancos de Dados
    - ▶ Java DB
    - ▶ Drivers
    - ▼ jdbc:derby://localhost:1527/sistema\_bancario [ap]
      - ▼ APP
        - ▶ Tabelas
          - Criar Tabela...
          - Recriar Tabela...
          - Executar Comando...
          - Atualizar
        - ▶ Views
        - ▶ Procedin
      - ▶ Outros esq
    - ▶ Servidores
    - ▶ Repositórios Mav
    - ▶ Nuvem
    - ▶ Builders do Hudson
    - ▶ Repositórios de Tarefas



INS

# Criação da tabela **contas** e inserção de dados iniciais

Digite os comandos abaixo:

```
CREATE TABLE contas (  
    nro_conta BIGINT NOT NULL,  
    saldo DECIMAL(8,2),  
    PRIMARY KEY (nro_conta)  
);  
  
INSERT INTO contas VALUES (123, 500.00);  
INSERT INTO contas VALUES (124, 1000.00);  
INSERT INTO contas VALUES (125, 2500.00);
```

Projetos Serviços x Arquivos

- Bancos de Dados
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sistema\_bancario [ap...]
    - APP
      - Tabelas
      - Views
      - Procedimentos
    - Outros esquemas
- Servidores
- Repositórios Maven
- Nuvm
- Builders do Hudson
- Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:15...] x

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
1 CREATE TABLE contas (  
2     nro_conta BIGINT NOT NULL,  
3     saldo DECIMAL(8,2),  
4     PRIMARY KEY (nro_conta)  
5 );  
6  
7 INSERT INTO contas VALUES (123, 500.00);  
8 INSERT INTO contas VALUES (124, 1000.00);  
9 INSERT INTO contas VALUES (125, 2500.00);
```



Projetos Serviços x Arquivos

- Bancos de Dados
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sistema\_bancario [ap
    - APP
      - Tabelas
      - Views
      - Procedimentos
      - Outros esquemas
- Servidores
- Repositórios Maven
- Nuvem
- Builders do Hudson
- Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:15...] x

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
1 CREATE TABLE contas (  
2     nro_conta BIGINT NOT NULL,  
3     saldo DECIMAL(8,2),  
4     PRIMARY KEY (nro_conta)  
5 );  
6  
7 INSERT INTO contas VALUES (123, 500.00);  
8 INSERT INTO contas VALUES (124, 1000.00);  
9 INSERT INTO contas VALUES (125, 2500.00);
```

- Navegar
- Formatar Alt+Shift+F
- Executar Seleção**
- Histórico SQL Ctrl+Alt+Shift+H
- Executar Arquivo Shift+F6
- Recortar Ctrl+X
- Copiar Ctrl+C
- Colar Ctrl+V
- Selecionar no Projetos
- Selecionar Conexão em Serviços

Projetos Serviços x Arquivos

- Bancos de Dados
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/sistema\_bancario [ap...]
    - APP
      - Tabelas
      - Views
      - Procedimentos
    - Outros esquemas
- Servidores
- Repositórios Maven
- Nuvem
- Builders do Hudson
- Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:1527/sistema\_bancario [app em APP]]

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
1 CREATE TABLE contas (  
2     nro_conta BIGINT NOT NULL,  
3     saldo DECIMAL(8,2),  
4     PRIMARY KEY (nro_conta)  
5 );  
6  
7 INSERT INTO contas VALUES (123, 500.00);  
8 INSERT INTO contas VALUES (124, 1000.00);  
9 INSERT INTO contas VALUES (125, 2500.00);
```

Saída

AppSelectContas (clean) x Processo do Banco de Dados Java DB x execução SQL 1 x

Executado com sucesso em 0,311 s, 0 linhas afetadas.  
Linha 1, coluna 1

Executado com sucesso em 0,093 s, 1 linhas afetadas.  
Linha 7, coluna 1

Executado com sucesso em 0,032 s, 1 linhas afetadas.  
Linha 8, coluna 1

Executado com sucesso em 0,015 s, 1 linhas afetadas.  
Linha 9, coluna 1

Execução finalizada após 0,451 s, 0 erro(s) encontrado(s).

← comandos foram executados com sucesso

Notificações Resultados do Teste Saída Resultados da Pesquisa

18 / 61  
9:21 INS



**Criação da tabela titulares e inserção de dados iniciais**



Digite os comandos abaixo:

```
CREATE TABLE titulares (  
    nro_titular BIGINT NOT NULL,  
    nome VARCHAR(255) NOT NULL,  
    rg VARCHAR(32) NOT NULL,  
    cpf VARCHAR(32) NOT NULL,  
    PRIMARY KEY (nro_titular)  
);  
  
INSERT INTO titulares VALUES (2, 'Marcos Antônio', '22333444', '09988877765');  
INSERT INTO titulares VALUES (3, 'Rosimeire Aparecida', '11222333', '08866655543');  
INSERT INTO titulares VALUES (4, 'Roberto Carlos', '33444555', '07755544432');  
INSERT INTO titulares VALUES (5, 'André Barros', '44555666', '06655533321');
```

- Projetos Servi... x Arquivos
- Bancos de Dados
    - Java DB
    - Drivers
    - jdbc:derby://localhost:1527/sistema\_bancario
      - APP
        - Tabelas
        - Views
        - Procedimentos
        - Outros esquemas
  - Servidores
  - Repositórios Maven
  - Nuvem
  - Builders do Hudson
  - Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:1527/sistema\_bancario [app em APP]]

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
1 CREATE TABLE contas (  
2     nro_conta BIGINT NOT NULL,  
3     saldo DECIMAL(8,2),  
4     PRIMARY KEY (nro_conta)  
5 );  
6  
7 INSERT INTO contas VALUES (123, 500.00);  
8 INSERT INTO contas VALUES (124, 1000.00);  
9 INSERT INTO contas VALUES (125, 2500.00);  
10  
11 CREATE TABLE titulares (  
12     nro_titular BIGINT NOT NULL,  
13     nome VARCHAR(255) NOT NULL,  
14     rg VARCHAR(32) NOT NULL,  
15     cpf VARCHAR(32) NOT NULL,  
16     PRIMARY KEY (nro_titular)  
17 );  
18  
19 INSERT INTO titulares VALUES (2, 'Marcos Antônio', '22333444', '09988877765');  
20 INSERT INTO titulares VALUES (3, 'Rosimeire Aparecida', '11222333', '08866655543');  
21 INSERT INTO titulares VALUES (4, 'Roberto Carlos', '33444555', '07755544432');  
22 INSERT INTO titulares VALUES (5, 'André Barros', '44555666', '06655533321');
```



SQL 1 [jdbc:derby://localhost:15...] x

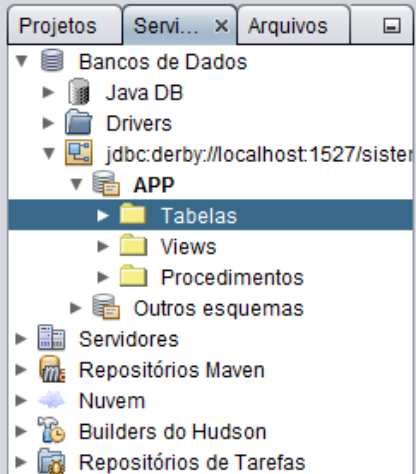
Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```

1 CREATE TABLE contas (
2     nro_conta BIGINT NOT NULL,
3     saldo DECIMAL(8,2),
4     PRIMARY KEY (nro_conta)
5 );
6
7 INSERT INTO contas VALUES (123, 500.00);
8 INSERT INTO contas VALUES (124, 1000.00);
9 INSERT INTO contas VALUES (125, 2500.00);
10
11 CREATE TABLE titulares (
12     nro_titular BIGINT NOT NULL,
13     nome VARCHAR(255) NOT NULL,
14     rg VARCHAR(32) NOT NULL,
15     cpf VARCHAR(32) NOT NULL,
16     PRIMARY KEY (nro_titular)
17 );
18
19 INSERT INTO titulares VALUES (2, 'Marcos
20 INSERT INTO titulares VALUES (3, 'Rosimeir
21 INSERT INTO titulares VALUES (4, 'Roberto
22 INSERT INTO titulares VALUES (5, 'André B

```

Navegar	
Formatar	Alt+Shift+F
<b>Executar Seleção</b>	
Histórico SQL	Ctrl+Alt+Shift
Executar Arquivo	Shift+F6
Recortar	Ctrl+X
Copiar	Ctrl+C
Colar	Ctrl+V
Selecionar no Projetos	
Selecionar Conexão em Serviços	



```
SQL 1 [jdbc:derby://localhost:1527/sistema_bancario [app em APP]]
Conexão: jdbc:derby://localhost:1527/sistema_bancario [app em APP]
1 CREATE TABLE contas (
2     nro_conta BIGINT NOT NULL,
3     saldo DECIMAL(8,2),
4     PRIMARY KEY (nro_conta)
5 );
6
7 INSERT INTO contas VALUES (123, 500.00);
8 INSERT INTO contas VALUES (124, 1000.00);
9 INSERT INTO contas VALUES (125, 2500.00);
10
11 CREATE TABLE titulares (
12     nro_titular BIGINT NOT NULL,
13     nome VARCHAR(255) NOT NULL;
```

Saída

AppSelectContas (clean) x Processo do Banco de Dados Java DB x execução SQL 1 x

Executado com sucesso em 0,172 s, 0 linhas afetadas.  
Linha 1, coluna 1

Executado com sucesso em 0,016 s, 1 linhas afetadas.  
Linha 9, coluna 1

Executado com sucesso em 0 s, 1 linhas afetadas.  
Linha 10, coluna 1

Executado com sucesso em 0,015 s, 1 linhas afetadas.  
Linha 11, coluna 1

Executado com sucesso em 0 s, 1 linhas afetadas.  
Linha 12, coluna 1

Execução finalizada após 0,203 s, 0 erro(s) encontrado(s).

← comandos executados com sucesso

Notificações Resultados do Teste Saída Resultados da Pesquisa Instrução(ões) SQL executada(s) com sucesso. 11:11/12/497 23/61 INS

# Criação da tabela **contas\_titulares** e inserção de dados iniciais



Digite os comandos abaixo:

```
CREATE TABLE contas_titulares (  
    nro_conta BIGINT NOT NULL,  
    nro_titular BIGINT NOT NULL,  
    PRIMARY KEY (nro_conta,nro_titular)  
);  
  
INSERT INTO contas_titulares VALUES (123,2);  
INSERT INTO contas_titulares VALUES (124,3);  
INSERT INTO contas_titulares VALUES (124,5);  
INSERT INTO contas_titulares VALUES (125,4);
```

- Projetos Servi... x Arquivos
- Bancos de Dados
    - Java DB
    - Drivers
    - jdbc:derby://localhost:1527/sistema\_bancario
      - APP
        - Tabelas
        - Views
        - Procedimentos
        - Outros esquemas
  - Servidores
  - Repositórios Maven
  - Nuvem
  - Builders do Hudson
  - Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:1527/sistema\_bancario]

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
3      saldo DECIMAL (8,2),
4      PRIMARY KEY (nro_conta)
5  );
6
7  INSERT INTO contas VALUES (123, 500.00);
8  INSERT INTO contas VALUES (124, 1000.00);
9  INSERT INTO contas VALUES (125, 2500.00);
10
11  CREATE TABLE titulares (
12      nro_titular BIGINT NOT NULL,
13      nome VARCHAR(255) NOT NULL,
14      rg VARCHAR(32) NOT NULL,
15      cpf VARCHAR(32) NOT NULL,
16      PRIMARY KEY (nro_titular)
17  );
18
19  INSERT INTO titulares VALUES (2, 'Marcos Antônio', '22333444', '09988877765');
20  INSERT INTO titulares VALUES (3, 'Rosimeire Aparecida', '11222333', '08866655543');
21  INSERT INTO titulares VALUES (4, 'Roberto Carlos', '33444555', '07755544432');
22  INSERT INTO titulares VALUES (5, 'André Barros', '44555666', '06655533321');
23
24  CREATE TABLE contas_titulares (
25      nro_conta BIGINT NOT NULL,
26      nro_titular BIGINT NOT NULL,
27      PRIMARY KEY (nro_conta, nro_titular)
28  );
29
30  INSERT INTO contas_titulares VALUES (123, 2);
31  INSERT INTO contas_titulares VALUES (124, 3);
32  INSERT INTO contas_titulares VALUES (124, 5);
33  INSERT INTO contas_titulares VALUES (125, 4);
34
```

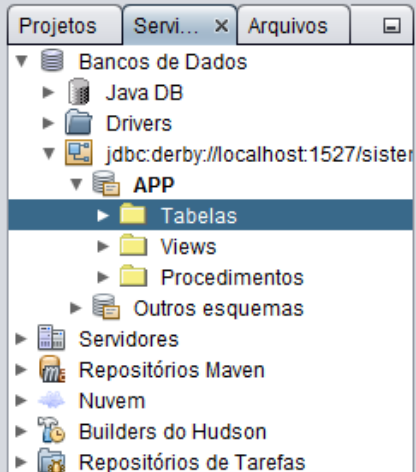
- Projetos Servi... x Arquivos
- Bancos de Dados
    - Java DB
    - Drivers
    - jdbc:derby://localhost:1527/sistema\_bancario
      - APP
        - Tabelas
        - Views
        - Procedimentos
        - Outros esquemas
  - Servidores
  - Repositórios Maven
  - Nuvem
  - Builders do Hudson
  - Repositórios de Tarefas

SQL 1 [jdbc:derby://localhost:1527/sistema\_bancario]

Conexão: jdbc:derby://localhost:1527/sistema\_bancario [app em APP]

```
3      saldo DECIMAL (8,2),
4      PRIMARY KEY (nro_conta)
5  );
6
7  INSERT INTO contas VALUES (123, 500.00);
8  INSERT INTO contas VALUES (124, 1000.00);
9  INSERT INTO contas VALUES (125, 2500.00);
10
11  CREATE TABLE titulares (
12      nro_titular BIGINT NOT NULL,
13      nome VARCHAR(255) NOT NULL,
14      rg VARCHAR(32) NOT NULL,
15      cpf VARCHAR(32) NOT NULL,
16      PRIMARY KEY (nro_titular)
17  );
18
19  INSERT INTO titulares VALUES (2, 'Marcos Antônio', '22333444', '09988877765');
20  INSERT INTO titulares VALUES (3, 'Rosimeire Aparecida', '11222333', '08866655543');
21  INSERT INTO titulares VALUES (4, 'Roberto Carlos', '33444555', '07755544432');
22  INSERT INTO titulares VALUES (5, 'André Barros', '44555666', '06655533321');
23
24  CREATE TABLE contas_titulares (
25      nro_conta BIGINT NOT NULL,
26      nro_titular BIGINT NOT NULL,
27      PRIMARY KEY (nro_conta, nro_titular)
28  );
29
30  INSERT INTO contas_titulares VALUES (123, 2);
31  INSERT INTO contas_titulares VALUES (124, 3);
32  INSERT INTO contas_titulares VALUES (124, 5);
33  INSERT INTO contas_titulares VALUES (125, 4);
34
```

Navegar	
Formatar	Alt+Shift+F
Executar Seleção	
Histórico SQL	Ctrl+Alt+Shift+H
Executar Arquivo	Shift+F6
Recortar	Ctrl+X
Copiar	Ctrl+C
Colar	Ctrl+V
Selecionar no Projetos	
Selecionar Conexão em Serviços	



```
SQL 1 [jdbc:derby://localhost:1527/sistema_bancario [app em APP]]
Conexão: jdbc:derby://localhost:1527/sistema_bancario [app em APP]
3      saldo DECIMAL (8,2),
4      PRIMARY KEY (nro_conta)
5  );
6
7  INSERT INTO contas VALUES (123, 500.00);
8  INSERT INTO contas VALUES (124, 1000.00);
9  INSERT INTO contas VALUES (125, 2500.00);
10
11 CREATE TABLE titulares (
12     nro_titular BIGINT NOT NULL,
13     nome VARCHAR(255) NOT NULL,
14     rg VARCHAR(32) NOT NULL,
15     cpf VARCHAR(32) NOT NULL
```

Saída

AppSelectContas (clean) x Processo do Banco de Dados Java DB x execução SQL 1 x

Executado com sucesso em 0,329 s, 0 linhas afetadas.  
Linha 1, coluna 1

Executado com sucesso em 0 s, 1 linhas afetadas.  
Linha 7, coluna 1

Executado com sucesso em 0,016 s, 1 linhas afetadas.  
Linha 8, coluna 1

Executado com sucesso em 0 s, 1 linhas afetadas.  
Linha 9, coluna 1

Executado com sucesso em 0,079 s, 1 linhas afetadas.  
Linha 10, coluna 1




Execução finalizada após 0,424 s, 0 erro(s) encontrado(s).

← comandos executados com sucesso

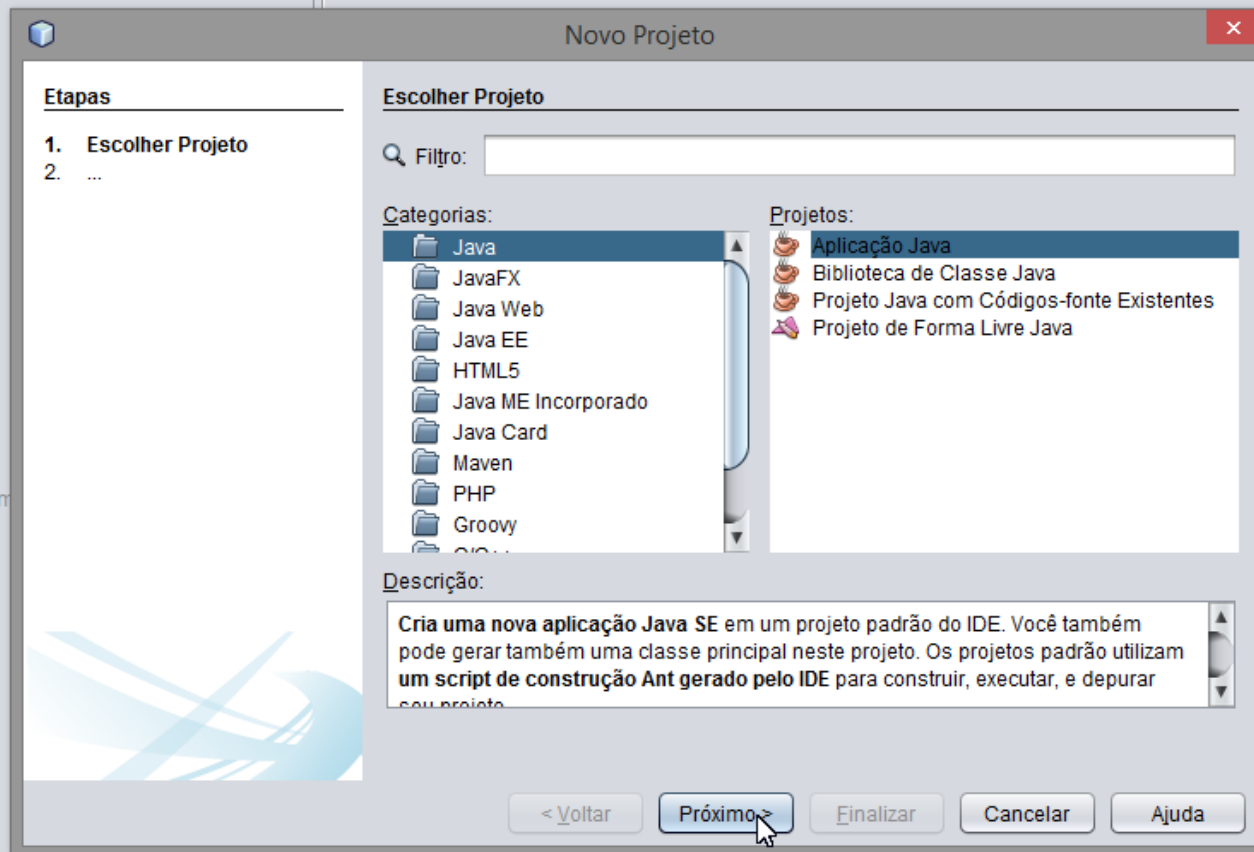
28 / 61

Notificações Resultados do Teste Saída Resultados da Pesquisa Instrução(ões) SQL executada(s) com sucesso. 24:11:31

# Criação de projeto JAVA e inclusão da biblioteca com o driver Java DB

-  Novo Projeto... Ctrl+Shift+N
-  Novo Arquivo... Ctrl+N
-  Abrir Projeto... Ctrl+Shift+O
  - Abrir Projeto Recente ▶
  - Fechar Projeto
  - Fechar Outros Projetos
  - Fechar Todos os Projetos
  - Abrir Arquivo...
  - Abrir Arquivo Recente ▶
- Grupos de Projetos...
- Propriedades do Projeto ()
- Importar Projeto ▶
- Exportar Projeto ▶
- Salvar Ctrl+S
  - Salvar como...
  - Salvar Tudo Ctrl+Shift+S
- Configurar Página...
- Imprimir... Ctrl+Alt+Shift+P
- Imprimir em HTML...
- Sair

Projetos x Serviços Arquivos



Projetos x Serviços Arquivos

selecione a pasta  
onde gravar o  
projeto



## Novo Aplicação Java

## Etapas

1. Escolher Projeto
2. Nome e Localização

## Nome e Localização

Nome do Projeto: SistemaBancarioComDao

Localização do Projeto: C:\Users\Tomaz\Dropbox\\_Docs\@Archive\2015\MACK-LP2-teoria05\netbeans

Procurar...

Pasta do Projeto: \_Docs\@Archive\2015\MACK-LP2-teoria05\netbeans\SistemaBancarioComDao

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

Pasta Bibliotecas:

Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

☐ Criar Classe Principal sistemabancariocomdao.SistemaBancarioComDao

&lt; Voltar

Próximo &gt;

Finalizar

Cancelar

Ajuda



Notificações

Resultados do Teste

Saída

Resultados da Pesquisa

1



Projetos x Serviços Arquivos



## Novo Aplicação Java



## Etapas

1. Escolher Projeto
2. **Nome e Localização**

## Nome e Localização

Nome do Projeto: SistemaBancarioComDao

Localização do Projeto: .\Users\Tomaz\Dropbox\\_Docs\@Archive\2015\MACK-LP2-lab05-dao\netbeans

Procurar...

Pasta do Projeto: ocs\@Archive\2015\MACK-LP2-lab05-dao\netbeans\SistemaBancarioComDao

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

Pasta Bibliotecas:

Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

☐ Criar Classe Principal sistemabancariocomdao.SistemaBancarioComDao

&lt; Voltar

Próximo &gt;

Finalizar

Cancelar

Ajuda



Notificações

Resultados do Teste

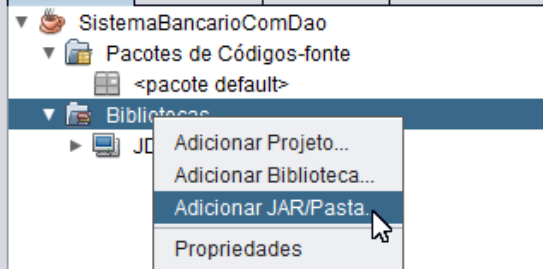
Saída

Resultados da Pesquisa



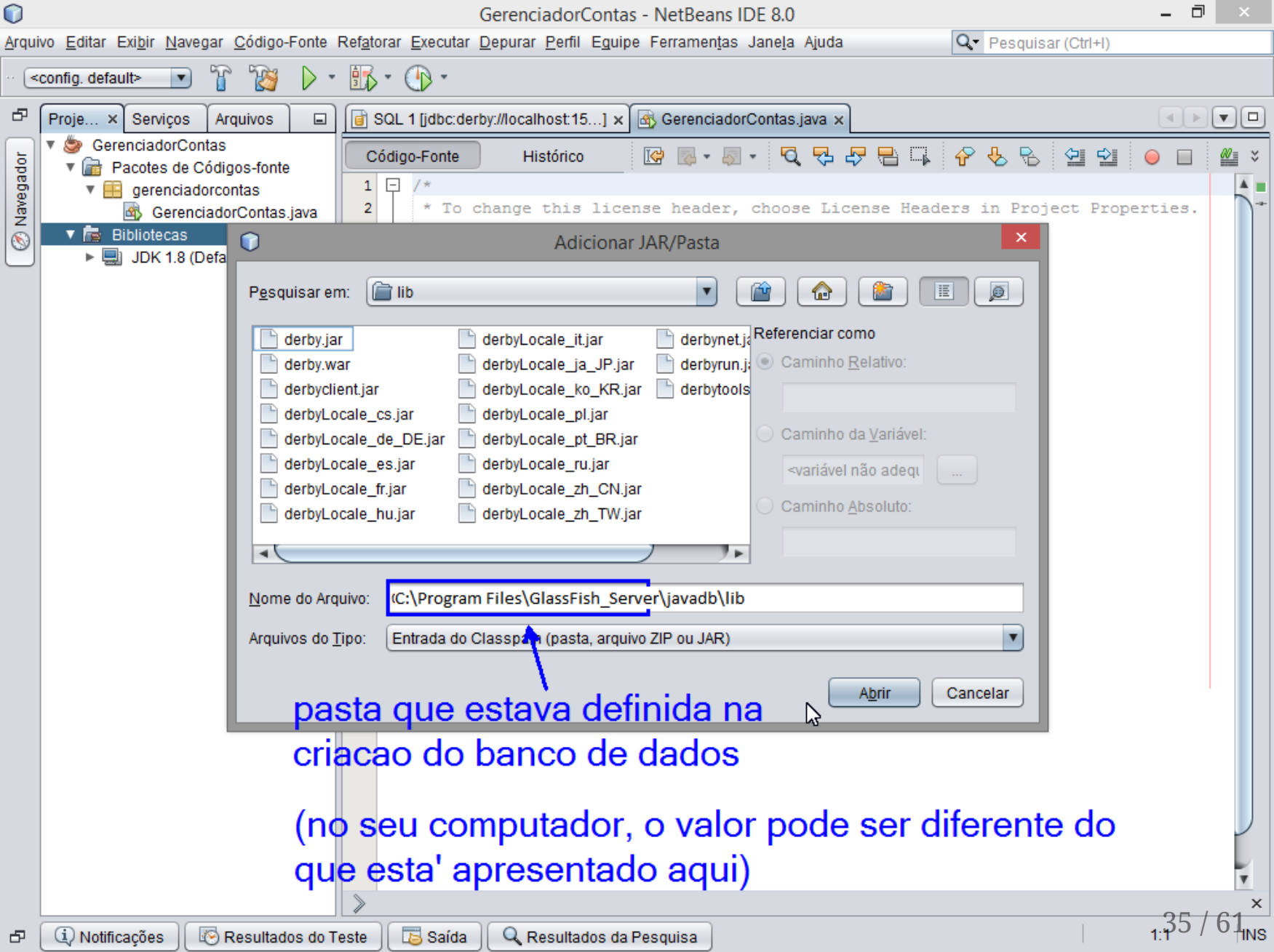
&lt;config. default&gt;

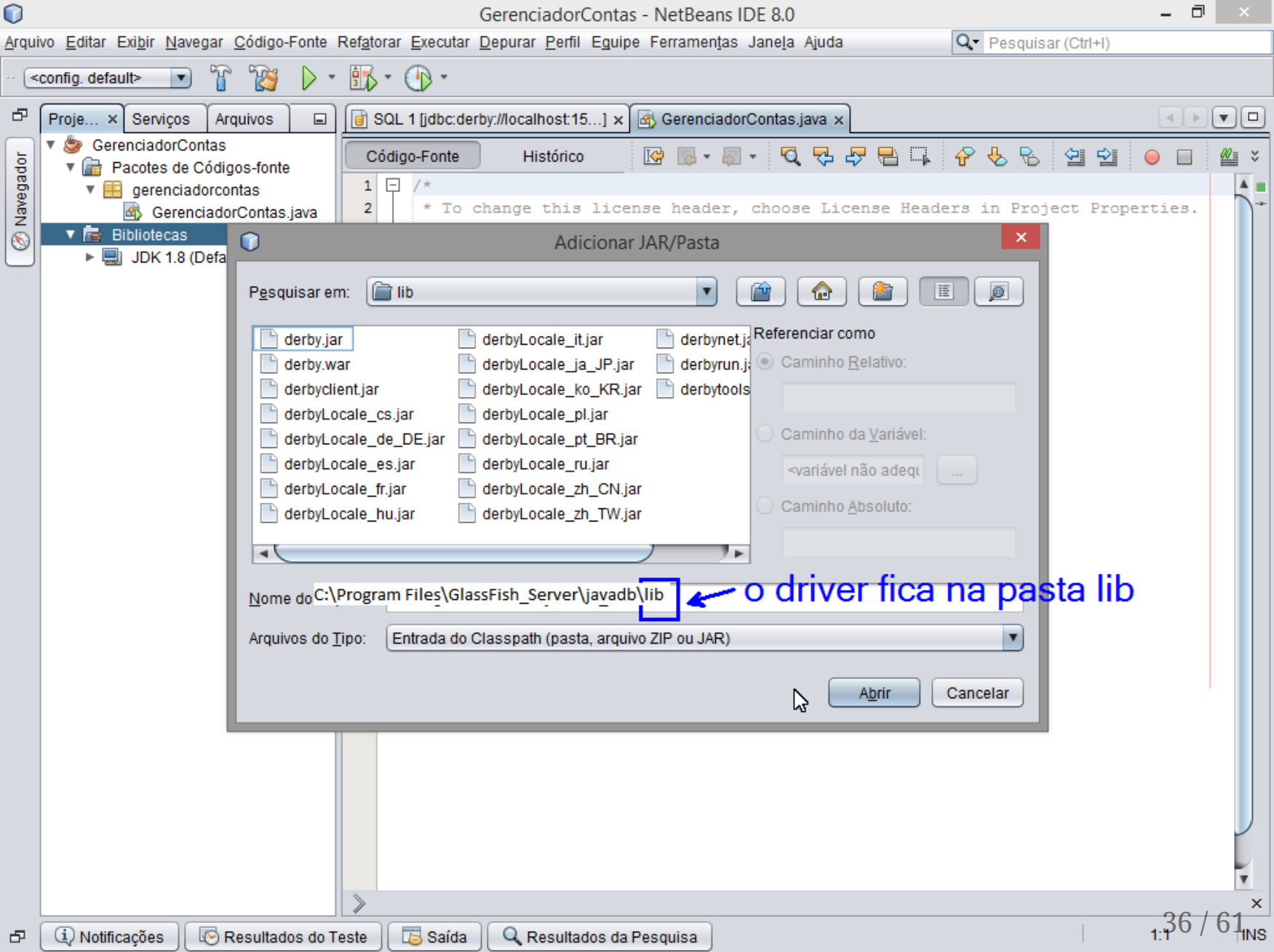
Projetos x Serviços Arquivos



run - Navegador x

&lt;Nenhuma View Disponível&gt;





&lt;config. default&gt;



Projetos x Serviços Arquivos

- ▼ SistemaBancarioComDao
  - ▼ Pacotes de Códigos-fonte
    - <pacote default>
  - ▼ Bibliotecas
    - ▶ derbyclient.jar
    - ▶ JDK 1.8 (Default)

run - Navegador x

&lt;Nenhuma View Disponível&gt;



Notificações



Resultados do Teste



Saída



Resultados da Pesquisa

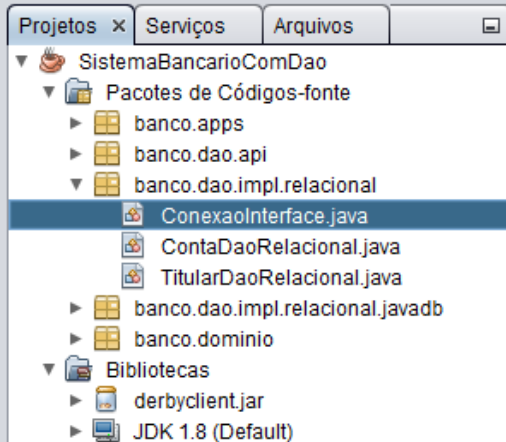


- ▼ SistemaBancarioComDao
  - ▼ Pacotes de Código-fonte
    - ▶ banco.apps
    - ▶ banco.dao.api
    - ▶ banco.dao.impl.relacional
    - ▶ banco.dao.impl.relacional.javadb
    - ▶ banco.dominio
  - ▼ Bibliotecas
    - ▶ derbyclient.jar
    - ▶ JDK 1.8 (Default)

crie a estrutura de pacotes

- ▼ AppSelectContasComDao
  - main(String[] args)

# Criação da interface de conexão



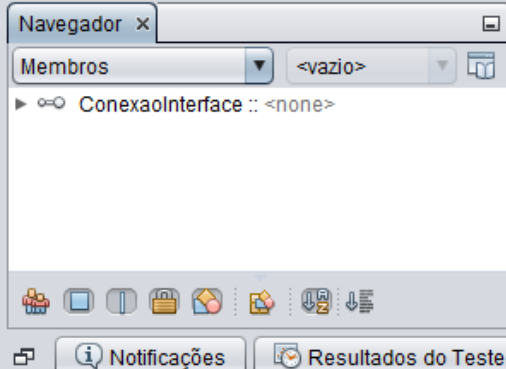
crie a interface  
ConexaoInterface

ConexaoInterface.java x

Código-Fonte

Histórico

```
1 package banco.dao.impl.relacional;
2
3 import java.sql.Connection;
4
5 public interface ConexaoInterface {
6     Connection getConnection();
7     void close();
8 }
9
```





# Criação da implementação da conexão ao Java DB

Projetos x Serviços Arquivos

- SistemaBancarioComDao
  - Pacotes de Códigos-fonte
    - banco.apps
    - banco.dao.api
    - banco.dao.impl.relacional
    - banco.dao.impl.relacional.javadb
      - ConexaoJavaDb.java
  - banco.dominio
  - Bibliotecas
    - derbyclient.jar
    - JDK 1.8 (Default)

crie a classe  
ConexaoJavaDb

ConexaoJavaDb.java x

```

8 public class ConexaoJavaDb implements ConexaoInterface {
9     private String usuario;
10    private String senha;
11    private String hostname;
12    private int porta;
13    private String nomeBancoDados;
14    private Connection conexao;
15    public ConexaoJavaDb(String usuario, String senha, String hostname, int porta, String n) {
16        this.usuario = usuario;
17        this.senha = senha;
18        this.hostname = hostname;
19        this.porta = porta;
20        this.nomeBancoDados = n;
21    }
22    @Override
23    public Connection getConnection() {
24        if (conexao == null) {
25            try {
26                Class.forName("org.apache.derby.jdbc.ClientDriver");
27                String url = "jdbc:derby://" + hostname + ":" + porta + "/" + nomeBancoDados;
28                conexao = DriverManager.getConnection(url, usuario, senha);
29            } catch (Exception ex) {
30                ex.printStackTrace();
31            }
32        }
33        return conexao;
34    }
35    @Override
36    public void close() {
37        try {
38            conexao.close();
39        } catch (SQLException ex) {
40            ex.printStackTrace();
41        }
42        conexao = null;
43    }
44 }
45

```

ConexaoJavaDb - Navegador x

Membros <vazio>

- ConexaoJavaDb :: ConexaoInterface
  - ConexaoJavaDb(String usuario, Str
  - close()
  - getConnection() : Connection
  - conexao : Connection
  - hostname : String

# Criação da classe de domínio **Conta**

Projetos x Serviços Arquivos

SistemaBancarioComDao

- Pacotes de Códigos-fonte
  - banco.apps
  - banco.dao.api
  - banco.dao.impl.relacional
  - banco.dao.impl.relacional.javadb
    - ConexaoJavaDb.java
  - banco.dominio
    - Conta.java
    - Titular.java
- Bibliotecas
  - derbyclient.jar
  - JDK 1.8 (Default)

crie a classe  
Conta

Conta.java x

```
1 package banco.dominio;
2
3 import java.math.BigDecimal;
4
5 public class Conta {
6     private long numero;
7     private BigDecimal saldo;
8
9     public Conta(long numero, BigDecimal saldo) {
10         this.numero = numero;
11         this.saldo = saldo;
12     }
13
14     public long getNumero() {
15         return numero;
16     }
17
18     public void setNumero(long numero) {
19         this.numero = numero;
20     }
21
22     public BigDecimal getSaldo() {
23         return saldo;
24     }
25
26     public void setSaldo(BigDecimal saldo) {
27         this.saldo = saldo;
28     }
29
30 }
31
```

Navegador x

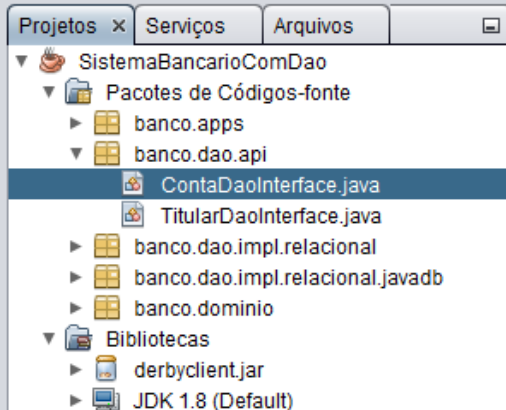
Membros <vazio>

Conta

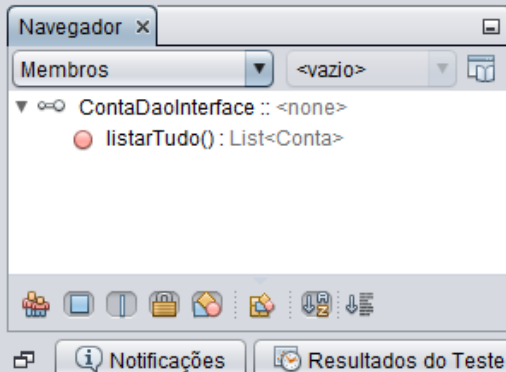
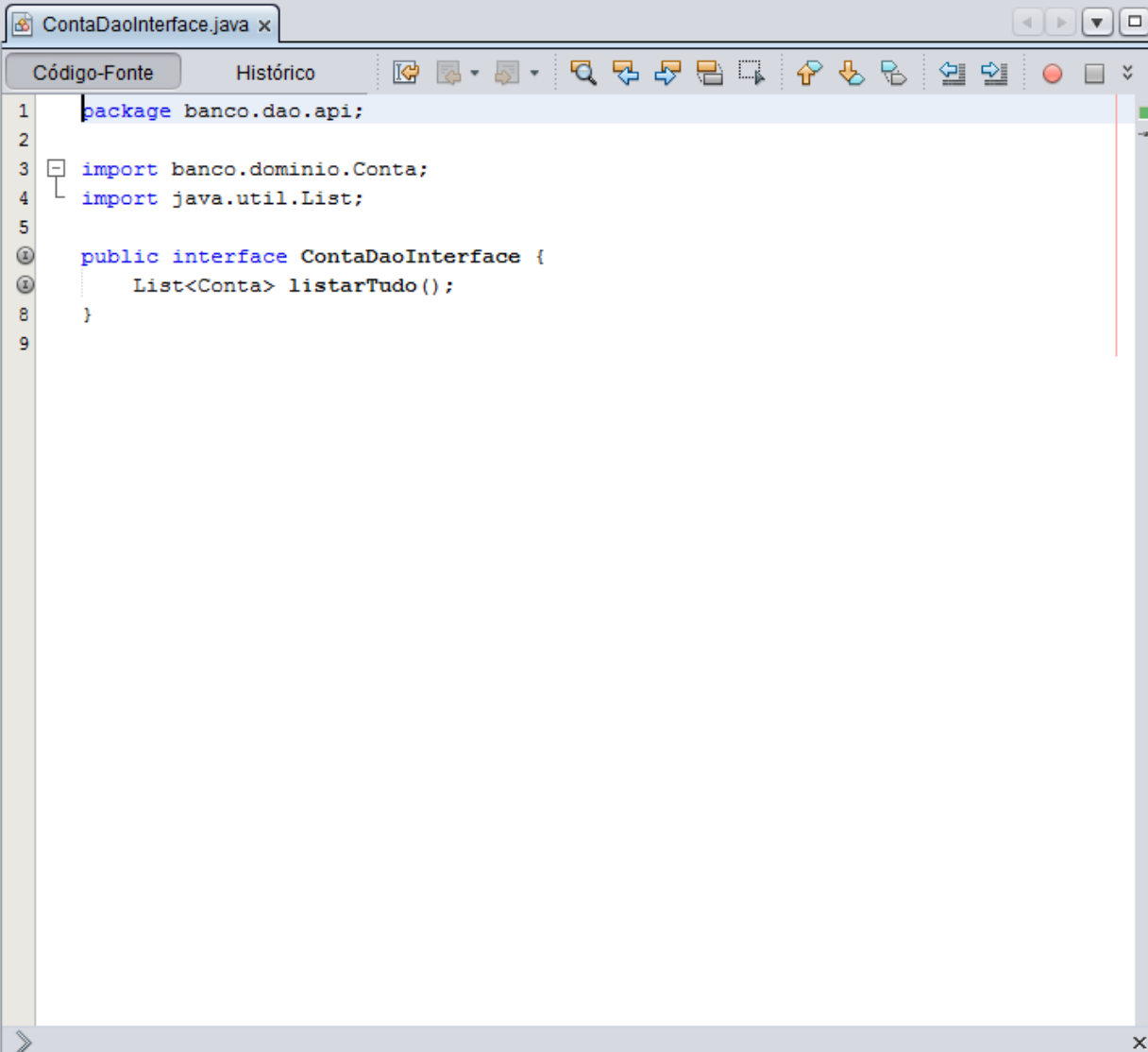
- Conta(long numero, BigDecimal saldo)
- getNumero() : long
- getSaldo() : BigDecimal
- setNumero(long numero)
- setSaldo(BigDecimal saldo)

Notificações Resultados do Teste Saída Resultados da Pesquisa

# Criação da interface com as operações de acesso aos dados das contas



crie a interface  
ContaDaoInterface



# Implementação das operações de acesso aos dados das contas



Projetos x Arquivos Serviços

SistemaBancarioComDao

- Pacotes de Códigos-fonte
  - banco.apps
  - banco.dao.api
  - banco.dao.impl.relacional
    - ContaDaoRelacional.java
    - TitularDaoRelacional.java
  - banco.dao.impl.relacional.javadb
  - banco.dominio
- Bibliotecas

criar a classe  
ContaDaoRelacional

listarTudo - Navegador x

Membros <vazio>

ContaDaoRelacional :: ContaDaoInterface

- ContaDaoRelacional(ConexaoInterface c)
- listarTudo(): List<Conta>
- conexao: ConexaoInterface

AppSelectContasComDao.java x ContaDaoInterface.java x ContaDaoRelacional.java x

```

10
11 public class ContaDaoRelacional implements ContaDaoInterface {
12
13     private ConexaoInterface conexao;
14
15     public ContaDaoRelacional(ConexaoInterface conexao) {
16         this.conexao = conexao;
17     }
18
19     @Override
20     public List<Conta> listarTudo() {
21         List<Conta> contas;
22         contas = new ArrayList<>();
23         try {
24             Statement st;
25             st = conexao.getConnection().createStatement();
26             String sql = "SELECT nro_conta, saldo FROM contas";
27             ResultSet resultados = st.executeQuery(sql);
28
29             while (resultados.next()) {
30                 long n = resultados.getLong("nro_conta");
31                 BigDecimal b = resultados.getBigDecimal("saldo");
32                 Conta c = new Conta(n, b);
33                 contas.add(c);
34             }
35
36             } catch (Exception ex) {
37                 ex.printStackTrace();
38             }
39         return contas;
40     }
41
42 }
43

```

Notificações Resultados do Teste

Saída

Resultados da Pesquisa

ContaDaoRelacional.java salvo.





**Criação da aplicação para listar todas as contas**



Projetos x Arquivos Serviços

- SistemaBancarioComDao
  - Pacotes de Códigos-fonte
    - banco.apps
      - AppSelectContasComDao.java**
      - banco.dao.api
      - banco.dao.impl.relacional
      - banco.dao.impl.relacional.javadb
      - banco.dominio
    - Bibliotecas

crie a classe  
AppSelectContasComDao

main - Navegador x

Membros

- AppSelectContasComDao
  - main(String[] args)

AppSelectContasComDao.java x

```
1 package banco.apps;
2
3 import banco.dao.api.ContaDaoInterface;
4 import banco.dao.impl.relacional.ConexaoInterface;
5 import banco.dao.impl.relacional.ContaDaoRelacional;
6 import banco.dao.impl.relacional.javadb.ConexaoJavaDb;
7 import banco.dominio.Conta;
8 import java.util.List;
9
10 public class AppSelectContasComDao {
11     public static void main(String[] args) {
12         ConexaoInterface conexao;
13         conexao = new ConexaoJavaDb("app", "app", "127.0.0.1", 1527, "sistema_
14         ContaDaoInterface dao;
15         dao = new ContaDaoRelacional(conexao);
16         List<Conta> todasContas;
17         todasContas = dao.listarTudo();
18         for (Conta c: todasContas) {
19             System.out.print("Nro: " + c.getNumero());
20             System.out.print(" - ");
21             System.out.println("Saldo: R$ " + c.getSaldo());
22         }
23         conexao.close();
24     }
25 }
26
27
```

# Execução da aplicação

Projetos x Serviços Arquivos

▼ SistemaBancarioComDao

- ▼ Pacotes de Códigos-fonte
  - ▼ banco.apps
    - AppSelectContasComDao.java
- ▼ Bibliotecas
  - derbyclient.jar
  - JDK 1.8 (Default)

Abrir	
Recortar	Ctrl+X
Copiar	Ctrl+C
Colar	Ctrl+V
Compilar Arquivo	F9
Executar Arquivo	Shift+F6
Depurar Arquivo	Ctrl+Shift+F5
Arquivo de Perfil	
Testar Arquivo	Ctrl+F6
Depurar Teste Arquivo	Ctrl+Shift+F6
Arquivo de Teste do Perfil	

Adicionar	
Deletar	Deletar
Salvar como Modelo...	

Localizar Usos	Alt+F7
Refatorar	►

Editor BeanInfo...

Membros do Arquivo	Ctrl+F12
Hierarquia do Arquivo	Alt+F12

Histórico	►
Ferramentas	►
Propriedades	

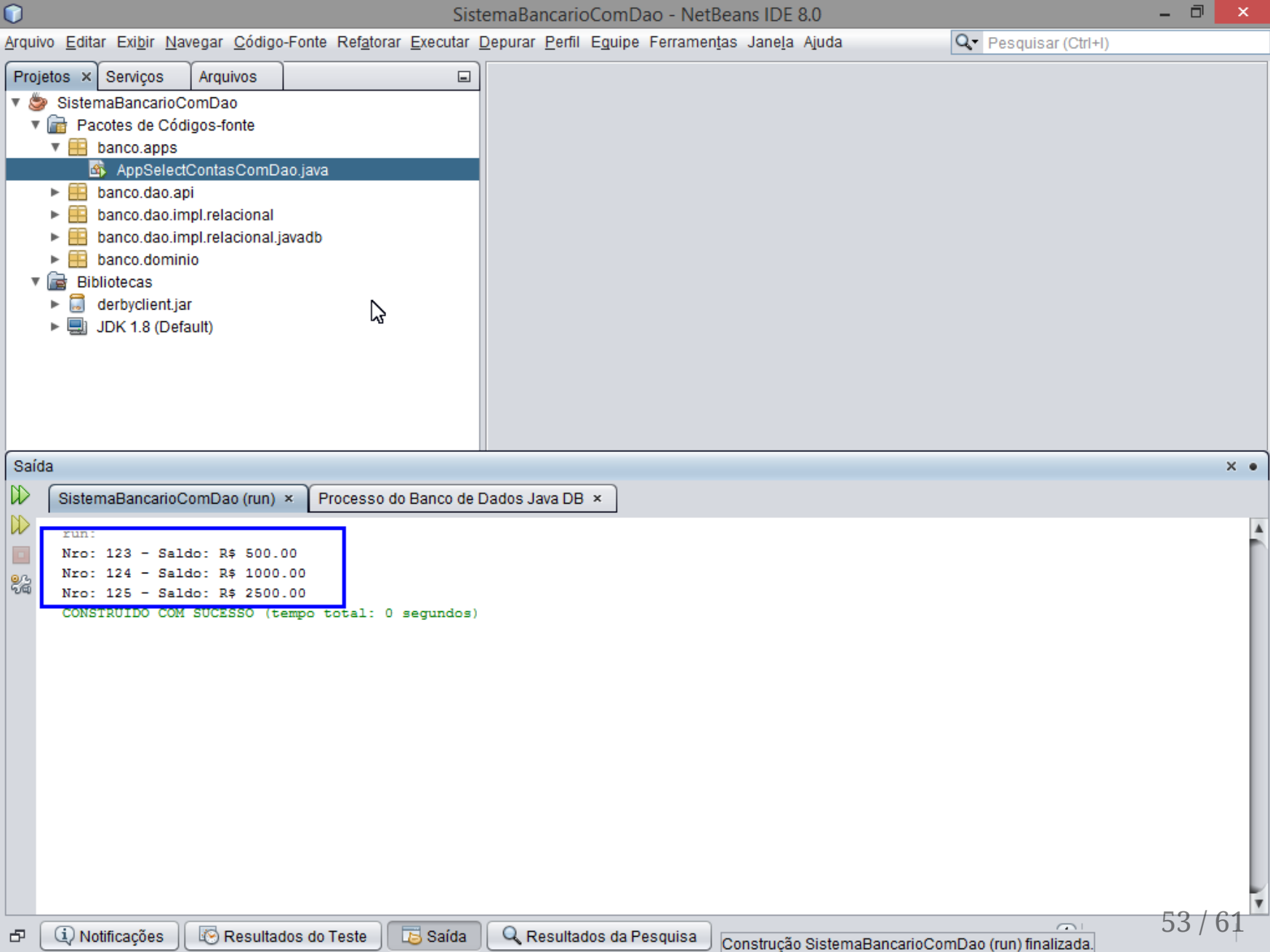
Navegador x

Membros &lt;v&gt;

▼ AppSelectContasComDao

- main(String[] args)





AppSelectContasComDao.java x

```
1 package banco.apps;
2
3 import banco.dao.api.ContaDaoInterface;
4 import banco.dao.impl.relacional.ConexaoInterface;
5 import banco.dao.impl.relacional.ContaDaoRelacional;
6 import banco.dao.impl.relacional.javadb.ConexaoJavaDb;
7 import banco.dominio.Conta;
8 import java.util.List;
9
10 public class AppSelectContasComDao {
11     public static void main(String[] args) {
12         ConexaoInterface conexao;
13         conexao = new ConexaoJavaDb("app", "app", "127.0.0.1", 1527, "sistema_bancario");
14         ContaDaoInterface dao;
15         dao = new ContaDaoRelacional(conexao);
16         List<Conta> todasContas;
17         todasContas = dao.listarTudo();
18         for (Conta c: todasContas) {
19             System.out.print("Nro: " + c.getNumero());
20             System.out.print(" - ");
21             System.out.println("Saldo: R$ " + c.getSaldo());
22         }
23         conexao.close();
24     }
25 }
26
27
```

Note que a AppSelectContasComDao não menciona diretamente nenhum item da API JDBC

# Exercícios



## Exercício 1

O tutorial apresentou um exemplo de utilização do padrão conhecido como DAO (Data Access Object). Seguindo os mesmos passos do tutorial, crie as interfaces e classes necessárias para listar os dados dos **titulares**.

Ao final, crie uma classe executável (com o nome **AppSelectTitularesComDao**) que consiga listar os dados de todos os titulares.



## Exercício 2

Altere a interface **ContaDaoInterface** de acordo com o código abaixo:

```
public interface ContaDaoInterface {  
    List<Conta> listarTudo();  
  
    // insere uma nova conta  
    void adicionar(Conta c);  
  
    // apaga uma conta  
    void remover(Conta c);  
  
    // atualiza os dados de uma conta já existente  
    void atualizar(Conta c);  
  
    // obtém a instância de Conta que possui o número informado  
    Conta buscarPeloNumero(long nroConta);  
  
    // mais operações  
}
```

## Exercício 3

Altere a classe **ContaDaoRelacional** de forma a implementar os novos métodos da interface.

Nesta implementação, preocupe-se em manipular apenas a tabela **contas**.

## Exercício 4

Desenvolva uma classe **GerenciarContas** que seja executável (ou seja, que tenha o método main implementado). A sua execução deverá apresentar um menu principal para o usuário com as opções:

- (1) Listar todas as contas.
- (2) Adicionar uma nova conta.
- (3) Remover uma conta.
- (4) Atualizar os dados de uma conta.
- (5) Sair

Todas as opções, com a exceção da opção (5), executam a sua função e voltam a apresenta o menu principal.

Na opção listar, o programa deverá imprimir as informações de todas as contas que estão cadastradas (uma linha para cada sala).

Na opção adicionar, o programa deverá solicitar o valor para cada campo da conta e inserir um novo registro na tabela.

(continua)

## **Exercício 4 (cont.)**

Na opção remover, o programa deverá solicitar o número da conta a ser apagada e, a seguir, removê-la da tabela.

Na opção atualizar, o programa deverá solicitar o número da conta a ser atualizada, solicitar os novos valores para os demais campos e, a seguir, atualizar a tabela.

Na opção sair, o programa deverá finalizar a execução.

**O programa deverá necessariamente utilizar as interfaces e classes DAO criadas nesta aula.**

Bom estudo!

