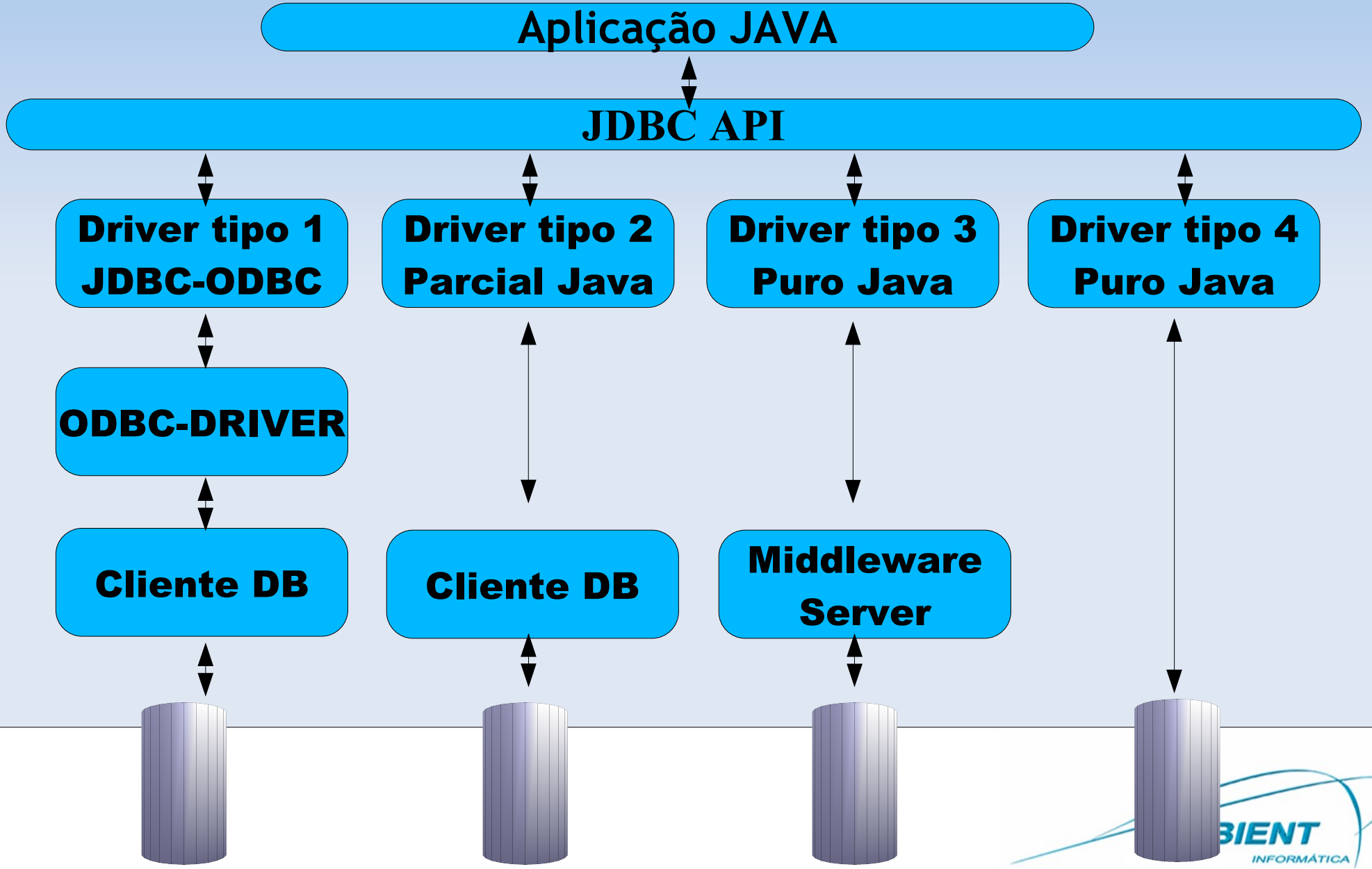
Aula 09

JDBC

Linhas Gerais

- JDBC – Java Database Connectivity
- Acesso a banco de dados;
- http://java.sun.com/jdbc
- Driver
- SQL;
- Classpath

Tipos de JDBC



Conectando no Banco de Dados

- Class.forName(“driver”);
- Lança exceção ClassNotFoundException;
- Url do banco de dados:
- jdbc:interbase://...
- jdbc:<subProtocolo>:<subNome>
- Connection c = DriverManager.getConnection(..);

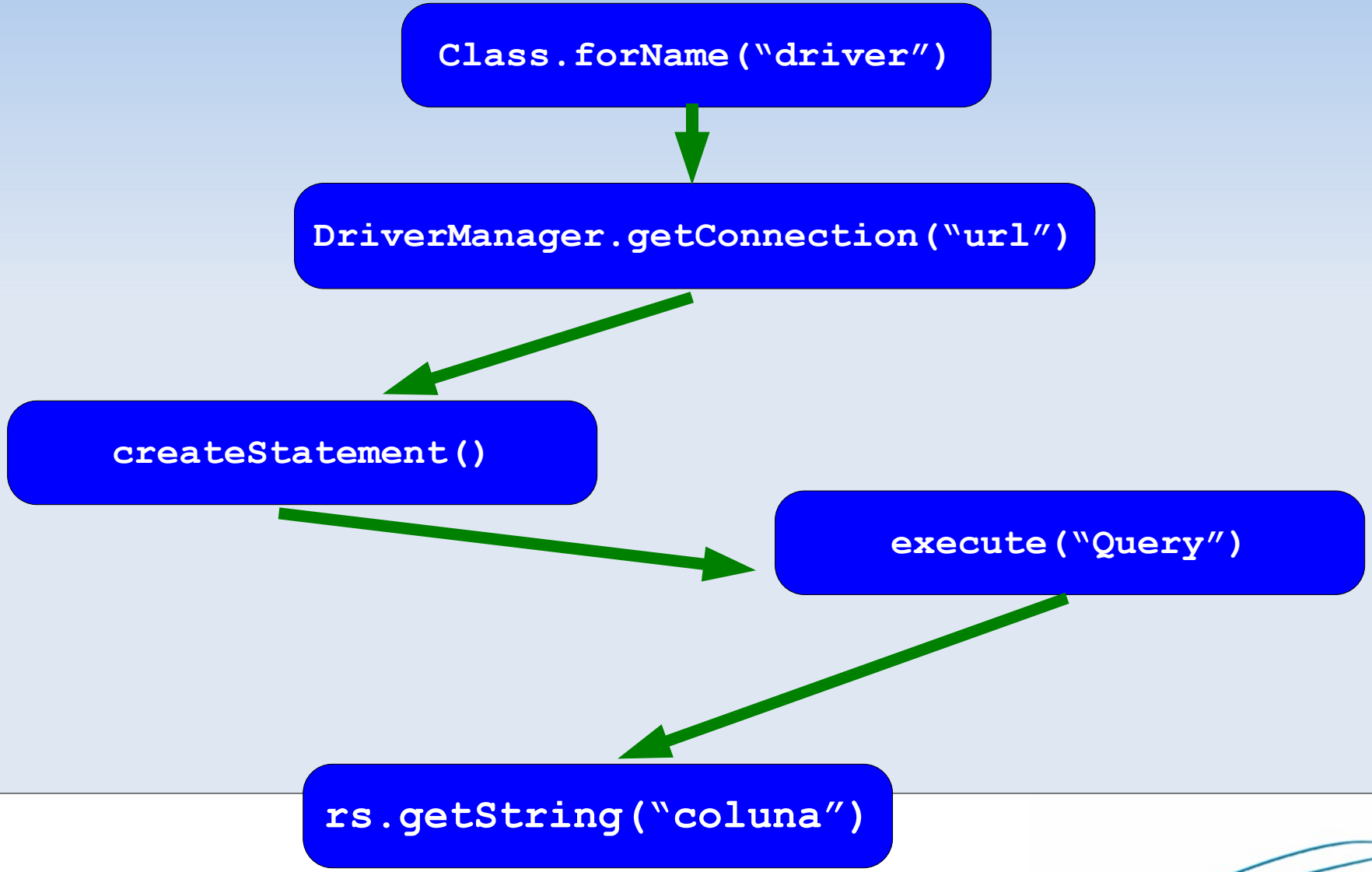
Enviando SQL's ao BD

- Statement stm = connection.createStatement()
- PreparedStatement stm = connection.prepareStatement(SQL)
- CallableStatement stm = connection.prepareCall(SQL)

Processando os dados

- ResultSet rs = statement....
- rs.getString();
- rs.getInt();
- rs.next()
- ResultSetMetaData rsmd = rs.getMetaData();
- rsmd.getColumnName();

... em resumo



SQL

- SELECT * FROM nomeDaTabela
- INSERT INTO nomeDaTabela (campo, campo2) VALUES (valor1, valor2)
- DELETE FROM nomeDaTabela WHERE campo = valor
- UPDATE INTO nomeDaTabela campo=valor WHERE campo=valor1

SELECT

- SELECT * FROM nomeDaTabela
- SELECT campo1, campo2 FROM nomeDaTabela

WHERE

- Especifica os critérios de seleção para a consulta;
- Suporta: =, >, <, LIKE
- LIKE com caracteres curingas % e _
- R% - todos os registros que iniciam com a letra R serão selecionados;
- _e% - indica um único curinga na sua posição;

INNER JOIN

- Mescla dados de múltiplas tabelas em um único resultado;
- ... INNER JOIN tabela2 ON tabela1.coluna = tabela2.coluna

INSERT

- Insere dados em uma tabela;
- INSERT INTO nomeDaTabela (coluna1, coluna2) VALUES (valor1, valor2)

UPDATE

- Altera dados em uma tabela;
- UPDATE nomeDaTabela SET coluna = valor WHERE coluna = valorAntigo

DELETE

- Exclui dados de uma tabela;
- DELETE FROM nomeDaTabela WHERE coluna = valorAntigo

Fechando as conexões

- statement.close();
- connection.close();

PreparedStatement

- Simplifica a representação dos parâmetros;
- Utiliz amos ? Para representar os parâmetros variáveis;
- Suporta updates(INSERT, UPDATE, DELETE) em batch;
  - Ex.: “SELECT * FROM tabela WHERE valor = ?”
  - pstm.setXXX(i, valor)
  - pstm.executeUpdate() ou
  - rs = pstm.executeQuery();

Execução em Batch

- Updates em batch:
- pstmt.setString(2, “String”);
- pstmt.addBatch();
- pstmt.executeBatch();

Transação

- ACID:
  - A – Atomicidade
  - C – Consistência
  - I – Isolação
  - D - Durabilidade

Transação: Atomicidade

- Trata o trabalho como parte indivisível (atômico). A transação deve ter todas as suas operações executadas em caso de sucesso ou nenhum resultado de alguma operação refletido sobre a base de dados em caso de falha.
- Commit
- RollBack

Transação: Consistência

- Regras de integridade dos dados são asseguradas, ou seja, as transações não podem quebrar as regras do Banco de Dados.

Transação: Isolamento

- Tudo se parece como se o trabalho estivesse isolado. O resultado de uma transação executada concorrentemente a outra deve ser o mesmo que o de sua execução de forma isolada. Operações exteriores a uma dada transação jamais verão esta transação em estados intermediários.

Transação: Durabilidade

- Os efeitos de uma transação em caso de sucesso (commit) são permanentes mesmo em presença de falhas.

Transação

- Connection.setAutoCommit(false);
- Connection.commit();
- Connection.rollback();

DÚVIDAS

AMBIENT
INFORMÁTICA

EXERCÍCIOS

- O que é JDBC?
- Quantos tipos de drivers JDBC existem ? Qual a diferença entre eles?
- Qual a última versão do JDBC existente?
- Qual o procedimento para conectar em um banco de dados?
- Qual a diferença dos métodos executeUpdate e executeQuery ?
- Como passar parâmetros para query?

FIM

AMBIENT
INFORMÁTICA