



Projeto de Lógica Especificações JML

por Paulo Vitor, Rita Lopes e Rita Cruz

Objeto de estudo

PROJETO BIBLIOTECA

- Criado por João Paulo F. da Silva
- Sistema de gerenciamento de biblioteca
- Java
- Disponível em: <https://github.com/paulojp-dev/projeto-biblioteca-java>

About

Software para biblioteca em java, simples. Sistema desenvolvido a fim de obter nota para a matéria de Programação OO do 3^a período de Análise e Desenvolvimento de Sistemas..

 Readme

 26 stars

 0 watching

 15 forks

Modelo do BD

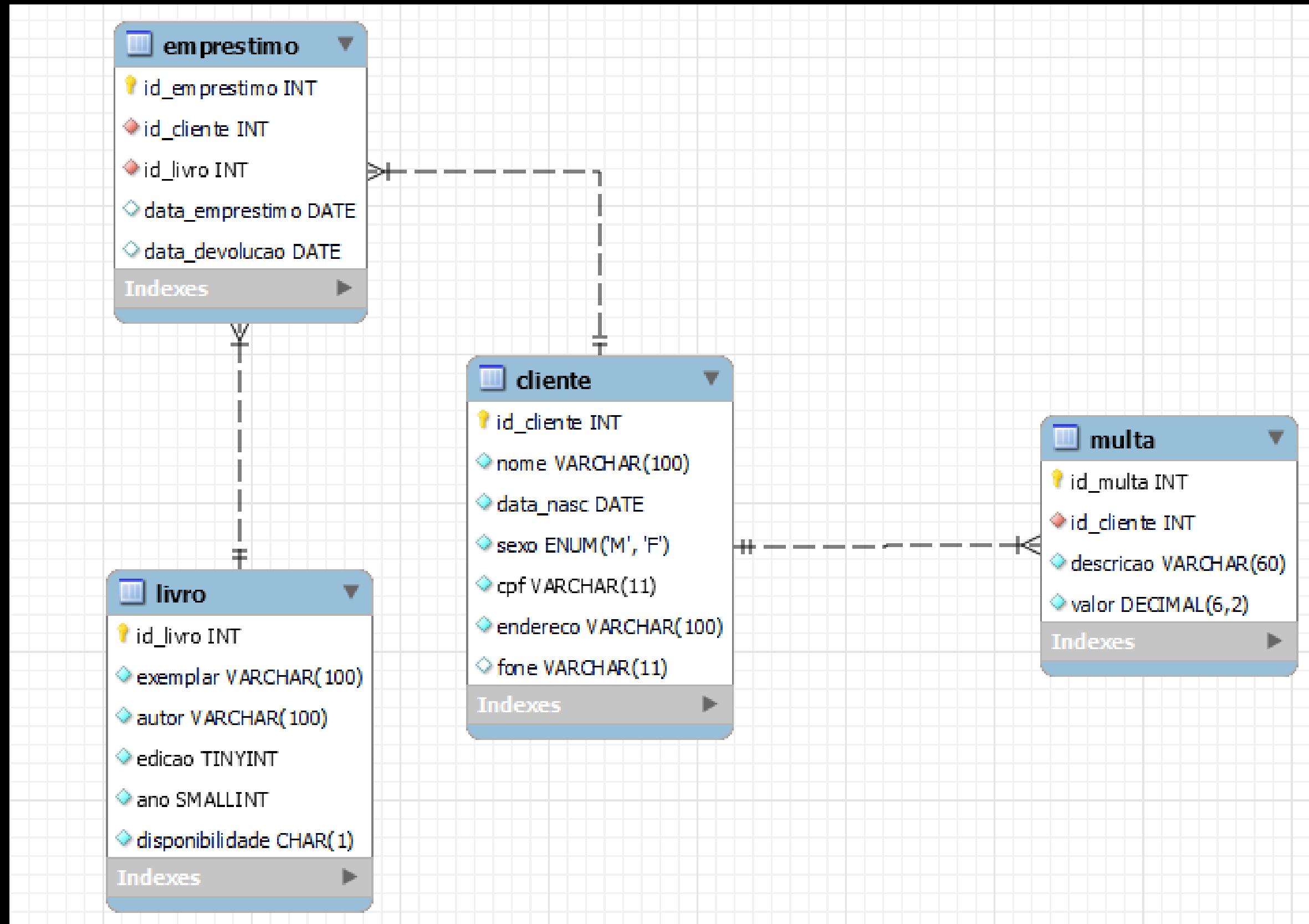
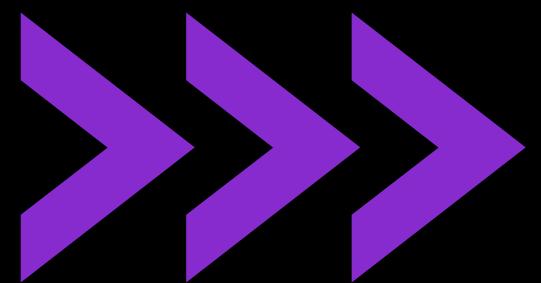


Diagrama de Classes



JFEmprestimo	JFMulta	JFCliente	JFLivro	Cliente	JFPrincipal	Livro
<pre> jT2dLivro JTextField jbPesquisar JButton jLabel4 JLabel jLabel5 JRadioButton telaPrincipal JFPrincipal tmEmprestimo DefaultTableModel jLabel6 JLabel jScrollPane2 JScrollPane jT01Emprestimo JTextField jTableCliente JTable verifica boolean jLabel11 JLabel jLabel2 JRadioButton jScrollPane3 JScrollPane jPanel1 JPanel livros List<Livro> jbSair JButton jLabel10 JLabel jLabel1 JLabel jLabel7 JLabel emprestimos List<Emprestimo> jTableEmprestimo JTable jLabel5 JLabel jScrollPane1 JScrollPane clientes List<Cliente> JT1dCliente JTextField jPanel4 JPanel jLabel8 JLabel jT4DataDevolucao JTextField bGPesquisa JButtonGroup jT3dataEmprestimo JTextField tmCliente DefaultTableModel jPanel2 JPanel jTableLivro JTable jbNovo JButton jClients JRadioButton jBcadstrar JButton jLabel3 JLabel jExcluir JButton jLabel9 JLabel jTpesquisar JTextField tmLivro DefaultTableModel enviaValor JFMulti jbDevolver JButton cadastroRegistro() void main(String) void pegalCliente() String mostraPesquisaEmprestimo(List<Emprestimo>) void desabilitaCamposEmprestimo() void diferecaData() long jbDevolverMouseClicked(MouseEvent) void jTableClientemouseClicked(MouseEvent) void salvaDataEmprestimo() String limpaCampoEmprestimo() void listaContatosCliente() void mostraDataDevolucao() void excluirRegistro() void salvaDataDevolucao() String alteraDisponibilidade(String) void limpaTabelaCliente() void pegDataDevolucaoTabela() String jBCadstrarActionPerformed(ActionEvent) void jbSairActionPerformed(ActionEvent) void disponibilidadeLivro() String pegalLivro() String listaContatosLivro() void jbNovoActionPerformed(ActionEvent) void verificaDisponibilidadeLivro() boolean passaValor(String) void limpaTabelaEmprestimo() void verificaDados() boolean listaContatosEmprestimo() void dataEmprestimo() void jBPesquisarActionPerformed(ActionEvent) void jTableEmprestimoMouseClicked(MouseEvent) void jTableLivroMouseClicked(MouseEvent) void formWindowClosed(WindowEvent) void mostraPesquisaCliente(List<Cliente>) void jBEExcluirActionPerformed(ActionEvent) void initComponents() void mostraPesquisaLivro(List<Livro>) void limpaTabelaLivro() void devolveLivro() void </pre>	<pre> jCBDescricao JComboBox jLabel2 JLabel jLabel9 JLabel jTpesquisar JTextField jTableMulta JTable jbPagar JButton jBExcluir JButton jLabel8 JLabel jPanel3 JPanel jNovo JButton jPanel1 JPanel jLabel4 JLabel jT2DataDia JTextField jBRegistrar JButton multas List<Multa> jTpesquisar JButton jT1dCliente JTextField jT3Valor JTextField jLabel7 JLabel jbSair JButton jLabel1 JLabel jbDesricao JButtonGroup jScrollPane1 JScrollPane jScrollPane3 JScrollPane jPanel2 JPanel jLabel10 JLabel tmMulta DefaultTableModel jT4ValorTotal JTextField jLabel3 JLabel jPanel4 JPanel receive(String, String) void limpaTabelaMulta() void desabilitaCampos() void mostraPesquisaMulta(List<Multa>) void pagarMulta() void pegalCliente() String cadastroRegistro() void initComponents() void jBPesquisarActionPerformed(ActionEvent) void formWindowClosed(WindowEvent) void listaContatosCliente() void jTableClientemouseClicked(MouseEvent) void jbPagarMouseClicked(MouseEvent) void jbNovoActionPerformed(ActionEvent) void jbSairActionPerformed(ActionEvent) void limpaTabelaMultaMouseClicked(MouseEvent) void habilitaCampos() void verificaDados() boolean mostraPesquisaCliente(List<Cliente>) void excluirRegistro() void jbRegistrarActionPerformed(ActionEvent) void main(String) void listaContatosMulta() void </pre>	<pre> jBSair JButton jLabel10 JLabel jLabel7 JLabel jTpesquisar JTextField jLabel2 JLabel jLabelPrincipal JFPrincipal jbPagar JButton jBExcluir JButton jLabel8 JLabel jPanel3 JPanel jNovo JButton jPanel1 JPanel jLabel4 JLabel jT2DataDia JTextField jBRegistrar JButton multas List<Multa> jTpesquisar JButton jT1dCliente JTextField jT3Sexo JTextField jT01d JTextField jT6fone JTextField jPanel1 JPanel jBAlterar JButton jLabel5 JLabel jBExcluir JButton jLabel7 JLabel jbExcluir JButton jLabel3 JLabel jLabel2 JLabel clientes List<Cliente> jT4CPF JTextField jTablePesquisa JTextField jLabel8 JLabel jPanel3 JPanel jT1Nome JTextField jPanel1 JPanel jT2DataAno JTextField tmCliente DefaultTableModel jLabel9 JLabel jBpesquisarActionPerformed(ActionEvent) void jbNovoActionPerformed(ActionEvent) void jBpesquisarActionPerformed(ActionEvent) void jTablePesquisaMouseClicked(MouseEvent) void jTableMultiMouseClicked(MouseEvent) void jBalterarActionPerformed(ActionEvent) void jBCadstrarActionPerformed(ActionEvent) void jbNovoActionPerformed(ActionEvent) void main(String) void </pre>	<pre> jT4Ano JTextField jTStatus JTextField jBAlterar JButton jBEExcluir JButton jbSair JButton jTExemplar JTextField jLabel4 JLabel jT2DataDia JTextField jBExcluir JButton jLabel6 JLabel jT01d JTextField jT6fone JTextField jPanel1 JPanel jBAlterar JButton jLabel5 JLabel jBExcluir JButton jLabel7 JLabel jbExcluir JButton jLabel3 JLabel jLabel2 JLabel clients List<Cliente> jT4CPF JTextField jTablePesquisa JTextField jLabel8 JLabel jPanel3 JPanel jT1Nome JTextField jPanel1 JPanel jT2DataAño JTextField tmCliente DefaultTableModel jLabel9 JLabel jBpesquisarActionPerformed(ActionEvent) void jbNovoActionPerformed(ActionEvent) void jBpesquisarActionPerformed(ActionEvent) void jTablePesquisaMouseClicked(MouseEvent) void jTableMultiMouseClicked(MouseEvent) void jBalterarActionPerformed(ActionEvent) void jBCadstrarActionPerformed(ActionEvent) void jbNovoActionPerformed(ActionEvent) void main(String) void </pre>	<pre> fone String dataNasc String sexo String id int endereco String name String setId(int) void jBCadastrar JButton jLabel1 JLabel jLabel5 JLabel jTpesquisar JTextField jTExemplar JTextField jLabel4 JLabel jT2DataDia JTextField jBExcluir JButton jLabel6 JLabel jT01d JTextField jT6fone JTextField jPanel1 JPanel jBAlterar JButton jLabel3 JLabel jLabel4 JLabel jLabel2 JLabel getld() int setSexo(String) void getEndereco() String getSexo() String getDataNasc() String getNome() String getCpf() void getFone() String setEndereco(String) void jbPesquisarActionPerformed(ActionEvent) void desabilitaCampos() void jbSairActionPerformed(ActionEvent) void main(String) void initComponents() void jbNovoActionPerformed(ActionEvent) void formWindowClosed(WindowEvent) void jBExcluirActionPerformed(ActionEvent) void jBalterarActionPerformed(ActionEvent) void jTablePesquisaMouseClicked(MouseEvent) void alteraRegistro() void limpaCampos() void habilitaCampos() void cadastroRegistro() void verificaDados() boolean listaContatos() void limpaCampos() void habilitaCampos() void limparTabela() void excluirRegistro() void jBCadstrarActionPerformed(ActionEvent) void data() String jbNovoActionPerformed(ActionEvent) void main(String) void </pre>	<pre> jMEmprimo JMenuItem jMenu6 JMenu jMenu7 JMenu jMenu3 JMenu jMLulta JMenuItem jMenu2 JMenu jMLivro JMenuItem jMLivro JMenuItem jMCiente JMenuItem jMenuBar1 JMenuBar setAno(short) void setDisponibilidade(String) void setExemplar(String) void setEdicao(byte) void getAno() short getDisponibilidade() String getAutor(String) void setId(int) void getMain(String) void getld() int getAno() int getEdicao() byte getExemplar() String getClienteActionPerformed(ActionEvent) void </pre>	<pre> id int exemplar String disponibilidade String autor String edicao byte ano short setAno(short) void setDisponibilidade(String) void setExemplar(String) void setEdicao(byte) void getAno() short getEdicao() byte getExemplar() String </pre>

Emprestimo	Multa	BdMulta	BdEmprestimo
<p>f <code>id_emprestimo</code> int f <code>id_cliente</code> int f <code>data_devolucao</code> String f <code>data_emprestimo</code> String f <code>id_livro</code> int</p> <p>m <code>setId_cliente(int)</code> void m <code>getData_emprestimo()</code> String m <code>setData_emprestimo(String)</code> void m <code>setData_devolucao(String)</code> void m <code>getId_emprestimo()</code> int m <code>getData_devolucao()</code> String m <code>setId_emprestimo(int)</code> void m <code>getId_cliente()</code> int m <code>setId_livro(int)</code> void m <code>getId_livro()</code> int</p>	<p>f <code>id_cliente</code> int f <code>descricao</code> String f <code>valor</code> float f <code>id_multa</code> int</p> <p>m <code>setValor(float)</code> void m <code>setId_cliente(int)</code> void m <code>getDescricao()</code> String m <code>getId_cliente()</code> int m <code>setDescricao(String)</code> void m <code>getValor()</code> float m <code>setId_multa(int)</code> void m <code>getId_multa()</code> int</p>	<p>f <code>conexao</code> Connection</p> <p>m <code>removeMultas(String)</code> void m <code>adicionaMulta(Multa)</code> void m <code>remove(int)</code> void m <code>totalMultaCliente(String)</code> String m <code>getLista(String)</code> List<Multa> m <code>getListaMultaPorCliente(String)</code> List<Multa></p>	<p>f <code>conexao</code> Connection</p> <p>m <code>getListaPorCliente(String)</code> List<Emprestimo> m <code>verificaMultaCliente(String)</code> boolean m <code>remove(int)</code> void m <code>getLista(String)</code> List<Emprestimo> m <code>adicionaEmprestimo(Emprestimo)</code> void</p>

BdLivro	BdCliente	JMLTestes	CriaConexao	Biblioteca
<p>f <code>conexao</code> Connection</p> <p>m <code>getLista(String)</code> List<Livro> m <code>alteraDisponibilidadeLivro(Livro)</code> void m <code>altera(Livro)</code> void m <code>adicionaLivro(Livro)</code> void m <code>remove(int)</code> void</p>	<p>f <code>conexao</code> Connection</p> <p>m <code>remove(int)</code> void m <code>getLista(String)</code> List<Cliente> m <code>adicionaCliente(Cliente)</code> void m <code>altera(Cliente)</code> void</p>	<p>f <code>i</code> int f <code>j</code> int</p> <p>m <code>add(int)</code> void m <code>main(String[])</code> void</p>	<p>m <code>getConexao()</code> Connection</p>	<p>m <code>main(String[])</code> void</p>

Análise do código fonte

ARQUIVOS ANALISADOS [8 CLASSES]

- Model
 - Cliente.java
 - Multa.java
 - Emprestimo.java
 - Livro.java
- Utils
 - BdLivro.java
 - BdCliente.java
 - BdMulta.java
 - BdEmprestimo.java

ITEM	QUANTIDADE
Número de Classes Total	8
Número de Classes Anotadas	8
Linhas de Código das Classes Anotadas	1051
Quantidade de Métodos Anotados	68
Herança e Interface	0
invariant	2
initially	0
constraint	0
assignable	45
requires	61
ensures	67
normal behaviour e exceptional behaviour	0
\forall	0
\exists	0
Modelos (represents)	0
behaviour subtyping	0

Quantificação do projeto

Análise do código fonte

ANOTAÇÕES ORDENADAS POR QUANTIDADE

ITEM	QUANTIDADE
ensures	67
requires	61
assignable	45
invariant	2
Herança e Interface	0
initially	0
constraint	0
normal behaviour e exceptional behaviour	0
\forall	0
\exists	0
Modelos (represents)	0
behaviour subtyping	0

Análise do código fonte

BDMULTA

- Responsável pelos acessos ao banco relativos à entidade multa.
- Total de anotações
 - requires [1]
 - ensures [2]
- Requires em quase todos os casos garante que os atributos passados para as consultas SQL **não serão vazios** (null ou "").
 - Isso preza pela **otimização** dos acessos ao banco, visto que abrir e fechar conexões consome recursos.
 - O intuito é fazer com que cada acesso seja significativo.
 - No caso de inserções, o requires garante que não há inserção de tuplas com valor null, o que no contexto desta aplicação não é interessante ou necessário.

"The server maintains several buffers for each client connection. One is used as a communications buffer for exchanging information with the client. Other buffers are maintained per client for reading tables and performing join and sort operations." (MySQL 5.0 Certification Study Guide, page 357)

Análise do código fonte

BDMULTA

```
95     // SELECT - Retorna uma lista com as multas de um determinado cliente
96     /*@ requires id_cliente != null;
97     @ requires id_cliente != "";
98     @*/
99     public List<Multta> getListaMulttaPorCliente(String id_cliente) throws SQLException{
100         // Prepara conexão p/ receber o comando SQL
101         String sql = "SELECT multa.id_multa, multa.id_cliente, multa.descricao, multa.valor " +
102             "FROM multa " +
103             "INNER JOIN cliente " +
104             "ON multa.id_cliente = cliente.id_cliente " +
105             "WHERE multa.id_cliente = ?;";
106         PreparedStatement stmt = this.conexao.prepareStatement(sql);
107         stmt.setString(1, id_cliente);
108
109         // Recebe o resultado da consulta SQL
110         ResultSet rs = stmt.executeQuery();
111
112         List<Multta> lista = new ArrayList<>();
113
114         // Enquanto existir registros, pega os valores do ReultSet e vai adicionando na lista
115         while(rs.next()) {
116             // A cada loop, é instanciado um novo objeto, p/ servir de ponte no envio de registros p/ a lista
117             Multta m = new Multta();
```

Em termos de qualidade de software e eficiência de recursos, é necessários pensar na alocação e desalocação dos buffers para consultas.

Trecho do método getListaMulttaPorCliente da classe BdMultta

Análise do código fonte

BDMULTA

- Requires visa **otimização** dos acessos ao banco.
- O ensures neste método garante que a string **retornoada não será nulo ou vazio**.
 - Isso é importante pois o retorno deste método é **diretamente lançado na GUI**, não há um tratamento em um método interface entre as duas camadas da aplicação, i.e., entre as duas classes.

```
137     // SELECT - Retorna o valor total das multas de um cliente
138     /*@ requires id_cliente != null;
139      @ requires id_cliente != "";
140      @ ensures \result != null
141      @ ensures \result != ""
142      */
143     public String totalMultCliente(String id_cliente) throws SQLException {
144         System.out.println("asdj89asduasd8asd8");
145         // Prepara conexão p/ receber o comando SQL
146         String sql = "SELECT SUM(multa.valor) AS totalMult FROM multa "
147             + "INNER JOIN cliente ON multa.id_cliente = cliente.id_cliente "
148             + "WHERE cliente.id_cliente = ? "
149             + "GROUP BY cliente.id_cliente;";
150         PreparedStatement stmt = this.conexao.prepareStatement(sql);
151         stmt.setString(1, id_cliente);
152
153         // Recebe o resultado da consulta SQL
154         ResultSet rs = stmt.executeQuery();
155
156         rs.next();
157         String totalMult = String.valueOf(rs.getFloat("totalMult"));
```

Trecho do método `totalMultCliente` da classe `BdMult`

Análise do código fonte

BDEMPRESTIMO

- Responsável pelos acessos ao banco relativos à entidade empréstimo.
- Total de anotações
 - requires [9]
- Requires visa otimização das conexões com o banco.

```
58     // SELECT - Retorna uma lista com o resultado da consulta
59     /*@ requires id != null;
60     @ requires id != "";
61     @*/
62     public List<Emprestimo> getLista(String id) throws SQLException{
63         // Prepara conexão p/ receber o comando SQL
64         String sql = "SELECT * FROM emprestimo WHERE id_emprestimo like ?";
65         PreparedStatement stmt = this.conexao.prepareStatement(sql);
66         stmt.setString(1, id);
67
68         // Recebe o resultado da consulta SQL
69         ResultSet rs = stmt.executeQuery();
70
71         List<Emprestimo> lista = new ArrayList<>();
72
73         // Enquanto existir registros, pega os valores do ReultSet e vai adicionando na lista
74         while(rs.next()) {
75             // A cada loop, é instanciado um novo objeto, p/ servir de ponte no envio de registros p/ a lista
76             Emprestimo e = new Emprestimo();
77
78             // "c" -> Registro novo - .setNome recebe o campo do banco de String "nome"
79             e.setId_emprestimo(Integer.valueOf(rs.getString("id_emprestimo")));
80             e.setId_cliente(Integer.valueOf(rs.getString("id_cliente")));
81             e.setId_livro(Integer.valueOf(rs.getString("id_livro")));
82             e.setData_emprestimo(rs.getString("data_emprestimo"));
83             e.setData_devolucao(rs.getString("data_devolucao"));
84
85             // Adiciona o registro na lista
86             lista.add(e);
```

Trecho do método `getLista` da classe `BdEmprestimo`

Análise do código fonte

BDLIVRO

- Responsável pelos acessos ao banco relativos à entidade Livro.
- Total de anotações
 - `requires` [4]
 - `assignable` [6]
 - `ensures` [2]
 - `invariant` [1]
- As pré-condições garantem que os dados que serão adicionados ao banco de dados não são nulos.
- As pós-condições aparecem garantindo que o retorno do método de pesquisa e a conexão não sejam nulos
- Os assignables são usados para garantir que não haverá mudança de valor em atributos na maioria dos métodos e no construtor para garantir a instanciação da variável de conexão

```
public class BdLivro {  
  
    /* ---CONEXÃO COM O BD-- */  
    /*@ spec_public @*/ private Connection conexao;  
    //@ public invariant conexao != null;  
  
    // Estabelece uma conexão  
    /* @ assignable this.conexao;  
     * ensures this.conexao != null;  
     */  
    ♦ PauloJP  
    public BdLivro() throws SQLException {  
        this.conexao = CriaConexao.getConexao();  
    }  
    /* <-CONEXÃO COM O BD--- */  
  
    /* ---LIVRO-> */  
  
    // CREATE - Adiciona um registro  
  
    /*@ public normal_behavior  
     * requires l != null;  
     * assignable \nothing;  
     */  
    ♦ PauloJP +1  
    public /*@ pure @*/ void adicionaLivro(Livro l) throws SQLException {  
        // Prepara conexão p/ receber o comando SQL  
        String sql = "INSERT INTO livro(exemplar, autor, edicao, ano, disponibilidade)"  
            + "VALUES(?, ?, ?, ?, ?);";  
        PreparedStatement stmt;  
        // stmt recebe o comando SQL
```

Análise do código fonte

BDCLIENTE

- Responsável pelos acessos ao banco relativos à entidade Cliente.
- Total de anotações
 - requires [2]
 - ensures [3]
 - assignable [5]
 - invariant [1]

```
PUBLIC CLASS BdCliente {  
  
    /* ----CONEXÃO COM O BD-> */  
    /*@ spec_public @*/ private Connection conexao;  
    //@ public invariant conexao != null;  
  
    // Estabelece uma conexão  
    /* @ assignable this.conexao;  
     * ensures this.conexao != null;  
     */  
    • PauloJP  
    public BdCliente() throws SQLException {  
        this.conexao = CriaConexao.getConexao();  
    }  
    /*...*/  
  
    /* ----CLIENTE-> */  
  
    // CREATE - Adiciona um registro  
    /*@     requires c != null;  
     * assignable \nothing;  
     */  
    • PauloJP  
    public void adicionaCliente(Cliente c) throws SQLException {  
        // Prepara conexão p/ receber o comando SQL  
        String sql = "INSERT INTO cliente(nome, data_nasc, sexo, cpf, endereco, fone)"  
                    + "VALUES(?, ?, ?, ?, ?, ?);  
        PreparedStatement stmt;  
        // stmt recebe o comando SQL  
        stmt = this.conexao.prepareStatement(sql);
```

Análise do código fonte

LIVRO

- Responsável pela Instanciação de um objeto do tipo livro, assim como os métodos getters e setters de seus atributos.
- Total de anotações
 - requires (19)
 - ensures (18)
 - assignable (12)

```
public class Livro {  
    private /*@ spec_public */ int id;  
    private /*@ spec_public */ String exemplar;  
    private /*@ spec_public */ String autor;  
    private /*@ spec_public */ byte edicao;  
    private /*@ spec_public */ short ano;  
    private /*@ spec_public */ String disponibilidade;  
  
    /*...*/  
    ♦ PauloJP  
    public Livro(int id, String exemplar, String autor, byte edicao, short ano, String disponibilidade) {  
        this.id = id;  
        this.exemplar = exemplar;  
        this.autor = autor;  
        this.edicao = edicao;  
        this.ano = ano;  
        this.disponibilidade = disponibilidade;  
    }  
  
    ♦ PauloJP  
    public Livro() {  
    }  
  
    /*@ requires 0 <= id;  
     * @ assignable \nothing;  
     * @ ensures \result == id;  
     */  
    ♦ PauloJP +1  
    public /*@ pure */ int getId() { return id; }
```

Análise do código fonte

CLIENTE

- Responsável pela Instanciação de um objeto do tipo cliente, assim como os métodos getters e setters.
- Total de anotações
 - requires [11]
 - ensures [20]

```
//@ requires nome.length() > 0 && nome.length() <= 100;
//@ requires dataNasc.length() == 10;
//@ requires sexo.length() == 1;
//@ requires cpf.length() == 11;
//@ requires endereco.length() > 0 && endereco.length() <= 100;
//@ ensures this.nome == nome;
//@ ensures this.dataNasc == dataNasc;
//@ ensures this.sexo == sexo;
//@ ensures this.cpf == cpf;
//@ ensures this.endereco == endereco;
//@ ensures this.fone == fone;
public Cliente(String nome, String dataNasc, String sexo, String cpf, String endereco, String fone) {
    this.nome = nome;
    this.dataNasc = dataNasc;
    this.sexo = sexo;
    this.cpf = cpf;
    this.endereco = endereco;
    this.fone = fone;
}

public Cliente() {

}

//@ ensures \result == id;
public /*@ pure @*/ int getId() {
    return id;
}

//@ requires id >= 0;
//@ ensures this.id == id;
//@ assignable this.id;
public void setId(int id) {
    this.id = id;
}

//@ ensures \result == nome;
//@ assignable \nothing;
```

Análise do código fonte

EMPRESTIMO

- Responsável pela Instanciação de um objeto do tipo emprestimo, assim como os métodos getters e setters. (data de empréstimo, data de devolução, quem pegou o livro, qual livro...)
- Total de anotações
 - requires (0)
 - ensures (10)

```
public class Emprestimo {  
  
    private /*@ spec_public */ int id_emprestimo;  
    private /*@ spec_public */ int id_cliente;  
    private /*@ spec_public */ int id_livro;  
    private /*@ spec_public */ String data_emprestimo;  
    private /*@ spec_public */ String data_devolucao;  
  
    public Emprestimo() {  
    }  
  
    /**@ ensures this.id_emprestimo == id_emprestimo;  
     * @ ensures this.id_cliente == id_cliente;  
     * @ ensures this.id_livro == id_livro;  
     * @ ensures this.data_emprestimo == data_emprestimo;  
     * @ ensures this.data_devolucao == data_devolucao;  
     * @  
     * @param id_emprestimo  
     * @param id_cliente  
     * @param id_livro  
     * @param data_emprestimo  
     * @param data_devolucao  
     */  
    public Emprestimo(int id_emprestimo, int id_cliente, int id_livro, String data_emprestimo, String data_devolucao) {  
        this.id_emprestimo = id_emprestimo;  
        this.id_cliente = id_cliente;  
        this.id_livro = id_livro;  
        this.data_emprestimo = data_emprestimo;  
        this.data_devolucao = data_devolucao;  
    }  
  
    /**@ ensures \result == id_emprestimo;  
     * @  
     * @return id_emprestimo;  
     */  
    public /*@ pure */ int getId_emprestimo() {  
        return id_emprestimo;  
    }  
  
    /**@ assignable this.id_emprestimo;  
     * @  
     * @param id_emprestimo  
     */  
    public void setId_emprestimo(int id_emprestimo) {  
        this.id_emprestimo = id_emprestimo;  
    }  
  
    /**@ ensures \result == id_cliente;  
     * @  
     * @return id_cliente;  
     */  
    public /*@ pure */ int getId_cliente() {  
        return id_cliente;  
    }
```

Análise do código fonte

MULTA

- Responsável pela Instanciação de um objeto do tipo multa, assim como os métodos getters e setters. (valor da multa, quem está sendo multado, descricao com o tipo da multa...)
- Total de anotações
 - requires (4)
 - ensures (12)

```
public class Multa {  
  
    private /*@ spec_public */ int id_multa;  
    private /*@ spec_public */ int id_cliente;  
    private /*@ spec_public */ String descricao;  
    private /*@ spec_public */ float valor;  
  
    //@ requires valor >= 0;  
    //@ requires descricao.length() <= 60;  
    //@ ensures this.id_multa == id_multa;  
    //@ ensures this.id_cliente == id_cliente;  
    //@ ensures this.descricao == descricao;  
    //@ ensures this.valor == valor;  
    public Multa(int id_multa, int id_emprestimo, String descricao, float valor, String pagamento)  
        this.id_multa = id_multa;  
        this.id_cliente = id_emprestimo;  
        this.descricao = descricao;  
        this.valor = valor;  
    }  
  
    public Multa() {  
    }  
  
    //@ ensures \result == id_multa;  
    //@ assignable \nothing;  
    public int getId_multa() {  
        return id_multa;  
    }  
  
    //@ ensures this.id_multa == id_multa;  
    //@ assignable this.id_multa;  
    public void setId_multa(int id_multa) {  
        this.id_multa = id_multa;  
    }  
  
    //@ ensures \result == id_cliente;  
    //@ assignable \nothing;  
    public int getId_cliente() {  
    }
```

Obrigado!
