

```
1 package DAO;
2
3 /**
4  *
5  * @author Edwar Saliba Júnior
6  */
7 import Conexao.Access;
8 import Conexao.Database;
9 import Dominio.Contato;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.util.ArrayList;
13
14 public class ContatoDAO {
15
16     private Database db;
17
18     public ContatoDAO() throws SQLException, ClassNotFoundException {
19         db = new Database(new Access("MySQL", "localhost",
20             3306, "agenda", "root", "123456"));
21         // MySQL.
22         db.setDbms(2);
23         db.connectionToDB();
24     }
25
26     public void inserir(Contato contato) {
27         try {
28             int cod = db.getIntFieldValue("contato", "MAX(idcontato)", "");
29             cod++;
30             contato.setIdContato(cod);
31
32             String[] campos = {"idcontato", "nome", "telefone"},
33                 valores = {String.valueOf(cod), contato.getNome(),
34                     contato.getTelefone()};
35
36             db.insertValues("contato", campos, valores);
37         } catch (SQLException ex) {
38             System.out.println("Erro de banco: " + ex);
39         }
40     }
41
42     public boolean excluir(Contato contato) {
43         try {
44             db.deleteValues("contato", "codigo = " + contato.getIdContato());
45             return true;
46         } catch (SQLException ex) {
47             System.out.println("Erro de banco: " + ex);
48             return false;
49         }
50     }
51
52     /**
53     * Método responsável por alterar os valores pertencentes a um determinado
54     * contato no banco de dados.
55     *
56     * @param contato tipo Contato
57     * @throws SQLException
58     */
59     public void alterarCliente(Contato contato) throws SQLException {
60         String[] campos = {"idcontato", "nome", "telefone"},
61             valores = {String.valueOf(contato.getIdContato()),
62                 contato.getNome(),
63                 contato.getTelefone()};
64         db.updateValues("contato", campos, valores, "codigo = "
65             + contato.getIdContato());
66     }
67 }
```

```

66     }
67
68     public Contato buscaContato(int idcontato, String nome) {
69         Contato ctt;
70         ResultSet rs;
71         String[] campos = {"idcontato", "nome", "telefone"};
72         try {
73             if ((idcontato > 0) || (nome != null)) {
74                 if (idcontato > 0) {
75                     rs = db.selection("contato", campos, true, "codigo = " +
76                         idcontato, true);
77                     if(rs != null){
78                         ctt = new Contato();
79                         ctt.setIdContato(rs.getInt("codigo"));
80                         ctt.setNome(rs.getString("nome"));
81                         ctt.setTelefone(rs.getString("telefone"));
82
83                         return ctt;
84                     }
85                     else{
86                         System.out.println("Contato não existe.");
87                         return null;
88                     }
89                 } else {
90                     rs = db.selection("SELECT * " +
91                         " FROM contato " +
92                         " WHERE nome = \' " + nome + "\'");
93                     if(rs != null && rs.next()){
94                         ctt = new Contato();
95                         ctt.setIdContato(rs.getInt("codigo"));
96                         ctt.setNome(rs.getString("nome"));
97                         ctt.setTelefone(rs.getString("telefone"));
98
99                         return ctt;
100                    }
101                    else{
102                        System.out.println("Contato não existe.");
103                        return null;
104                    }
105                }
106            } else {
107                System.out.println("Parâmetros inválidos. Contato não encontrado.");
108                return null;
109            }
110        } catch (SQLException ex) {
111            System.out.println("Erro de banco: " + ex);
112            return null;
113        }
114    }
115
116    public ArrayList buscaContatos() throws SQLException{
117        ArrayList<Contato> array = new ArrayList();
118        ResultSet rs;
119        String[] campos = {"idcontato", "nome", "telefone"};
120        rs = db.selection("contato", campos, true, "", false);
121        while(rs.next()){
122            Contato ctt = new Contato();
123            ctt.setIdContato(rs.getInt("codigo"));
124            ctt.setNome(rs.getString("nome"));
125            ctt.setTelefone(rs.getString("telefone"));
126            array.add(ctt);
127        }
128        return array;
129    }
130}

```

