

## JDBC - Continuação

- Implemente a remoção de registros:

```
public void remove(Aluno aluno) {
    try {
        PreparedStatement stmt = connection.prepareStatement("delete
from alunos where id=?");
        stmt.setLong(1, aluno.getId());
        stmt.execute();
        stmt.close();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

- modifique a classe CriaAluno, acrescentando um código para testar o método de remoção (não esqueça de setar o **id** do registro a ser modificado):

```
// removendo registro
AlunoDAO dao = new AlunoDAO();
dao.remove(aluno);
System.out.println("Removido!");
```

- implemente a alteração de registros:

```
public void altera(Aluno aluno) {
    String sql = "update alunos set nome=?, email=?, endereco=? where
id=?";

    try {
        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setString(1, aluno.getNome());
        stmt.setString(2, aluno.getEmail());
        stmt.setString(3, aluno.getEndereco());
        stmt.setLong(4, aluno.getId());
        stmt.execute();
        stmt.close();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

- modifique a classe CriaAluno, acrescentando um código para testar o método de alteração (não esqueça de setar o **id** do registro a ser modificado):

```
// alterando registro
AlunoDAO dao = new AlunoDAO();
dao.altera(aluno);
System.out.println("Alterado!");
```

- implemente a listagem dos registros:

```
public List<Aluno> getLista() { // importar java.util
```

```

List<Aluno> alunos = new ArrayList<Aluno>(); // importar java.util
PreparedStatement stmt;
try {
    stmt = this.connection.prepareStatement("select * from alunos
order by nome");

    ResultSet rs = stmt.executeQuery(); // importar java.sql

    while (rs.next()) {
        Aluno aluno = new Aluno();
        aluno.setId(rs.getLong("id"));
        aluno.setNome(rs.getString("nome"));
        aluno.setEmail(rs.getString("email"));
        aluno.setEndereco(rs.getString("endereco"));

        alunos.add(aluno);
    }
    rs.close();
    stmt.close();
} catch (SQLException e) {
    throw new RuntimeException(e);
}
return alunos;
}

```

- modifique a classe CriaAluno, acrescentando um código para testar o método de listagem:

```

// listando registros
AlunoDAO dao = new AlunoDAO();
List<Aluno> alunos = dao.getList();
for (Aluno aluno1 : alunos){
    System.out.println("Nome: "+aluno1.getNome()+" Email: "+aluno1
.getEmail()+" Endereço: "+ aluno1.getEndereco());
}

```

- Use os métodos criados para inserir, remover, alterar e listar para fazer testes com o banco de dados através da aplicação Java. Usando o comando SQL select, cheque no BD as alterações realizadas.
- **EXERCÍCIO:** Vamos fazer um cadastro de contatos de uma agenda:
  - Crie um banco de dados agendadb com uma tabela contatos contendo os campos id, nome, e-mail, endereço;
  - Crie uma aplicação Java e implemente as operações de inserção, alteração, remoção, busca por nome e listagem de contatos;
  - Anexe o projeto compactado na atividade correspondente no TIDIA.