

FMG - FACULDADE METODISTA GRANBERY

Docente: Marco Antônio Araújo

Discentes:

Ítalo Francis de Oliveira Leite

Josué Pilati de Oliveira

Lucas de Oliveira Soares Machado Linhares

RELATÓRIO DO TRABALHO DE MANUTENÇÃO DE SOFTWARE

<https://github.com/lucas7linhares/SCA>

Juiz de Fora
2018














1. AJUSTE NO DOCUMENTO DE REQUISITO E PROTÓTIPOS DE TELA

- a. O requisito funcional 1 foi complementado com informações do que deve conter na tela.
- b. Nos requisitos funcionais 3, 5, 7, 9 e 11 não era especificado que teria um botão “incluir” e nem qual seria sua função. Portanto, o texto foi complementado com tais informações.
- c. O requisito funcional 15 estava confuso, com informações misturadas e de outros requisitos. O texto foi reorganizado e explicado de forma coerente.
- d. Nos protótipos de tela, a tela referente à “Manter turma” estava incompleta, contendo apenas os campos de texto e combobox onde seriam inseridos as informações, porém não havia nenhum label indicando o que deveria ser preenchido. Esses labels foram inseridos e a tela foi corrigida.
- e. Já a tela “Pesquisar aluno” faltava o campo texto onde seria digitada a pesquisa. O campo foi inserido e a tela também foi corrigida.

2. CORREÇÃO DO SCRIPT DE CRIAÇÃO DO BANCO DE DADOS.

3. CORREÇÃO DE IMPORTAÇÕES DE BIBLIOTECAS E JARS:

- a. Erros:

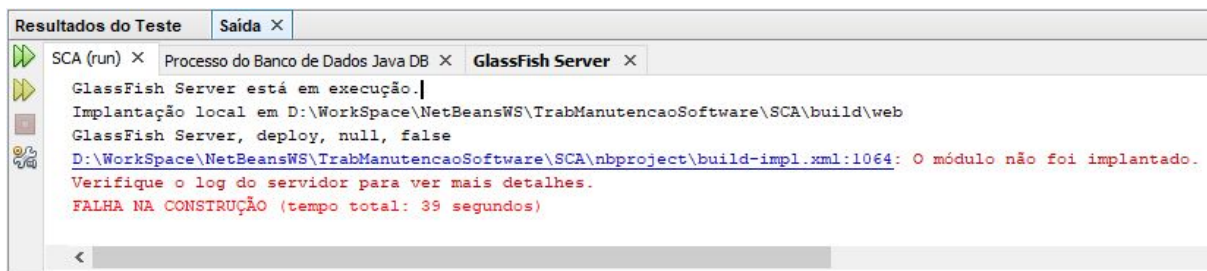
Nome	Pacote
 lib/mysql-connector-java-5.0.4-bin.jar	<input checked="" type="checkbox"/>
 Missing library: hibernate-support	<input checked="" type="checkbox"/>
 Missing library: ejb3-persistence	<input checked="" type="checkbox"/>
 Broken reference: commons-beanutils-1.8.2.jar	<input checked="" type="checkbox"/>
 Broken reference: commons-collections-3.2.1.jar	<input checked="" type="checkbox"/>
 Broken reference: commons-digester-2.1.jar	<input checked="" type="checkbox"/>
 Broken reference: commons-javaflow-20060411.jar	<input checked="" type="checkbox"/>
 Broken reference: commons-logging-1.1.jar	<input checked="" type="checkbox"/>
 Broken reference: iText-2.1.7.jar	<input checked="" type="checkbox"/>
 Broken reference: jasperreports-5.0.4.jar	<input checked="" type="checkbox"/>
 Broken reference: jstl.jar	<input checked="" type="checkbox"/>
 Broken reference: mysql-connector-java-5.1.23-bin.jar	<input checked="" type="checkbox"/>
 Broken reference: standard.jar	<input checked="" type="checkbox"/>

- b. Para correção foram removidos as bibliotecas e jars e adicionados os jars da pasta lib, do projeto, e a biblioteca Hibernate:

Nome	Pacote
lib/mysql-connector-java-5.0.4-bin.jar	<input checked="" type="checkbox"/>
lib/commons-beanutils-1.8.2.jar	<input checked="" type="checkbox"/>
lib/commons-collections-3.2.1.jar	<input checked="" type="checkbox"/>
lib/commons-digester-2.1.jar	<input checked="" type="checkbox"/>
lib/commons-javaflow-20060411.jar	<input checked="" type="checkbox"/>
lib/commons-logging-1.1.jar	<input checked="" type="checkbox"/>
lib/iText-2.1.7.jar	<input checked="" type="checkbox"/>
lib/jasperreports-5.0.4.jar	<input checked="" type="checkbox"/>
Hibernate 4.3.x(JPA2.1)	<input checked="" type="checkbox"/>

4. CORREÇÃO DA CONSTRUÇÃO DO PROGRAMA:

- a. Erro na construção:



- b. Correção do código *web.xml*:

- i. Correção do nome do servlet *ManterCursoController*:

```
<servlet>
  <servlet-name>manterCursoController</servlet-name>
  <servlet-class>controller.ManterCursoController</servlet-class>
</servlet>
```

- ii. Remoção da declaração repetida do servlet *ManterMatriculaController*:

```
<servlet>
  <servlet-name>ManterMatriculaController</servlet-name>
  <servlet-class>controller.ManterMatriculaController</servlet-class>
</servlet>
<servlet>
  <servlet-name>ManterMatriculaController</servlet-name>
  <servlet-class>controller.ManterMatriculaController</servlet-class>
</servlet>
```

- iii. Correção do nome do servlet *ManterCursoController* ao mapear o mesmo:

```
<servlet-mapping>
  <servlet-name>manterCursoController</servlet-name>
  <url-pattern>/manterCursoController</url-pattern>
</servlet-mapping>
```

5. AJUSTES DO BANCO DE DADOS:

- a. Ajuste das configurações do hibernate para acesso ao banco local:

```
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/sca</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">password</property>
    <property name="hibernate.current_session_context_class">thread</property>
    <property name="hibernate.query.factory_class">org.hibernate.hql.internal.classic.ClassicQueryTranslatorFactory</property>
    <property name="hibernate.show_sql">true</property>
    <mapping class="model.Professor"/>
    <mapping class="model.Horario"/>
    <mapping class="model.Aluno"/>
    <mapping class="model.AvaliacaoId"/>
    <mapping class="model.Disciplina"/>
    <mapping class="model.Curso"/>
    <mapping class="model.Turma"/>
    <mapping class="model.Avaliacao"/>
  </session-factory>
</hibernate-configuration>
```

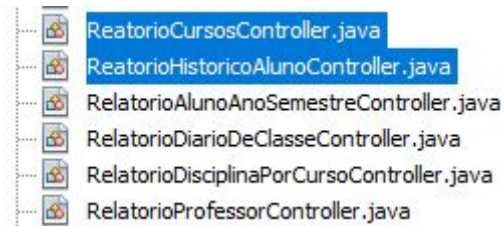
- b. Ajuste na classe BD:

```
public class BD {
  public static Connection getConexao() throws SQLException, ClassNotFoundException{
    Connection conexao=null;
    Class.forName("com.mysql.jdbc.Driver");
    conexao = DriverManager.getConnection("jdbc:mysql://localhost/sca", "root", "password");
    return conexao;
  }
}
```

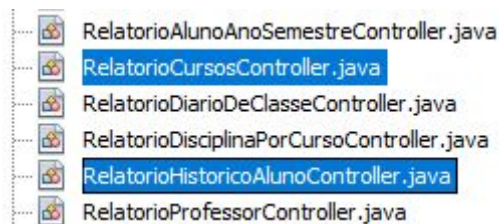
- c. Remoção dos dados que estavam incoerentes e ajuste de palavras com caracteres especiais.

6. CORREÇÃO DE NOME DE CLASSES:

- a. Nomes incorretos das classes *ReatorioCursosController* e *ReatorioHistoricoAlunoController*:



- b. Refatoração dos nomes das classes para *RelatorioCursosController* e *RelatorioHistoricoAlunoController*:

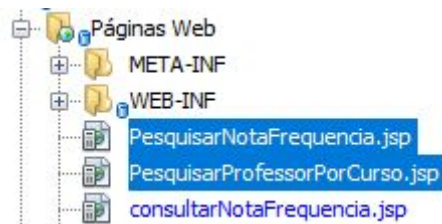


- c. Ajustes na referência, dos nomes dos servlets, no arquivo *web.xml*:

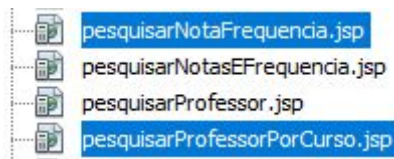
```
<servlet>
  <servlet-name>RelatorioCursosController</servlet-name>
  <servlet-class>controller.RelatorioCursosController</servlet-class>
</servlet>
<servlet>
  <servlet-name>RelatorioHistoricoAlunoController</servlet-name>
  <servlet-class>controller.RelatorioHistoricoAlunoController</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>RelatorioCursosController</servlet-name>
  <url-pattern>/RelatorioCursosController</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>RelatorioHistoricoAlunoController</servlet-name>
  <url-pattern>/RelatorioHistoricoAlunoController</url-pattern>
</servlet-mapping>
```

7. CORREÇÃO NOME DE PÁGINAS JSP:

- a. Nomes fora do padrão das páginas JSP:



- b. Correção dos nomes das páginas:



- c. Ajuste nos controllers para que mantenham a referência correta as páginas JSP:

```
RequestDispatcher janela = request.getRequestDispatcher("/PesquisarNotaFrequencia.jsp");  
RequestDispatcher janela = request.getRequestDispatcher("/PesquisarProfessorPorCurso.jsp");  
  
RequestDispatcher janela = request.getRequestDispatcher("/pesquisarNotaFrequencia.jsp");  
RequestDispatcher janela = request.getRequestDispatcher("/pesquisarProfessorPorCurso.jsp");
```

8. CORREÇÃO DAS CLASSES DAO:

- a. Trocado o método “getCurrentSession()” pelo “openSession()” em todas as classes DAO:

```
Session session = HibernateUtil.getSessionFactory().openSession();
```

- b. Inserido o método “session.close()” antes do retorno de todas as funções em todas as classes DAO:

```
session.close();
```

- c. Inserido as linhas em verde, para tratamento de sessão, e removido as linhas vermelhas:

```
+ public static List<Aluno> obterAlunos() throws ClassNotFoundException, SQLException {  
+     Transaction tx = null;  
+     Session session = HibernateUtil.getSessionFactory().openSession();  
-     session.beginTransaction();  
-     session.clear();  
-     List<Aluno> alunos = session.createCriteria(Aluno.class).list();  
-     session.close();  
-     return alunos;  
+     List<Aluno> alunos = null;  
+     try {  
+         Transaction transaction = session.beginTransaction();  
+         tx = session.getTransaction();  
+         alunos = session.createCriteria(Aluno.class).list();  
+         if (!transaction.wasCommitted()) {  
+             transaction.commit();  
+         }  
+     } catch (Exception e) {  
+         if (tx != null) {  
+             tx.rollback();  
+         }  
+         e.printStackTrace();  
+     } finally {  
+         session.close();  
+         return alunos;  
+     }  
+ }
```

9. AJUSTE DE LAZY INITIALIZE:

- a. Inserido a linha abaixo no arquivo de configuração do Hibernate:

```
<property name="hibernate.enable_lazy_load_no_trans">true</property>
```

10. AJUSTE NOS NOMES DE SERVLETS NAS PÁGINAS JSP:

- a. Corrigido nome do servlet "ManterCursoController" nas páginas "manterCurso.jsp" e "pesquisarCurso.jsp":

```
<form action="manterCursoController?acao=prepararOperacao">  
<form action="ManterCursoController?acao=prepararOperacao">
```

11. AJUSTE NA PÁGINA DE MENUS:

- a. Correção dos links de referência das classes controller de relatório:

```
<a href="ReatorioCursosController?acao=prepararOperacao">Relatório de Cursos</a></br>  
<a href="RelatorioCursosController?acao=prepararOperacao">Relatório de Cursos</a></br>  
<a href="ReatorioHistoricoAlunoController?acao=prepararOperacao">Historico</a></br>  
<a href="RelatorioHistoricoAlunoController?acao=prepararOperacao">Historico</a></br>
```

12. AJUSTE NA PÁGINA pesquisarTurma.jsp:

- a. Adicionado action para o formulário de busca:

```
<form>  
<form action="PesquisarTurmaController">
```

- b. Corrigido título da página:

```
<title>Documento sem título</title>  
<title>Pesquisar turma</title>
```

- c. Ajustado botão de envio do formulário:

```
<td><input type="button" value="Pesquisar"></td>  
<td><input type="submit" name="pesquisar" value="Pesquisar"></td>
```

- d. Ajustado input do campo disciplina:

```
<td>Nome da Disciplina:</td><td><input type="text" nome="txtDisciplina"></td>  
<td>Nome da Disciplina:</td><td><input type="text" name="txtDisciplina"></td>
```

e. Adicionado método de envio do formulário:

```
<form action="PesquisarTurmaController">
<form action="PesquisarTurmaController?filtro=1" method="post">
```

13. AJUSTE NO FILTRO DE TURMAS:

a. Captura dos campos do filtro:

```
try {
    request.setAttribute("turmas", Turma.obterTurmas());
    request.setAttribute("disciplinas", Disciplina.obterDisciplinas());
    String filtro = request.getParameter("filtro");
    if (filtro != null) {
        String codCurso = request.getParameter("optCurso");
        String ano = request.getParameter("optAno");
        String semestre = request.getParameter("optSemestre");
        String disciplina = request.getParameter("txtDisciplina");
        request.setAttribute("turmas", Turma.buscarTurmas(codCurso, ano, semestre, disciplina));
    } else {
        request.setAttribute("turmas", Turma.obterTurmas());
    }
    request.setAttribute("cursos", Curso.obterCursos());
    RequestDispatcher view = request.getRequestDispatcher("/pesquisarTurma.jsp");
    view.forward(request, response);
}
```

b. Criação do método de busca com filtros na classe TurmaDAO:

```
public static List<Turma> buscarTurmas(String codCurso, String ano, String semestre, String disciplina)
{
    Transaction tx = null;
    Session session = HibernateUtil.getSessionFactory().openSession();
    List<Turma> turmas = null;
    try {
        Transaction transaction = session.beginTransaction();
        tx = session.getTransaction();
        String sql = "";
        if (!codCurso.equals("")) {
            sql += " codCurso=" + codCurso + " and";
        }
        if (!ano.equals("")) {
            sql += " ano=" + ano + " and";
        }
        if (!semestre.equals("")) {
            sql += " semestre=" + semestre + " and";
        }
        if (!disciplina.equals("")) {
            sql += " d.nome like '%" + disciplina + "%' and";
        }
        sql = sql.substring(0, (sql.length() - 3));
        turmas = session.createQuery("from Turma where " + sql).list();
        if (!transaction.wasCommitted()) {
            transaction.commit();
        }
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        e.printStackTrace();
    } finally {
        session.close();
        return turmas;
    }
}
```


c. Criação do método de busca com filtros na classe Turma:

```
public static List<Turma> buscarTurmas(String codCurso, String ano, String semestre, String disciplina)
    return TurmaDAO.buscarTurmas(codCurso, ano, semestre, disciplina);
}
```

d. Corrigido busca que trazia dados repetidos:

```
Criteria crit = session.createCriteria(Turma.class);
crit.setResultTransformer(Criteria.DISTINCT_ROOT_ENTITY);
turmas = crit.list();
```

14. CRIAÇÃO DE TELA INICIAL PARA ACESSO AO SISTEMA DE ACORDO COM PERFIL:

a. Criação da nova página index:

```
<div class="block">
  <div class="position-vertical-center">
    <div class="position-center">
      <center>
        <h1>SCA</h1>
        <b>Selecione perfil:</b>
        <form method="post" action="LoginSistema">
          <input type="radio" name="perfil" value="professor"/> Professor
          <input type="radio" name="perfil" value="secretaria"/> Secretária
          <input type="radio" name="perfil" value="aluno"/> Aluno
          <br><br>
          <button type="submit" name="acao" value="entrar">Entrar</button>
        </form>
      </center>
    </div>
  </div>
</div>
```

b. Criação de página de menus de acordo com perfil acessado:

```
<:if test="${perfil == 'secretaria'}">
  <a href="PesquisarCursoController?Manter Cursos/a"> </a> </br>
  <a href="PesquisarDisciplinaController?filtro=0">Manter Disciplinas</a> </br>
  <a href="PesquisarProfessorController">Manter Professores</a> </br>
  <a href="PesquisarAlunoController">Manter Alunos</a> </br>
  <a href="PesquisarTurmaController">Manter Turmas</a></br>
</:if>
<:if test="${perfil == 'professor'}">
  <a href="PesquisarMatrizCurricularController?acao=prepararOperacao">Consultar Matriz Curricular</a> <br />
  <a href="PesquisarProfessorPorCursoController?acao=prepararOperacao"> Consultar Professores Vinculados a cada Curso</a>
  <a href="PesquisarNotaFrequenciaController?acao=prepararOperacao"> Manter Nota e Frequência</a> <br />
</:if>
<:if test="${perfil == 'aluno'}">
  <a href="RelatorioCursosController?acao=prepararOperacao">Relatório de Cursos</a></br>
  <a href="RelatorioProfessorController?acao=prepararOperacao">Relatório de Professores Por Titulação</a></br>
  <a href="RelatorioDisciplinaPorCursoController?acao=prepararOperacao">Relatório de Disciplina Por Cursos</a></br>
  <a href="RelatorioAlunoAnoSemestreController?acao=prepararOperacao">Relatório de Aluno Ano/Semestre</a></br>
  <a href="RelatorioDiarioDeClasseController?acao=prepararOperacao">Diário de Classe</a></br>
  <a href="RelatorioHistoricoAlunoController?acao=prepararOperacao">Historico</a></br>
</:if>
```

c. Criação do servlet LoginSistema:

```
response.setContentType("text/html;charset=UTF-8");
String perfil = request.getParameter("perfil");
request.setAttribute("perfil", perfil);
RequestDispatcher view = request.getRequestDispatcher("/menus.jsp");
view.forward(request, response);
```

15. CRIAÇÃO DE OBTENÇÃO DE SEMESTRE E ANO PARA FILTROS DE BUSCA:

a. Criação dos métodos de busca de semestre e ano na classe Turma:

```
public static List<String> buscarAnos() throws ClassNotFoundException, SQLException {
    return TurmaDAO.buscarAnos();
}

public static List<String> buscarSemestres() throws ClassNotFoundException, SQLException {
    return TurmaDAO.buscarSemestres();
}
```

b. Criação do método de busca de ano na classe TurmaDAO:

```
public static List<String> buscarAnos() throws ClassNotFoundException, SQLException {
    Transaction tx = null;
    Session session = HibernateUtil.getSessionFactory().openSession();
    List<String> anos = null;
    try {
        Transaction transaction = session.beginTransaction();
        tx = session.getTransaction();
        anos = session.createQuery("select t.ano from Turma t group by ano").list();
        if (!transaction.wasCommitted()) {
            transaction.commit();
        }
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        e.printStackTrace();
    } finally {
        session.close();
        return anos;
    }
}
```

c. Criação do método de busca de semestre na classe TurmaDAO:

```
public static List<String> buscarSemestres() throws ClassNotFoundException, SQLException {
    Transaction tx = null;
    Session session = HibernateUtil.getSessionFactory().openSession();
    List<String> semestres = null;
    try {
        Transaction transaction = session.beginTransaction();
        tx = session.getTransaction();
        semestres = session.createQuery("select t.semestre from Turma t group by semestre").list();
        if (!transaction.wasCommitted()) {
            transaction.commit();
        }
    } catch (Exception e) {
        if (tx != null) {
            tx.rollback();
        }
        e.printStackTrace();
    } finally {
        session.close();
        return semestres;
    }
}
```

d. Ajuste nas páginas de filtros:

```
<option value="2013" selected>2013</option>
<option value="2012" selected>2012</option>
<option value="2011" selected>2011</option>
<option value="2010" selected>2010</option>
<c:forEach items="${anos}" var="ano">
    <option value="${ano}">${ano}</option>
</c:forEach>
```

```
<option value="1" selected>1</option>
<option value="2" selected>2</option>

<c:forEach items="${semestres}" var="semestre">
    <option value="${semestre}">${semestre}</option>
</c:forEach>
```

16. ADIÇÃO DA TERCEIRA NOTA PARA ALUNO:

- a. Adicionado o atributo de terceira nota nas páginas manterAlunoNotaFrequencia.jsp e manterNotaFrequencia.jsp.

- b. Adicionado atributo de terceira nota na classe Avaliacao e no servlet ManterAlunoNotaFrequenciaController.

```
@Column(name = "nota3", precision = 12, scale = 0)
public Float getNota3() {
    return this.nota3;
}

public void setNota3(Float nota3) {
    this.nota3 = nota3;
}

int matricula = Integer.parseInt(request.getParameter("txtMatricula"));
float nota1 = 0;
float nota2 = 0;
+ float nota3 = 0;
int numFaltas = 0;
float notaProvaFinal = 0;
if(!request.getParameter("txtNota1").equals("")){
@@ -68,6 +69,10 @@ public void confirmarOperacao(HttpServletRequest request, HttpServletResponse re
    nota2 = Float.parseFloat(request.getParameter("txtNota2"));
}

+ if(!request.getParameter("txtNota3").equals("")){
+     nota3 = Float.parseFloat(request.getParameter("txtNota3"));
+ }
+
if(!request.getParameter("txtNumFaltas").equals("")){
    numFaltas = Integer.parseInt(request.getParameter("txtNumFaltas"));
}

@@ -83,6 +88,7 @@ public void confirmarOperacao(HttpServletRequest request, HttpServletResponse re
    try {
        avaliacao.setNota1(nota1);
        avaliacao.setNota2(nota2);
+       avaliacao.setNota3(nota3);
        avaliacao.setNumFaltas(numFaltas);
        avaliacao.setNotaProvaFinal(notaProvaFinal);
        avaliacao.alterar();
    }
}
```

- c. Feito modificações no script de criação do BD:

```
`codTurma` int(11) NOT NULL,
`nota1` float DEFAULT NULL,
`nota2` float DEFAULT NULL,
+ `nota3` float DEFAULT NULL,
`numFaltas` int(11) DEFAULT NULL,
`notaProvaFinal` float DEFAULT NULL,
PRIMARY KEY (`matriculaAluno`,`codTurma`),
@@ -87,7 +88,7 @@ CREATE TABLE `avaliacao` (

LOCK TABLES `avaliacao` WRITE;
/*!40000 ALTER TABLE `avaliacao` DISABLE KEYS */;
-INSERT INTO `avaliacao` VALUES (6,4,NULL,NULL,NULL,NULL),(6,5,NULL,NULL,NULL,NULL);
+INSERT INTO `avaliacao` VALUES (6,4,NULL,NULL,NULL,NULL,NULL),(6,5,NULL,NULL,NULL,NULL);
```