Universidade Federal de Juiz de Fora
Departamento de Ciências da Computação

Trabalho da Disciplina DCC078 ASPECTOS AVANÇADOS EM ENGENHARIA DE SOFTWARE

Alunos: Cláudio Nazareth Lopes
Ramon Vaz de Mello Larivoir
Professor: Marco Antônio Pereira Araújo

Dezembro
2018

1 Apresentação

Neste trabalho foi realizado a técnica de refatoração, que consiste em uma alteração feita na estrutura do software para tornar mais fácil seu entendimento e reduzir o custo de manutenção sem alterar o comportamento do sistema.

Esta parte do trabalho é a terceira parte de todo o trabalho, e foi utilizado toda a estrutura de código implementada anteriormente na primeira parte. A primeira parte consiste na criação de um projeto com as funcionalidades parecidas com o iFood utilizando padrões de projeto para a implementação.

Com esse projeto implementado, foi realizado um reconhecimento para identificar os "maus cheiros" no código. Os "maus cheiros" são possíveis partes do código que necessitam de refatoração. Código duplicado; Método longo; Classes grandes; Lista de parâmetros longa; Alteração divergente; Cirurgia com rifle; Inveja de dados; Grupos de dados; Obsessão primitiva; Comando switch; Hierarquia paralelas de herança; Classes ociosas; Generalidade especulativa; Campo temporário; Cadeias de menagens; Intermediário; Intimidade inadequada; Classes alternativas com interfaces diferentes; Biblioteca de classes incompleta; Classes de dados; Herança recusada; e Comentários são os tipos de "maus cheiros".

Após a identificação dos "maus cheiros" foi feita uma analise se existia a necessidade de se aplicar as técnicas de refatoração. Onde foi necessário, identificamos qual das técnicas seria necessária ser utilizada para, assim, poder aplicar. O resultado dessa analise está descrita na Seção 2.

2 Aplicação da Refatoração

2.1 Código Duplicado

Código duplicado foi o "mau cheiro" mais presente no projeto. Porém, em nem todos foram aplicadas as técnicas de refatoração.

Da Figura 1 a Figura 6 foram encontrados códigos duplicados nas classes ClienteDAO; EmpresaDAO; FuncionarioDAO; PedidoDAO; ProdutoDAO; e PeordutoEmpresaDAO na função Update, porém, não foi aplicada nenhuma técnica de refatoração.

Da Figura 7 a Figura 16 foram encontrados códigos duplicados nas classes ClienteDAO; EmpresaDAO; FuncionarioDAO; PedidoDAO; ProdutoDAO; e PeordutoEmpresaDAO na função Find, mas também não foi utilizada nenhuma técnica de refatoração.

Da Figura 17 a Figura 20 foram encontrados códigos duplicados nas clas-

ses EmpresaDAO; PedidoDAO; ProdutoDAO; e PeordutoEmpresaDAO na função List, e também não foi aplicada nenhuma técnica de refatoração.

```
public void update(String nome, String telefone, String email, String senha) throws SQLException,
 ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update cliente set telefone = '" + telefone + "' where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 1: Classe ClienteDAO função Update

```
public void update(int id, String nome, String email) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update empresa set email = '" + email + "', nome = '" + nome +
        "' where id_empresa = " + id + "");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 2: Classe EmpresaDAO função Update

Da Figura 21 e 26 foram encontrados códigos duplicados nas classes ClienteDAO; EmpresaDAO; FuncionarioDAO; PedidoDAO; ProdutoDAO; e PeordutoEmpresaDAO na função Delete e, neste caso, foi aplicado a técnica de refatoração **Subir Método na Hieraquia** junto com o padrão de projeto *Template Method*, mostrados da Figura 27 a Figura 33.

2.2 Lista de Parâmetros Longa

Na Figura 34 e Figura 35 foram encontrados o "mau cheiro" lista longa de parâmetros longa, nas classes Cliente e Funcionário no seu respectivo construtor e foi aplicado a técnica de refatoração **Preserva Objeto Inteiro** juntamente com o pedrão de projeto *Composite*, mostrados da Figura 36 a Figura 39.

2.3 Classes de Dados

Na Figura 40 foi encontrado o "mau cheiro" classe de dados na classe Memento, mas nenhuma técnica de refatoração foi realizada.

2.4 Campo Temporário

Da Figura 41 a Figura 92 foi identificado o "mau Cheiro" Campo Temporário, e foram executadas suas refatorações.

```java
public void update (String nome, String telefone) throws SQLException, ClassNotFoundException{
    Connection conn = null;
    Statement st = null;
    try{
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update funcionario set telefone = '" + telefone + "' where nome = '" + nome + "'");
    }catch(SQLException e){
        System.out.println("Erro no SQL");
        throw e;
    }finally{
        closeResources(conn, st);
    }
}
```

Figura 3: Classe FuncionarioDAO função Update

3 Conclusão

Todo código está sujeito a "maus cheiros", e quanto mais cedo eles forem identificados, mais fácil será de refatorá-los. Porém, como foi visto, nem todo "mau cheiro" significa que o código deve ser alterado ou que algo esteja incorreto, algumas vezes é necessário que este tipo de situação ocorra no código para que o entendimento seja mais claro.

Um bom planejamento de projeto permite mapear ou identificar todos os possíveis "maus cheiros" que possam aparecer no código e, assim, as medidas necessárias poderão ser tomadas logo no início da implementação, poupando tempo e esforço de uma refatoração complexa e demorada.

Um código sem "maus cheiros", além de facilitar a manutenção, também facilita o entendimento da finalidade de cada função, fazendo com que o próximo desenvolvedor que olhar o código, saberá interpretar perfeitamente a sua ideia inicial e toda sua lógica para resolver o problema proposto.

```java
public void update(int id, String nome, String email) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update produto set email = '" + email + "', nome = '" + nome + "'
         where id_produto = " + id + "");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 4: Classe PedidoDAO função Update

```java
public void update(int id, String nome, String email) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update produto set email = '" + email + "', nome = '" + nome + "'
         where id_produto = " + id + "");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 5: Classe ProdutoDAO função Update

```java
public void update(int id, String nome, String email) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("update produto set email = '" + email + "', nome = '" + nome + "'
         where id_produto = " + id + "");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 6: Classe ProdutoEmpresaDAO função Update

```java
public Cliente find(int id) throws SQLException, ClassNotFoundException {
    conn = null;
    st = null;
    rs = null;
    Cliente cliente = new Cliente();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from cliente where id_cliente = " + id + "");
        while (rs.next()) {
            cliente
                    .setEmail(rs.getString("email"))
                    .setNome(rs.getString("nome"))
                    .setSenha(rs.getString("senha"))
                    .setTelefone(rs.getString("telefone"))
                    .setId(id);

        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return cliente;
}
```

Figura 7: Classe ClienteDAO função Find

```java
public Empresa find(int id) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Empresa empresa = new Empresa();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from empresa where id_empresa = " + id + "");
        while (rs.next()) {
            empresa
                    .setId(rs.getInt("id_empresa"))
                    .setNome(rs.getString("nome"))
                    .setEmail(rs.getString("email"))
                    .setSenha(rs.getString("senha"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return empresa;
}
```

Figura 8: Classe EmpresaDAO função Find

```java
public String find (int id) throws SQLException, ClassNotFoundException{
    conn = null;
    st = null;
    rs = null;
    Funcionario funcionario = null;
    try{
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select nome, email from funcionario where id = '" + id + "'");
        while(rs.next()){
            funcionario = Factory.createFuncionario(rs.getString("funcao"));
            funcionario.setEmail(rs.getString("emial"));
            funcionario.setNome(rs.getString("nome"));
        }
    }catch(SQLException e){
        System.out.println("Erro no SQL");
        throw e;
    }finally{
        closeResources(conn, st);
    }
    return funcionario.getFuncao();
}
```

Figura 9: Classe FuncionarioDAO função Find

```java
public Pedido find(int id) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Pedido pedido = new Pedido();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from pedido where id_pedido = " + id + "");
        while (rs.next()) {
            pedido
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return pedido;
}
```

Figura 10: Classe PedidoDAO função Find

```java
public Produto find(int id) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Produto produto = new Item();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from produto where id_produto = " + id + "");
        while (rs.next()) {
            produto
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return produto;
}
```

Figura 11: Classe ProdutoDAO função Find

```java
public Produto find(int id) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Produto produto = new Item();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from produto where id_produto = " + id + "");
        while (rs.next()) {
            produto
                    .setId(rs.getInt("id_empresa"))
                    .setNome(rs.getString("nome"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return produto;
}
```

Figura 12: Classe ProdutoEmpresaDAO função Find

```java
public Cliente find(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Cliente cliente = new Cliente();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select nome, email from cliente where nome = '" + nome + "'");
        while (rs.next()) {
            cliente.setEmail(rs.getString("emial"));
            cliente.setNome(rs.getString("nome"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return cliente;
}
```

Figura 13: Classe ClienteDAO função Find

```java
public Funcionario find (String nome) throws SQLException, ClassNotFoundException{
    conn = null;
    st = null;
    rs = null;
    Funcionario funcionario = null;
    try{
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select nome, email from funcionario where nome = '" + nome + "'");
        while(rs.next()){
            funcionario = Factory.createFuncionario(rs.getString("funcao"));
            funcionario.setEmail(rs.getString("emial"));
            funcionario.setNome(rs.getString("nome"));
        }
    }catch(SQLException e){
        System.out.println("Erro no SQL");
        throw e;
    }finally{
        closeResources(conn, st);
    }
    return funcionario;
}
```

Figura 14: Classe FuncionarioDAO função Find

```java
public Pedido find(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Pedido pedido = new Pedido();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from pedido where nome = '" + nome + "'");
        while (rs.next()) {
            pedido
                    .setId(rs.getInt("id_pedido"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return pedido;
}
```

Figura 15: Classe PedidoDAO função Find

```
public Produto find(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    ResultSet rs = null;
    Produto produto = new Item();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select * from produto where nome = '" + nome + "'");
        while (rs.next()) {
            produto
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
        }
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
    return produto;
}
```

Figura 16: Classe ProdutoDAO função Find

```
public List<Produto> listProdutos(int id_empresa) {
    List<Produto> produtos = new ArrayList<Produto>();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select p.* from produto as p join produto_empresa as pe on pe.id_produto
        = p.id_produto where pe.id_empresa = " + id_empresa + "");
        while(rs.next()) {
            Produto produto = new Item();
            produto
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"))
                    .setId(rs.getInt("id_produto"));
            produtos.add(produto);
        }
    } catch (SQLException ex) {
        Logger.getLogger(EmpresaDAO.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(EmpresaDAO.class.getName()).log(Level.SEVERE, null, ex);
    }
    return produtos;
}
```

Figura 17: Classe EmpresaDAO função List

```
public List<Produto> listAll() {
    List<Produto> produtos = new ArrayList<Produto>();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select id_empresa, nome from empresa");
        while (rs.next()) {
            Produto produto = new Item();
            produto
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
            produtos.add(produto);
        }
    } catch (SQLException ex) {
        Logger.getLogger(PedidoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(PedidoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        closeResources(conn, st, rs);
        return produtos;
    }
}
```

Figura 18: Classe PedidoDAO função List

```
public List<Produto> listAll() {
    List<Produto> produtos = new ArrayList<Produto>();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select id_empresa, nome from empresa");
        while (rs.next()) {
            Produto produto = new Item();
            produto
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
            produtos.add(produto);
        }
    } catch (SQLException ex) {
        Logger.getLogger(ProdutoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(ProdutoDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        closeResources(conn, st, rs);
        return produtos;
    }
}
```

Figura 19: Classe ProdutoDAO função List

```java
public List<Produto> listAll() {
    List<Produto> produtos = new ArrayList<Produto>();
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        rs = st.executeQuery("select id_empresa, nome from empresa");
        while (rs.next()) {
            Produto produto = new Item();
            produto
                    .setId(rs.getInt("id_produto"))
                    .setNome(rs.getString("nome"))
                    .setValor(rs.getInt("valor"));
            produtos.add(produto);
        }
    } catch (SQLException ex) {
        Logger.getLogger(ProdutoEmpresaDAO.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(ProdutoEmpresaDAO.class.getName()).log(Level.SEVERE, null, ex);
    } finally {
        closeResources(conn, st, rs);
        return produtos;
    }
}
```

Figura 20: Classe ProdutoEmpresaDAO função List

```java
public void delete(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from cliente where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 21: Classe ClienteDAO função Delete

```java
public void delete(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from empresa where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 22: Classe EmpresaDAO função Delete

```java
public void delete(String nome) throws SQLException, ClassNotFoundException{
    Connection conn = null;
    Statement st = null;
    try{
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from funcionario where nome = '" + nome + "'");
    }catch(SQLException e){
        System.out.println("Erro no SQL");
    }finally{
        closeResources(conn, st);
    }
}
```

Figura 23: Classe FuncionarioDAO função Delete

```java
public void delete(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from empresa where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 24: Classe PedidoDAO função Delete

```java
public void delete(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from empresa where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 25: Classe ProdutoDAO função Delete

```java
public void delete(String nome) throws SQLException, ClassNotFoundException {
    conn = null;
    st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute("delete from empresa where nome = '" + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
        throw e;
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 26: Classe ProdutoEmpresaDAO função Delete

```
public abstract String getSQLDelete();

public void delete(String nome) throws SQLException, ClassNotFoundException {
    Connection conn = null;
    Statement st = null;
    try {
        conn = DatabaseLocator.getInstance().getConnection();
        st = conn.createStatement();
        st.execute(getSQL()  + nome + "'");
    } catch (SQLException e) {
        System.out.println("Erro no SQL");
    } finally {
        closeResources(conn, st);
    }
}
```

Figura 27: Classe DAO função Delete

```
public String getSQLDelete(){
  return ("delete from cliente where nome = '");
}
```

Figura 28: Classe ClienteDAO refatorado

```
public String getSQLDelete(){
  return ("delete from empresa where nome = '");
}
```

Figura 29: Classe EmpresaDAO refatorado

```
public String getSQLDelete(){
  return ("delete from funcionario where nome = '");
}
```

Figura 30: Classe FuncionarioDAO refatorado

```
public String getSQLDelete(){
  return ("delete from empresa where nome = '");
}
```

Figura 31: Classe PedidoDAO refatorado

```
public String getSQLDelete(){
  return ("delete from empresa where nome = '");
}
```

Figura 32: Classe ProdutoDAO refatorado

```
public String getSQLDelete(){
  return ("delete from empresa where nome = '");
}
```

Figura 33: Classe ProdutoEmpresaDAO refatorado

```
public Cliente(String nome, String email, String telefone, String senha) {
    this.nome = nome;
    this.email = email;
    this.telefone = telefone;
    this.senha = senha;
    this.id = ++cont;
}
```

Figura 34: Classe Cliente Construtor

```
public Funcionario(String nome, String email, String funcao, String funcionarioSuperior) {
    this.nome = nome;
    this.email = email;
    this.funcionarioSuperior = Factory.createFuncionario(funcionarioSuperior);
    this.funcao = funcao;
}
```

Figura 35: Classe Funcionário Construtor

```
public Cliente() {
    this.id = ++cont;
}
```

Figura 36: Classe Cliente Refatorado parte 1

```
if(nome.equals("") || email.equals("")) {
   response.sendRedirect("index.jsp");
} else {
    Cliente cliente = new Cliente();
    cliente.setNome(request.getParameter("textNome"))
           .setEmail(request.getParameter("textEmail"))
           .setTelefone(request.getParameter("textTelefone"))
           .setSenha(request.getParameter("textSenha"));
    try{
        ClienteDAO.getInstance().save(cliente);
        response.sendRedirect("CadastrarSucesso.jsp");
    }catch(SQLException ex){
        response.sendRedirect("Erro.jsp");
        ex.printStackTrace();
    }catch(ClassNotFoundException ex){
        ex.printStackTrace();
    }
}
```

Figura 37: Classe Cliente Refatorado parte 2

```
public Funcionario() {
}
```

Figura 38: Classe Cliente Funcionário parte 1

```
if(nome.equals("") || email.equals("") || senha.equals("")) {
   request.getRequestDispatcher("CadastrarFuncionario.jsp").forward(request, response);
} else {
    Funcionario funcionario = Factory.createFuncionario(request.getParameter("funcao"));
    funcionario
            .setEmail(request.getParameter("email"))
            .setNome(request.getParameter("nome"))
            .setSenha(request.getParameter("senha"));
    try{
        FuncionarioDAO.getInstance().save(funcionario);
        request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
    }catch(SQLException ex){
        response.sendRedirect("Erro.jsp");
        ex.printStackTrace();
    }catch(ClassNotFoundException ex){
        ex.printStackTrace();
    }
}
```

Figura 39: Classe Cliente Funcionário parte 2

```java
package controller;

import state.PedidoEstado;

public class Memento {
    private PedidoEstado estado;

    public PedidoEstado getEstadoSalvo(){
        return estado;
    }

    public String toString(){
        return estado.getEstado();
    }

    public Memento(PedidoEstado estado) {
        this.estado = estado;
    }
}
```

Figura 40: Classe Memento

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {
    int id = Integer.parseInt(request.getParameter("id_funcionario"));
    int id_pedido = Integer.parseInt(request.getParameter("id_pedido"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("FuncionarioPedidoStatus.jsp");
    dispatcher.forward(request, response);
    try{
        Funcionario funcionario = Factory.createFuncionario(FuncionarioDAO.getInstance().find(id));
        Pedido pedido = PedidoDAO.getInstance().find(id_pedido);
        funcionario.atualizarPedido(PedidoDAO.getInstance().find(id_pedido));
        response.sendRedirect("contatoSucesso.jsp");
    }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
}
```

Figura 41: Classe AtualizaPedido

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {
    RequestDispatcher dispatcher = request.getRequestDispatcher("FuncionarioPedidoStatus.jsp");
    dispatcher.forward(request, response);
    try{
        Funcionario funcionario = Factory.createFuncionario(FuncionarioDAO.getInstance().find(Integer.parseInt
        (request.getParameter("id_funcionario"))));
        Pedido pedido = PedidoDAO.getInstance().find(Integer.parseInt(request.getParameter("id_pedido")));
        funcionario.atualizarPedido(PedidoDAO.getInstance().find(Integer.parseInt(request.getParameter
        ("id_pedido"))));
        response.sendRedirect("contatoSucesso.jsp");
    }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
}
```

Figura 42: Classe AtualizaPedido refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");
    String email = request.getParameter("textEmail");
    String telefone = request.getParameter("textTelefone");
    String senha = request.getParameter("textSenha");

    if(nome.equals("") || email.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        Cliente cliente = new Cliente(nome, email, telefone, senha);
        try{
            ClienteDAO.getInstance().save(cliente);
            response.sendRedirect("CadastrarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 43: Classe CadastrarCliente

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("") || email.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        Cliente cliente = new Cliente(request.getParameter("textNome"), request.getParameter("textEmail"),
         request.getParameter("textTelefone"), request.getParameter("textSenha"));
        try{
            ClienteDAO.getInstance().save(cliente);
            response.sendRedirect("CadastrarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 44: Classe CadastrarCliente refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");
    String senha = request.getParameter("textSenha");
    String email = request.getParameter("textEmail");

    if(nome.equals("") || senha.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        Empresa empresa = new Empresa();
        empresa
                .setNome(nome)
                .setSenha(senha)
                .setEmail(email);
        try{
            EmpresaDAO.getInstance().save(empresa);
            response.sendRedirect("CadastrarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 45: Classe CadastrarEmpresa

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("") || senha.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        Empresa empresa = new Empresa();
        empresa
                .setNome(request.getParameter("textNome"))
                .setSenha(request.getParameter("textSenha"))
                .setEmail(request.getParameter("textEmail"));
        try{
            EmpresaDAO.getInstance().save(empresa);
            response.sendRedirect("CadastrarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 46: Classe CadastrarEmpresa refatorado

```java
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    String nome = request.getParameter("nome");
    String email = request.getParameter("email");
    String funcao = request.getParameter("funcao");
    String senha = request.getParameter("senha");
    int id_empresa = Integer.parseInt(request.getParameter("id_empresa"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("EmpresaIndex.jsp");
    request.setAttribute("id_empresa", id_empresa);
    if(nome.equals("") || email.equals("") || senha.equals("")) {
       dispatcher = request.getRequestDispatcher("CadastrarFuncionario.jsp");
       dispatcher.forward(request, response);
    } else {
        Funcionario funcionario = Factory.createFuncionario(funcao);
        funcionario
                .setEmail(email)
                .setNome(nome)
                .setSenha(senha);
        try{
            FuncionarioDAO.getInstance().save(funcionario);
            dispatcher.forward(request, response);
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 47: Classe CadastrarFuncionario

```java
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    if(nome.equals("") || email.equals("") || senha.equals("")) {
       request.getRequestDispatcher("CadastrarFuncionario.jsp").forward(request, response);
    } else {
        Funcionario funcionario = Factory.createFuncionario(request.getParameter("funcao"));
        funcionario
                .setEmail(request.getParameter("email"))
                .setNome(request.getParameter("nome"))
                .setSenha(request.getParameter("senha"));
        try{
            FuncionarioDAO.getInstance().save(funcionario);
            request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 48: Classe CadastrarFuncionario refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException, ClassNotFoundException {
    int id_cliente = Integer.parseInt(request.getParameter("id_cliente"));
    int id_empresa = Integer.parseInt(request.getParameter("id_empresa"));
    String[] items = request.getParameterValues("item");
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteConfirmarPedido.jsp");
    request.setAttribute("id_cliente", id_cliente);
    request.setAttribute("id_empresa", id_empresa);
    try {
        Class classe = Class.forName("pagamento." + request.getParameter("pagamento"));
        Object objeto = classe.newInstance();
        FormaPagamento fp = (FormaPagamento) objeto;
        List<Produto> itens = new ArrayList<Produto>();
        float total = 0;
        if (items != null) {
            for (String item : items) {
                int id_produto = Integer.parseInt(item);
                Produto p = ProdutoDAO.getInstance().find(id_produto);
                total = total + p.getValor();
                itens.add(p);
            }
            total = total - (total * fp.getDesconto() / 100);
            request.setAttribute("pagamento", fp);
            request.setAttribute("total", total);
            request.setAttribute("itens", itens);
            dispatcher.forward(request, response);
        } else {
            List<Produto> produtos = EmpresaDAO.getInstance().listProdutos(id_empresa);
            dispatcher = request.getRequestDispatcher("ClienteProdutosEmpresa.jsp");
            FormaPagamento dinheiro = new Dinheiro();
            FormaPagamento cartao = new Cartao();
            request.setAttribute("dinheiro", dinheiro);
            request.setAttribute("cartao", cartao);
            request.setAttribute("produtos", produtos);
            dispatcher.forward(request, response);
        }
    } catch (SQLException ex) {
        Logger.getLogger(ClienteConfirmarPedido.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        Logger.getLogger(ClienteConfirmarPedido.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        Logger.getLogger(ClienteConfirmarPedido.class.getName()).log(Level.SEVERE, null, ex);
```

Figura 49: Classe ClienteConfirmaPadido

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException,
 ClassNotFoundException {
    String[] items = request.getParameterValues("item");
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    try {
        Class classe = Class.forName("pagamento." + request.getParameter("pagamento"));
        Object objeto = classe.newInstance();
        List<Produto> itens = new ArrayList<Produto>();
        float total = 0;
        if (items != null) {
            for (String item : items) {
                Produto p = ProdutoDAO.getInstance().find(Integer.parseInt(item));
                total = total + p.getValor();
                itens.add(p);
            }
            total = total - (total * fp.getDesconto() / 100);
            request.setAttribute("pagamento", (FormaPagamento) objeto);
            request.setAttribute("total", total);
            request.setAttribute("itens", itens);
            request.getRequestDispatcher("ClienteConfirmarPedido.jsp").forward(request, response);
        } else {
            List<Produto> produtos = EmpresaDAO.getInstance().listProdutos(Integer.parseInt(request.getParameter("id_empresa")));
            FormaPagamento dinheiro = new Dinheiro();
            FormaPagamento cartao = new Cartao();
            request.setAttribute("dinheiro", dinheiro);
            request.setAttribute("cartao", cartao);
            request.setAttribute("produtos", produtos);
            request.getRequestDispatcher("ClienteProdutosEmpresa.jsp").forward(request, response);
        }
    } catch (SQLException ex) {
        Logger.getLogger(ClienteConfirmarPedido.class.getName()).log(Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        Logger.getLogger(ClienteConfirmarPedido.class.getName()).log(Level.SEVERE, null, ex);
```

Figura 50: Classe ClienteConfirmaPadido refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    int id = Integer.parseInt(request.getParameter("id_cliente"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteIndex.jsp");
    request.setAttribute("id_cliente", id);
    dispatcher.forward(request, response);

}
```

Figura 51: Classe ClienteIndex

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.getRequestDispatcher("ClienteIndex.jsp").forward(request, response);

}
```

Figura 52: Classe ClienteIndex refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_cliente"));
    List<Empresa> empresas = EmpresaDAO.getInstance().listAll();
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteNovoPedido.jsp");
    request.setAttribute("empresas", empresas);
    request.setAttribute("id_cliente", id);
    dispatcher.forward(request, response);

}
```

Figura 53: Classe ClienteNovoPedido

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("empresas", EmpresaDAO.getInstance().listAll());
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.getRequestDispatcher("ClienteNovoPedido.jsp").forward(request, response);

}
```

Figura 54: Classe ClienteNovoPedido refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException, ClassNotFoundException {
    String[] items = request.getParameterValues("item");
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    try {
        List<Produto> itens = new ArrayList<Produto>();
        for (String item : items) {
            Produto p = ProdutoDAO.getInstance().find(Integer.parseInt(item));
            itens.add(p);
        }
        Pedido pedido = new Pedido();
        pedido.setProduto(itens);
        pedido.setPedidoEstado(new PedidoEstadoEmProdução());
        request.setAttribute("itens", itens);
        request.getRequestDispatcher("ClientePedidoStatus.jsp").forward(request, response);
        List<Produto> produtos = EmpresaDAO.getInstance().listProdutos
        (Integer.parseInt(request.getParameter("id_empresa")));
        FormaPagamento dinheiro = new Dinheiro();
        FormaPagamento cartao = new Cartao();
        request.setAttribute("dinheiro", dinheiro);
        request.setAttribute("cartao", cartao);
        request.setAttribute("produtos", produtos);
        request.getRequestDispatcher("ClienteProdutosEmpresa.jsp").forward(request, response);
    } catch (SQLException ex) {
        Logger.getLogger(ClientePedidoConcluido.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Figura 55: Classe ClientePedidoConcluido

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException, ClassNotFoundException {
    int id_cliente = Integer.parseInt(request.getParameter("id_cliente"));
    int id_empresa = Integer.parseInt(request.getParameter("id_empresa"));
    String[] items = request.getParameterValues("item");
    float total = Float.parseFloat(request.getParameter("total"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClientePedidoStatus.jsp");
    request.setAttribute("id_cliente", id_cliente);
    request.setAttribute("id_empresa", id_empresa);
    try {
        List<Produto> itens = new ArrayList<Produto>();
        for (String item : items) {
            int id_produto = Integer.parseInt(item);
            Produto p = ProdutoDAO.getInstance().find(id_produto);
            itens.add(p);
        }
        Pedido pedido = new Pedido();
        pedido.setProduto(itens);
        pedido.setPedidoEstado(new PedidoEstadoEmProdução());
        request.setAttribute("itens", itens);
        dispatcher.forward(request, response);
        List<Produto> produtos = EmpresaDAO.getInstance().listProdutos(id_empresa);
        dispatcher = request.getRequestDispatcher("ClienteProdutosEmpresa.jsp");
        FormaPagamento dinheiro = new Dinheiro();
        FormaPagamento cartao = new Cartao();
        request.setAttribute("dinheiro", dinheiro);
        request.setAttribute("cartao", cartao);
        request.setAttribute("produtos", produtos);
        dispatcher.forward(request, response);
```

Figura 56: Classe ClientePedidoConcluido refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException, ClassNotFoundException {
    int id_cliente = Integer.parseInt(request.getParameter("id_cliente"));
    int id_empresa = Integer.parseInt(request.getParameter("id_empresa"));
    String[] produtos = request.getParameterValues("item");
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteFormaPagamento.jsp");
    FormaPagamento dinheiro = new Dinheiro();
    FormaPagamento cartao = new Cartao();
    request.setAttribute("dinheiro", dinheiro);
    request.setAttribute("cartao", cartao);
    request.setAttribute("produtos", produtos);
    request.setAttribute("id_cliente", id_cliente);
    request.setAttribute("id_empresa", id_empresa);
    dispatcher.forward(request, response);

}
```

Figura 57: Classe ClientePedidoStatus

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException, ClassNotFoundException {
    FormaPagamento dinheiro = new Dinheiro();
    FormaPagamento cartao = new Cartao();
    request.setAttribute("dinheiro", dinheiro);
    request.setAttribute("cartao", cartao);
    request.setAttribute("produtos", request.getParameterValues("item"));
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    request.getRequestDispatcher("ClienteFormaPagamento.jsp").forward(request, response);

}
```

Figura 58: Classe ClientePedidoStatus refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id_cliente = Integer.parseInt(request.getParameter("id_cliente"));
    int id_empresa = Integer.parseInt(request.getParameter("id_empresa"));
    List<Produto> produtos = EmpresaDAO.getInstance().listProdutos(id_empresa);
    RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteProdutosEmpresa.jsp");
    FormaPagamento dinheiro = new Dinheiro();
    FormaPagamento cartao = new Cartao();
    request.setAttribute("dinheiro", dinheiro);
    request.setAttribute("cartao", cartao);
    request.setAttribute("produtos", produtos);
    request.setAttribute("id_cliente", id_cliente);
    request.setAttribute("id_empresa", id_empresa);
    dispatcher.forward(request, response);
    Pedido pedido = new Pedido();
    try{
        pedido.setProduto(produtos);
        PedidoDAO.getInstance().save(id_empresa, id_cliente,produtos);
        response.sendRedirect("CadastrarSucesso.jsp");
    }catch(ClassNotFoundException ex){
        ex.printStackTrace();
    } catch (SQLException ex) {
        Logger.getLogger(ClienteProdutosEmpresa.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Figura 59: Classe ClientesProdutosEmpresa

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    FormaPagamento dinheiro = new Dinheiro();
    FormaPagamento cartao = new Cartao();
    request.setAttribute("dinheiro", dinheiro);
    request.setAttribute("cartao", cartao);
    request.setAttribute("produtos", EmpresaDAO.getInstance().listProdutos(id_empresa));
    request.setAttribute("id_cliente", Integer.parseInt(request.getParameter("id_cliente")));
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    request.getRequestDispatcher("ClienteProdutosEmpresa.jsp").forward(request, response);
    Pedido pedido = new Pedido();
    try{
        pedido.setProduto(produtos);
        PedidoDAO.getInstance().save(id_empresa, id_cliente,produtos);
        response.sendRedirect("CadastrarSucesso.jsp");
    }catch(ClassNotFoundException ex){
        ex.printStackTrace();
    } catch (SQLException ex) {
        Logger.getLogger(ClienteProdutosEmpresa.class.getName()).log(Level.SEVERE, null, ex);
    }

}
```

Figura 60: Classe ClientesProdutosEmpresa refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            ClienteDAO.getInstance().delete(nome);
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 61: Classe DeletarCliente

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            ClienteDAO.getInstance().delete(request.getParameter("textNome"));
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 62: Classe DeletarCliente refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            EmpresaDAO.getInstance().delete(nome);
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 63: Classe DeletarEmpresa

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            EmpresaDAO.getInstance().delete(request.getParameter("textNome"));
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 64: Classe DeletarEmpresa refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        //Contato contato = new Contato(nome, null);
        try{
            FuncionarioDAO.getInstance().delete(nome);
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 65: Classe DeletarFuncionario

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            FuncionarioDAO.getInstance().delete(request.getParameter("textNome"));
            response.sendRedirect("apagarSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 66: Classe DeletarFuncionario refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("EmpresaIndex.jsp");
    request.setAttribute("id_empresa", id);
    dispatcher.forward(request, response);

}
```

Figura 67: Classe EmpresaIndex

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
}
```

Figura 68: Classe EmpresaIndex refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    List<Produto> produtos = EmpresaDAO.getInstance().listProdutos(id);
    RequestDispatcher dispatcher = request.getRequestDispatcher("EmpresaProdutos.jsp");
    request.setAttribute("produtos", produtos);
    request.setAttribute("id_empresa", id);
    dispatcher.forward(request, response);

}
```

Figura 69: Classe EmpresasProdutos

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("produtos", EmpresaDAO.getInstance().listProdutos(Integer.parseInt
    (request.getParameter("id_empresa"))));
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    request.getRequestDispatcher("EmpresaProdutos.jsp").forward(request, response);

}
```

Figura 70: Classe EmpresasProdutos refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IO
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("CadastrarFuncionario.j
    request.setAttribute("id_empresa", id);
    dispatcher.forward(request, response);
}
```

CampoTemporario.png

Figura 71: Classe FuncionarioCadastrarForm

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
    request.getRequestDispatcher("CadastrarFuncionario.jsp").forward(request, response);
}
```

Figura 72: Classe FuncionarioCadastrarForm refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    String email = request.getParameter("email");
    String senha = request.getParameter("senha");

    if(email.equals("") || senha.equals("")) {
       response.sendRedirect("LoginCliente.jsp");
    } else {
        try{
            Cliente cliente = ClienteDAO.getInstance().login(email, senha);
            if(cliente.getNome() != null) {
                RequestDispatcher dispatcher = request.getRequestDispatcher("ClienteIndex.jsp");
                request.setAttribute("id_cliente", cliente.getId());
                dispatcher.forward(request, response);
            } else {
                response.sendRedirect("LoginCliente.jsp");
            }
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 73: Classe LoginCliente

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {

    if(email.equals("") || senha.equals("")) {
       response.sendRedirect("LoginCliente.jsp");
    } else {
        try{
            Cliente cliente = ClienteDAO.getInstance().login(request.getParameter("email"),
             request.getParameter("senha"));
            if(cliente.getNome() != null) {
                request.setAttribute("id_cliente", cliente.getId());
                request.getRequestDispatcher("ClienteIndex.jsp").forward(request, response);
            } else {
                response.sendRedirect("LoginCliente.jsp");
            }
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 74: Classe LoginCliente refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    String email = request.getParameter("email");
    String senha = request.getParameter("senha");

    if(email.equals("") || senha.equals("")) {
       response.sendRedirect("LoginEmpresa.jsp");
    } else {
        try{
            Empresa empresa = EmpresaDAO.getInstance().login(email, senha);
            if(empresa.getNome() != null) {
                RequestDispatcher dispatcher = request.getRequestDispatcher("EmpresaIndex.jsp");
                request.setAttribute("id_empresa", empresa.getId());
                dispatcher.forward(request, response);
            } else {
                response.sendRedirect("LoginEmpresa.jsp");
            }
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 75: Classe LoginEmpresa

```
public class LoginEmpresa implements Action{
    public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
     ServletException {

        if(email.equals("") || senha.equals("")) {
           response.sendRedirect("LoginEmpresa.jsp");
        } else {
            try{
                Empresa empresa = EmpresaDAO.getInstance().login(request.getParameter("email"),
                 request.getParameter("senha"));
                if(empresa.getNome() != null) {
                    request.setAttribute("id_empresa", empresa.getId());
                    request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
                } else {
                    response.sendRedirect("LoginEmpresa.jsp");
                }
            }catch(SQLException ex){
                response.sendRedirect("Erro.jsp");
                ex.printStackTrace();
            }catch(ClassNotFoundException ex){
                ex.printStackTrace();
            }
        }
    }
}
```

Figura 76: Classe LoginEmpresa refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");
    String senha = request.getParameter("textSenha");

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            String fundionario = null;
            request.setAttribute(fundionario, FuncionarioDAO.getInstance().find(nome));
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 77: Classe LoginFuncionario

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            String fundionario = null;
            request.setAttribute(fundionario, FuncionarioDAO.getInstance().find(request.getParameter("textNome")))
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 78: Classe LoginFuncionario refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    String nome = request.getParameter("nome");
    String valor = request.getParameter("valor");
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("ProdutoCadastrar.jsp");
    request.setAttribute("id_empresa", id);

    if(nome.equals("") || valor.equals("")) {
       dispatcher = request.getRequestDispatcher("ProdutoCadastrar.jsp");
       dispatcher.forward(request, response);
    } else {
        int val = Integer.parseInt(valor);
        Produto produto = new Item();
        produto
                .setNome(nome)
                .setValor(val);
        try{
            ProdutoDAO.getInstance().save(id, produto);
            dispatcher.forward(request, response);
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 79: Classe ProdutoCadastrar

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));

    if(request.getParameter("nome").equals("") || request.getParameter("valor").equals("")) {
       request.getRequestDispatcher("ProdutoCadastrar.jsp").forward(request, response);
    } else {
        Produto produto = new Item();
        produto
                .setNome(request.getParameter("nome"))
                .setValor(Integer.parseInt(request.getParameter("valor")));
        try{
            ProdutoDAO.getInstance().save(Integer.parseInt(request.getParameter("id_empresa")), produto);
            request.getRequestDispatcher("ProdutoCadastrar.jsp").forward(request, response);
        }catch(SQLException ex){
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 80: Classe ProdutoCadastrar refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    RequestDispatcher dispatcher = request.getRequestDispatcher("ProdutoCadastrar.jsp");
    request.setAttribute("id_empresa", id);
    dispatcher.forward(request, response);
}
```

Figura 81: Classe ProdutoCastrarForm

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa"));
    request.getRequestDispatcher("ProdutoCadastrar.jsp").forward(request, response);
}
```

Figura 82: Classe ProdutoCastrarForm refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");
    String telefone = request.getParameter("textTelefone");
    String email = request.getParameter("textEmail");
    String senha = request.getParameter("textSenha");

    if(nome.equals("") || telefone.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            ClienteDAO.getInstance().update(nome, telefone, email, senha);
            response.sendRedirect("contatoSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 83: Classe UpdateCliente

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(nome.equals("") || telefone.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            ClienteDAO.getInstance().update(request.getParameter("textNome"),
             request.getParameter("textTelefone"), request.getParameter("textEmail"),
            request.getParameter("textSenha"));
            response.sendRedirect("contatoSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 84: Classe UpdateCliente refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_cliente"));
    try {
        Cliente cliente = ClienteDAO.getInstance().find(id);
        RequestDispatcher dispatcher = request.getRequestDispatcher("UpdateCliente.jsp");
        request.setAttribute("cliente", cliente);
        dispatcher.forward(request, response);
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }

}
```

Figura 85: Classe UpdateClienteForm

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    try {
        request.setAttribute("cliente", ClienteDAO.getInstance().find(Integer.parseInt
        (request.getParameter("id_cliente"))));
        request.getRequestDispatcher("UpdateCliente.jsp").forward(request, response);
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }

}
```

Figura 86: Classe UpdateClienteForm refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    String nome = request.getParameter("textNome");
    String email = request.getParameter("textEmail");
    String senha = request.getParameter("textSenha");
    int id = Integer.parseInt(request.getParameter("id_empresa"));

    RequestDispatcher dispatcher = request.getRequestDispatcher("EmpresaIndex.jsp");
    request.setAttribute("id_empresa", id);

    if (nome.equals("") || email.equals("")) {
        dispatcher.forward(request, response);
    } else {
        try {
            if (senha.equals("")) {
                EmpresaDAO.getInstance().update(id, nome, email);
            } else {
                EmpresaDAO.getInstance().update(id, nome, email, senha);
            }
            dispatcher.forward(request, response);
        } catch (SQLException ex) {
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        }
    }
}
```

Figura 87: Classe UpdateEmpresa

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));

    if (request.getParameter("textNome").equals("") || request.getParameter("textEmail").equals("")) {
        request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
    } else {
        try {
            if (request.getParameter("textSenha").equals("")) {
                EmpresaDAO.getInstance().update(Integer.parseInt(request.getParameter("id_empresa")),
                 request.getParameter("textNome"), request.getParameter("textEmail"));
            } else {
                EmpresaDAO.getInstance().update(Integer.parseInt(request.getParameter("id_empresa")),
                 request.getParameter("textNome"), request.getParameter("textEmail"),
                request.getParameter("textSenha"));
            }
            request.getRequestDispatcher("EmpresaIndex.jsp").forward(request, response);
        } catch (SQLException ex) {
            response.sendRedirect("Erro.jsp");
            ex.printStackTrace();
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        }
    }
}
```

Figura 88: Classe UpdateEmpresa refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    int id = Integer.parseInt(request.getParameter("id_empresa"));
    try {
        Empresa empresa = EmpresaDAO.getInstance().find(id);
        RequestDispatcher dispatcher = request.getRequestDispatcher("UpdateEmpresa.jsp");
        request.setAttribute("id_empresa", id);
        request.setAttribute("empresa", empresa);
        dispatcher.forward(request, response);
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }

}
```

Figura 89: Classe UpdateEmpresaForm

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException,
 ServletException {
    try {
        request.setAttribute("id_empresa", Integer.parseInt(request.getParameter("id_empresa")));
        request.setAttribute("empresa", EmpresaDAO.getInstance().find(Integer.parseInt
        (request.getParameter("id_empresa")));
        request.getRequestDispatcher("UpdateEmpresa.jsp").forward(request, response);
    } catch (SQLException ex) {
        ex.printStackTrace();
    } catch (ClassNotFoundException ex) {
        ex.printStackTrace();
    }

}
```

Figura 90: Classe UpdateEmpresaForm refatorado

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String nome = request.getParameter("textNome");
    String telefone = request.getParameter("textTelefone");

    if(nome.equals("") || telefone.equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            FuncionarioDAO.getInstance().update(nome, telefone);
            response.sendRedirect("contatoSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 91: Classe UpdateFuncionario

```
public void execute(HttpServletRequest request, HttpServletResponse response) throws IOException {

    if(request.getParameter("textNome").equals("") || request.getParameter("textTelefone").equals("")) {
       response.sendRedirect("index.jsp");
    } else {
        try{
            FuncionarioDAO.getInstance().update(request.getParameter("textNome"),
             request.getParameter("textTelefone"));
            response.sendRedirect("contatoSucesso.jsp");
        }catch(SQLException ex){
            response.sendRedirect("contatoErro.jsp");
            ex.printStackTrace();
        }catch(ClassNotFoundException ex){
            ex.printStackTrace();
        }
    }
}
```

Figura 92: Classe UpdateFuncionario refatorado