



Universidade Federal Fluminense

Trabalho de Engenharia de Software I

Sistema de revistas de uma livraria

Alunos:

Paulo Henrique T. Borges
Claudio dos Santos
Leonardo Ramos
João Lucas Rulff

Professora: Teresa Cristina de Aguiar

Índice

Padrões de projeto e ferramentas.....	3
Imagens das telas de projeto.....	4
Classes de Modelo.....	5
Classes DAO.....	12
Classes de Serviço.....	25
Classes de Controle.....	27
Classes de utilitários.....	38
JSPs do projeto.....	41
Scripts e plugins utilizados no front-end.....	51
Javascripts escritos por nós usando jQuery.....	54
Arquivo XML com as declarações dos Servlets e Filter...	58
Arquivo de conexão com o Banco de Dados.....	60
Interface Homem-Máquina – Telas.....	61
Projeto de Banco de Dados.....	67
Diagrama ER.....	69
Scripts SQL do banco.....	70
Diagrama de Classes.....	73
Diagrama de Sequência.....	75
Mais informações sobre o código.....	76
Descrição do caso de uso real.....	77

Código fonte da implementação:

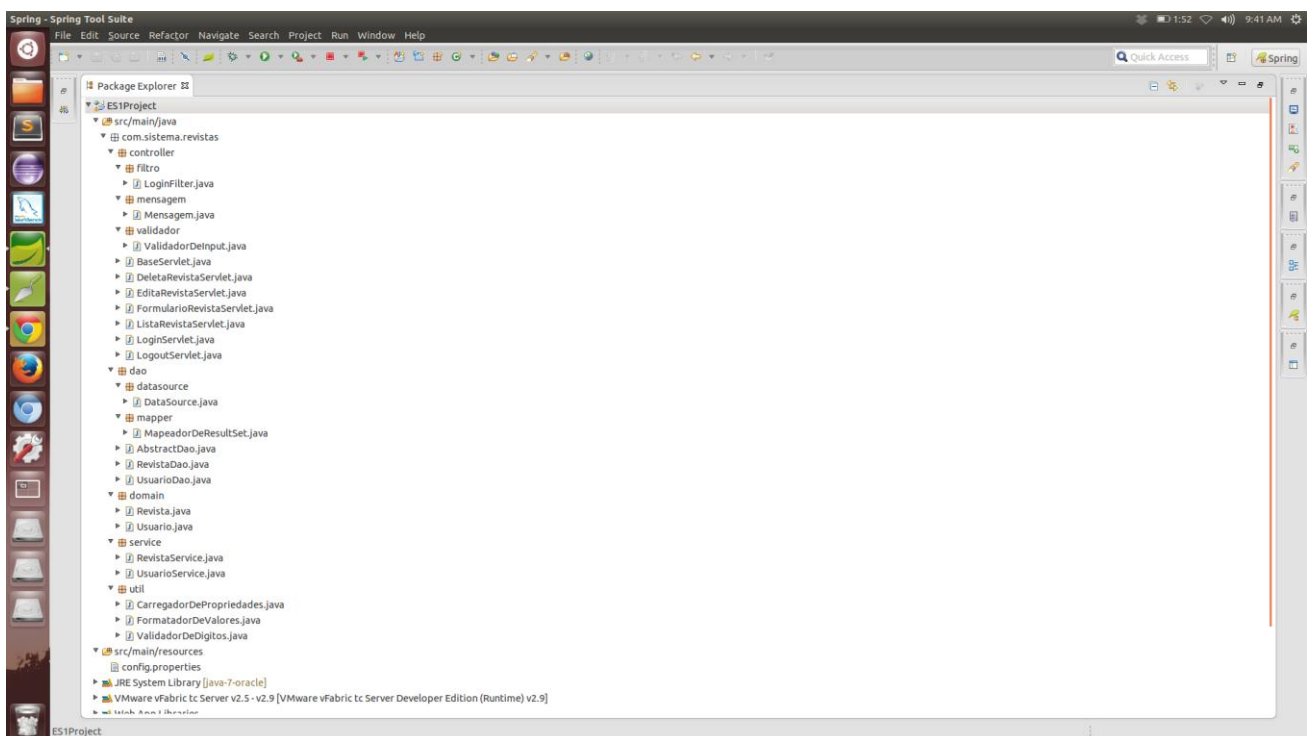
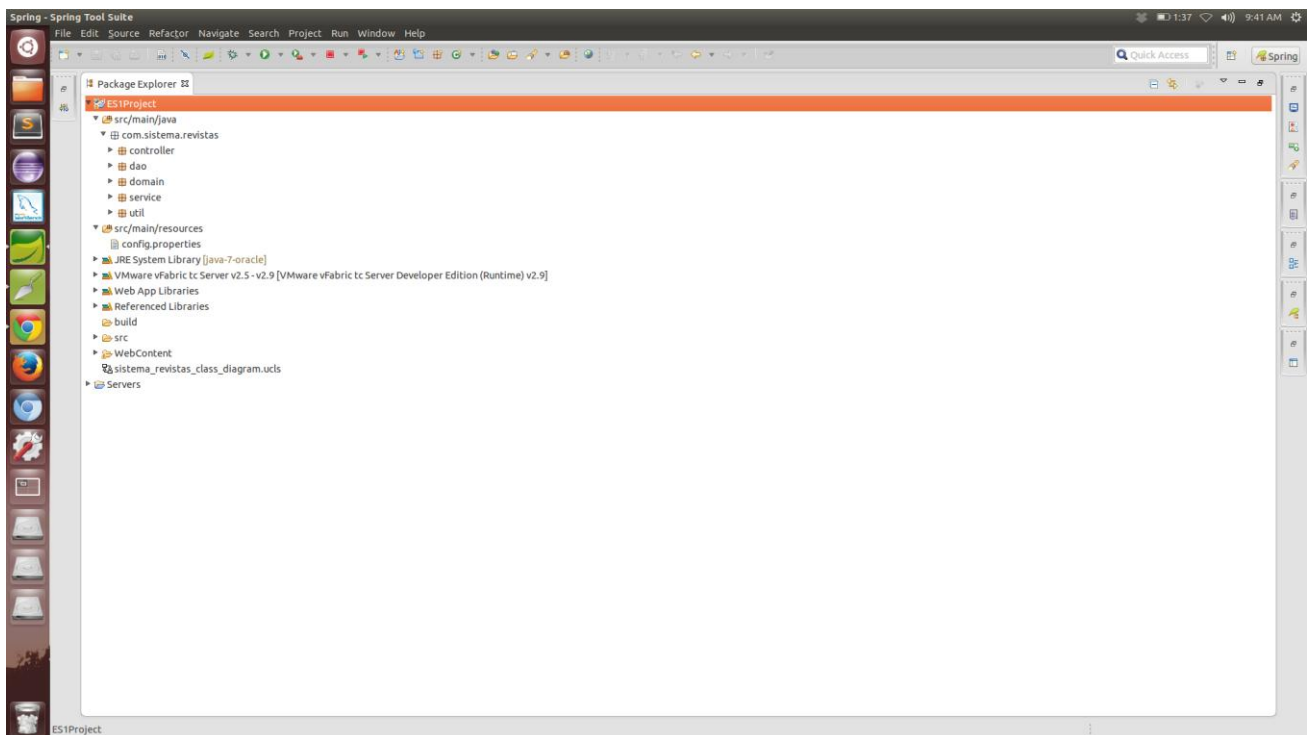
Design Patterns usadas na implementação do projeto:

- MVC
- Singleton
- DAO, Generic DAO (DAO abstrato e genérico)
- Factory Method (Alternative Constructor)
- Builder

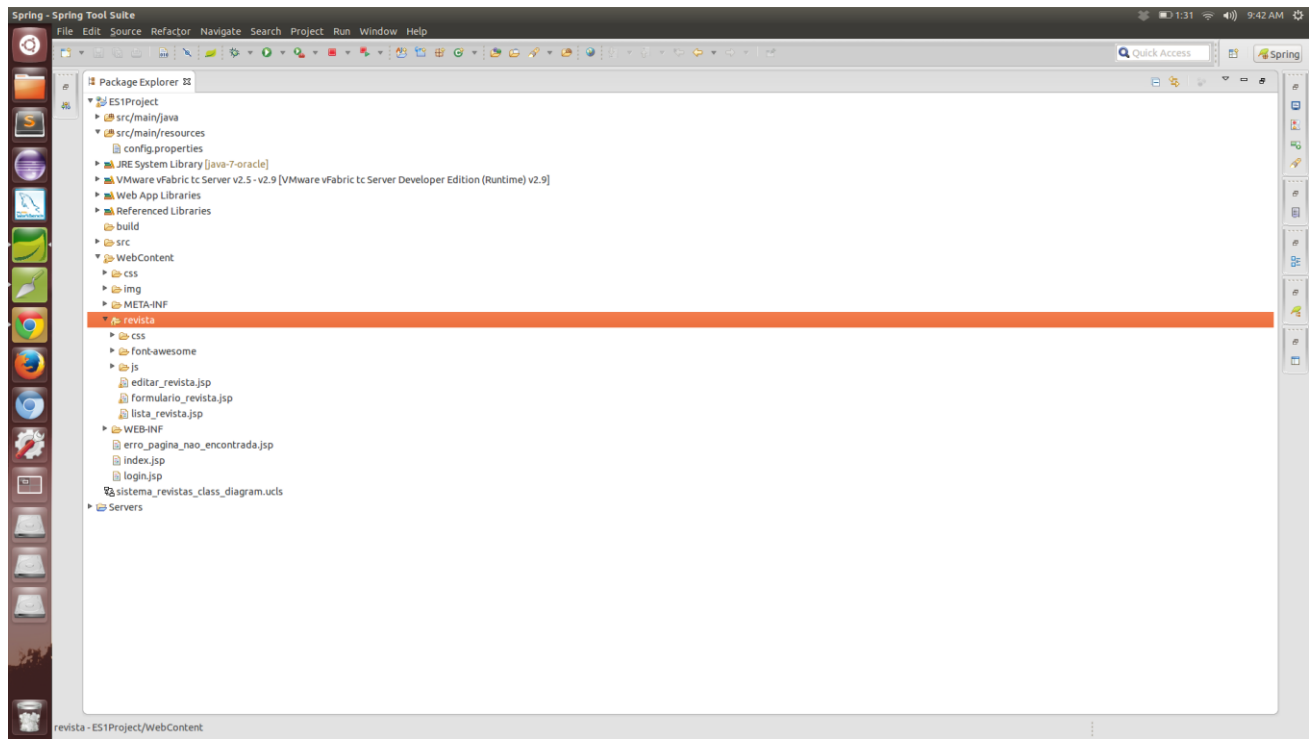
Ferramentas utilizadas:

- Eclipse Spring Tool Suite (STS) para Java EE
- Astah Professional
- Github para versionamento
- Servidor Jetty - plugin pro STS

Exibição do projeto distribuído em camadas de pacotes no Eclipse Spring Tool Suite (STS – IDE de Java EE):



Pasta de WebContent com as views e arquivos referentes às telas :



Classes de Modelo (pacote 'domain'):

Revista.java :

```
package com.sistema.revistas.domain;

import java.math.BigDecimal;
import com.sistema.revistas.util.FormatadorDeValores;

public class Revista {

    private Long id;
    private String nome;
    private Boolean temDigital;
    private String tema;
    private String descricao;
    private BigDecimal precoDe;
    private BigDecimal precoPor;
    private Boolean estaDeletado;

    public static class RevistaBuilder {

        private Long id;
        private String nome;
        private Boolean temDigital;
        private String tema;
        private String descricao;
        private BigDecimal precoDe;
        private BigDecimal precoPor;
        private Boolean estaDeletado = false;

        public RevistaBuilder id(Long val) {
            id = val;
            return this;
        }

        public RevistaBuilder nome(String val) {
            nome = val;
            return this;
        }

        public RevistaBuilder temDigital(Boolean val) {
            temDigital = val;
            return this;
        }

        public RevistaBuilder tema(String val) {
            tipo = val;
            return this;
        }
    }
}
```

```

    public RevistaBuilder descricao(String val) {
        descricao = val;
        return this;
    }

    public RevistaBuilder precoDe(BigDecimal val) {
        precoDe = val;
        return this;
    }

    public RevistaBuilder precoPor(BigDecimal val) {
        precoPor = val;
        return this;
    }

    public RevistaBuilder estaDeletado(Boolean val) {
        estaDeletado = val;
        return this;
    }

    public Revista build() {
        return new Revista(this);
    }
}

public Revista(RevistaBuilder revistaBuilder) {
    this.nome = revistaBuilder.nome;
    this.id = revistaBuilder.id;
    this.temDigital = revistaBuilder.temDigital;
    this.tema = revistaBuilder.tema;
    this.descricao = revistaBuilder.descricao;
    this.precoDe = revistaBuilder.precoDe;
    this.precoPor = revistaBuilder.precoPor;
    this.estaDeletado = revistaBuilder.estaDeletado;
}

public Long getId() {
    return id;
}

public String getNome() {
    return nome;
}

public String getTema() {
    return tema;
}

public String getDescricao() {

```

```

        return descricao;
    }

    public Boolean getTemDigital() {
        return temDigital;
    }

    public BigDecimal getPrecoDe() {
        return precoDe;
    }

    public BigDecimal getPrecoPor() {
        return precoPor;
    }

    public Boolean getEstaDeletado() {
        return estaDeletado;
    }

    public String getPrecoDeFormatado() {
        return FormatadorDeValores.formataNumero(precoDe);
    }

    public String getPrecoPorFormatado() {
        return FormatadorDeValores.formataNumero(precoPor);
    }
}

```


Usuario.java :

```
package com.sistema.revistas.domain;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class Usuario {

    private Long id;
    private String nome;
    private String sobrenome;
    private String senha;
    private Boolean usuarioGerente;
    private Boolean estaDeletado;

    public static class UsuarioBuilder {

        private Long id;
        private String nome;
        private String sobrenome;
        private String senha;
        private Boolean usuarioGerente;
        private Boolean estaDeletado = false;

        public UsuarioBuilder id(Long val) {
            id = val;
            return this;
        }

        public UsuarioBuilder nome(String val) {
            nome = val;
            return this;
        }

        public UsuarioBuilder estaDeletado(Boolean val) {
            estaDeletado = val;
            return this;
        }

        public UsuarioBuilder sobrenome(String val) {
            sobrenome = val;
            return this;
        }

        public UsuarioBuilder senha(String val) {
            senha = val;
            return this;
        }
    }
}
```

```

public UsuarioBuilder usuarioGerente(Boolean val) {
    usuarioGerente = val;
    return this;
}

public Usuario build() {
    return new Usuario(this);
}

}

public Usuario(UsuarioBuilder usuarioBuilder) {
    this.id = usuarioBuilder.id;
    this.nome = usuarioBuilder.nome;
    this.sobrenome = usuarioBuilder.sobrenome;
    this.senha = tornaSenhaSegura(usuarioBuilder.senha);
    this.usuarioGerente = usuarioBuilder.usuarioGerente;
    this.estaDeletado = usuarioBuilder.estaDeletado;
}

private String tornaSenhaSegura(String senhaParaHash) {
    String salt = "sistemarevista";
    String senhaGerada = null;

    try {
        MessageDigest md = MessageDigest.getInstance("SHA-512");
        md.update(salt.getBytes());

        byte[] bytes = md.digest(senhaParaHash.getBytes());
        StringBuilder sb = new StringBuilder();

        for(int i = 0; i < bytes.length; i++) {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100,
16).substring(1));
        }

        senhaGerada = sb.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }

    return senhaGerada;
}

public Long getId() {
    return id;
}

public String getNome() {
    return nome;
}

```

```
}

public String getSobrenome() {
    return sobrenome;
}

public String getSenha() {
    return senha;
}

public Boolean getUsuarioGerente() {
    return usuarioGerente;
}

public Boolean getEstaDeletado() {
    return estaDeletado;
}

}
```

Classes DAO (pacote 'DAO'):

GenericDao.java :

```
package com.sistema.revistas.dao;

import java.util.List;
import com.sistema.revistas.dao.datasource.DataSource;

public abstract class GenericDao<T> {

    protected DataSource dataSource;
    protected static final String CAMINHO_DO_ARQUIVO = "src/main/resources/config.properties";

    public abstract Boolean insereObjeto(T objeto);
    public abstract Boolean atualizaObjeto(T objeto);
    public abstract Boolean deletaElemento(T objeto);
    public abstract List<T> encontraTodosOsObjetos();
    public abstract T encontraObjetoPorId(Long id);

}
```

DataSource.java :

```
package com.sistema.revistas.dao.datasource;

public class DataSource {

    private String url;
    private String username;
    private String password;
    private String driverClass;

    public DataSource(String url, String username, String password, String driverClass) {
        this.url = url;
        this.username = username;
        this.password = password;
        this.driverClass = driverClass;
    }

    public String getUrl() {
        return url;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }

    public String getDriverClass() {
        return driverClass;
    }

}
```

RevistaDao.java :

```
package com.sistema.revistas.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.dao.datasource.DataSource;
import com.sistema.revistas.dao.mapper.MapeadorDeResultSet;
import com.sistema.revistas.util.CarregadorDePropriedades;

public class RevistaDao extends GenericDao<Revista> {

    private static class Loader {
        private static RevistaDao revistaDaoInstancia = new RevistaDao();
    }

    private RevistaDao() {
        Properties propriedades =
            CarregadorDePropriedades.carregaPropriedades(CAMINHO_DO_ARQUIVO);

        dataSource = new DataSource(propriedades.getProperty("dataSource.url"),
            propriedades.getProperty("dataSource.username"),
            propriedades.getProperty("dataSource.password"),

            propriedades.getProperty("dataSource.driver"));
    }

    public static RevistaDao getInstanciaDeRevistaDao() {
        return Loader.revistaDaoInstancia;
    }

    @Override
    public Boolean insereObjeto(Revista revista) {
        try {
            Class.forName(dataSource.getDriverClass());
            Connection connection = DriverManager.getConnection(dataSource.getUrl(),
                dataSource.getUsername(), dataSource.getPassword());

            String sql = "INSERT INTO REVISTA (NM_NOME, BO_TEM_DIGITAL, NR_PRECO_DE,
                NR_PRECO_POR, " +
                "BO_ESTA_DELETADO, NM_DESCRICAO, NM_TEMA) VALUES (?, ?, ?, ?, ?, ?, ?)";
```

```

        PreparedStatement statement = connection.prepareStatement(sql);

        statement.setString(1, revista.getNome());
        statement.setBoolean(2, revista.getTemDigital());
        statement.setBigDecimal(3, revista.getPrecoDe());
        statement.setBigDecimal(4, revista.getPrecoPor());
        statement.setBoolean(5, revista.getEstaDeletado());
        statement.setString(6, revista.getDescricao());
        statement.setString(7, revista.getTema());

        statement.execute();

        fechaRecursos(connection, statement);
        return true;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return false;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

@Override
public Boolean atualizaObjeto(Revista revista) {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "UPDATE REVISTA SET NM_NOME = ?, BO_TEM_DIGITAL = ?, NR_PRECO_DE =
?, NR_PRECO_POR = ?, BO_ESTA_DELETADO = ?, "
                    + "NM_DESCRICAO = ?, NM_TEMA = ? WHERE CD_REVISTA = ?";

        PreparedStatement statement = connection.prepareStatement(sql);

        statement.setString(1, revista.getNome());
        statement.setBoolean(2, revista.getTemDigital());
        statement.setBigDecimal(3, revista.getPrecoDe());
        statement.setBigDecimal(4, revista.getPrecoPor());
        statement.setBoolean(5, revista.getEstaDeletado());
        statement.setString(6, revista.getDescricao());
        statement.setString(7, revista.getTema());
        statement.setLong(8, revista.getId());

        statement.execute();

        fechaRecursos(connection, statement);
        return true;
    } catch (ClassNotFoundException e) {

```

```

        e.printStackTrace();
        return false;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

@Override
public Boolean deletaElemento(Revista revista) {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "UPDATE REVISTA SET BO_ESTA_DELETADO = ? WHERE CD_REVISTA = ?";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setBoolean(1, true);
        statement.setLong(2, revista.getId());
        statement.execute();

        fechaRecursos(connection, statement);
        return true;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return false;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

@Override
public List<Revista> encontraTodosOsObjetos() {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "SELECT * FROM REVISTA WHERE BO_ESTA_DELETADO = 0";
        Statement statement = connection.createStatement();
        ResultSet rs = statement.executeQuery(sql);

        ArrayList<Revista> ListaDeRevistas = new ArrayList<Revista>();
        while (rs.next()) {
            ListaDeRevistas.add(MapeadorDeResultSet.mapeiaRevista(rs));
        }

        fechaRecursos(connection, statement, rs);
        return ListaDeRevistas;
    }
}

```



```

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return new ArrayList<Revista>();
    } catch (SQLException e) {
        e.printStackTrace();
        return new ArrayList<Revista>();
    }
}

@Override
public Revista encontraObjetoPorId(Long id) {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "SELECT * FROM REVISTA WHERE CD_REVISTA = ? AND BO_ESTA_DELETADO =
0";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setLong(1, id);

        ResultSet rs = statement.executeQuery();

        if (rs.next()) {
            Revista revista = MapeadorDeResultSet.mapeiaRevista(rs);
            fechaRecursos(connection, statement, rs);

            return revista;
        }

        fechaRecursos(connection, statement, rs);
        return null;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return null;
    } catch (SQLException e) {
        e.printStackTrace();
        return null;
    }
}

private void fechaRecursos(Connection connection, Statement statement) throws SQLException {
    fechaRecursos(connection, statement, null);
}

private void fechaRecursos(Connection connection, Statement statement, ResultSet resultSet) throws
SQLException {
    if(resultSet != null) {
        resultSet.close();
    }
}

```

```
        if(statement != null) {  
            statement.close();  
        }  
  
        if(connection != null) {  
            connection.close();  
        }  
    }  
}
```

UsuarioDao.java :

```
package com.sistema.revistas.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;
import com.sistema.revistas.domain.Usuario;
import com.sistema.revistas.dao.datasource.DataSource;
import com.sistema.revistas.dao.mapper.MapeadorDeResultSet;
import com.sistema.revistas.util.CarregadorDePropriedades;

public class UsuarioDao extends GenericDao<Usuario> {

    private static class Loader {
        private static UsuarioDao usuarioDaoInstancia = new UsuarioDao();
    }

    private UsuarioDao() {
        Properties propiedades =
            CarregadorDePropriedades.carregaPropriedades(CAMINHO_DO_ARQUIVO);

        dataSource = new DataSource(propiedades.getProperty("dataSource.url"),
            propiedades.getProperty("dataSource.username"),
            propiedades.getProperty("dataSource.password"),
            propiedades.getProperty("dataSource.driver"));
    }

    public static UsuarioDao getInstanciaDeUsuarioDao() {
        return Loader.usuarioDaoInstancia;
    }

    public Usuario autenticaUsuario(Usuario usuario) {
        try {
            Class.forName(dataSource.getDriverClass());
            Connection connection = DriverManager.getConnection(dataSource.getUrl(),
                dataSource.getUsername(), dataSource.getPassword());

            String sql = "SELECT * FROM USUARIO WHERE NM_NOME = ? AND NM_SENHA = ? AND\n"
                + "BO_E_GERENTE = 1 AND BO_ESTA_DELETADO = 0";
            PreparedStatement statement = connection.prepareStatement(sql);
            statement.setString(1, usuario.getNome());
```

```

        statement.setString(2, usuario.getSenha());

        ResultSet rs = statement.executeQuery();

        if (rs.next()) {
            Usuario usuarioDoBanco = MapeadorDeResultSet.mapeiaUsuario(rs);
            fechaRecursos(connection, statement, rs);

            return usuarioDoBanco;
        }

        fechaRecursos(connection, statement, rs);
        return null;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return null;
    } catch (SQLException e) {
        e.printStackTrace();
        return null;
    }
}

@Override
public Boolean insereObjeto(Usuario usuario) {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "INSERT INTO USUARIO (NM_NOME, NM_SOBRENOME, NM_SENHA, " +
"BO_ESTA_DELETADO, BO_E_GERENTE) VALUES (?, ?, ?, ?, ?)";

        PreparedStatement statement = connection.prepareStatement(sql);

        statement.setString(1, usuario.getNome());
        statement.setString(2, usuario.getSobrenome());
        statement.setString(3, usuario.getSenha());
        statement.setBoolean(4, usuario.getEstaDeletado());
        statement.setBoolean(5, usuario.getUsuarioGerente());

        statement.execute();

        fechaRecursos(connection, statement);
        return true;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return false;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

```

```

    }

    @Override
    public Boolean atualizaObjeto(Usuario usuario) {
        try {
            Class.forName(dataSource.getDriverClass());
            Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

            String sql = "UPDATE USUARIO SET NM_NOME = ?, NM_SOBRENOME = ?, NM_SENHA = ?,
BO_ESTA_DELETADO = ?, BO_E_GERENTE = ?"
                        + " WHERE CD_USUARIO = ?";

            PreparedStatement statement = connection.prepareStatement(sql);

            statement.setString(1, usuario.getNome());
            statement.setString(2, usuario.getSobrenome());
            statement.setString(3, usuario.getSenha());
            statement.setBoolean(4, usuario.getEstaDeletado());
            statement.setBoolean(5, usuario.getUsuarioGerente());
            statement.setLong(6, usuario.getId());

            statement.execute();

            fechaRecursos(connection, statement);
            return true;
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            return false;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    @Override
    public Boolean deletaElemento(Usuario usuario) {
        try {
            Class.forName(dataSource.getDriverClass());
            Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

            String sql = "UPDATE USUARIO SET BO_ESTA_DELETADO = ? WHERE CD_USUARIO = ?";

            PreparedStatement statement = connection.prepareStatement(sql);
            statement.setBoolean(1, true);
            statement.setLong(2, usuario.getId());
            statement.execute();

            fechaRecursos(connection, statement);
            return true;
        }
    }

```

```

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return false;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

@Override
public List<Usuario> encontraTodosOsObjetos() {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "SELECT * FROM USUARIO WHERE BO_ESTA_DELETADO = 0";
        Statement statement = connection.createStatement();
        ResultSet rs = statement.executeQuery(sql);

        ArrayList<Usuario> ListaDeUsuarios = new ArrayList<Usuario>();
        while (rs.next()) {
            ListaDeUsuarios.add(MapeadorDeResultSet.mapeiaUsuario(rs));
        }

        fechaRecursos(connection, statement, rs);
        return ListaDeUsuarios;
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
        return new ArrayList<Usuario>();
    } catch (SQLException e) {
        e.printStackTrace();
        return new ArrayList<Usuario>();
    }
}

@Override
public Usuario encontraObjetoPorId(Long id) {
    try {
        Class.forName(dataSource.getDriverClass());
        Connection connection = DriverManager.getConnection(dataSource.getUrl(),
dataSource.getUsername(), dataSource.getPassword());

        String sql = "SELECT * FROM USUARIO WHERE CD_USUARIO = ? AND BO_ESTA_DELETADO =
0";

        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setLong(1, id);

        ResultSet rs = statement.executeQuery();

        if (rs.next()) {

```

```

        Usuario usuario = MapeadorDeResultSet.mapeiaUsuario(rs);
        fechaRecursos(connection, statement, rs);

        return usuario;
    }

    fechaRecursos(connection, statement, rs);
    return null;
} catch (ClassNotFoundException e) {
    e.printStackTrace();
    return null;
} catch (SQLException e) {
    e.printStackTrace();
    return null;
}
}

private void fechaRecursos(Connection connection, Statement statement) throws SQLException {
    fechaRecursos(connection, statement, null);
}

private void fechaRecursos(Connection connection, Statement statement, ResultSet resultSet) throws
SQLException {
    if(resultSet != null) {
        resultSet.close();
    }

    if(statement != null) {
        statement.close();
    }

    if(connection != null) {
        connection.close();
    }
}
}

```

MapeadorDeResultSet.java :

```
package com.sistema.revistas.dao.mapper;

import java.sql.ResultSet;
import java.sql.SQLException;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.domain.Usuario;

public class MapeadorDeResultSet {

    public static Usuario mapeiaUsuario(ResultSet rs) throws SQLException {
        Usuario usuario = new
        Usuario.UsuarioBuilder().nome(rs.getString("NM_NOME")).sobrenome(rs.getString("NM_SOBRENOME")).

        senha(rs.getString("NM_SENHA")).usuarioGerente(rs.getBoolean("BO_E_GERENTE")).id(rs.getLong("CD_USUA
RIO")).

        estaDeletado(rs.getBoolean("BO_ESTA_DELETADO")).build();

        return usuario;
    }

    public static Revista mapeiaRevista(ResultSet rs) throws SQLException {
        Revista revista = new
        Revista.RevistaBuilder().id(rs.getLong("CD_REVISTA")).nome(rs.getString("NM_NOME")).

        temDigital(rs.getBoolean("BO_TEM_DIGITAL")).precoDe(rs.getBigDecimal("NR_PRECO_DE")).

        precoPor(rs.getBigDecimal("NR_PRECO_POR")).estaDeletado(rs.getBoolean("BO_ESTA_DELETADO")).descricao
(rs.getString("NM_DESCRICAO")).

        tema(rs.getString("NM_TEMA")).build();

        return revista;
    }
}
```


Classes de Serviço (pacote 'service'):

RevistaService.java :

```
package com.sistema.revistas.service;

import java.util.List;
import com.sistema.revistas.dao.RevistaDao;
import com.sistema.revistas.domain.Revista;

public class RevistaService {

    private RevistaDao revistaDao;

    private static class Loader {
        private static RevistaService revistaServiceInstancia = new RevistaService();
    }

    private RevistaService() {
        revistaDao = RevistaDao.getInstanciaDeRevistaDao();
    }

    public static RevistaService getInstanciaDeRevistaService() {
        return Loader.revistaServiceInstancia;
    }

    public List<Revista> consultaTodasAsRevistas() {
        return revistaDao.encontraTodosOsObjetos();
    }

    public Boolean excluiRevista(Revista revista) {
        return revistaDao.deletaElemento(revista);
    }

    public Revista consultaRevistaPorId(Long id) {
        return revistaDao.encontraObjetoPorId(id);
    }

    public Boolean criaRevista(Revista revista) {
        return revistaDao.insereObjeto(revista);
    }

    public Boolean atualizaRevista(Revista revista) {
        return revistaDao.atualizaObjeto(revista);
    }
}
```

UsuarioService.java :

```
package com.sistema.revistas.service;
import java.util.List;
import com.sistema.revistas.dao.UsuarioDao;
import com.sistema.revistas.domain.Usuario;

public class UsuarioService {

    private UsuarioDao usuarioDao;

    private static class Loader {
        private static UsuarioService usuarioServiceInstancia = new UsuarioService();
    }

    private UsuarioService() {
        usuarioDao = UsuarioDao.getInstanciaDeUsuarioDao();
    }

    public static UsuarioService getInstanciaDeUsuarioService() {
        return Loader.usuarioServiceInstancia;
    }

    public Usuario autenticaLoginDeUsuario(Usuario usuario) {
        return usuarioDao.autenticaUsuario(usuario);
    }

    public Boolean removeUsuario(Usuario usuario) {
        return usuarioDao.deletaElemento(usuario);
    }

    public Boolean insereUsuario(Usuario usuario) {
        return usuarioDao.insereObjeto(usuario);
    }

    public Boolean atualizaUsuario(Usuario usuario) {
        return usuarioDao.atualizaObjeto(usuario);
    }

    public Usuario consultaUsuarioPorId(Long id) {
        return usuarioDao.encontraObjetoPorId(id);
    }

    public List<Usuario> consultaTodosOsUsuarios() {
        return usuarioDao.encontraTodosOsObjetos();
    }
}
```

Classes de Controle (pacote 'controller'):

BaseServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.controller.mensagem.Mensagem;

public abstract class BaseServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public void renderizaJSP(HttpServletRequest request, HttpServletResponse response, String urlPagina)
throws IOException {
        if (request.getParameter("mensagem") == null) {
            request.getSession().setAttribute("mensagem", Mensagem.mensagemNulaFactory());
        }

        response.sendRedirect(urlPagina);
    }
}
```

DeletaRevistaServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.controller.mensagem.Mensagem;
import com.sistema.revistas.controller.validador.ValidadorDeInput;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.service.RevistaService;

public class DeletaRevistaServlet extends BaseServlet {

    private static final long serialVersionUID = 1L;
    private RevistaService revistaService;

    public DeletaRevistaServlet() {
        super();
        revistaService = RevistaService.getInstanciaDeRevistaService();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if (!ValidadorDeInput.validaId(request)) {
            request.setAttribute("mensagem", Mensagem.erroIdInvalidoFactory());
        } else {
            Long id = new Long(request.getParameter("id"));
            Revista revista = revistaService.consultaRevistaPorId(id);
            excluiRevista(request, revista);
        }

        response.sendRedirect("listaRevista?mensagem=true");
    }

    private void excluiRevista(HttpServletRequest request, Revista revista) {
        if (revista != null && revistaService.excluiRevista(revista)) {
            request.getSession().setAttribute("mensagem",
Mensagem.deletarComSucessoFactory());
        } else {
            request.getSession().setAttribute("mensagem",
Mensagem.deletarComErroFactory());
        }
    }
}
```

EditaRevistaServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import java.math.BigDecimal;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.controller.mensagem.Mensagem;
import com.sistema.revistas.controller.validador.ValidadorDeInput;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.service.RevistaService;

public class EditaRevistaServlet extends BaseServlet {

    private static final long serialVersionUID = 1L;
    private RevistaService revistaService;

    public EditaRevistaServlet() {
        super();
        revistaService = RevistaService.getInstanciaDeRevistaService();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if (!ValidadorDeInput.validaId(request)) {
            request.getSession().setAttribute("revista", null);
        } else {
            Long id = new Long(request.getParameter("id"));
            Revista revista = revistaService.consultaRevistaPorId(id);
            request.getSession().setAttribute("revista", revista);
        }
        renderizaJSP(request, response, "editar_revista.jsp");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if (ValidadorDeInput.validaEdicaoDeRevista(request)) {
            Revista revista = new Revista.RevistaBuilder().id(new
Long(request.getParameter("id"))).nome(request.getParameter("nome")).
temDigital(new
Boolean(request.getParameter("temDigital"))).precoDe(new BigDecimal(request.getParameter("precoDe"))).
precoPor(new
BigDecimal(request.getParameter("precoPor"))).estaDeletado(new Boolean(request.getParameter("estaDeletado"))).
descricao(request.getParameter("descricao")).tema(request.getParameter("tema")).build();
            editaRevista(request, revista);
        }
        response.sendRedirect("editaRevista?id=" + request.getParameter("id")+ "&mensagem=true");
    }

    private void editaRevista(HttpServletRequest request, Revista revista) {
        if (revistaService.atualizaRevista(revista)) {
            request.getSession().setAttribute("mensagem",
Mensagem.editarComSucessoFactory());
        } else {
            request.getSession().setAttribute("mensagem", Mensagem.editarComErroFactory());
        }
    }
}
```

FormularioRevistaServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import java.math.BigDecimal;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.controller.mensagem.Mensagem;
import com.sistema.revistas.controller.validador.ValidadorDeInput;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.service.RevistaService;

public class FormularioRevistaServlet extends BaseServlet{

    private static final long serialVersionUID = 1L;
    private RevistaService revistaService;

    public FormularioRevistaServlet() {
        super();
        revistaService = RevistaService.getInstanciaDeRevistaService();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        renderizaJSP(request, response, "formulario_revista.jsp");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        if (ValidadorDeInput.validaCriacaoDeRevista(request)) {
            Revista revista = new
Revista.RevistaBuilder().nome(request.getParameter("nome")).
                temDigital(new
Boolean(request.getParameter("temDigital"))).precoPor(new BigDecimal(request.getParameter("precoPor"))).
                precoDe(new
BigDecimal(request.getParameter("precoDe"))).descricao(request.getParameter("descricao")).
                tema(request.getParameter("tema")).build();

            cadastraRevista(request, revista);
        }

        response.sendRedirect("formularioRevista?mensagem=true");
    }

    private void cadastraRevista(HttpServletRequest request, Revista revista) {
        if (revistaService.criaRevista(revista)) {
            request.getSession().setAttribute("mensagem",
Mensagem.cadastrarComSucessoFactory());
        } else {
            request.getSession().setAttribute("mensagem",
Mensagem.cadastrarComErroFactory());
        }
    }
}
```

ListaRevistaServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.domain.Revista;
import com.sistema.revistas.service.RevistaService;

public class ListaRevistaServlet extends BaseServlet {

    private static final long serialVersionUID = 1L;
    private RevistaService revistaService;

    public ListaRevistaServlet() {
        super();
        revistaService = RevistaService.getInstanciaDeRevistaService();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        List<Revista> revistas = revistaService.consultaTodasAsRevistas();
        request.getSession().setAttribute("revistas", revistas);

        renderizaJSP(request, response, "lista_revista.jsp");
    }
}
```

LoginServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.controller.mensagem.Mensagem;
import com.sistema.revistas.domain.Usuario;
import com.sistema.revistas.service.UsuarioService;

public class LoginServlet extends BaseServlet {

    private static final long serialVersionUID = 1L;
    private UsuarioService usuarioService;

    public LoginServlet() {
        super();
        usuarioService = UsuarioService.getInstanciaDeUsuarioService();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        renderizaJSP(request, response, "login.jsp");
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        Usuario usuario = new
Usuario.UsuarioBuilder().nome(request.getParameter("nome")).senha(request.getParameter("senha")).build();
        Usuario usuarioDoBanco = usuarioService.autenticaLoginDeUsuario(usuario);

        if (usuarioDoBanco == null) {
            request.getSession().setAttribute("mensagem", Mensagem.loginComErroFactory());
            response.sendRedirect("login?mensagem=true");
        } else {
            //Sessão setada para uma hora
            request.getSession().setAttribute("usuario", usuario);
            request.getSession().setMaxInactiveInterval(30 * 2 * 60);
            response.sendRedirect("revista/listaRevista");
        }
    }
}
```


LogoutServlet.java :

```
package com.sistema.revistas.controller;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class LogoutServlet extends BaseServlet {

    private static final long serialVersionUID = 1L;

    public LogoutServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        request.getSession().invalidate();
        response.sendRedirect("http://localhost:8080/ES1Project/");
    }
}
```

LoginFilter.java :

```
package com.sistema.revistas.controller.filtro;

import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.sistema.revistas.domain.Usuario;

public class LoginFilter implements Filter {

    public LoginFilter() {
        super();
    }

    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws
    IOException, ServletException {
        HttpServletRequest httpRequest = (HttpServletRequest) request;
        HttpServletResponse httpResponse = (HttpServletResponse) response;

        Usuario usuario = (Usuario) httpRequest.getSession().getAttribute("usuario");

        if (usuario == null) {
            httpResponse.sendRedirect("http://localhost:8080/ES1Project/login");
        } else {
            chain.doFilter(request, response);
        }
    }

    @Override
    public void destroy() {}

    @Override
    public void init(FilterConfig arg0) throws ServletException {}
}
```

Mensagem :

```
package com.sistema.revistas.controller.mensagem;

public class Mensagem {

    private String tipo;
    private String texto;

    private Mensagem(String tipo, String texto) {
        this.tipo = tipo;
        this.texto = texto;
    }

    public String getTipo() {
        return tipo;
    }

    public String getTexto() {
        return texto;
    }

    public static Mensagem cadastrarComSucessoFactory() {
        return new Mensagem("mensagemSucesso", "Revista cadastrada com sucesso.");
    }

    public static Mensagem cadastrarComErroFactory() {
        return new Mensagem("mensagemErro", "Erro ao cadastrar revista.");
    }

    public static Mensagem editarComSucessoFactory() {
        return new Mensagem("mensagemSucesso", "Revista editada com sucesso.");
    }

    public static Mensagem deletarComErroFactory() {
        return new Mensagem("mensagemErro", "Erro ao excluir revista.");
    }

    public static Mensagem deletarComSucessoFactory() {
        return new Mensagem("mensagemSucesso", "Revista excluída com sucesso.");
    }

    public static Mensagem editarComErroFactory() {
        return new Mensagem("mensagemErro", "Erro ao editar revista.");
    }

    public static Mensagem loginComErroFactory() {
        return new Mensagem("mensagemErro", "Senha ou nome de usuário inválidos.");
    }

    public static Mensagem erroIdInvalidoFactory() {
        return new Mensagem("mensagemErro", "Id de revista inválido.");
    }

    public static Mensagem mensagemNulaFactory() {
        return null;
    }
}
```

ValidadorDeInput.java :

```
package com.sistema.revistas.controller.validador;

import java.math.BigDecimal;
import javax.servlet.http.HttpServletRequest;
import com.sistema.revistas.util.ValidadorDeDigitos;

public class ValidadorDeInput {

    public static Boolean validaId(HttpServletRequest request) {
        if (!ValidadorDeDigitos.validaNumero(request.getParameter("id"))) {
            return false;
        }

        return true;
    }

    public static Boolean validaEdicaoDeRevista(HttpServletRequest request) {
        if (validaCamposObrigatoriosDeRevista(request) &&
            validaCamposBooleano(request.getParameter("temDigital"))
            && validaPrecos(request.getParameter("precoDe"),
            request.getParameter("precoPor")) &&
            validaCamposBooleano(request.getParameter("estaDeletado")) &&
            ValidadorDeDigitos.validaNumero(request.getParameter("id"))) {

            return true;
        }

        return false;
    }

    public static Boolean validaCriacaoDeRevista(HttpServletRequest request) {
        if (validaCamposObrigatoriosDeRevista(request) &&
            validaCamposBooleano(request.getParameter("temDigital"))
            && validaPrecos(request.getParameter("precoDe"),
            request.getParameter("precoPor"))) {

            return true;
        }

        return false;
    }

    private static Boolean validaPrecos(String precoDe, String precoPor) {
        try {
            BigDecimal precoDeBigDecimal = new BigDecimal(precoDe);
            BigDecimal precoPorBigDecimal = new BigDecimal(precoPor);

            if (precoDeBigDecimal.compareTo(precoPorBigDecimal) < 0) {
                return false;
            }

            return true;
        } catch (Exception e) {
            return false;
        }
    }

    private static Boolean validaCamposBooleano(String temDigital) {
        if (!Boolean.TRUE.toString().equals(temDigital) &&
            !Boolean.FALSE.toString().equals(temDigital)) {
            return false;
        }

        return true;
    }
}
```

```
        private static Boolean validaCamposObrigatoriosDeRevista(HttpServletRequest request) {  
            return request.getParameter("nome") != null && request.getParameter("tema") != null &&  
request.getParameter("precoDe") != null &&  
            request.getParameter("precoPor") != null &&  
request.getParameter("temDigital") != null;  
        }  
  
    }
```

Classes de utilitários (pacote 'util'):

CarregadorDePropriedades.java :

```
package com.sistema.revistas.util;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;

public class CarregadorDePropriedades {

    public static Properties carregaPropriedades(String caminhoDoArquivo) {
        Properties propriedades = null;
        InputStream input = null;

        try {
            propriedades = new Properties();
            input = new FileInputStream(caminhoDoArquivo);
            propriedades.load(input);

            return propriedades;
        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            if (input != null) {
                try {
                    input.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }

        return propriedades;
    }
}
```

FormatadorDeValores.java :

```
package com.sistema.revistas.util;

import java.math.BigDecimal;
import java.text.DecimalFormat;
import java.text.DecimalFormatSymbols;
import java.util.Locale;

public class FormatadorDeValores {

    public static String formataNumero(BigDecimal value) {
        Locale local = new Locale("pt", "BR");
        DecimalFormat df = new DecimalFormat("#,##0.00", new DecimalFormatSymbols(local));

        return "R$ " + df.format(value);
    }
}
```

ValidadorDeDigitos.java :

```
package com.sistema.revistas.util;

public class ValidadorDeDigitos {

    private static final String QUANTITY_REGEX = "^(0|[1-9][0-9]*)$";

    public static boolean validaNumero(String input) {
        return input != null && input.matches(QUANTITY_REGEX);
    }

}
```


JSPs do projeto:

erro_pagina_nao_encontrada.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Página não encontrada</title>
<link rel="stylesheet" href="http://localhost:8080/ES1Project/css/erro.css">
</head>
<body>
    <div class="image-container"></div>
</body>
</html>
```

index.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Home</title>
</head>
<body>
    <%
        String redirectURL = "http://localhost:8080/ES1Project/revista/listaRevista";
        response.sendRedirect(redirectURL);
    %>
</body>
</html>
```

login.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<!--[if lt IE 7]> <html class="lt-ie9 lt-ie8 lt-ie7" lang="en"> <![endif]-->
<!--[if IE 7]> <html class="lt-ie9 lt-ie8" lang="en"> <![endif]-->
<!--[if IE 8]> <html class="lt-ie9" lang="en"> <![endif]-->
<!--[if gt IE 8]><!--> <html lang="en"> <!--<![endif]-->
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Sistema revistas - login</title>
  <link rel="stylesheet" href="css/style.css">
  <!--[if lt IE 9]><script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>
<body>
  <div class="title-container">
    <h1 class="title">Sistema Revistas</h1>
  </div>
  <div class="login-form">

    <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemErro'}">
      <div class="errormsgbox">${mensagem.texto}</div>
    </c:if>

    <form method="post" action="login" class="login">
      <p>
        <label for="login">Usuário:</label>
        <input type="text" name="nome" id="login">
      </p>

      <p>
        <label for="password">Senha:</label>
        <input type="password" name="senha" id="password">
      </p>

      <p class="login-submit">
        <button type="submit" class="login-button">Login</button>
      </p>

    </form>
  </div>
</body>
</html>
```

editar_revista.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Sistema Revistas - editar revista</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.css" rel="stylesheet">

    <!-- Add custom CSS here -->
    <link href="css/sb-admin.css" rel="stylesheet">
    <link rel="stylesheet" href="font-awesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="css/validationEngine.jquery.css">
    <link rel="stylesheet" href="css/jquery-ui.css">

  </head>

  <body>

    <div id="wrapper">

      <!-- Sidebar -->
      <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="">Sistema Revistas</a>
        </div>

        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse navbar-ex1-collapse">
          <ul class="nav navbar-nav side-nav">
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" id="menu-top" data-toggle="dropdown"><i class="fa fa-caret-square-o-down"></i> Gerenciar revistas <b class="caret"></b></a>
              <ul id="dropdown-menu-items" class="dropdown-menu">
                <li><a href="http://localhost:8080/ES1Project/revista/listaRevista"><i class="fa fa-table"></i> Exibir revistas cadastradas</a></li>
                <li><a href="http://localhost:8080/ES1Project/revista/formularioRevista"><i class="fa fa-edit"></i> Cadastrar nova revista</a></li>
              </ul>
            </li>

            <ul class="nav navbar-nav navbar-right navbar-user">
              <li class="dropdown user-dropdown">
                <a href="#" class="dropdown-toggle" data-toggle="dropdown"><i class="fa fa-user"></i>
                  ${usuario.nome} <b class="caret"></b></a>
                <ul class="dropdown-menu">
```

```

        <li><a href="http://localhost:8080/ES1Project/logout"><i class="fa fa-power-off"></i> Log
Out</a></li>
    </ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</nav>

<div id="page-wrapper">

    <div class="row">
        <div class="col-lg-12">
            <h1>Editar Revista <small> Entre com os dados</small></h1>
        </div>
    </div><!-- /.row -->
    <div class="form-magazine">

        <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemErro'}">
            <div class="errormsgbox">${mensagem.texto}</div>
        </c:if>
        <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemSucesso'}">
            <div class="successbox">${mensagem.texto}</div>
        </c:if>

        <div class="row">
            <div class="col-lg-6">

                <form class="form" method="post" action="editaRevista" role="form">

                    <input type="hidden" name="id" value="${revista.id}" class="form-
control" maxlength="100">
                    <input type="hidden" name="estaDeletado"
value="${revista.estaDeletado}" class="form-control" maxlength="100">

                    <div class="form-group">
                        <label class="non-radio-label">Nome da revista: <span class="required-field">*</span></label>
                        <input name="nome" value="${revista.nome}" autocomplete="off" class="form-control
nome" maxlength="100">
                    </div>

                    <div class="form-group">
                        <label class="non-radio-label">Tema: <span class="required-field">*</span></label>
                        <input name="tema" value="${revista.tema}" autocomplete="off" class="form-control
tema" maxlength="100">
                    </div>

                    <div class="form-group">
                        <label class="non-radio-label">Descrição da revista: </label>
                        <input name="descricao" value="${revista.descricao}" autocomplete="off" class="form-
control descricao" maxlength="100">
                    </div>

                    <div class="form-group">
                        <label class="non-radio-label">Preço Por: <span class="required-field">*</span></label>
                        <input name="precoPor" value="${revista.precoPor}" class="form-control precoPor"
maxlength="10">
                    </div>

                    <div class="form-group">
                        <label class="non-radio-label">Preço De: <span class="required-field">*</span></label>
                        <input name="precoDe" value="${revista.precoDe}" autocomplete="off" class="form-
control precoDe" maxlength="10">
                    </div>

```

```

        <c:if test="${revista.temDigital == true}">
        <div class="form-group">
            <label class="radio-label">Tem formato digital: <span class="required-
field">* </span></label>
            <label class="radio-inline">
                <input type="radio" name="temDigital" id="optionsRadiosInline1"
value="true" checked> Sim
            </label>
            <label class="radio-inline">
                <input type="radio" name="temDigital" id="optionsRadiosInline2"
value="false">Não
            </label>
        </div>
        </c:if>

        <c:if test="${revista.temDigital != true}">
        <div class="form-group">
            <label class="radio-label">Tem formato digital: <span class="required-
field">* </span></label>
            <label class="radio-inline">
                <input type="radio" name="temDigital" id="optionsRadiosInline1"
value="true"> Sim
            </label>
            <label class="radio-inline">
                <input type="radio" name="temDigital" id="optionsRadiosInline2"
value="false" checked> Não
            </label>
        </div>
        </c:if>

        <c:if test="${revista != null}">
            <button type="submit" class="btn btn-default">Editar revista</button>
        </c:if>
    </form>

    <br />
    <br />
    <br />
    <a class="back_button"
href="http://localhost:8080/ES1Project/revista/listaRevista">voltar para lista de revistas</a>
    </div>
</div><!-- /.row -->
</div>
</div><!-- /#page-wrapper -->

</div><!-- /#wrapper -->

<!-- JavaScript -->
<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.js"></script>
<script src="js/showhidemenu.js"></script>
<script src="js/jquery.mask.min.js"></script>
    <script src="js/jquery.validate.min.js"></script>
    <script src="js/jquery.validationEngine-pt_BR.js"></script>
    <script src="js/jquery.validationEngine.js"></script>
<script src="js/validate.revista.js"></script>

</body>
</html>

```

formulario_revista.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Sistema Revistas - cadastrar revista</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.css" rel="stylesheet">

    <!-- Add custom CSS here -->
    <link href="css/sb-admin.css" rel="stylesheet">
    <link rel="stylesheet" href="font-awesome/css/font-awesome.min.css">
    <link rel="stylesheet" href="css/validationEngine.jquery.css">
    <link rel="stylesheet" href="css/jquery-ui.css">

  </head>

  <body>

    <div id="wrapper">

      <!-- Sidebar -->
      <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="">Sistema Revistas</a>
        </div>

        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse navbar-ex1-collapse">
          <ul class="nav navbar-nav side-nav">
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" id="menu-top" data-toggle="dropdown"><i class="fa fa-caret-square-o-down"></i> Gerenciar revistas <b class="caret"></b></a>
              <ul id="dropdown-menu-items" class="dropdown-menu">
                <li><a href="http://localhost:8080/ES1Project/revista/listaRevista"><i class="fa fa-table"></i> Exibir revistas cadastradas</a></li>
                <li><a href="http://localhost:8080/ES1Project/revista/formularioRevista"><i class="fa fa-edit"></i> Cadastrar nova revista</a></li>
              </ul>
            </li>
          </ul>

          <ul class="nav navbar-nav navbar-right navbar-user">
            <li class="dropdown user-dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"><i class="fa fa-user"></i>
                ${usuario.nome} <b class="caret"></b></a>
              <ul class="dropdown-menu">
```

```

        <li><a href="http://localhost:8080/ES1Project/logout"><i class="fa fa-power-off"></i> Log
Out</a></li>
    </ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</nav>

<div id="page-wrapper">

    <div class="row">
        <div class="col-lg-12">
            <h1>Cadastrar Revista <small> Entre com os dados</small></h1>
        </div>
    </div><!-- /.row -->

    <div class="form-magazine">

        <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemErro'}">
            <div class="errormsgbox">${mensagem.texto}</div>
        </c:if>
        <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemSucesso'}">
            <div class="successbox">${mensagem.texto}</div>
        </c:if>

    <div class="row">
        <div class="col-lg-6">

            <form class="form" method="post" action="formularioRevista" role="form">

                <div class="form-group">
                    <label class="non-radio-label">Nome da revista: <span class="required-field">*</span></label>
                    <input name="nome" autocomplete="off" class="form-control nome" maxlength="100">
                </div>

                <div class="form-group">
                    <label class="non-radio-label">Tema: <span class="required-field">*</span></label>
                    <input name="tema" autocomplete="off" class="form-control tema" maxlength="100">
                </div>

                <div class="form-group">
                    <label class="non-radio-label">Descrição da revista: </label>
                    <input name="descricao" autocomplete="off" class="form-control descricao"
maxlength="100">
                </div>

                <div class="form-group">
                    <label class="non-radio-label">Preço Por: <span class="required-field">*</span></label>
                    <input name="precoPor" autocomplete="off" class="form-control precoPor"
maxlength="10">
                </div>

                <div class="form-group">
                    <label class="non-radio-label">Preço De: <span class="required-field">*</span></label>
                    <input name="precoDe" autocomplete="off" class="form-control precoDe"
maxlength="10">
                </div>

                <div class="form-group">
                    <label class="radio-label">Tem formato digital: <span class="required-field">*</span>
                    <label class="radio-inline">
                        <input type="radio" name="temDigital" id="optionsRadiosInline1" value="true"
checked> Sim

```



```

        </label>
        <label class="radio-inline">
            <input type="radio" name="temDigital" id="optionsRadiosInline2" value="false">Não
        </label>
    </div>

    <button type="submit" class="btn btn-default">Cadastrar revista</button>
    <button type="reset" class="btn btn-default">Resetar formulário</button>

</form>

</div>
</div><!-- /.row -->
</div>
</div><!-- /#page-wrapper -->

</div><!-- /#wrapper -->

<!-- JavaScript -->
<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.js"></script>
<script src="js/showhidemenu.js"></script>
<script src="js/jquery.mask.min.js"></script>
    <script src="js/jquery.validate.min.js"></script>
    <script src="js/jquery.validationEngine-pt_BR.js"></script>
    <script src="js/jquery.validationEngine.js"></script>
<script src="js/validate.revista.js"></script>

</body>
</html>

```

lista_revista.jsp :

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="">

<title>Sistema Revistas - lista de revista</title>

<!-- Bootstrap core CSS -->
<link href="css/bootstrap.css" rel="stylesheet">

<!-- Custom CSS -->
<link href="css/sb-admin.css" rel="stylesheet">
<link rel="stylesheet" href="font-awesome/css/font-awesome.min.css">
<link href="css/lightbox.css" rel="stylesheet">

</head>

<body>

<div id="wrapper">

<!-- Sidebar -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
<!-- Brand and toggle get grouped for better mobile display -->
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-
collapse">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="">Sistema Revistas</a>
</div>

<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse navbar-ex1-collapse">
<ul class="nav navbar-nav side-nav">
<li class="dropdown">
<a href="#" class="dropdown-toggle" id="menu-top" data-toggle="dropdown"><i class="fa fa-caret-
square-o-down"></i> Gerenciar revistas <b class="caret"></b></a>
<ul id="dropdown-menu-items" class="dropdown-menu">
<li><a href="http://localhost:8080/ES1Project/revista/listaRevista"><i class="fa fa-
table"></i> Exibir revistas cadastradas</a></li>
<li><a href="http://localhost:8080/ES1Project/revista/formularioRevista"><i class="fa fa-
edit"></i> Cadastrar nova revista</a></li>
</ul>
</li>
</ul>

<ul class="nav navbar-nav navbar-right navbar-user">
<li class="dropdown user-dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown"><i class="fa fa-user"></i>
${usuario.nome} <b class="caret"></b></a>
<ul class="dropdown-menu">
```

```

        <li><a href="http://localhost:8080/ES1Project/logout"><i class="fa fa-power-off"></i> Log
Out</a></li>
    </ul>
</li>
</ul>
</div><!-- /.navbar-collapse -->
</nav>

<div id="page-wrapper">

    <div class="row">
        <div class="col-lg-12">
            <h1>Lista de Revistas</h1>
        </div>
    </div><!-- /.row -->

    <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemErro'}">
        <div class="errormsgbox">${mensagem.texto}</div>
    </c:if>
    <c:if test="${mensagem != null && mensagem.tipo eq 'mensagemSucesso'}">
        <div class="successbox">${mensagem.texto}</div>
    </c:if>

    <div class="col-lg-6">
        <div class="table-responsive">

            <c:if test="${revistas == null || fn:length(revistas) eq 0}">
                <h4 class="no_magazines">Não há revistas cadastradas.</h4>
            </c:if>

            <c:if test="${revistas != null && fn:length(revistas) gt 0}">

                <table class="table table-hover table-striped tablesorter">
                    <thead>
                        <tr>
                            <th>id <i class="fa fa-sort"></i></th>
                            <th>Nome <i class="fa fa-sort"></i></th>
                            <th>Tema <i class="fa fa-sort"></i></th>
                            <th>Descrição <i class="fa fa-sort"></i></th>
                            <th>Formato Digital <i class="fa fa-sort"></i></th>
                            <th>Preço De <i class="fa fa-sort"></i></th>
                            <th>Preço Por <i class="fa fa-sort"></i></th>
                            <th>Gerenciar</th>
                        </tr>
                    </thead>
                    <tbody>
                        <c:forEach items="${revistas}" var="revista" varStatus="status">
                            <tr>
                                <td>${revista.id}</td>
                                <td>${revista.nome}</td>
                                <td>${revista.tema}</td>
                                <td>${revista.descricao}</td>
                                <c:if test="${revista.temDigital == true}">
                                    <td>Sim</td>
                                </c:if>
                                <c:if test="${revista.temDigital != true}">
                                    <td>Não</td>
                                </c:if>
                                <td>${revista.precoDeFormatado}</td>
                                <td>${revista.precoPorFormatado}</td>
                                <td><a class="manage-magazine"
href="http://localhost:8080/ES1Project/revista/editaRevista?id=${revista.id}">editar</a>
                                <a href="javascript:void(0)" class="manage-magazine" onclick =
"summonLightBox('http://localhost:8080/ES1Project/revista/deletaRevista?id=${revista.id}')">excluir</a></td>
                                </tr>
                        </c:forEach>
                    </tbody>
                </table>
            </c:if>
        </div>
    </div>

```

```

                </table>

            </c:if>
        </div>
    </div>

</div><!-- /#page-wrapper -->

</div><!-- /#wrapper -->

<!-- JavaScript -->
<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.js"></script>
<script src="js/showhidemenu.js"></script>

<!-- Page Specific Plugins -->
<script src="js/tablesorter/jquery.tablesorter.js"></script>
<script src="js/tablesorter/tables.js"></script>
<script src="js/lightbox.js"></script>

        <div id="light" class="white_content hidden"></div>
        <div id="fade" class="black_overlay hidden"></div>
    </body>
</html>

```

Javascripts e plugins utilizados no front-end:

Plugins utilizados:

- jquery-1.10.2.js
- bootstrap.js
- jquery.tablesorter.js
- tables.js
- validate.min.js
- validationEngine-pt_BR.js
- jquery.validationEngine.js
- jquery.mask.min.js

JavaScripts escritos por nós usando jQuery:

lightbox.js :

```
$(function() {  
    $(document).on('click', "#fade", function() {  
        killLightBoxWithFire();  
    });  
});  
  
function summonLightBox(url) {  
    var content = '<p class="delete_message">Você tem certeza que deseja excluir a revista?</p><br/><a  
class="select_lightbox_option" id="optionInLightBox" href="'+url+'" >Sim</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a  
id="optionInLightBox" href="javascript:void(0)" onclick = "killLightBoxWithFire()"  
class="select_lightbox_option" >Não</a>';  
    $('#light').removeClass("hidden").html(content);  
    $('#fade').removeClass("hidden");  
}  
  
function killLightBoxWithFire() {  
    $('#light').addClass("hidden");  
    $('#fade').addClass("hidden");  
}
```

showhidemenu.js :

```
$(document).ready(function() {  
    $("#menu-top").click(function() {  
        if($("#dropdown-menu-items").is(':hidden')) {  
            $("#dropdown-menu-items").show(130);  
        } else {  
            $("#dropdown-menu-items").hide(130);  
        }  
    });  
})
```

validate.revista.js :

```
$(function() {
    var doubleTypeInputs = new Array();

    var form = $(".form");

    var priceFrom = $(".precoDe");
    var priceFor = $(".precoPor");

    doubleTypeInputs.push(priceFor);
    doubleTypeInputs.push(priceFrom);

    // Apply Double Type masks
    $.each(doubleTypeInputs, function(index, data) {
        data.mask('09999.00');
    });

    $.validator.addMethod("greaterThan",

    function (value, element, param) {
        var $min = $(param);
        if (this.settings.onfocusout) {
            $min.off(".validate-greaterThan").on("blur.validate-greaterThan", function () {
                $(element).valid();
            });
        }

        return parseInt(value) > parseInt($min.val());
    }, "Preço de deve ser maior que preço por.");

    form.validate({
        submitHandler : function(form) {
            form.submit();
        },
        rules : {
            "nome" : {
                required : true
            },
            "tema" : {
                required : true
            },
            "precoDe" : {
                required : true,
                number: true,
                greaterThan: '.precoPor'
            },
            "precoPor" : {
                required : true,
                number: true
            }
        },
        messages : {
            "nome" : {
                required : "Este campo é obrigatório.",
            },
            "tema" : {
                required : "Este campo é obrigatório.",
            },
            "precoDe" : {
                required : "Este campo é obrigatório.",
                number: "Este campo deve ser um número."
            },
            "precoPor" : {
```



```
        required : "Este campo é obrigatório.",  
        number: "Este campo deve ser um número."  
    }  
});  
});
```

Arquivo web.xml com as declarações dos Servlets e Filter:

Web.xml :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
metadata-complete="true" version="3.0">
  <display-name>ES1 Sistema Revistas</display-name>

  <servlet>
    <servlet-name>FormularioRevistaServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.FormularioRevistaServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>FormularioRevistaServlet</servlet-name>
    <url-pattern>/revista/formularioRevista</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>ListaRevistaServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.ListaRevistaServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>ListaRevistaServlet</servlet-name>
    <url-pattern>/revista/listaRevista</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>EditaRevistaServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.EditaRevistaServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>EditaRevistaServlet</servlet-name>
    <url-pattern>/revista/editaRevista</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>DeletaRevistaServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.DeletaRevistaServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>DeletaRevistaServlet</servlet-name>
    <url-pattern>/revista/deletaRevista</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>LoginServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.LoginServlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
```

```

    <url-pattern>/login</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>LogoutServlet</servlet-name>
    <servlet-class>com.sistema.revistas.controller.LogoutServlet</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>LogoutServlet</servlet-name>
    <url-pattern>/logout</url-pattern>
</servlet-mapping>

<filter>
    <filter-name>LoginFilter</filter-name>
    <filter-class>com.sistema.revistas.controller.filtro.LoginFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>LoginFilter</filter-name>
    <url-pattern>/revista/*</url-pattern>
</filter-mapping>

<error-page>
    <error-code>404</error-code>
    <location>/erro_pagina_nao_encontrada.jsp</location>
</error-page>
</web-app>

```

Arquivo de configuração de conexão com o banco de dados:

Config.properties :

```
dataSource.driver=com.mysql.jdbc.Driver
dataSource.url=jdbc:mysql://localhost/sistemarevistas
dataSource.username=root
dataSource.password=root
dataSource.dialect=org.hibernate.dialect.MySQLDialect
dataSource.strategy=update
dataSource.hibernate.temp.use_jdbc_metadata_defaults=false
```

Interface homem-máquina - telas:

erro_pagina_nao_encontrada.jsp:



formulario_revista.jsp :

A screenshot of a web browser window displaying a form titled 'Cadastrar Revista' with the subtitle 'Entre com os dados'. The browser's address bar shows 'localhost:8080/ES1Project/revista/formulario_revista.jsp'. The form contains the following fields: 'Nome da revista: *' (text input), 'Tema: *' (text input), 'Descrição da revista:' (text input), 'Preço Por: *' (text input), and 'Preço De: *' (text input). Below these is a radio button group for 'Tem formato digital: *' with options 'Sim' and 'Não'. At the bottom of the form are two buttons: 'Cadastrar revista' and 'Resetar formulário'. The browser's title bar reads 'Sistema Revistas - cadastrar revista - Google Chrome'. On the left, a sidebar shows a menu with 'Sistema Revistas' and 'Gerenciar revistas'. The top right of the browser window shows the user is logged in as 'Administrador'.

formulario_revista.jsp – dados inválidos no cadastro:

Sistema Revistas - cadastrar revista - Google Chrome

localhost:8080/E51Project/revista/formulario_revista.jsp

Sistema Revistas

Administrador

Cadastrar Revista

Entre com os dados

Nome da revista: * Este campo é obrigatório.

Tema: * Este campo é obrigatório.

Descrição da revista:

Preço Por: * Este campo é obrigatório.

Preço De: * Este campo é obrigatório.

Tem formato digital: * ☒ Sim ☐ Não

Cadastrar revista Resetar formulário

formulario_revista.jsp – dados cadastro com sucesso:

Sistema Revistas - cadastrar revista - Google Chrome

localhost:8080/E51Project/revista/formulario_revista.jsp

Sistema Revistas

Administrador

Cadastrar Revista

Entre com os dados

Revista cadastrada com sucesso.

Nome da revista: *

Tema: *

Descrição da revista:

Preço Por: *

Preço De: *

Tem formato digital: * ☒ Sim ☐ Não

Cadastrar revista Resetar formulário

editar_revista.jsp:

Sistema Revistas - editar revista - Google Chrome

localhost:8080/ES1Project/revista/editar_revista.jsp

Sistema Revistas

Gerenciar revistas

Editar Revista

Entre com os dados

Nome da revista: *

Revista 1

Tema: *

Informática

Descrição da revista:

Revista sobre informática

Preço Por: *

11.00

Preço De: *

12.00

Tem formato digital: * ☒ Sim ☐ Não

Editar revista

[voltar para lista de revistas](#)

editar_revista.jsp – dados inválidos a edição:

Sistema Revistas - editar revista - Google Chrome

localhost:8080/ES1Project/revista/editar_revista.jsp

Sistema Revistas

Gerenciar revistas

Editar Revista

Entre com os dados

Nome da revista: *

Revista 1

Tema: *

Este campo é obrigatório.

Descrição da revista:

Revista sobre informática

Preço Por: *

13.00

Preço De: *

12.00

Preço de deve ser maior que preço por.

Tem formato digital: * ☒ Sim ☐ Não

Editar revista

[voltar para lista de revistas](#)

editar_revista.jsp – edição realizada com sucesso:

The screenshot shows a web browser window titled 'Sistema Revistas - editar revista - Google Chrome'. The address bar shows 'localhost:8080/E51Project/revista/editar_revista.jsp'. The page has a dark sidebar on the left with a menu containing 'Gerenciar revistas'. The main content area is titled 'Editar Revista' with a subtitle 'Entre com os dados'. A green message box at the top says 'Revista editada com sucesso.'. Below this are several form fields: 'Nome da revista:' with the value 'Revista 1', 'Tema:' with 'Informática', 'Descrição da revista:' with 'Revista sobre informática', 'Preço Por:' with '11.00', and 'Preço De:' with '12.00'. There is a radio button for 'Tem formato digital:' with 'Sim' selected. At the bottom, there is an 'Editar revista' button and a link 'voltar para lista de revistas'.

lista_revista.jsp – lightbox com confirmação de exclusão de uma revista:

The screenshot shows a web browser window titled 'Sistema Revistas - lista de revista - Google Chrome'. The address bar shows 'localhost:8080/E51Project/revista/lista_revista.jsp'. The page has a dark sidebar on the left with a menu containing 'Gerenciar revistas'. The main content area is titled 'Lista de Revistas'. It contains a table with 8 columns: 'Id', 'Nome', 'Tema', 'Descrição', 'Formato Digital', 'Preço De', 'Preço Por', and 'Gerenciar'. A lightbox dialog is open in the center of the screen with the text 'Você tem certeza que deseja excluir a revista?' and two buttons: 'Sim' and 'Não'.

Id	Nome	Tema	Descrição	Formato Digital	Preço De	Preço Por	Gerenciar
7	Revista 1	Informática	Revista sobre informática	Sim	R\$ 12,00	R\$ 11,00	editar excluir
8	Revista 2	Biologia			R\$ 10,00	R\$ 95,00	editar excluir
9	Revista 3	Informática			R\$ 10,00	R\$ 83,00	editar excluir
10	Revista 3	Química			R\$ 10,00	R\$ 76,00	editar excluir
17	Revista 7	Jogos			R\$ 10,00	R\$ 78,00	editar excluir
19	Revista 8	Biologia			R\$ 10,00	R\$ 15,00	editar excluir

lista_revista.jsp revista excluída com sucesso:

Sistema Revistas - lista de revista - Google Chrome

localhost:8080/E51Project/revista/lista_revista.jsp

Sistema Revistas

Administrador

Lista de Revistas

Revista excluída com sucesso.

id	Nome	Tema	Descrição	Formato Digital	Preço De	Preço Por	Gerenciar
7	Revista 1	Informática	Revista sobre informática	Sim	R\$ 12,00	R\$ 11,00	editar excluir
8	Revista 2	Biologia	Revista sobre biologia	Não	R\$ 99,00	R\$ 85,00	editar excluir
9	Revista 3	Informática	Revista sobre informática	Sim	R\$ 86,00	R\$ 83,00	editar excluir
10	Revista 3	Química	Revista sobre química	Não	R\$ 78,00	R\$ 76,00	editar excluir
17	Revista 7	Jogos	Revista sobre jogos.	Não	R\$ 80,00	R\$ 78,00	editar excluir

lista_revista.jsp:

Sistema Revistas - lista de revista - Google Chrome

localhost:8080/E51Project/revista/lista_revista.jsp

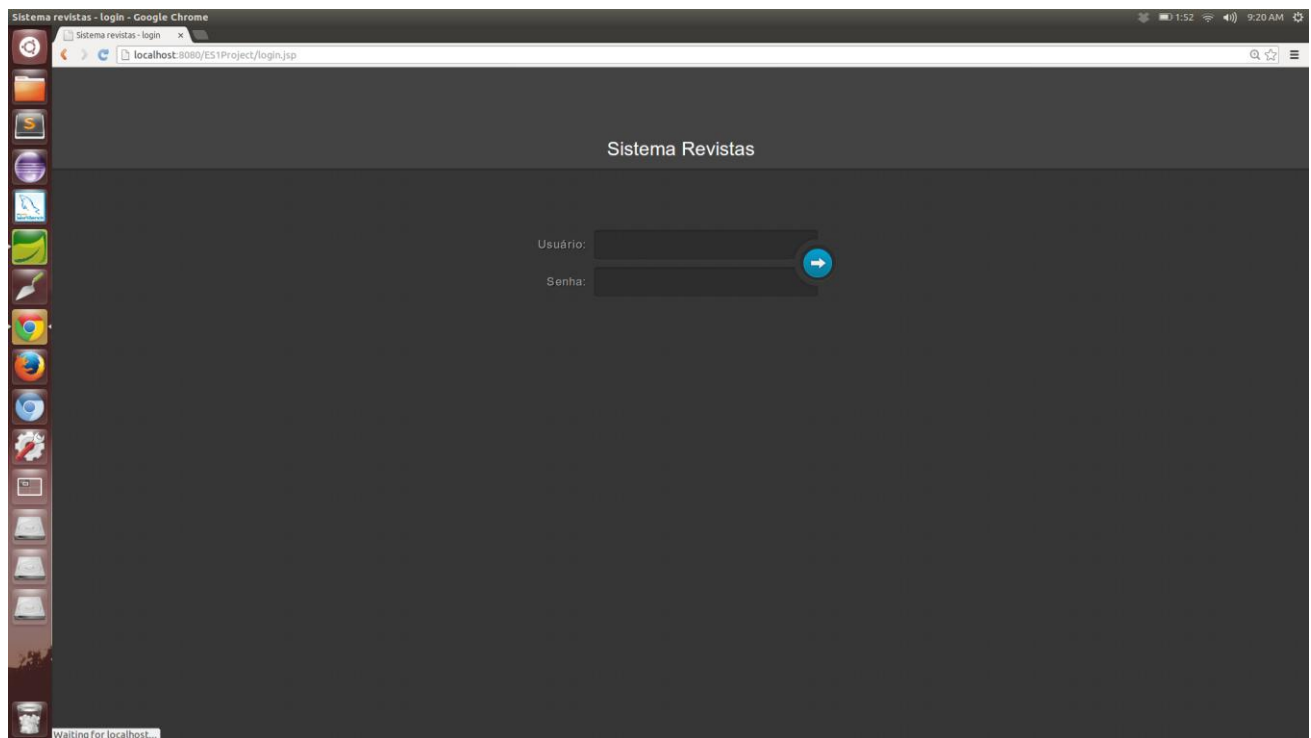
Sistema Revistas

Administrador

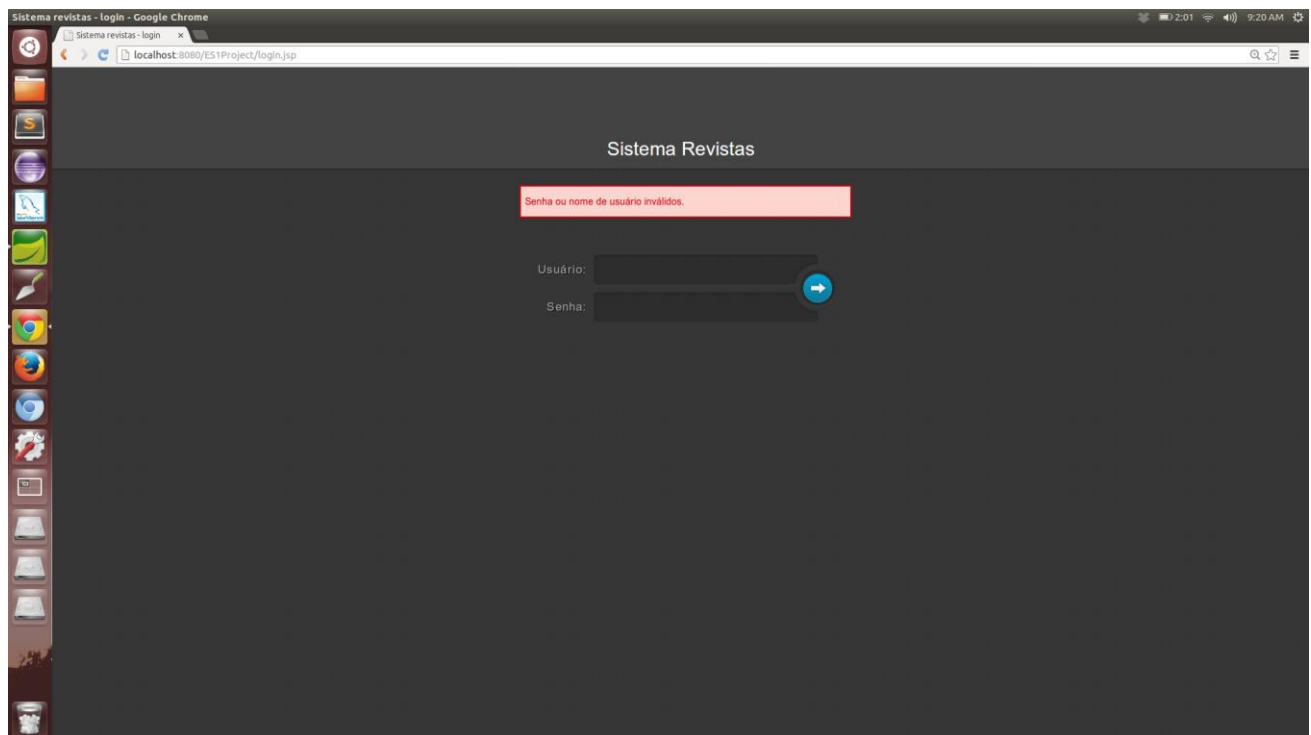
Lista de Revistas

id	Nome	Tema	Descrição	Formato Digital	Preço De	Preço Por	Gerenciar
7	Revista 1	Informática	Revista sobre informática	Sim	R\$ 12,00	R\$ 11,00	editar excluir
8	Revista 2	Biologia	Revista sobre biologia	Não	R\$ 99,00	R\$ 85,00	editar excluir
9	Revista 3	Informática	Revista sobre informática	Sim	R\$ 86,00	R\$ 83,00	editar excluir
10	Revista 3	Química	Revista sobre química	Não	R\$ 78,00	R\$ 76,00	editar excluir
17	Revista 7	Jogos	Revista sobre jogos.	Não	R\$ 80,00	R\$ 78,00	editar excluir
18	Revista 7	Informática	Revista sobre informática.	Não	R\$ 50,00	R\$ 45,00	editar excluir

login.jsp:



login.jsp – nome de usuário ou senha inválidos:



Cliente(id,nome,idade,cpf,telefone,endereco)

Assinatura(id,período,end_entrega,id_revista(FK_revista_id))

fazAssinatura(id_cliente(FK_Cliente_id),idassinatura(FK_Assinatura_id))

Revista(id,nome,tipo,descrição,precoPor,precoDe,id_editor(FK_Editora_id),id_gerente(FK_Gerente_codGerente))

Tema(id,nome,descrição)

PertenceRevistaTema(id_revista(FK_Revista_id),idtema(FK_Tema_id))

Editora(id,nome,endereço)

Redator(id,nome,idade,nacionalidade,id_especialidade(FK_Especialidade_id))

Especialidade(id,descrição)

Escreve(id_revista(FK_Revista_id),idredator(FK_Redator_id))

Pagamento(id,preco)

PagamentoViaBoleto(idpagamento(FK_Pagamento_id),numeroBoleto)

PagamentoViaCartao(idpagamento(FK_Pagamento_id),numeroCartao)

contemPagamentoAssinatura(idpagamento(FK_Pagamento_id),idAssinatura(FK_Assinatura_id))

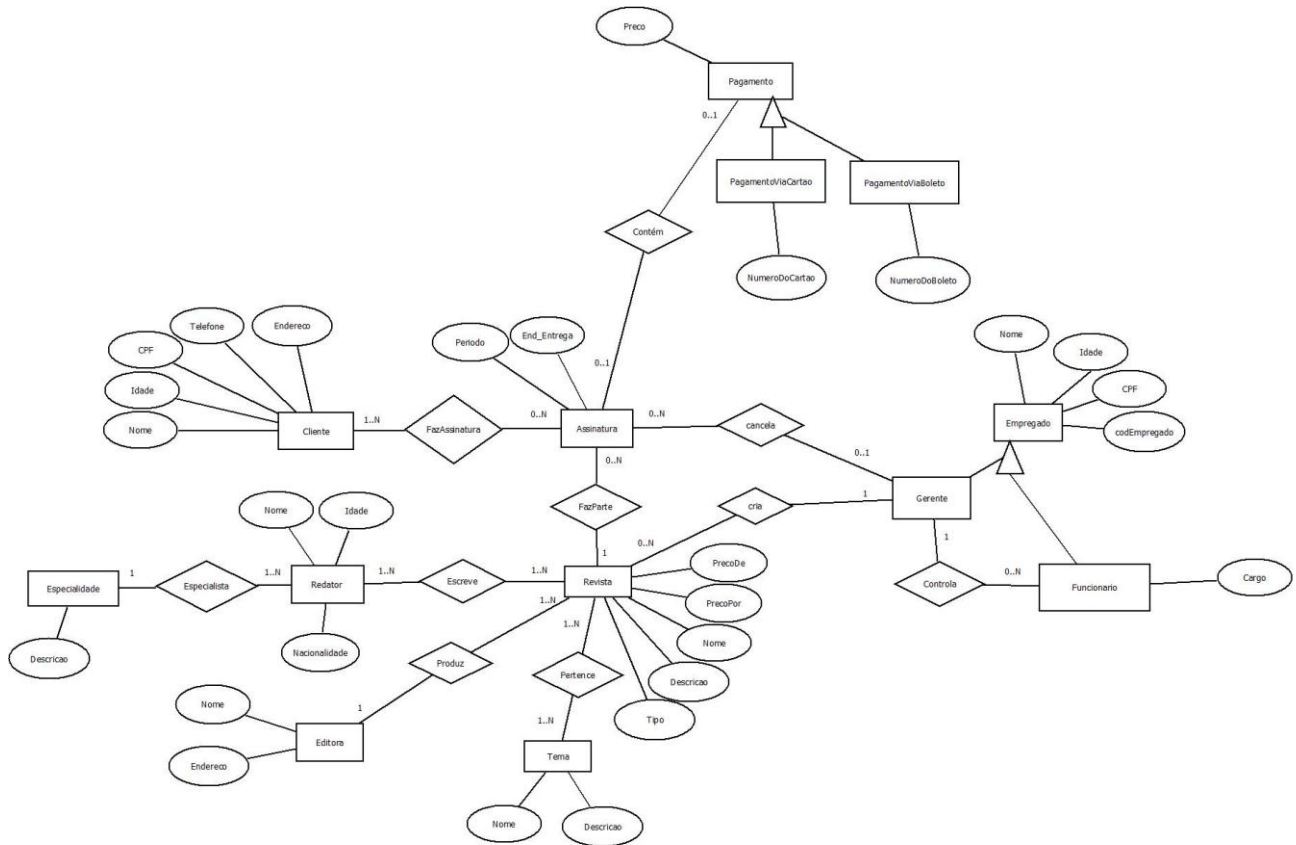
CancelaGerenteAssinatura(idgerente(FK_Gerente_id),idassinatura(FK_Assinatura_id))

Empregado(codEmpregado,nome,idade,cpf)

Gerente(codGerente,codEmpregado(FK_Empregado_id))

Funcionario(codEmpregado(FK_Empregado_codEmpregado),Cargo,codGerente(FK_Gerente_codGerente))

Diagrama ER:



Scripts SQL do banco:

```
Create or replace table cliente(  
    id:integer,  
    nome:varchar2(35) not null,  
    cpf: char(11) not null,  
    telefone: char(11) not null,  
    endereço: varchar2(35) not null,  
    constraint PK_cliente primary key (id)  
);
```

```
create or replace table assinatura(  
    id: integer,  
    periodo: integer not null,  
    ende_entrega: varchar2(35) not null,  
    id_revista:integer not null,  
    constraint PK_assinatura primary key (id),  
    constraint FK_revista_id FOREIGN KEY (id_revista) REFERENCES revista(id)  
);
```

```
create or replace table fazAssinatura(  
    id_revista: integer not null,  
    id_assinatura: integer not null,  
    constraint PK_fazAssinatura primary key (id_revista,id_assinatura),  
    constraint FK_revista_id FOREIGN KEY (id_revista) REFERENCES revista(id),  
    constraint FK_assinatura_id FOREIGN KEY (id_assinatura) REFERENCES assinatura(id)  
);
```

```
create or replace table revista(  
    id:integer,  
    nome:varchar2(35) not null,  
    descricao:varchar2(120) not null,  
    precoPor:float,  
    precoDe:float not null,  
    id_editor:integer not null,  
    id_gerente:integer not null,  
    constraint PK_revista primary key (id),  
    constraint FK_editora_id FOREIGN KEY (id_editor) REFERENCES editora(id),  
    constraint FK_gerente_id FOREIGN KEY (id_gerente) REFERENCES gerente(codGerente)  
);
```

```
create or replace table tema(  
    id: integer,  
    nome:varchar2(35) not null,  
    descricao:varchar2(35) not null,  
    constraint PK_tema primary key (id)  
);
```

```
create or replace table PertenceRevistaTema(  
    id_revista:integer,
```

```

        id_tema:integer,
        constraint PK_pertenceRevistaTema primary key (id_tema,id_revista),
        constraint FK_revista_id FOREIGN KEY (id_revista) REFERENCES revista(id),
        constraint FK_tema_id FOREIGN KEY (id_tema) REFERENCES tema(id)
    );

create or replace table editora(
    id:integer,
    nome:varchar2(35) not null,
    endereco:varchar2(35) not null,
    constraint PK_editora primary key (id)
);

create or replace table redator(
    id:integer,
    nome: varchar2(35) not null,
    nacionalidade: varchar2(35) not null,
    id_especialidade: integer not null,
    constraint PK_redator primary key (id),
    constraint FK_especialidade_id FOREIGN KEY (id_especialidade) REFERENCES especialidade(id)
);

create or replace table especialidade(
    id:integer,
    descricao:varchar2(35),
    constraint PK_especialidade primary key (id)
);

create or replace table escreve(
    id_revista:integer,
    id_redator:integer,
    constraint PK_escreve primary key (id_redator,id_revista),
    constraint FK_revista_id FOREIGN KEY (id_revista) REFERENCES revista(id),
    constraint FK_redator_id FOREIGN KEY (id_redator) REFERENCES redator(id)
);

create or replace table pagamento(
    id: integer,
    preco: float,
    constraint PK_pagamento primary key (id)
);

create or replace table pagamentoViaBoleto(
    id_pagamento:integer,
    numeroBoleto:integer,
    constraint PK_pagamentoViaBoleto primary key (id_pagamento,numeroBoleto),
    constraint FK_pagamento_id foreign key (id_pagamento) references pagamento(id)
);

create or replace table pagamentoViaCartao(
    id_pagamento:integer,

```

```

        numeroCartao:integer,
        constraint PK_pagamentoViaCartao primary key (id_pagamento,numeroCartao),
        constraint FK_pagamento_id foreign key (id_pagamento) references pagamento(id)
    );

create or replace table contemPagamentoAssinatura(
    id_pagamento:integer,
    id_assinatura:integer,
    constraint PK_contemPagamentoAssinatura primary key (id_pagamento,id_assinatura),
    constraint FK_pagamento_id foreign key (id_pagamento) references pagamento(id),
    constraint FK_assinatura_id foreign key (id_assinatura) references assinatura(id)
);

create or replace table cancelaGerenteAssinatura(
    id_gerente:integer,
    id_assinatura:integer,
    constraint PK_cancelaGerenteAssinatura primary key (id_gerente,id_assinatura),
    constraint FK_gerente_id foreign key (id_gerente) references gerente(codGerente),
    constraint FK_assinatura_id foreign key (id_assinatura) references assinatura(id)
);

create or replace table empregado(
    codEmpregado:integer,
    nome:varchar2(35) not null,
    idade:integer not null,
    cpf:char(11) not null,
    constraint PK_empregado primary key (codEmpregado)
);

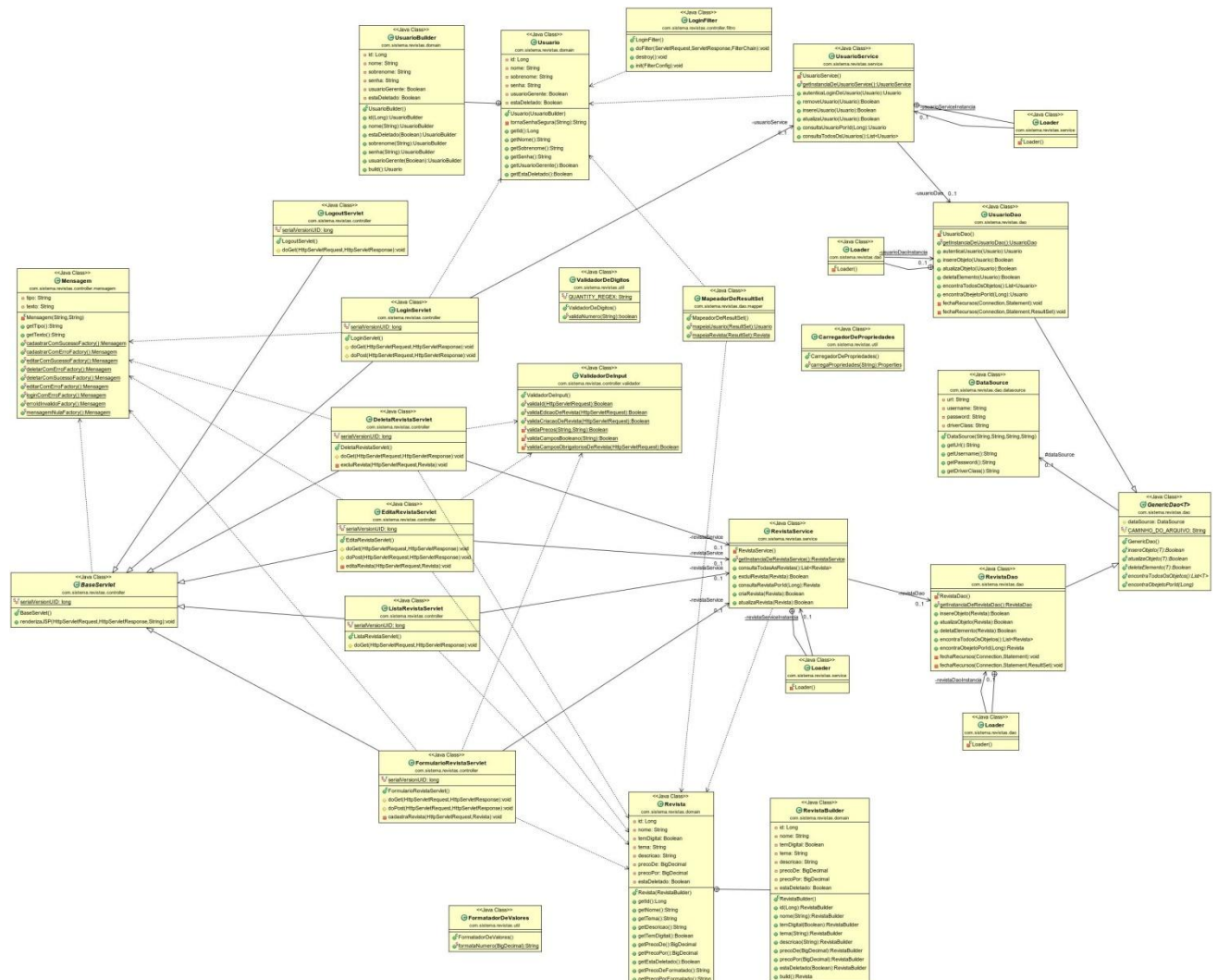
create or replace table gerente(
    codEmpregado:integer,
    codGerente:integer,
    constraint PK_gerente primary key (codEmpregado,codGerente),
    constraint FK_Empregado_id foreign key (codEmpregado) references empregado(codEmpregado)
);

create or replace table funcionario(
    codEmpregado:integer,
    codFuncionario:integer,
    cargo:varchar2(35) not null,
    id_gerente:integer not null,
    constraint PK_funcionario primary key (codEmpregado,codFuncionario),
    constraint FK_Empregado_id foreign key (codEmpregado) references empregado(codEmpregado),
    constraint FK_gerente_id foreign key (id_gerente) references gerente(codGerente)
);

```


Diagrama de Classes:

Gerado pelo Eclipse (STS):



Feito no Astah Professional:

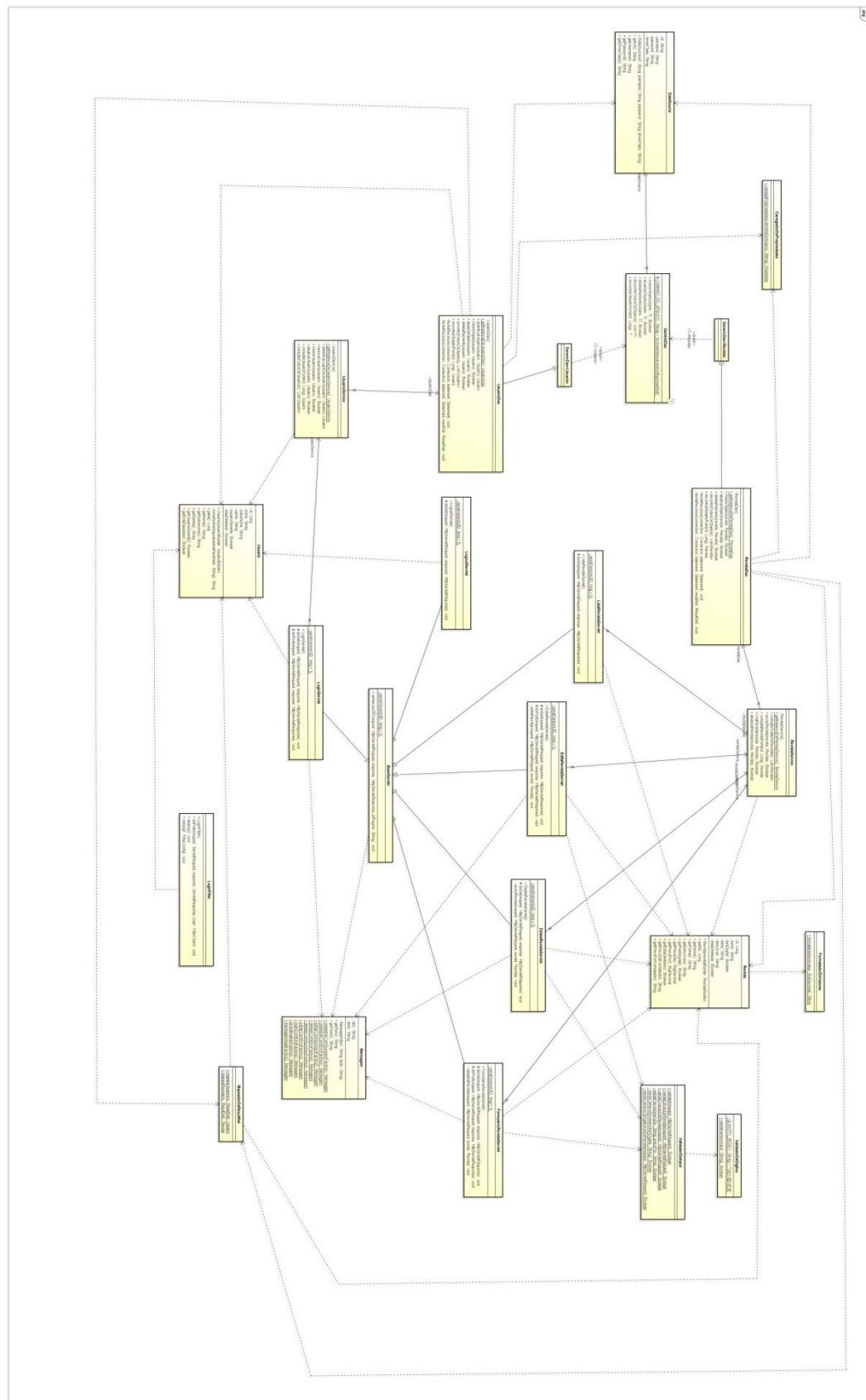
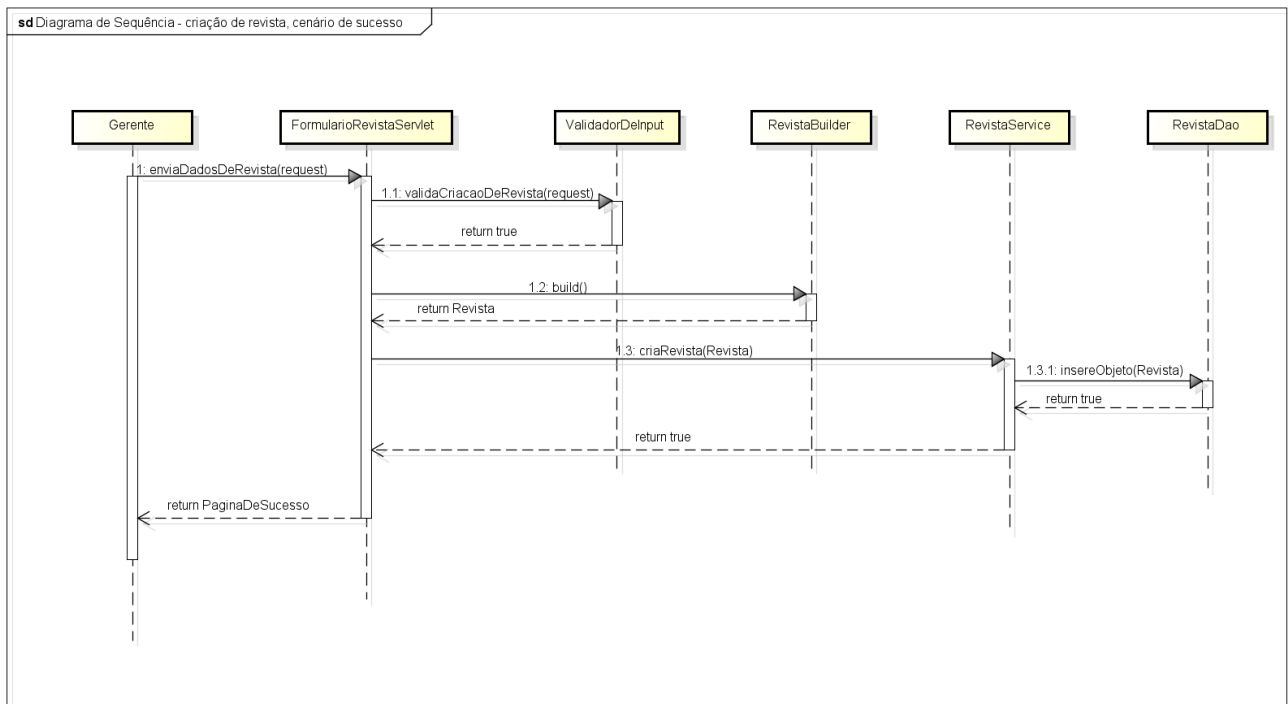


Diagrama de Sequência – Criação de Revista Cenário de Sucesso



powered by Astah

Mais informações sobre o código fonte em:

<https://github.com/LeonardoRamos/Trabalho-ES1>

Descrição do caso de uso real:

- Cenário principal:

- 1 - Gerente faz o login no sistema inserindo usuário e senha nos campos apresentados na primeira página do sistema.
- 2 - Sistema apresenta página lista de revistas mostrando uma lista com dados de cada revista e as opções de editar e excluir revista. Na página há o botão gerenciar revistas. Gerente clica em gerenciar revistas e seleciona a opção cadastrar revista.
- 3 – Sistema apresenta tela cadastrar revista.
- 4 - Gerente insere os dados da revista nos campos de cadastro da revista. Nome da Revista(String), Tema(String), Descrição da Revista(String), Preço por(Big Decimal), Preço de(Big Decimal), Tem formato digital (seleção).
- 5 – Gerente clica em cadastrar revista e o sistema gera uma mensagem de revista cadastrada com sucesso.

- Cenário alternativo 1: Gerente insere dados de usuário e senha errados

- 1 - Gerente faz o login no sistema inserindo usuário e senha nos campos apresentados na primeira página do sistema.
- 2- Ao validar dados de login, o sistema comunica um erro e apresenta uma mensagem de senha ou nome de usuário inválidos.
- 3 – Sistema se mantém na página de login.

- Cenário alternativo 2: Gerente deixa campos em branco ao cadastrar revista

- 1 - Gerente faz o login no sistema inserindo usuário e senha nos campos apresentados na primeira página do sistema.
- 2 - Sistema apresenta página lista de revistas mostrando uma lista com dados de cada revista e as opções de editar e excluir revista. Na página há o botão gerenciar revistas. Gerente clica em gerenciar revistas e seleciona a opção cadastrar revista.
- 3 – Sistema apresenta tela cadastrar revista.
- 4 - Gerente insere os dados da revista nos campos de cadastro da revista. Nome da Revista (String), Tema(String), Descrição da Revista(String), Preço por(Big Decimal), Preço de(Big Decimal), Tem formato digital (seleção).
- 5 – Ao validar os dados inseridos, o sistema comunica que todos os campos não foram preenchidos por meio da mensagem de campo obrigatório ao lado de cada campo não preenchido.

6 – Sistema se mantém na página de cadastro de revista.