

Sistema Triar

Realizando seleções de maneira simples

Sumário

1. Introdução

2. Camada de Acesso a Dados

2.1 Modelo ER

2.2 Package Model

2.2 Package DAO

3. Camada de Apresentação

3.1 Package Controller

3.2 Package Converter

3.3 Template do Site

3.4 Navegação

3.4 Telas

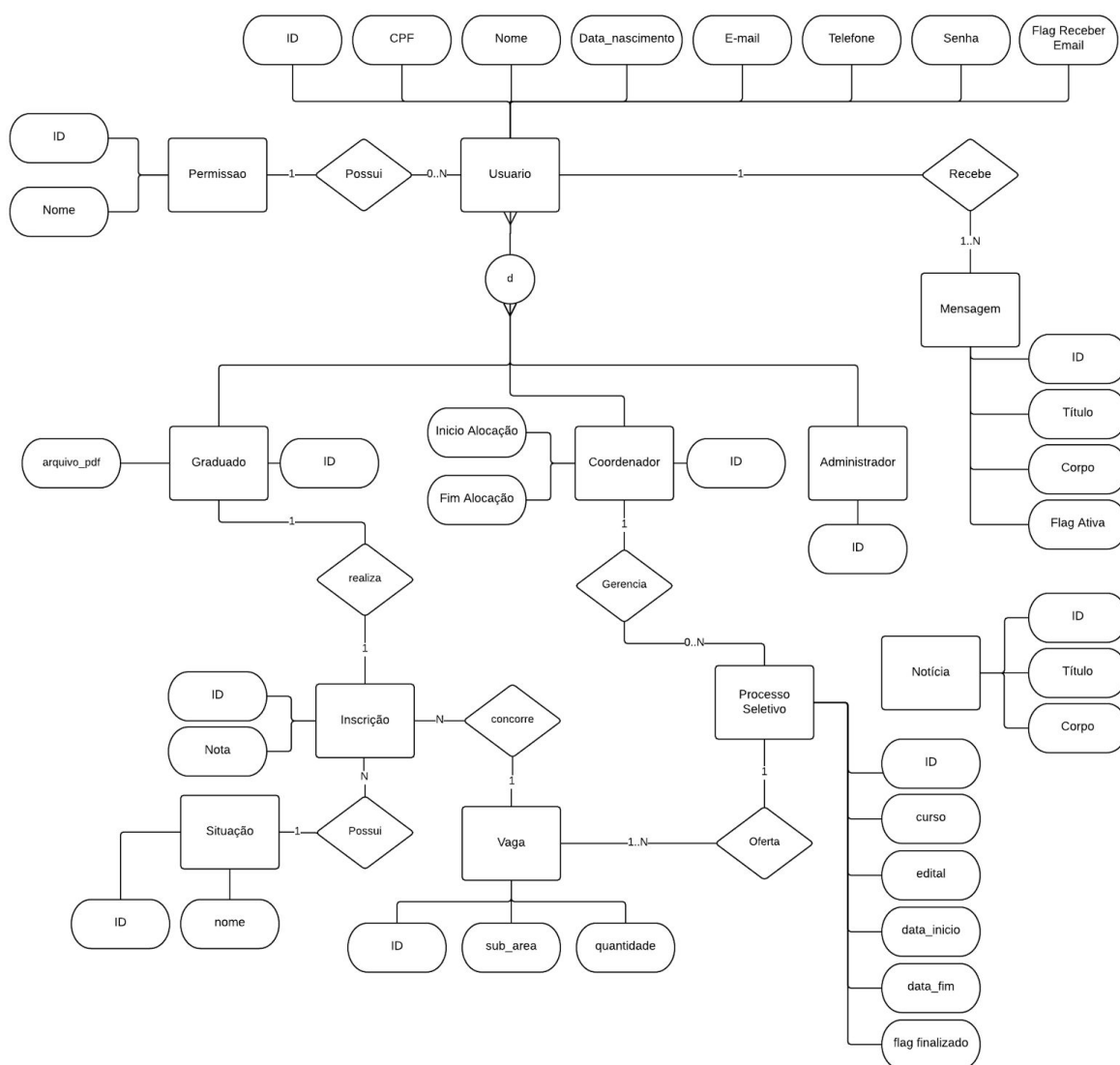
1. Introdução

Esse documento registra o desenvolvimento da camada de aplicação utilizando o framework JSF e a camada de acesso aos dados que foi implementada utilizando o framework JPA (realizando um mapeamento objeto-relacional) juntamente com o banco de dados MySQL.

O projeto está no github: <https://github.com/rodrigondec/triar>

2. Camada de Acesso a Dados

2.1 Modelo ER



2.2 Package Model

Neste pacote estão as classes que implementam o modelo no padrão MVC do sistema. São utilizadas pelo JPA para implementar as entidades no banco de dados e suas relações.

2.2.1 Administrador.java

```
public class Administrador {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idadministrador;  
  
    @OneToOne  
    @JoinColumn(name="idusuario")  
    private Usuario usuario;
```

2.2.2 Coordenador.java

```
public class Coordenador {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idcoordenador;  
    @Temporal(TemporalType.DATE)  
    @Column(nullable=false)  
    private Date inicio_alocacao;  
    @Temporal(TemporalType.DATE)  
    @Column(nullable=false)  
    private Date fim_alocacao;  
  
    @OneToOne  
    @JoinColumn(name="idusuario")  
    private Usuario usuario;  
  
    @OneToMany  
    @JoinColumn(name="idcoordenador")  
    private List<ProcessoSeletivo> processos;
```

2.2.3 Graduado.java

```
public class Graduado {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idgraduado;  
    private File identidade;  
  
    @OneToOne  
    @JoinColumn(name="idusuario")  
    private Usuario usuario;  
  
    @OneToMany  
    @JoinColumn(name="idgraduado")  
    private List<Inscricao> inscricoes;
```

2.2.4 Inscricao.java

```
public class Inscricao {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idinscricao;  
    private int nota;  
  
    @ManyToOne  
    @JoinColumn(name="idgraduado")  
    private Graduado graduado;  
  
    @ManyToOne  
    @JoinColumn(name="idvaga")  
    private Vaga vaga;  
  
    @ManyToOne  
    @JoinColumn(name="idsituacao")  
    private Situacao situacao;
```

2.2.5 Mensagem.java

```
public class Mensagem {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idmensagem;  
    @Column(nullable=false)  
    private String titulo;  
    @Column(nullable=false)  
    private String corpo;  
    @Column(nullable=false)  
    private boolean ativa;  
  
    @ManyToOne  
    @JoinColumn(name="idusuario")  
    private Usuario usuario;
```

2.2.6 Noticia.java

```
public class Noticia {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idnoticia;  
    @Column(nullable=false)  
    private String titulo;  
    @Column(nullable=false)  
    private String corpo;
```

2.2.7 Permissao.java

```
public class Permissao {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int idpermissao;
    @Column(nullable=false, unique=true)
    private String nome;

    @OneToMany
    @JoinColumn(name="idusuario")
    private List<Usuario> usuarios;
```

2.2.8 ProcessoSeletivo.java

```
public class ProcessoSeletivo {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int idprocesso;
    @Column(nullable=false)
    private String curso;
    // @Column(nullable=false)
    private File edital;
    @Temporal(TemporalType.DATE)
    @Column(nullable=false)
    private Date data_inicio;
    @Temporal(TemporalType.DATE)
    @Column(nullable=false)
    private Date data_fim;
    @Column(nullable=false)
    boolean finalizado = false;

    @ManyToOne
    @JoinColumn(name="idcoordenador")
    private Coordenador coordenador;

    @OneToMany
    @JoinColumn(name="idvaga")
    private List<Vaga> vagas;
```

2.2.9 Situacao.java

```
public class Situacao {
    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private int idsituacao;
    @Column(nullable=false, unique=true)
    private String nome;

    @OneToMany
    @JoinColumn(name="idinscricao")
    private List<Inscricao> inscricoes;
```

2.2.10 Usuario.java

```
public class Usuario {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idusuario;  
    @Column(nullable=false, unique=true)  
    private String cpf;  
    @Column(nullable=false)  
    private String nome;  
    @Temporal(TemporalType.DATE)  
    @Column(nullable=false)  
    private Date data_nascimento;  
    @Column(nullable=false, unique=true)  
    private String email;  
    private String telefone;  
    @Column(nullable=false)  
    private String senha;  
    @Column(nullable=false)  
    private Boolean receber_email = true;  
  
    @ManyToOne  
    @JoinColumn(name="idpermissao")  
    private Permissao permissao;  
  
    @OneToMany  
    @JoinColumn(name="idmensagem")  
    private List<Mensagem> mensagens;
```

2.2.11 Vaga.java

```
public class Vaga {  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int idvaga;  
    @Column(nullable=false)  
    private String sub_area;  
    private int quantidade;  
  
    @ManyToOne  
    @JoinColumn(name="idprocesso")  
    private ProcessoSeletivo processo;  
  
    @OneToMany  
    @JoinColumn(name="idinscricao")  
    private List<Inscricao> inscricoes;
```


2.2 Package DAO

Neste pacote estão as classes que realizam as consultas no banco de dados. Todas as classes desse pacote possuem métodos para criar, atualizar e remover registros nas tabelas/entidades no banco.

2.2.1 AdministradorDAO.java

Métodos

```
public Administrador buscarAdministrador(int idusuario){
```

Método que retorna um administrador que possua o idusuario passado. Query:
"select a from Administrador a where a.usuario.idusuario= :idusuario"

```
public List<Administrador> listarAdministradores() {
```

Método que retorna uma lista de administradores. Query:
"select a from Administrador a"

2.2.2 CoordenadorDAO.java

Métodos

```
public Coordenador buscarCoordenador(int idusuario){
```

Método que retorna um coordenador que possua o idusuario passado. Query:
"select c from Coordenador c where c.usuario.idusuario= :idusuario"

```
public List<ProcessoSeletivo> listarProcessos(int idcoordenador){
```

Método que retorna uma lista de processos seletivos que sejam gerenciados pelo coordenador passado. Query:
"select p from ProcessoSeletivo p where p.coordenador.idcoordenador= :idcoordenador"

2.2.3 GraduadoDAO.java

Métodos

```
public Graduado buscarGraduado(int idusuario){
```

Método que retorna um graduado que possua o idusuario passado. Query:
"select g from Graduado g where g.usuario.idusuario= :idusuario"

```
public List<ProcessoSeletivo> listarProcessos(int idgraduado){
```

Método que retorna uma lista de processos seletivos em que o graduado passado esteja inscrito. Query:

```
"select curso, data_inicio, data_fim, edital from processo_seletivo natural join vaga natural join inscricao natural join graduado where idgraduado = :idgraduado"
```

2.2.4 InscricaoDAO.java

Métodos

```
public List<Inscricao> listarInscricoes() {
```

Método que retorna uma lista de inscrições. Query:

```
"select i from Inscricao i"
```

2.2.5 MensagemDAO.java

Métodos

```
public List<Mensagem> listarMensagens() {
```

Método que retorna uma lista de mensagens. Query:

```
"select m from Mensagem m"
```

2.2.6 NoticiaDAO.java

Métodos

```
public List<Noticia> listarNoticias() {
```

Método que retorna uma lista de notícias. Query:

```
"select n from Noticia n"
```

2.2.7 PermissaoDAO.java

Métodos

```
public String buscarNome(int idpermissao){
```

Método que retorna o nome da permissão que possua o idpermissao passado, utilizando o método find() do EntityManager.

```
public List<Permissao> listarPermissoes() {
```

Método que retorna uma lista de permissões. Query:
"select p from Permissao p"

2.2.8 ProcessoSeletivoDAO.java

Métodos

```
public List<ProcessoSeletivo> listarProcessos(){
```

Método que retorna uma lista de processos seletivos. Query:
"select p from ProcessoSeletivo p"

```
public ProcessoSeletivo buscarIdprocesso(int idprocesso) {
```

Método que retorna o processo seletivo que possua o idprocesso passado, utilizando o método find() do EntityManager.

2.2.9 SituacaoDAO.java

Métodos

```
public List<Situacao> listarSituacoes() {
```

Método que retorna uma lista de situações. Query:
"select s from Situacao s"

2.2.10 UsuarioDAO.java

Métodos

```
public Usuario buscarIdusuario(int idusuario){
```

Método que retorna o usuário que possua o idusuario passado, utilizando o método find() do EntityManager.

```
public Usuario buscarEmail(String email){
```

Método que retorna um usuário que possua o email passado. Query:
"select u from Usuario u where u.email= :email"

```
public List<Mensagem> listarMensagens(int idusuario) {
```

Método que retorna uma lista de mensagens que sejam do usuário passado. Query:
"select m from Mensagem m where m.usuario.idusuario= :idusuario"

```
public boolean temNotificacao(int idusuario){
```

Método que retorna um bool dizendo se existe alguma mensagem não lida do usuário passado. Query:

```
"select m from mensagem m where m.ativa = 1 and m.usuario.idusuario= :idusuario"
```

```
public List<Usuario> listarUsuarios() {
```

Método que retorna uma lista de usuários. Query:

```
"select u from Usuario u"
```

2.2.11 VagaDAO.java

Métodos

3. Camada de Apresentação

3.1 Package Controller

Neste pacote estão as classes de controle no padrão MVC implementado no sistema. Todas as classes desse pacote possuem métodos de get e set para seus atributos (exceto para os atributos DAO, que possuem notação @Inject).

3.1.1 AdministradorMB.java

Atributos

```
private Administrador administrador;  
  
private AdministradorDAO administradorDAO;  
  
private List<Administrador> administradores;  
    Lista de todos os administradores do sistema
```

Métodos

```
public String getUrlCadastrar(){  
    Retorna a string com o caminho da página de cadastro  
  
public String getUrlListar(){  
    Retorna a string com o caminho da página de listagem  
  
public String cadastrar(){  
    Método de cadastro
```

3.1.2 CoordenadorMB.java

Atributos

```
private Coordenador coordenador  
  
private CoordenadorDAO coordenadorDAO  
  
private List<ProcessoSeletivo> processos  
    Lista de processos seletivos gerenciados pelo coordenador
```

Métodos

```
public String validar(){
```

Método que irá criar/atualizar um coordenador de acordo com o caso de uso *Validar Coordenador*.

```
public String geturlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String geturlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.3 GraduadoMB.java

Atributos

```
private Graduado graduado
```

```
private GraduadoDAO graduadoDAO
```

Métodos

```
public String geturlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String geturlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.4 InscricaoMB.java

Atributos

```
private Inscricao inscricao
```

Atributo de “persistência” da inscrição

```
private InscricaoDAO inscricaoDAO
```

Atributo de acesso aos dados da inscrição

```
private List<Inscricao> inscricoes
```

Lista de todas as inscrições cadastradas

Métodos

```
public String getUrlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.5 MensagemMB.java

Atributos

Métodos

```
public String getUrlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String getUrlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.6 NoticiaMB.java

Atributos

```
private Noticia noticia
```

Atributo de “persistência” da notícia

```
private NoticiaDAO noticiaDAO
```

Atributo de acesso aos dados da notícia

Métodos

```
public String geturlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String geturlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.7 PermissaoMB.java

Atributos

Métodos

```
public String geturlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String geturlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.8 ProcessoSeletivoMB.java

Atributos

```
private ProcessoSeletivo processo
```

```
private ProcessoSeletivoDAO processoDAO
```

```
private List<ProcessoSeletivo> processos
```

Lista de todos os processos seletivos do sistema

Métodos

```
public String finalizar(){
```

Método para finalizar um processo seletivo


```
public String geturlListar(){  
    Retorna a string com o caminho da página de listagem  
  
public String cadastrar(){  
    Método de cadastro
```

3.1.9 SituacaoMB.java

Atributos

```
private Situacao situacao  
    Atributo de “persistência” da situação  
  
private SituacaoDAO situacaoDAO  
    Atributo de acesso aos dados da situação  
  
private List<Situacao> situacoes;  
    Lista de todas as situações do sistema
```

Métodos

```
public String geturlCadastrar(){  
    Retorna a string com o caminho da página de cadastro  
  
public String geturlListar(){  
    Retorna a string com o caminho da página de listagem  
  
public String cadastrar(){  
    Método de cadastro
```

3.1.10 UsuarioMB.java

Atributos

```
private Usuario usuario  
    Atributo de “persistência” do usuário  
  
private UsuarioDAO usuarioDAO  
    Atributo de acesso aos dados do usuário
```

```
private List<Mensagem> listaMensagens
```

Lista de todas as mensagens de um determinado usuário

```
private List<Usuario> usuarios;
```

Lista de todos os usuários do sistema

Métodos

```
public String getMenu(){
```

Método que retorna o nome do arquivo do menu específico para um determinado tipo de usuário

```
public String getHome(){
```

Método que retorna a url da página inicial para um determinado tipo de usuário

```
public String login(){
```

Método que realiza o login no sistema

```
public String getUrlCadastrar(){
```

Retorna a string com o caminho da página de cadastro

```
public String getUrlListar(){
```

Retorna a string com o caminho da página de listagem

```
public String cadastrar(){
```

Método de cadastro

3.1.11 VagaMB.java

Atributos

```
private Vaga vaga
```

Atributo de “persistência” da vaga

```
private VagaDAO vagaDAO
```

Atributo de acesso aos dados da vaga

Métodos

```
public String getUrlListar(){
```

Retorna a string com o caminho da página de listagem

3.2 Package Converter

Neste pacote estão as classes que implementam os FacesConverter necessários para o pacote Controller.

3.2.1 UsuarioCV.java

```
public class UsuarioCV implements Converter{
    @Inject
    private UsuarioDAO usuarioDAO;

    @Override
    public Object getAsObject(FacesContext context, UIComponent component, String idusuario) {
        int id = Integer.parseInt(idusuario);

        Usuario usuario = usuarioDAO.buscarIdusuario(id);
        System.out.println(usuario.getNome());
        return usuario;
    }

    @Override
    public String getAsString(FacesContext context, UIComponent component, Object usuario) {
        if (!(usuario instanceof Usuario)) {
            return null;
        }
        return String.valueOf(((Usuario) usuario).getIdusuario());
    }
}
```

3.2.2 ProcessoCV.java

```
public class ProcessoCV implements Converter{
    @Inject
    ProcessoSeletivoDAO processoDAO;

    @Override
    public Object getAsObject(FacesContext arg0, UIComponent arg1, String idprocesso) {
        int id = Integer.parseInt(idprocesso);

        ProcessoSeletivo processo = processoDAO.buscarIdprocesso(id);
        System.out.println(processo.getCurso());
        return processo;
    }

    @Override
    public String getAsString(FacesContext arg0, UIComponent arg1, Object processo) {
        if (!(processo instanceof ProcessoSeletivo)) {
            return null;
        }
        return String.valueOf(((ProcessoSeletivo) processo).getIdprocesso());
    }
}
```

3.3 Template do Site

Todas as páginas do Sistema Triar utilizam um template que possui o framework Bootstrap. O corpo do template é dividido entre menu, header, conteúdo e footer; sendo o menu e o footer arquivos prontos que são incluídos. Como demonstra a imagem abaixo:

```
<h:body>
  <div id="wrap">
    <ui:include src="#{usuarioMB.menu}" />

    <div class="fluid-container">
      <div class="row-fluid text-center">
        <ui:insert name="header">Título</ui:insert>
      </div>
      <div class="row-fluid">
        <ui:insert name="conteudo">Conteúdo</ui:insert>
      </div>
    </div>
  </div>

  <div id="footer">
    <div class="container text-center">
      <ui:include src="footer.xhtml" />
    </div>
  </div>

  <h:outputScript library="webjars" name="jquery/2.1.4/jquery.min.js"/>
  <h:outputScript library="webjars" name="bootstrap/3.3.6/js/bootstrap.min.js" />
</h:body>
```

3.4 Navegação

A navegação entre as páginas do sistema é feita através de uma barra de menu superior, e há um menu para cada tipo de usuário. Sendo esses o usuário administrador, coordenador, graduado e externo. O UsuarioMB.getMenu() é o que ‘decide’ qual dos menus incluir.

3.4 Telas

3.4.1 Administrador

Index.xhtml

Página que possui listagem de usuários e processos seletivos registrados

Páginas de cadastro

Uma coleção de páginas de cadastro para as classes de Modelo do sistema

Páginas de listagem

Uma coleção de páginas de listagem para as classes de Modelo do sistema

Validar_coordenador.xhtml

Página para validar um coordenador no sistema

3.4.2 Coordenador

Index.xhtml

Página que possui listagem dos processos seletivos gerenciados pelo coordenador

Finalizar_processo.xhtml

Página para finalizar um processo seletivo

Registrar_processo.xhtml

Página para cadastrar um novo processo seletivo

3.4.3 Graduado

Index.xhtml

Página que possui uma listagem de processo seletivos

Minhas_inscrições.xhtml

Página que possui uma listagem dos processos seletivos em que o graduado está inscrito

Registrar_inscrição.xhtml

Página para registrar uma inscrição em um processo seletivo

3.4.4 Externo

Index.xhtml

Página que possui uma listagem de processos seletivos