Sistema Triar

Realizando seleções de maneira simples

Sumário

1. Introdução

Esse documento registra o desenvolvimento da camada de negócios utilizando o framework EJB.

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

2. Camada de Negócios

2.1 Package Service

Neste pacote estão as classes que implementam as regras de negócio fazendo a mediação entre os controllers e as DAO's .

2.1.1 AdministradorService.java

Métodos

public List<Administrador> listarAdministradores()

Método para listar todos os administradores

public void cadastrarAdministrador(Administrador administrador)

Método para cadastrar um novo administrador

2.1.2 CoordenadorService.java

Métodos

public Coordenador getCoordenador(int idusuario)

Método que retorna um coordenador de acordo com o idusuario

public List<Coordenador> listarCoordenadores()

Método para listar todos os coordenadores

public void cadastrarCoordenador(Coordenador coordenador)

Método para cadastrar um novo coordenador

public List<ProcessoSeletivo> listarProcessos(int idcoordenador)

Método para listar os processos seletivos de um coordenador

2.1.3 GraduadoService.java

Métodos

public Graduado getGraduado(int idusuario)

Método que retorna um graduado de acordo com o idusuario

public void cadastrarGraduado(Graduado graduado)

Método para cadastrar um novo graduado

public List<Graduado> listarGraduados()

Método para listar todos os graduados

2.1.4 InscricaoService.java

Métodos

public List<Inscricao> listarInscricoes()

Método para listar todas as insrições

public List<Vaga> listarVagas(int idprocesso)

Método para listar todas as vagas de acordo com o idprocesso

public void cadastrarInscricao(Inscricao inscricao)

Método para cadastrar uma nova inscrição

public void atualizarInscricao(Inscricao inscricao)

Método para atualizar uma inscrição

public void removerInscricao(int idinscricao)

Método para remover uma inscrição

public List<Inscricao> listarInscricoesPorProcesso(ProcessoSeletivo processo)

Método para listar todas as inscrições de acordo com um processo

2.1.5 MensagemService.java

Métodos

public void cadastrarMensagem(Mensagem mensagem)

Método para cadastrar uma nova mensagem

public List<Mensagem> listarMensagens()

Método para listar todas as mensagens

public void atualizarMensagem(Mensagem m)

Método para atualizar uma mensagem

public void deletarMensagem(Mensagem m)

Método para remover uma mensagem

2.1.6 NoticiaService.java

Métodos

public List<Noticia> listarNoticias()

Método para listar todas as notícias

public void cadastrarNoticia(Noticia noticia)

Método para cadastrar uma nova notícia

public void atualizarNoticia(Noticia noticia)

Método para atualizar uma notícia

public void removerNoticia(int idnoticia)

Método para remover uma notícia

2.1.7 ProcessoSeletivoService.java

Métodos

public void cadastrarProcessoSeletivo(ProcessoSeletivo processo)

Método para cadastrar um novo processo seletivo

public void cadastrarVaga(Vaga vaga)

Método para cadastrar uma nova vaga

public List<ProcessoSeletivo> listarProcessos()

Método para listar todos os processos seletivos

public void atualizarProcesso(ProcessoSeletivo processo)

Método para atualizar um processo seletivo

public List<ProcessoSeletivo> listarProcessosPorGraduado(Graduado graduado)

Método para listar todos os processos de acordo com um graduado

2.1.8 UsuarioService.java

Métodos

public int login(String email, String senha)

Método para verificar se há um usuário com esse email e senha. Retorna o idusuario se existir

public Usuario getUsuario(int idusuario)

Método que retorna um graduado de acordo com o idusuario

public List<Mensagem> listarMensagens(int idusuario)

Método para listar todas as mensagens de acordo com idusuario

public boolean temNotificacao(int idusuario)

Método para verificar se o usuário tem uma mensagem não lida

public void cadastrarUsuario(Usuario usuario)

Método para cadastrar um novo usuário

public List<Usuario> listarUsuarios()

Método para listar todos os usuários

public void atualizarUsuario(Usuario u)

Método para atualizar um usuário

2.2 Package Session

Neste pacote está a classe que implementa o SessionContext do FacesContext ExternalContext 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 SessionContext.java

Métodos

public void encerrarSessao()

Método para encerrar a sessão

public Object getAttribute(String nome)

Método para pegar o atributo de acordo com o nome

public void setAttribute(String nome, Object valor)

Método para setar um atributo de acordo com o nome e valor

public void removeAttribute(String nome)

Método para remover um atributo

public boolean isUsuarioLogado()

Método para verificar se há um usuário logado

public Usuario getUsuarioLogado()

Método para retornar o usuário logado

public void setUsuarioLogado(Usuario usuario)

Método para setar o usuário logado

3. Camada de Apresentação

3.1 Package FIlter

Neste pacote estão as classes que implementam a interface javax.servlet.Filter.

3.1.1 LoginFilter

Esse filtro gerencia o acesso às páginas dentro da pasta “/interna/*”, liberando o acesso apenas se tiver um usuário logado

3.1.2 AdministradorFilter

Esse filtro gerencia o acesso às páginas dentro da pasta “/interna/administrador/*”, liberando o acesso apenas se o usuário logado for um administrador

3.1.3 CoordenadorFilter

Esse filtro gerencia o acesso às páginas dentro da pasta “/interna/coordenador/*”, liberando o acesso apenas se o usuário logado for um coordenador

3.1.4 GraduadoFilter

Esse filtro gerencia o acesso às páginas dentro da pasta “/interna/graduado/*”, liberando o acesso apenas se o usuário logado for um graduado

3.2 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 Service, que possuem notação @EJB).

3.2.1 AdministradorMB.java

Atributos

private Administrador administrador;

private AdministradorService administradorService;

private List<Administrador> administradores;

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

3.2.2 CadastrarInscricaoMB.java

Atributos

private Inscricao inscricao;

Private ProcessoSeletivo processo;

Private Vaga vaga;

Private List<Vaga> vagas;

private InscricaoService inscricaoService;

Métodos

public String geturlProcesso()

Retorna a string com o caminho da página de seleção do processo

public String geturlVaga()

Retorna a string com o caminho da página de seleção da vaga

public String armazenar()

Método para armazenar o processo selecionado

public String cadastrar()

Método de cadastro da inscrição para a vaga selecionada

3.2.3 CadastrarProcessoMB.java

Atributos

private ProcessoSeletivo processo;

Private Vaga vaga;

Private ProcessoSeletivoService processoService;

Private CoordenadorService coordenadorService;

Métodos

public String geturlProcesso()

Retorna a string com o caminho da página de registro do processo

public String geturlVaga()

Retorna a string com o caminho da página de registro da vaga

public String armazenar()

Método para armazenar o processo seletivo

public String cadastrar()

Método de cadastro o processo e vaga

3.2.4 ConfiguracaoMB.java

Atributos

private Usuario usuario;

private String senha1;

private String senha2;

UsuarioService usuarioService;

Métodos

public void alterarSenha()

Método para alterar senha do usuário

public void atualizarUsuario()

Método para atualizar dados do usuário

3.2.5 CoordenadorMB.java

Atributos

private Coordenador coordenador

private CoordenadorService coordenadorService

private List<ProcessoSeletivo> processos

Métodos

public String validar(){

Método que irá 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

3.2.6 FinlizarProcessoMB.java

Atributos

private ProcessoSeletivo processo;

private ProcessoSeletivoService processoService;

private CoordenadorService coordenadorService;

private InscricaoService inscricaoService;

private List<Inscricao> inscricoes;

Métodos

public String geturlNotas()

Método que retorna a url da página de registro de notas

public String geturlProcesso()

Método que retorna a url da página de seleção do processo seletivo

public String armazenar()

Método para armazenar o processo seletivo escolhido

public String finalizar()

Método para registrar as notas e situações das inscrições do processo seletivo

3.2.7 GraduadoMB.java

Atributos

private Graduado graduado

private List<ProcessoSeletivo> processos;

private GraduadoService graduadoService;

private ProcessoSeletivoService processoService;

private List<Graduado> graduados;

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

3.2.8 LoginMB.java

Atributos

private UsuarioService usuarioService;

private String email;

private String senha;

Métodos

public String doLogin()

Método que realiza o login no sistema

public String doLogout()

Método que realiza o logout no sistema

3.2.9 MensagemMB.java

Atributos

private Mensagem mensagem;

private MensagemService mensagemService;

private List<Mensagem> mensagens;

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

Public void ler()

Método para alterar a flag ‘não lida’ da mensagem para false

Public void deletar()

Método para deletar uma mensagem

3.2.10 NoticiaMB.java

Atributos

private Noticia noticia;

NoticiaService noticiaService;

private List<Noticia> noticias;

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

3.2.11 ProcessoSeletivoMB.java

Atributos

private ProcessoSeletivoService processoService;

private List<ProcessoSeletivo> processos;

Métodos

public String geturlListar(){

Retorna a string com o caminho da página de listagem

3.2.9 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

3.2.10 UsuarioMB.java

Atributos

private Usuario usuario;

UsuarioService usuarioService;

private List<Mensagem> mensagens;

private List<Usuario> usuarios;

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 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 getnotificacaoSimbolo()

Método que retorna o tipo do simbolo dependendo se o usuário tem notificação

public String cadastrar()

Método de cadastro