**EventoController - Controllers**

*@Controller*

public class EventoController {

*@Autowired*

private EventoRepository er;

*@Autowired*

private ConvidadoRepository cr;

*@RequestMapping*(value="/cadastrarEvento", method=*RequestMethod*.***GET***)

public String form(){

return "evento/formEvento";

}

*@RequestMapping*(value="/cadastrarEvento", method=*RequestMethod*.***POST***)

public String form(*@Valid* Evento evento, BindingResult result, RedirectAttributes attributes){

if(result.hasErrors()){

attributes.addFlashAttribute("mensagem", "Verifique os campos!");

return "redirect:/cadastrarEvento";

}

er.save(evento);

attributes.addFlashAttribute("mensagem", "Evento cadastrado com sucesso!");

return "redirect:/cadastrarEvento";

}

*@RequestMapping*("/eventos")

public ModelAndView listaEventos(){

ModelAndView mv = new ModelAndView("listaEventos");

Iterable<Evento> eventos = er.findAll();

mv.addObject("eventos", eventos);

return mv;

}

*@RequestMapping*(value="/{codigo}", method=*RequestMethod*.***GET***)

public ModelAndView detalhesEvento(*@PathVariable*("codigo") long codigo){

Evento evento = er.findByCodigo(codigo);

ModelAndView mv = new ModelAndView("evento/detalhesEvento");

mv.addObject("evento", evento);

Iterable<Convidado> convidados = cr.findByEvento(evento);

mv.addObject("convidados", convidados);

return mv;

}

*@RequestMapping*("/deletarEvento")

public String deletarEvento(long codigo){

Evento evento = er.findByCodigo(codigo);

er.delete(evento);

return "redirect:/eventos";

}

*@RequestMapping*(value="/{codigo}", method=*RequestMethod*.***POST***)

public String detalhesEventoPost(*@PathVariable*("codigo") long codigo, *@Valid* Convidado convidado, BindingResult result, RedirectAttributes attributes){

if(result.hasErrors()){

attributes.addFlashAttribute("mensagem", "Verifique os campos!");

return "redirect:/{codigo}";

}

Evento evento = er.findByCodigo(codigo);

convidado.setEvento(evento);

cr.save(convidado);

attributes.addFlashAttribute("mensagem", "Convidado adicionado com sucesso!");

return "redirect:/{codigo}";

}

*@RequestMapping*("/deletarConvidado")

public String deletarConvidado(String rg){

Convidado convidado = cr.findByRg(rg);

cr.delete(convidado);

Evento evento = convidado.getEvento();

long codigoLong = evento.getCodigo();

String codigo = "" + codigoLong;

return "redirect:/" + codigo;

}

}

**Convidado – Model**

*@Entity*

public class Convidado {

*@Id*

*@NotEmpty*

private String rg;

*@NotEmpty*

private String nomeConvidado;

*@ManyToOne*

private Evento evento;

public String getRg() {

return rg;

}

public void setRg(String rg) {

this.rg = rg;

}

public String getNomeConvidado() {

return nomeConvidado;

}

public void setNomeConvidado(String nomeConvidado) {

this.nomeConvidado = nomeConvidado;

}

public Evento getEvento() {

return evento;

}

public void setEvento(Evento evento) {

this.evento = evento;

}

}

**Role – Model**

*@Entity*

public class Role implements GrantedAuthority{

*@Id*

private String nomeRole;

*@ManyToMany*(mappedBy = "roles")

private List<Usuario> usuarios;

public String getNomeRole() {

return nomeRole;

}

public void setNomeRole(String nomeRole) {

this.nomeRole = nomeRole;

}

public List<Usuario> getUsuarios() {

return usuarios;

}

public void setUsuarios(List<Usuario> usuarios) {

this.usuarios = usuarios;

}

*@Override*

public String getAuthority() {

// **TODO** Auto-generated method stub

return this.nomeRole;

}

}

**Evento – Model**

*@Entity*

public class Evento implements Serializable{

private static final long ***serialVersionUID*** = 1L;

*@Id*

*@GeneratedValue*(strategy= *GenerationType*.***AUTO***)

private long codigo;

*@NotEmpty*

private String nome;

*@NotEmpty*

private String local;

*@NotEmpty*

private String data;

*@NotEmpty*

private String horario;

*@OneToMany*

private List<Convidado> convidados;

public long getCodigo() {

return codigo;

}

public void setCodigo(long codigo) {

this.codigo = codigo;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getLocal() {

return local;

}

public void setLocal(String local) {

this.local = local;

}

public String getData() {

return data;

}

public void setData(String data) {

this.data = data;

}

public String getHorario() {

return horario;

}

public void setHorario(String horario) {

this.horario = horario;

}

}

**Usuario - Model**

*@Entity*

public class Usuario implements UserDetails, Serializable{

private static final long ***serialVersionUID*** = 1L;

*@Id*

private String login;

private String nomeCompleto;

*@NotEmpty*

private String senha;

*@ManyToMany*

*@JoinTable*(

name = "usuarios\_roles",

joinColumns = *@JoinColumn*(

name = "usuario\_id", referencedColumnName = "login"),

inverseJoinColumns = *@JoinColumn*(

name = "role\_id", referencedColumnName = "nomeRole"))

private List<Role> roles;

public String getLogin() {

return login;

}

public void setLogin(String login) {

this.login = login;

}

public String getNomeCompleto() {

return nomeCompleto;

}

public void setNomeCompleto(String nomeCompleto) {

this.nomeCompleto = nomeCompleto;

}

public String getSenha() {

return senha;

}

public void setSenha(String senha) {

this.senha = senha;

}

public List<Role> getRoles() {

return roles;

}

public void setRoles(List<Role> roles) {

this.roles = roles;

}

*@Override*

public Collection<? extends GrantedAuthority> getAuthorities() {

// **TODO** Auto-generated method stub

return (Collection<? extends GrantedAuthority>) this.roles;

}

*@Override*

public String getPassword() {

// **TODO** Auto-generated method stub

return this.senha;

}

*@Override*

public String getUsername() {

// **TODO** Auto-generated method stub

return this.login;

}

*@Override*

public boolean isAccountNonExpired() {

// **TODO** Auto-generated method stub

return true;

}

*@Override*

public boolean isAccountNonLocked() {

// **TODO** Auto-generated method stub

return true;

}

*@Override*

public boolean isCredentialsNonExpired() {

// **TODO** Auto-generated method stub

return true;

}

*@Override*

public boolean isEnabled() {

// **TODO** Auto-generated method stub

return true;

}

}

**ConvidadoRepository**

public interface ConvidadoRepository extends CrudRepository<Convidado, String>{

Iterable<Convidado> findByEvento(Evento evento);

Convidado findByRg(String rg);

}

**UsuarioRepository**

public interface UsuarioRepository extends CrudRepository<Usuario, String>{

Usuario findByLogin(String login);

}

**EventoRepository**

public interface EventoRepository extends CrudRepository<Evento, String>{

Evento findByCodigo(long codigo);

}