

CLASSE MÉTODOS

1. Classe Condutor

Métodos:

conduzirEvento(Evento evento)

Exibe uma mensagem informando que o condutor está conduzindo um determinado evento.

removerEvento(Evento evento)

Exibe uma mensagem indicando que o condutor foi removido da participação em um evento.

2. Classe Espectador

Métodos:

cadastrarEvento()

Exibe uma mensagem indicando que o espectador foi cadastrado em um evento.

sairEvento()

Exibe uma mensagem indicando que o espectador foi retirado de um evento.

3. Classe Evento

Métodos:

mostrarInformacaoEvento()

Exibe informações completas sobre o evento, incluindo ID, nome, descrição e datas.

cadastrarEvento()

Exibe mensagem indicando que o evento foi cadastrado.

removerEvento()

Exibe mensagem indicando que o evento foi removido.

4. Classe EventoAdministrado

Método sobrescrito:

mostrarInformacaoEvento()

Exibe informações específicas de um evento administrado pelo usuário.

5. Classe EventoParticipado

Método sobrescrito:

mostrarInformacaoEvento()

Exibe informações específicas de um evento do qual o usuário participou.

6. Classe Gerente

Métodos:

`getEventosIds()`

Retorna uma lista contendo apenas os IDs dos eventos que o gerente administra.

`cadastrarEvento(Evento evento)`

Exibe mensagem informando que o gerente cadastrou um evento.

`removerEvento(Evento evento)`

Exibe mensagem informando que o gerente removeu um evento.

7. Classe PasswordResetToken

Métodos (Getters e Setters):

`getId() / setId()`

`getToken() / setToken()`

`getExpiration() / setExpiration()`

`getUsuarioId() / setUsuarioId()`

`getUsuarioTipo() / setUsuarioTipo()`

Métodos usados para acessar e modificar informações relacionadas ao token de recuperação de senha.

8. Classe Pessoa

Métodos:

`atualizarStatusAutomaticamente()`

Atualiza automaticamente o status do usuário com base no preenchimento dos campos obrigatórios.

`desativar()`

Define o status do usuário como INATIVO.

`isCampoVazio(String campo)` (método privado)

Verifica se um campo é nulo ou vazio.

9. Classe Staff

Métodos:

`atribuirEvento(Evento evento)`

Exibe mensagem informando que um evento foi atribuído ao membro da equipe (staff).

`removerEvento(Evento evento)`

Exibe mensagem indicando que o staff foi removido de um evento.

REPOSITORY MÉTODOS

1. CondutorRepository

- `findByEmail(String email)`

Busca um condutor pelo e-mail informado.

- `findByCpf(String cpf)`

Busca um condutor pelo CPF informado.

- `existsByEmailOrCpf(String email, String cpf)`

Verifica se já existe um condutor com o mesmo e-mail ou CPF.

2. EspectadorRepository

- `findByEmail(String email)`

Busca um espectador pelo e-mail informado.

- `findByCpf(String cpf)`

Busca um espectador pelo CPF informado.

- `existsByEmailOrCpf(String email, String cpf)`

Verifica se já existe um espectador com o e-mail ou CPF informado.

- `findByResetToken(String resetToken)`

Busca um espectador através do token de recuperação de senha.

3. EventoRepository

(Não possui métodos adicionais além dos herdados do JpaRepository)

Métodos herdados:

`save()`

`findById()`

`findAll()`

`deleteById()`

... entre outros métodos padrão do Spring Data.

4. GerenteRepository

- `findByEmail(String email)`

Busca um gerente pelo e-mail.

- `findByCpf(String cpf)`

Busca um gerente pelo CPF.

- `existsByEmailOrCpf(String email, String cpf)`

Verifica se já existe um gerente com e-mail ou CPF no sistema.

5. PasswordResetTokenRepository

- `findByToken(String token)`

Busca um token de redefinição de senha específico.

- `deleteByUsuarioidAndUsuarioTipo(Integer usuarioid, String usuarioTipo)`

Apaga todos os tokens relacionados a um usuário específico e do tipo indicado (Ex.: Espectador, Gerente etc.).

6. StaffRepository

- `findByEmail(String email)`

Busca um funcionário (staff) pelo e-mail.

- `existsByEmailOrCpf(String email, String cpf)`

Verifica se já existe um staff cadastrado com o e-mail ou CPF informado.