**Actores:**

* Utilizador não registado
* Utilizador registado
  + Organizador
  + Participante
  + Convidado

**Casos utilização:**

**Login:**

Actor: Utilizador não registado

Acção: Login

Prioridade: Alta

**Registo:**

Actor: Utilizador não registado

Accção: Registar utilizador

Prioridade: Alta

**Criar evento:**

Actor: Utilizador registado

Acção: Criar evento privado\público

Prioridade: Alta

**Adicionar participante evento:**

Actor: Utilizador registado

Acção: Adicionar o próprio à lista de participantes

Prioridade: Alta

**Remover participante evento:**

Actor: Utilizador participante

Acção: Remover o próprio à lista de participantes

Prioridade: Alta

**Convidar evento privado:**

Actor: Organizador evento

Acção: Convidar participante para evento privado

Prioridade: Média

**Convidar evento público:**

Actor: Utilizador registado

Acção: Convidar evento público

Prioridade: Baixa

**Lista utilizadores:**

Actor: Utilizador registado

Acção: Procurar eventos públicos

Prioridade: Alta

**Lista eventos públicos:**

Actor: Utilizador não registado

Acção: Procurar eventos públicos

Prioridade: Alta

**Lista eventos criados:**

Actor: Organizador

Acção: Procurar eventos criados

Prioridade: Alta

**Lista eventos privados:**

Actor: Utilizador convidado

Acção: Procurar eventos para os quais foi convidado

Prioridade: Média

**Resposta convite:**

Actor: Utilizador convidado

Acção: Aceitar/Rejeitar convite evento

Prioridade: Média

**Modelo entidade-associação:**

**Entidades:**

Utilizador(uid, nome, data nascimento(dia, mes, ano), email)

Evento (eid, tema, tipo, data ocorrência(dia, mes, ano), tempo(hora, minuto))

**Associações:**

Organiza (Utilizador, Evento ) 1:N p/t

Participa (Utilizador, Evento) N:N p/t

Convidado (Utilizador, Evento) N:N p/t

**Modelo relacional:**

Utilizador(uid, nome, diaNascimento, mesNascimento, anoNascimento, email)

Evento(eid, tema, tipo, diaOcorrencia, mesOcorrencia, anoOcorrencia, horaTempo, minutoTempo, #eid->Utilizador NN)

Participa(#uid->Utilizador NN, #eid->Evento NN, estado)

Convidado(#uid->Utilizador NN, #eid->Evento NN, estado)

**SQL:**

CREATE TABLE utilizador (uid integer primary key, nome varchar(255) not null, password varchar(255) not null, nascimento date, email varchar(255) not null);

CREATE TABLE evento(eid integer primary key, tema varchar(255) not null, descricao varchar(255), data date, tipo boolean, tempo time, uid references Utilizador);

CREATE TABLE participa(uid references Utilizador, eid references Evento, estado);

CREATE TABLE convidado(uid references Utilizador, eid references Evento, estado);

**Queries:**

**Login:**

SELECT password FROM utilizador WHERE nome””;

**Registo:**

INSERT INTO utilizador VALUES (NULL, nome, password, dataNascimento, email);

**Criar evento:**

INSERT INTO evento VALUES(NULL, tema, data, tipo, tempo, id organizador);

**Convidar evento:**

INSERT INTO convidado VALUES(id convidado, id evento, true);

**Adicionar participante:**

INSERT INTO participa VALUES(id convidado, id evento, true);

**Remover participante:**

UPDATE participa SET estado=false WHERE uid= and eid=;

**Lista eventos públicos:**

SELECT \* FROM evento WHERE dataOcorrencia > hoje AND tipo=false;

**Lista eventos privados para os quais foi convidado:**

SELECT \* FROM convidado WHERE uid="" and estado=false;

**Lista utilizadores:**

SELECT nome, dataNascimento, email FROM utilizador

**Lista eventos criados:**

SELECT \* FROM evento WHERE uid=;

**Aceitar convite = Adicionar participante**

**Rejeitar convite:**

UPDATE convidado SET estado=false WHERE uid= and eid=;

**Regras:**

CREATE RULE removeConvidado AS

ON INSERT INTO participa

WHERE tipo==true // Evento privado

DO (

UPDATE convidado SET estado=false

WHERE uid=participa.uid AND eid=participa.eid);