

Base de Dados

LEI

Projeto “ANL Eventos” - Natação

Ano lectivo 2021/2022



202100296_202100299_202100984_GestaoEventosNatacao_LuisDamas_5f11h

**Nome: Francisco Silva
Nome: Rui Barroso
Nome: Gonalo Vieira**

**Número: 202100984
Número:202100299
Número: 202100296**

**Turma: 5 Turno: 5F11h
Professor: Luís Damas**

Índice

• Sumário.....	3
• Definição do do problema.....	4
• Modelo Entidade-Relação.....	6
• Diagrama Entidade-Relação (DE).....	11
• Modelo Relacional.....	12
• Scripts - Anexos.....	15
• Conclusão.....	16

Sumário

No âmbito da disciplina de Base de Dados, foi nos pedido para o desenvolvimento de uma base de dados responsável por gerir um evento desportivo, cujas modalidades ficaram ao nosso critério.

Nesta 1ª foi nos pedido para definir um Modelo Entidade-Relação que responde aos requisitos e necessidades do Enunciado e a partir desse Modelo construir um Diagrama Entidade-Relação

Neste relatório vamos definir o tema, o problema segundo os requisitos, apresentar a nossa proposta do Modelo Entidade-Relação, o nosso diagrama, e o nosso Modelo relacional. E por fim as funcionalidades da nossa base de dados.

Definição do do problema

A “ANL Eventos” necessita de um sistema para a gestão de eventos desportivos especializados que integre, tanto a gestão da componente desportiva, como a componente de alojamento dos participantes.
Eventos constituídos por variadas provas de diferentes modalidades de Natação, individuais e coletivas.

O sistema necessita de:

Permitir um indivíduo ser um atleta, um treinador e fazer parte da staff ao mesmo tempo. Um atleta não é obrigado a ter um treinador;

Gerir (inserir, atualizar, remover e consultar) a informação relacionada com os requisitos identificados;

Para cada evento terá de ser feito o registo da entidade específica que o organiza, período em que decorre, descrição do local, etc.;

Cada evento é constituído por um conjunto pré-determinado de provas de modalidades específicas que deve também ser persistido com a indicação de datas, horários, duração, local onde se realizam as atividades, resultados, etc.;

O registo de resultados de cada prova deve considerar também o critério de classificação (crescente: melhores classificados têm menores valores (ex.: tempo em atletismo), decrescente: melhores classificados têm maiores valores (ex.: peso em halterofilismo);

O registo de alojamentos deve permitir identificar características que os podem diferenciar a partir de uma lista pré-definida de valores. Por exemplo, deve ser possível indicar se possui microondas, aquecimento central, vista mar, terraço, varanda, etc. Sempre que se registre uma destas características deve ser armazenado o IP da máquina e a data e hora em que o valor foi selecionado;

Cada alojamento deve registar a sua localização (longitude e latitude) e ainda o distrito, concelho e freguesia em que se situa, de acordo com a seguinte lista normalizada 2;

Deverá existir informação sobre produtos e serviços que poderão ser fornecidos de acordo com o tipo de alojamento.

Para cada Evento deverá guardar a seguinte informação: quem o organiza; o local; a data; a duração; e as provas(uma ou mais);

Para cada Prova deverá guardar a seguinte informação: a modalidade; os participantes; as equipas caso seja uma modalidade coletiva; escalão; sexo; o resultado;

Para cada Modalidade deverá guardar a seguinte informação: nome; descrição; se é coletivo

Para cada Equipa deverá guardar a seguinte informação: o número, os membros e treinador(que podem pertencer a várias equipas); o nome da equipa(que tem de ser único); as provas em que participam;

Para cada Atleta deverá guardar a seguinte informação: o nome; a data de nascimento; a idade; o número identificação(único); telemóvel(único); peso; altura; escalão; sexo; alojamento;

Para cada Treinador deverá guardar a seguinte informação: o nome; a data de nascimento; a idade; o número identificação(único); telemóvel(único); título profissional; equipa; alojamento;

Para cada membro da Staff deverá guardar a seguinte informação: o nome; a data de nascimento; a idade; o número identificação(único); telemóvel(único); funções; provas;

Sobre o Alojamento deverá guardar a seguinte informação: a localização(o distrito, o concelho e a freguesia) do alojamento dos participantes; os produtos e serviços que poderão ser fornecidos de acordo com o tipo de alojamento e os detalhes dos alojamentos;

O sistema deve também guardar os comboios que tem parceria com os eventos e os respetivos bilhetes,;

Modelo Entidade-Relação

Entidades

- A entidade Indivíduo(Nome; Data de Nascimento; Idade(derivada); Número Identificação; Telemóvel(opcional))
Devido à repetição de atributos decidimos criar uma generalização que engloba 3 outras entidades.
Esta entidade contém a informação básica de todos os indivíduos que de alguma maneira participam na realização do evento, seja participação direta (ex:atleta) ou indireta (ex:staff).
Chave - Número Identificação.
- A entidade Atleta(Peso; Altura; Escalão(derivada); Sexo;)
Representa os indivíduos que participam em provas.
Esta entidade contém informação extra relevante à condição física do indivíduo.
Chave - Número Identificação.
- A entidade Treinador(Título Profissional)
São os representantes de uma equipa ou Atleta.
Esta entidade contém a informação do nível profissional do Treinador.
Chave - Número Identificação.
- A entidade Staff(Título Profissional; Número fiscal)
São os representantes das pessoas que trabalham para o funcionamento dos eventos(funcionários, organizadores, etc)
Esta entidade contém a informação do nível profissional e do seu NIF.
Chave - Número Identificação.
- A entidade Função(Nome, Descrição)
Contém as funções necessárias para o funcionamento de um evento.

- A entidade Equipa(Sigla; Nome; Número de Membros(Derivado))
Esta entidade contém a informação em relação às equipas.
Chave - Sigla.
- A entidade Competição(ID;Número de Participantes(Derivado), Hora da competição)
Esta entidade representa uma competição de uma prova de um evento.
Chave - ID.
- A entidade Evento(Nome; Local; Data de Início; Data Final; Duração; Estado (concluído?))
Esta entidade contém a informação em relação a cada Evento.
Chave - Nome.
- A entidade Prova(ID;Se é coletiva ou individual; Escalão; Sexo Permitido(M - masculino, F - feminino, U - unisexo))
Esta entidade contém as informações sobre as provas e requisitos.
Chave - ID.
- A entidade Modalidade(Nome; Descrição;)
Esta entidade contém as Modalidades que estão presentes nas provas.
Chave - Nome.
- A entidade Alojamento(ID;Nº de Divisões; Nº de WC's;Cozinha partilhada; Area; Coordenadas; Morada)
Esta entidade contém a informação relacionada com os Alojamentos dos indivíduos que participam nos eventos.
Chave - ID.
- A entidade Item(Nome; Descrição; IP; Data e Hora)
Esta entidade representa os itens que podem estar presentes num alojamento.
Chave - IP.
- A entidade Serviço(Descrição; Produto)
Esta entidade contém todos os Serviços.
Chave - Produto.
- A entidade Fornecedor(Nome; Descrição)
Esta entidade contém todos os fornecedores.
Chave - Nome.

- A entidade Tipologia(Nome; Descrição)
Esta entidade contém as tipologias.
Chave - Nome.
- A entidade Distrito(ID; Nome;)
Esta entidade contém os Distritos que estão onde alguns participantes dos eventos.
Chave - ID.
- A entidade Conselho(ID; Nome;)
Esta entidade contém os Conselhos que estão onde alguns participantes dos eventos.
Chave - ID.
- A entidade Freguesia(ID; Nome;)
Esta entidade contém as freguesias onde estão alguns participantes dos eventos.
Chave - ID.
- A entidade Comboio(ID; Tipo; Preço; Lugares)
Esta entidade contém os comboios aos quais um evento tem parceria.
Chave - ID.
- A entidade Viagem(Bilhete; Preço; email;, Feedback; Avaliação; data/hora de reservar; data/hora de viagem; caminho)
Esta entidade contém as deslocações ,que são compradas através do portal da CP-Comboios aos quais um evento tem parceria.
Chave - ID.
- A entidade Linha(ID, Caminho)
Esta entidade representam as linhas de comboio suportadas pelo sistema e contém as localidades por onde passam.
Chave - ID.

Relações

- Indivíduo - Atleta
1 Indivíduo pode ser 1 atleta
1 atleta tem obrigatoriamente de ser 1 Indivíduo
- Indivíduo - Treinador
1 Indivíduo pode ser 1 Treinador
1 Treinador tem obrigatoriamente de ser 1 Indivíduo
- Indivíduo - Staff
1 Indivíduo pode ser 1 Staff
1 Staff tem obrigatoriamente de ser 1 Indivíduo
- Indivíduo - Alojamento
1 indivíduo tem que ter 1 Alojamento
1 Alojamento pode ter vários indivíduos
- Staff - Evento - Função
1 Staff pode ter várias funções em vários Eventos
1 Evento pode ter vários funcionários(Staff) e várias funções
1 Função pode ter vários funcionários em vários eventos
- Staff - Evento
1 Staff pode organizar vários Eventos
1 Evento tem de ser organizado por 1 Staff (organizador)
- Staff - Staff
1 Staff é o chefe de vários staff
1 Staff pode ter 1 Staff(Chefe)
- Treinador - Equipa
1 Treinador pode ter várias Equipas
1 Equipa tem que ter 1 Treinador
- Atleta - Equipa
1 Atleta pode participar em várias Equipa
1 Equipa pode ter vários atletas
- Atleta - Competição
1 Atleta pode participar em várias competições
1 competição pode ter vários atletas
Esta relação contém os seguintes atributos (Classificação(Composto): Prêmio;
Lugar;).

- Equipa - Competição
 - 1 Equipa pode participar em várias competições
 - 1 competição pode ter vários Equipas
 - Esta relação contém os seguintes atributos (Classificação(Composto): Prémio; Lugar;).
- Competição - Prova
 - 1 Competição tem de conter 1 prova
 - 1 Prova pode conter várias competições
- Competição - Evento
 - 1 Evento pode conter várias competições
 - 1 Competição pertence a 1 Evento
- Prova - Modalidade
 - 1 Prova tem que ter pelo menos 1 Modalidade
 - 1 Modalidade pode ter várias Provas
- Evento - Freguesia
 - 1 Evento é organizado em uma Freguesia
 - 1 Freguesia pode ter vários eventos
- Alojamento - Freguesia
 - 1 Alojamento tem de ter 1 Freguesia
 - 1 Freguesia pode vários Alojamento
- Alojamento - Serviço
 - 1 Alojamento pode ter vários Serviços
 - 1 Serviço pode ter 1 ou mais Alojamentos
- Alojamento - Item
 - 1 Alojamento pode ter vários Itens
 - 1 Item pode ter vários Alojamentos
- Concelho - Freguesia
 - 1 Conselho pode ter vários Freguesia
 - 1 Freguesia tem 1 Conselho
- Distrito - Concelho
 - 1 Distrito pode ter vários Concelho
 - 1 Concelho tem de ter 1 Distrito
- Serviço - Fornecedor - Tipologia
 - 1 Serviço pode ser fornecido em vários Tipologias por vários Fornecedores
 - 1 Fornecedor pode fornecer vários Serviços por Tipologia
 - 1 Tipologia pode ter vários Serviços de vários Fornecedores
 - Esta relação contém o seguinte atributo (Preço)

- Linha - Comboio
 - 1 linha pode ter vários comboios
 - 1 comboio tem que ter só 1 linha

- Comboio - Evento (parceria)
 - 1 Comboio pode ter vários eventos
 - 1 Evento pode ter vários comboios

- Viagem - Atleta - Comboio
 - 1 Viagem tem de ter 1 atleta e vários comboios
 - 1 Atleta pode ter várias viagens e comboios
 - 1 Comboio pode ter várias viagens e atletas
 - Esta relação contém o seguinte atributo (lugar)

- viagem - evento
 - 1 viagem contém vários eventos
 - 1 evento contém várias viagens
 - Esta relação contém o seguinte atributo (quantidade)

- viagem - Atleta- evento
 - 1 Viagem tem de ter 1 atleta e 1 evento
 - 1 Atleta pode ter várias viagens e vários eventos
 - 1 evento pode ter várias viagens e atletas

Diagrama Entidade - Relação

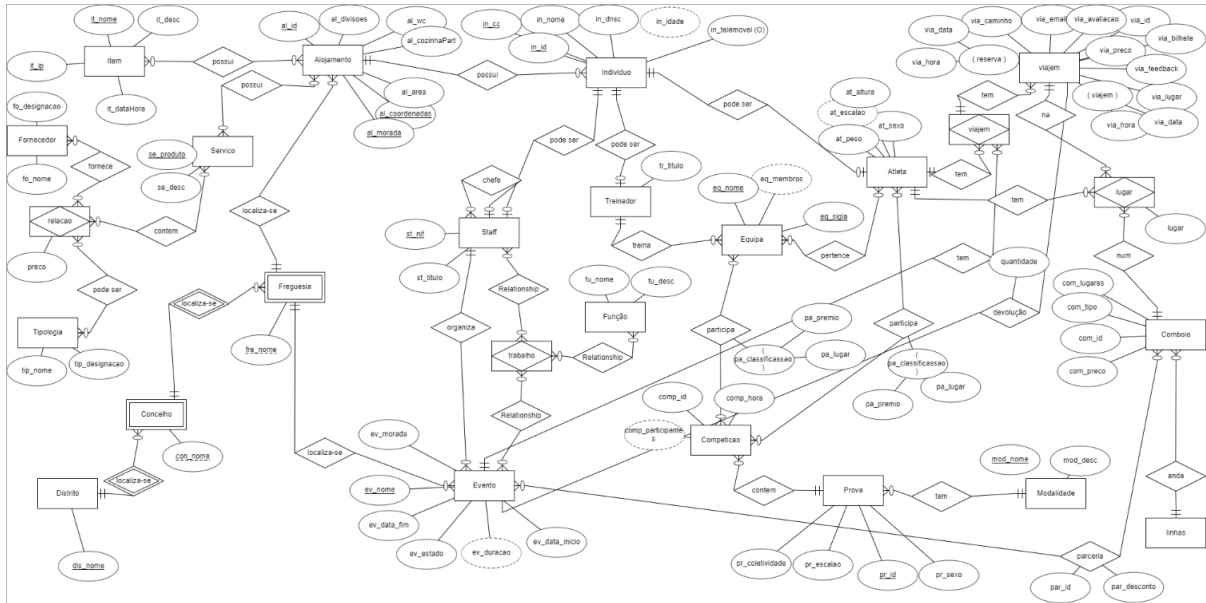


Fig. - MER

Modelo Relacional

- individuo (in_id int, in_cc varchar, in_nome varchar, in_dnsc date, in_idade int, in_telemovei char, al_id int)
FK al_id references alojamento(al_id)
- atleta(at_id int, at_peso double, at_altura double, at_escalao int, at_sexo char)
FK at_id references individuo(in_id)
- treinador(tr_id int, tr_titulo varchar)
FK tr_id references individuo(in_id)
- staff(st_id int, st_nif char, st_titulo varchar)
FK st_id references individuo(in_id)
- equipa(eq_sigla char; eq_nome varchar; eq_membros int, tr_id int)
FK tr_id references treinador (tr_id)
- equipa_membros(eq_sigla char, at_id int)
FK eq_sigla references equipa(eq_sigla)
FK at_id references atleta(at_id)
- competicao(comp_id int, comp_participantes int, comp_hora time, comp_evento varchar, comp_prova int)
FK comp_evento references evento(ev_nome)
FK comp_prova references prova(pr_id)
- participantes(comp_id int, at_id int, pa_lugar int, pa_premio varchar)
FK comp_id references competicao(comp_id)
FK at_id references atleta(at_id)
- equipas_participantes(comp_id int, eq_sigla varchar, pa_lugar int, pa_premio varchar)
FK comp_id references competicao(comp_id)
FK eq_sigla references equipa(eq_sigla)

- staff_rank(st_id int, st_chefe_id int)
FK st_id references staff(st_id)
FK st_chefe_id references staff(st_id)
- evento(ev_nome varchar, ev_local id, ev_data_inicio datetime, ev_data_fim datetime, ev_duracao double; ev_estado bool, ev_organizador varchar)
FK ev_local references freguesia(fre_id)
FK ev_organizador references staff(in_cc)
- prova(pr_id int, pr_coletividade bool, pr_escalao int, pr_sexo char, pr_modalidade varchar)
FK pr_modalidade references modalidade(mod_nome)
- modalidade(mod_nome varchar, mod_desc varchar)
- funcao(fu_nome varchar, fu_descricao varchar)
- trabalho(tr_evento varchar, tr_nome varchar, tr_st_id int)
FK tr_evento references evento(ev_nome)
FK tr_nome references funcao(fu_nome)
FK tr_st_id references staff(st_id)
- alojamento(al_id int, al_divisoas int, al_wc int, al_cozinhaPart bool, al_area double, al_coordenadas char, al_morada varchar, al_local varchar)
FK al_local references freguesia(fre_nome)
- item(it_nome varchar, it_desc varchar, it_ip char, it_dataHora datetime)
- alojamento_item(al_id int, it_nome varchar)
FK al_id references alojamento(al_id)
FK it_nome references item(it_nome)
- servico(se_desc varchar; se_produto varchar)
- alojamento_servico(al_id int, se_produto varchar)
FK al_id references alojamento(al_id)
FK se_produto references servico(se_produto)
- fornecedor(fo_nome varchar, fo_desc varchar)

- tipologia(tip_nome varchar, tip_desc varchar)
- servico_preco(fo_nome varchar, se_produto varchar, tip_nome varchar, se_preco double)
FK se_produto references servico(se_produto)
FK fo_nome references fornecedor(fo_nome)
FK tip_nome references tipologia(fo_nome)
- distrito(ds_id int, ds_nome varchar)
- conselho(con_id int, con_nome varchar, con_distrito varchar)
FK con_distrito references distrito(ds_nome)
- freguesia(fre_id int, fre_nome varchar, fre_conselho varchar)
FK fre_conselho references conselho(con_nome)
- parcerias(par_id int, par_com_id int, par_ev_nome varchar, par_desconto int)
FK par_com_id references comboios(com_id),
FK par_ev_nome references eventos(ev_nome)

linhas(li_id int, li_caminho varchar);

comboios(com_id int, com_li_id int, com_tipo varchar, com_preco double,
com_lugares int not null);
FK com_li_id references linhas (li_id)

viagens(via_bilhete varchar, via_at_id int, via_ev_nome varchar, via_preco double,
via_email varchar, via_feedback varchar, via_avaliacao int, via_res_data_hora
datetime, via_data_hora datetime, via_caminho varchar,);
FK via_at_id references atletas(at_id),
FK key via_ev_nome references eventos(ev_nome)

lugar(lu_viajem varchar, lu_com_id int, lu_at_id int, lu_lugar int);
FK lu_viajem references viagens(via_bilhete),
FK lu_com_id references comboios(com_id),
FK lu_at_id references atletas(at_id)

```
devolucoes(dev_evento varchar, dev_via_id varchar, dev_dinheiro_devolvido
double);
FK dev_via_id references viagens(via_bilhete),
FK dev_evento references eventos(ev_nome)
```

Funcionalidades

-- Functions

-- Recebe um int e retorna um varchar

-- x representa a posição e é retornado uma string com a classificação

f_Classification(x int)

-- calcula a idade, recebe uma data de nascimento

f_Age(x date)

-- Identifica se um atleta já possui um bilhete para a sua deslocação a um evento específico;

f_hasTicket(evento_nome varchar(30), atleta_id int)

-- Calcula o nº de bilhetes adquiridos para uma comboio específico.

f_trainTickets(train_id int)

-- Calcula o caminho mais adequado entre 2 locais.

f_trainPath(origin int, destination int)

-- Views

-- View que mostra os resultados de todas as competições

v_Resultados

-- View que mostra toda a informação dos atletas

v_Atletas;

-- View que mostra as equipas que participam em cada evento

v_Equipas;

-- View que mostra todas as provas de cada evento

v_provas_por_evento;

-- View para informação dos produtos de um alojamento
v_alojamentos_info;

-- Procedures

-- SP1

-- Cria uma nova competição num evento determinado

sp_criar_comp(in prova int, in evento varchar(30), in hora time)

-- SP2

-- Adiciona um atleta à lista de atletas/equipas que irão competir na competição indicada

sp_adicionar_participante(in id_comp int, in atleta_id int, in sigla char(3))

-- SP3

-- Regista o resultado do participante na competição indicada

sp_registar_resultado(in id_comp int, in participante_id int, posicao int)

-- SP4

-- Regista o resultado da equipa e seus participantes na competição indicada

sp_registar_resultado_equipa(in id_comp int, in sigla char(3), posicao int)

-- SP5

-- Remove a competição identificada no parâmetro, nas seguintes circunstâncias:

-- a. Caso não existam resultados associados à competição (ou outros registos que sejam dependentes);

-- b. Caso existam resultados associados à competição e tenha sido enviado "True" no parâmetro force;

-- c. Caso contrário, devolve um erro.

sp_remover_comp(in id_comp int, in forc bool)

-- SP6

-- Cria uma nova prova com uma cópia de todos os dados existentes na prova indicada como parâmetro

procedure sp_clonar_prova(in id_prova int)

-- SP7

-- Adiciona um indivíduo à base de dados

add_individual(cc char(10), nome varchar(30), birthdate date, phone char(9), al_id int)

-- SP8

-- Cria um evento

create_evento(in evento varchar(30), in local_id int, in data_inicio date, in data_final date, in organizador_id int)

```

-- SP9
-- Remove relações entre empregados
-- Recebe como parametros o BI do empregado e um bool se true então
-- a. remove todos os registo que tenham esse BI como chefe
-- b. false remove o registo que tenham esse BI como subordinado
sp_remove_staff_rank(in staf_id int, in boss bool)

-- SP10
-- Calcular o Atleta com melhores resultados num evento
-- (mais 1º lugares em competições individuais e coletivas)
sp_get_best_atlete(out best_atlete_id int)

-- SP11
-- Remover Atleta da base de dados
sp_remove_atlete(in atleta_id int)

-- SP12
-- Remover Equipa da base de dados
sp_remove_team(in team_sigla varchar(3))

-- SP13
-- Mostra todos atletas da freguesia recebida como parametro
sp_fre_atletes(in freguesia int)

-- SP14
-- Dado um evento e um atleta, e com base nas localizações do evento e da morada do
-- atleta, efetua a compra de um bilhete.
-- O nº bilhete e o lugar são gerados aleatoriamente.
sp_get_ticket(in atleta_id varchar(8), in evento_nome varchar(30), email varchar(100),
feedback varchar(500), avaliacao int)

-- SP15
-- Dado um evento, procede à devolução do valor de desconto a todos os atletas que
-- utilizaram a parceria CP-Comboios;
sp_rembolse(in evento_nome varchar(30))

-- SP16
-- Cancela um comboio, e envia um email a cada atleta que tem bilhete.
sp_cancel_trip(in comboio_id int)

```

-- Triggers

-- Regista na tabela de tbl_logs na operação update à tabela participantes os seguintes dados:

- o Data e hora da operação;
 - o Identificação do atleta;
 - o Identificação da prova;
 - o Resultado anterior;
 - o Novo resultado;
 - o Posição (1º, 2º, ...).
- result_change

-- Regista na tabela de tbl_logs na operação delete à tabela participantes os seguintes dados:

- o Data e hora da operação;
 - o Identificação do atleta;
 - o Identificação da prova;
 - o Resultado anterior;
 - o Novo resultado (-1);
 - o Posição (1º, 2º, ...).
- result_delete

-- Não permite que um atleta adquira mais que um bilhete para um mesmo evento e não permite ultrapassar o nº de lugares disponíveis para um comboio.

check_viajens

Anexos

- create.sql
- logic.sql
- populate.sql
- queries.sql
- results.sql
- test_triggers.sql
- test.sql

Conclusão

Numa avaliação global, consideramos que os objetivos principais foram atingidos com sucesso. No final deste projeto sentimo-nos aptos para fazer uma análise de um problema prático real, traduzi-lo para o contexto de bases de dados e executá-lo.

A elaboração deste projeto foi desafiante para todos os membros do grupo. Principalmente na descrição do problema e na identificação de algumas entidades e na análise dos novos requisitos pedidos(Comboios etc).

Na implementação do código o que nós tivemos mais dificuldade foi:
No cálculo do caminho que um atleta percorreu na viagem.