



NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Escola de Condução

2ªFASE

MiEI/LEI - 2021/2022

Grupo: G44

Turno: P4

José Trigueiro - 58119

Martim Costa - 57876

Vasco Malta - 58592

1 - Descrição do Projeto

Neste projeto pretendemos implementar uma base de dados para uma escola de condução rodoviária.

Na escola existem dois tipos de pessoas (Alunos e Funcionários).

Os funcionários subdividem-se em instrutores e funcionários de limpeza.

Os alunos podem aprender uma ou várias categorias de veículos, sendo que cada veículo tem uma categoria associada.

A escola tem parcerias com oficinas nas quais realizam a manutenção dos veículos.

A escola de condução tem várias salas de aula, nestas são lecionadas aulas teóricas, e as aulas práticas são lecionadas num veículo. Os funcionários de limpeza realizam a limpeza das salas de aula.

1.1- Pessoas

As pessoas têm como identificador único o seu número de cartão de cidadão, além disso têm nome, data de nascimento, sexo, telemóvel, email e morada. As pessoas ou são alunos ou funcionários.

1.1.1- Alunos

Os alunos aprendem uma ou várias de categorias de veículos, podendo ir tanto a aulas práticas como aulas teóricas.

1.1.2- Funcionários

Os funcionários têm salário e data de entrada em serviço.

Eles subdividem-se em instrutores e funcionários de limpeza.

Os instrutores podem dar tanto aulas práticas como teóricas.

Os funcionários de limpeza limpam o estabelecimento incluindo as salas de aula.

1.2- Categorias

As categorias são identificadas unicamente pelo id da categoria(ex: A, B, etc.)

Cada categoria pode ter 0 ou vários alunos atribuídos. Em relação aos veículos, a categoria tem que ter pelo menos um associado.

1.3- Veículos

Os veículos serão usados para lecionar os alunos nas aulas práticas. São identificados pela matrícula, e têm como atributos o tipo(ligeiro, pesado, etc.), marca, modelo e o ano. Cada veículo está relacionado a uma só categoria e tem que estar associado a pelo menos a uma aula prática.

1.4- Oficina

A escola de condução tem parcerias com oficinas, para caso de avaria e manutenção dos veículos. A oficina é identificada pelo nipc(Número de Identificação de Pessoa Coletiva). Tem como atributos o nome, a morada, o email, e o número de telefone.

1.5- Aulas

As aulas têm data e hora e estão divididas em dois tipos:

- Teóricas: Cada teórica é dada em uma única sala.

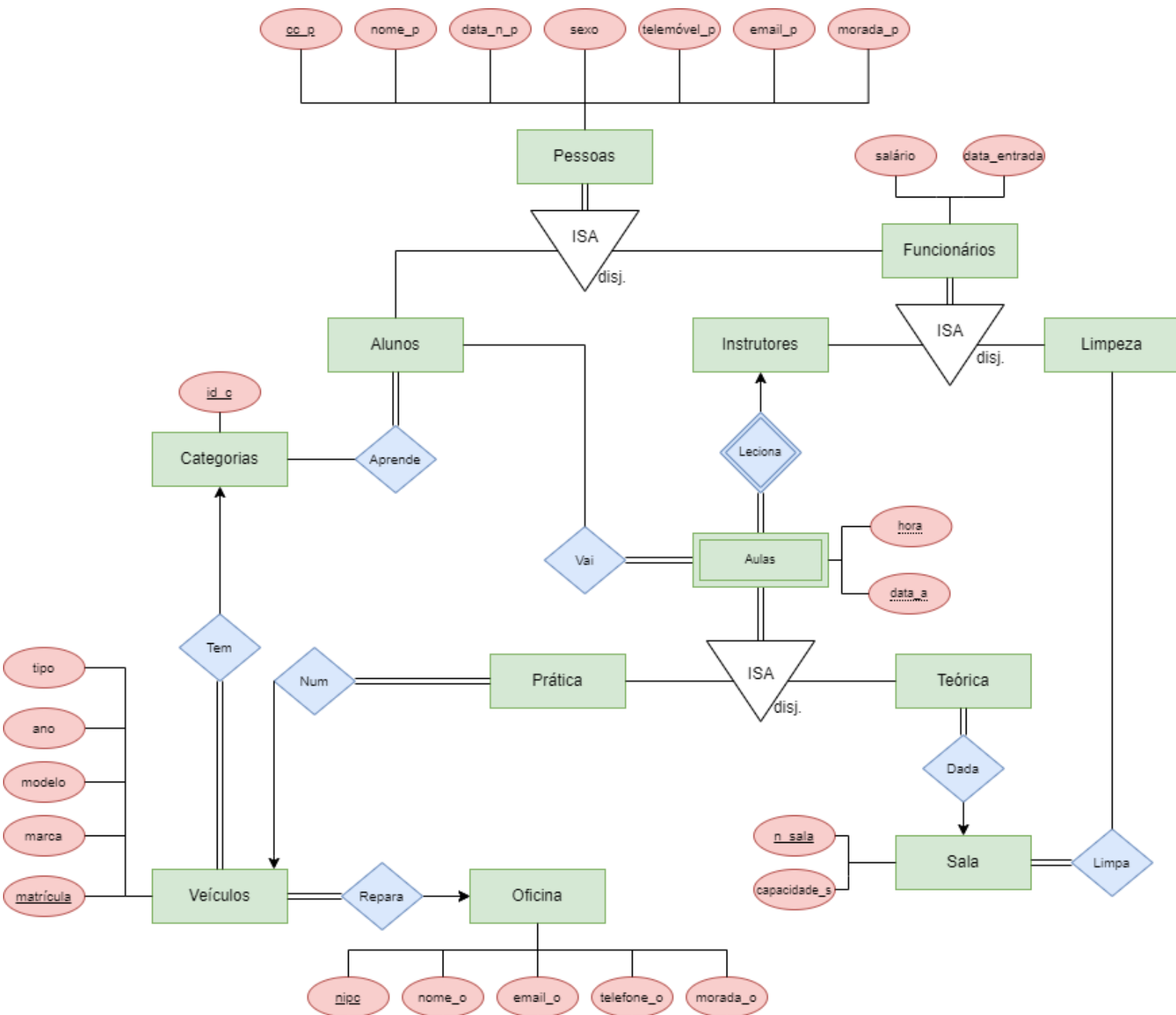
- Práticas: Cada prática é dada em um único veículo.

Cada aula é lecionada por um instrutor e dada a um ou mais alunos.

1.6- Salas

As salas são onde serão lecionadas as aulas teóricas. São identificadas pelo número da sala e têm uma certa capacidade.

2 - Modelo ER



Legenda: Verde: Entidades / Azul: Relações / Vermelho: Atributos.

3 - Esquema Relacional

Entidades:

Pessoas (cc_p, nome, data_n_p, sexo, telemóvel_p, email_p, morada_p)

Alunos (cc_p)

cc_p é chave estrangeira de Pessoas

Funcionários (cc_p, salário, data_entrada)

cc_p é chave estrangeira de Pessoas.

Categorias (id_c)

Veículos (matricula, marca, modelo, ano, tipo, id_c, nipc)

id_c é chave estrangeira de Categorias;

nipc é chave estrangeira de Oficinas.

Oficinas (nipc, nome_o, email_o, telefone_o, morada_o)

Instrutores(cc_p)

cc_p é chave estrangeira de Funcionários.

Limpeza(cc_p)

cc_p é chave estrangeira de Funcionários.

Aulas (cc_p, hora, data)

cc_p é chave estrangeira de Instrutores;

hora e data são discriminantes.

Teóricas (cc_p, hora, data, n_sala)

cc_p, hora e data são chaves estrangeiras de Aulas;

n_sala é chave estrangeira de Sala.

Práticas (cc_p, hora, data, matricula)

cc_p, hora e data são chaves estrangeiras de Aulas;

matricula é chave estrangeira de Veículos.

Sala (n_sala, capacidade_s)

Relações:

Aprende (cc_p, id_c)

cc_p é chave estrangeira de Alunos;

id_c é chave estrangeira de Categorias.

Vai (cc_p_al, cc_p_au, hora, data_a)

cc_p_al é chave estrangeira de Alunos;

cc_p_au, hora e data_a são chaves estrangeiras de Aulas.

Limpa (cc_p, n_sala)

cc_p é chave estrangeira de Limpeza;

n_sala é chave estrangeira de Sala.

4 - Código SQL (triggers e views)

4.1 - Triggers:

4.1.1: Alunos:

- Este trigger permite inserir um aluno de uma só vez fazendo-o na tabela “pessoas” e “alunos”:

```
create or replace trigger insere_aluno
instead of insert on pessoas_alunos
for each row
begin
    insert into pessoas values(:new.cc_p, :new.nome_p, :new.data_n_p, :new.sexo, :new.telemovel_p, :new.email_p, :new.morada_p);
    insert into alunos values(:new.cc_p);
end;
```

- Este trigger permite atualizar a informação de um aluno de uma só vez fazendo-o na tabela “pessoas” e “alunos”:

```
create or replace trigger atualiza_aluno
instead of update on pessoas_alunos
for each row
begin
    update pessoas set cc_p = :new.cc_p, nome_p=:new.nome_p,data_n_p= :new.data_n_p, sexo=:new.sexo,telemovel_p= :new.telemovel_p,email_p= :new.email_p,morada_p= :new.morada_p where cc_p
    update alunos set cc_p = :new.cc_p where cc_p = :new.cc_p;
end;
```

- Este trigger permite eliminar um aluno de uma só vez, removendo primeiro na tabela “aprende” em seguida em “alunos” e por fim em “pessoas”:

```
create or replace trigger apaga_aluno
instead of delete on pessoas_alunos
for each row
begin
    delete from aprende where cc_aluno = :old.cc_p;
    delete from alunos where cc_p = :old.cc_p;
    delete from pessoas where cc_p = :old.cc_p;
end;
```

4.1.2: Instrutores:

- Este trigger permite inserir um funcionário de instrução de uma só vez fazendo-o na tabela “pessoas” em seguida em “funcionarios” e por fim em “instrutores”:

```
create or replace trigger insere_instrutor
instead of insert on pessoas_instrutores
for each row
begin
insert into pessoas values(:new.cc_p, :new.nome_p, :new.data_n_p, :new.sexo, :new.telemovel_p, :new.email_p, :new.morada_p);
insert into funcionarios values(:new.cc_p,:new.salario,:new.data_entrada);
insert into instrutores values(:new.cc_p);
end;
```

- Este trigger permite atualizar a informação de um funcionário de instrução de uma só vez fazendo-o na tabela “pessoas”, em seguida em “funcionarios” e por fim em “instrutores”:

```
create or replace trigger atualiza_instrutor
instead of update on pessoas_instrutores
for each row
begin
update pessoas set cc_p = :new.cc_p, nome_p=:new.nome_p,data_n_p= :new.data_n_p, sexo=:new.sexo,telemovel_p= :new.telemovel_p,email_p= :new.email_p,morada_p= :new.morada_p where cc_p = :new.cc_p;
update funcionarios set cc_p = :new.cc_p, salario= :new.salario, data_entrada=:new.data_entrada where cc_p = :new.cc_p;
update instrutores set cc_p=:new.cc_p where cc_p=:new.cc_p;
end;
```

- Este trigger permite eliminar um funcionário de instrução de uma só vez, removendo primeiro na tabela “instrutores” em seguida em “funcionarios” e por fim em “pessoas”:

```
create or replace trigger apaga_instrutor
instead of delete on pessoas_instrutores
for each row
begin
delete from instrutores where cc_p=: old.cc_p;
delete from funcionarios where cc_p= : old.cc_p;
delete from pessoas where cc_p=: old.cc_p;
end;
```

4.1.3: Limpeza:

- Este trigger permite inserir um funcionário de limpeza de uma só vez fazendo-o na tabela “pessoas” em seguida em “funcionarios” e por fim em “limpeza”:

```
create or replace trigger insere_limpeza
instead of insert on pessoas_limpeza
for each row
begin
    insert into pessoas values(:new.cc_p, :new.nome_p, :new.data_n_p, :new.sexo, :new.telemovel_p, :new.email_p, :new.morada_p);
    insert into funcionarios values(:new.cc_p,:new.salario,:new.data_entrada);
    insert into limpeza values(:new.cc_p);
end;
```

- Este trigger permite atualizar a informação de um funcionário de limpeza de uma só vez fazendo-o na tabela “pessoas”, em seguida em “funcionarios” e por fim em “limpeza”:

```
create or replace trigger atualiza_limpeza
instead of update on pessoas_limpeza
for each row
begin
    update pessoas set cc_p = :new.cc_p, nome_p=:new.nome_p,data_n_p= :new.data_n_p, sexo=:new.sexo,telemovel_p= :new.telemovel_p,email_p= :new.email_p,morada_p= :new.morada_p where cc_p = :new.cc_p;
    update funcionarios set cc_p = :new.cc_p, salario=:new.salario, data_entrada=:new.data_entrada where cc_p = :new.cc_p;
    update limpeza set cc_p=:new.cc_p where cc_p=:new.cc_p;
end;
```

- Este trigger permite eliminar um funcionário de limpeza de uma só vez, removendo primeiro na tabela “limpa”, de seguida em “limpeza”, depois em “funcionarios” e por fim em “pessoas”:

```
create or replace trigger apaga_limpeza
instead of delete on pessoas_limpeza
for each row
begin
    delete from limpa where cc_limpeza = :old.cc_p;
    delete from limpeza where cc_p = :old.cc_p;
    delete from funcionarios where cc_p= : old.cc_p;
    delete from pessoas where cc_p=: old.cc_p;
end;
```


4.1.4: Aulas:

- Este trigger permite inserir um aula teórica de uma só vez fazendo-a na tabela “aulas” e em seguida em “teorica”:

```
create or replace trigger insere_teoricas
instead of insert on aulas_teoricas
for each row
begin
    insert into aulas values(:new.cc_p, :new.hora, :new.data_a);
    insert into teorica values(:new.cc_p, :new.hora, :new.data_a, :new.n_sala);
end;
```

- Este trigger permite inserir um aula prática de uma só vez fazendo-a na tabela “aulas” e em seguida em “pratica”:

```
create or replace trigger insere_praticas
instead of insert on aulas_praticas
for each row
begin
    insert into aulas values(:new.cc_p, :new.hora, :new.data_a);
    insert into pratica values(:new.cc_p, :new.hora, :new.data_a, :new.matricula);
end;
/
```

4.1.5: Controlo:

- Este trigger só permite pessoas maiores de 18 anos de idade:

```
create or replace trigger control_idade
before insert on pessoas
for each row
begin
    if :new.data_n_p > add_months(sysdate , -12*18) then
        RAISE_APPLICATION_ERROR(-20001, 'NAO TEM IDADE SUFICIENTE');
    end if;
end;
```

- Este trigger só permite o registo de aulas inferiores à data corrente:

```
create or replace trigger control_data_aulas
before insert on aulas
for each row
begin
    if sysdate < :new.data_a then
        RAISE_APPLICATION_ERROR(-20001, 'DATA INVALIDA');
    end if;
end;
```

- Este trigger só permite o registo de veículos com data inferior à atual:

```
create or replace trigger control_ano_veiculo
before insert on veiculos
for each row
begin
    if :new.ano > TO_NUMBER(to_char(sysdate, 'YYYY')) then
        RAISE_APPLICATION_ERROR(-20001, 'ANO INVALIDO');
    end if;
end;
```

4.2 - Views:

4.2.1: Alunos:

- Esta view junta a tabela “pessoas” e “alunos” para termos acesso a todos os dados dos alunos:

```
create or replace view pessoas_alunos AS
SELECT *
FROM pessoas inner join alunos using(cc_p)
;
```

4.2.2: Funcionários:

- Esta view junta a tabela “pessoas” e “funcionarios” para simplificar a view “pessoas_instrutores” e “pessoas_limpeza”:

```
create or replace view pessoas_funcionarios AS
SELECT *
FROM pessoas inner join funcionarios using (cc_p)
;
```

- Esta view junta a tabela “pessoas_funcionarios” e “instrutores” para termos acesso a todos os dados dos funcionários de instrução :

```
create or replace view pessoas_instrutores AS
SELECT *
FROM pessoas_funcionarios inner join instrutores using(cc_p)
;
```

- Esta view junta a tabela “pessoas_funcionarios” e “limpeza” para termos acesso a todos os dados dos funcionários de limpeza:

```
create or replace view pessoas_limpeza AS
SELECT *
FROM pessoas_funcionarios inner join limpeza using (cc_p)
;
```

4.2.3: Aulas:

- Esta view junta a tabela “aulas” e “pratica” para termos acesso a todos os dados das aulas práticas:

```
create or replace view aulas_praticas As
Select *
From aulas inner join pratica using(cc_p, hora, data_a)
;
```

- Esta view junta a tabela “aulas” e “teorica” para termos acesso a todos os dados das aulas teórica:

```
create or replace view aulas_teoricas As
Select *
From aulas inner join teorica using(cc_p, hora, data_a)
;
```

5 - Limitações

A nossa base de dados tem como limitações que:

- Só serão aceites pessoas com idade superior ou igual a 18 anos;
- O ordenado mínimo é de 740 e no máximo 9999;
- As aulas têm hora certa/exata(e.g. 15:00, 16:00, 18:00, etc), sendo essa hora entre as 9h e as 20h (ou seja as aulas começam às 9 e terminam às 21h), tendo sempre duração de uma hora;
- Ao inserir um veículo o seu ano não pode ser superior ao corrente;
- Não é possível registar aulas superiores à data corrente;
- O email tem de conter um “@” e um “.”;
- A capacidade de uma sala de aula não pode ser negativa;
- Não é possível inserir novas salas.

6 - Consultas interessantes

Algumas das consultas que achamos relevantes são:

6.1 - Quanto gasta a escola de condução por mês em salários?

```
select sum(salario)
from funcionarios
```

Esta consulta permite saber à escola quanto é gasto mensalmente em ordenados.

6.2 - Quais as categorias mais frequentadas na escola de condução ordenadas por ordem decrescente de frequência?

```
select id_c, count(cc_aluno) as alunos
from aprende
group by id_c
order by alunos desc ;
```

Esta consulta permite à escola melhor visualizar quais as categorias com maior incidência.

6.3 - Instrutores ordenados pelo número de aulas que deram e a respetiva quantidade.

```
select cc_p, count(hora) as aulas
from aulas
group by cc_p
order by aulas desc;
```

Esta consulta permite à escola melhor visualizar a quantidade de aulas já dadas por um instrutor, para possivelmente lhe dar um aumento salarial.

6.4 - As oficinas com mais veículos associados por ordem de quantidade.

```
select nipc, nome_o, count(matricula) as veiculos
from veiculos natural join oficina
group by nipc, nome_o
order by veiculos desc;
```

Esta consulta permite à escola visualizar as oficinas que mais frequenta para uma possível parceria futura.

6.5 - Quais os veículos mais usados por ordem decrescente e o seu respetivo número de aulas total?

```
select matricula, count(hora) as aulas  
from pratica  
group by matricula  
order by aulas desc;
```

Esta consulta é bastante importante para a escola poder saber o nível de utilização de cada viatura.

6.6 - Quais os instrutores que dão tanto aulas teóricas como praticas?

```
select nome_p, cc_p from teorica inner join pessoas_instrutores using (cc_p)  
intersect  
select nome_p, cc_p from pratica inner join pessoas_instrutores using (cc_p);
```

Esta consulta permite à escola saber que instrutores estão habilitados e dão tanto aulas teóricas quanto práticas.

7 - Interface e Funcionalidades

A nossa aplicação APEX tem um menu no lado esquerdo chamado “Início” que clicando nele mostra as ligações para todas as páginas do site. Cada uma das ligações direciona o utilizador a uma tabela que apresenta os dados relacionados com a mesma. Em parte delas é possível alterar os dados clicando no pequeno **lápiz** que aparece como por exemplo nos alunos em que é possível alterar alguns dados. Apenas as aulas, salas e registos é que não é possível fazer qualquer alteração aos seus elementos registados. Nestas só é permitido registar novas aulas. Para inserir, existe um pequeno botão no lado direito em cima da tabela e irá aparecer uma janela modal.

Para efetuar a eliminação de um elemento de uma tabela tem de se clicar no pequeno **lápiz** que aparece atrás de cada linha da tabela e clicar em "Eliminar". Resumindo, a opção de eliminação de um elemento é no mesmo local que a edição.

Existe uma página que se chama “Consultas” em que existem 6 botões em que cada um leva a uma página diferente onde aparece a respetiva tabela da consulta. Essa tabela não pode ser alterada manualmente, no entanto é atualizada conforme novos dados são inseridos na Base de Dados.

Outro ponto importante a referir é que para aceder às tabelas das Aulas Teóricas e Práticas tem de se aceder à página “Aulas” e no fim da página contém dois botões, cada um no extremo da página, que levam para as respetivas tabelas.

O servidor usado para o APEX foi o servidor local.

8 - Manual do Utilizador

Edição

Para editar uma linha de uma tabela tem que ir à página da mesma (voltando a referir que nem todas as tabelas dão para editar). De seguida procura-se na tabela o elemento pelo qual queremos editar e clicar no “lápiz” que se encontra à esquerda dessa linha. Altera-se os atributos pretendidos e clica-se em “Submeter”. Se pretender cancelar clique em "Cancelar".

Eliminação

Para eliminar uma linha da tabela, faça os mesmo passos que em **Editar** mas em vez de alterar os valores e submeter simplesmente clique em “Eliminar”.

Inserção

Para inserir um novo elemento tem que ir à página em que pretende inserir e clicar no botão que está à direita em cima da tabela. Depois de inserir todos os dados clicar em “Submeter”. Se pretender cancelar clique em "Cancelar".

Filtragem e Procura

Em todas as tabelas interativas é possível procurar por um valor em específico(nome, matrícula, etc. Basta clicar na caixa de texto no canto superior esquerdo da tabela.

Podemos também, ao clicar em “Actions” filtrar resultados, usando operadores como o diferente (!=), maior (>), menor (<), etc...

9 - Alterações

Fizemos uma pequena alteração em relação à Fase 1 do projeto, consideramos que a entidade Aulas, Teórica e Prática são registos de aulas que já ocorreram, não se trata de marcações de aulas.