Relatório Final

Bases de dados

Tema do trabalho:

Sistema de gestão de um clube de vídeo

Grupo 12 – Turno 8

Olena Bernatska nº 41692

David de Oliveira Pinto Gago nº 41710

João Pedro Guita de Almeida nº 42009

Índice

Introdução

Com este trabalho pretendemos desenvolver a base de dados que armazena a informação necessária para gerir o sistema de gestão de um clube de vídeo. Portanto incluiremos no nosso trabalho dados relativamente a pessoas que estão subdivididas em clientes e funcionários, a filmes, aluguer, género, realizador.

Cada cliente tem um nome, um número de cliente, idade, a morada, o número de telemóvel. Cada funcionário tem um nome, um número de funcionário, idade, a morada, o número de telemóvel e salário.

A cada funcionário está associado um único cargo, que por sua vez tem um nome do seu cargo e um salário fixo correspondente.

Cada cliente poderá alugar vários filmes.

Cada filme tem um título, um número de filme, data que neste caso é a data de lançamento e duração e número de cópias existentes no clube, realizador, actores. Os filmes podem ter vários géneros.

Sendo um clube de vídeo é razoável existir, pelo menos, uma cópia de cada filme, pelo que cada exemplar encontra-se identificado com um número de cópia, não existindo números de cópias iguais entre filmes iguais. Cada cópia de um filme poderá ser alugada mais do que uma vez, mas não em simultâneo.

Deve ser permitido o aluguer de filmes, em que cada aluguer é caracterizado por número de aluguer, data de aluguer, data de devolução, funcionário, cliente. Só deverá ser possível alugar um filme se houver cópias deste em stock. Cada aluguer terá um funcionário responsável e um cliente associado.

O pagamento será efectuado no momento da devolução e o seu valor poderá ter ou não um acrescento consoante o número de dias que o filme esteve alugado.

Alterações

Comparativamente com a descrição inicial, não existem muitas diferenças conceptuais a não ser que agora um filme pode ter mais do que um realizador.

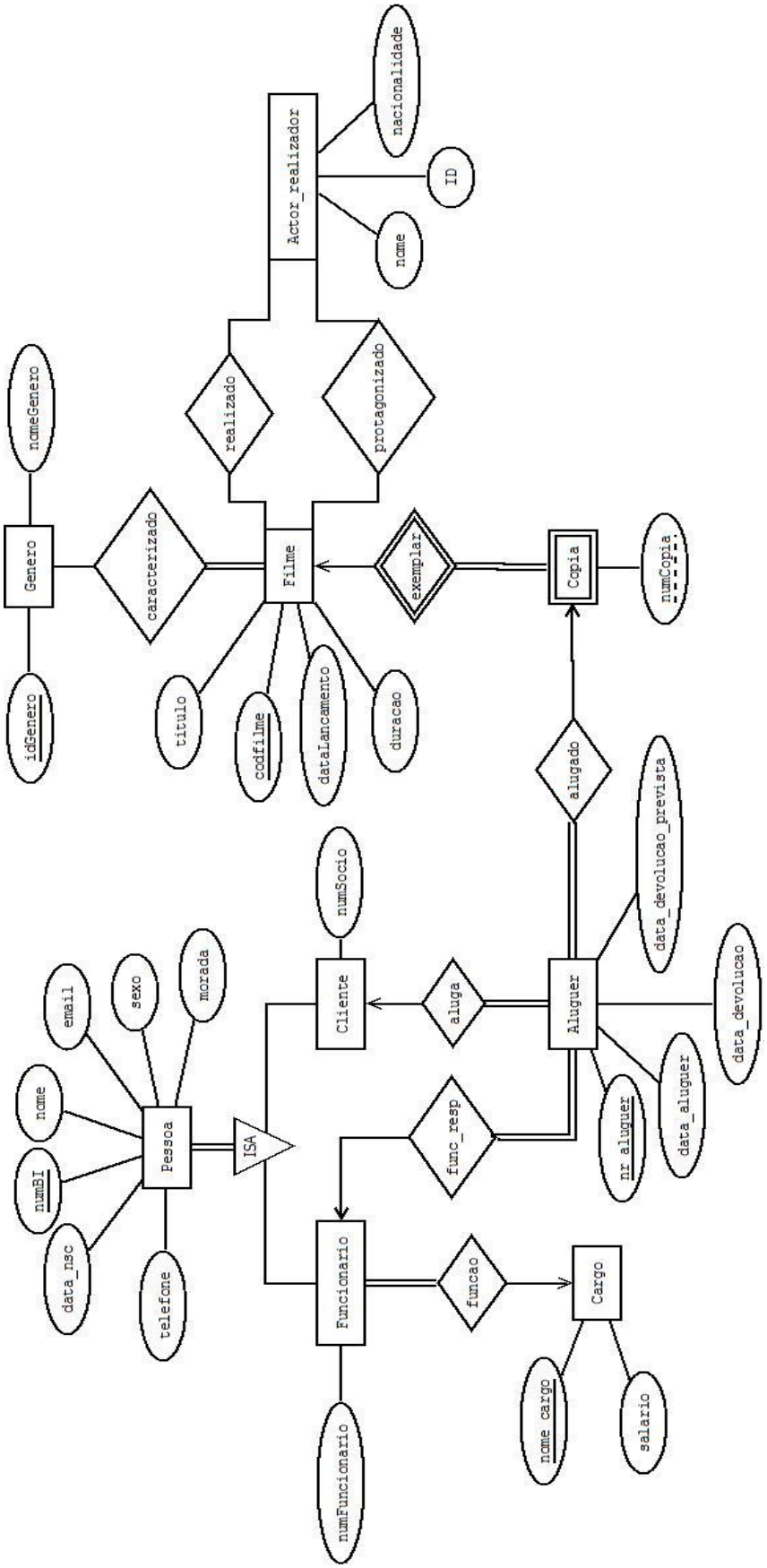
A maior parte das modificações dizem respeito à modelação de base de dados. Na Tabela das Pessoas mudámos o atributo idade para data de nascimento e agora não será possível ter mais do que um número de telefone a cada pessoa, tendo assim sido retirada a entidade Telefone e adicionado o atributo telefone à entidade Pessoa, esta alteração limita o número de telefones de uma pessoa a apenas um.

Foi retirada a entidade Multa e adicionado o atributo Data de Devolução Prevista à tabela Aluguer sendo assim possível controlar tanto se um aluguer já foi acabado e se é necessário aplicar uma multa.

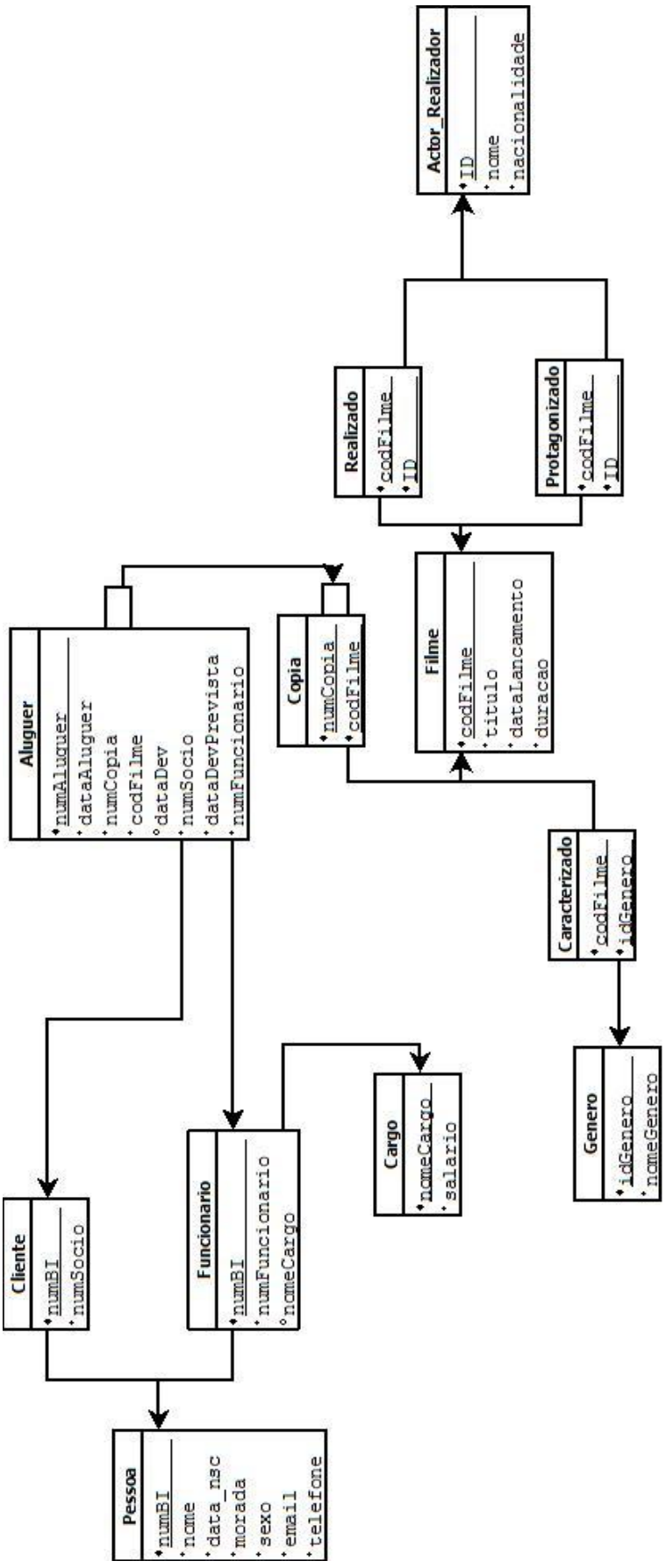
A entidade Copia foi alterada, sendo agora uma entidade fraca, assim cada cópia é referente a um filme e não precisa de um identificador global.

Alterou-se a participação da entidade Filme nas relações Caracterizado, Protagonizado e Realizado para participação parcial, possibilitando assim a existência de filmes sem actores, etc.

Modelo de dados ER



Modelo Relacional



Discussão das decisões tomadas

Uma vez que o conjunto de entidades *Cliente* e o conjunto de entidades *Funcionário* possuem atributos em comum, optámos por fazer uma generalização, criando o conjunto de entidades *Pessoa*. *Pessoa* é uma generalização total, porque não podem existir pessoas que não são funcionários nem clientes. *Cliente* e *Funcionário* são especializações sobrepostas pois uma pessoa pode pertencer a ambos.

Um clube de vídeo possui vários exemplares de cada filme, daí ser necessário a criação de um conjunto de entidade *Cópia*, é com ela que iremos efetuar as operações de aluguer. A sua chave primária será o número de cópia e terá um mapeamento N:1 entre *Cópia* e *Filme,* total do lado da *Cópia.*

Cada *Aluguer* é feito sobre uma só *Cópia* de cada vez.

A relação *função* é de N:1 com a participação total por parte da entidade *Funcionário* pois cada funcionário apenas pode ter um cargo, mas pode haver vários funcionários com o mesmo cargo.

A relação *func_resp* tem cardinalidade 1:N com participação total da entidade Aluguer, porque cada aluguer tem que ter um funcionário associado, mas cada funcionário poderá ser responsável por mais do que um aluguer

Cada filme pode ter vários géneros e cada género tem vários filmes associados, logo a cardinalidade da relação *caracterizado* é N:N.

Um filme pode ter mais do que um realizador e actor, e cada realizador/actor pode estar associado a mais do que um filme, nuns como actor e noutros como realizador. Assim a cardinalidade das relações *realizado* e *protagonizado* é, em ambas, de N:N.

Decidiu-se fazer entidades separadas para pessoa que podem ser clientes ou funcionários, e pessoas do mundo do espetáculo. Esta separação deve-se a ser necessária informação como número de BI, morada, telefone para clientes e funcionários, e essa informação sobre actores/realizadores não ser geralmente do conhecimento público.

Código SQL

```
-----------------------------------------------------
---------------------- Tabelas ----------------------
-----------------------------------------------------

drop table Pessoa cascade constraints;
create table Pessoa(
    numBI varchar2(8) not null,
    nome varchar2(50) not null,
    data_nsc timestamp not null,
    morada varchar2(50) not null,
    sexo char(1) not null check ( sexo in ( 'F' , 'M' ) ),
    email varchar2(50) not null,
    telefone varchar(15) not null,
    primary key (numBI)
);

drop table Cliente cascade constraints;
create table Cliente(
    numBI varchar2(8) not null,
    numSocio number(8) not null,
    primary key (numSocio),
    foreign key (numBI) references Pessoa(numBI)
);

drop table Cargo cascade constraints;
create table Cargo(
    nomeCargo varchar2(50) not null,
    salario number(8) not null,
    primary key (nomeCargo)
);

drop table Funcionario cascade constraints;
create table Funcionario(
    numBI varchar2(8) not null,
    numFuncionario number(8) not null,
    nomeCargo varchar2(50) not null,
    primary key (numFuncionario),
    foreign key (numBI) references Pessoa(numBI),
    foreign key (nomeCargo) references Cargo(nomeCargo)
);

drop table Filme cascade constraints;
create table Filme(
    codFilme number(8) not null,
    titulo varchar2(50) not null,
    dataLancamento timestamp not null,
    duracao number(4) not null,
    primary key (codFilme)
);

drop table Genero cascade constraints;
create table Genero(
    idGenero number(8) not null,
    nomeGenero varchar2(50) not null,
    primary key (idGenero),
    unique (nomeGenero)
);
```

```
drop table Caracterizado cascade constraints;
create table Caracterizado(
    codFilme number(8) not null,
    idGenero number(8) not null,
    primary key (codFilme, idGenero),
    foreign key (idGenero) references Genero(idGenero),
    foreign key (codFilme) references Filme(codFilme)
);

drop table Actor_Realizador cascade constraints;
create table Actor_Realizador(
    ID number (10) not null,
    nome varchar2(50) not null,
    nacionalidade varchar2(50) not null,
    primary key(ID)
);

drop table Realizado cascade constraints;
create table Realizado(
    codFilme number(8) not null,
    ID number (10) not null,
    primary key(codFilme, ID),
    foreign key(codFilme) references Filme(codFilme),
    foreign key(ID) references Actor_Realizador(ID)
);

drop table Protagonizado cascade constraints;
create table Protagonizado(
    codFilme number(8) not null,
    ID number (10) not null,
    primary key(codFilme, ID),
    foreign key(codFilme) references Filme(codFilme),
    foreign key(ID) references Actor_Realizador(ID)
);

drop table Copia cascade constraints;
create table Copia(
    numCopia number(10) not null,
    codFilme number(8) not null,
    primary key (numCopia,codFilme),
    foreign key (codFilme) references Filme(codFilme)
);

drop table Aluguer cascade constraints;
create table Aluguer(
    numAluguer number(10) not null,
    dataAluguer timestamp not null,
    numCopia number(10) not null,
    codFilme number(8) not null,
    dataDev timestamp,
    numSocio number(8) not null,
    dataDevPrevista timestamp not null,
    numFuncionario number(8) not null,
    primary key(numAluguer),
    foreign key (numCopia,codFilme) references
Copia(numCopia,codFilme),
    foreign key (numSocio) references Cliente(numSocio),
    foreign key (numFuncionario) references
Funcionario(numFuncionario)
);
```

```
-----------------------------------------------------
-------------------- Sequencias --------------------
-----------------------------------------------------

drop sequence seq_socio;
create sequence seq_socio
start with 1
increment by 1;

drop sequence seq_funcionario;
create sequence seq_funcionario
start with 1
increment by 1;

drop sequence seq_aluguer;
create sequence seq_aluguer
start with 1
increment by 1;

drop sequence seq_filme;
create sequence seq_filme
start with 1
increment by 1;

drop sequence seq_actor;
create sequence seq_actor
start with 1
increment by 1;

drop sequence seq_genero;
create sequence seq_genero
start with 1
increment by 1;

-----------------------------------------------------
----------------------- Views -----------------------
-----------------------------------------------------

create or replace view clientes as(
    select     "PESSOA"."NUMBI" as "NUMBI",
     "PESSOA"."NOME" as "NOME",
     "PESSOA"."DATA_NSC" as "DATA_NSC",
     "PESSOA"."MORADA" as "MORADA",
     "PESSOA"."SEXO" as "SEXO",
     "PESSOA"."EMAIL" as "EMAIL",
     "PESSOA"."TELEFONE" as "TELEFONE"
 from "CLIENTE" "CLIENTE",
     "PESSOA" "PESSOA"
 where   "PESSOA"."NUMBI"="CLIENTE"."NUMBI"
);

create or replace view funcionarios as(
select     "PESSOA"."NOME" as "NOME",
     "PESSOA"."DATA_NSC" as "DATA_NSC",
     "PESSOA"."MORADA" as "MORADA",
     "PESSOA"."SEXO" as "SEXO",
     "PESSOA"."EMAIL" as "EMAIL",
     "PESSOA"."TELEFONE" as "TELEFONE",
     "FUNCIONARIO"."NOMECARGO" as "NOMECARGO",
     "PESSOA"."NUMBI" as "NUMBI"
```

```
 from     "FUNCIONARIO" "FUNCIONARIO",
     "PESSOA" "PESSOA"
 where    "PESSOA"."NUMBI"="FUNCIONARIO"."NUMBI"
);


create or replace view generos_filme as (
    select     "FILME"."CODFILME" as "CODFILME",
         "CARACTERIZADO"."IDGENERO" as "IDGENERO"
     from     "CARACTERIZADO" "CARACTERIZADO",
         "FILME" "FILME"
     where    "FILME"."CODFILME"="CARACTERIZADO"."CODFILME"
);

create or replace view protagonistas_filme as(
select     "FILME"."CODFILME" as "CODFILME",
    "PROTAGONIZADO"."ID" as "ID"
 from     "PROTAGONIZADO" "PROTAGONIZADO",
     "FILME" "FILME"
 where    "FILME"."CODFILME"="PROTAGONIZADO"."CODFILME"
);

create or replace view realizadores_filme as(
select     "FILME"."CODFILME" as "CODFILME" ,
    "REALIZADO"."ID" as "ID"
 from     "REALIZADO" "REALIZADO",
     "FILME" "FILME"
 where    "FILME"."CODFILME"="REALIZADO"."CODFILME"
);

create or replace view copias_livres as(
    select "COPIA"."NUMCOPIA" as "NUMCOPIA",
         "COPIA"."CODFILME" as "CODFILME"
    from "COPIA" "COPIA"
    where ("NUMCOPIA","CODFILME") not in
        (select     "ALUGUER"."NUMCOPIA" as "NUMCOPIA",
             "ALUGUER"."CODFILME" as "CODFILME"
         from     "ALUGUER" "ALUGUER"
         where      "ALUGUER"."DATADEV" is null)
);

create or replace view generos_filme_master as (
    select    "CARACTERIZADO".rowid as "rowid",
        "FILME"."CODFILME" as "CODFILME",
         "CARACTERIZADO"."IDGENERO" as "IDGENERO"
     from     "CARACTERIZADO" "CARACTERIZADO",
         "FILME" "FILME"
     where    "FILME"."CODFILME"="CARACTERIZADO"."CODFILME"
);

create or replace view protagonistas_filme_master as(
select     "PROTAGONIZADO".rowid as "rowid",
"FILME"."CODFILME" as "CODFILME",
    "PROTAGONIZADO"."ID" as "ID"
 from     "PROTAGONIZADO" "PROTAGONIZADO",
     "FILME" "FILME"
 where    "FILME"."CODFILME"="PROTAGONIZADO"."CODFILME"
);

create or replace view realizadores_filme_master as(
select    "REALIZADO".rowid as "rowid" ,
```

```
"FILME"."CODFILME" as "CODFILME" ,
    "REALIZADO"."ID" as "ID"
 from     "REALIZADO" "REALIZADO",
     "FILME" "FILME"
 where   "FILME"."CODFILME"="REALIZADO"."CODFILME"
);

-----------------------------------------------------
-------------------- Triggers ---------------------
-----------------------------------------------------

-- Manipulacao do cliente
create or replace trigger add_cliente
  instead of insert on clientes
  for each row
  declare
    x number;
  begin
    select count(*) into x from pessoa where :new.numBI = numBI;
    if x=0 then
      insert into pessoa values (:new.numBI, :new.nome, :new.data_nsc,
:new.morada, :new.sexo, :new.email, :new.telefone);
       end if;
  insert into cliente values (:new.numBI, seq_socio.nextval);
  end;
/

create or replace trigger del_cliente
  instead of delete on clientes
  for each row
  declare x number;
  begin
    SELECT COUNT(*) INTO x FROM funcionario where numBI = :old.numBI;
    delete from cliente where numBI = :old.numBI;
    if x=0 then
      delete from pessoa where numBI = :old.numBI;
    end if;
  end;
/

create or replace trigger update_cliente
  instead of update on clientes
  for each row
  begin
    if :new.numbi = :old.numbi then
      update pessoa
      set nome = :new.nome, data_nsc = :new.data_nsc, morada =
:new.morada, sexo = :new.sexo, email = :new.email, telefone =
:new.telefone
          where numBI = :new.numBI;
    end if;
  end;
/

-- Manipulacao do funcionario
create or replace trigger add_funcionario
  instead of insert on funcionarios
  for each row
  declare
    x number;
  begin
```

```
    select count(*) into x from pessoa where :new.numBI = numBI;
    if x=0 then
      insert into pessoa values (:new.numBI, :new.nome, :new.data_nsc,
:new.morada, :new.sexo, :new.email, :new.telefone);
        end if;
  insert into funcionario values (:new.numBI, seq_funcionario.nextval,
:new.nomecargo);
  end;
/

create or replace trigger del_funcionario
  instead of delete on funcionarios
  for each row
  declare x number;
  begin
    SELECT COUNT(*) INTO x FROM cliente where numBI = :old.numBI;
    delete from funcionario where numBI = :old.numBI;
    if x=0 then
      delete from pessoa where numBI = :old.numBI;
    end if;
  end;
/

create or replace trigger update_funcionario
  instead of update on funcionarios
  for each row
  begin
    if :new.numbi = :old.numbi then
      update pessoa
      set nome = :new.nome, data_nsc = :new.data_nsc, morada =
:new.morada, sexo = :new.sexo,
          email = :new.email, telefone = :new.telefone where numbi =
:new.numbi;
      update funcionario
      set nomecargo = :new.nomecargo where numbi = :new.numbi;
    end if;
  end;
/

-- apagar atributos dos filmes
create or replace trigger del_filme
  after delete on filme
  for each row
  begin
    delete from caracterizado where codFilme=:old.codFilme;
    delete from protagonizado where codFilme=:old.codFilme;
    delete from realizado where codFilme=:old.codFilme;
  end;
/

create or replace trigger del_protagonista
  after delete on actor_realizador
  for each row
  begin
    delete from protagonizado where id = :old.id;
  end;
/

create or replace trigger del_realizador
  after delete on actor_realizador
  for each row
```

```
  begin
    delete from realizado where id = :old.id;
  end;
/

-- adicionar nova copia do filme
create or replace trigger adiciona_copia
  before insert on copia
  for each row
  declare
    numCopias number;
  begin
    select count(codFilme) into numCopias from copia where
codFilme=:new.codfilme;
    :new.numCopia := numCopias +1;
  end;
/

-- verifica se ha copias do filme disponiveis
create or replace trigger adiciona_aluguer
  before insert on aluguer
  for each row
  declare
    xx number;
  begin
    if :new.datadev != null and :new.dataaluguer > :new.datadev then
      raise_application_error(-20798, ' Datas invalidas.');
    end if;
    if :new.dataaluguer > :new.datadevprevista then
      raise_application_error(-20799, ' Data ou prazo invalidos.');
    end if;
    select min(numCopia) into xx from copias_livres where
codFilme=:new.codfilme;
    if xx>0 then
      :new.numcopia := xx;
    else
      raise_application_error(-20799, 'Nao existem copias do filme
disponiveis.');
    end if;
  end;
/

--verifica consistencia das datas
create or replace trigger dif_datas
  before update on aluguer
  for each row
  begin
    if :new.dataaluguer > :new.datadev then
      raise_application_error(-20798, ' Datas invalidas.');
    end if;
    if :new.dataaluguer > :new.datadevprevista then
      raise_application_error(-20799, ' Data ou prazo invalidos.');
    end if;
  end;
/

-- Manipulacao da master table dos generos
create or replace trigger ins_generos_filme2
  instead of insert on generos_filme_master
  for each row
  begin
```

```
    insert into caracterizado values(:new.codfilme, :new.idgenero);
  end;
/

create or replace trigger del_generos_filme2
  instead of delete on generos_filme_master
  for each row
  begin
    delete from caracterizado where rowid=:old."rowid";
  end;
/

create or replace trigger up_generos_filme2
  instead of update on generos_filme_master
  for each row
  begin
    update caracterizado
      set idgenero=:new.idgenero where rowid=:old."rowid";
  end;
/

-- Manipulacao da master table dos protagonistas
create or replace trigger insert_protagonista
  instead of insert on protagonistas_filme_master
  for each row
  begin
    insert into protagonizado values(:new.codfilme, :new.id);
  end;
/

create or replace trigger delete_protagonista
  instead of delete on protagonistas_filme_master
  for each row
  begin
    delete from protagonizado where rowid=:old."rowid";
  end;
/

create or replace trigger update_protagonista
  instead of update on protagonistas_filme_master
  for each row
  begin
    update protagonizado
      set id=:new.id where rowid=:old."rowid";
  end;
/

-- Manipulacao da master table dos realizadores
create or replace trigger insert_realizador
  instead of insert on realizadores_filme_master
  for each row
  begin
    insert into realizado values(:new.codfilme, :new.id);
  end;
/

create or replace trigger delete_realizador
  instead of delete on realizadores_filme_master
  for each row
  begin
    delete from realizado where rowid=:old."rowid";
```

```
  end;
/

create or replace trigger update_realizador
  instead of update on realizadores_filme_master
  for each row
  begin
    update realizado
      set id=:new.id where rowid=:old."rowid";
  end;
/

-------------------------------------------------------
---------------------- Funcoes ------------------------
-------------------------------------------------------

create or replace function multa (date1 in date, date2 in date)
return number is
begin
if date1<date2 then
return 2*(date2-date1);
else return 0;
end if;
end;
/

create or replace function preco (date1 in date, date2 in date)
return number is
begin
return date2-date1;
end;
/

-------------------------------------------------------
---------------------- Dados --------------------------
-------------------------------------------------------

-- Pessoa(numBI, nome, data_nsc, morada, sexo, email, telefone)
INSERT INTO Pessoa values('12345679', 'Maria', to_date('23-04-1980',
'DD-MM-YYYY'), 'R.Amalia', 'F', 'maria@rev.com',
'289123426');
INSERT INTO Pessoa values('12345677', 'Paula', to_date('01-09-1991',
'DD-MM-YYYY'), 'R.Augusta', 'F', 'paula@rev.com',
'289123455');
INSERT INTO Pessoa values('12345671', 'David', to_date('12-02-1980',
'DD-MM-YYYY'), 'Av.5 de Outubro', 'M',
'david@rev.com', '289123456');
INSERT INTO Pessoa values('12345672', 'Jose', to_date('25-04-1974',
'DD-MM-YYYY'), 'R. Santo António', 'M',
'jose@rev.com', '289123123');
INSERT INTO Pessoa values('12345673', 'Salvio', to_date('30-12-1988',
'DD-MM-YYYY'), 'Praceta Augusto', 'M',
'salvio@rev.com', '289123343');
INSERT INTO Pessoa values('12345681', 'Ana', to_date('02-04-1990',
'DD-MM-YYYY'), 'Avenida de Liberdade', 'F',
'ana@gmail.com', '218596123');
INSERT INTO Pessoa values('12345683', 'Claudio', to_date('18-09-1975',
'DD-MM-YYYY'), 'Rua de Faro', 'M',
'claudio@rev.com', '214563289');
INSERT INTO Pessoa values('12345685', 'Pedro', to_date('05-05-1987',
'DD-MM-YYYY'), 'Rua das Flores', 'M',
```

```
'pedro@gmail.com', '214563456');
INSERT INTO Pessoa values('12325688', 'Miguel', to_date('29-03-1973',
'DD-MM-YYYY'), 'Avenida Casal Ribeiro', 'M',
'miguel@gmail.com', '214563196');
INSERT INTO Pessoa values('12345688', 'Teresa', to_date('03-11-1995',
'DD-MM-YYYY'), 'Avenida Fontes Pereira', 'F',
'teresa@gmail.com', '214269197');


-- Cliente(numBI, numSocio)
INSERT INTO Cliente values('12345673', seq_socio.nextval);
INSERT INTO Cliente values('12345679', seq_socio.nextval);
INSERT INTO Cliente values('12345677', seq_socio.nextval);
INSERT INTO Cliente values('12345681', seq_socio.nextval);
INSERT INTO Cliente values('12345683', seq_socio.nextval);
INSERT INTO Cliente values('12345688', seq_socio.nextval);



-- Cargo(nomeCargo, salario)
INSERT INTO Cargo values('Balcao', '450');
INSERT INTO Cargo values('Patrao', '5000');
INSERT INTO Cargo values('Voluntário', '0');
INSERT INTO Cargo values('Gerente', '525');

-- Funcionario(numBI, numFuncionario, nomeCargo)
INSERT INTO Funcionario values('12345671',
seq_funcionario.nextval,'Balcao');
INSERT INTO Funcionario values('12345672',
seq_funcionario.nextval,'Patrao');
INSERT INTO Funcionario values('12345673',
seq_funcionario.nextval,'Balcao');
INSERT INTO Funcionario values('12345688',
seq_funcionario.nextval,'Voluntário');
INSERT INTO Funcionario values('12325688',
seq_funcionario.nextval,'Gerente');
INSERT INTO Funcionario values('12345685',
seq_funcionario.nextval,'Balcao');


-- Filme(codFilme, titulo, dataLamcamento, duracao)
INSERT INTO Filme values(seq_filme.nextval, 'Happy',
to_date('2011.10.15', 'YYYY.MM.DD'), '93');
INSERT INTO Filme values(seq_filme.nextval, 'Psycho',
to_date('1960.11.22', 'YYYY.MM.DD'), '109');
INSERT INTO Filme values(seq_filme.nextval, 'Harry Potter and the
Sorcerers Stone', to_date('2001.10.15', 'YYYY.MM.DD'), '152');
INSERT INTO Filme values(seq_filme.nextval, 'Pulp Fiction',
to_date('1994.11.25', 'YYYY.MM.DD'), '154');
INSERT INTO Filme values(seq_filme.nextval, 'Lord Of The Rings',
to_date('2001.12.21', 'YYYY.MM.DD'), '178');
INSERT INTO Filme values(seq_filme.nextval, 'Wildlife',
to_date('2014.02.23', 'YYYY.MM.DD'), '80');
INSERT INTO Filme values(seq_filme.nextval, 'Jurassic Park',
to_date('1993.06.09', 'YYYY.MM.DD'), '127');
INSERT INTO Filme values(seq_filme.nextval, 'Norbit',
to_date('2007.03.15', 'YYYY.MM.DD'), '102');
INSERT INTO Filme values(seq_filme.nextval, 'Harry Potter and the
Deathly Hallows: Part 2', to_date('2011.07.14', 'YYYY.MM.DD'), '130');
INSERT INTO Filme values(seq_filme.nextval, 'Burlesque',
to_date('2010.12.30', 'YYYY.MM.DD'), '119');
```

```
INSERT INTO Filme values(seq_filme.nextval, 'Noah',
to_date('2014.04.10', 'YYYY.MM.DD'), '138');


-- Actor_Realizador(ID, nome, nacionalidade)
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Daniel
Radcliffe','UK');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Emma
Watson','UK');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Rupert
Grint','UK');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Chris
Columbus','USA');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Steven
Spielberg','USA');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Jeff
Goldblum','USA');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Eddie
Murphy','USA');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'David
Yates','UK');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Russell
Crowe','New Zealand');
INSERT INTO Actor_Realizador values(seq_actor.nextval,'Russell
Crowe','New Zealand');


-- Realizado(codFilme, ID)
INSERT INTO Realizado values(3,4);
INSERT INTO Realizado values(7,5);
INSERT INTO Realizado values(9,8);
INSERT INTO Realizado values(9,9);


-- Protagonizado(codFilme, ID)
INSERT INTO Protagonizado values(3,1);
INSERT INTO Protagonizado values(3,2);
INSERT INTO Protagonizado values(3,3);
INSERT INTO Protagonizado values(7,6);
INSERT INTO Protagonizado values(8,7);
INSERT INTO Protagonizado values(9,1);
INSERT INTO Protagonizado values(9,2);
INSERT INTO Protagonizado values(9,3);
INSERT INTO Protagonizado values(11,9);


-- Genero(idGenero, nomeGenero)
INSERT INTO Genero values(seq_genero.nextval,'Fantasia');
INSERT INTO Genero values(seq_genero.nextval,'Aventura');
INSERT INTO Genero values(seq_genero.nextval, 'Terror');
INSERT INTO Genero values(seq_genero.nextval, 'Documentário');
INSERT INTO Genero values(seq_genero.nextval, 'Policial');
INSERT INTO Genero values(seq_genero.nextval, 'Comédia');
INSERT INTO Genero values(seq_genero.nextval, 'Musical');
INSERT INTO Genero values(seq_genero.nextval, 'Drama');


-- Caracterizado(codFilme, idGenero)
INSERT INTO Caracterizado values(3,1);
INSERT INTO Caracterizado values(3,2);
INSERT INTO Caracterizado values(6,4);
```

```
INSERT INTO Caracterizado values(11,2);
INSERT INTO Caracterizado values(10,8);
INSERT INTO Caracterizado values(11,8);
INSERT INTO Caracterizado values(8,7);

-- Copia(numCopia, codFilme)
INSERT INTO Copia values(null, 3);
INSERT INTO Copia values(null, 3);
INSERT INTO Copia values(null, 11);
INSERT INTO Copia values(null, 8);
INSERT INTO Copia values(null, 9);
INSERT INTO Copia values(null, 4);
INSERT INTO Copia values(null, 5);
INSERT INTO Copia values(null, 9);
INSERT INTO Copia values(null, 1);
INSERT INTO Copia values(null, 2);
INSERT INTO Copia values(null, 6);
INSERT INTO Copia values(null, 7);
INSERT INTO Copia values(null, 5);


-- Aluguer (numAluguer, dataAluguer, numCopia, codFilme, dataDev,
numSocio, dataDevPrevista, numFuncionario )
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.04.10',
'YYYY.MM.DD'),null,3, null, 2,to_date('2014.04.11', 'YYYY.MM.DD') ,
1);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.06.10',
'YYYY.MM.DD'),null,3,  null,2,to_date('2014.06.11', 'YYYY.MM.DD'), 3);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.05.22',
'YYYY.MM.DD'),null,11,  to_date('2014.05.23', 'YYYY.MM.DD'), 6,
to_date('2014.05.23', 'YYYY.MM.DD'), 3);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.06.23',
'YYYY.MM.DD'),null,9,  to_date('2014.07.23',
'YYYY.MM.DD'),6,to_date('2014.06.24', 'YYYY.MM.DD'), 1);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.05.23',
'YYYY.MM.DD'),null,8, to_date('2014.05.28', 'YYYY.MM.DD'), 3,
to_date('2014.05.24', 'YYYY.MM.DD'), 2);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.06.23',
'YYYY.MM.DD'),null,5,  null,5,to_date('2014.06.24', 'YYYY.MM.DD'), 1);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.05.23',
'YYYY.MM.DD'),null,4, to_date('2014.06.05', 'YYYY.MM.DD'), 4,
to_date('2014.05.24', 'YYYY.MM.DD'), 2);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.06.23',
'YYYY.MM.DD'),null,5,  null,5,to_date('2014.06.24', 'YYYY.MM.DD'), 1);
INSERT INTO Aluguer values(seq_aluguer.nextval, to_date('2014.05.23',
'YYYY.MM.DD'),null,4, to_date('2014.06.09', 'YYYY.MM.DD'), 4,
to_date('2014.05.24', 'YYYY.MM.DD'), 2);
```

Limitações /Opções tomadas na implementação da BD

Na forma como está criada a nossa base de dados não é possível apagar funcionários que tenham sido responsáveis por alugueres, nem clientes que tenham feito alugueres, a solução para despedir funcionário é criar o cargo "Despedido". E também apenas é possível eliminar um género que não tenha nenhum filme associado.

Partimos do princípio que quando uma cópia é perdida, é imediatamente reposta, logo não permitimos apagar cópias de filmes.

Foram criadas três vistas para os filmes com o intuito de possibilitar a criação de master detail forms com as relações realizado, protagonizado e caracterizado.

Foram criadas as vistas clientes e funcionários para inserir, editar e eliminar clientes/funcionários.

A vista copias_livres serve para verificar a disponibilidade de cópias do filme desejado ao criar um novo aluguer.

Criaram-se três vistas para inserir e eliminar géneros, protagonistas e realizadores de filmes. Podiam-se ter utilizado as vistas criadas para os master detail forms mas estas só foram criadas posteriormente.

Criaram-se triggers para substituir a inserção, actualização e eliminação nas vistas clientes e funcionários, verificando assim se a pessoa já existe antes de a adicionar como cliente ou funcionário e evitando apagar uma pessoa que seja ao mesmo tempo cliente e funcionário.

O trigger adiciona_copia calcula o próximo número de cópia daquele filme a ser adicionado.

O trigger adiciona_aluguer, antes de inserir um novo aluguer verifica se as datas inseridas são válidas e encontra uma cópia livre do filme escolhido.

O trigger dif_datas verifica se as datas de um aluguer são válidas quando se actualiza o update.

Foram criados triggers para tratar da inserção, actualização e eliminação de elementos dos filmes a partir dos master table forms.

As funções multa e preço calculam o preço a pagar pelo aluguer e a multa caso haja atraso na entrega.

Descrição da Interface

- Consultar e editar os dados das pessoas, como clientes (sócios) ou funcionários;
- Consultar, adicionar, eliminar e editar a informação detalhada dos funcionários;
- Consultar, adicionar, editar e eliminar clientes;
- Consultar, adicionar, editar e eliminar cargos;
- Consultar, adicionar, editar e eliminar novos realizadores e actores;
- Consultar e adicionar novos filmes, assim como editar e eliminar a sua informação;
- Consultar, eliminar e adicionar novos géneros;
- Consultar e adicionar número de cópias;
- Consultar o lucro do clube de vídeo por períodos (mês, trimestre, semestre, ano)

Identificação das Funcionalidades

O primeiro ponto está implementado na pagina inicial, e na pagina de gestão e de pesquisas.

O segundo ponto está definido em todas as listagens, inclusivamente de pessoas, clientes e funcionários, filmes, cópias, actores e realizadores e géneros, implementados quer na secção de pesquisas, quer na de gestão.

O terceiro ponto está implementado na gestão das cópias em que o número de cópias é um somatório de cópias associadas a um mesmo filme e o lucro, localizado na gestão em que é um valor a que à soma do valor adquirido com todos os alugueres é descontado o salário de cada funcionário.

O quarto ponto está definido podendo-se adicionar uma pessoa como cliente ou funcionário em gestão, ou ainda actualizar os dados referentes a estes nas listagens de clientes ou funcionários, sendo a forma de actualizar as informações de actores e realizadores da mesma forma. Poder-se-á adicionar um género, na secção de gestão, em géneros, ou alterar o nome associado a um neste mesmo sitio, o mesmo se aplica a cargos, mas na página de cargos, também em gestão e a alugueres na página de alugueres.

Em relação ao quinto ponto, este foi implementado nas páginas Detalhes de Filme; Copias ao adicionar uma nova cópia; Alugueres ao adicionar um novo; ao adicionar um novo funcionário é-nos dada uma LOV para escolhermos o cargo deste; ao adicionar elementos a filmes também nos são apresentadas LOV's de filmes, generos, protagonistas/realizadores; nos master forms são-nos apresentados LOV's ao adicionar novas linhas nos detalhes.

Em relação ao sexto ponto, existem breadcrumbs em todas as páginas para facilitar a navegação, estando todas ligadas à pagina inicial.

O sétimo ponto está implementado na página Gestao/Filmes, onde carregando no título de um filme, somos redireccionados para a página Detalhes de Filme com o código do filme como parâmetro.

As páginas de lucro e de detalhes de filme, em gestão, implementam o oitavo ponto apresentando-nos uma select list e apenas apresentando dados se tivermos escolhido algum elemento dessa lista.

Em relação ao nono ponto, existem três Master Forms: Master Form Géneros, Master Form Realizadores e Master Form Protagonistas.

Na página de criação de novos alugueres podemos observar que a data de aluguer se ainda não estiver preenchida, é automaticamente preenchida com o valor do dia corrente assim como a data de devolução prevista é automaticamente preenchida com o valor do dia seguinte.

Na página de calculo do lucro observamos a utilização das funções preço e multa em toda a tabela de alugueres sendo posteriormente tudo somado e subtraido o valor dos salários de todos os funcionários do clube de vídeo.

Manual do Utilizador

No início da aplicação existem duas páginas de entrada Gestão e Pesquisa por filmes onde existem ligações para as páginas subsequentes.

Gestão

1. Pessoa

Aqui encontrará a informação geral de todas as pessoas (clientes, funcionários) associadas ao videoclube: nºBI, nome, data de nascimento, morada, email, sexo, telefone, tipo.

a) Inserir nova cliente ou funcionário
   1. Clique em "Novo cliente" ou "Novo funcionário";
   2. Preencha os respectivos campos com os dados correspondentes;
   3. Clique em "Create".

2.Lista de Clientes

a)Inserir nova cliente
   1. Clique em "Novo cliente";
   2. Preencha os respectivos campos com os dados correspondentes;
   3. Clique em "Create".

b) Editar
   1.Clique no lápis à esquerda da linha correspondente;
   2. Modifique os dados correspondentes
   3.Clique em "Apply changes".

c) Remover (Só será possível a remoção dos clientes que ainda não têm um aluguer associado)
   1. Clique no lápis à esquerda da linha correspondente;
   2.Clique em "Delete".

3.Lista de Funcionários

Aqui encontrará a informação relativa aos funcionários: nºBI, nome, data de nascimento, morada, sexo, email, telefone e o cargo associado. O número de funcionário é gerado automaticamente.

a)Inserir novo funcionário
   1. Clique em "Novo funcionário";
   2. Preencha os respectivos campos com os dados correspondentes;
   3. Clique em "Create".

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique os dados correspondentes
3. Clique em “Apply changes”.

c)Remover (Só será possível a remoção dos funcionários que ainda não têm um aluguer associado)

1. Clique no lápis à esquerda da linha correspondente;
2. Clique em “Delete”.

4.Cargos

Aqui encontrará a informação relativa aos cargos existentes: nome do cargo como o salário correspondente.

a)Inserir novo cargo

1. Clique em “Create”;
2. Preencha os respectivos campos com os dados correspondentes;
3. Clique em “Create”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique o valor associado ao salário (neste caso não é possível alterar o nome de cargo)

Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Clique em “Delete”.

5.Filmes

Aqui encontrará a informação relativa aos filmes existentes: título, duração ano de lançamento.

a)Inserir novo filme

1. Clique em “Create”;
2. Preencha os respectivos campos com os dados correspondentes;
3. Clique em “Create”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique os dados correspondentes;
3. Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Clique em “Delete”.

6.Detalhes de Filme

Aqui encontrará a informação sobre filmes existentes no vídeo clube. Ao escolher filme será mostrada a informação existente no dado filme: nome, duração, ano. Nome, nacionalidade dos respectivos actores, como também dos realizadores. E por fim todos os géneros do dado filme.

7.Cópia

Aqui encontrará o registo de número de cópias existentes em cada filme.

a)Inserir nova cópia

1. Clique em “Create”;
2. Selecione o filme onde quer adicionar a cópia;
3. Clique em “Create”.

8.Actores/Realizadores

Aqui encontrará a informação relativa aos realizadores e actores : o nome e a nacionalidade.

a)Inserir novo realizador/actor

1. Clique em “Create”;
2. Preencha os respectivos campos com os dados correspondentes;
3. Clique em “Create”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique os dados correspondentes;
3. Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Clique em “Delete”.

9.Alugueres

Aqui encontrará o registo de todos os alugueres como a sua informação associada: número de aluguer, filme, número de cópia, nome do sócio, nome do funcionário, data de aluguer, prazo, e data de devolução (deve ser null, enquanto a devolução do filme não foi feita)

a)Inserir novo aluguer

1. Clique em “Create”;
2. Preencha os respectivos campos com os dados correspondentes;
3. Clique em “Create”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique os dados correspondentes;
3. Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Clique em “Delete”.

d)Histórico

1. Clique em “Histórico” e vai encontrar o registo de todos os alugueres feitos no clube de vídeo.

10.Adicionar elementos ao filme

1. Adicionar Géneros

a)Inserir género

1. Clique em “Create”;
2. Insira o género;
3. Clique em “Create”.

1. Adicionar Protagonistas

a)Inserir protagonista

1. Clique em “Create”;
2. Insira actor;
3. Clique em “Create”.

2. Adicionar Realizador

a)Inserir realizador

1. Clique em “Create”;
2. Insira realizador;
3. Clique em “Create”.

11.Lucro

Aqui encontrará o registo do lucro do vídeo filme em períodos, lucro mensal, trimestral, semestral, anual.

12.Master Forme Filme Géneros

Aqui encontrará a informação relativa aos filmes.

a)Inserir novo género

1. Clique em “Add Row”;
2. Escolhe o género;
3. Clique em “Apply Changes”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique os géneros correspondentes;
3. Clique em “Apply changes”.

13. Master Forme Filme Protagonistas

Aqui encontrará a informação relativa aos filmes.

a)Inserir novo actor

1. Clique em “Add Row”;
2. Escolhe o Actor;
3. Clique em “Apply Changes”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique o nome de actor;
3. Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Escolhe o actor que quer remover;
3. Clique em “Delete”.

14. Master Forme Filme Realizadores

Aqui encontrará a informação relativa aos filmes.

a)Inserir novo realizador

1. Clique em “Add Row”;
2. Escolhe o realizador;
3. Clique em “Apply Changes”.

b)Editar

1. Clique no lápis à esquerda da linha correspondente;
2. Modifique o nome do realizador;
3. Clique em “Apply changes”.

c)Remover

1. Clique no lápis à esquerda da linha correspondente;
2. Escolhe o realizador que quer remover;
3. Clique em “Delete”.

Pesquisa de Filmes

1. Pesquisa por Género

Aqui encontrará o registo de filmes que tem géneros associados pela ordem alfabética dos géneros, como a informação associado ao filme: titulo, duração e ano.

2. Pesquisa por Título

Aqui encontrará o registo de todos títulos dos filmes existentes no vídeo clube.

3. Pesquisa por Actor

Aqui será mostrado todos os actores que participam no filme, agrupados por título do filme.

4. Pesquisa por Realizador

Aqui será mostrado todos os realizadores do filme, agrupados por título.

5. Novidades

Aqui encontrará o registo de todos os filmes com a data de lançamento recente, a três meses.

Passos de demonstração das funcionalidades da BD

- Entrar na página inicial
- Seguir para Gestão
- Ir para a página Pessoas
- Adicionar um funcionário novo
- Editar ou apagar um funcionário da lista (preferencialmente o Pedro que não tem alugueres associados)
- Voltar a Gestao e ir para Copias
- Adicionar uma cópia de um filme
- Voltar a Gestao e ir para Detalhes de Filme
- Seleccionar um filme
- Voltar a Gestao e ir para Filmes
- Carregar no título de um filme (Harry Potter é uma boa escolha)
- Voltar a Gestao e ir para Master Form Filmes Protagonistas
- Carregar em edit no filme que se quer editar
- Voltar a Gestao e ir para Alugueres
- Criar um novo aluguer
- Voltar para Gestao e ir para Lucro
- Selecionar o período pretendido (salários elevados e poucos alugueres dão algum prejuízo)

Conclusão

Foi implementado com sucesso o sistema de gestão de um clube de vídeo.

A base de dados criada permite adição e alteração dos dados contidos e a sua consistencia é controlada através de triggers.

Foi criada uma interface e a sua utilização é simples e intuitiva, além de incluir todos os pontos pedidos no enunciado.

Este trabalho serviu para consolidarmos o nosso conhecimento sobre a matéria dada nas aulas e aprendemos também como criar uma base de dados que poderá ser aplicada ao mundo real.