

Base de Dados

Parte 3

Grupo 30 - Turno BD817957L02

Docente - Ricardo Eugénio Proença Rodrigues

Nome/Número	% de trabalho realizado	Total de horas
Duarte Faria 79856	50%	20
Vasco Morganho 81920	50%	20

Link para a aplicação PHP: <http://web.ist.utl.pt/ist181920/index.php>

Comandos de criação da Base de Dados

```
create table Camara
(numCamara integer not null,
 constraint pk_camara primary key(numCamara));
```

```
create table Video
(dataHoraInicio timestamp not null,
 dataHoraFim timestamp not null,
 numCamara integer not null,
 check (dataHoraInicio < dataHoraFim),
 constraint pk_video primary key(dataHoraInicio, numCamara),
 constraint fk_video_camara foreign key(numCamara) references Camara(numCamara) ON DELETE
 CASCADE ON UPDATE CASCADE);
```

```
create table SegmentoVideo
(numSegmento integer not null,
 duracao time not null,
 dataHoraInicio timestamp not null,
 numCamara integer not null,
 constraint pk_segmento primary key(numSegmento, dataHoraInicio, numCamara),
 constraint fk_segmento_video foreign key(dataHoraInicio, numCamara) references Video ON DELETE
 CASCADE ON UPDATE CASCADE);
```

```
create table Local
(moradaLocal varchar(255) not null,
 constraint pk_local primary key(moradaLocal));
```

```
create table Vigia
(moradaLocal varchar(255) not null,
 numCamara integer not null,
 constraint fk_vigia_local foreign key(moradaLocal) references Local(moradaLocal) ON DELETE
 CASCADE ON UPDATE CASCADE,
 constraint fk_vigia_camara foreign key(numCamara) references Camara(numCamara) ON DELETE
 CASCADE ON UPDATE CASCADE,
 constraint pk_vigia primary key(moradaLocal, numCamara));
```

```
create table ProcessoSocorro
(numProcessoSocorro integer,
 constraint pk_processo primary key(numProcessoSocorro));
```

```
create table EventoEmergencia
(moradaLocal varchar(255) not null,
 instanteChamada timestamp not null,
 nomePessoa varchar(255) not null unique,
 numTelefone varchar(255) not null unique,
 numProcessoSocorro integer,
```

```
constraint fk_evento_local foreign key (moradaLocal) references Local(moradaLocal) ON DELETE
CASCADE ON UPDATE CASCADE,
constraint fk_evento_processo foreign key(numProcessoSocorro) references
ProcessoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE CASCADE,
constraint pk_evento primary key(instanteChamada, numTelefone));
```

```
create table EntidadeMeio
(nomeEntidade varchar(255) not null,
constraint pk_entidade primary key(nomeEntidade));
```

```
create table Meio
(numMeio integer not null,
nomeMeio varchar(255) not null,
nomeEntidade varchar(255) not null,
constraint fk_meio_entidade foreign key(nomeEntidade) references EntidadeMeio(nomeEntidade) ON
DELETE CASCADE ON UPDATE CASCADE,
constraint pk_meio primary key (numMeio, nomeEntidade));
```

```
create table MeioCombate
(numMeio integer not null,
nomeEntidade varchar(255) not null,
constraint fk_meioCombate_meio foreign key(numMeio, nomeEntidade) references Meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
constraint pk_meioCombate primary key(numMeio, nomeEntidade));
```

```
create table MeioApoio
(numMeio integer not null,
nomeEntidade varchar(255) not null,
constraint fk_meioApoio_meio foreign key(numMeio, nomeEntidade) references Meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
constraint pk_meioApoio primary key(numMeio, nomeEntidade));
```

```
create table MeioSocorro
(numMeio integer not null,
nomeEntidade varchar(255) not null,
constraint fk_meioSocorro_meio foreign key(numMeio, nomeEntidade) references Meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
constraint pk_meioSocorro primary key(numMeio, nomeEntidade));
```

```
create table Transporta
(numMeio integer not null,
nomeEntidade varchar(255) not null,
numVitimas integer not null,
numProcessoSocorro integer not null,
constraint fk_transporta_meioSocorro foreign key(numMeio,nomeEntidade) references MeioSocorro ON
DELETE CASCADE ON UPDATE CASCADE,
constraint fk_transporta_processo foreign key(numProcessoSocorro) references ProcessoSocorro ON
DELETE CASCADE ON UPDATE CASCADE,
constraint pk_transporta primary key(numMeio, nomeEntidade, numProcessoSocorro));
```

```
create table Alocado
```

```
(numMeio integer not null,
 nomeEntidade varchar(255) not null,
 numHoras integer not null,
 numProcessoSocorro integer not null,
 constraint fk_alocado_meioApoio foreign key(numMeio, nomeEntidade) references MeioApoio ON
DELETE CASCADE ON UPDATE CASCADE,
 constraint pk_alocado primary key (numMeio, nomeEntidade, numProcessoSocorro));
```

```
create table Acciona
(numMeio integer not null,
 nomeEntidade varchar(255) not null,
 numProcessoSocorro integer not null,
 constraint fk_acciona_meio foreign key(numMeio, nomeEntidade) references Meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
 constraint fk_acciona_processo foreign key(numProcessoSocorro) references
ProcessoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE CASCADE,
 constraint pk_acciona primary key(numMeio, nomeEntidade, numProcessoSocorro));
```

```
create table Coordenador
(idCoordenador integer not null,
 constraint pk_coordenador primary key(idCoordenador));
```

```
create table Audita
(idCoordenador integer not null,
 numMeio integer not null,
 nomeEntidade varchar(255) not null,
 numProcessoSocorro integer not null,
 dataHoraInicio timestamp not null,
 dataHoraFim timestamp not null,
 dataAuditoria timestamp not null,
 texto varchar(255) not null,
 check (dataHoraInicio < dataHoraFim AND dataAuditoria >= CURRENT_TIMESTAMP),
 constraint fk_audita_acciona foreign key(numMeio, nomeEntidade, numProcessoSocorro) references
Acciona(numMeio, nomeEntidade, numProcessoSocorro) ON DELETE CASCADE ON UPDATE CASCADE,
 constraint fk_audita_coordenador foreign key(idCoordenador) references Coordenador(idCoordenador) ON
DELETE CASCADE ON UPDATE CASCADE,
 constraint pk_audita primary key(idCoordenador, numMeio, nomeEntidade, numProcessoSocorro));
```

```
create table Solicita
(idCoordenador integer not null,
 dataHoraInicioVideo timestamp not null,
 numCamara integer not null,
 dataHoraInicio timestamp not null,
 dataHoraFim timestamp not null,
 constraint fk_solicita_coordenador foreign key(idCoordenador) references Coordenador(idCoordenador) ON
DELETE CASCADE ON UPDATE CASCADE,
 constraint fk_solicita_video foreign key(dataHoraInicioVideo, numCamara) references Video(dataHoraInicio,
numCamara) ON DELETE CASCADE ON UPDATE CASCADE,
 constraint pk_solicita primary key(idCoordenador, dataHoraInicioVideo, numCamara));
```

Consultas SQL

1. Qual é o processo de socorro que envolveu maior número de meios distintos;

```
SELECT numProcessoSocorro
      FROM ((SELECT numProcessoSocorro, count(numMeio) AS nMeios
              FROM Acciona
              GROUP BY numProcessoSocorro) as t1
      CROSS JOIN
              (SELECT max(t1.nMeios) AS mMeios
              FROM (SELECT numProcessoSocorro, count(numMeio) AS nMeios
                    FROM Acciona
                    GROUP BY numProcessoSocorro) as t1
              ) as t)
      as t2
WHERE t2.nMeios = t2.mMeios;
```

2. Qual a entidade fornecedora de meios que participou em mais processos de socorro no Verão de 2018;

```
SELECT nomeEntidade
      FROM ((SELECT instanteChamada, count(numProcessoSocorro) AS nPross, nomeEntidade
              FROM Acciona NATURAL JOIN EventoEmergencia
              WHERE instanteChamada >= '2018-06-21 00:00:00' AND instanteChamada <= '2018-09-23
23:59:59'
              GROUP BY instanteChamada, nomeEntidade) as t1
      CROSS JOIN
              (SELECT max(t1.nPross) AS maxEnt
              FROM (SELECT instanteChamada, count(numProcessoSocorro) AS nPross, nomeEntidade
                    FROM Acciona NATURAL JOIN EventoEmergencia
                    WHERE instanteChamada >= '2018-06-21 00:00:00' AND instanteChamada
<= '2018-09-23 23:59:59'
                    GROUP BY instanteChamada, nomeEntidade) as t1
              ) as t)
      as t2
WHERE t2.nPross = t2.maxEnt;
```

3. Quais são os processos de socorro, referente a eventos de emergência em 2018 de Oliveira do Hospital, onde existe pelo menos um acionamento de meios que não foi alvo de auditoria;

```
SELECT numProcessoSocorro
      FROM EventoEmergencia NATURAL JOIN Acciona
      WHERE instanteChamada >= '2018-01-01 00:00:00' AND instanteChamada <= '2018-12-31 23:59:59'
AND moradaLocal = 'Oliveira do Hospital'
      EXCEPT
SELECT numProcessoSocorro
      FROM Audita;
```

4. Quantos segmentos de vídeo com duração superior a 60 segundos, foram gravados em câmeras de vigilância de Monchique durante o mês de Agosto de 2018;

```
SELECT count(numSegmento) as numSegmentos
FROM SegmentoVideo NATURAL JOIN Vigia
WHERE dataHoraInicio <='2018-08-31 23:59:59' AND dataHoraInicio >='2018-08-01 00:00:00' AND
moradaLocal = 'Monchique' AND duracao > '00:01:00';
```

5. Liste os Meios de combate que não foram usados como Meios de Apoio em nenhum processo de socorro;

```
SELECT numMeio, nomeEntidade
      FROM MeioCombate
      EXCEPT
SELECT numMeio, nomeEntidade
      FROM MeioApoio;
```

6. Liste as entidades que forneceram meios de combate a todos os Processos de socorro que acionaram meios;

```
SELECT DISTINCT T.nomeEntidade
FROM Acciona as T
WHERE NOT EXISTS(
      (SELECT numProcessoSocorro
      FROM ProcessoSocorro)
      EXCEPT
      (SELECT Acciona.numProcessoSocorro
      FROM MeioCombate, Acciona
      WHERE Acciona.numMeio = MeioCombate.numMeio AND
      T.nomeEntidade = Acciona.nomeEntidade));
```

Explicação da arquitetura da aplicação PHP e das relações entre os diversos ficheiros

No index contêm links para as listas das tabelas de Processos de Socorro, Locais, Eventos de Emergência, Meios, Meios de Combate, Meios de Socorro, Meios de Apoio e Entidades. No início de cada página anteriormente referida existe um link para a criação de uma nova entrada na respectiva tabela e em cada linha da tabela existe um link para remover a dita linha. No caso dos Processos de Socorro ainda é possível listar os meios accionados por cada processo, accionar novos meios e associá-los a eventos de emergência que não têm processo de socorro atribuído. Na página que lista os Locais existe em cada linha da tabela um link para listar os Meios de Socorro que operaram no respetivo local.

Quando queremos inserir uma nova entrada numa dada tabela somos redirecionados para uma página que contém um formulário. Após preencher todos os campos somos redirecionados para outra página de confirmação que valida os novos dados inseridos e mostra uma mensagem de sucesso ou erro.

Nas páginas dos Meios de Combate, Meios de Socorro e Meios de Apoio é possível editar a entidade de um determinado meio. O processo de inserção e validação dos novos dados é idêntico ao “inserir”.