

Departamento de Informática

MIEI

Relatório da 2ª fase do Projeto de Base de Dados

Grupo 40

# Hotel Fctnense

Bases de Dados

2016-2017

Ana Margarida Duarte de Sousa (47406)

João André Furtado Teixeira (48047)

Mafalda Filipa Caldeira Camilo (47981)

**Turno Prático 4**

**Professor:** Nuno C. Marques

# Índice

<b>Descrição do tema do projeto e objetivos da base de dados .....</b>	<b>3</b>
<b>Modelo ER .....</b>	<b>5</b>
<b>Código SQL de criação da base de dados relacional para o projeto .....</b>	<b>6</b>
Tabelas .....	7
Código SQL .....	8
Criação de Tabelas.....	8
Números de Sequência.....	11
Views.....	11
Triggers.....	13
Inserts.....	18
<b>Atribuição de privilégios .....</b>	<b>20</b>
<b>Manual de Utilizador da Aplicação APEX.....</b>	<b>21</b>

# Descrição do tema do projeto e objetivos da base de dados

São imensos os dados a ser geridos num hotel, desde as principais informações dos funcionários e clientes, até ao mínimo pormenor dos produtos de uma cozinha de um hotel, que já por si só é uma base de dados. Decidimos portanto num conceito mais geral e menos pormenorizado (focando-nos mais nas reservas dos clientes e no papel dos funcionários) organizar uma base de dados completa relativa a um Hotel, que será portanto o tema principal do projeto do nosso grupo.

De uma forma resumida, tal base de dados permitirá que sejam introduzidos dados relativos aos funcionários, clientes e administradores de um hotel, bem como dados relativos às reservas de quartos por parte desses mesmos clientes.

Relativamente a mudanças nas informações que irão ser guardadas nesta base de dados completa sobre o Hotel Fctnense, decidimos retirar a parte relativa ao minibar e, em vez disso, adicionar uma entidade relativa ao checkout dos clientes, de forma a focar-nos num conceito mais geral deste hotel, e adicionar algo que também será mais útil relativamente ao funcionamento do mesmo.

Tal como foi referido na descrição do tema anterior, mais precisamente, a base de dados deverá guardar, para cada pessoa, o nif, que os distinguirá, o nome, o email, o número de telefone e o sexo. Apesar destes atributos em comum, há que distinguir as pessoas que interagem de formas diferentes num hotel e portanto mais especificamente teremos então os clientes, cuja informação acerca da nacionalidade e do número de cliente deverá ser armazenada, e ainda os funcionários, que têm como atributo o número de Funcionário, e os administradores, sendo que estes últimos têm como atributo, para além do número de administrador, os cargos correspondentes.

É de frisar que não vamos impor que um trabalhador do hotel não possa ser um cliente do mesmo, portanto poderá acontecer que um determinado funcionário ou administrador possa também ser um cliente do hotel e usufruir dessa mesma qualidade. De forma a distinguir as pessoas que são clientes, funcionários, ou os dois, basta verificar se contêm um número de cliente e/ou número de funcionário.

A todo o funcionário está associado um dado trabalho num sector do hotel e não se impede que um funcionário trabalhe em mais do que um sector. Por exemplo, o funcionário com nif correspondente 123456789 pode trabalhar no sector da receção desempenhando um certo cargo e ao mesmo tempo trabalhar no sector de limpezas desempenhando outro cargo. Para além disto há que guardar informação acerca de quem é o responsável por cada função desempenhada por um dado funcionário num sector sendo que todos os responsáveis pelos funcionários são do tipo administrador.

Os sectores serão distinguidos entre si por um código único e terão uma breve descrição sobre o que representam. Por exemplo, o sector 1 poderá ser o sector da limpeza e como descrição consideramos o facto de este ser responsável pela limpeza dos quartos e do hotel em si.

Quanto às reservas do hotel, estas são unicamente distinguidas pelo seu número de reserva e será também armazenada informação acerca das datas da mesma, da capacidade do quarto de cada reserva e ainda o seu preço. A cada reserva está associado somente um cliente, e obviamente, não poderá existir reservas sem um cliente associado. Por fim, cada reserva terá um quarto reservado para esta.

Cada quarto será apenas distinguido pelo seu número, e será também guardada

informação acerca do piso onde este se encontra, e ainda, terão um código de Quarto associado.

Os quartos poderão (ou não) ter várias facilidades. Quanto às últimas estas serão distinguidas pelo seu código único e também será preciso guardar uma pequena descrição sobre essa mesma facilidade. Por exemplo, o código 'F' representa a facilidade quarto para fumadores.

Existem vários tipos de quartos num hotel e cada tipo tem uma determinada capacidade máxima de hóspedes associada e ainda um preço base de cada códigoQ, sendo este o código associado que irá distinguir cada tipo de quarto. Neste caso, imaginemos por exemplo que um tipo de quarto com o código 'T1' dará respeito a quartos standard com capacidade máxima de 4, que tem um preço base de 200€.

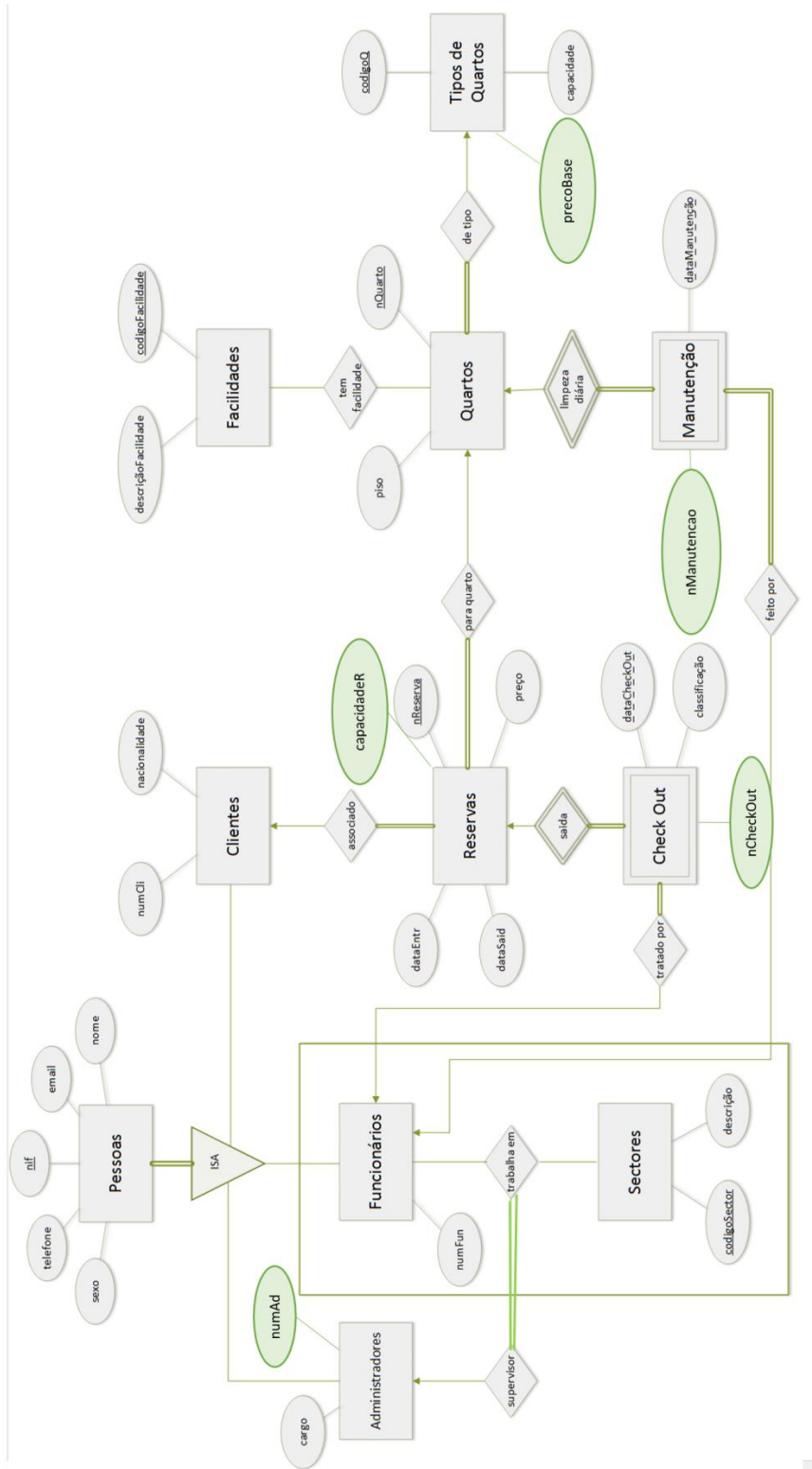
Gostaríamos ainda de armazenar informação acerca da manutenção realizada nos quartos, ou seja, informação sobre a limpeza diária efetuada pelos funcionários do hotel nos quartos e a respectiva data em que foram realizados. Note-se que assumimos que um funcionário faz uma limpeza geral por dia em cada quarto, sendo que um mesmo funcionário pode limpar vários quartos diferentes no mesmo dia mas um quarto só terá uma limpeza por dia por um funcionário. Por exemplo, o funcionário com nif 123456789 efetuou uma limpeza geral no quarto 1 no dia 01-01-2018 e esse mesmo funcionário pode também efetuar uma limpeza no quarto 2, no mesmo dia.

Por fim, associada à reserva de cada quarto iremos guardar informação acerca do checkOut de uma reserva, sendo este responsável por um funcionário e que contém uma data de checkOut e uma classificação de satisfação referente à reserva e estadia do hotel, sendo que esta classificação será de 1,2,3,4 ou 5 estrelas. É importante referir que consideramos que a data de checkOut poderá anteceder a data de Saída de uma reserva, não as considerando portanto iguais.

## Alterações em relação à versão preliminar

Em relação à versão inicial do projecto alteramos alguns aspectos, nomeadamente inserimos alguns atributos que se mostraram necessários durante o desenvolvimento da aplicação.

❖ Diagrama final:



Na entidade Administradores foi adicionado um atributo numAd que corresponde ao número de administrador e tal como nos clientes e funcionários esse número serve também como identificador do administrador na base de dados.

Na entidade Reservas foi adicionado um atributo capacidadeR, servindo assim para ser guardado na base a capacidade nos quartos que o cliente pretende reservar, assim será possível no momento da reserva fazer a escolha do quarto com dimensões adequadas.

Na entidade Tipos de Quartos foi adicionado o atributo precoBase, este atributo guarda o preço base de cada tipologia de quartos, por exemplo os quartos 'T0' custam 100€ os 'T1' custam 200€.

Na entidade Manutenção e Checkout foram adicionados os atributos nManutenção e nCheckOut, respetivamente, sendo que estes atributos foram exclusivamente inseridos por causa do apex, servindo assim como um counter para o número de chave do apex.

# Código SQL de criação da base de dados relacional para o projeto

## Tabelas

Nota: A chave primária de cada relação é formada pelos atributos sublinhados.

- Pessoas (nif, nome, email, sexo, telefone);
- Funcionários (nifFun, numFun);
- Clientes (nifCli, numCli, nacionalidade);
- Administradores (nifAd, cargo, numAd);
- Sectores (codigoSector, descricao);
- trabalhaEm (nifFun, codigoSector, nifAd);
- tipoQuartos (codigoQ, capacidade, precoBase);
- Quartos (nQuarto, codigoQ, piso);
- Manutenção (nManutencao, dataManutencao, nQuarto, nifFunMan);
- Facilidades (codigoFacilid, descrFacilid);
- temFacilidades (nQuarto, codigoFacilid);
- Reservas (nReserva, nifCli, nQuarto, preco, dataEntrada, dataSaida);
- CheckOut (dataCheckOut, nReserva, classificacao, nifFun, nCheckOut);

# Código SQL

De forma a proporcionar um melhor entender do nosso código e, em especial, das restrições de integridade, adicionamos alguns comentários que poderão ser úteis.

## Criação de Tabelas

```
drop table pessoas cascade constraints;
```

### --CRIAR TABELA PESSOAS

```
create table pessoas (  
  nif number(9),  
  nome varchar2(30) not null,  
  email varchar2(60) not null,  
  sexo char(1) not null CHECK (sexo IN ( 'F' , 'M' )), -- sexo tem de ser feminino(F) ou masculino(M)  
  telefone number (15),  
  primary key(nif),  
  unique (email), -- tem que haver um email diferente por pessoa  
  check (nif between 100000000 and 999999999) -- Garantir que o nif se encontra entre estes numeros  
);
```

```
drop table funcionarios cascade constraints;
```

### --CRIAR TABELA FUNCIONARIOS

```
create table funcionarios (  
  nifFun number(9),  
  numFun number(5) not null,  
  primary key (nifFun),  
  foreign key(nifFun) references pessoas(nif),  
  unique (numFun) -- tem que haver um numero de funcionario(numFun) diferente para cada funcionario  
);
```

```
drop table clientes cascade constraints;
```

### --CRIAR TABELA CLIENTES

```
create table clientes (  
  nifCli number(9),  
  numCli number(5) not null,  
  nacionalidade varchar2(30),  
  primary key (nifCli),  
  foreign key(nifCli) references pessoas(nif),  
  unique (numCli) -- tem que haver um numero de cliente(numCli) diferente para cada cliente  
);
```

```
drop table administradores cascade constraints;
```

### --CRIAR TABELA ADMINISTRADORES

```
create table administradores (  
  nifAd number(9),  
  cargo varchar2(20)not null,  
  numAd number(5),
```



```

primary key (nifAd),
foreign key(nifAd) references pessoas(nif),
check (cargo in ('G', 'SG', 'P')),-- G -> Gerente | SP -> subGerente | P -> patroa , so existe estes 3
cargos para administradores
unique(numAd)
);

```

```

drop table sectores cascade constraints;

```

**--CRIAR TABELA SECTORES**

```

create table sectores (
codigoSector varchar2(2),
descricao varchar2(30),
primary key (codigoSector)
);

```

```

drop table trabalhaEm cascade constraints;

```

**--CRIAR TABELA TRABALHA EM**

```

create table trabalhaEm (
nifFun number(9),
codigoSector varchar2(2),
nifAd number(9),
primary key (nifFun,codigoSector),
foreign Key (nifFun) references funcionarios(nifFun),
foreign key (codigoSector) references sectores(codigoSector),
foreign key (nifAd) references administradores(nifAd)
);

```

```

drop table tipoQuartos cascade constraints;

```

**--CRIAR TABELA TIPO QUARTOS**

```

create table tipoQuartos (
codigoQ varchar(2),
capacidade number(1) not null,
precoBase number(3) not null,
primary key (codigoQ)
);

```

```

drop table quartos cascade constraints;

```

**--CRIAR TABELA QUARTOS**

```

create table quartos (
nQuarto number(2),
codigoQ varchar2(2),
pisso number(2) not null,
primary key (nQuarto),
foreign key (codigoQ) references tipoQuartos(codigoQ)
);

```

```

drop table manutencao cascade constraints;

```

**--CRIAR TABELA MANUTENCAO**

```

create table manutencao (
nManutencao number(5),
dataManutencao DATE,

```

```

nQuarto number(2),
nifFunMan number(9),
primary key (dataManutencao,nQuarto),
foreign key (nQuarto) references quartos(nQuarto),
foreign key (nifFunMan) references funcionarios(nifFun),
unique(nManutencao)
);

```

```

drop table facilidades cascade constraints;

```

**--CRIAR TABELA FACILIDADES**

```

create table facilidades (
codigoFacilid varchar2(3),
descrFacilid varchar2(30) not null,
primary key (codigoFacilid)
);

```

```

drop table temFacilidades cascade constraints;

```

**--CRIAR TABELA TEM FACILIDADES**

```

create table temFacilidades (
nQuarto number(2),
codigoFacilid varchar2(3),
primary key (nQuarto,codigoFacilid),
foreign key (nQuarto) references quartos(nQuarto),
foreign key (codigoFacilid) references facilidades(codigoFacilid)
);

```

```

drop table reservas cascade constraints;

```

**--CRIAR TABELA RESERVAS**

```

create table reservas (
nReserva number(10),
nifCli number(9),
nQuarto number(2) not null,
preco number(10) not null,
dataEntrada DATE not null,
dataSaida DATE not null,
capacidadeR number(1) not null,
primary key (nReserva),
foreign key (nifCli) references clientes(nifCli),
foreign key (nQuarto) references quartos(nQuarto),
check (preco > 0),
check (dataEntrada < dataSaida) -- o preco para a reserva de um quarto(preco) terá que ser
positivo nao nulo e a data de entrada inferior a de saida
);

```

```

drop table checkOut cascade constraints;

```

**--CRIAR TABELA CHECKOUT**

```

create table checkOut(
dataCheckOut DATE,
nReserva number(10),
classificacao number(1)not null,
nifFun number(9),

```

```

nCheckOut number(5),
primary key (dataCheckOut, nReserva),
foreign key (nifFun) references funcionarios(nifFun),
foreign key (nReserva) references reservas(nReserva),
check (classificacao in (1,2,3,4,5)), -- no checkOut o cliente deixa sempre uma classificacao pelo
servico/estadia do hotel tendo essa a variar de 1 a 5
unique(nCheckOut)
);

```

## Números de sequência

### --CRIAR SEQUENCIA PARA NUMERO FUNCIONARIO

```

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

```

### --CRIAR SEQUENCIA PARA NUMERO CLIENTE

```

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

```

### --CRIAR SEQUENCIA PARA NUMERO ADMINISTRADORES

```

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

```

### --CRIAR SEQUENCIA PARA NUMERO RESERVA

```

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

```

### --CRIAR SEQUENCIA PARA NUMERO CHECKOUT

```

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

```

### --CRIAR SEQUENCIA PARA NUMERO MANUTENCAO

```

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

```

## Views

### --CRIAR VIEW CLIENTES

```

drop view v_clientes cascade constraints;
create or replace view v_clientes as
select numCli, nome, nif, nacionalidade, email, sexo, telefone
from clientes inner join pessoas on(pessoas.nif = clientes.nifCli)
order by numCli;

```

### --CRIAR VIEW ADMINISTRADORES

```

drop view v_administradores cascade constraints;

```

```
create or replace view v_administradores as
select numAd, nome, nif, cargo, email, sexo, telefone
from administradores inner join pessoas on(pessoas.nif = administradores.nifAd)
order by numAd;
```

#### **--CRIAR VIEW FUNCIONARIOS**

```
drop view v_funcionarios cascade constraints;
create or replace view v_funcionarios as
select numFun, nif, nome, email, sexo, telefone
from funcionarios inner join pessoas on(pessoas.nif = funcionarios.nifFun)
order by numFun;
```

#### **--CRIAR VIEW RESERVAS**

```
drop view v_reservas cascade constraints;
create or replace view v_reservas as
select nReserva, capacidadeR , nQuarto, preco, dataEntrada, dataSaida, nifCli, nome, email,
sexo, telefone, nacionalidade
from reservas inner join clientes using(nifCli) inner join pessoas on(pessoas.nif = nifCli)
order by nReserva desc;
```

#### **--CRIAR VIEW CHECKOUT**

```
drop view v_checkout cascade constraints;
create or replace view v_checkout as
select nCheckOut, nReserva, classificacao, nifCli, dataCheckout, nifFun
from checkOut natural join reservas
order by nCheckOut desc;
```

#### **--CRIAR VIEW FACILIDADES DOS QUARTOS**

```
drop view v_facilidadesQuarto cascade constraints;
create or replace view v_facilidadesQuarto as
select ROWID pk, nQuarto, codigoFacilid
from temFacilidades;
```

#### **--CRIAR VIEW MANUTENCAO**

```
drop view v_manutencao cascade constraints;
create or replace view v_manutencao as
select nManutencao, dataManutencao, nQuarto, nifFunMan, nome
from manutencao inner join funcionarios on(nifFunMan = nifFun) inner join pessoas
on(nifFunMan = nif)
order by dataManutencao desc;
```

#### **--CRIAR VIEW PARA TRABALHA EM**

```
drop view v_trabalhaEm cascade constraints;
create or replace view v_trabalhaEm as
select ROWID pk, nifFun, codigoSector, nifAd
```

```
from trabalhaEm;
```

#### **--CRIAR VIEW DE QUARTOS**

```
drop view v_quartos cascade constraints;  
create or replace view v_quartos as  
select nQuarto, nQuarto as numeroQuarto, codigoQ , piso from quartos;
```

#### **--CRIAR VIEW PARA TIPO DE QUARTOS**

```
drop view v_tipoQuartos;  
create or replace view v_tipoQuartos as  
select * from tipoQuartos;
```

### Triggers

#### **--INSERIR ADMINISTRADORES**

```
create or replace trigger insert_administradores  
instead of insert on v_administradores  
referencing new as NEW  
for each row  
  declare numero number;  
begin  
  select count(*) into numero -- verificar se a pessoa já existe na base de dados através do seu  
  nif  
  from pessoas  
  where nif=:NEW.nif;  
  
  if numero > 0 then --se já existir então inserir só na tabela dos administradores  
    insert into administradores  
    values (:NEW.nif, :NEW.cargo, seq_administradores.nextval);  
  end if;  
  
  if numero < 1 then --se não existir, inserir tanto na tabela pessoas como administradores  
    insert into pessoas  
    values (:NEW.nif, :NEW.nome, :NEW.email, :NEW.sexo, :NEW.telefone);  
  
    insert into administradores  
    values (:NEW.nif, :NEW.cargo, seq_administradores.nextval);  
  end if;  
end;  
/
```

#### **--UPDATE ADMINISTRADORES**

```
create or replace trigger update_administradores  
instead of update on v_administradores
```

```

referencing new as new old as old
for each row
begin
    update pessoas
    set nome=:new.nome,
    email=:new.email,
    sexo=:new.sexo,
    telefone=:new.telefone
    where nif=:new.nif;

    update administradores
    set cargo = :new.cargo
    where nifAd = :new.nif;
end;
/

```

#### **--INSERIR FUNCIONARIOS**

```

create or replace trigger insert_funcionarios
instead of insert on v_funcionarios
referencing new as NEW
for each row
    declare numero number;
begin
    select count(*) into numero -- verificar se a pessoa já existe na base de dados através do seu
nif
    from pessoas
    where nif=:NEW.nif;

    if numero > 0 then --se já existir então inserir só na tabela dos funcionários
        insert into funcionarios
        values (:NEW.nif, seq_funcionarios.nextval);
    end if;

    if numero < 1 then --se não existir, inserir tanto na tabela pessoas como funcionários
        insert into pessoas
        values (:NEW.nif, :NEW.nome, :NEW.email, :NEW.sexo, :NEW.telefone);

        insert into funcionarios
        values (:NEW.nif, seq_funcionarios.nextval);
    end if;
end;
/

```

#### **--UPDATE FUNCIONARIOS**

```

create or replace trigger update_funcionarios

```

```
instead of update on v_funcionarios
referencing new as new old as old
for each row
begin
```

```
    update pessoas
    set nome=:new.nome,
        email=:new.email,
        sexo=:new.sexo,
        telefone=:new.telefone
    where nif=:new.nif;
```

```
end;
```

```
/
```

#### **--INSERIR NA TABELA TRABALHA EM**

```
create or replace trigger insert_trabalhaem
```

```
instead of insert on v_trabalhaem
```

```
referencing new as NEW
```

```
for each row
```

```
begin
```

```
    insert into trabalhaem (nifFun ,codigoSector , nifAd)
    values (:NEW.nifFun,:NEW.codigoSector,:NEW.nifAd);
```

```
end;
```

```
/
```

#### **--DELETE NA TABELA TRABALHA EM**

```
create or replace TRIGGER delete_trabalhaEm
```

```
instead of delete on v_trabalhaEm
```

```
referencing old as OLD
```

```
for each row
```

```
begin
```

```
    delete from trabalhaem where nifFun = :OLD.nifFun;
```

```
end;
```

```
/
```

#### **--UPDATE CLIENTES**

```
create or replace trigger update_clientes
```

```
instead of update on v_clientes
```

```
referencing new as NEW old as OLD
```

```
for each row
```

```
begin
```

```
    update pessoas
    set nome = :NEW.nome,
        email = :NEW.email,
        telefone = :NEW.telefone,
        sexo = :NEW.sexo
```

```

where nif = :NEW.nif;

update clientes
set nacionalidade = :NEW.nacionalidade
where nifCli = :NEW.nif;
end;
/

--INSERIR RESERVA
create or replace trigger insert_reserva
instead of insert on v_reservas
referencing new as NEW
for each row
declare quarto number; cliente number; precoR number;
begin
    select count(*) into cliente --ver se o cliente que está a reservar já existe na base de dados
    através do seu nif
    from pessoas where nif=:NEW.nifCli;

    if cliente < 1 then --se nao existir insersir tanto na tabela pessoas como em clientes
        insert into pessoas values (:NEW.nifCli, :NEW.nome, :NEW.email, :NEW.sexo,
:NEW.telefone);
        insert into clientes values (:NEW.nifCli, seq_clientes.nextval, :NEW.nacionalidade);
    end if;

    --escolher o quarto adequado para reserva: de todos os quartos existentes no hotel com a
    capacidade desejada, retirar os quartos que se encontram indisponiveis nas datas do cliente, e
    escolher o quarto com menor número de quarto
    select min(nQuarto) into quarto from ((
        select nQuarto from quartos inner join tipoQuartos using(codigoQ) where
:NEW.capacidadeR = capacidade)
    minus
    (select nQuarto from reservas
    where :NEW.dataEntrada < dataSaida and :NEW.dataSaida > dataEntrada));

    --determinar o preço da reserva do cliente: o preço base do tipo de quarto (T0, T1, T2,
    T3) que serve para o cliente vezes o número de noites que o cliente se encontra no hotel
    select precoBase * (:NEW.datasaida-:NEW.dataentrada) into precoR
    from tipoquartos
    where :NEW.capacidadeR = capacidade;

    insert into reservas values(seq_reservas.nextval, :NEW.nifcli, quarto, precoR,
:NEW.dataEntrada, :NEW.dataSaida, :NEW.capacidadeR);

    if quarto < 1 then

```



```

        raise_application_error(-20002,'Lamentamos mas nao temos disponibilidade para as
suas datas');
    end if;
end;
/

```

#### **--INSERIR CHECKOUT**

```

create or replace trigger insert_checkout
instead of insert on v_checkout
referencing new as NEW
for each row
    declare quarto number;
begin
    insert into checkOut
    values (:NEW.dataCheckout, :NEW.nReserva, :NEW.classificacao, :NEW.nifFun,
seq_checkout.nextval);
end;
/

```

#### **--INSERIR MANUTENCAO**

```

create or replace trigger insert_manutencao
instead of insert on v_manutencao
referencing new as NEW
for each row
begin
    insert into manutencao values(seq_manutencao.nextval, :NEW.dataManutencao,
:NEW.nQuarto, :NEW.nifFunMan);
end;
/

```

#### **--INSERIR FACILIDADES EM QUARTO**

```

create or replace trigger insert_temfacilidadeQuarto
instead of insert on v_facilidadesQuarto
referencing new as NEW
for each row
begin
    insert into temfacilidades (nQuarto,codigoFacilid)
    values (:NEW.nQuarto,:NEW.codigoFacilid);
end;
/

```

#### **--DELETE FACILIDADES DO QUARTO**

```

create or replace trigger delete_temfacilidadeQuarto
instead of delete on v_facilidadesQuarto
referencing old as OLD

```

```

for each row
begin
    delete from temfacilidades where ROWID = :OLD.pk;
end;
/

```

#### **--ALTERAR PRECO BASE DO TIPO DE QUARTO**

```

create or replace trigger update_preco
instead of update on v_tipoQuartos
referencing new as NEW old as OLD
for each row
begin
    update tipoQuartos
    set precoBase = :NEW.precoBase
    where capacidade = :NEW.capacidade;
end;
/

```

### **Inserts**

#### **/Inserir administradores/**

De início iremos inserir alguns administradores na base de dados, sendo que poderão ser adicionados novos administradores e editadas as suas informações na aplicação.

Exemplo de inserção: insert into v\_administradores values (1, 'Ana Sousa', 123456789, 'G', 'ana.sousa@gmail.com', 'F', 922209090);

#### **/Inserir funcionários/**

De início iremos inserir alguns funcionários na base de dados, sendo que poderão ser adicionados novos funcionários e editadas as suas informações na aplicação.

Exemplo de inserção: insert into v\_funcionarios values (1, 126385724, 'Albertina Silva', 'albertina@gmail.com', 'F', 978365849 );

#### **/Inserir Sector/**

Os sectores serão inseridos também logo no início da criação da base de dados, não será permitida a eliminação de sectores nem a criação de novos, logo o hotel irá sempre ter os mesmos sectores ao longo do tempo.

Exemplo de inserção: insert into Sectors values ('L','Limpeza');

#### **/Inserir trabalhaEm/**

Os funcionários com o respectivo nif são associados a um sector que contém um supervisor para cada um deles. Na aplicação poderemos despedir ou contratar um funcionário para vários sectores.

Exemplo de inserção: insert into trabalhaEm values (126385724, 'L', 123456789);

#### **/Inserir tipoQuartos/**

Os tipos de quartos serão inseridos e determinados logo no início da criação da base de dados e não serão permitidas novas inserções de tipos de quartos nem remoções dos mesmos, portanto o hotel terá sempre os mesmos tipos de quartos.

Exemplo de inserção: insert into tipoQuartos values ('T0',2, 100);

#### **/Inserir facilidades/**

As facilidades serão inseridas e determinadas logo no início da criação da base de dados e não serão permitidas novas inserções de facilidades nem remoções das mesmas, portanto o hotel terá sempre as mesmas facilidades.

Exemplo de inserção: insert into facilidades values ('F','Fumador');

#### **/Inserir quartos/**

Os quartos serão inseridos e determinados logo no início da criação da base de dados e não serão permitidas novas inserções de quartos nem remoções dos mesmos, portanto o hotel terá sempre os mesmos quartos.

Exemplo de inserção: insert into quartos values (1,'T0',2);

#### **/Inserir temFacilidades/**

Cada quarto poderá ter várias facilidades, e no início iremos fazer algumas inserções de facilidades nos quartos, sendo que posteriormente na aplicação poderemos retirar ou inserir novas facilidades.

Exemplo de inserção: insert into temFacilidades values(1,'F');

#### **/Inserir manutenção/**

Cada quarto terá que ter uma manutenção realizada por um funcionário, de forma a manter os quartos sempre acessíveis. No início da aplicação iremos ter alguns registos de manutenções já inseridas e poderão ser adicionadas novas ao longo do tempo através da aplicação.

Exemplo de inserção: insert into manutencao values (seq\_manutencao.nextval, to\_date('2017-06-01', 'YYYY-MM-DD'), 1, 126385724);

#### **/Inserir reservas/**

Serão inseridas algumas reservas no início da aplicação e portanto serão também criados novos clientes na base de dados. Posteriormente na aplicação poderão ser inseridas novas reservas e consequentemente clientes.

Exemplo de inserção: insert into v\_reservas values (1, 2, 6, 100, to\_date('2017-06-01', 'YYYY-MM-DD'), to\_date('2017-06-02', 'YYYY-MM-DD'), 122222229, 'Catarina Silva', 'catarina@gmail.com', 'F', 967854354, 'Portugal');

### **/Inserir checkout/**

Após uma reserva feita por um cliente este poderá apresentar um checkout numa data dando uma certa classificação. Iremos inserir no início já alguns checkouts referentes às reservas existentes e posteriormente podem ser inseridos novos checkouts de reservas ao longo do tempo na aplicação.

Exemplo de inserção: insert into checkOut values (to\_date('2017-06-02', 'YYYY-MM-DD'), 1, 5, 137597504, seq\_checkOut.nextval);

## **Atribuição de Privilégios**

```
grant insert on v_reservas to BD47406;  
grant select on v_tipoQuartos to BD47406;
```

```
grant select on v_funcionarios to BD47981;  
grant insert on v_checkout to BD47981;  
grant select on v_clientes to BD47981;  
grant select on v_facilidadesquarto to BD47981;  
grant insert, select on v_manutencao to BD47981;  
grant select on v_quartos to BD47981;  
grant select, insert on v_reservas to BD47981;  
grant select on v_trabalhaEm to BD47981;
```

O user BD47406 é um cliente e portanto só consegue inserir reservas e ter acesso aos vários tipos de quarto.

O user BD47981 é um funcionário e portanto tem acesso aos registos dos vários funcionários, clientes, facilidades dos quartos, manutenções, quartos, reservas e trabalhaEm. Para além disso, pode inserir um checkout, manutenção e reserva.

# Manual de Utilizador Da Aplicação APEX

Existem 9 tabs onde é possível consultar e manipular todas as tabelas existentes na base de dados. Começamos pelos Administradores, onde na página principal é possível consultar e editar os administradores existentes e os seus atributos correspondentes.

Numad	Nome	Nif	Cargo	Email	Sexo	Telefone
1	Ana Sousa	123456789	G	ana.sousa@gmail.com	F	922209090
2	Joao Teixeira	134567899	SG	joateixeira@gmail.com	M	967845323
3	Mafalda Camilo	145678901	P	maf.camilo@gmail.com	F	956827468
4	Miguel Pereira	156789012	G	miguel.pereira@gmail.com	M	937563837
5	Claudio Jesus	167890123	SG	claudio.jesus@gmail.com	M	947563846
6	Barbara Cruz	178901234	P	barbara.cruz@gmail.com	F	9143645736

Para editar administradores é possível alterar os campos: Nome, Cargo, Email, Sexo e Telefone.

Para inserir administradores novos basta clicar no botão "create" e depois é necessário preencher os campos: Nome, Nif, Cargo, Email, Sexo, Telefone.

Inserir Administradores

\*Nome

\*Nif

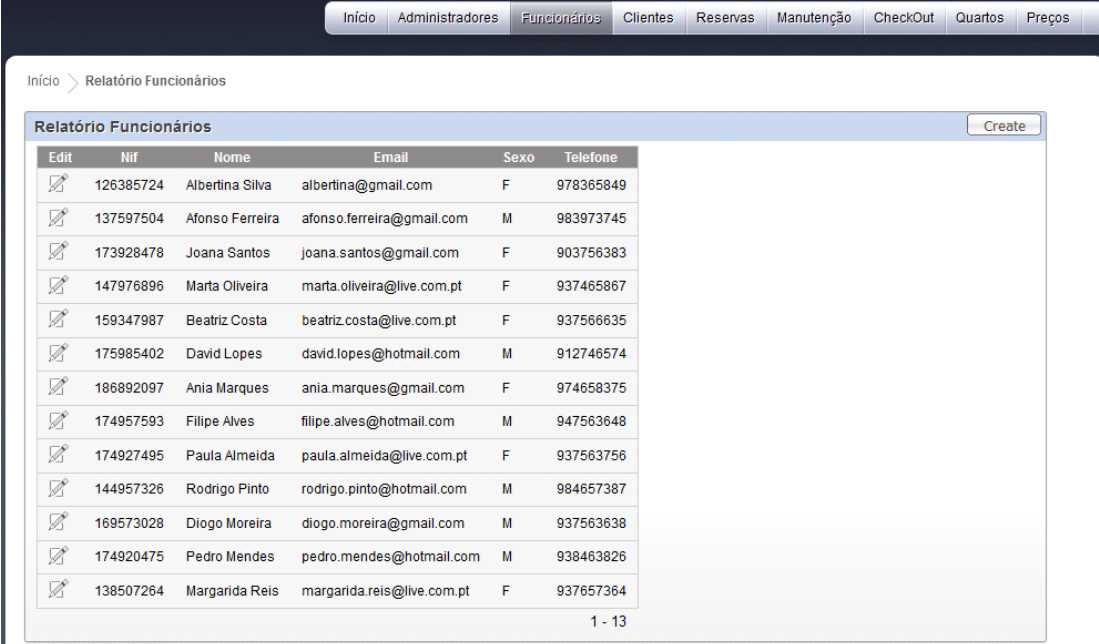
\*Cargo Gerente ▼

\*Email

\*Sexo Feminino ▼

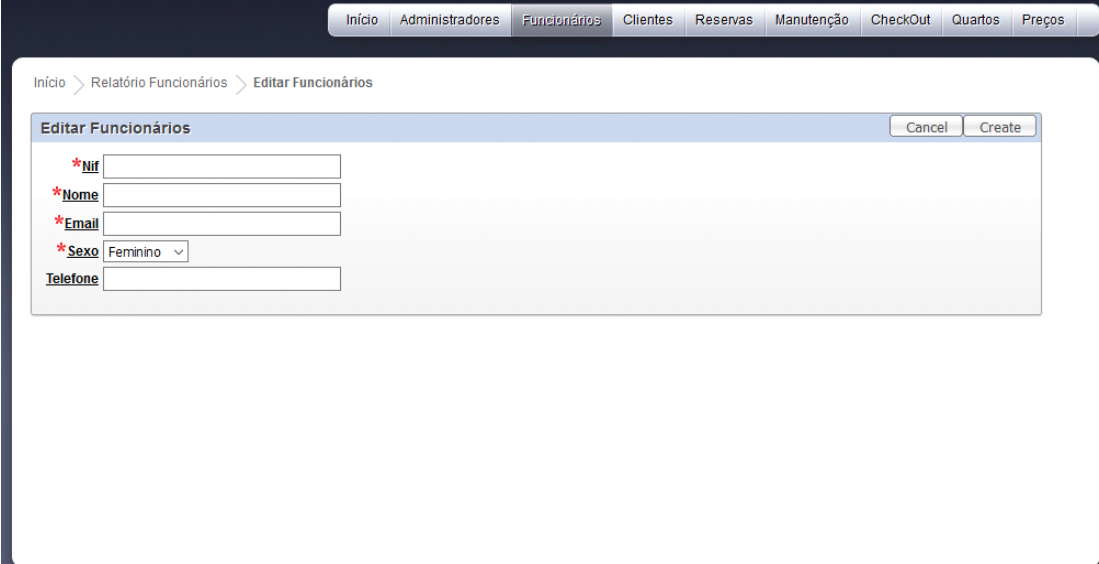
Telefone


Na tab Funcionários temos o registo dos Funcionários, onde é possível consultar, editar e inserir funcionários.




Edit	Nif	Nome	Email	Sexo	Telefone
	126385724	Albertina Silva	albertina@gmail.com	F	978365849
	137597504	Afonso Ferreira	afonso.ferreira@gmail.com	M	983973745
	173928478	Joana Santos	joana.santos@gmail.com	F	903756383
	147976896	Marta Oliveira	marta.oliveira@live.com.pt	F	937465867
	159347987	Beatriz Costa	beatriz.costa@live.com.pt	F	937566635
	175985402	David Lopes	david.lopes@hotmail.com	M	912746574
	186892097	Ania Marques	ania.marques@gmail.com	F	974658375
	174957593	Filipe Alves	filipe.alves@hotmail.com	M	947563648
	174927495	Paula Almeida	paula.almeida@live.com.pt	F	937563756
	144957326	Rodrigo Pinto	rodrigo.pinto@hotmail.com	M	984657387
	169573028	Diogo Moreira	diogo.moreira@gmail.com	M	937563638
	174920475	Pedro Mendes	pedro.mendes@hotmail.com	M	938463826
	138507264	Margarida Reis	margarida.reis@live.com.pt	F	937657364

Para inserir novos Funcionários basta clicar no botão "create" e preencher os campos: Nif, Nome, Email, Sexo, Telefone.



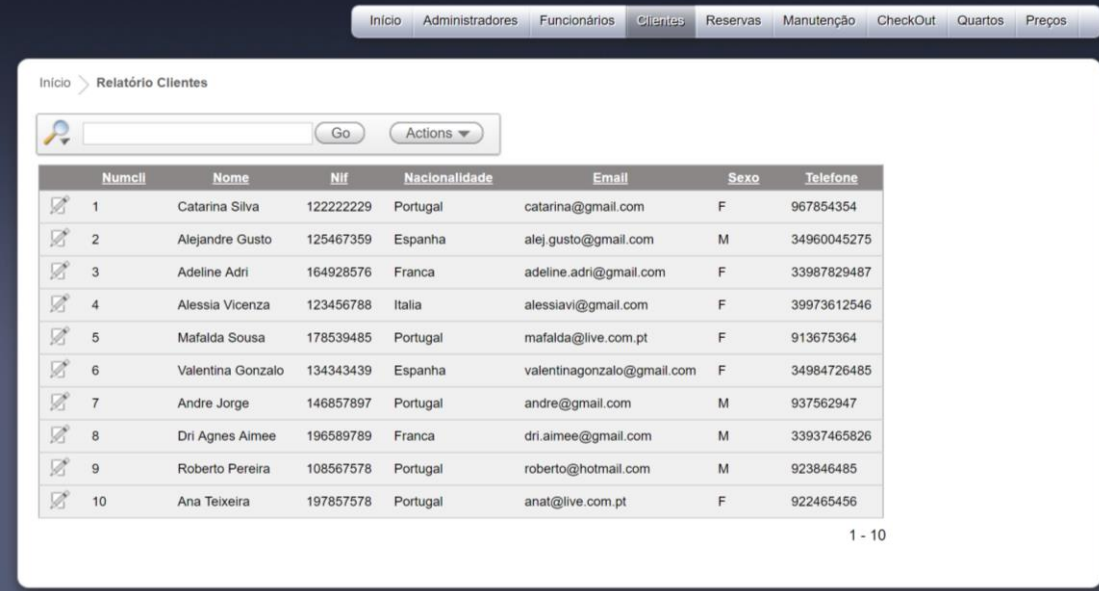
Caso queira editar um Funcionário basta clicar no icon  no lado esquerdo da respetiva linha e dentro dessa tab é possível consultar, inserir, editar e eliminar a relação que existe de um dado funcionário com um dado sector, ou seja, empregar um funcionário a um sector ou despedi-lo desse sector.



1 of 13

1 - 2

Na tab Clientes é possível consultar e editar a informação dos clientes existentes.



Numcli	Nome	Nif	Nacionalidade	Email	Sexo	Telefone
1	Catarina Silva	122222229	Portugal	catarina@gmail.com	F	967854354
2	Alejandro Gusto	125467359	Espanha	alej.gusto@gmail.com	M	34960045275
3	Adeline Adri	164928576	Franca	adeline.adri@gmail.com	F	33987829487
4	Alessia Vicenza	123456788	Italia	alessiavi@gmail.com	F	39973612546
5	Mafalda Sousa	178539485	Portugal	mafalda@live.com.pt	F	913675364
6	Valentina Gonzalo	134343439	Espanha	valentinagonzalo@gmail.com	F	34984726485
7	Andre Jorge	146857897	Portugal	andre@gmail.com	M	937562947
8	Dri Agnes Aimee	196589789	Franca	dri.aimee@gmail.com	M	33937465826
9	Roberto Pereira	108567578	Portugal	roberto@hotmail.com	M	923846485
10	Ana Teixeira	197857578	Portugal	anat@live.com.pt	F	922465456

1 - 10

Caso queira editar cada Cliente, é possível alterar os campos: Nome, Nacionalidade, Email, Sexo, Telefone.

Welcome: UBD40 Logout

Início | Administradores | Funcionários | **Cientes** | Reservas | Manutenção | CheckOut | Quartos | Preços

Início > Relatório Clientes > Editar Clientes

Editar Clientes

Cancel

Apply Changes

\*Nome

Catarina Silva

\*Nif

122222229

Nacionalidade

Portugal

\*Email

catarina@gmail.com

\*Sexo

Feminino

Telefone











967854354

Na tab Reservas é possível consultar e criar as reservas realizadas no Hotel, como também o somatório de todos os preços das reservas em "Total dinheiro das reservas".

Início | Administradores | Funcionários | Clientes | **Reservas** | Manutenção | CheckOut | Quartos | Preços

Início > Relatório Reservas

Go Actions Create

	Número	Cap	Quarto	Preço	Entrada	Saida	Nif	Nome	Email	Sexo	Telefone	Nacionalidade
	10	8	8	400	2017-06-16	2017-06-17	197857578	Ana Teixeira	anat@live.com.pt	F	922465456	Portugal
	9	2	6	200	2017-05-31	2017-06-02	108567578	Roberto Pereira	roberto@hotmail.com	M	923846485	Portugal
	8	4	5	200	2017-06-02	2017-06-03	196589789	Dri Agnes Aimee	dri.aimee@gmail.com	M	33937465826	Franca
	7	4	4	400	2017-06-02	2017-06-04	146857897	Andre Jorge	andre@gmail.com	M	937562947	Portugal
	6	6	9	300	2017-06-02	2017-06-03	134343439	Valentina Gonzalo	valentinagonzalo@gmail.com	F	34984726485	Espanha
	5	6	3	900	2017-06-01	2017-06-04	178539485	Mafalda Sousa	mafalda@live.com.pt	F	913675364	Portugal
	4	4	4	200	2017-06-01	2017-06-02	123456788	Alessia Vicenza	alessiavi@gmail.com	F	39973612546	Italia
	3	2	1	300	2017-06-03	2017-06-06	164928576	Adeline Adri	adeline.adri@gmail.com	F	33987829487	Franca
	2	4	2	600	2017-06-01	2017-06-04	125467359	Alejandro Gusto	alej.gusto@gmail.com	M	34960045275	Espanha
	1	2	1	100	2017-06-01	2017-06-02	122222229	Catarina Silva	catarina@gmail.com	F	967854354	Portugal

Total dinheiro das reservas: 3600

1 - 1

1 - 10



Ao criar cada reserva, é possível preencher os campos: Número Pessoas, DataEntrada, DataSaida, NifCli, Nome, Email, Sexo, Telefone, Nacionalidade.

Início > Relatório Reservas > Inserir Reservas

Inserir Reservas Cancel Create

\*Numero Pessoas 2 ▼

\*DataEntrada

\*DataSaida

\*Nifcli

\*Nome

\*Email

\*Sexo Feminino ▼

Telefone

Nacionalidade

Na tab da Manutenção é possível consultar e criar as manutenções do Hotel, em Nquarto é possível consultar os quartos que se pretende através do seu número. Por exemplo Nquarto está a 1 logo conseguimos ver todas as manutenções realizadas nesse quarto mostrando os campos Nmanutenção, Data Manutencao, Quarto, Nif Funcionário, Nome.

Início > Relatório Manutenção

NQuarto 1 ▼

Go Actions Create

	Nmanutencao	Data Manutencao	Quarto	Nif Funcionário	Nome
	3	2017-05-03	1	126385724	Albertina Silva

1 - 1

Ao criar cada manutenção, é possível preencher os campos: Data Manutenção, Quarto, Nif.

Initial > Administradores > Funcionários > Clientes > Reservas > Manutenção > CheckOut > Quartos > Preços

Initial > Relatório Manutenção > Inserir Manutenção

Inserir Manutenção

Cancel

Create

\*Data Manutenção

\*Quarto

1

Nif

126385724

Na tab CheckOut é possível consultar e criar checkouts feitos no Hotel.








Initial > Administradores > Funcionários > Clientes > Reservas > Manutenção > CheckOut > Quartos > Preços

Initial > Relatório CheckOut

Go

Actions

Create

Ncheckout	Reserva	Classificação	Nif Cliente	Data Checkout	Nif Funcionário
 7	7	4	146857897	2017-06-04	137597504
 6	6	4	134343439	2017-06-03	174927495
 5	5	5	178539485	2017-06-04	174927495
 4	4	5	123456788	2017-06-02	137597504
 3	3	3	164928576	2017-06-06	137597504
 2	2	4	125467359	2017-06-04	174927495
 1	1	5	122222229	2017-06-02	137597504

1 - 7

Ao criar cada CheckOut, é possível preencher os campos: Nreserva, Classificação, Nif Cliente, Data Checkout, Nif Funcionário (sendo que este só pode trabalhar na receção).

Na tab Quartos, é possível consultar os quartos e editar as facilidades de um dado quarto do Hotel. Com o filtro auxiliar "CodigoQ" podemos seleccionar o tipo de quarto e no canto inferior esquerdo mostra-nos a quantidade desse tipo de quarto que existe no Hotel. Por exemplo com CodigoQ a T0 existem 4 quartos desse tipo.

Edit	Numero Quarto	Tipo	Piso
	1	T0	2
	2	T1	2
	3	T2	2
	4	T1	2
	5	T1	2
	6	T0	3
	7	T1	3
	8	T3	3
	9	T2	3
	10	T2	3
	11	T0	3
	12	T2	3
	13	T1	3
	14	T1	3
	15	T1	4

row(s) 1 - 15 of 20 Next

Número De Quartos:  
4

1 - 1

Ao editar cada quarto, só podemos visualizar os campos de um quarto que são os seguintes: NumeroQuarto, Tipo e Piso. Podemos também adicionar, apagar e editar as facilidades que existem nesse dado quarto com o campo CodigoFacilid.

Initial > Administradores > Funcionários > Clientes > Reservas > Manutenção > CheckOut > Quartos > Preços

Initial > Relatório Quartos > Detalhes Quarto

V Quartos

\*Numeroquarto 1

Tipo T0

\*Piso 2

1 of 20

V Facilidadesquarto Detail

Apagar Facilidade

Adicionar Facilidade

☐

Codigoofacilid

☐

Fumador

☐

Hidromassagem

☐

Minibar

1 - 3

Por fim, na tab Preços podemos consultar e editar os preços referenciados de cada tipo de quarto, tendo os campos Tipo, Capacidade e Preço Base.

Initial > Relatório Tipo Quartos

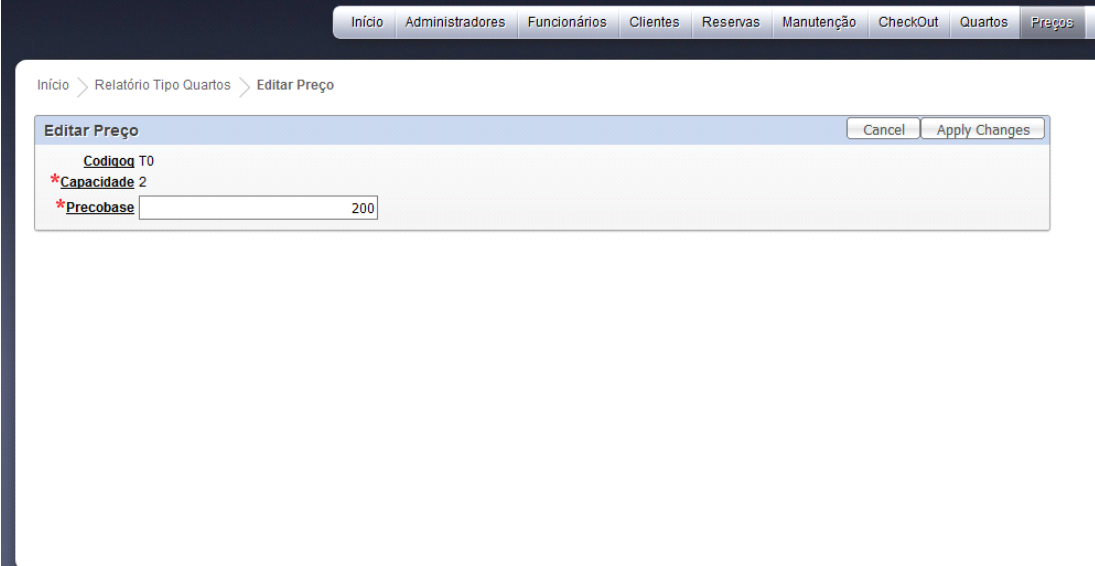
Go

Actions

	Tipo	Capacidade	Preço Base
	T0	2	200
	T1	4	200
	T2	6	300
	T3	8	400

1 - 4

Ao editar cada preço, é possível apenas editar o campo Preço base.



Início > Relatório Tipo Quartos > Editar Preço

**Editar Preço** Cancel Apply Changes

Codigo T0

\*Capacidade 2

\*Preço base 200