

Base de Dados

Versão preliminar do esquema da Base de Dados

27/04/2014

**José Daniel Horta nº 32006
João Diogo Lemos nº 42013
Tiago Alexandre Ferreira nº 42087
Turno Prático 2**

ÍNDICE

1	Nova Surf Camp.....	2
2	Introdução e objectivos da Base de Dados	4
3	Modelo Entidade-Relação	5
4	Inserção de dados SQL	7

1 NOVA SURF CAMP

A Nova Surf Camp é uma escola de surf que aposta na prática e no ensino do surf, atendendo a alunos de todas as faixas etárias, de ambos os sexos e de todos os níveis.

Situada na Costa da Caparica – Portugal, esta escola conta com a ajuda de instrutores certificados pela Federação Portuguesa de Surf de modo a proporcionar a melhor experiência de Surf nas 11 praias que esta Costa disponibiliza.

Para além de ser uma escola de Surf, o Surf Camp é também um *hostel* que conta com 12 quartos. Dos quais 9 com wc privados e 3 com wc partilhados. Uma sala de convívio com *tv*, jogos de tabuleiro, filmes, computadores, consolas para todos os gostos e *wi-fi* grátis.

Para além do ensino, o Surf Camp também dispõe de um conjunto de serviços tais como: alojamento no nosso *hostel*, actividades ao ar livre, passeios turísticos e encomenda de pranchas.

Para simplificar, organizámos os serviços do surf camp em pacotes de acordo com a seguinte tabela:

PACKAGES SurfCamp + Hostel <u>Prices Per person</u>		Weekend package <u>SurfCamp + Hostel</u> (Includes Welcome dinner, 2X Nights, 2X Breakfast, 1X Surf Lesson and 2X Days of gear renting, beach transfer and insurance(ONLY APPLICABLE FOR SURF LESSONS))	Let's Surf package <u>SurfCamp + Hostel</u> (Includes Welcome dinner, 7X Nights, 7X Breakfast, 5X Surf Lesson, Equipment (Wetsuit + Board), beach transfer and insurance(ONLY APPLICABLE FOR SURF LESSONS))	Surf trip package <u>SurfCamp + Hostel</u> (Includes Welcome dinner, 7X Nights, 7X Breakfast, 3X Surf Lesson, 2X Day Surfari, Equipment (Wetsuit + Board), beach transfer and insurance(ONLY APPLICABLE FOR SURF LESSONS))	Beginner Surfer Pack SurfCamp + Hostel (Includes 7X Nights in a 6 bed shared dorm, with breakfast, 5X Surf Lesson including gear and transportation to the beach) Note: A 14 night package also available
Low Season Oct 14 – Dec 27 Jan 2 – May 31	DOUBLE	€ 115	€ 455	€ 505	€ 359 (7 NIGHTS)
	SHARED	€ 95	€ 385	€ 435	
Medium Season June 1 – June 25 Sept 17 – Oct 13	DOUBLE	€ 125	€ 495	€ 545	€ 399 (7 NIGHTS)
	SHARED	€ 115	€ 420	€ 470	
High Season June 26 – Sept 16 Dec 28 – Jan 1	DOUBLE	€ 145	€ 565	€ 615	€499 (7 NIGHTS)
	SHARED	€ 125	€ 495	€ 545	

Caso o cliente pretenda ter apenas aulas de Surf os preços são os seguintes:

SURF LESSONS PRICES	
Group Surf Lessons Duration: 2Hours	€ 25 per person
Private Surf Lessons Individual/ 1 ON 1 Duration: 2 Hours	€ 75 per person
Surf Session or Lesson With Photographer High resolution photos supplied at end of session Qualified Photographer	Group Price - €50 per session Individual Price - € 20 per hour

Todo o material necessário para praticar surf é disponibilizado pelo Surf Camp.

2 INTRODUÇÃO E OBJECTIVOS DA BASE DE DADOS

Este trabalho tem por objectivo criar uma base de dados de gestão do nosso Surf Camp, que permite um cliente praticar uma actividade individual ou um pacote com direito a estadia e equipamento pela parte do Surf Camp. Simplificámos o modelo inicial não atendendo a parte das sessões fotográficas nem reservas em grupo.

Na nossa base de dados serão armazenados dados relativos a:

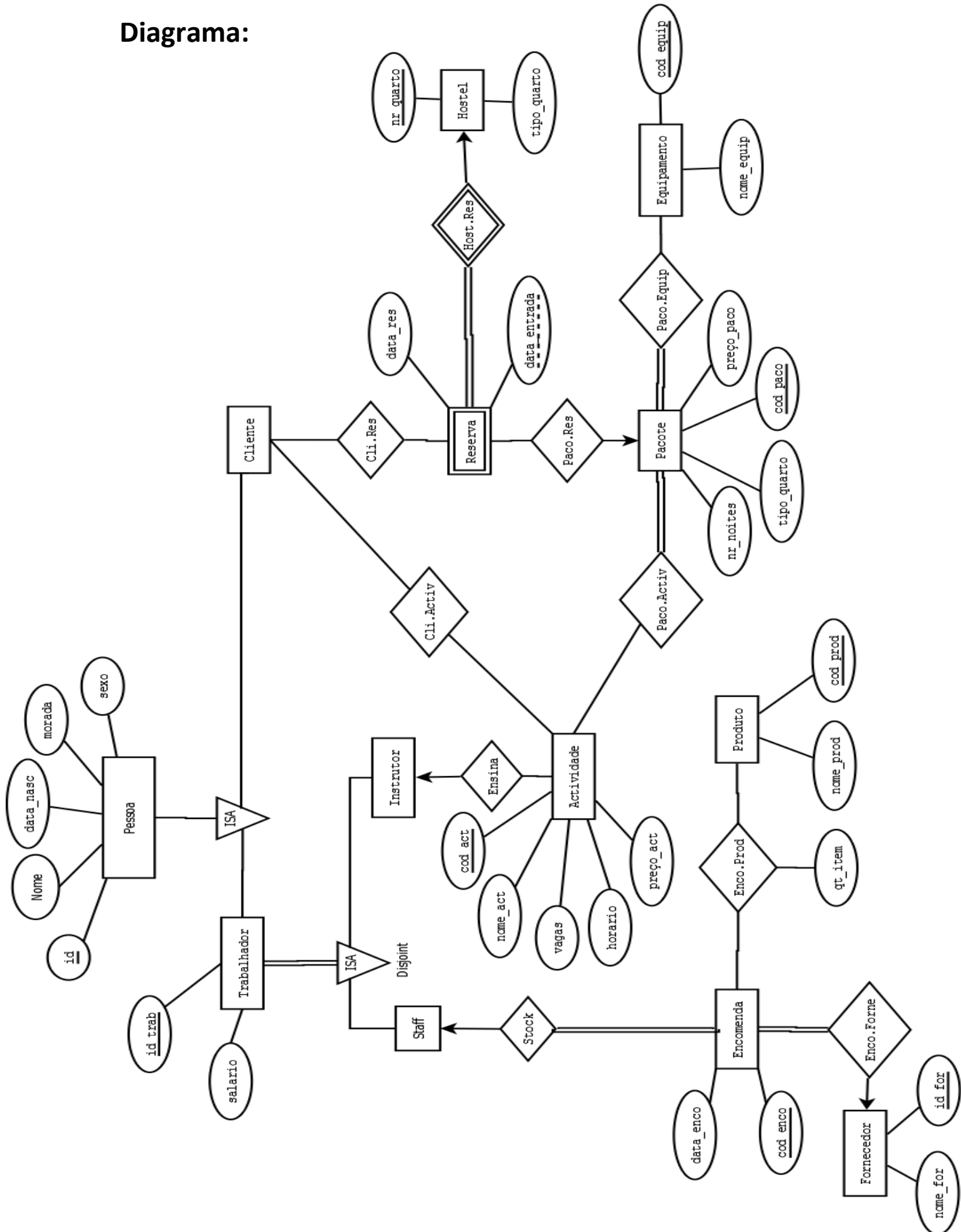
- **Reserva** : armazenando os dados relativamente às reservas de quartos, data de entrada e data da reserva
- **Hostel** : armazenando o identificador do quarto e tipo, se é partilhado ou duplo
- **Pacote** : armazenando o número de noites incluído no pacote, o preço, o código que identifica o pacote, o tipo de quarto e o código do equipamento disponibilizado
- **Equipamento** : armazenando o nome do equipamento e o seu código identificativo
- **Encomenda**: armazenando os dados de uma encomenda para fazer stock de produtos para o hostel ou para o equipamento disponibilizado para as actividades contendo a sua data de encomenda, numero de encomenda, id identificativo do membro que encomenda assim como o seu código de trabalhador e o código do fornecedor
- **Produto**: armazenando o nome do produto, código identificativo do produto
- **Encomenda_Produto** : relação que contém a quantidade de itens a serem encomendados o código do produto e o número de encomenda que está associado
- **Actividade**: armazenando o código da actividade, o seu nome, as vagas disponíveis, o preço da actividade e o seu horário
- **Pessoas**: armazenando todos os clientes e trabalhadores da Surf Camp com os respectivos dados, ID, nome, morada, data de nascimento e sexo
- **Cliente** : armazenando os clientes que são uma generalização de Pessoas
- **Trabalhador** : que são uma generalização de Pessoas contendo o respectivo salario e ID de trabalhador
- **Instrutor**: uma generalização de trabalhador que ensina/apoia os clientes na prática das actividades
- **Staff**: generalização de trabalhador que trata da parte das limpezas, manutenção e stock de produtos

O **objectivo** deste projecto é desenvolver de raiz uma base de dados funcional, ao longo deste projecto vamos obter as bases necessárias à concepção, construção e análise de bases de dados relacionais.

Nesta fase é necessário saber modelar o modelo Entidade-Relação, o modelo relacional de dados e também saber os fundamentos da linguagem SQL, para realizar o script que cria as tabelas juntamente com as suas restrições.

3 MODELO ENTIDADE-RELAÇÃO

Diagrama:



Passagem Modelo ER para o Modelo Relacional (já simplificado):

Pessoa({id, nome, data_nasc, morada, sexo})

Trabalhador({id, id_trab, salario})

Cliente({id})

Staff({id, id_trab})

Instrutor({id, id_trab})

Encomenda({cod_enco, id, id_trab, id_for, data_enco})

Fornecedor(id_for, nome_for)

Enco.Prod({cod_enco, cod_prod, qt_Item})

Produto({cod_prod, nome_prod})

Actividade({cod_act, id, id_trab, nome_act, vagas, horario, preço_act})

Pacote({cod_paco, preço_paco, tipo_quarto, nr_noites})

Reserva({nr_quarto, data_entrada, cod_paco, data_res});

Hostel({nr_quarto, tipo_quarto})

Equipamento({id_equip, nome_equip})

Cli.Activ ({cod_act, id});

Paco.Activ ({cod_act, cod_paco});

Cli.Res ({id, nr_quarto, data_entrada});

Paco.Equip ({cod_paco, cod_equip});

4 INSERÇÃO DE DADOS SQL

```
drop table Pessoa cascade constraints;
create table Pessoa(
  id number(9) not null,
  nome varchar2(40) not null,
  morada varchar2(40) not null,
  data_nasc date not null,
  sexo char(1) not null check(sexo in('F','M')),
  primary key(id)
);
```

```
drop table Cliente cascade constraints;
create table Cliente(
  id number(9) not null,
  primary key(id),
  foreign key (id) references Pessoa
);
```

```
drop table Trabalhador cascade constraints;
create table Trabalhador(
  id number(9) not null,
  id_trab number(9) not null,
  salario number(5) not null,
  primary key(id, id_trab),
  foreign key(id) references Pessoa
);
```

```
drop table Staff cascade constraints;
create table Staff(
  id number(9) not null,
  id_trab number(9) not null,
  primary key(id, id_trab),
  foreign key(id, id_trab) references Trabalhador(id,id_trab)
);
```

```
drop table Instrutor cascade constraints;
create table Instrutor(
  id number(9) not null,
  id_trab number(9) not null,
  primary key(id, id_trab),
  foreign key(id, id_trab) references Trabalhador(id,id_trab)
);
```



```
drop table Produto cascade constraints;
create table Produto(
cod_prod number(9) not null,
nome_prod varchar2(40) not null,
primary key(cod_prod)
);
```

```
drop table Encomenda cascade constraints;
create table Encomenda(
cod_enco number(9) not null,
data_enco date not null,
id number(9) not null,
id_trab number(9) not null,
id_for number(9) not null,
primary key(cod_enco),
foreign key(id) references Pessoa,
foreign key(id, id_trab) references Trabalhador(id,id_trab)
);
```

```
drop table Fornecedor cascade constraints;
create table Fornecedor(
id_for number(9) not null,
nome_for varchar2(40) not null,
primary key(id_for)
);
```

```
drop table Actividade cascade constraints;
create table Actividade(
cod_act number(9) not null,
nome_act varchar2(40) not null,
vagas number(9) not null,
horario number(6) not null,
preco_act number(7) not null,
id number(9) not null,
id_trab number(9) not null,
primary key(cod_act),
foreign key(id, id_trab) references Trabalhador(id,id_trab)
);
```

```
drop table Pacote cascade constraints;
create table Pacote(
cod_paco number(9) not null,
preco_paco number(7) not null,
tipo_quarto char(1) not null check(tipo_quarto in('D','S')),
nr_noites number(2) not null,
primary key(cod_paco)
);
```

```

drop table Reserva cascade constraints;
create table Reserva(
nr_quarto number(4) not null,
data_entrada date not null,
cod_paco number(9) not null,
data_res date not null,
primary key(nr_quarto, data_entrada),
foreign key (nr_quarto) references Hostel,
foreign key (cod_paco) references Pacote
);

```

```

drop table Hostel cascade constraints;
create table Hostel(
nr_quarto number(4) not null,
tipo_quarto char(1) not null check(tipo_quarto in('D','S')),
primary key(nr_quarto)
);

```

```

drop table Equipamento cascade constraints;
create table Equipamento(
cod_equip number(4) not null,
nome_equip varchar2(40) not null,
primary key(cod_equip)
);

```

```

drop table EncoProd cascade constraints;
create table EncoProd(
cod_prod number(9) not null,
cod_enco number(9) not null,
qt_item number(4) not null,
primary key(cod_prod, cod_enco),
foreign key(cod_prod) references Produto,
foreign key(cod_enco) references Encomenda
);

```

```

drop table CliActiv cascade constraints;
create table CliActiv (
cod_act number(9) not null,
id number(9) not null,
primary key(cod_act,id),
foreign key(cod_act) references Actividade,
foreign key(id) references Cliente
);

```

```

drop table PacoActiv cascade constraints;
create table PacoActiv(
cod_act number(9) not null,
cod_paco number(9) not null,
qt_item number(4) not null,
primary key(cod_act, cod_paco),
foreign key(cod_act) references Actividade,
foreign key(cod_paco) references Pacote
);

```

```

drop table CliRes cascade constraints;
create table CliRes(
id number(9) not null,
data_entrada date not null,
nr_quarto number(4) not null,
primary key(id, data_entrada,nr_quarto),
foreign key(id) references Cliente,
foreign key(data_entrada,nr_quarto) references Reserva(data_entrada,nr_quarto)
);

```

```

drop table PacoEquip cascade constraints;
create table PacoEquip(
cod_paco number(9) not null,
cod_equip number(9) not null,
primary key(cod_paco, cod_equip),
foreign key(cod_paco) references Pacote,
foreign key(cod_equip) references Equipamento
);

```

```

drop sequence seq_Pessoa;
create sequence seq_Pessoa increment by 1 start with 1;
drop sequence seq_Trabalhador;
create sequence seq_Trabalhador increment by 1 start with 1;
drop sequence seq_Cliente;
create sequence seq_Cliente increment by 1 start with 1;
drop sequence seq_Staff;
create sequence seq_Staff increment by 1 start with 1;
drop sequence seq_Instrutor;
create sequence seq_Instrutor increment by 1 start with 1;
drop sequence seq_Encomenda;
create sequence seq_Encomenda increment by 1 start with 1;
drop sequence seq_Fornecedor;
create sequence seq_Fornecedor increment by 1 start with 1;
drop sequence seq_EncoProd;
create sequence seq_EncoProd increment by 1 start with 1;
drop sequence seq_Produto;
create sequence seq_Produto increment by 1 start with 1;
drop sequence seq_Actividade;

```

```
create sequence seq_Actividade increment by 1 start with 1;
drop sequence seq_Pacote;
create sequence seq_Pacote increment by 1 start with 1;
drop sequence seq_Reserva;
create sequence seq_Reserva increment by 1 start with 1;
drop sequence seq_Hostel;
create sequence seq_Hostel increment by 1 start with 1;
drop sequence seq_Equipamento;
create sequence seq_Equipamento increment by 1 start with 1;
drop sequence seq_CliActiv;
create sequence seq_CliActiv increment by 1 start with 1;
drop sequence seq_PacoActiv;
create sequence seq_PacoActiv increment by 1 start with 1;
drop sequence seq_CliRes;
create sequence seq_CliRes increment by 1 start with 1;
drop sequence seq_PacoRes;
create sequence seq_PacoRes increment by 1 start with 1;
```