

# MODELAÇÃO CONCEPTUAL

## 1. FÓRMULA 1

Pretende-se armazenar a informação relativa a uma época do campeonato de Fórmula 1.

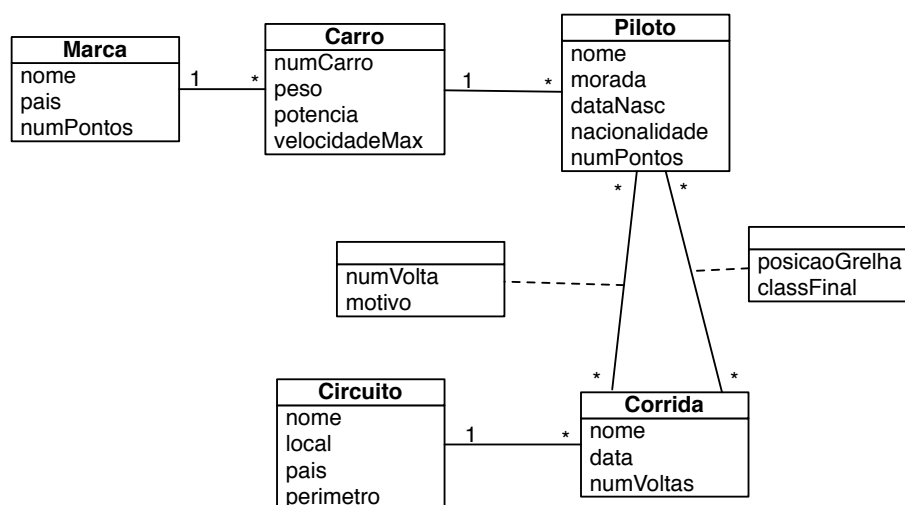
De cada marca participante no campeonato pretende-se armazenar o seu nome, país de origem, nº atual de pontos no campeonato de marcas e quais os carros inscritos. De cada carro interessa saber o seu peso, potência e velocidade máxima.

Relativamente aos pilotos participantes é necessário conhecer o seu nome, morada, idade, nacionalidade e nº atual de pontos no campeonato de pilotos. Um piloto só pode conduzir um carro ao longo da época, embora um determinado carro possa ser conduzido por mais de um piloto. Esta situação, embora não muito frequente, pode surgir, por exemplo, devido ao afastamento de um piloto ferido num acidente.

Uma época é constituída por um conjunto de corridas que se realizam em circuitos e em datas definidas no início da época. Para uma determinada corrida pode-se, ao longo da época e por razões várias, alterar-se o circuito onde esta se realiza. Em situações excepcionais pode acontecer também a realização de duas corridas no mesmo circuito. Cada corrida tem um número determinado de voltas no circuito onde é realizada. De cada circuito pretende-se saber o nome, local, país e perímetro.

No que diz respeito à realização de uma corrida interessa saber quais os pilotos que participaram, as posições que ocuparam na grelha de partida e a classificação final. Relativamente à corrida interessa também saber quais os pilotos que desistiram, em que volta ocorreu e qual o motivo da desistência. [Baseado num exercício de António Brito]

i. Modelo conceptual



ii. Esquema relacional

Marca (idMarca, nome, pais, numPontos)

Carro (idCarro, numCarro, peso, potencia, velocidadeMax, idMarca -> Marca)

Piloto (idPiloto, nome, morada, idade, nacionalidade, numPontos, idCarro -> Carro)

Circuito (idCircuito, nome, local, pais, perimetro)

Corrida (idCorrida, nome, data, numVoltas, idCircuito->Circuito)

Participacao (idPiloto -> Piloto, idCorrida -> Corrida, posicaoGrelha, classifFinal)

Desistencia (idPiloto -> Piloto, idCorrida -> Corrida, numVolta, motivo)

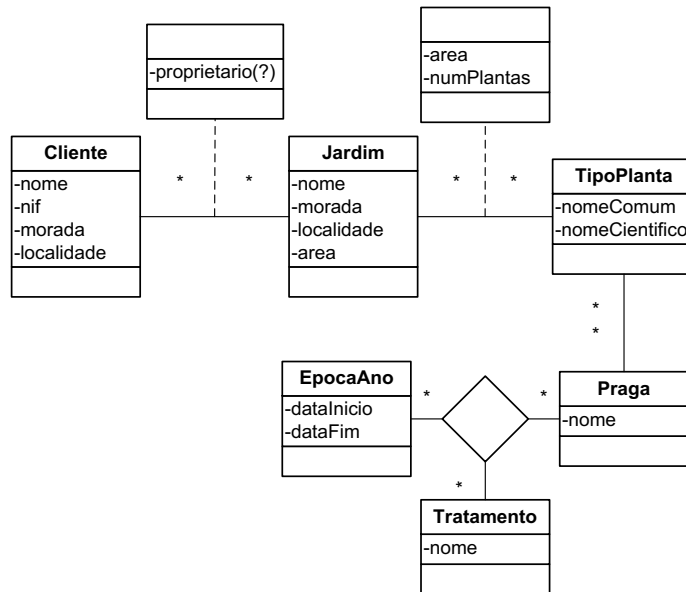
## 2. JARDINEIRO

Um jardineiro tem a seu cargo a manutenção dos jardins dos seus clientes. Assim, de cada cliente quer saber o seu nome, NIF, morada e localidade. Estes dados são necessários para efeitos de faturação.

De cada jardim é necessário saber o nome, morada, localidade, área (em hectares), o cliente responsável por esse jardim e saber se ele é proprietário do jardim ou apenas o explora. De cada jardim é necessário saber os tipos de plantas que lá se encontram, área ocupada por cada um desses tipos de planta e número de plantas de cada tipo. Sobre os tipos de planta guarda-se o nome pelo qual é conhecido (por exemplo: laranjeira) e o respetivo nome científico (por exemplo: Citrus sinensis).

Para cada tipo de planta interessa ainda saber as pragas que as afetam (por exemplo, as laranjeiras podem ter cochonilha). As épocas do ano caracterizam-se por uma data de início e uma data de fim. É também necessário saber que tratamentos são eficazes para o tratamento de cada praga numa dada época, considerando que uma praga pode afetar uma planta em várias épocas. [Baseado num exercício de João Mendes Moreira]

i. Modelo conceptual



## ii. Esquema relacional

Cliente (idCliente, nome, nif, morada, localidade)

Jardim (idJardim, nome, morada, localidade, area)

Proprietario (idCliente -> Cliente, idJardim -> Jardim, proprietario?)

TipoPlanta (idTipoPlanta, nomeComum, nomeCientifico)

JardimPlantas (idJardim -> Jardim, idTipoPlanta -> TipoPlanta, area, numPlantas)

Praga (idPraga, nome)

PragaTipoPlanta (idPraga -> Praga, idTipoPlanta -> TipoPlanta)

Tratamento (idTratamento, nome)

EpocaAno (idEpocaAno, dataInicio, dataFim)

PragaTratamentoEpoca (idPraga -> Praga, idTratamento -> Tratamento, idEpocaAno -> EpocaAno)

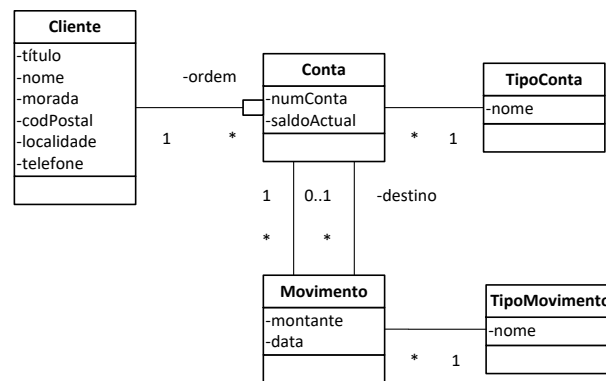
## 3. BANCO MUITOSJUROS

Construa um modelo conceptual para modelar a realidade do banco MuitosJuros, tendo em conta que:

- os clientes têm um título, nome, morada, código postal, localidade e telefone;
- as contas estão associadas a um número de conta, um tipo de conta e ao saldo atual;
- podem ser efetuados movimentos de vários tipos e montantes;
- cada movimento deve estar associado a uma data;

- v. cada conta pode ter um ou mais titulares com ordem diferente (p. ex.: 1º e 2º titular);
- vi. interessa saber quais são os primeiros titulares de uma dada conta, as datas em que uma conta teve movimentos e quem são os clientes que possuem uma conta de um dado tipo. [Exercício de Carla Teixeira Lopes]

i. Modelo conceptual



ii. Esquema relacional

Cliente (idCliente, titulo, nome, morada, codPostal, localidade, telefone)

TipoConta (idTipoConta, nome)

Conta (idConta, numConta, saldoActual, idTipoConta -> TipoConta)

Titular (idConta -> Conta, ordem, idCliente -> Cliente)

TipoMovimento (idTipoMovimento, nome)

Movimento (idMovimento, montante, data, idTipoMovimento -> TipoMovimento, origem -> Conta, destino -> Conta)

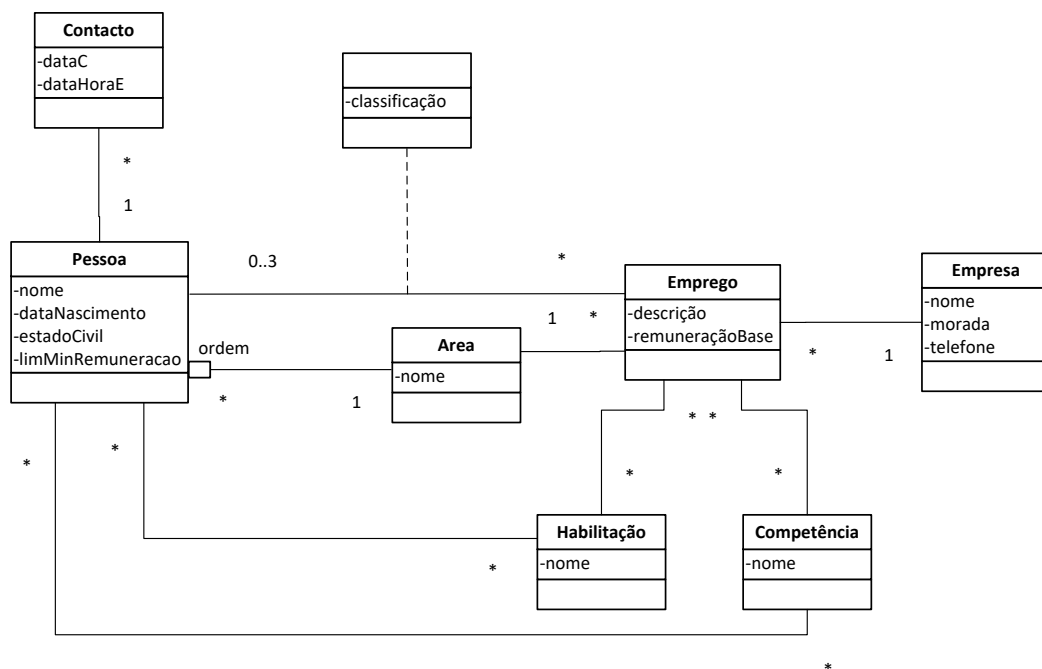
## 4. AGÊNCIA DE EMPREGO

Uma agência de emprego tem como objetivo encontrar o pessoal adequado ao preenchimento de vagas fornecidas pelas empresas. Assim, quando uma empresa quer recrutar pessoas, tem de fornecer à agência a área de trabalho a que a oferta de emprego se refere, uma descrição desse trabalho, habilitações e competências necessárias, remuneração base e indicação da empresa proponente. Da empresa é necessário saber o nome, morada e telefone.

Quando uma pessoa recorre à agência para procurar emprego tem de dar o nome, data de nascimento, estado civil, habilitações, competências, limite mínimo de remuneração e uma lista com as áreas de trabalho ordenada de acordo com a preferência da pessoa. As habilitações, as competências e as áreas de trabalho definidas tanto pelas empresas como pelas pessoas são seleccionadas de uma lista definida pela agência.

O sistema informático deverá atribuir automaticamente pessoas à procura de emprego às ofertas feitas pelas empresas. Assim, para cada oferta, são atribuídas três pessoas que são contactadas para comparecerem a uma entrevista individual. O sistema deve registar a data do contacto, assim como a data e hora da entrevista. No caso de a pessoa ter sido seleccionada para mais do que uma oferta de emprego, só irá a uma entrevista. Nas entrevistas é atribuída uma classificação por cada um dos empregos a que essa pessoa tenha sido atribuída. [Exercício de João Mendes Moreira]

#### i. Modelo conceptual



#### ii. Esquema relacional

Pessoa (idPessoa, nome, dataNascimento, estadoCivil, limMinRemuneracao)

Contacto (idContacto, dataC, dataHoraE, idPessoa -> Pessoa)

Empresa (idEmpresa, nome, morada, telefone)

Area (idArea, nome)

PessoaArea (idPessoa -> Pessoa, ordem, idArea -> Area)

Emprego (idEmprego, descricao, remuneracaoBase, idEmpresa -> Empresa, idArea -> Area)

EmpregoPessoa (idEmprego -> Emprego, idPessoa -> Pessoa, classificacao)

Habilitacao (idHabilitacao, nome)

Competencia (idCompetencia, nome)

PessoaHabilitacao (idPessoa -> Pessoa, idHabilitacao -> Habilitacao)

PessoaCompetencia (idPessoa -> Pessoa, idCompetencia -> Competencia)

EmpregoHabilitacao (idEmprego -> Emprego, idHabilitacao -> Habilitacao)

EmpregoCompetencia (idEmprego -> Emprego, idCompetencia -> Competencia)

## 5. OFICINA

Um concessionário de automóveis pretende informatizar o seu serviço de reparações em oficina. De cada cliente interessa guardar o nome, a morada e o telefone. Interessa também saber a matrícula, a marca e o modelo de cada carro, assim como quem é o cliente proprietário desse carro. Sobre os funcionários da oficina pretende-se guardar o nome, a morada, o telefone, a especialidade desse funcionário e o respetivo custo horário. Sabe-se que o custo horário da mão-de-obra depende da especialidade (por exemplo: os eletricitas têm um preço, os chapeiros outro preço, etc.).

A oficina guarda o registo de todas as peças em armazém. Sobre cada peça sabe-se: o código, a designação, o custo unitário e a quantidade disponível. É ainda necessário saber quais os modelos de automóveis compatíveis com cada peça.

Por cada reparação realizada é guardada informação do veículo, do cliente (por omissão é o proprietário do veículo, mas pode ser outra pessoa qualquer), as datas de início e de fim da reparação, e ainda: (1) todas as peças utilizadas na reparação (como é óbvio, só podem ser utilizadas peças compatíveis com o modelo do automóvel em reparação) e respetiva quantidade; e (2) o nº de horas despendidas por cada funcionário na reparação. [Exercício de João Mendes Moreira]

### i. Modelo conceptual



```
FuncionarioReparacao (idFuncionario -> Funcionario, idReparacao -> Reparacao,
numHoras)
```

## 6. LIGA DE FUTEBOL

A entidade responsável pelos principais campeonatos de futebol de um país à beira mar plantado pretende informatizar os seus serviços. Para tal, pediu ajuda aos alunos da FEUP-MIEIC.

Nessa entidade, designada por liga, estão inscritos clubes. Cada clube tem várias equipas e cada equipa só participa num campeonato em cada época. Um clube pode ter uma equipa por cada classe etária distinta (juvenis, juniores, etc.) e por género (masculino ou feminino). Do mesmo modo há, por época, um campeonato por classe etária e por género. As equipas só podem participar no campeonato da mesma faixa etária e género.

Sobre cada equipa é necessário saber os jogadores inscritos. Em cada época existem dois períodos para a inscrição de jogadores. Em cada momento deve ser possível saber quais os jogadores legalmente inscritos na liga e qual o clube a que cada um está vinculado. Num dado momento um jogador só pode estar vinculado a um clube. A equipa a que o jogador pertence é determinada no início da época pela sua faixa etária e pelo seu género.

No início da época é definido um calendário de jogos para cada campeonato. Sobre cada jogo deve constar: a jornada a que diz respeito, data prevista de realização, as equipas em confronto (visitada e visitante) e o estádio onde se realiza o jogo e que é, por omissão, o estádio do clube visitado, mas pode, por algum motivo, ser outro. A data do jogo pode ser alterada pela liga a pedido dos clubes.

Sobre os jogos é necessário guardar informação sobre os diferentes eventos que ocorrem ao longo do jogo. Esses eventos estão tipificados (ex.: golo, cartão amarelo/vermelho, substituição, etc.). Obviamente que sobre: os cartões disciplinares, é necessário saber a quem foi mostrado; sobre os golos, quem o marcou e a favor de que equipa (é que há golos na própria baliza!); e sobre as substituições, quem entrou e quem saiu. Pode haver outros tipos de eventos que se assume só necessitarem de saber o jogador envolvido. É relevante saber, para todos os eventos, o minuto de jogo em que ocorreu.

Para cada classe defina os atributos adequados. Por exemplo: os clubes têm nome, morada, ano de fundação, etc. [Exercício de João Mendes Moreira]

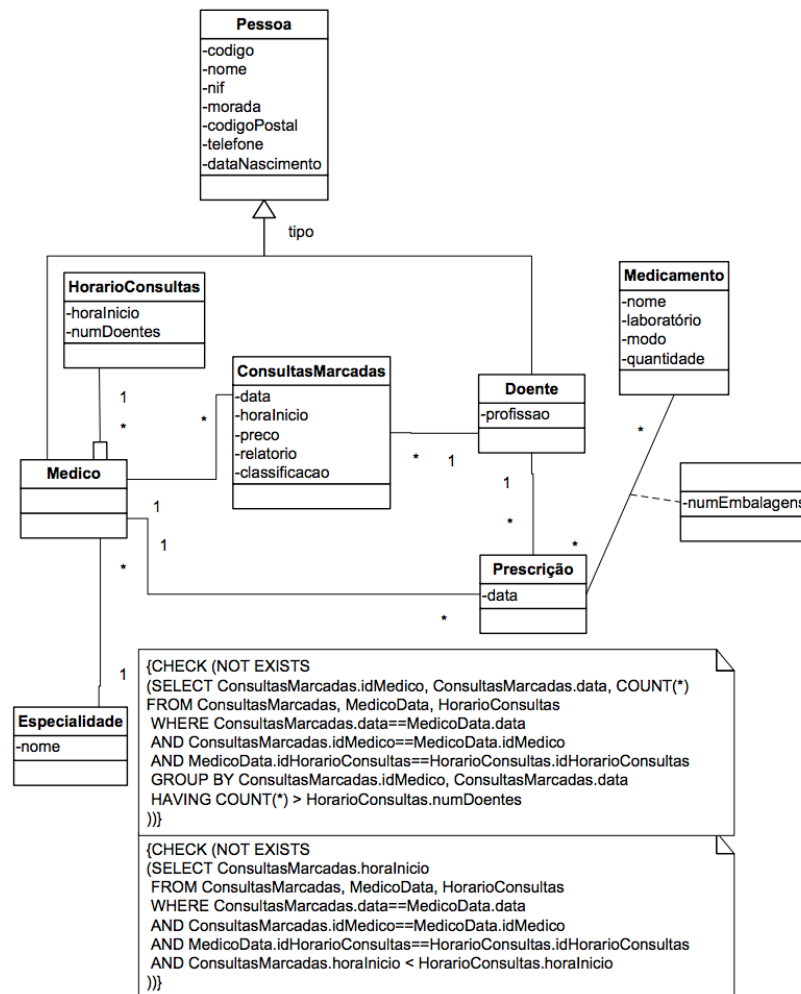
- i. Modelo conceptual





Pretende-se também guardar as prescrições de medicamentos feitas pelos médicos aos doentes. Dos medicamentos interessa saber o nome, laboratório, modo de administração, a quantidade de medicamentos na embalagem e a quantidade de embalagens prescritas. [Baseado num exercício de Gabriel David]

## 9



ii. Esquema relacional

Especialidade (idEspecialidade, nome)

Medico (idMedico, codigo, nome, nif, morada, codigoPostal, telefone, dataNascimento, idEspecialidade -> Especialidade)

Doente (idDoente, codigo, nome, nif, morada, codigoPostal, telefone, dataNascimento)

HorarioConsulta (idHorarioConsulta, horaInicio, numDoentes)

MedicoConsulta (idMedico -> Medico, data, idHorarioConsulta -> HorarioConsulta)

ConsultasMarcadas (idConsultaMarcada, data, horaInicio, preco, relatorio, classificacao, idMedico -> Medico, idDoente -> Doente)

Medicamento (idMedicamento, nome, laboratorio, modo, quantidade)

Prescricao (idPrescricao, data, idMedico -> Medico, idDoente -> Doente)

PrescricaoMedicamento (idPrescricao -> Prescricao, idMedicamento -> Medicamento, numEmbalagens)

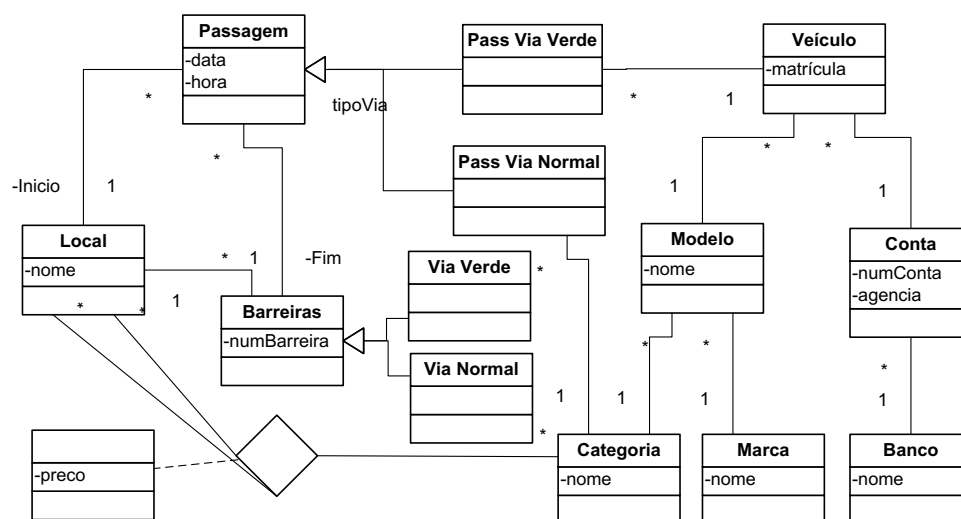
## 8. BRISA

A BRISA quer manter a informação sobre a utilização das suas autoestradas de acordo com:

- as autoestradas existentes vão desde a A1 (Norte) até à A8 (Loures);
- as portagens são cobradas nas barreiras, as quais têm um número próprio e estão implantadas num local;
- o valor da portagem depende da classe do veículo (de 1 a 4) e é aditivo (Porto -> Coimbra = Porto -> Aveiro + Aveiro-> Coimbra) e simétrico;
- interessa saber qual o dia e a hora em que a portagem foi paga, o percurso a que se refere e a matrícula e a marca do respetivo veículo;
- alguns dos veículos são verdes, i.e., têm um dispositivo de identificação automática e é necessário saber qual o número de conta, banco e agência responsável pelos respetivos pagamentos.

NOTA: deve ser possível representar os custos dos percursos independentemente de já ter havido alguém a pagar essa portagem. [Baseado num exercício de Gabriel David e Ana Paiva]

- Modelo conceptual



Só se devem definir os preços entre locais contíguos na auto-estrada para evitar ambiguidades.

Os registos das passagens são efectuados automaticamente através da detecção do sinal da via verde (Pass Via Verde) ou da indicação do portageiro (Pass Via Normal).

- Esquema relacional

Local (idLocal, nome)

Barreiras (idBarreiras, numBarreira, tipoVia, idLocal -> Local)

Categoria (idCategoria, nome)

Preco (idCategoria -> Categoria, origem Local, destino -> Local, preco)

Marca (idMarca, nome)

Modelo (idModelo, nome, idMarca -> Marca, idCategoria -> Categoria)

Banco (idBanco, nome)

Conta (idConta, numConta, agencia, idBanco -> Banco)

Veiculo (idVeiculo, matricula, idModelo -> Modelo, idConta -> Conta)

Passagem (idPassagem, data, hora, inicio -> Barreira, fim -> Barreira)

PassagemVerde (idPassagem -> Passagem, idVeiculo -> Veiculo)

PassagemNormal (idPassagem -> Passagem, idCategoria -> Categoria)

## 9. RED BULL AIR RACE

Pretende-se armazenar informação relativa a uma época da competição Red Bull Air Race. Nesta competição participam pilotos que se organizam em equipas. Cada piloto compete ao longo da época com um único modelo de avião.

Cada piloto é identificado por um número e pretende-se armazenar o seu primeiro e último nome, bem como a sua nacionalidade e data de nascimento. Relativamente à equipa, apenas é necessário saber a sua designação e país de origem. No entanto, cada modelo de avião é caracterizado pela sua potência, velocidade máxima, comprimento, largura e peso.

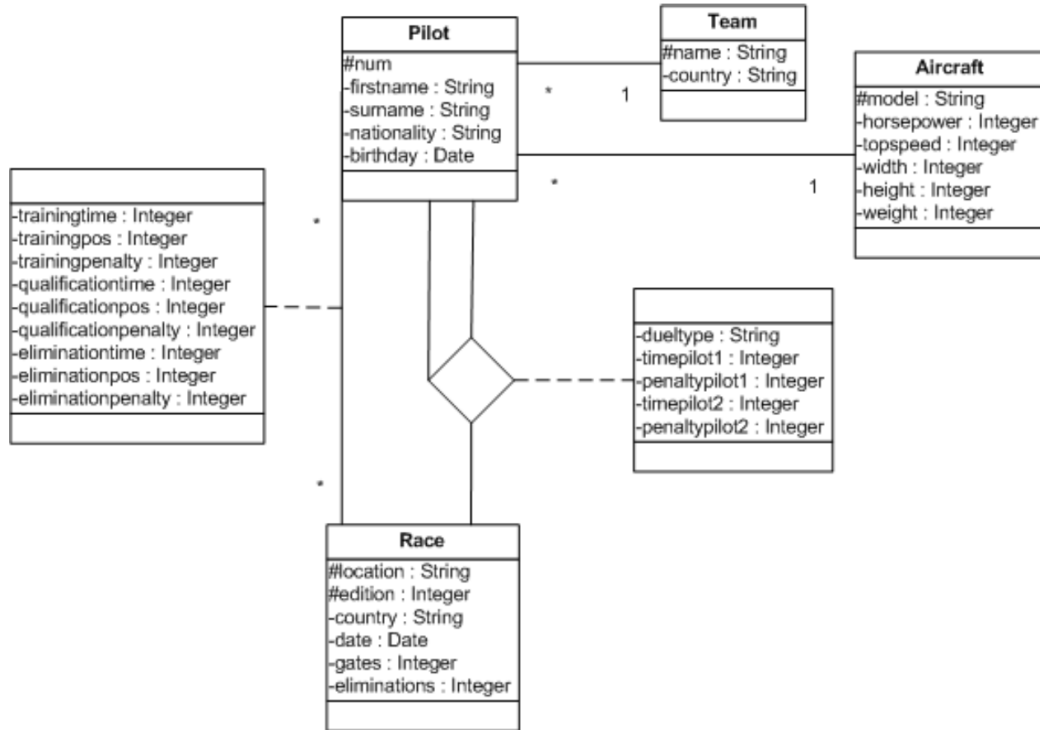
Ao longo da época, os pilotos competem em corridas que se realizam numa determinada cidade, num dado ano. É interessante saber em que país e data cada corrida se realizou bem como o número de pódios da mesma e quantos pilotos são eliminados na primeira fase.

As corridas seguem uma ‘mecânica a dois tempos’. A primeira fase consiste em percorrer o percurso no menor tempo possível, combatendo somente contra o relógio. Esta fase é dividida em etapas: Treino Livre, Qualificação e Eliminação. Para cada etapa é interessante guardar o tempo, a posição relativa de cada piloto bem como os segundos de penalização. Os tempos obtidos na fase de Treino Livre ditam a ordem de saída para a Qualificação e a ordem desta para a Eliminação. O piloto com o pior tempo da Qualificação não segue para a Eliminação e os quatro piores da Eliminação não seguem para a Segunda Fase.

A Segunda Fase é uma etapa de Duelo na qual os pilotos competem diretamente 2 a 2 desde os Quartos-de-Final até à Final. É importante guardar a informação relativa a quem participou em que duelo, em que fase, com que tempo e com que penalidade.

O sistema de pontuação é linear, atribuindo 6 pontos ao vencedor até 1 ponto ao 6º classificado. [Baseado num exercício de Gabriel David, Vasco Vinhas e André Restivo]

## i. Modelo conceptual



## ii. Esquema relacional

Team (name, country)

Aircraft (model, horsepower, topspeed, width, height, weight)

Pilot (num, firstname, surname, nationality, birthday, name -> Team, model -> Aircraft)

Race (location, edition, country, date, gates, eliminations)

Participation (num -> Pilot, [location, edition] -> Race, trainingtime, trainingpos, trainingpenalty, qualificationtime, qualificationpos, qualificationpenalty)

Duel (num<sub>pilot1</sub> -> Pilot, num<sub>pilot2</sub> -> Pilot, [location, edition] -> Race, duetype, timepilot1, timepilot2, penaltypilot1, penaltypilot2)