

Faculdade De Engenharia Universidade Do Porto
Departamento de Engenharia Informática

Aeroporto

Bases de Dados (BDAD)
Mestrado Integrado em Engenharia Informática e Computação

Fábio Lino: 201206080
Gil Teixeira: 201505735
Pedro Tavares: 201406991

Turma: 5 Grupo: 507
Professor: Henrique Cardoso

Ano Letivo: 2016-2017

ÍNDICE

Aeroporto	v
1 Introdução	v
2 Contexto	v
3 Modelo Conceptual	vi
4 Esquema Relacional	vii
5 Dependências Funcionais e Formas Normais	vii
6 Restrições à Base de Dados	viii

AEROPORTO

1

Introdução

Este relatório tem como objetivo descrever o trabalho realizado no âmbito da disciplina de Bases de Dados, chamado "Aeroporto". O objetivo deste trabalho é criar uma base de dados para um sistema de regulação de voos num dado aeroporto.

2

Contexto

Pretende-se então guardar a informação sobre um voo, para isso criamos a classe Voo que está associada a um único Avião, a pelo menos um Funcionário e poderá estar também associada a Passageiros. Um dado Avião está associado a um Modelo, a uma Companhia e poderá ter um Hangar ou não. Um Passageiro que pretenda então viajar passa a estar associado a um Estado para esse Voo. Uma Bagagem está associada a um Passageiro e ao(s) Voo(s) em que esteve presente. Um Funcionário pode pertencer a Secções do Aeroporto, o qual pode ter várias Funções por cada Secção a que pertença. Por fim para generalizar a informação de Passageiro e Funcionário, ambos descendem da classe Pessoa que contém os dados comuns aos dois.

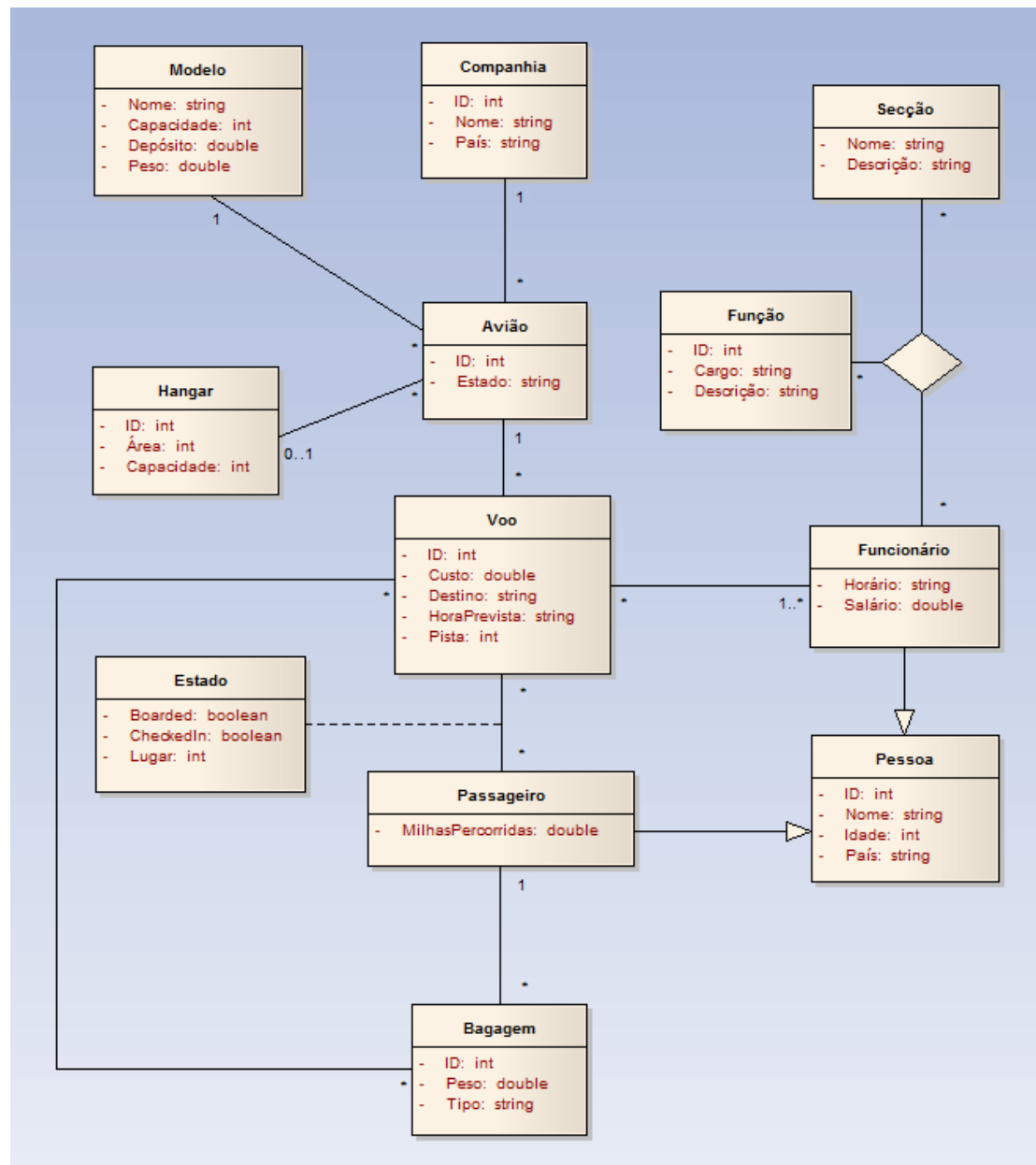


Figura 1: Diagrama UML

Modelo(MNome, Capacidade, Depósito, Peso)
 Hangar(HID, Área, Capacidade)
 Companhia(CID, Nome, País)
 Avião(AID, ModeloNome->Modelo, HangarID->Hangar, CompanhiaID->Companhia, Estado)
 Pessoa(PID, Nome, Idade, País)
 Passageiro(PPID->Pessoa, MilhasPercorridas)
 Bagagem(BID, PassageiroID->Passageiro, Peso, Tipo)
 Transporta(BagagemID->Bagagem, VooID->Voo)
 Funcionário(PFID->Pessoa, Horário, Salário)
 Secção(SNome, Descrição)
 Função(FID, Cargo, Descrição)
 Trabalho(FuncionárioID->Funcionário, SecçãoNome->Secção, FunçãoID->Função)
 Estado(PassageiroID->Passageiro, VooID->Voo, Boarded, CheckedIN, Lugar)
 Voo(VID, AviãoID->Avião, Custo, Destino, HoraPrevista, Pista)

MNome -> Capacidade, Depósito, Peso
 HID -> Área, Capacidade
 CID -> Nome, País
 AID -> ModeloNome, HangarID, CompanhiaID, Estado
 PID -> Nome, Idade, País
 PPID -> MilhasPercorridas BID -> PassageiroID, Peso, Tipo
 PFID -> Horário, Salário
 SNome -> Descrição
 FID -> Cargo, Descrição
 PPID, PFID -> Boarded, CheckedIN, Lugar
 VID -> AviãoID, Custo, Destino, HoraPrevista, Pista

Como cada Dependência funcional tem a chave da relação do lado esquerdo, esta está de acordo com a Forma Normal Boyce-Codd e a 3ª Forma Normal. Assim, não existe qualquer violação a qualquer das formas, pois cada relação tem um identificador único que é chave ou a chave é o conjunto de todos os elementos da relação.

A capacidade, o depósito e o peso de cada Modelo devem ser número positivos.
(CHECK(Capacidade > 0), CHECK(Deposito > 0), CHECK(Peso > 0))

A área e a capacidade de cada Hangar devem ser número positivos.
(CHECK(Area > 0), CHECK(Capacidade > 0))

O nome da Companhia deve ser único e não nulo.
(UNIQUE NOT NULL)

O país da Companhia deve ser não nulo.
(NOT NULL)

O nome do modelo e a companhia de cada Avião não podem ser nulos.
(NOT NULL)

O nome e país de cada Pessoa não podem ser nulos.
(NOT NULL)

A idade de cada Pessoa tem de ser maior ou igual a zero.
(CHECK(Idade >= 0))

As milhas percorridas por cada passageiro têm de ser maior ou igual a zero.
(CHECK(MilhasPercorridas >= 0))

Cada Bagagem tem que ter um passageiro.
(NOT NULL)

O peso de cada Bagagem tem de ser maior ou igual a zero.
(CHECK(Peso >= 0))

O salário de cada funcionário tem de ser maior ou igual a zero.
(CHECK(Salário >= 0))

O Cargo de cada Função não pode ser nulo.
(NOT NULL)

Os valores de Boarded só podem variar entre 0 e 1.
CHECK(Boarded = 0 OR Boarded = 1)

Os valores de CheckedIN só podem variar entre 0 e 1.
CHECK(CheckedIn = 0 OR CheckedIn = 1)

O número de Lugar de cada Estado tem de ser positivo.
(CHECK(Lugar > 0))

Cada voo tem que ter um Avião.
(NOT NULL)

O Custo de cada Voo tem de ser um número positivo.
(CHECK(Custo > 0))

O destino de um Voo não pode ser Nulo.
(NOT NULL)

A número da pista de cada Voo tem de ser maior ou igual a zero.
(CHECK(Pista >= 0))