2018/19



BRUNO FILIPE CARDOSO MICAELO – UP201706044

MIGUEL DELGADO PINTO – UP201706156

PEDRO MIGUEL RODRIGUES FERRAZ ESTEVES – UP201705160

Gestão de Recursos de Biblioteca

Base de Dados

# Introdução

O projeto assenta na criação de uma base dados para auxiliar a gestão de uma biblioteca.

Como qualquer biblioteca, nela podem ser requisitados livros de diversos géneros. Os utilizadores têm também ao seu dispor vários elementos multimédia, como filmes, álbuns de música e software.

Se não pretenderem fazer uma requisição, podem usufruir de salas disponíveis para reserva, equipadas com computadores, projetores, televisões, entre outros.

# Planeamento da Biblioteca

A informação de qualquer indivíduo relacionado com a biblioteca será armazenada na classe **Pessoa**. Estes são caracterizados pelo nome, data de nascimento, número de telemóvel e número de cartão de cidadão.

Os trabalhadores na biblioteca, classe **Funcionário**, derivada de Pessoa, possuem um ID próprio, um salário, hora de entrada e hora de saída e informação complementar, como número de contribuinte e morada.

A classe **Utilizador**, derivada de Pessoa, facilita a organização do sistema e a identificação dos usuários da biblioteca.

Cada utilizador pode realizar uma **Requisição** de um **Exemplar** de uma certa **Publicação**.

As requisições têm um ID, uma data e uma hora caraterísticas. As publicações possuem um ID, um nome, um género e uma idade mínima. É necessário que o exemplar esteja disponível para requisição, uma vez que certos exemplares devem estar permanentemente na biblioteca.

As publicações estão organizadas por classes derivadas do tipo Publicação, que podem ser: **Livro**, **Filme**, **Álbum** ou **Software**.

Um livro é caracterizado por uma editora e pelo número da edição, sendo que está associado a pelo menos um **Autor** (cada autor tem um ID e um nome).

Um filme distingue-se pelo realizador e pelo estúdio.

Um álbum define-se pelo seu produtor.

Note-se que um álbum conta com a participação de pelo menos um **Artista**, sendo este caracterizado por um ID e pelo seu nome.

Um software classifica-se pela sua versão e pelo developer.

Caso um utilizador esteja atrasado na devolução de um exemplar, a classe de associação **Atraso** guarda informação sobre os dias de atraso e a multa.

Os utilizadores podem também usufruir de **Salas** disponíveis para **Reserva**.

As salas são caracterizadas por um número, um tipo de sala e capacidade. Têm também **Equipamento** específico (cada um com um ID, marca e modelo). Para facilitar a organização, a cada equipamento está associado uma classe **TipoEquipamento**. Existe um número limitado de tipos de equipamento: Leitor de CDs, Leitor de DVDs, Leitor de VHS, Computador, Projetor e Televisão, cada um com um propósito específico.

A reserva destas salas tem um ID, um motivo, uma data e hora específicas e uma duração.

Como se trata de uma biblioteca de excelência, os seus bens estão sempre nas melhores condições. Para tal, os funcionários realizam atividades de manutenção (classe **Tipo de Manutenção**) em certos exemplares. São caracterizadas pelo seu nome.

# Diagrama UML

O diagrama UML encontra-se em anexo (no ficheiro **Biblioteca\_UML.jpg**).

# Modelo Relacional

Pessoa(CartaoCidadao, Nome, DataNascimento, Telefone)

CartaoCidadao -> {Nome, DataNascimento, Telefone}

A chave principal da relação Pessoa é CartaoCidadao.

Utilizador(CartaoCidadao -> Pessoa)

A chave principal da relação Utilizador é CartaoCidadao, que serve também como chave estrangeira para a relação Pessoa.

Funcionario(CartaoCidadao -> Pessoa, Salario, Contribuinte, Morada, HoraEntrada, HoraSaida)

CartaoCidadao -> {Contribuinte, Morada, HotaEntrada, HoraSaida}

Tal como para a relação Utilizador, a chave principal é CartaoCidadao, que é também uma chave estrangeira para Pessoa.

Reserva(IDReserva, Motivo, Data, Hora, Duracao, CCutilizador -> Utilizador)

IDReserva -> {Motivo, Data, Hora, Duracao, CCutilizador}

A chave principal para Reserva é IDReserva, tendo também CCutilizador como chave estrangeira para uma relação Utilizador.

Sala(Numero, Tipo, Capacidade)

Numero -> {Tipo, Capacidade}

A chave principal para Sala é Numero.

ReservaDeSala(IDReserva -> Reserva, NumeroSala -> Sala)

A relação ReservaDeSala tem como chave principal a chave composta por IDReserva e NumeroSala, chaves estrangeiras para Reserva e Sala, respetivamente.

Equipamento(IDEquipamento, Marca, Modelo, NumeroSala -> Sala, NomeTipo -> TipoEquipamento)

IDEquipamento -> {Marca, Modelo, NumeroSala, NomeTipo}

Modelo -> {Marca, NomeTipo}

A chave principal para Equipamento é IDEquipamento, tendo NumeroSala e NomeTipo como chaves estrangeiras para Sala e TipoEquipamento, respetivamente.

TipoEquipamento(Nome, Proposito)

Nome -> {Proposito}

A chave principal para TipoEquipamento é Nome.

Requisicao(IDRequisicao, Data, Hora, DiasAtraso, Multa, CCutilizador -> Utilizador)

IDRequisicao -> {Data, Hora, DiasAtraso, Multa, CCutilizador}

A chave principal para Requisicao é IDRequisicao, tendo como chave estrangeira para Utilizador CCutilizador.

Exemplar(IDExemplar, PossivelRequisitar, IDSala -> Sala, IDPublicacao -> Publicacao)

IDExemplar -> {PossivelRequisitar, IDSala, IDPublicacao}

A chave principal para Exemplar é IDExemplar, com IDSala e IDPublicacao como chaves estrangeiras para as relações Sala e Publicacao, respetivamente.

TipoDeManutencao(Nome)

A chave principal para TipoManutencao é Nome.

AtoDeManutencao(CCfuncionario -> Funcionario, NomeManutencao -> TipodeManutencao, IDExemplar -> Exemplar)

AtoDeManutencao tem como chave principal a chave composta por Ccfuncionario, NomeManutencao e IDExemplar, que servem também como chaves estrangeiras para as relações Funcionario, TipodeManutencao e Exemplar.

Publicacao(IDPublicacao, Nome, Genero, IdadeMinima)

IDPublicacao -> {Nome, Genero, IdadeMinima}

A chave principal da relação Publicacao é IDPublicacao.

Livro(IDPublicacao -> Publicacao, Editora, Edicao)

IDPublicacao -> {Editora, Edicao}

A chave principal de Livro é IDPublicacao, que é também uma chave estrangeira para Publicacao.

Autor(IDAutor, Nome)

IDAutor -> {Nome}

A chave principal para Autor é IDAutor.

Autoria(IDPublicacao -> Livro, IDAutor -> Autor)

Autoria tem como chave principal a chave composta por IDPublicacao e IDAutor, que servem também como chaves estrangeiras para as relações Livro e Autor, respetivamente.

Software(IDPublicacao -> Publicacao, Versao, Developer)

IDPublicacao -> {Versao, Developer}

A chave principal de Software é IDPublicacao, que é também uma chave estrangeira para Publicacao.

Album(IDPublicacao -> Publicacao, Produtor)

IDPublicacao -> {Produtor}

A chave principal de Album é IDPublicacao, que é também uma chave estrangeira para Publicacao.

Artista(IDArtista, Nome)

IDArtista -> {Nome}

A relação Artista tem como chave principal IDArtista.

Interpreta(IDPublicacao -> Album, IDArtista -> Artista)

A relação Interpreta tem como chave principal a chave composta por IDPublicacao e IDArtista, também chaves estrangeiras para Album e Artista, respetivamente.

Filme(IDPublicacao -> Publicacao, Realizador, Estudio)

IDPublicacao -> {Publicacao, Realizador, Estudio}

A chave principal de Filme é IDPublicacao, que é também uma chave estrangeira para Publicacao.

# Análise das Relações

De modo a analisar as dependências funcionais de cada relação e as conseguintes formas normais, é necessário analisar os fechos dos lados esquerdos de cada dependência funcional.

Relação Pessoa:

* {CartaoCidadao}+ = {CartaoCidadao, Nome, DataNascimento, Telefone}

Relação Utilizador:

* {CartaoCidadao}+ = {CartaoCidadao}

Relação Funcionario:

* {CartaoCidadao}+ = {CartaoCidadao, Salario, Contribuinte, Morada, HoraEntrada, HoraSaida}

Relação Reserva:

* {IDReserva}+ = {IDReserva, Motivo, Data, Hora, Duracao, CCutilizador}

Relação Sala:

* {Numero}+ = {Numero, Tipo, Capacidade}

Relação ReservaDeSala:

* {IDReserva, NumeroSala}+ = {IDReserva, NumeroSala}

Relação Equipamento:

* {IDEquipamento}+ = {IDEquipamento, Marca, Modelo, NumeroSala, NomeTipo}
* {Modelo}+ = {Modelo, Marca, NomeTipo}

Relação TipoEquipamento:

* {Nome}+ = {Nome, Proposito}

Relação Requisicao:

* {IDRequisicao}+ = {IDRequisição, Data, Hora, DiasAtraso, Multa, CCutilizador}

Relação Exemplar:

* {IDExemplar}+ = {IDExemplar, PossivelRequisitar, IDSala, IDPublicacao}

Relação TipoDeManutencao:

* {Nome}+ = {Nome}

Relação AtoDeManutencao:

* {Ccfuncionario, NomeManutencao, IDExemplar}+ = {Ccfuncionario, NomeManutencao, IDExemplar}

Relação Publicacao:

* {IDPublicacao}+ = {IDPublicaçao, Nome, Genero, IdadeMinima}

Relação Livro:

* {IDPublicacao}+ = {IDPublicacao, Editora, Edicao}

Relação Autor:

* {IDAutor}+ = {IDAutor, Nome}

Relação Autoria:

* {IDPublicacao, IDAutor}+ = {IDPublicacao, IDAutor}

Relação Software:

* {IDPublicacao}+ = {IDPublicacao, Versao, Developer}

Relação Album:

* {IDPublicacao}+ = {IDPublicacao, Produtor}

Relação Artista:

* {IDArtista}+ = {IDArtista, Nome}

Relação Interpreta:

* {IDPublicacao, IDArtista}+ = {IDPublicacao, IDArtista}

Relação Filme:

* {IDPublicacao}+ = {IDPublicacao, Realizador, Estudio}

Tendo em conta a análise dos fechos de cada dependência funcional, pode-se concluir que a maioria destas respeita a Terceira Forma Normal (respeitam a primeira [cada atributo contém apenas valores atómicos] e segunda [os atributos membros de uma chave são primos] formas normais e todos os atributos não primos são dependentes de uma chave de forma não transitiva) e na BCNF (o lado esquerdo de cada dependência corresponde a uma chave). A única exceção é o conjunto de dependências correspondentes à relação Equipamento, na qual Modelo -> {Marca, NomeTipo} corresponde a uma violação da BCNF (Modelo não faz parte de uma chave).

De modo a resolver a violação da BCNF na relação Equipamento, alterou-se a classe Equipamento, passando os atributos Modelo, Marca e NomeTipo para uma classe nova Modelo, cuja chave principal é o nome do modelo (tendo em conta que este é primo). Resumidamente, o modelo relacional muda na seguinte forma:

Equipamento(IDEquipamento, NumeroSala -> Sala, NomeModelo -> Modelo)

IDEquipamento -> {NumeroSala, NomeModelo}

Modelo(NomeModelo, Marca, NomeTipo -> TipoEquipamento)

NomeModelo -> {Marca, NomeTipo}

Nesta forma, não existem violações da BCNF (e consequentemente das formas normais das quais é um super set [Primeira, Segunda e Terceira formas normais]).