

Licenciatura em Engenharia Informática  
Departamento de Eletrónica, Telecomunicações e Informática  
Universidade de Aveiro  
Bases de Dados  
2020/2021

# **Projeto Final**

## Base de Dados de uma Biblioteca

Andreia de Sá Portela, 97953  
Miguel Rocha Ferreira, 98599

## Índice

Introdução	3
I - Análise de requisitos	4
II - DER	5
III - Esquema Relacional da BD	6
IV - Diagrama da Base de Dados	7
V - Normalização	8
VI - Interfaces	9
VII - SQL DDL em SQL Server	10
VIII - SQL DML em SQL Server	11
Entregáveis	13

## Introdução

Para o nosso projeto final de Bases de Dados, decidimos criar uma base de dados (em SQL) para uma Biblioteca Municipal, assim como uma interface de gestão de dados (realizada em C#) com o intuito de aplicar os conhecimentos adquiridos ao longo do semestre.

Este sistema seria utilizado em bibliotecas de forma a poder gerir todos os seus itens, as suas reservas, os seus membros e os seus funcionários, assim como tratar de transações entre os membros e a biblioteca.

Este relatório divide-se em 6 partes:

- Análise de Requisitos
- DER
- Esquema Relacional
- Normalização
- SQL DDL
- SQL DML

Todas as transações descritas neste trabalho poderão ser visualizadas usando a interface gráfica *LibraryBD* disponível no GitHub, implementada utilizando a ferramenta Visual Studio.

Usamos os slides disponibilizados pelos professores, os guiões práticos e a documentação SQL e Visual Studio encontrada na Internet.

## **I - Análise de requisitos**

A biblioteca tem um nome, uma morada e um número de telefone. Ela disponibiliza vários itens como livros, filmes, CDs, jornais, revistas e periféricos a membros da biblioteca.

Um livro tem um autor, um título, um código ISBN, uma editora, um ano de publicação e um género.

Um filme tem um título, um realizador, um género e um ano de lançamento.

Um CD tem um título, um artista, um género e um ano de lançamento.

Ambos o jornal e a revista tem uma marca, o tipo(diário, semanal, mensal...), a data de lançamento e a edição.

Cada produto disponibilizado pela biblioteca tem uma seção associada: infantil, juvenil ou adulta.

Um periférico tem um tipo e uma marca.

Cada item tem um ID único dentro da biblioteca.

Os membros da biblioteca têm que ter nome, morada, mail, ID de membro e NIF.

A biblioteca tem um ou mais funcionários caracterizados por nome, data de nascimento, morada, email, ID de funcionário, NIF, SSN, data de início de contrato e data de fim de contrato . Um desses funcionários é o gerente da biblioteca e são definidas também as datas de início e fim da posição de gerente (não pode haver dois gerentes em simultâneo).

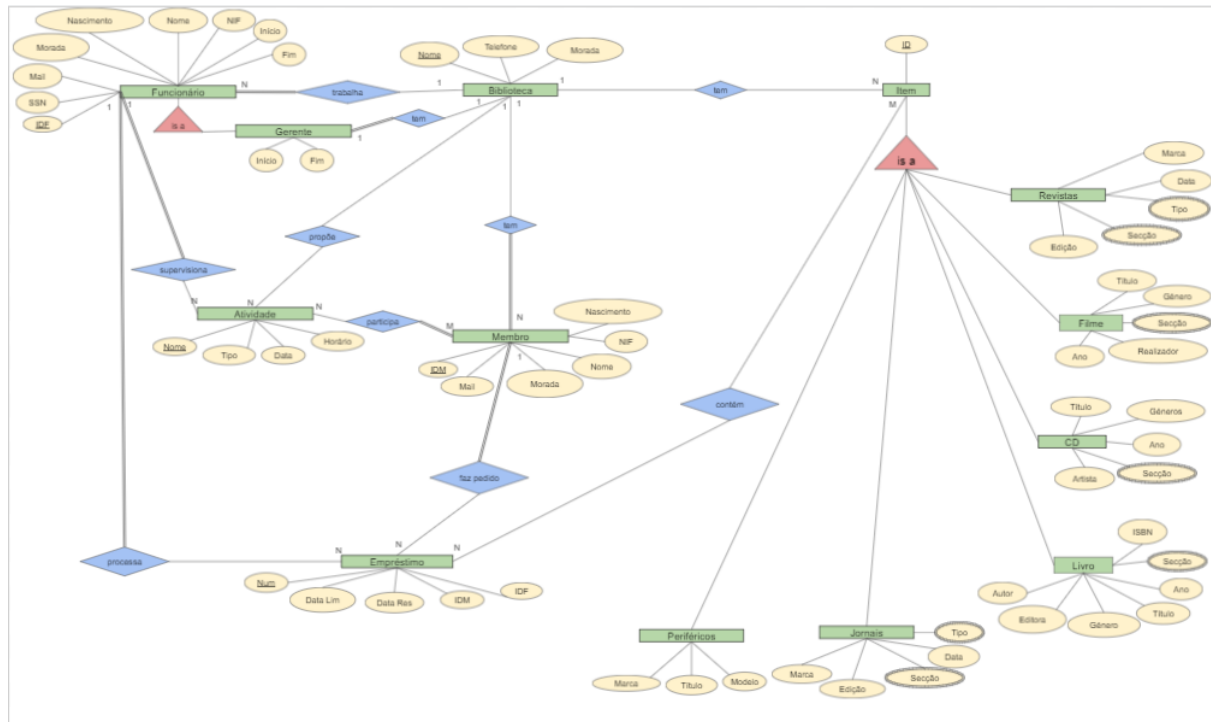
Os empréstimos têm um número de pedido, o id do membro que reservou, o id do funcionário que efetuou a reserva e as datas do empréstimo e de entrega.

A biblioteca propõe várias atividades. Os membros podem participar nas atividades que quiserem desde que não sejam no mesmo dia. Cada atividade tem uma data, um horário e um tipo (filme, leitura, teatro, etc). Apenas um funcionário é responsável por um tipo de atividade.

## II - DER

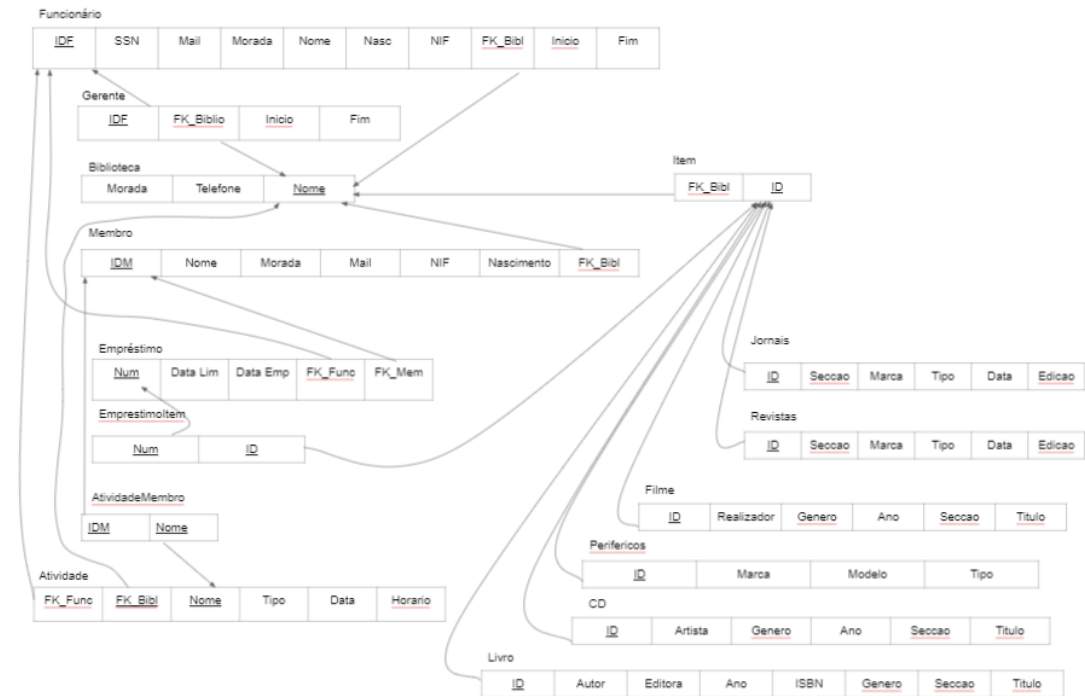
Após reunir os requisitos, construímos o Diagrama Entidade Relação correspondente.

Esta versão do DER é uma versão final do mesmo. Este DER sofreu várias alterações desde o início do semestre, especialmente após a reunião com o professor, nomeadamente a adição das entidades ‘Atividades’ e ‘Item’, entre outras.

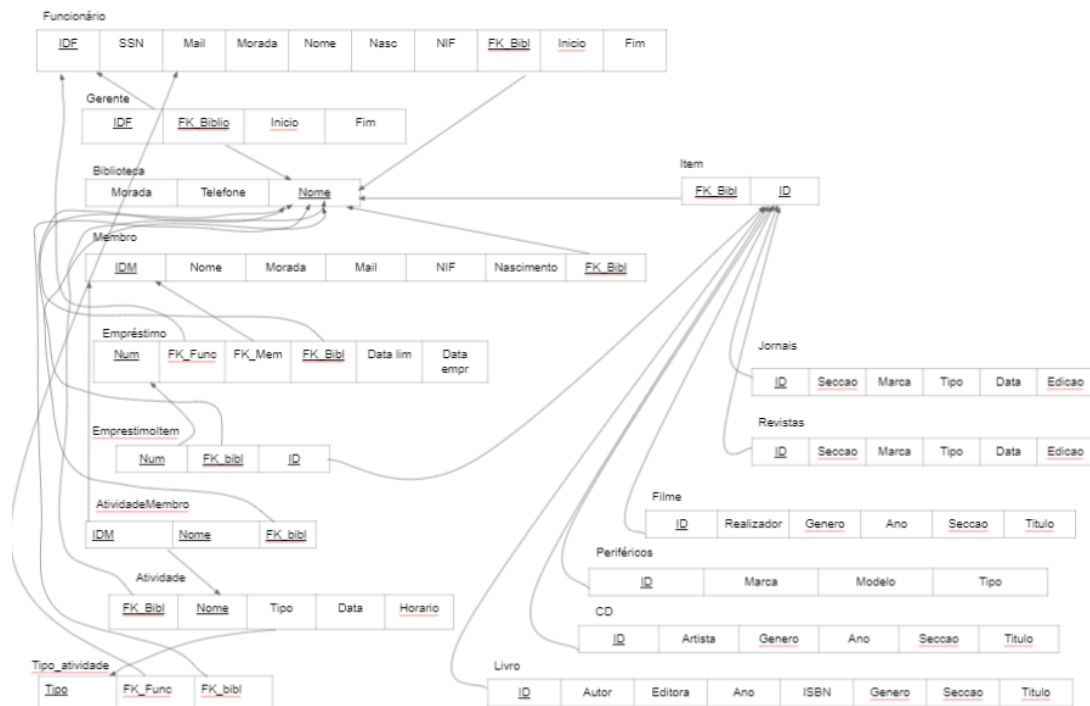


### III - Esquema Relacional da BD

O diagrama representado no ponto anterior, traduziu-se no seguinte esquema relacional. Este diagrama, tal como o diagrama anterior, também foi sofrendo alterações ao longo do semestre. Esta versão reflete as mudanças que efetuamos no DER.

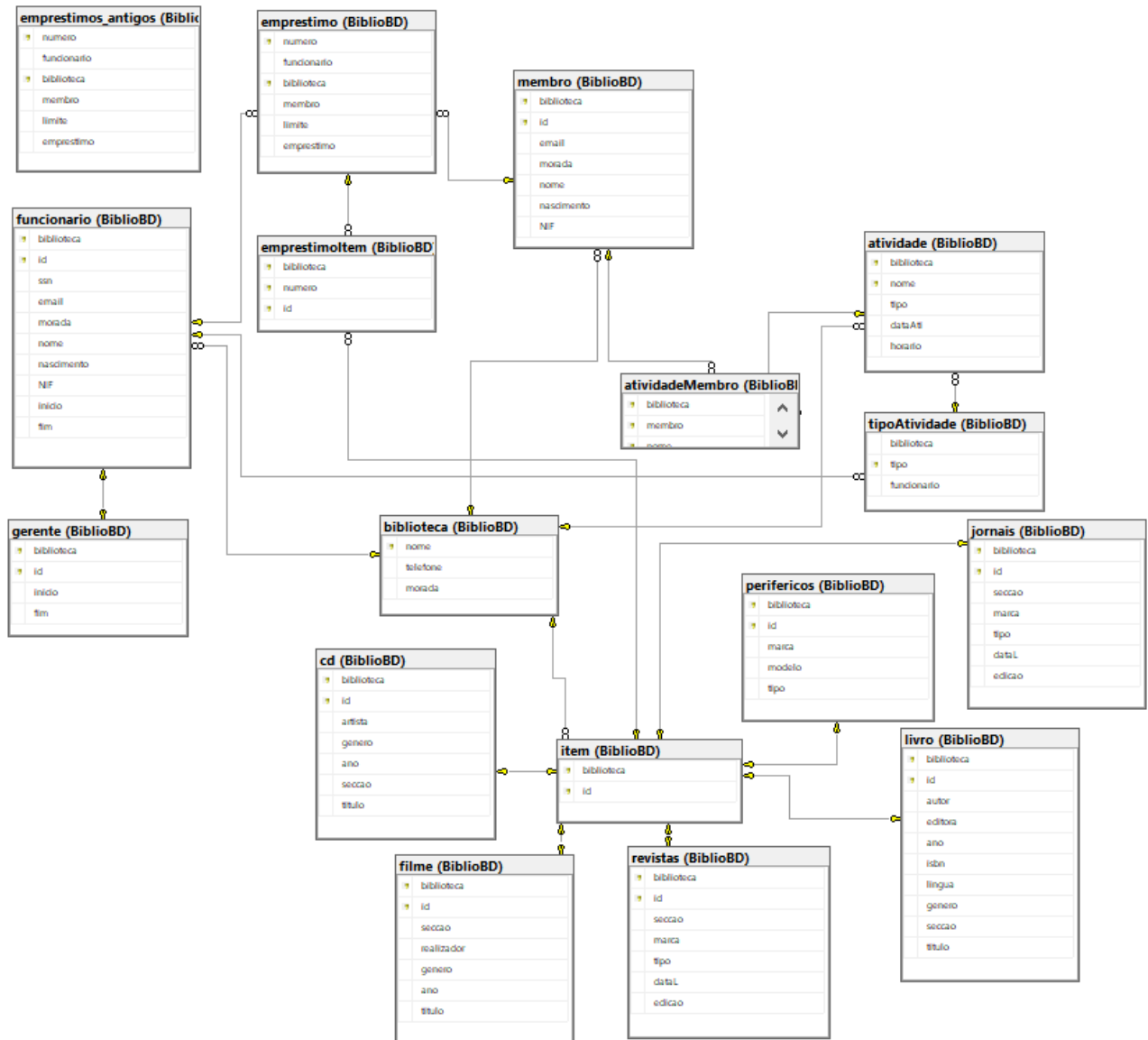


No entanto, após a normalização das entidades (ver ponto seguinte), obtivemos o seguinte esquema.



## IV - Diagrama da Base de Dados

Nas múltiplas entidades definidas, temos as seguintes dependências:



## V - Normalização

Nas múltiplas entidades definidas, temos as seguintes dependências:

### **Funcionário:**

{IDF,FK\_Bibl} → {SSN,Mail,Morada,Nome, Nasc,NIF,Inicio,Fim}

### **Gerente:**

{IDF,FK\_Bibl} → {Inicio,Fim}

### **Biblioteca:**

Nome → {Morada,Telefone}

### **Membro:**

{IDM,FK\_Bibl} → {Nome,Morada,Mail,NIF,Nasc}

### **Empréstimo:**

{Data lim,Data emp} → Num (temos a entidade EmpréstimoItem)

Num → {FK\_Func,FK\_Mem}

### **Atividade:**

{nome,FK\_Bibl} → {Tipo,data,horario}

Tipo → Funcionário (criamos a entidade Tipo\_atividade)

### **Jornais e revistas:**

ID → {seccao,marca,tipo,data,edicao}

### **Filme:**

ID → {realizador,genero,ano,seccao,titulo}

### **Periféricos:**

ID → {marca,modelo,tipo}

### **CD:**

ID→{artista,genero,ano,seccao,titulo}

### **Livro:**

ID → {autor,editora,ano,ISBN,genero,seccao,titulo}



## VI - Interfaces

Como já referimos anteriormente, recorreremos aos Windows Forms e à linguagem C# de forma a fazer a interface gráfica da *LibraryBD*.

Para este projeto criamos 6 Forms:

- Home
  - Interface com um Menu que nos permite escolher as opções listadas abaixo
- Consultar
  - Interface que nos permite consultar os dados de cada entidade da nossa base de dados (funcionário, membro, empréstimo, itens, etc...)
- Emprestar
  - Interface que nos permite criar empréstimos e adicionar itens aos mesmos, associando-os ao membro e funcionário indicados.
- Entregar
  - Interface que nos permite entregar empréstimos ativos, fazendo isso item a item ou entregando o empréstimo completo. Permite também estender um empréstimo por 15 dias.
- Atividades
  - Interface que permite consultar, adicionar e remover atividades, assim como membros que queiram participar nas mesmas.
- Informações da Biblioteca
  - Interface que possui os dados da biblioteca

## VII - SQL DDL em SQL Server

Primeiramente, criamos o esquema “*BiblioBD*” na base de dados correspondente ao nosso grupo (P9G4) que o professor nos facultou.

Utilizando o Esquema mostrado anteriormente, definimos as entidades em SQL com os vários tipos que lhes eram associados, as suas primary keys e relações.

Todas as instruções efetuadas podem ser lidas no ficheiro *DDL.sql* enviado em anexo com o projeto.

## VIII - SQL DML em SQL Server

Antes de definir consultas, triggers, etc, começamos por inserir alguns valores base à nossa Base de Dados. Estas inserções podem ser lidas no ficheiro *Inserts.sql*.

De seguida, decidimos fazer algumas consultas a fim de ver se a nossa base de dados continha os dados e as relações corretas. Estas consultas e os resultados correspondentes podem ser vistos no documento *Consultas.pdf* e o código no ficheiro *consultas.sql*.

Nesta fase, começamos a criar Stored Procedures, User Defined Functions, Triggers e Índices, usando quando necessário Cursores. Todas estas ações podem ser vistas em detalhe nos documentos *biblioteca.sql*, *consultar.sql*, *emprestimos.sql* e *atividades.sql* (cada documento contém as ações que lhe pertencem a fim de as poder utilizar na interface gráfica). Segue uma lista descrevendo o que contém cada ficheiro:

- biblioteca.sql
  - Procedure para buscar info da biblioteca
- consultar.sql
  - Function para procurar reservas de um membro
  - Trigger para verificar as datas de um funcionário ao fazer insert
  - Trigger para verificar as datas de um gerente ao fazer insert usando cursores
  - Function para procurar membro por id
  - Function para procurar membro por nome
  - Procedure para adicionar membro (id é incrementado automaticamente)
  - Procedure para adicionar funcionário (id é incrementado automaticamente)
  - Procedure para eliminar membro por id
  - Procedure para eliminar funcionário por id
  - Procedure para eliminar item por id
  - Functions para procurar empréstimo por número, idf, idm e todas as combinações possíveis
  - Procedure para editar um membro
  - Procedure para verificar disponibilidade de um item por id
- emprestimos.sql
  - Indice Non-Clustered que vai ordenar os empréstimos pelo ID do Membro
  - Indice Non-Clustered que vai ordenar os itens dos empréstimos pelo número de empréstimo
  - Procedure para estender a data limite de um empréstimo (aumenta 15 dias da data limite)
  - Function que verifica se um determinado item já está emprestado
  - Function que verifica se um utilizador tem mais que 5 itens emprestados
  - Function que verifica se um determinado utilizador tem algum empréstimo em atraso
  - Function que verifica se um determinado funcionário ainda trabalha na biblioteca
  - Procedure que adiciona um novo empréstimo
  - Procedure que adiciona um item a um certo empréstimo

- Procedure que elimina um certo item emprestado da tabela de itensEmprestados (funciona como uma entrega de um único item que estava emprestado)
  - Procedure que elimina um empréstimo da tabela de empréstimos, e elimina também os seus itens associados (funciona como uma entrega de todos os itens que estavam emprestados)
  - Trigger que, quando um empréstimo é apagado, o transfere para a tabela 'emprestimos\_antigos' de forma a manter um historial de empréstimos
  - Procedure para obter nome de item de acordo com o id
- atividades.sql
    - Procedure para adicionar atividade
    - Function para obter atividades
    - Function para obter membros de atividade pelo nome da atividade
    - Procedure para eliminar atividades por nome
    - Function para obter atividades de um membro pelo id do membro
    - Function para obter atividades de um membro pelo id do membro e tipo de atividade
    - Function para obter atividades por tipo
    - Procedure para adicionar membro a um atividade
    - Procedure para remover membro de uma atividade

## **Entregáveis**

Todos os ficheiros mencionados neste relatório serão entregues juntamente com o mesmo.

A distribuição dos ficheiros está explicada no ficheiro README.md presente no diretório raiz da pasta submetida,

Será também utilizado um vídeo de demonstração onde são exploradas as funcionalidades principais do nosso projeto.

Link do vídeo: <https://www.shrunkn.com/a9Pxt>