

# Relatório Final

Gestão de Bibliotecas

Base de Dados



BIBLIOTECA  
UNIVERSITARIA

Projeto realizado por:  
Fábio Silva [73786]  
Rui Oliveira [73179]

Docente:  
Carlos Costa  
Joaquim Sousa Pinto  
Sérgio Matos

## Índice

[Índice](#)

[Introdução](#)

[Diagrama Entidade Relação](#)

[Esquema relacional](#)

[SQL DDL](#)

[Normalização](#)

[Índices](#)

[Triggers](#)

[User Defenition Functions](#)

[DML associado à interface gráfica](#)

## Introdução

Este projeto surgiu no âmbito da cadeira Base de Dados, e como tal, tem como objectivo a implementação de uma base de dados e uma interface para manipulação da mesma. A base de dados escolhida foi uma base de dados para gestão de bibliotecas.

O procedimento da realização foi de acordo com o proposto pelos guiões práticos da cadeira. Começou-se por fazer uma análise de requisitos, reunindo informações sobre as principais entidades e as principais operações sobre estas. Posteriormente foi feito o desenho conceptual, modelo relacional e a sua implementação. Sobre este foram introduzidos dados e feitas algumas queries. Paralelamente foi desenvolvida a interface gráfica do sistema e foram implementados alguns stored procedures/triggers/user definition functions de acordo com as necessidades.

O presente relatório está organizado pela ordem de implementação.

Após uma minuciosa análise chegou-se à conclusão que a aplicação de gestão de bibliotecas deveria permitir realizar as seguintes ações:

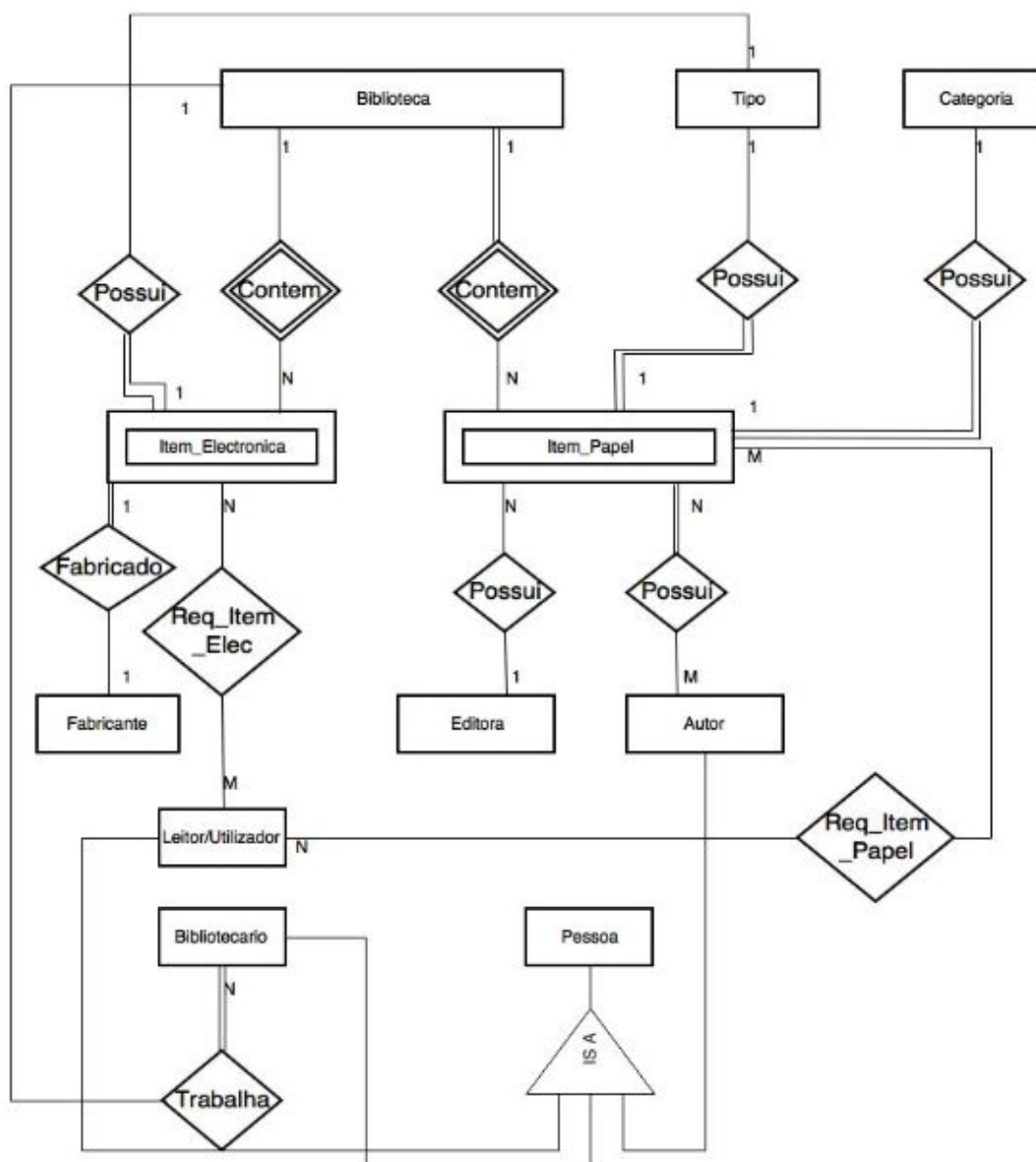
- Inserir, eliminar e fazer update de todas as entidades referidas anteriormente;
- Atribuir à pessoa um leitor, um bibliotecária ou um autor;
- Realizar uma requisição para um dado item de papel ou item eletrónica;
- Consultar o numero total de dinheiro ganho pela biblioteca em multas;
- Consultar o número total de requisições efetuadas pelo leitor.

## Diagrama Entidade Relação

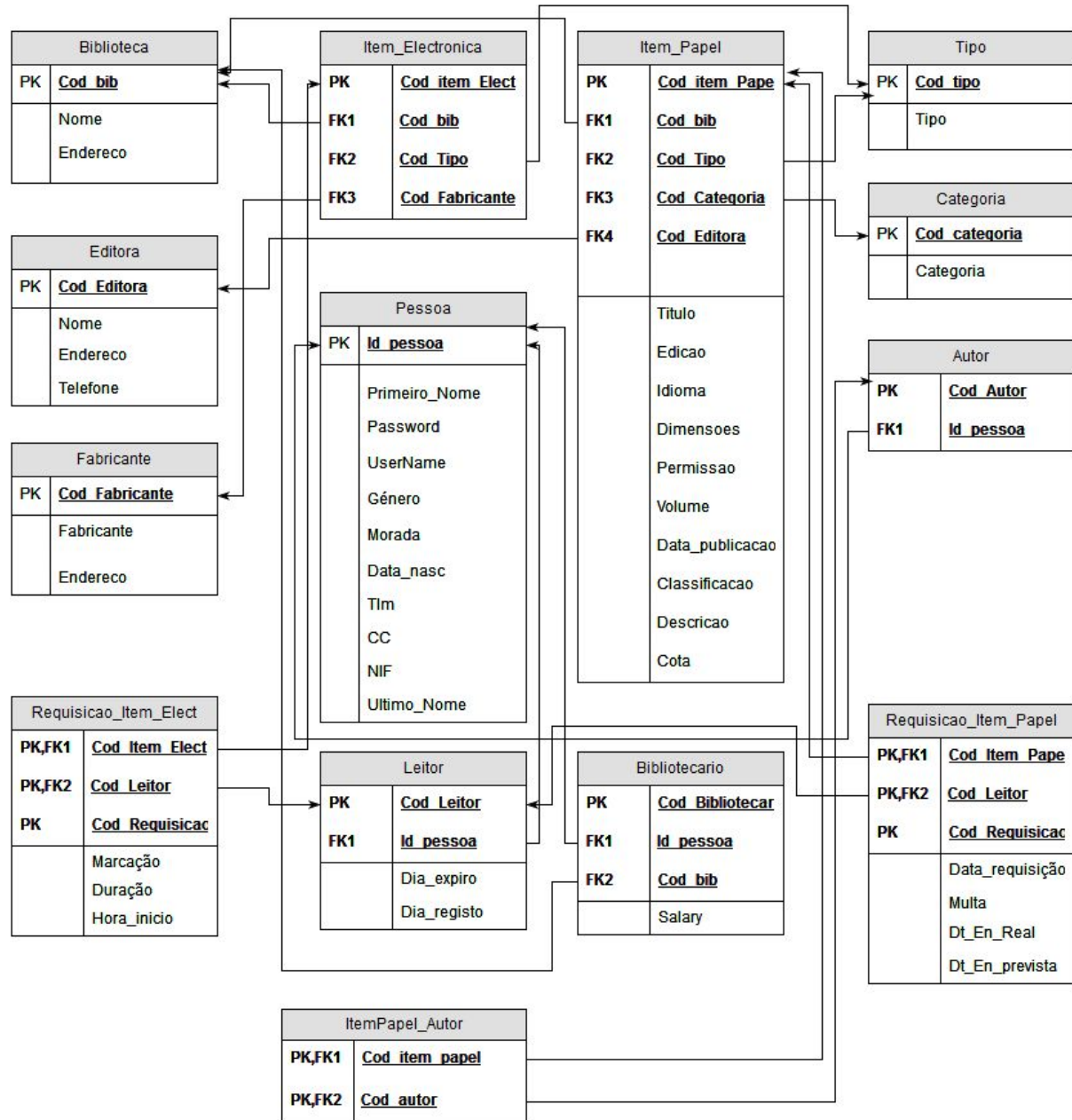
Numa primeira análise foi feita uma seleção das entidades mais importantes que compõem a nossa base de dados:

- Item Papel
- Item Eletrónica
- Requisição Item papel
- Requisição Item Eletrónica
- Categoria
- Tipo
- Fabricante
- Editora
- Pessoa
- Autor
- Leitor
- Bibliotecário
- Biblioteca

Após seleção das entidades procedeu-se à elaboração do seguinte diagrama entidade relação.



## Esquema relacional



## SQL DDL

```
CREATE TABLE PROJECT.BIBLIOTECA(  
  COD_BIB INTEGER CHECK (COD_BIB LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  NOME Varchar(70) NOT NULL,  
  ENDereco Varchar(50) NOT NULL,  
  CONSTRAINT PKBIBLIOTECA PRIMARY KEY(COD_BIB));  
  
CREATE TABLE PROJECT.EDITORIA(  
  COD_EDITORIA INTEGER CHECK(COD_EDITORIA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  NOME Varchar(70) NOT NULL,  
  ENDereco Varchar(50) DEFAULT('UNKNOWN'),  
  TELEFONE INTEGER CHECK( TELEFONE LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' ),  
  CONSTRAINT PKEDITORIA PRIMARY KEY(COD_EDITORIA));  
  
CREATE TABLE PROJECT.FABRICANTE(  
  COD_FABRICANTE INTEGER CHECK( COD_FABRICANTE LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  FABRICANTE Varchar(50) NOT NULL,  
  ENDereco Varchar(50) DEFAULT('UNKNOWN'),  
  TELEFONE INTEGER CHECK(TELEFONE LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' ),  
  CONSTRAINT PKFABRICANTE PRIMARY KEY(COD_FABRICANTE));  
  
CREATE TABLE PROJECT.TIPO(  
  COD_TIPO INTEGER CHECK(COD_TIPO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  TIPO Varchar(50) NOT NULL,  
  CONSTRAINT PKTIPO PRIMARY KEY(COD_TIPO));  
  
CREATE TABLE PROJECT.CATEGORIA(  
  COD_CATEGORIA INTEGER CHECK(COD_CATEGORIA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  CATEGORIA Varchar(50) NOT NULL,  
  CONSTRAINT PKCATEGORIA PRIMARY KEY(COD_CATEGORIA));  
  
CREATE TABLE PROJECT.PESSOA(  
  ID_PESSOA INTEGER CHECK(ID_PESSOA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  PRIMEIRO_NOME Varchar(70) NOT NULL,  
  ULTIMO_NOME Varchar(70),  
  MORADA Varchar(50) NOT NULL,  
  GENERO Char(1) CHECK(GENERO='F' OR GENERO='M'),  
  DATA_NASC DATE ,  
  TLM INTEGER,  
  CC INTEGER CHECK(CC LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  NIF INTEGER CHECK(NIF LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  CONSTRAINT PKPESSOA PRIMARY KEY(ID_PESSOA));  
  
CREATE TABLE PROJECT.AUTOR(  
  COD_AUTOR INTEGER CHECK( COD_AUTOR LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' ),  
  ID_PESSOA INTEGER CHECK( ID_PESSOA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' ),  
  CONSTRAINT FKAUTOR FOREIGN KEY(ID_PESSOA) REFERENCES PROJECT.PESSOA(ID_PESSOA) ON DELETE SET NULL,  
  CONSTRAINT PKAUTOR PRIMARY KEY(COD_AUTOR));
```



```
CREATE TABLE PROJECT.LEITOR(
  COD_LEITOR INTEGER CHECK(COD_LEITOR LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  ID_PESSOA INTEGER CHECK(ID_PESSOA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  USERNAME Varchar(10) NOT NULL,
  PASS Varchar(15) NOT NULL,
  DATA_EXPIRO DATE NOT NULL,
  DATA_REGISTO DATE NOT NULL ,
  COD_BIB INTEGER CHECK(COD_BIB LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  CONSTRAINT PKLEITOR_COD_BIB FOREIGN KEY(COD_BIB) REFERENCES PROJECT.BIBLIOTECA(COD_BIB) ON DELETE SET NULL,
  CONSTRAINT PKLEITOR PRIMARY KEY(COD_LEITOR),
  CONSTRAINT FKLEITOR_ID_PESSOA FOREIGN KEY(ID_PESSOA) REFERENCES PROJECT.PESSOA(ID_PESSOA) ON DELETE SET NULL,
  CHECK (DATA_REGISTO<DATA_EXPIRO));
```

```
CREATE TABLE PROJECT.BIBLIOTECARIO(
  COD_BIBLIOTECARIO INTEGER CHECK(COD_BIBLIOTECARIO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  ID_PESSOA INTEGER CHECK(ID_PESSOA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  COD_BIB INTEGER CHECK(COD_BIB LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  SALARY DECIMAL(6,2) NOT NULL CHECK(SALARY < 2000),
  USERNAME Varchar(10) NOT NULL,
  PASS Varchar(15) NOT NULL,
  CONSTRAINT PKBIBLIOTECARIO PRIMARY KEY(COD_BIBLIOTECARIO),
  CONSTRAINT FKBIOTECARIO_ID_PESSOA FOREIGN KEY(ID_PESSOA) REFERENCES PROJECT.PESSOA(ID_PESSOA) ON DELETE SET NULL,
  CONSTRAINT FKBIOTECARIO_COD_BIB FOREIGN KEY(COD_BIB) REFERENCES PROJECT.BIBLIOTECA(COD_BIB) ON DELETE SET NULL);
```

```
CREATE TABLE PROJECT.ITEM_PAPEL(
  COD_ITEM_PAPEL INTEGER CHECK( COD_ITEM_PAPEL LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  COD_BIB INTEGER CHECK( COD_BIB LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  COD_TIPO INTEGER CHECK( COD_TIPO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  COD_CATEGORIA INTEGER CHECK( COD_CATEGORIA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  COD_EDITORA INTEGER CHECK( COD_EDITORA LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  TITULO Varchar(100) NOT NULL,
  EDICAO Integer NOT NULL CHECK( EDICAO LIKE '[0-9][0-9][0-9]'),
  IDIOMA Varchar(10) NOT NULL,
  DIMENSOES Varchar(12) DEFAULT('UNKNOWN'),
  PERMISSAO BIT NOT NULL CHECK(PERMISSAO=0 or PERMISSAO=1),
  VOLUME Integer CHECK(VOLUME LIKE '[0-9][0-9][0-9]') ,
  DATA_PUB DATE ,
  CLASSIFICACAO INTEGER CHECK(CLASSIFICACAO < 11),
  DESCRICAO Varchar(250) DEFAULT('NO DESCRIPTION'),
  COTA Integer NOT NULL,
  CONSTRAINT PKITEM_PAPEL PRIMARY KEY(COD_ITEM_PAPEL),
  CONSTRAINT FKITEM_PAPEL_COD_BIB FOREIGN KEY(COD_BIB) REFERENCES PROJECT.BIBLIOTECA(COD_BIB) ON DELETE SET NULL,
  CONSTRAINT FKITEM_PAPEL_COD_TIPO FOREIGN KEY(COD_TIPO) REFERENCES PROJECT.TIPO(COD_TIPO) ON DELETE SET NULL,
  CONSTRAINT FKITEM_PAPEL_COD_CATEGORIA FOREIGN KEY(COD_CATEGORIA) REFERENCES PROJECT.CATEGORIA(COD_CATEGORIA) ON DELETE SET NULL,
  CONSTRAINT FKITEM_PAPEL_COD_EDITORA FOREIGN KEY(COD_EDITORA) REFERENCES PROJECT.EDITORIA(COD_EDITORA) ON DELETE SET NULL);
```



```
CREATE TABLE PROJECT.ITEM_PAPEL_AUTOR(  
  COD_ITEM_PAPEL INTEGER CHECK(COD_ITEM_PAPEL LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  COD_AUTOR INTEGER CHECK(COD_AUTOR LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  CONSTRAINT FKITEM_PAPEL_AUTOR_ITEM FOREIGN KEY(COD_ITEM_PAPEL) REFERENCES PROJECT.ITEM_PAPEL(COD_ITEM_PAPEL),  
  CONSTRAINT FKITEM_PAPEL_AUTOR_AUTOR FOREIGN KEY(COD_AUTOR) REFERENCES PROJECT.AUTOR(COD_AUTOR),  
  CONSTRAINT PKITEM_PAPEL_AUTOR PRIMARY KEY(COD_ITEM_PAPEL, COD_AUTOR));  
  
CREATE TABLE PROJECT.REQUISICAO_ITEM_ELECT(  
  COD_REQUISICAO INTEGER CHECK(COD_REQUISICAO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  COD_ITEM_ELECT INTEGER CHECK(COD_ITEM_ELECT LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  COD_LEITOR INTEGER CHECK(COD_LEITOR LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  MARCAcao DATE DEFAULT('01/01/0000'),  
  DURACAO INTEGER DEFAULT('0'),  
  INICIO TIME DEFAULT('00:00:00'),  
  CONSTRAINT FKREQUISICAO_ITEM_ELECT_COD_ITEM_ELECT FOREIGN KEY(COD_ITEM_ELECT) REFERENCES PROJECT.ITEM_ELECTRONICA(COD_ITEM_ELECT),  
  CONSTRAINT FKREQUISICAO_ITEM_ELECT_COD_LEITOR FOREIGN KEY(COD_LEITOR) REFERENCES PROJECT.LEITOR(COD_LEITOR),  
  CONSTRAINT PKREQUISICAO_ITEM_ELECT PRIMARY KEY(COD_REQUISICAO));  
  
CREATE TABLE PROJECT.REQUISICAO_ITEM_PAPEL(  
  COD_ITEM_PAPEL INTEGER CHECK(COD_ITEM_PAPEL LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  COD_LEITOR INTEGER CHECK(COD_LEITOR LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  COD_REQUISICAO INTEGER CHECK(COD_REQUISICAO LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  
  DATA_REQUISICAO DATE DEFAULT('01/01/0000'),  
  DT_EN_REAL DATE DEFAULT('01/01/0000'),  
  DT_EN_PREVISTA DATE DEFAULT('01/01/0000'),  
  MULTA DECIMAL(5,2) CHECK(MULTA < 200),  
  CONSTRAINT FKREQUISICAO_ITEM_PAPEL_COD_ITEM_PAPEL FOREIGN KEY(COD_ITEM_PAPEL) REFERENCES PROJECT.ITEM_PAPEL(COD_ITEM_PAPEL),  
  CONSTRAINT FKREQUISICAO_ITEM_PAPEL_COD_LEITOR FOREIGN KEY(COD_LEITOR) REFERENCES PROJECT.LEITOR(COD_LEITOR),  
  CONSTRAINT PKREQUISICAO_ITEM_PAPEL PRIMARY KEY(COD_REQUISICAO));
```

## Normalização

Após análise do modelo relacional concluiu-se que não existe nenhuma violação das formas normais, estando a base de dados na 3 Forma Normal.

## Índices

Após análise do modelo relacional e considerando as queries utilizadas no projeto chegou-se à conclusão que não era necessário a criação de novos índices para além dos que são criados por *default* para chaves primárias.

## Triggers

De forma a verificar a consistência da informação inserida pelo utilizador, procedeu-se à criação de triggers. Isto permitiu verificar os dados antes de os inserir na base de dados. Assim foi possível garantir a integridade do sistema e certificar que os dados são válidos.

O seguinte trigger, "**marcacao\_on\_req\_item\_elect**", possui as seguintes funcionalidades:

1. Verifica se a data inserida pelo utilizador é superior ou igual à data atual;
2. Consoante o dia da semana correspondente à marcação da requisição proposta pelo utilizador, o sistema verifica se a data de inicio da requisição mais o tempo corresponde à duração da requisição estão dentro do horário da biblioteca.

```
GO
CREATE TRIGGER MARCAÇÃO_ON_REQ_ITEM_ELECT ON PROJECT.REQUISICAO_ITEM_ELECT
INSTEAD OF INSERT, UPDATE
AS
BEGIN
    DECLARE @DOM VARCHAR(10)
    DECLARE @DATE DATE
    DECLARE @DURACAO INTEGER
    DECLARE @INICIO TIME
    DECLARE @FREETIME INTEGER
    DECLARE @BEGINS TIME = '10:00:00'
    DECLARE @ENDF TIME = '19:00:00'
    DECLARE @ENDS TIME = '13:00:00'
    DECLARE @ACTUAL_DATE DATE
    DECLARE @action as char(1);
    DECLARE @CIE INT
    DECLARE @CL INT
    DECLARE @CR INT
    SELECT @DATE=MARCAÇÃO, @DURACAO=DURACAO, @INICIO=INICIO, @CIE=COD_ITEM_ELECT, @CL=COD_LEITOR, @CR=COD_REQUISICAO FROM inserted
    SET @DOM = dbo.GETDOM(@DATE)
    SET @ACTUAL_DATE = GETDATE()

    SET @action = 'I';
    IF EXISTS(SELECT * FROM DELETED)
    BEGIN
        SET @action=
        CASE
            WHEN EXISTS(SELECT * FROM INSERTED) THEN 'U'
            ELSE 'D'
        END
    END

    IF @DATE < @ACTUAL_DATE
    BEGIN
        RAISERROR('Data inválida!',16,1);
        RETURN;
    END
END
ELSE
```

```
IF @DOM = 'Monday' OR @DOM = 'Tuesday' OR @DOM = 'Wednesday' OR @DOM = 'Thursday' OR @DOM = 'Friday'
BEGIN
    IF @INICIO >= @BEGINS AND @INICIO <= @ENDF
    BEGIN
        SET @FREETIME = (SELECT DATEDIFF(MI, @INICIO, @ENDF))
        IF @DURACAO <= @FREETIME
        IF @action='I'
            INSERT INTO PROJECT.REQUISICAO_ITEM_ELECT SELECT * FROM inserted;
        ELSE
            UPDATE PROJECT.REQUISICAO_ITEM_ELECT
            SET COD_ITEM_ELECT=@CIE, COD_LEITOR=@CL, MARCAÇÃO=@DATE, DURACAO=@DURACAO, INICIO=@INICIO
            WHERE COD_REQUISICAO=@CR
        ELSE
            RAISERROR('A duração da marcação tem que ser compatível com o horário da biblioteca!',16,1);
        END
    ELSE
        RAISERROR('O início da marcação tem que ser compatível com o horário da biblioteca!',16,1);
    END
    IF @DOM = 'Saturday'
    BEGIN
        IF @INICIO >= @BEGINS AND @INICIO <= @ENDS
        BEGIN
            SET @FREETIME = (SELECT DATEDIFF(MI, @INICIO, @ENDS))
            IF @DURACAO <= @FREETIME
            IF @action='I'
                INSERT INTO PROJECT.REQUISICAO_ITEM_ELECT SELECT * FROM inserted;
            ELSE
                UPDATE PROJECT.REQUISICAO_ITEM_ELECT
                SET COD_ITEM_ELECT=@CIE, COD_LEITOR=@CL, MARCAÇÃO=@DATE, DURACAO=@DURACAO, INICIO=@INICIO
                WHERE COD_REQUISICAO=@CR
            ELSE
                RAISERROR('A duração da marcação tem que ser compatível com o horário da biblioteca!',16,1);
            END
        ELSE
            RAISERROR('O início da marcação tem que ser compatível com o horário da biblioteca!',16,1);
        END
    END
    IF @DOM='Sunday'
    BEGIN
        RAISERROR('O horário da biblioteca é de Segunda a Sábado. Por favor escolha outro dia da semana!',16,1);
    END
END
```

A seleção de triggers que se seguem têm como principal objectivo eliminar os problemas de “dependências”, isto é, quando uma chave primária de uma relação é referenciada noutra entidade. Optou-se por eliminar os tuplos onde a chave é referenciada e o tuplo na relação onde é chave primária, similar à opção não suportada em sql server (“*delete on cascade*”). Outra possibilidade seria colocar a “*null*” onde a chave é referenciada e apagar na relação em que é chave primária.

O trigger “**deleteitempapel**” foi criado com vista a apagar as entradas da tabela **item\_papel** e onde este é referenciado (“**requisicao\_item\_papel\_autor**”).

```
CREATE TRIGGER DELETEITEMPAPEL ON PROJECT.ITEM_PAPEL
INSTEAD OF DELETE
AS
BEGIN
    DELETE FROM PROJECT.ITEM_PAPEL_AUTOR WHERE COD_ITEM_PAPEL = (SELECT COD_ITEM_PAPEL FROM deleted);
    DELETE FROM PROJECT.ITEM_PAPEL WHERE COD_ITEM_PAPEL = (SELECT COD_ITEM_PAPEL FROM deleted);
END
```

O trigger “**deleteleitor**” foi criado com vista a apagar as entradas da tabela “**leitor**” e onde este é referenciado ( “**requisicao\_item\_elect**” e “**requisicao\_item\_papel**”).

```
CREATE TRIGGER DELETELEITOR ON PROJECT.LEITOR
INSTEAD OF DELETE
AS
BEGIN
    DELETE FROM PROJECT.REQUISICAO_ITEM_ELECT where COD_LEITOR = (SELECT COD_LEITOR FROM deleted) ;
    DELETE FROM PROJECT.REQUISICAO_ITEM_PAPEL where COD_LEITOR = (SELECT COD_LEITOR FROM deleted);
    DELETE FROM PROJECT.LEITOR WHERE COD_LEITOR = (SELECT COD_LEITOR FROM deleted);
END
```

O trigger “**delete\_item\_elect**” foi criado com vista a apagar as entradas da tabela “**item\_electronica**” e onde este é referenciado (“**requisicao\_item\_elect**”).

```
GO
CREATE TRIGGER DELETE_ITEM_ELECT ON PROJECT.ITEM_ELECTRONICA
INSTEAD OF DELETE
AS
BEGIN
    DELETE FROM PROJECT.REQUISICAO_ITEM_ELECT WHERE COD_ITEM_ELECT = (SELECT COD_ITEM_ELECT FROM deleted);
    DELETE FROM PROJECT.ITEM_ELECTRONICA WHERE COD_ITEM_ELECT = (SELECT COD_ITEM_ELECT FROM deleted);
END
```

O trigger “**deleteautor**” foi criado com vista a apagar as entradas da tabela “**autor**” e onde este é referenciado (“**item\_papel\_autor**”).

```
CREATE TRIGGER DELETEAUTOR ON PROJECT.AUTOR
INSTEAD OF DELETE
AS
BEGIN
    DELETE FROM PROJECT.ITEM_PAPEL_AUTOR WHERE COD_AUTOR = (SELECT COD_AUTOR FROM deleted);
    DELETE FROM PROJECT.AUTOR WHERE COD_AUTOR = (SELECT COD_AUTOR FROM deleted);
END
```

O trigger “**update\_RIP**” foi criando com vista a validar os dados inseridos. São efetuadas validações das datas, tais como, se a data de entrega é superior à data de requisição e se a data de requisição é superior à data de entrega prevista do item.



Verifica-se também a relação entre a data prevista e a data real e atribui-se uma multa de acordo com políticas internas (0.25€ por dia de atraso).

Para além dos aspetos evidenciados anteriormente é também verificado se o item de papel em questão tem permissão para poder ser requisitado.

```

go
CREATE TRIGGER update_RIP ON PROJECT.REQUISICAO_ITEM_PAPEL
INSTEAD OF INSERT, UPDATE
AS
BEGIN
    DECLARE @DATA_PREV DATE;
    DECLARE @DATA_REAL DATE;
    DECLARE @DATA_DIF INT;
    DECLARE @DATA_REQ DATE;
    DECLARE @DATA_RR INT;
    DECLARE @DATA_RP INT;
    DECLARE @MULTA DECIMAL(5,2);
    DECLARE @COD_REQ INT;
    DECLARE @COD_IP INT;
    DECLARE @COD_L INT;
    DECLARE @action AS char(1);
    DECLARE @PERMISSION AS BIT;
    SELECT @COD_REQ=COD_REQUISICAO, @COD_IP=COD_ITEM_PAPEL, @COD_L=COD_LEITOR, @DATA_REQ=DATA_REQUISICAO, @DATA_PREV= DT_EN_PREVISTA ,
           @DATA_REAL= DT_EN_REAL, @MULTA=MULTA FROM inserted;
    SET @DATA_RR= (SELECT DATEDIFF(day, @DATA_REAL, @DATA_REQ))
    SET @DATA_RP= (SELECT DATEDIFF(day, @DATA_PREV, @DATA_REQ))
    SET @DATA_DIF=(SELECT DATEDIFF(day, @DATA_PREV, @DATA_REAL))
    SET @PERMISSION = dbo.GETPERMISSION(@COD_IP)
    SET @action = 'I';
    IF EXISTS(SELECT * FROM DELETED)

    BEGIN
        SET @action=
        CASE
            WHEN EXISTS(SELECT * FROM INSERTED) THEN 'U'
            ELSE 'D'
        END
    END

    IF @PERMISSION=0
    BEGIN
        RAISERROR('O item de papel selecionado não tem permissão de requisição!',16,1);
        RETURN;
    END

```

```

IF @DATA_RR >0
BEGIN
    RAISERROR('A data da requisição tem que ser uma data mais recente do que a data de entrega do item!',16,1);
    RETURN;
END
IF @DATA_RP >0
BEGIN
    RAISERROR('A data da requisição tem que ser uma data mais recente do que a data prevista de entrega do item!',16,1);
    RETURN;
END
if @DATA_DIF>0
BEGIN
    IF @action='I'
    BEGIN
        SET @MULTA= ABS(@DATA_DIF)*0.25;
        INSERT INTO PROJECT.REQUISICAO_ITEM_PAPEL (COD_REQUISICAO, COD_ITEM_PAPEL, COD_LEITOR, DATA_REQUISICAO, DT_EN_REAL, DT_EN_PREVISTA, MULTA)
        VALUES (@COD_REQ, @COD_IP, @COD_L, @DATA_REQ, @DATA_REAL,@DATA_PREV, @MULTA);
    END
    ELSE
    BEGIN
        SET @MULTA= ABS(@DATA_DIF)*0.25;
        UPDATE PROJECT.REQUISICAO_ITEM_PAPEL SET COD_ITEM_PAPEL= @COD_IP, COD_LEITOR= @COD_L, DATA_REQUISICAO=@DATA_REQ,
        DT_EN_REAL=@DATA_REAL, DT_EN_PREVISTA=@DATA_PREV, MULTA=@MULTA WHERE PROJECT.REQUISICAO_ITEM_PAPEL.COD_REQUISICAO=@COD_REQ;
    END
END
ELSE

```

```

BEGIN
  IF @action='I'
  BEGIN
    SET @MULTA=0;
    INSERT INTO PROJECT.REQUISICAO_ITEM_PAPEL (COD_REQUISICAO, COD_ITEM_PAPEL, COD_LEITOR, DATA_REQUISICAO, DT_EN_REAL, DT_EN_PREVISTA, MULTA)
    VALUES (@COD_REQ, @COD_IP, @COD_L, @DATA_REQ, @DATA_REAL, @DATA_PREV, @MULTA);
  END
  ELSE
  BEGIN
    SET @MULTA= 0
    UPDATE PROJECT.REQUISICAO_ITEM_PAPEL SET COD_ITEM_PAPEL= @COD_IP, COD_LEITOR= @COD_L, DATA_REQUISICAO=@DATA_REQ, DT_EN_REAL=@DATA_REAL,
    DT_EN_PREVISTA=@DATA_PREV, MULTA=@MULTA WHERE PROJECT.REQUISICAO_ITEM_PAPEL.COD_REQUISICAO=@COD_REQ;
  END
END

```

## User Defenition Functions

A UDF “**getfreepeople**” retorna todos as pessoas que não são bibliotecários, autores ou leitores. Isto permite que se possa saber se um determinado “**id\_pessoa**” está ou não associado a um subtipo de pessoa. Assim sendo, é assim possível associar uma pessoa “livre” a um dos seus subtipos.

```

CREATE FUNCTION GETFREEPEOPLE()
RETURNS TABLE
AS
RETURN
(Select PROJECT.PESSOA.ID_PESSOA,PROJECT.PESSOA.PRIMEIRO_NOME, PROJECT.PESSOA.ULTIMO_NOME,PROJECT.PESSOA.MORADA, PROJECT.PESSOA.GENERO,
PROJECT.PESSOA.DATA_NASC, PROJECT.PESSOA.TLM, PROJECT.PESSOA.CC, PROJECT.PESSOA.NIF
FROM PROJECT.PESSOA LEFT JOIN PROJECT.LEITOR ON PROJECT.PESSOA.ID_PESSOA=PROJECT.LEITOR.ID_PESSOA
LEFT JOIN PROJECT.BIBLIOTECARIO ON PROJECT.PESSOA.ID_PESSOA=PROJECT.BIBLIOTECARIO.ID_PESSOA
LEFT JOIN PROJECT.AUTOR ON PROJECT.PESSOA.ID_PESSOA=PROJECT.AUTOR.ID_PESSOA
WHERE PROJECT.LEITOR.USERNAME is NULL and PROJECT.BIBLIOTECARIO.SALARY is NULL and PROJECT.AUTOR.COD_AUTOR is Null)

```

A UDF “**getdom**” retorna o dia da semana de uma determinada data. Esta udf é utilizada no trigger “**marcacao\_on\_req\_item\_elect**”.

```

CREATE FUNCTION GETDOM (@mrc Date) RETURNS VARCHAR(10)
AS
BEGIN
  DECLARE @DOM VARCHAR(10)
  SET @DOM = (Select DATENAME(dw, (@mrc)))
RETURN @DOM
END

```

A UDF “**GetPermission**” retorna a permissão de requisição para um dado “**item\_papel**”. Esta UDF é utilizada no trigger “**update\_rip**” para permitir a requisição de um item de papel ou não.

```

CREATE FUNCTION GETPERMISSION (@book_id INT) RETURNS BIT
AS
BEGIN
  DECLARE @perm BIT
  SELECT @perm=PERMISSAO FROM PROJECT.ITEM_PAPEL WHERE COD_ITEM_PAPEL=@book_id
RETURN @perm
END

```

A UDF “**Req\_Pessoa**” retorna o número total de requisições efetuadas por um dado leitor (requisições de itens de papel e eletrónicos).

```
CREATE FUNCTION REQ_PESSOA (@ID INT) RETURNS INT
AS
BEGIN
    DECLARE @NR_REQ_PAPEL INT
    DECLARE @NR_REQ_ELECT INT
    SELECT @NR_REQ_PAPEL = count(*)
    FROM PROJECT.REQUISICAO_ITEM_PAPEL
    WHERE PROJECT.REQUISICAO_ITEM_PAPEL.COD_LEITOR=@ID

    SELECT @NR_REQ_ELECT = count(*)
    FROM PROJECT.REQUISICAO_ITEM_ELECT
    WHERE PROJECT.REQUISICAO_ITEM_ELECT.COD_LEITOR=@ID

    RETURN @NR_REQ_PAPEL+@NR_REQ_ELECT
END
```

A UDF **TotalMoney** retorna o número total de dinheiro ganho pela biblioteca em multas relativamente às requisições item papel.

```
CREATE FUNCTION TotalMoney() RETURNS Decimal(10,2)
AS
BEGIN
    DECLARE @total Decimal(10,2)
    SELECT @total = sum(MULTA)
    FROM PROJECT.REQUISICAO_ITEM_PAPEL

    RETURN @TOTAL
END
```

A implementação de stored procedures no projeto teve como principal objetivo aumentar a segurança e performance na inserção e update de tuplos relacionados com a entidade pessoa e seus subtipos. Para aumentar ainda mais a segurança utilizou-se stored procedures com encriptação.

Relativamente às restantes entidades, a inserção, updates e deletes foram realizados com queries *ad hoc*.

O stored procedure “**update\_leitor**” foi criado para facilitar o update de informação relativa ao leitor e, por sua vez, à pessoa, ou seja, é feito um update às informações enquanto “leitor”, p.e “username” e também é feito um update às suas informações enquanto pessoa, p.e “primeiro nome”.

```

CREATE PROC UPDATE_LEITOR @ID_PESSOA INT, @COD_LEITOR INT, @COD_BIB INT, @USERNAME VARCHAR(10), @PASS VARCHAR(15),
                           @DATA_EXP DATE, @DATA_REG DATE, @PN VARCHAR(70), @UN VARCHAR(70), @MORADA VARCHAR(50),
                           @GENERO CHAR(1), @DATA_NASC DATE, @TLM INT, @CC INT, @NIF INT
AS
BEGIN
    BEGIN TRANSACTION;
    BEGIN TRY
        UPDATE PROJECT.LEITOR SET ID_PESSOA=@ID_PESSOA, USERNAME=@USERNAME, PASS=@PASS, DATA_EXPIRO=@DATA_EXP, DATA_REGISTO=@DATA_REG, COD_BIB=@COD_BIB
        WHERE PROJECT.LEITOR.COD_LEITOR=@COD_LEITOR;
        UPDATE PROJECT.PESSOA SET PRIMEIRO_NOME=@PN, ULTIMO_NOME=@UN, CC=@CC, NIF=@NIF, MORADA=@MORADA, GENERO=@GENERO, TLM=@TLM, DATA_NASC=@DATA_NASC
        WHERE PROJECT.PESSOA.ID_PESSOA=@ID_PESSOA;
    END TRY

    BEGIN CATCH
        SELECT
            ERROR_NUMBER() AS ErrorNumber
            ,ERROR_SEVERITY() AS ErrorSeverity
            ,ERROR_STATE() AS ErrorState
            ,ERROR_PROCEDURE() AS ErrorProcedure
            ,ERROR_LINE() AS ErrorLine
            ,ERROR_MESSAGE() AS ErrorMessage;

        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
    END CATCH;
    IF @@TRANCOUNT > 0
        COMMIT TRANSACTION;
END

```

O Stored Procedure “**Inserir\_Leitor**” foi criado para facilitar a inserção de novos tuplos na relação Leitor.

```

GO
CREATE PROC INSERT_LEITOR @COD_LEITOR INT, @ID_PESSOA INT, @USERNAME VARCHAR(10), @PASS VARCHAR(15),
                           @DATA_EXP DATE, @DATA_REG DATE, @COD_BIB INT
WITH ENCRYPTION
AS
BEGIN
    BEGIN TRANSACTION;
    BEGIN TRY
        INSERT INTO PROJECT.LEITOR VALUES( @COD_LEITOR, @ID_PESSOA, @USERNAME, @PASS, @DATA_EXP, @DATA_REG, @COD_BIB);
    END TRY

    BEGIN CATCH
        SELECT
            ERROR_NUMBER() AS ERRORNUMBER
            ,ERROR_SEVERITY() AS ERRORSEVERITY
            ,ERROR_STATE() AS ERRORSTATE
            ,ERROR_PROCEDURE() AS ERRORPROCEDURE
            ,ERROR_LINE() AS ERRORLINE
            ,ERROR_MESSAGE() AS ERRORMESSAGE;

        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;
    END CATCH;
    IF @@TRANCOUNT > 0
        COMMIT TRANSACTION;
END

```



## DML associado à interface gráfica

### Interface Pessoa

- Selecionar tuplos associados à entidade **"Pessoa"**  
`SELECT * FROM Project.Pessoa`
- Inserção na base de dados associados à entidade **"Pessoa"** usando a stored procedure **"insert\_pessoa"**  
`INSERT_PESSOA @IDPessoa, @primeiroNome, @ultimoNome, @Morada, @Genero, @DataNasc, @TIm, @Cc, @Nif`
- Update de tuplos associados à entidade **"Pessoa"**  
`UPDATE PROJECT.PESSOA " &  
"SET PRIMEIRO_NOME = @primeiroNome, " &  
" ULTIMO_NOME = @ultimoNome, " &  
" CC = @Cc, " &  
" NIF = @Nif, " &  
" MORADA = @Morada, " &  
" GENERO = @Genero, " &  
" TLM = @TIm, " &  
" DATA_NASC = @DataNasc " &  
"WHERE ID_Pessoa = @IDPessoa"`
- Delete dos valores associados à entidade **"Pessoa"**  
`DELETE FROM PROJECT.PESSOA WHERE ID_PESSOA=@IDPessoa`

Gestão - Biblioteca

Ficheiro Ver

### Pessoa

105336450	Diane Wright	Primeiro Nome:	Diane	Último Nome:	Wright
111111111	Evelyn Palmer	CC:	870139049	NIF:	867850491
112754140	Sara Harris	Morada:	73 Bashford Pass		
116676511	Jessica Price	Género:	F	TLM:	265689753
123123523	Fabio Silva	ID Pessoa:	105336450	Data Nascimento:	20 de junho de 1950
124284077	Elizabeth Harper	Listar por:	<input type="text"/>		
127109022	Shawn Ramos	Items encontrados:	102		
131131657	Eugene Fowler				
139318281	Steve Sullivan				
139615259	Robin Bryant				
145659490	Lisa Gomez				
147099555	Sharon Carpenter				
151004571	Joshua Burton				
155009575	Gregory Morris				
159825383	Laura Vasquez				
171119050	Patricia Porter				
180004583	Kimberly Reynolds				
184180146	Johnny Smith				
188068612	Paul Lee				
220659110	Alan Diaz				
232116579	Judith Turner				
236988641	Joshua Bryant				
240962038	Phillip Gonzalez				
246904955	Teresa Daniels				
260242899	John McDonald				
263328257	Evelyn Palmer				

Adicionar Editar Eliminar

## Interface Leitor

- Selecionar tuplos associados à entidade “**Leitor**”

```
SELECT * FROM PROJECT.LEITOR JOIN PROJECT.PESSOA ON
PROJECT.LEITOR.ID_PESSOA = PROJECT.PESSOA.ID_PESSOA
```

- Retorna o número total de requisições efetuadas por um dado “**leitor**”

Insertão dos valores associados à entidade pessoa “insert\_leitor”

```
INSERT_LEITOR @COD_LEITOR, @ID_PESSOA, @USERNAME, @PASS, @DATA_EXP,
@DATA_REG, @COD_BIB
```

- Update dos valores associados á entidade “**Leitor**” usando a stored procedure “**update\_leitor**”

```
UPDATE_LEITOR @ID_PESSOA, @COD_LEITOR, @COD_BIB, @USERNAME,
@PASS,@DATA_EXP, @DATA_REG,@PN,@UN, @MORADA, @GENERO,@DATA_NASC,@TLM,
@CC, @NIF
```

- Delete dos valores associados à entidade “**Leitor**”

```
DELETE FROM PROJECT.LEITOR WHERE COD_LEITOR=@CodLeitor
```

Gestão - Biblioteca

Ficheiro Ver

### Leitor

116676296	Earl Chavez	Primeiro Nome:	Earl	Último Nome:	Chavez
124468179	Jerry Ryan	CC:	574193604	NIF:	104534423
154013758	Keith Reyes	Morada:	497 Jana Parkway		
163029281	Christopher Hall	Género:	M	TLM:	851134448
190570745	Frank Vasquez	ID Pessoa:	895663527	Data Nascimento:	1 de outubro de 1990
227908645	Gerald Reid	Código Biblioteca:	489839364	Cód Leitor:	116676296
228780187	Shawn Reed	Username:	Imorenow	Password:	*****
232076691	Diane Wright	Data Registo:	8 de março de 1951	Data Expiro:	6 de dezembro de 2016
240422150	Melissa Jackson	Listar por:			
259857626	Barbara Riley				
287970127	Roger Baker				
297686702	Albert Cox				
301813036	Carolyn Kelly				
302312732	Scott Henderson				
430852166	Jimmy Kelley				
455246060	William Cunningham				
460862688	Betty Carter				
462878572	Willie Davis				
506040594	Ann Owens				
521999502	Elizabeth Wood				
537059759	Wayne Oliver				
586079881	Phillip Ford				
587939284	Phillip Flores				
607663372	Harry Nguyen				
608583108	Teresa Daniels				
608686896	Barbara Coleman				

Ver nr requisições    Adicionar    Editar    Eliminar

## Interface Bibliotecário

- Selecionar tuplos associados à entidade “**Bibliotecário**”  

```
SELECT * FROM PROJECT.BIBLIOTECARIO JOIN PROJECT.PESSOA ON
PROJECT.BIBLIOTECARIO.ID_PESSOA = PROJECT.PESSOA.ID_PESSOA
```
- Seleção de todas as pessoas que não são “**bibliotecários**”, autores ou leitores usando a udf “**getfreepeople**”  

```
SELECT * FROM GETFREEPEOPLE()
```
- Inserção na base de dados associados à entidade “**Bibliotecário**” usando a stored procedure “**insert\_bibliotecário**”  

```
INSERT_BIBLIOTECARIO @COD_BIBLIOTECARIO , @ID_PESSOA, @COD_BIB,
@SALARY, @USERNAME, @PASS
```
- Update dos valores associados à entidade “**Bibliotecário**” usando a stored procedure “**update\_bibliotecario**”  

```
UPDATE_BIBLIOTECARIO @ID_PESSOA, @COD_BIBLIOTECARIO ,
@COD_BIB,@USERNAME, @PASS,@SALARY,@PN,@UN, @MORADA,
@GENERO,@DATA_NASC, @TLM, @CC, @NIF
```
- Delete dos valores associados à entidade “**bibliotecário**”  

```
DELETE FROM PROJECT.BIBLIOTECARIO WHERE COD_BIBLIOTECARIO=@CodBib
```

Gestão - Biblioteca

Ficheiro Ver

### Bibliotecário

114931833 Joe Spencer 156531779 Paul Lee 174767727 Gloria Jordan 192304120 Billy Bishop 205633644 Julia Carpenter 245885021 Shawn Ramos 288175200 Judith Turner 295317920 Rachel Cunningham 318454647 Mildred Richards 324838887 Jimmy Campbell 418070000 Sara Harris 464720217 Patricia Porter 527956630 Julie Foster 539986331 Gregory Grant 648972074 Steve Sullivan 673735110 Alan Diaz 776419502 Evelyn Ganett 777777777 Evelyn Palmer 818333566 Kathleen Mendoza 963576156 Amy Hanson 976509722 Phillip Gonzalez	Primeiro Nome:	<input type="text" value="Joe"/>	Último Nome:	<input type="text" value="Spencer"/>
	CC:	<input type="text" value="568932515"/>	NIF:	<input type="text" value="315765884"/>
	Morada:	<input type="text" value="83 Del Sol Park"/>		
	Género:	<input type="text" value="M"/>	TLM:	<input type="text" value="547582105"/>
	ID Pessoa:	<input type="text" value="846057983"/>	Data Nascimento:	<input type="text" value="8 de julho de 2012"/>
	Código Biblioteca:	<input type="text" value="459567487"/>	Cód Bibliotecario:	<input type="text" value="114931833"/>
	Username:	<input type="text" value="jbrooks9"/>	Password:	<input type="text" value="*****"/>
	Salário:	<input type="text" value="1412,47"/>	Items encontrados:	<input type="text" value="21"/>
	Listar por:	<input type="text"/>		

Adicionar

Editar

Eliminar

## Interface Autor

- Selecciona os valores associados á entidade “**Autor**”  
SELECT \* FROM PROJECT.AUTOR JOIN PROJECT.PESSOA ON  
PROJECT.AUTOR.ID\_PESSOA = PROJECT.PESSOA.ID\_PESSOA
- Seleção de todas as pessoas que não são bibliotecários, autores ou leitores usando a udf “**getfreepeople**”  
SELECT \* FROM GETFREEPEOPLE()
- Inserção na base de dados associados à entidade “**Autor**” usando a stored procedure “**insert\_autor**”  
INSERT\_AUTOR @ID\_PESSOA, @COD\_AUTOR
- Update dos tuplos associado à entidade “**Autor**” utilizando a stored procedure “**update\_autor**”  
UPDATE\_AUTOR @ID\_PESSOA, @COD\_AUTOR, @PN, @UN, @MORADA,  
@GENERO, @DATA\_NASC, @TLM, @CC, @NIF
- Delete dos valores associados à entidade “**Autor**”  
DELETE FROM PROJECT.AUTOR WHERE COD\_AUTOR=@CodAutor

Gestão - Biblioteca

Ficheiro Ver

### Autor

106851922 Ernest Rose  
111111111 Lois Mills  
143168977 Lisa Gomez  
164344875 Jessica Flores  
186408685 Antonio Mills  
192989535 Laura Vasquez  
201046426 Harold Stewart  
260902387 Jean White  
273080549 Edward Dean  
330561587 Kelly Fisher  
414599518 Angela Chapman  
624217038 Lois Mills  
632133582 Teresa James  
651401707 Eugene Fowler  
681207145 Jessica Price  
693809173 Timothy Reynolds  
778081660 Alice Richardson  
850260593 Lillian Hart  
918298736 Julie Scott  
945985228 Tina Matthews  
950623714 Christine Turner

Primeiro Nome:	<input type="text" value="Ernest"/>	Último Nome:	<input type="text" value="Rose"/>
CC:	<input type="text" value="812815742"/>	NIF:	<input type="text" value="652216924"/>
Morada:	<input type="text" value="6550 Roxbury Crossing"/>		
Género:	<input type="text" value="M"/>	TLM:	<input type="text" value="420970850"/>
ID Pessoa:	<input type="text" value="924924066"/>	Data Nascimento:	<input type="text" value="26 de março de 1973"/>
Código Autor:	<input type="text" value="106851922"/>		
Listar por:	<input type="text" value=""/>	Items encontrados:	<input type="text" value="21"/>

Adicionar

Editar

Eliminar

## Interface Item Papel

- Selecionar tuplos associados à entidade “Item Papel”  
SELECT \* FROM PROJECT.ITEM\_PAPEL
- Update dos valores associados à entidade “Item Papel”  
UPDATE PROJECT.ITEM\_PAPEL " &  
"SET COD\_BIB = @CodBiblioteca, " &  
" COD\_TIPO = @CodTipo, " &  
" COD\_CATEGORIA = @CodCategoria, " &  
" COD\_EDITORA = @CodEditora, " &  
" TITULO = @Titulo, " &  
" EDICAO = @Edicao, " &  
" IDIOMA = @Idioma, " &  
" DIMENSOES = @Dimensoes, " &  
" PERMISSAO = @Permissao, " &  
" VOLUME = @Volume, " &  
" DATA\_PUB = @DataPub, " &  
" CLASSIFICACAO = @Classificacao, " &  
" DESCRICAO = @Descricao, " &  
" COTA = @Cota " &  
"WHERE COD\_ITEM\_PAPEL = @CodItemPapel"
- Inserção dos valores associados à entidade “Item Papel”  
INSERT INTO PROJECT.ITEM\_PAPEL (COD\_ITEM\_PAPEL, COD\_BIB, COD\_TIPO, COD\_CATEGORIA, COD\_EDITORA, " &  
"TITULO, EDICAO, IDIOMA, DIMENSOES, PERMISSAO, VOLUME, DATA\_PUB, CLASSIFICACAO, DESCRICAO, COTA" &  
")VALUES (@CodItemPapel, @CodBiblioteca, @CodTipo, @CodCategoria, @CodEditora, @Titulo, @Edicao" &  
", @Idioma, @Dimensoes, @Permissao, @Volume, @DataPub, @Classificacao, @Descricao, @Cota)
- Delete dos valores associados à entidade “Item Papel”  
DELETE FROM PROJECT.ITEM\_PAPEL WHERE COD\_ITEM\_PAPEL=@CodItemPapel

Gestão - Biblioteca

Ficheiro Ver

### Items Papel

103433106	459567487	Proin risu
138621256	459567487	Sed vel e
148179869	459567487	Nulla mol
211703536	459567487	Vivamus
236334445	459567487	Vestibulu
284952329	459567487	Vivamus
305350731	459567487	bd
309442507	459567487	In hac ha
316388775	459567487	Curabitur
342884241	459567487	In est risu
374044458	459567487	Fusce pc
384315964	459567487	Proin eu
400679020	459567487	Curabitur
426140206	459567487	Aliquam e
452982806	459567487	Nam du
474268437	459567487	Vivamus
481665630	459567487	Varius ut
513828495	459567487	Donec d
515312293	459567487	Duis bibe
524493996	459567487	Proin at t
562070871	459567487	Nulla fac
568137681	459567487	Duis bibe
595182780	459567487	Phasellus
606330592	459567487	Duis aliq
705439439	459567487	In quis ju
725190922	459567487	Proin risu

Cód. Item papel:	<input type="text" value="103433106"/>	Cod. Biblioteca:	<input type="text" value="459567487"/>
Cod. Tipo:	<input type="text" value="532711178"/>	Cod. Categoria:	<input type="text" value="955805865"/>
Cod. Editora:	<input type="text" value="167943717"/>	Cod. Autor:	<input type="text" value="201046426"/>
Título:	<input type="text" value="Proin risu."/>	Edição:	<input type="text" value="607"/>
Idioma:	<input type="text" value="Danish"/>	Dimensões:	<input type="text" value="45"/>
Permissao:	<input type="text" value="True"/>	Data publicação:	<input type="text" value="17 de abril de 1854"/>
Volume:	<input type="text" value="906"/>	Descrição:	<input type="text" value="Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus."/>
Classificação:	<input type="text" value="1"/>		
Cota:	<input type="text" value="865"/>		
Listar por:	<input type="text" value=""/>	Items encontrados:	<input type="text" value="35"/>

Requisitar Adicionar Editar Eliminar

### Interface Item eletrónica

- Inserção dos valores associados à entidade “**Item Electrónica**”  
INSERT INTO PROJECT.ITEM\_ELECTRONICA (COD\_ITEM\_ELECT, COD\_BIB, COD\_TIPO, COD\_FABRICANTE) " &  
"VALUES (@CodItemElect, @CodBiblioteca, @CodTipo, @CodFabricante)
- Selecionar tuplos associados à entidade “**Item Electrónica**”  
SELECT \* FROM PROJECT.ITEM\_ELECTRONICA
- Update dos valores associados à entidade “**Item Electrónica**”  
UPDATE PROJECT.ITEM\_ELECTRONICA " &  
"SET COD\_BIB = @CodBib, " &  
" COD\_Tipo = @CodTipo, " &  
" COD\_FABRICANTE = @CodFabricante " &  
"WHERE COD\_ITEM\_ELECT = @CodIE"
- Delete dos valores associados à entidade “**Item Electrónica**”  
DELETE FROM PROJECT.ITEM\_ELECTRONICA WHERE  
COD\_ITEM\_ELECT=@CodItemElect

Gestão - Biblioteca

Ficheiro Ver

#### Item Electrónica

111641958	459567487	Cód. Item eletrónica:	<input type="text" value="111641958"/>
111826740	489839364	Cod. Biblioteca:	<input type="text" value="459567487"/>
114244766	489839364	Cod. Tipo:	<input type="text" value="148959750"/>
117548760	489839364	Cod. Fabricante:	<input type="text" value="174590652"/>
128273770	459567487	Listar por:	Items encontrados:
133177494	489839364	<input type="text"/>	<input type="text" value="50"/>
136708691	459567487		
146443088	489839364		
161796192	489839364		
167995771	459567487		
172984605	489839364		
179184869	459567487		
181289716	489839364		
181595344	489839364		
194412290	459567487		
201764688	459567487		
245792738	459567487		
279709940	459567487		
288779722	459567487		
291978042	489839364		
318640219	489839364		
321772100	489839364		
327686163	489839364		
330899467	459567487		
372482218	489839364		
408959641	459567487		

Requisitar

Adicionar

Editar

Eliminar

## Interface Requisição Item Papel

- Selecionar tuplos associados à entidade **"Requisição Item Papel"**  
SELECT \* FROM PROJECT.REQUISICAO\_ITEM\_PAPEL
- Inserção dos valores associados à entidade **"Requisição Item Papel"**  
INSERT INTO PROJECT.REQUISICAO\_ITEM\_PAPEL (COD\_REQUISICAO, COD\_ITEM\_PAPEL, COD\_LEITOR, DATA\_REQUISICAO, DT\_EN\_REAL, DT\_EN\_PREVISTA, MULTA) " & "VALUES (@CodRequisicao, @CodItemPapel, @CodLeitor, @Data, @DataReal, @DataPrev, @Multa)
- Update dos valores associados à entidade **"Requisição Item Papel"**  
UPDATE PROJECT.REQUISICAO\_ITEM\_PAPEL " &  
"SET COD\_ITEM\_PAPEL = @CodItemPapel, " &  
" COD\_LEITOR = @CodLeitor, " &  
" DATA\_REQUISICAO = @DataReq, " &  
" DT\_EN\_REAL = @DataReal, " &  
" DT\_EN\_PREVISTA = @DataPrev, " &  
" MULTA = @Multa " &  
"WHERE COD\_REQUISICAO = @CodReq"
- Delete dos valores associados à entidade **"Requisição Item Papel"**  
DELETE FROM PROJECT.REQUISICAO\_ITEM\_PAPEL WHERE  
COD\_REQUISICAO=@CodReqItemPapel "
- Retorna o número total de dinheiro ganho pela biblioteca em multas relativamente a **"Requisição Item Papel"**.  
Select dbo.TotalMoney()

Gestão - Biblioteca

Ficheiro Ver

### Requisição Item Papel

116915680	384315964	7596958	Cód. Item papel:	<input type="text" value="384315964"/>
116947853	774098197	4308521	Cod. Leitor:	<input type="text" value="759695863"/>
119548139	725190922	9904131	Cod. Requesição:	<input type="text" value="116915680"/>
165601835	905939611	9630670	Data Requesição:	<input type="text" value="20 de setembro de 2017"/>
182095056	774098197	8898906	Multa:	<input type="text" value="0,00"/>
199724756	725190922	2320766	Data Entrega Real:	<input type="text" value="15 de outubro de 2017"/>
211446392	725190922	6673677	Data Entrega Prevista:	<input type="text" value="3 de novembro de 2017"/>
214428753	562070871	1166762	Listar por:	<input type="text" value="Items encontrados:"/>
245211704	595182780	6086868		<input type="text" value="34"/>
252370406	342884241	1720034		
266022482	103433106	1166762		
279853609	474268437	8343068		
280562866	774098197	1540137		
293041481	374044458	9630670		
306629962	760647127	3018130		
306754163	103433106	1166762		
332731373	774098197	2279086		
334740166	540069592	3023127		
342423133	103433106	1166762		
410383468	400679020	1166762		
445874605	568137681	4308521		
490788598	342884241	8667054		
501066285	452982806	2404221		
543161673	624872252	1540137		
553746486	316388775	1166762		
578022134	211703536	2287801		

Ganhos em Multas

Adicionar

Editar

Eliminar



## Interface Requisição Item Electrónica

- Selecionar tuplos associados à entidade **"Requisição Item Electrónica"**  
SELECT \* FROM PROJECT.REQUISICAO\_ITEM\_ELECT
- Inserção dos valores associados à entidade **"Requisição Item Electrónica"**  
INSERT INTO PROJECT.REQUISICAO\_ITEM\_ELECT (COD\_REQUISICAO, COD\_ITEM\_ELECT, COD\_LEITOR, MARCACAO, DURACAO, INICIO) " &  
"VALUES (@CodRequisicao, @CodItemElect, @CodLeitor, @Marcacao, @Duracao, @Inicio)
- Update dos valores associados à entidade **"Requisição Item Electrónica"**  
UPDATE PROJECT.REQUISICAO\_ITEM\_ELECT " &  
"SET COD\_ITEM\_ELECT = @CodItemElect, " &  
" COD\_LEITOR = @CodLeitor, " &  
" MARCACAO = @Marcacao, " &  
" DURACAO = @Duracao, " &  
" INICIO = @Inicio " &  
"WHERE COD\_REQUISICAO = @CodReq"
- Delete dos valores associados à entidade **"Requisição Item Electrónica"**  
DELETE FROM PROJECT.REQUISICAO\_ITEM\_ELECT WHERE  
COD\_REQUISICAO=@CodReqItemElect

Gestão - Biblioteca

Ficheiro Ver

### Requisição Item Electrónica

101858355	117548760	8099596	117548760
114401327	372482218	6654070	809959621
121421215	593999042	9486256	101858355
138978020	937384268	9058948	16 de abril de 2031
148853564	146449088	2404221	36
177643705	111641958	1166762	16:37:00
187735734	111826740	5860798	Items encontrados:
206079729	318640219	7362303	42
228783258	111641958	1166762	
231225442	111641958	4308521	
238534086	518641035	1244681	
261273361	291978042	1720034	
283690214	730260886	6993782	
324302552	181595344	1905707	
332754232	146449088	1166762	
332773971	997080750	6545898	
356580687	167995771	4552460	
360509174	925879303	2879701	
418731905	161796192	7431273	
426826230	111641958	1166762	
520506842	111641958	1630292	
530557849	891229258	8343068	
531299056	851701837	7596958	
618561781	111641958	6993782	
642244475	408959641	3018130	
642401061	288779722	6623309	

Adicionar

Editar

Eliminar

## Interface Biblioteca

- Selecionar tuplos associados à entidade “**Biblioteca**”  
`SELECT * FROM PROJECT.BIBLIOTECA`
- Inserção dos valores associados à entidade “**Biblioteca**”  
`INSERT INTO PROJECT.BIBLIOTECA (COD_BIB, NOME, ENDERECO) " &  
"VALUES (@CodBiblioteca, @NomeBiblioteca, @Endereco)`
- Update dos valores associados à entidade “**Biblioteca**”  
`UPDATE PROJECT.BIBLIOTECA " &  
"SET NOME = @NomeBiblioteca, " &  
" ENDERECO = @Endereco " &  
"WHERE COD_BIB = @CodBiblioteca"`
- Delete dos valores associados à entidade “**Biblioteca**”  
`DELETE FROM PROJECT.BIBLIOTECA WHERE COD_BIB=@CodBib`

Gestão - Biblioteca

Ficheiro Ver

### Biblioteca

459567487 Shuffleston  
489839364 MeejoNome: Código: Endereço: Listar por:  
Items encontrados:  

Adicionar

Editar

Eliminar

## Interface Editora

- Selecionar tuplos associados à entidade **"Editora"**  
`SELECT * FROM PROJECT.EDITORA`
- Inserção dos valores associados à entidade **"Editora"**  
`"INSERT INTO PROJECT.EDITORA (COD_EDITORA, NOME, ENDERECO, TELEFONE) " &`  
`"VALUES (@CodEditora, @NomeEditora, @Endereco, @Telefone) "`
- Update dos valores associados à entidade **"Editora"**  
`UPDATE PROJECT.EDITORA " &`  
`"SET NOME = @NomeEditora, " &`  
`" ENDERECO = @Endereco, " &`  
`" TELEFONE = @Telefone " &`  
`"WHERE COD_EDITORA = @CodEditora"`
- Delete dos valores associados à entidade **"Editora"**  
`DELETE FROM PROJECT.EDITORA WHERE COD_EDITORA=@CodEditora "`

Gestão - Biblioteca

Ficheiro Ver

### Editoras

167943717	Eayo
177309404	Ntags
182524111	Vitz
200168965	Photobug
282214846	Pixonyx
299025989	Skynoodle
323204689	Gabtune
332595421	Zoomdog
403852746	Dabshots
425594255	Kimia
467727971	Katz
476804744	Trilith
530032174	Edgeblab
601876949	Thoughtsphere
697620551	Zoomdog
811171250	Jabbertype
815034592	Blogpad
896791081	Mycat
974012405	Mydeo
997491239	Camido

Cód. Editora:	<input type="text" value="167943717"/>
Nome:	<input type="text" value="Eayo"/>
Endereço:	<input type="text" value="0 Melrose Circle"/>
Telefone:	<input type="text" value="332219077"/>
Listar por:	<input type="text" value=""/>
Items encontrados:	<input type="text" value="20"/>

Adicionar

Editar

Eliminar

## Interface Fabricante

- Selecionar tuplos associados à entidade **"Fabricante"**  
`SELECT * FROM PROJECT.FABRICANTE`
- Inserção dos valores associados à entidade **"Fabricante"**  
`INSERT INTO PROJECT.FABRICANTE (COD_FABRICANTE, FABRICANTE, ENDERECO, TELEFONE) " &"VALUES (@CodFab, @Fabricante, @Endereco, @Telefone)`
- Update dos valores associados à entidade Fabricante  
`UPDATE PROJECT.FABRICANTE " &`  
`"SET FABRICANTE = @Fabricante, " &`  
`" ENDERECO = @Endereco, " &`  
`" TELEFONE = @Telefone " &`  
`"WHERE COD_FABRICANTE = @CodFab"`
- Delete dos valores associados à entidade **"Fabricante"**  
`DELETE PROJECT.FABRICANTE WHERE COD_FABRICANTE=@CodFab`

Gestão - Biblioteca

Ficheiro Ver

### Fabricantes

174590652	Skinte
194385028	Cogilith
215556991	Lazz
216803344	Voormm
222605031	Pxoape
284148830	Lajo
302240742	Plajo
347333417	Dynava
348826538	Flashdog
367811486	Lajo
437802750	Zoomlounge
464864323	Jatri
484659799	Tagfeed
520517617	Oyonder
526219895	Skivee
570499148	Yakidoo
623751718	Photobug
722906192	Digitube
730602687	Bubbltube
790618733	Mudo
849817268	Realblab
855076671	Riffwire
902224226	Tavu
970607372	Browsetype

Cód. Fabricante:	<input type="text" value="174590652"/>
Fabricante:	<input type="text" value="Skinte"/>
Endereço:	<input type="text" value="15250 Charing Cross Circle"/>
Telefone:	<input type="text" value="470994336"/>
Listar por:	Items encontrados:
<input type="text" value="v"/>	<input type="text" value="24"/>

Adicionar

Editar

Eliminar

## Interface Tipo

- Selecionar tuplos associados à entidade “**Tipo**”  
`SELECT * FROM PROJECT.TIPO`
- Inserção dos valores associados à entidade “**Tipo**”  
`INSERT INTO PROJECT.TIPO (COD_FABRICANTE, FABRICANTE) " &  
"VALUES (@CodTipo, @Tipo) "`
- Update dos valores associados à entidade tipo  
`UPDATE PROJECT.TIPO " &  
"SET TIPO = @Tipo " &  
"WHERE COD_TIPO = @CodTipo"`
- Delete dos valores associados à entidade “**Tipo**”  
`DELETE FROM PROJECT.TIPO WHERE COD_TIPO=@CodTipo`

Gestão - Biblioteca

Ficheiro Ver

### Tipos

109613135	Atlas
120795219	Mapa
137127978	Computador
138269439	Revista
148959750	Filme
157582769	Dicionário
257702078	Magazine
298410535	Enciclopedia
361054743	CD
532711178	Jornal
659995103	Manual
764917029	Manuscrito
849125117	Livro

Cód. tipo:

109613135

Tipo:

Atlas

Listar por:

Items encontrados:

13

Adicionar

Editar

Eliminar

## Interface Categoria

- Selecionar tuplos associados à entidade “**Categoria**”  
`SELECT * FROM PROJECT.CATEGORIA`
- Inserção dos valores associados à entidade “**Categoria**”  
`INSERT INTO PROJECT.CATEGORIA (COD_CATEGORIA, CATEGORIA) " &  
"VALUES (@CodCategoria, @Categoria)`
- Update dos valores associados à entidade “**Categoria**”  
`UPDATE PROJECT.CATEGORIA " &  
"SET CATEGORIA = @Categoria " &  
"WHERE COD_CATEGORIA = @CodCategoria"`
- Delete dos valores associados à entidade “**Categoria**”  
`DELETE FROM PROJECT.CATEGORIA WHERE COD_CATEGORIA=@CodCategoria "`

Gestão - Biblioteca

Ficheiro Ver

### Categorias

122913963	Comedia
161046158	Suspense
183127204	Dança
198722418	Aventura
248903715	Prosa
491229858	Fantasia
543277800	Animacao
596761513	Policial
739752666	Ficcao cientifica
759517874	Documentario
852852852	SQL
924077881	Drama
955805865	Terror
996867209	Comedia Romantica

Cód. Categoria:

122913963

Categoria:

Comedia

Listar por:

▼

Items encontrados:

14

Adicionar

Editar

Eliminar

