



Universidade de Aveiro
Engenharia de Computadores e Telemática
Base de Dados

Relatório sobre o projeto final

Grupo 7,
Turma Prática 2,
Autores:

- Guilherme Amaral Ribeiro Pereira: 93134
- José Luís Rodrigues Costa: 92996

Contexto

A nossa proposta de trabalho consiste em criar um sistema que permite aos organizadores de eventos musicais planejar da melhor forma os mesmos. Desde poder procurar artistas e locais para o festival, como contratar staff e artistas para os concertos, como escolher patrocinadores que possam reduzir o custo do festival.

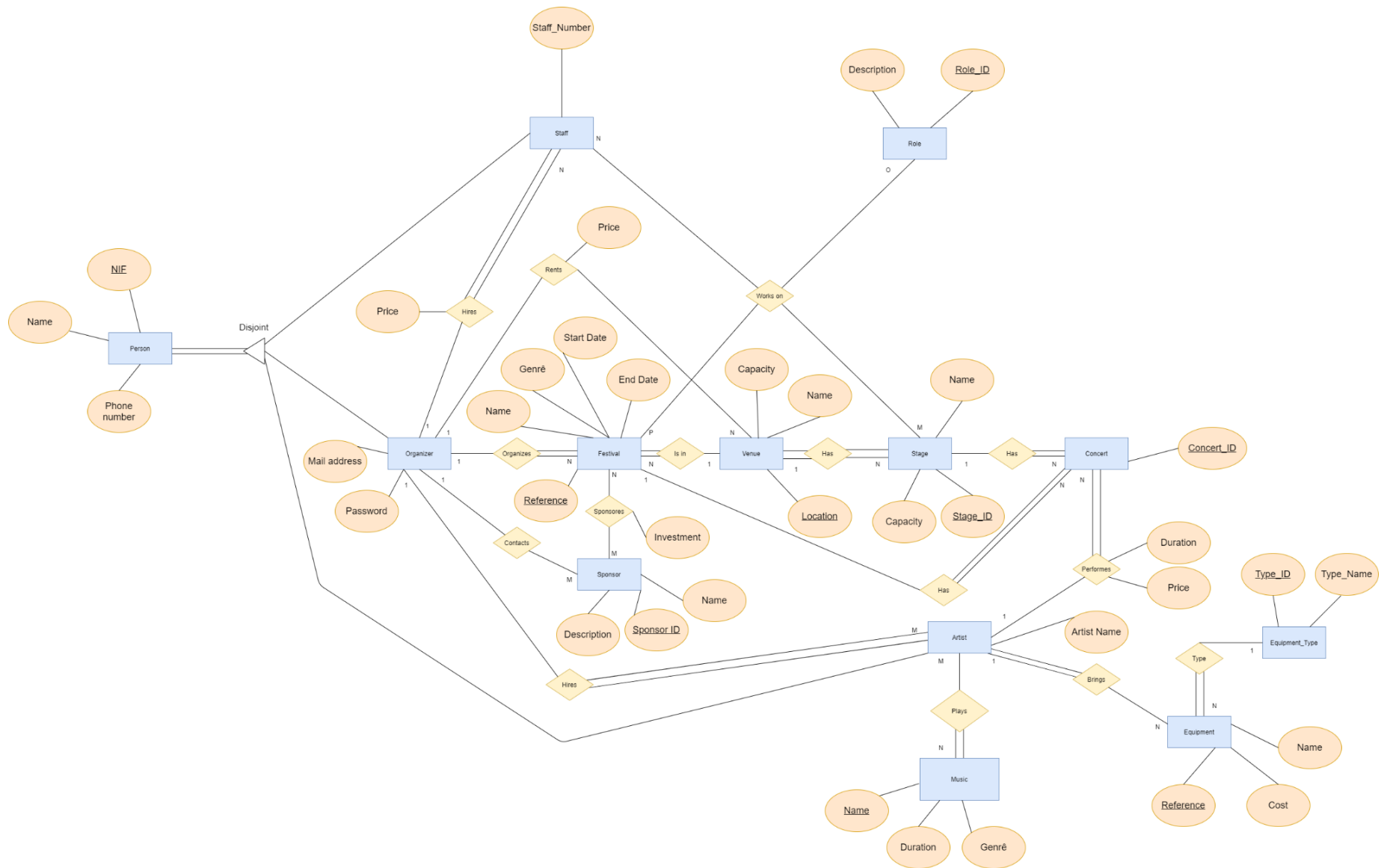
Permitir também que o organizador possa ir acompanhando algumas estatísticas do evento como por exemplo poder consultar o custo que de momento está associado ao festival. As músicas que irão ser tocadas nos concertos etc.

Análise de Requisitos

- Permitir que o organizador possa procurar e escolher o local para o festival.
- Os organizadores devem ter associados a si um NIF, um nome, um número de telemóvel, um endereço de email e uma Password Encriptada.
- O local por sua vez terá associado uma capacidade um nome, uma localização e vários palcos.
- Os palcos têm associados a si um nome, uma capacidade, um ID único e vários concertos que lá irão acontecer.
- Permitir que o organizador crie um festival tendo este associado um local uma data um gênero um nome e patrocinadores, artistas e staff se existirem.
- Permitir ao utilizador procurar e escolher patrocinadores para o festival.

- Os patrocinadores têm a si associados um nome, uma descrição e um ID único. E na relação que tem com o festival terá de estar o valor indicado do patrocínio.
- Permitir que o organizador procure por artistas para associar ao festival.
- Os artistas devem ter a si associados um NIF, um nome, um endereço email, um nome de artista, as músicas que vão tocar nos concertos e o equipamento que traz.
- O equipamento tem um nome, uma referência e um tipo.
- Cada performance do artista num concerto tem uma duração e um preço.
- As músicas devem ter um Nome, uma duração e um género.
- Permitir que o organizador procure por staff para contratar para o festival por um certo preço.
- O staff terá de ter associado a si um NIF, um nome, um número de telefone, um organizador para o qual trabalha e um número de staff.
- O organizador deve poder consultar o custo que no momento está associado ao festival.
- O organizador deve poder num certo momento obter uma lista com as músicas que vão ser tocadas no festival.
- Deve ser possível obter uma lista com o tempo de cada concerto.
- Deve ser possível ao organizador obter informação do equipamento.
- Permitir que o organizador apague um plano de um festival.
- Permite que o organizador reveja as informações de um festival que tenha planeado.

Diagrama entidade Relação



Esquema relacional da base de dados

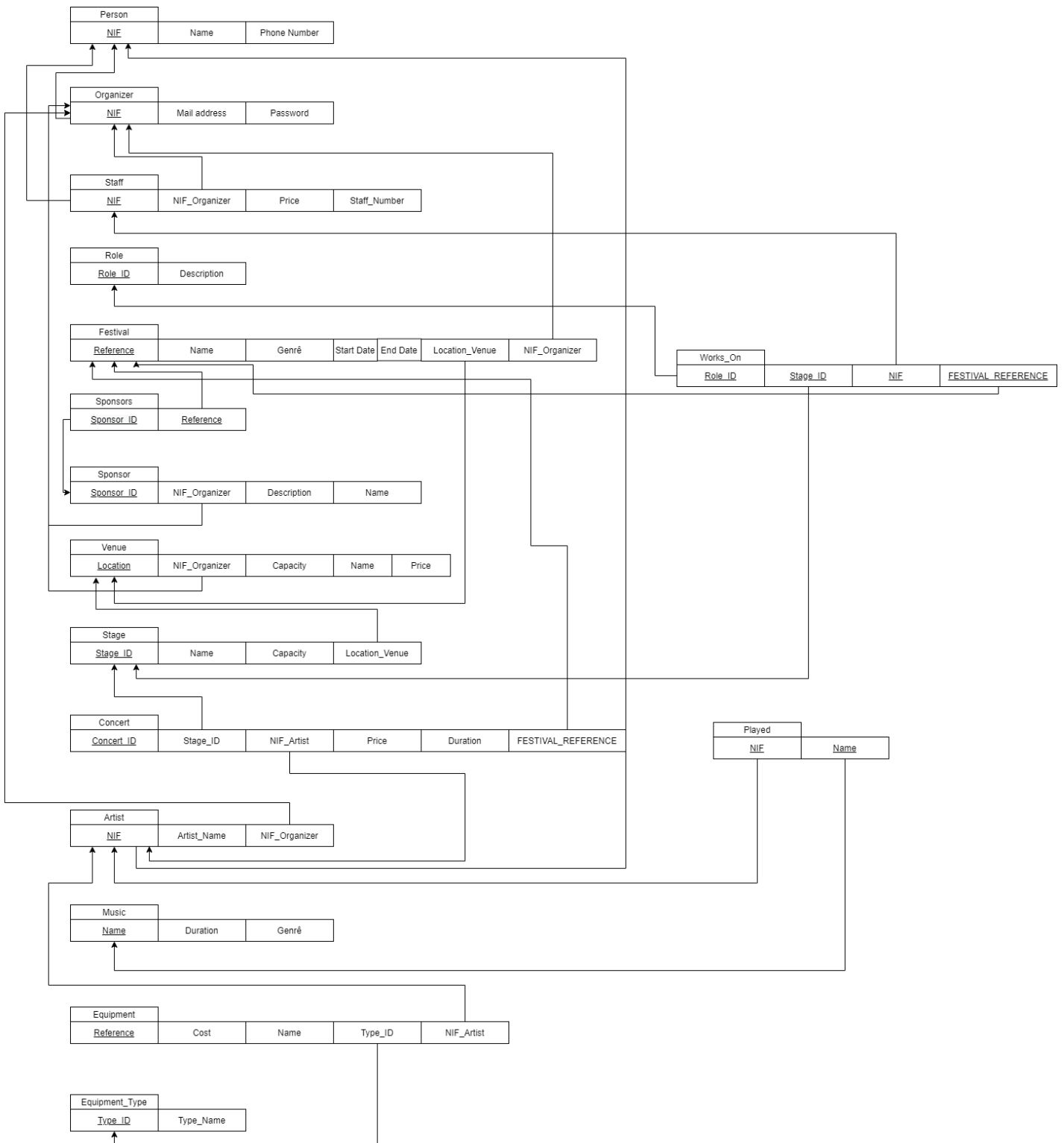
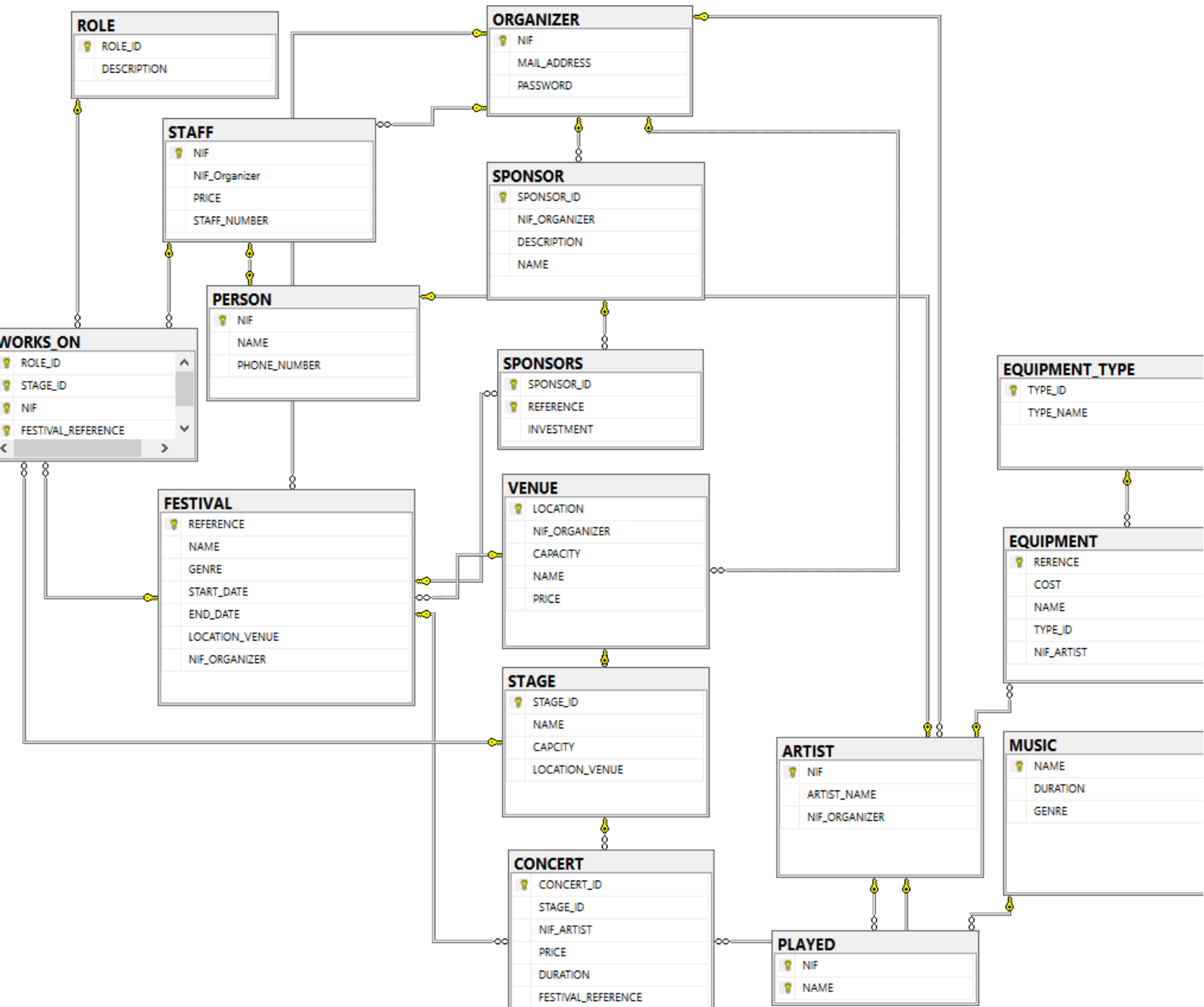


Diagrama da base de dados



SQL Estrutura da base de dados

```
CREATE TABLE PERSON (  
    NIF                CHAR(9)          NOT NULL,  
    [NAME]             VARCHAR(256)     NOT NULL,  
    PHONE_NUMBER       INT              NOT NULL,  
    Constraint PERSON_PK PRIMARY KEY (NIF),  
    Constraint PERSON_PH UNIQUE (PHONE_NUMBER )  
);
```

```
CREATE TABLE ORGANIZER(  
    NIF                CHAR(9)          NOT NULL,  
    MAIL_ADDRESS       VARCHAR(256)     NOT NULL,  
    [PASSWORD]         VARBINARY(128)   NOT NULL,  
    Constraint ORGANIZER_PK PRIMARY KEY(NIF),  
    Constraint ORGANIZER_UQ UNIQUE (MAIL_ADDRESS)  
);
```

```
CREATE TABLE STAFF(  
    NIF                CHAR(9)          NOT NULL,  
    NIF_Organizer       CHAR(9)          NOT NULL,  
    PRICE              INT              NOT NULL ,  
    STAFF_NUMBER       INT              NOT NULL,  
    Constraint STAFF_PK PRIMARY KEY(NIF),  
    Constraint CheckPrice CHECK(PRICE > 0),  
    FOREIGN KEY (NIF) REFERENCES PERSON(NIF),  
    FOREIGN KEY (NIF_Organizer) REFERENCES ORGANIZER(NIF)  
);
```

```
CREATE TABLE [ROLE](  
    ROLE_ID            INT              NOT NULL,  
    [DESCRIPTION]      VARCHAR(256)     NOT NULL,  
    Constraint ROLE_ID_PK PRIMARY KEY(ROLE_ID)  
);
```

```

CREATE TABLE VENUE(
    [LOCATION]          VARCHAR(256)          NOT NULL,
    NIF_ORGANIZER      CHAR(9)                ,
    CAPACITY           INT                    NOT NULL,
    [NAME]             VARCHAR(256)          NOT NULL,
    PRICE              INT                    NOT NULL,
    CONSTRAINT VENUE_PK PRIMARY KEY(LOCATION),
    CONSTRAINT CheckPriceVenue CHECK(PRICE > 0),
    CONSTRAINT CheckCapacityVenue CHECK(CAPACITY > 0),
    CONSTRAINT VENUE_FK_ORGANIZER FOREIGN KEY(NIF_ORGANIZER) REFERENCES
ORGANIZER(NIF)
);

```

```

CREATE TABLE FESTIVAL(
    REFERENCE          CHAR(9)                NOT NULL,
    [NAME]             VARCHAR(256)          NOT NULL,
    GENRE              VARCHAR(128)          NOT NULL,
    [START_DATE]       DATETIME               NOT NULL,
    END_DATE           DATETIME               NOT NULL,
    LOCATION_VENUE     VARCHAR(256)          NOT NULL,
    NIF_ORGANIZER      CHAR(9)                NOT NULL,
    Constraint FESTIVAL_PK PRIMARY KEY(REFERENCE),
    Constraint FESTIVAL_FK_ORGANIZER FOREIGN KEY(NIF_ORGANIZER)
REFERENCES ORGANIZER(NIF),
    Constraint FESTIVAL_FK_VENUE FOREIGN KEY(LOCATION_VENUE) REFERENCES
VENUE(LOCATION)
);

```

```

CREATE TABLE ARTIST(
    NIF                CHAR(9)                NOT NULL,
    ARTIST_NAME        VARCHAR(128)          NOT NULL,
    NIF_ORGANIZER      CHAR(9)                NOT NULL,
    Constraint P_NIF_PK PRIMARY KEY(NIF),
    Constraint P_NIF_FK FOREIGN KEY (NIF) REFERENCES PERSON(NIF),
    Constraint O_NIF_FK FOREIGN KEY (NIF_ORGANIZER) REFERENCES
ORGANIZER(NIF)
);

```

```

CREATE TABLE EQUIPMENT_TYPE(
    [TYPE_ID]          INT                    NOT NULL,
    [TYPE_NAME]        VARCHAR(128)          NOT NULL,
    Constraint TYPE_ID_PK PRIMARY KEY([TYPE_ID]));

```



```

CREATE TABLE EQUIPMENT(
    RERENCE          INT                NOT NULL,
    COST              VARCHAR(128)       NOT NULL,
    [NAME]            VARCHAR(128)       NOT NULL,
    [TYPE_ID]         INT                NOT NULL,
    NIF_ARTIST        CHAR(9)            ,
    Constraint RERENCE_PK PRIMARY KEY(RERENCE),
    Constraint TYPE_ID_FK FOREIGN KEY ([TYPE_ID]) REFERENCES
EQUIPMENT_TYPE([TYPE_ID]),
    CONSTRAINT CheckPriceEquipment CHECK(COST > 0),
    Constraint ART_NIF_FK FOREIGN KEY (NIF_ARTIST) REFERENCES
ARTIST(NIF)
);

```

```

CREATE TABLE MUSIC(
    [NAME]            VARCHAR(256)       NOT NULL,
    DURATION           TIME              NOT NULL,
    GENRE              VARCHAR(128)       NOT NULL,
    Constraint NAME_NIF_PK PRIMARY KEY([NAME])
);

```

```

CREATE TABLE STAGE(
    STAGE_ID           INT                NOT NULL,
    [NAME]            VARCHAR(256)       NOT NULL,
    CAPCITY            INT                NOT NULL,
    LOCATION_VENUE     VARCHAR(256)       NOT NULL,
    CONSTRAINT STAGE_PK PRIMARY KEY(STAGE_ID),
    CONSTRAINT CheckCapacityStage CHECK(CAPCITY > 0),
    CONSTRAINT STAGE_FK_VENUE FOREIGN KEY(LOCATION_VENUE) REFERENCES
VENUE(LOCATION)
);

```

```

CREATE TABLE WORKS_ON(
    ROLE_ID            INT                NOT NULL,
    STAGE_ID           INT                NOT NULL,
    NIF                CHAR(9)           NOT NULL,
    FESTIVAL_REFERENCE CHAR(9)           NOT NULL,
    Constraint WORKS_ON_PK PRIMARY
KEY(ROLE_ID,STAGE_ID,NIF,FESTIVAL_REFERENCE),
    CONSTRAINT WORKS_ON_FK_ROLE_ID FOREIGN KEY(ROLE_ID) REFERENCES
[ROLE](ROLE_ID),

```

```

        CONSTRAINT WORKS_ON_FK_STAGE_ID FOREIGN KEY(STAGE_ID) REFERENCES
STAGE(STAGE_ID),
        CONSTRAINT WORKS_ON_FK_NIF FOREIGN KEY(NIF) REFERENCES STAFF(NIF),
        CONSTRAINT WORKS_ON_FK_FESTIVAL FOREIGN KEY (FESTIVAL_REFERENCE)
REFERENCES FESTIVAL(REFERENCE)

);

```

```

CREATE TABLE CONCERT(
    CONCERT_ID      INT          NOT NULL,
    STAGE_ID        INT          NOT NULL,
    NIF_ARTIST      CHAR(9)      NOT NULL,
    PRICE           INT          NOT NULL,
    DURATION        TIME         NOT NULL,
    FESTIVAL_REFERENCE CHAR(9) NOT NULL,
    CONSTRAINT CONCERT_PK PRIMARY KEY(CONCERT_ID),
    CONSTRAINT CONCERT_FK_STAGE FOREIGN KEY(STAGE_ID) REFERENCES
STAGE(STAGE_ID),
    CONSTRAINT CONCERT_FK_ARTIST FOREIGN KEY(NIF_ARTIST) REFERENCES
ARTIST(NIF),
    CONSTRAINT CheckConcertPrice CHECK(PRICE > 0),
    CONSTRAINT CONCERT_FK_FESTIVAL FOREIGN KEY(FESTIVAL_REFERENCE)
REFERENCES FESTIVAL(REFERENCE)
);

```

```

CREATE TABLE PLAYED(
    NIF              CHAR(9)      NOT NULL,
    [NAME]           VARCHAR(256) NOT NULL,
    CONSTRAINT PLAYED_PK PRIMARY KEY(NIF,[NAME]),
    CONSTRAINT PLAYED_FK_NIF FOREIGN KEY(NIF) REFERENCES ARTIST(NIF),
    CONSTRAINT PLAYED_FK_NAME FOREIGN KEY([NAME]) REFERENCES
MUSIC([NAME]),
);

```

```

CREATE TABLE SPONSOR(
    SPONSOR_ID      INT          NOT NULL,
    NIF_ORGANIZER    CHAR(9)      ,
    [DESCRIPTION]    VARCHAR(256) NOT NULL,
    [NAME]           VARCHAR(256) NOT NULL,
    CONSTRAINT SPONSOR_PK PRIMARY KEY(SPONSOR_ID),
    CONSTRAINT SPONSOR_FK_ORGANIZER FOREIGN KEY(NIF_ORGANIZER)
REFERENCES ORGANIZER(NIF)
);

```

```
CREATE TABLE SPONSORS(  
    SPONSOR_ID          INT                NOT NULL,  
    REFERENCE           CHAR(9)            NOT NULL,  
    INVESTMENT          INT                NOT NULL,  
    CONSTRAINT SPONSORS_PK PRIMARY KEY(SPONSOR_ID,REFERENCE),  
    CONSTRAINT SPONSORS_FK_REF FOREIGN KEY(REFERENCE) REFERENCES  
FESTIVAL(REFERENCE),  
    CONSTRAINT CheckSponsorsInvestment CHECK (INVESTMENT > 0),  
    CONSTRAINT SPONSOR_ID FOREIGN KEY(SPONSOR_ID) REFERENCES  
SPONSOR(SPONSOR_ID),  
);
```

User Defined Functions

No desenvolvimento deste projeto foram criadas dezassete UDFs maioritariamente com o propósito de efetuar consultas a base de dados, no entanto também foram criadas udfs que retornam valores como por exemplo o custo associado ao festival num determinado momento. (Podem ser vistas no ficheiro FESTIVAL_ORGANIZER_UDF.sql ou no scriptBD).

A lista das UDFs e uma pequena descrição do que elas fazem pode ser vista na seguinte tabela:

SearchVenue	Procurar por Locais para o Festival
SearchFestival	Procurar os Festivais que o organizador já criou
SearchSponsor	Procurar por patrocínios contratados pelo organizador para associar ao festival
GetOrganizerInfo	Pedir a informação sobre o organizador
GetStaffInfo	Procurar pela informação dos staffs associados ao organizador
GetStages	Procurar os palcos que estão num local
GetArtists	Devolve os artistas associados ao organizador que ainda podemos associar ao festival.
GetArtistMusic	Devolve as músicas que um artista tem
GetPriceAll	Devolve o custo que de momento está associado ao festival, custo do local do staff dos artistas e o desconto dos sponsors
CheckCredentials	Verifica se a palavra passe está correta

GetFestivalMusics	Devolve todas as músicas associadas de momento ao festival
FestivalGetAllStaff	Devolve todos os staffs associados a um festival
FestivalGetAllArtists	Devolve todos os artistas associados a um festival
GetFestivalEquipment	Devolve a lista de equipamentos que está associada ao festival, ou seja os equipamentos associados aos artistas do festival
GetFestivalStages	Devolve os palcos associados a um festival
GetFestivalSponsors	Devolve os Patrocinadores associados a um festival
GetFestivalMainInfo	Devolve a informação de mais alto nível sobre o festival como por exemplo o nome as datas e o local

Stored Procedures

Desenvolvemos também as seguintes stored procedures que foram criadas para adicionar, remover ou alterar dados na nossa base de dados. (Podem ser vistas no ficheiro FESTIVAL_ORGANIZER_SP.sql ou no scriptBD).

InsertOrganizer	Insere na base de dados um organizador que é pessoa (Transação)
InsertStaff	Insere na base de dados um staff que é pessoa associado a um organizador (Transação)
InsertArtist	Insere na base de dados um artista que é pessoa associado a um

	organizador (Transação)
InsertSponsor	Insere na base de dados um Patrocinador associado a um organizador
SponsorFestival	Cria a ligação entre o patrocinador e o festival com um custo
InsertMusic	Insere na base de dados uma música e insere a ligação entre o artista e a música (transação)
InsertEquipment	Insere um equipamento na base de dados associado ao artista com um certo tipo de equipamento
DeleteFestival	Apaga um festival, todas as dependências associadas ao mesmo (transação)
StaffWorksOn	Insere na base de dados a relação entre o staff o onde em que palco do festival e com que role ele irá trabalhar
UpdateVenue	Associa a venue ao organizador quando este a aluga
AddConcert	Adicionar um concerto ao festival
AddOrganizer	Insere na base de dados um organizador com a palavra passe encriptada (Transação)
InsertFestival	Insere na base de dados um festival

Triggers

Foram também desenvolvidos quatro triggers pois é necessário haver algum controlo no que toca a contratar pessoas à mesma hora que não faria sentido, ou planear um festival num local onde já iria acontecer outro entre essas datas. (Podem ser vistos no ficheiro FESTIVAL_ORGANIZER_Triggers.sql ou no scriptBD).

A tabela seguinte apresenta os quatro triggers e respetiva descrição:

CheckVenueAvailable	Verifica se a data de início e a data de fim são válidas. Se não são numa data que já passou ou se não há conflito entre as duas.
CheckVenueAlreadyOccupied	Verifica se não existe já um festival planeado para um local nos mesmos dias.
CheckStaffAlreadyWorking	Verifica se um staff já está a trabalhar noutro festival que acontece à mesma hora.
CheckFestivalConcertSchedule	Verifica se a duração dos concertos num certo palco não ultrapassa a duração do festival

Views

As views existentes foram desenvolvidas com o propósito de oferecer uma camada de proteção entre a base de dados e aplicação mas também para otimizar o processo de pesquisa. (Podem ser vistas no ficheiro FESTIVAL_ORGANIZER_VIEWS.sql ou no script_BD).

SearchOrganizer	View com todos os organizadores na base de dados
SearchEquipmentType	View com todo o tipo de equipamentos existentes

SearchRole	View com todo o tipo de Roles de staff existentes
-------------------	---

Indexes

Os índices foram criados nas nos atributos que eram mais pesquisados de forma a acelerar o processo de pesquisa.

(Podem ser vistas no ficheiro FESTIVAL_ORGANIZER_INDEX.sql ou no scriptBD).

ixStaffForeing	NONCLUSTERED
ixStageVenue	NONCLUSTERED
ixVenueOrganizer	NONCLUSTERED
ixFestivalOrganizer	NONCLUSTERED
ixConcertFestival	NONCLUSTERED
ixSponsor	NONCLUSTERED

Transações

Foram usadas transações para manter a integridade dos dados dentro das nossas Stored procedures que faziam mais do que uma instrução de insert update e delete.

Notas Sobre a projeto

Existe na base de dados do grupo um organizador que já tem associado artistas staff festivais etc, caso seja necessário testar as funcionalidades sem

criar um novo organizador e ter de inserir os valores através da interface. esse organizador é o Guilherme Costa e a palavra passe é teste.

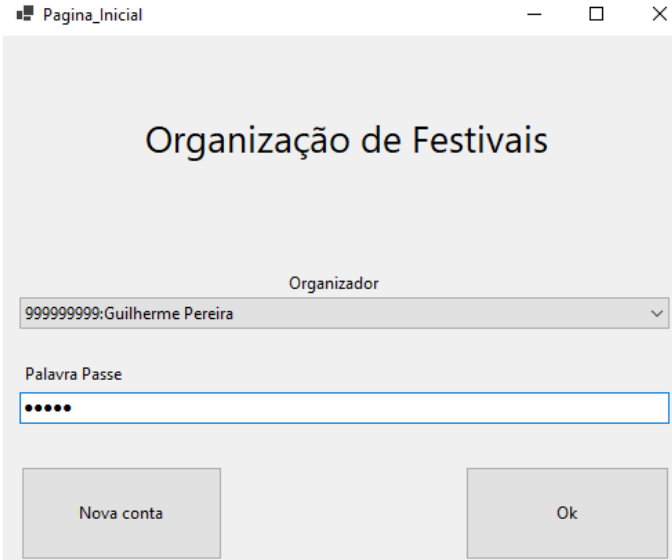
Outra nota é que a aplicação está-se a ligar à base de dados do grupo no entanto pode ser alterado mudando a string de conexão que se encontra no ficheiro Class1.cs que se encontra no código fonte da interface.

Pode se replicar o projeto através do scriptBD.sql disponível no ficheiro de entrega, e também por motivos de visualização existem à parte ficheiros sql que contém as tabelas, os índices, as views, os triggers, as UDFs e as SP.

Overview de alguns formulários da interface

Na seguinte secção podemos ver algumas das 15 interfaces do projeto (que foram desenvolvidas em C#) que contém as features mais relevantes

Um vídeo demonstrativo da aplicação a funcionar pode ser encontrado no seguinte [link](#).



Pagina_Inicial

Organização de Festivais

Organizador

999999999:Guilherme Pereira

Palavra Passe


.....

Nova conta

Ok

Página Inicial de login do organizador (Na Base de dados do grupo existe um Organizador Guilherme Costa cuja palavra passe é teste que já contém festivais planeados etc).

Página_organizador



Nome

Guilherme Pereira

Número de telefone

933123321

email

guilherme@ua.pt

Editar

Novo Festival

Contratar Staff

Contratar Artista

Adicionar Patrocinador

Apagar um festival

Ver cartaz de um festival

Obter informações sobre os festivais

Página Inicial do organizador onde podemos escolher quais operações realizar.

Página_Festivais

Novo Festival

Referencia (9 digitos)

999777555

Genero Musical

EDM

Nome

Second Sky

Data Inicio (ano-dia-mes)

2027-01-01

Hora de Inicio

12:00:00

Data Fim (ano-dia-mes)

2027-01-02

Hora de fim

12:00:00

Local

R. das Portas de Santo Antão 96, 1150-269 Lisboa

Arrendar Local

Sponsor

Sponsor ID: 89, Nome: Fanta Descrição: Produtor de bebidas refrescantes

Adicionar Patrocinios

Staff

NIF: 192837465, Telemóvel: 987654322, Nome: Guilherme Staff, Preço: 700 Palco: 3 Tarefa: Cleaning

Adicionar Staff

Concerto

Criar Concerto

Palcos

Id Palco	Nome	Capacidade
3	Palco Secundário	1000
4	Palco Principal	3300

Ok

Ver Cartaz Atual

Preço total

1300

Atualizar

Página de planeamento de festivais onde podemos alugar palcos escolher staff escolher patrocinadores criar concertos ver o cartaz de músicas e o preço atual do festival.

Página_Festival_Info

Ver Informação sobre um festival

666555888:Secret Sky

Informação Geral

Staff

Artistas

Equipamentos

Palcos

Patrocínios

NOME: Palco Principal
ID: 1
Capacidade: 15000

NOME: Palco Secundário
ID: 2
Capacidade: 5000

Página que permite obter as informações sobre todos os festivais que foram planeados pelo organizador.