

# Cadeia de Discotecas

Relatório do projeto de BDAD

Turma 3, Grupo 4

André Lino dos Santos (up201907879)

Edgar Ferreira da Torre (up201906573)

Maria José Valente da Silva Carneiro (up201907726)

Abril 2021

## Conteúdo

<b>A</b>	<b>Definição do modelo conceptual</b>	<b>3</b>
A.1	Descrição do modelo conceptual . . . . .	3
A.2	Diagrama de Classes em UML . . . . .	5
A.3	Diagrama de Classes em UML - revisto . . . . .	6
<b>B</b>	<b>Definição do esquema relacional</b>	<b>7</b>
<b>C</b>	<b>Análises de Dependências Funcionais e Formas Normais</b>	<b>9</b>
<b>D</b>	<b>Restrições</b>	<b>13</b>

## A Definição do modelo conceptual

### A.1 Descrição do modelo conceptual

(Alterações efetuadas à descrição após a revisão encontram-se em *italico*)

O modelo conceptual que apresentamos baseia-se na gestão de uma cadeia de discotecas.

De cada discoteca é necessário guardar o seu nome, proprietário, a sua localização e área, *que deve ser maior que a soma das áreas dos espaços interiores. Para a frequentar, cada pessoa, da qual interessa saber o número do BI, nome, número de telemóvel e idade, deverá ser membro da mesma, guardando-se o seu número de membro e tipo de afiliação, isto é, se é um cliente VIP ou não.* É de realçar que os membros têm de ser maiores de idade.

Cada discoteca é composta por vários espaços: lounges, caixas de pagamento, bengaleiros e pistas, sendo todos identificados pelo seu id. De todos os espaços interessa também saber a respetiva área. No caso dos bengaleiros, deve-se guardar o número máximo de casacos e o preço por casaco; nas caixas de pagamento, o dinheiro em cada caixa e nas pistas, o nome e género de música (*funk, trance, 90s, reggaeton, kizomba, house*).

Os membros da discoteca com afiliação especial (VIP) poderão efetuar reservas para a utilização do lounge. Dessa reserva interessa saber a dia, a hora, o número de garrafas e o número de sofás a reservar (de 0 a 5).

*Relativamente aos artistas, é necessário saber o nome, o número de telemóvel, o respetivo cachê e o seu tipo (se é residente ou convidado). Caso os artistas sejam convidados, é importante registar a hora de começo e término da atuação e a sua duração, que não deverá ser menor que 30 minutos, nem maior que 4 horas. Estes podem atuar em várias pistas. Cada pista pode ter ainda um artista residente, que não tem atuação fixa (atuam quando não há nenhum artista convidado em pista).*

As pistas agregam bares, identificados por um id, aos quais estarão associados bebidas, das quais interessa saber o nome, marca, stock (*entre 50 e 2400*), preço e teor alcoólico. *O preço das bebidas varia com o teor alcoólico: se for menor que 1.2, o preço tem de ser menor que 4 euros; se for maior que 1.2, o preço varia entre 4 euros e 50 euros.*

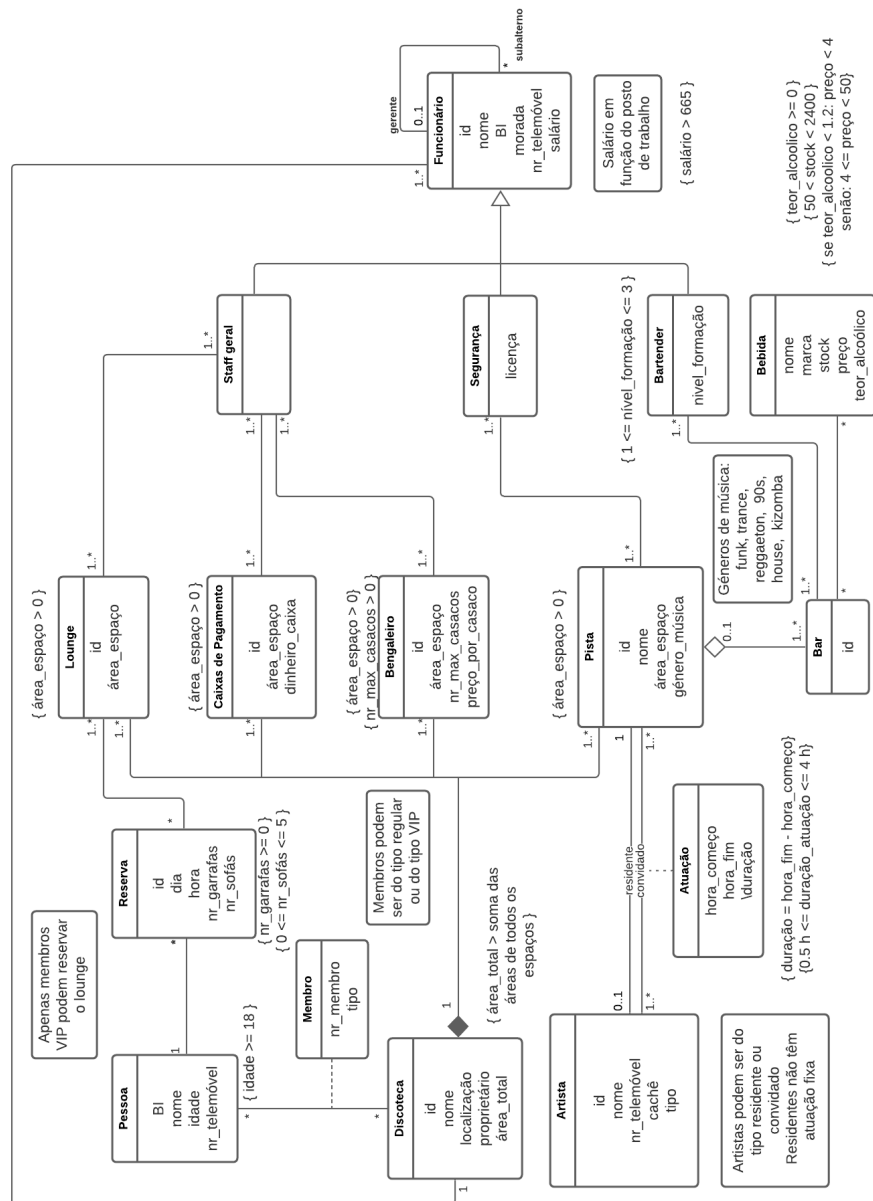
Em todos os espaços existem funcionários de diferentes tipos (staff geral, seguranças e bartenders) que respondem às necessidades existentes de cada um. De cada funcionário guarda-se o nome, morada, salário (que varia com a função do trabalhador e *tem de ser maior que o salário mínimo, que é 665€*), número de telemóvel e de cartão de cidadão. Existe uma hierarquia de funcionários, que

podem ser gerentes ou os seus subalternos.

No caso particular dos seguranças, para desempenharem a sua função, é necessário a posse de uma licença. Estes atuam nas pistas e, consequentemente nos seus bares, onde também trabalham os bartenders. Destes é registado o seu nível de formação, *entre 1 e 3*. O staff geral distribui-se pelos bengaleiros, lounge e pelas caixas de pagamento.



## 6



## B Definição do esquema relacional

**Pessoa** (BI, nome, idade, nrTelemóvel)

**Discoteca** (id, nome, localização, proprietário, áreaTotal)

**Membro** (BI  $\rightarrow$  Pessoa, idDiscoteca  $\rightarrow$  Discoteca, nrMembro, tipo)  
tipo  $\in \{ \text{"regular"}, \text{"VIP"} \}$

**Reserva** (id, dia, hora, nrGarrafas, nrSofás, BI  $\rightarrow$  Pessoa)

**Lounge** (id, áreaEspaço, idDiscoteca  $\rightarrow$  Discoteca)

**ReservaLounge** (idReserva  $\rightarrow$  Reserva, idLounge  $\rightarrow$  Lounge)

**CaixasPagamento** (id, áreaEspaço, dinheiroCaixa, idDiscoteca  $\rightarrow$  Discoteca)

**Bengaleiro** (id, áreaEspaço, nrMaxCasacos, preçoCasaco, idDiscoteca  $\rightarrow$  Discoteca)

**Artista** (id, nome, nrTelemóvel, cachê, tipo)  
tipo  $\in \{ \text{"convidado"}, \text{"residente"} \}$

**Pista** (id, nome, áreaEspaço, géneroMúsica, idResidente  $\rightarrow$  Artista, idDiscoteca  $\rightarrow$  Discoteca)  
géneroMúsica  $\in \{ \text{"funk"}, \text{"trance"}, \text{"house"}, \text{"90s"}, \text{"kizomba"}, \text{"reggaeton"} \}$

**Atuação** (idArtista  $\rightarrow$  Artista, idPista  $\rightarrow$  Pista, horaComeço, horaFim, duração)

**Bar** (id, idPista  $\rightarrow$  Pista)

**Bebida** (nome, marca, stock, preço, teorAlcoólico)

**BarBebida** (idBar  $\rightarrow$  Bar,  $\{ \text{nome}, \text{marca} \}$   $\rightarrow$  Bebida)

**Funcionário** (id, nome, nrTelemóvel, BI, morada, salário, idDiscoteca  $\rightarrow$  Discoteca)

**Hierarquia** (idSubalterno  $\rightarrow$  Funcionário, idGerente  $\rightarrow$  Funcionário)

**StaffGeral** (id  $\rightarrow$  Funcionário)

**Segurança** (id  $\rightarrow$  Funcionário, licença)

**Bartender** (id  $\rightarrow$  Funcionário, nívelFormação)

**LoungeStaffGeral** (idStaff  $\rightarrow$  StaffGeral, idLounge  $\rightarrow$  Lounge)

**CaixasStaffGeral** (idStaff  $\rightarrow$  StaffGeral, idCaixa  $\rightarrow$  Caixas)

**BengaleiroStaffGeral** (idStaff  $\rightarrow$  StaffGeral, idBengaleiro  $\rightarrow$  Bengaleiro)

**BarBartender** (idBartender  $\rightarrow$  Bartender, idBar  $\rightarrow$  Bar)

**PistaSegurança** (idSegurança  $\rightarrow$  Segurança, idPista  $\rightarrow$  Pista)



## C Análises de Dependências Funcionais e Formas Normais

### Pessoa

$$\{BI\} \rightarrow \{nome, idade, nrTelemóvel\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

### Discoteca

$$\begin{aligned} \{id\} &\rightarrow \{nome, localização, proprietário, áreaTotal\} \\ \{nome, localização\} &\rightarrow \{id\} \end{aligned}$$

As dependências funcionais não triviais são do tipo  $\overline{A} \rightarrow \overline{B}$  e  $\overline{A}$  é uma superchave em ambos os casos: na primeira dependência,  $\overline{A}$  é a chave mínima (em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação) e na segunda é uma superchave dado que  $\{nome, localização\}^+ = \{nome, localização, id\}^+ = \{nome, localização, id, proprietário, áreaTotal\}$ . Deste modo, a relação encontra-se na BCNF e, consequentemente, na 3NF.

### Membro

$$\begin{aligned} \{BI, idDiscoteca\} &\rightarrow \{nrMembro, tipo\} \\ \{nrMembro, idDiscoteca\} &\rightarrow \{BI\} \end{aligned}$$

As dependências funcionais não triviais são do tipo  $\overline{A} \rightarrow \overline{B}$  e  $\overline{A}$  é uma superchave em ambos os casos: na primeira dependência,  $\overline{A}$  é também uma superchave (em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação) e na segunda é uma superchave dado que  $\{nrMembro, idDiscoteca\}^+ = \{nrMembro, idDiscoteca, BI\}^+ = \{nrMembro, idDiscoteca, BI, tipo\}$ . Deste modo, a relação encontra-se na BCNF e, consequentemente, na 3NF.

### Reserva

$$\{id\} \rightarrow \{dia, hora, nrGarrafas, nrSofas, BI\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

### ReservaLounge

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

### CaixasPagamento

$$\{id\} \rightarrow \{áreaEspaço, dinheiroCaixa, idDiscoteca\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, conseqüentemente, na 3NF.

### Bengaleiro

$$\{id\} \rightarrow \{áreaEspaço, nrMaxCasacos, preçoCasaco, idDiscoteca\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, conseqüentemente, na 3NF.

### Artista

$$\{id\} \rightarrow \{nome, nrTelemóvel, cachê, tipo\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, conseqüentemente, na 3NF.

### Pista

$$\begin{aligned} \{id\} &\rightarrow \{nome, áreaEspaço, géneroMúsica, idResidente, idDiscoteca\} \\ \{nome, idDiscoteca\} &\rightarrow \{id\} \\ \{residente\} &\rightarrow \{id\} \end{aligned}$$

As dependências funcionais não triviais são do tipo  $\overline{A} \rightarrow \overline{B}$  e  $\overline{A}$  é uma superchave em todos os casos: na primeira dependência,  $\overline{A}$  é uma chave mínima (em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação); na segunda, é uma superchave dado que  $\{nome, idDiscoteca\}^+ = \{nome, idDiscoteca, idPista\}^+ = \{nome, idDiscoteca, idPista, áreaEspaço, géneroMúsica, residente\}$  e na terceira é uma chave mínima dado que  $\{residente\}^+ = \{residente, idPista\}^+ = \{residente, idPista, nome, idDiscoteca, idPista, áreaEspaço, géneroMúsica, residente\}$ . Deste modo, a relação encontra-se na BCNF e, conseqüentemente, na 3NF.

### Atuação

$$\begin{aligned} \{idArtista, idPista\} &\rightarrow \{horaComeço, horaFim, duração\} \\ \{horaComeço, horaFim\} &\rightarrow \{duração\} \end{aligned}$$

A relação respeita a 1NF dado que os valores de todos os atributos são atômicos. Quanto à 2NF, na primeira dependência não trivial em que  $\overline{A} \rightarrow \overline{B}$ , temos que

$\overline{A}$  é chave candidata devido ao seu fecho conter todos os atributos da relação. Na segunda dependência não trivial,  $\{horaComeço, horaFim\}$  não pode ser considerado um subconjunto de uma chave e o atributo  $\{duração\}$  não é primo. Assim, podemos concluir que a relação respeita a 2NF.

Quanto à 3NF, dado que na segunda dependência funcional não trivial, em que  $\overline{A} \rightarrow \overline{B}$ ,  $\overline{A}$  não é uma superchave nem  $\overline{B}$  tem apenas atributos primos, então a relação viola a 3NF e, consequentemente, a BCNF.

Isto deve-se ao atributo *duração* ser um elemento derivado dos atributos *horaComeço* e *horaFim*, pelo que não nos parece vantajoso em termos de design decompor esta relação, de modo que esta violação irá ser ignorada.

### Bar

$$\{id\} \rightarrow \{idPista\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

### Bebida

$$\{nome, marca\} \rightarrow \{stock, preço, teorAlcoólico\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

### BarBebida

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

### Funcionário

$$\begin{aligned} \{id\} &\rightarrow \{nome, nrTelemóvel, BI, morada, salário, idDiscoteca\} \\ \{BI, idDiscoteca\} &\rightarrow \{id\} \end{aligned}$$

As dependências funcionais não triviais são do tipo  $\overline{A} \rightarrow \overline{B}$  e  $\overline{A}$  é uma superchave (chave mínima) em ambos os casos: na primeira dependência, o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação e na segunda o fecho é  $\{BI, idDiscoteca\}^+ = \{BI, idDiscoteca, id\}^+ = \{id, BI, morada, nrTelemóvel, salário, idDiscoteca\}$ . Deste modo, a relação encontra-se na BCNF e, consequentemente, na 3NF.

### Hierarquia

$$\{idSubalterno\} \rightarrow \{idGerente\}$$

A dependência funcional é do tipo  $\overline{A} \rightarrow \overline{B}$  e, como  $\overline{B} \subseteq \overline{A}$ , dado que gerentes também podem ser subalternos de outros gerentes, então a dependência funcional é trivial pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **StaffGeral**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **Segurança**

$$\begin{aligned}\{\text{id}\} &\rightarrow \{\text{licença}\} \\ \{\text{licença}\} &\rightarrow \{\text{id}\}\end{aligned}$$

As dependências funcionais não triviais são do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

#### **Bartender**

$$\{\text{id}\} \rightarrow \{\text{nívelFormação}\}$$

A dependência funcional não trivial é do tipo  $\overline{A} \rightarrow \overline{B}$  e como  $\overline{A}$  é sempre uma superchave (chave mínima, em que o fecho de  $\overline{A}$  inclui diretamente todos os atributos da relação), a relação encontra-se na BCNF e, consequentemente, na 3NF.

#### **LoungeStaffGeral**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **CaixasStaffGeral**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **BengaleiroStaffGeral**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **BarBartender**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

#### **PistaSegurança**

Esta relação não tem dependências funcionais não triviais pelo que se encontra na BCNF e, consequentemente, também se encontra na 3NF.

## D Restrições

### Pessoa

- Cada pessoa tem um número de BI definido e deve ter no máximo 8 dígitos.  
BI INTEGER NOT NULL CHECK (BI <= 99999999)
- Cada pessoa tem um nome definido.  
nome TEXT NOT NULL
- A idade tem um valor definido e deve ser maior ou igual a 18.  
idade INTEGER NOT NULL CHECK (idade >= 18)
- Cada pessoa é identificada pelo seu BI.  
PRIMARY KEY (BI)

### Discoteca

- Cada discoteca tem um id definido.  
id INTEGER NOT NULL
- A discoteca tem um nome definido.  
nome TEXT NOT NULL
- A discoteca tem uma localização definido.  
localização TEXT NOT NULL
- A discoteca tem um proprietário definido.  
proprietário TEXT NOT NULL
- O nome e a localização da discoteca definem-na unicamente.  
UNIQUE (nome, localização)
- Cada discoteca é identificada pelo seu id.  
PRIMARY KEY (id)
- A restrição correspondente à área total da discoteca ter que ser maior que a soma das áreas dos espaços interiores irá ser implementada posteriormente.

### Membro

- Cada membro faz referência a uma Pessoa.  
BI INTEGER NOT NULL REFERENCES Pessoa
- Cada membro faz referência a uma Discoteca.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca
- O número de membro tem um valor definido.  
nrMembro INTEGER NOT NULL
- O membro apenas pode ser do tipo Regular ou VIP.  
tipo TEXT CHECK (tipo in ('regular','VIP'))

- O número de membro e o id da discoteca à qual pertence são únicos.  
UNIQUE (nrMembro, idDiscoteca)
- Cada membro é identificado pelo seu BI e pelo id da discoteca à qual pertence.  
PRIMARY KEY (BI, idDiscoteca)
- A restrição em que apenas membros VIP podem reservar o lounge irá ser implementada posteriormente.

#### Reserva

- Cada reserva tem um id definido.  
id INTEGER NOT NULL
- A reserva tem um dia definido.  
dia TEXT NOT NULL
- A reserva tem uma hora definida.  
hora TEXT NOT NULL
- O número de garrafas predefinido é 0 e deve ser maior ou igual a 0.  
nrGarrafas INTEGER DEFAULT 0 CHECK (nrGarrafas >= 0)
- O número de sofás predefinido é 0 e deve estar compreendido entre 0 e 5, inclusive.  
nrSofas INTEGER DEFAULT 0 CHECK (nrSofas >= 0 AND nrSofas <= 5)
- Cada reserva tem uma pessoa definida, referenciada pelo seu BI.  
BI INTEGER NOT NULL REFERENCES Pessoa
- Cada reserva é identificada pelo seu id.  
PRIMARY KEY (id)

#### Lounge

- Cada lounge tem um id definido.  
id INTEGER NOT NULL
- A área das caixas tem de ser maior que 0.  
areaEspaco REAL CHECK (areaEspaco > 0)
- Cada lounge faz referência ao id da discoteca correspondente.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca
- Cada lounge é identificado pelo seu id.  
PRIMARY KEY (id)

### **ReservaLounge**

- Cada reserva do lounge faz referência ao seu id.  
idReserva INTEGER NOT NULL REFERENCES Reserva
- Cada reserva do lounge faz referência ao id do lounge onde foi feita.  
idLounge INTEGER NOT NULL REFERENCES Lounge
- Cada reserva do lounge é identificada pelo id da reserva e pelo id do Lounge onde foi feita.  
PRIMARY KEY (idReserva, idLounge)

### **CaixasPagamento**

- Cada espaço de caixas de pagamento tem um id definido.  
id INTEGER NOT NULL
- A área das caixas tem de ser maior que 0.  
areaEspaco REAL CHECK (areaEspaco > 0)
- O dinheiro em caixa tem de estar definido.  
dinheiroCaixa REAL NOT NULL
- Cada espaço de caixas de pagamento faz referência ao id da discoteca correspondente.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca
- Cada espaço de caixas de pagamento é identificado pelo seu id.  
PRIMARY KEY (id)

### **Bengaleiro**

- Cada bengaleiro tem um id definido.  
id INTEGER NOT NULL
- A área do bengaleiro tem de ser maior que 0.  
areaEspaco REAL CHECK (areaEspaco > 0)
- O número de casacos tem de estar definido e tem de ser maior que 0.  
nrMaxCasacos INTEGER NOT NULL CHECK (nrMaxCasacos > 0)
- O preço por casaco tem de estar definido e tem de ser maior ou igual a 0.  
precoCasaco REAL NOT NULL CHECK (precoCasaco >= 0)
- Cada bengaleiro faz referência ao id da discoteca correspondente.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca
- Cada bengaleiro é identificado pelo seu id.  
PRIMARY KEY (id)

### **Artista**

- Cada artista tem um id definido.  
id INTEGER NOT NULL
- O artista tem um nome definido.  
nome TEXT NOT NULL
- O artista tem um número de telemóvel definido, que deve ter 9 dígitos e começar por 91, 92, 93 ou 96.  
nrTelemovel INTEGER NOT NULL CHECK ((nrTelemovel >= 910000000 AND nrTelemovel <= 939999999) OR (nrTelemovel >= 960000000 AND nrTelemovel <= 969999999))
- o artista apenas pode ser do tipo convidado ou residente.  
tipo TEXT CHECK (tipo in ('convidado','residente'))
- Cada artista é identificado pelo seu id.  
PRIMARY KEY (id)

### **Pista**

- Cada pista tem um id definido.  
id INTEGER NOT NULL
- Cada pista tem um nome definido.  
nome TEXT NOT NULL
- A área da pista tem de ser maior que 0.  
areaEspaco REAL CHECK (areaEspaco > 0)
- O género de música apenas pode ser do tipo funk, trance, house, 90's, kizomba ou reggaeton.  
generoMusica TEXT CHECK (generoMusica in ('funk', 'trance', 'house', '90s', 'kizomba', 'reggaeton'))
- Cada pista tem associado um artista residente único, exclusivo à mesma.  
idResidente INTEGER REFERENCES Artista UNIQUE
- Cada pista faz referência ao id da discoteca correspondente.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca
- O nome da pista e o id da discoteca a que pertence definem-na unicamente.  
UNIQUE (nome, idDiscoteca)
- Cada pista é identificada pelo seu id.  
PRIMARY KEY (id)

### **Atuação**

- Cada atuação faz referência ao id do artista.  
idArtista INTEGER NOT NULL REFERENCES Artista



- Cada atuação faz referência ao id da sua pista.  
idPista INTEGER NOT NULL REFERENCES Pista
- A atuação tem hora de começo definida.  
horaComeço TEXT NOT NULL
- A atuação tem hora de término definida.  
horaFim TEXT NOT NULL
- A duração da atuação deve ser de 0.5 a 4 horas.  
duracao TEXT NOT NULL CHECK (duracao >= '00:30' AND duracao <= '04:00')
- Cada atuação é definida pelo id do seu artista e da pista onde atua.  
PRIMARY KEY (idArtista, idPista)
- O cálculo do atributo derivado duração a partir da hora de começo e fim da atuação irá ser implementado posteriormente.
- A restrição em que apenas artistas do tipo convidado poderão ter uma atuação irá ser implementada posteriormente.

#### Bar

- Cada bar tem um id definido.  
id INTEGER NOT NULL
- Cada bar tem associado a sua pista.  
idPista INTEGER REFERENCES Pista
- Cada bar é identificado pelo seu id.  
PRIMARY KEY (id)

#### Bebida

- Cada bebida tem um nome definido.  
nome TEXT NOT NULL
- Cada bebida tem uma marca definido.  
marca TEXT NOT NULL
- Cada bebida tem um stock definido que deve estar entre 50 e 2400.  
stock INTEGER NOT NULL CHECK (stock > 50 AND stock < 2400)
- Cada bebida tem um preço definido.  
preco REAL NOT NULL
- Cada bebida tem um teor alcoólico definido que deve ser maior ou igual a zero.  
teorAlcoolico REAL NOT NULL CHECK (teorAlcoolico >= 0)

- O preço de cada bebida varia com o teor alcoólico: se o teor alcoólico for menor que 1.2, o preço é menor ou igual a 4€; se for maior, o preço está entre 4€ e 50€.  
CHECK ((teorAlcoolico < 1.2 AND preco < 4.0) OR (teorAlcoolico >= 1.2 AND preco >= 4.0 AND preco < 50.0))
- Cada bebida é identificada pelo seu nome e marca.  
PRIMARY KEY (nome, marca)

### BarBebida

- Cada bebida num bar faz referência ao id do bar em questão.  
idBar INTEGER NOT NULL REFERENCES Bar
- Cada bebida num bar tem um nome definido.  
nome TEXT NOT NULL
- Cada bebida num bar tem uma marca definida.  
marca TEXT NOT NULL
- Cada bebida num bar faz referência ao nome e marca da mesma.  
FOREIGN KEY (nome, marca) REFERENCES Bebida
- Cada bebida num bar é identificada pelo seu nome, marca e pelo id do bar em questão.  
PRIMARY KEY (idBar, nome, marca)

### Funcionário

- Cada funcionário tem um id definido.  
id INTEGER NOT NULL
- Cada funcionário tem um nome definido.  
nome TEXT NOT NULL
- Cada funcionário tem um número de telemóvel definido, que deve ter 9 dígitos e começar por 91, 92, 93 ou 96.  
nrTelemovel INTEGER NOT NULL CHECK ((nrTelemovel >= 910000000 AND nrTelemovel <= 939999999) OR (nrTelemovel >= 960000000 AND nrTelemovel <= 969999999))
- Cada funcionário tem um bi definido e deve ter no máximo 8 dígitos.  
BI INTEGER NOT NULL CHECK (BI <= 99999999)
- Cada funcionário tem um salário definido e maior que 665 (salário mínimo).  
salario INTEGER NOT NULL CHECK (salario > 665)
- Cada funcionário faz referência ao id da discoteca onde trabalha.  
idDiscoteca INTEGER NOT NULL REFERENCES Discoteca

- O número do BI do funcionário e o id da Discoteca onde trabalha definem-no unicamente.  
UNIQUE (BI, idDiscoteca)
- Cada funcionário é identificado pelo seu id.  
PRIMARY KEY (id)
- A restrição do salário ser em função do posto de trabalho irá ser implementada posteriormente.

### Hierarquia

- Cada relação de hierarquia tem associado o id do funcionário subalterno.  
idSubalterno INTEGER NOT NULL REFERENCES Funcionario
- Cada relação de hierarquia tem associado o id do funcionário gerente/superior.  
idGerente INTEGER NOT NULL REFERENCES Funcionario
- Cada relação de hierarquia é identificada pelo seu id de funcionário.  
PRIMARY KEY (idSubalterno)

### StaffGeral

- Cada membro do staff geral está associado a um funcionário.  
id INTEGER NOT NULL REFERENCES Funcionario
- Cada membro do staff geral é identificado pelo seu id.  
PRIMARY KEY (id)

### Segurança

- Cada segurança está associado a um funcionário.  
id INTEGER NOT NULL REFERENCES Funcionario PRIMARY KEY
- Cada segurança deverá ter uma licença de trabalho definida, única a cada um e com um máximo de 6 dígitos.  
licenca INTEGER NOT NULL UNIQUE CHECK (licenca <= 999999)
- Cada segurança é identificado pelo seu id.  
PRIMARY KEY (id)

### Bartender

- Cada bartender está associado a um funcionário.  
id INTEGER NOT NULL REFERENCES Funcionario
- Cada bartender deverá ter um nível de formação entre 1 e 3.  
nivelFormacao INTEGER NOT NULL CHECK (nivelFormacao >= 1 AND nivelFormacao <= 3)
- Cada bartender é identificado pelo seu id.  
PRIMARY KEY (id)

### **LoungeStaffGeral**

- Cada membro do staff geral num lounge faz referência ao seu id.  
idStaff INTEGER NOT NULL REFERENCES StaffGeral
- Cada membro do staff geral num lounge faz referência ao id desse mesmo Lounge.  
idLounge INTEGER NOT NULL REFERENCES Lounge
- Cada membro do staff geral é identificado pelo seu id de staff e pelo id do lounge ao qual pertence.  
PRIMARY KEY (idStaff, idLounge)

### **CaixasStaffGeral**

- Cada membro do staff geral nas caixas de pagamento faz referência ao seu id.  
idStaff INTEGER NOT NULL REFERENCES StaffGeral
- Cada membro do staff geral nas caixas de pagamento faz referência ao id dessas mesmas caixas.  
idCaixas INTEGER NOT NULL REFERENCES CaixasPagamento
- Cada membro do staff geral é identificado pelo seu id de staff e pelo id das caixas de pagamento às quais pertence.  
PRIMARY KEY (idStaff, idCaixas)

### **BengaleiroStaffGeral**

- Cada membro do staff geral num bengaleiro faz referência ao seu id.  
idStaff INTEGER NOT NULL REFERENCES StaffGeral
- Cada membro do staff geral num bengaleiro faz referência ao id desse mesmo bengaleiro.  
idBengaleiro INTEGER NOT NULL REFERENCES Bengaleiro
- Cada membro do staff geral é identificado pelo seu id de staff e pelo id do bengaleiro ao qual pertence.  
PRIMARY KEY (idStaff, idBengaleiro)

### **BarBartender**

- Cada bartender num bar faz referência ao seu id.  
idBartender INTEGER NOT NULL REFERENCES Bartender
- Cada bartender num bar faz referência ao id desse mesmo bar.  
idBar INTEGER NOT NULL REFERENCES Bar
- Cada bartender num bar é identificado pelo seu id de bartender e pelo id do bar ao qual pertence.  
PRIMARY KEY (idBartender, idBar)

### **PistaSegurança**

- Cada segurança numa pista faz referência ao seu id.  
idSegurança INTEGER NOT NULL REFERENCES Segurança
- Cada segurança numa pista faz referência ao id dessa mesma pista.  
idPista INTEGER NOT NULL REFERENCES Pista
- Cada segurança numa pista é identificado pelo seu id de segurança e pelo id da pista à qual pertence.  
PRIMARY KEY (idSegurança, idPista)