



Projeto GDI

Repositório

<https://github.com/cebms/projeto-GDI>

Integrantes

Carlos Eduardo Bezerra Mendes Silva - cebms

Gabriel Nogueira de Moura Pereira - gnmp

João Pedro Mafaldo de Paula - jmp

Jose Clebson de Souza Oliveira - jcso2

Mariana Guimarães Coêlho - mgc3

MiniMundo



A base de dados foi projetada para uma plataforma de **streaming musical**, onde usuários podem criar músicas, playlists e eventos, além de interagir entre si e assinar planos premium.

Usuários

Cada usuário é identificado pelo seu CPF, além de ter informações como nome, e-mail e telefones de contato. Alguns usuários podem se tornar criadores de conteúdo, registrando uma conta bancária para recebimento de pagamentos, podendo criar músicas e publicar álbuns. Além disso, a plataforma pode ter usuários com assinaturas, que terão acesso às músicas e poderão interagir com outros usuários.

Músicas e Álbuns

Os criadores podem lançar álbuns, que contêm diversas músicas, cada uma com um título, duração, gênero e imagem de capa. Todas as músicas estão associadas a um álbum e podem ser publicadas na plataforma.

Eventos

A plataforma permite criar eventos associados aos criadores. Os criadores podem organizar eventos musicais, registrados com um número identificador, data e endereço (CEP e número).

Assinaturas

Usuários podem optar por planos de assinatura premium, pagando valores diferenciados conforme o tipo de plano escolhido (mensal, anual, etc.). Também há a possibilidade de aplicar descontos.

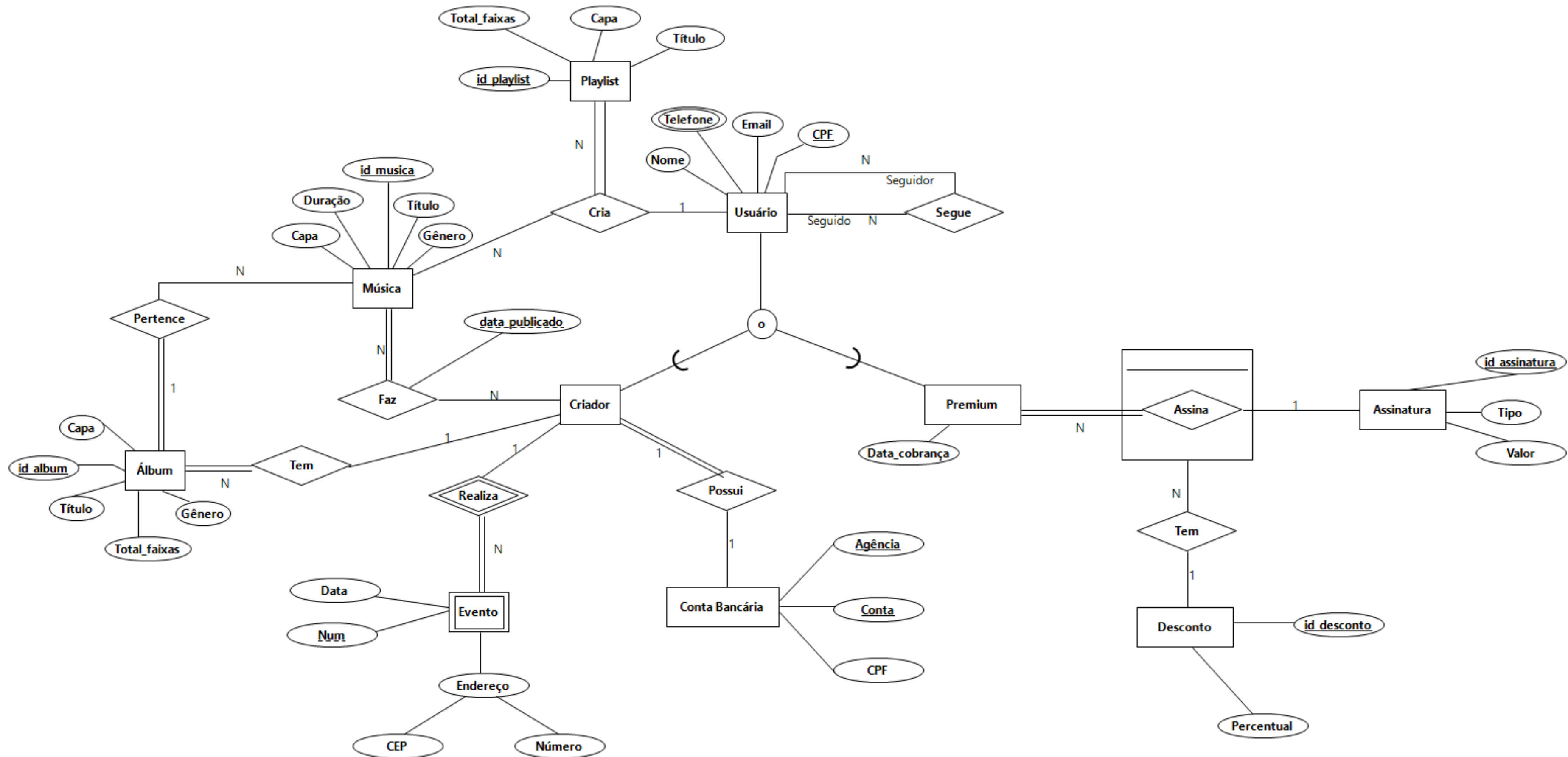
Playlist

Usuários podem criar playlists personalizadas contendo diferentes músicas. A playlist contém o número de faixas e uma imagem de capa.

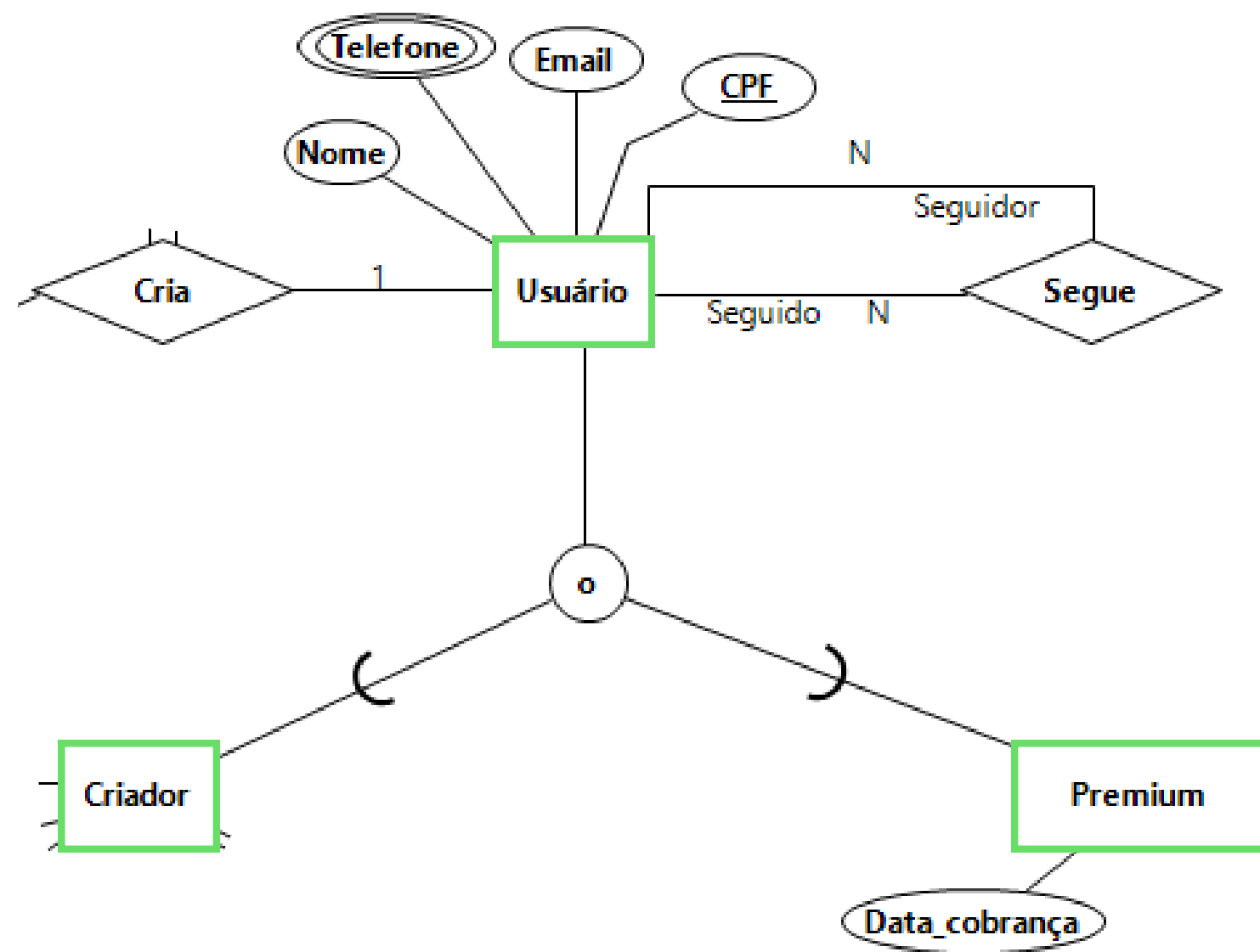
Projeto Conceitual



Pag 3



Usuário, Criador e Premium



Usuário(CPF, nome, email, telefone1, telefone2)

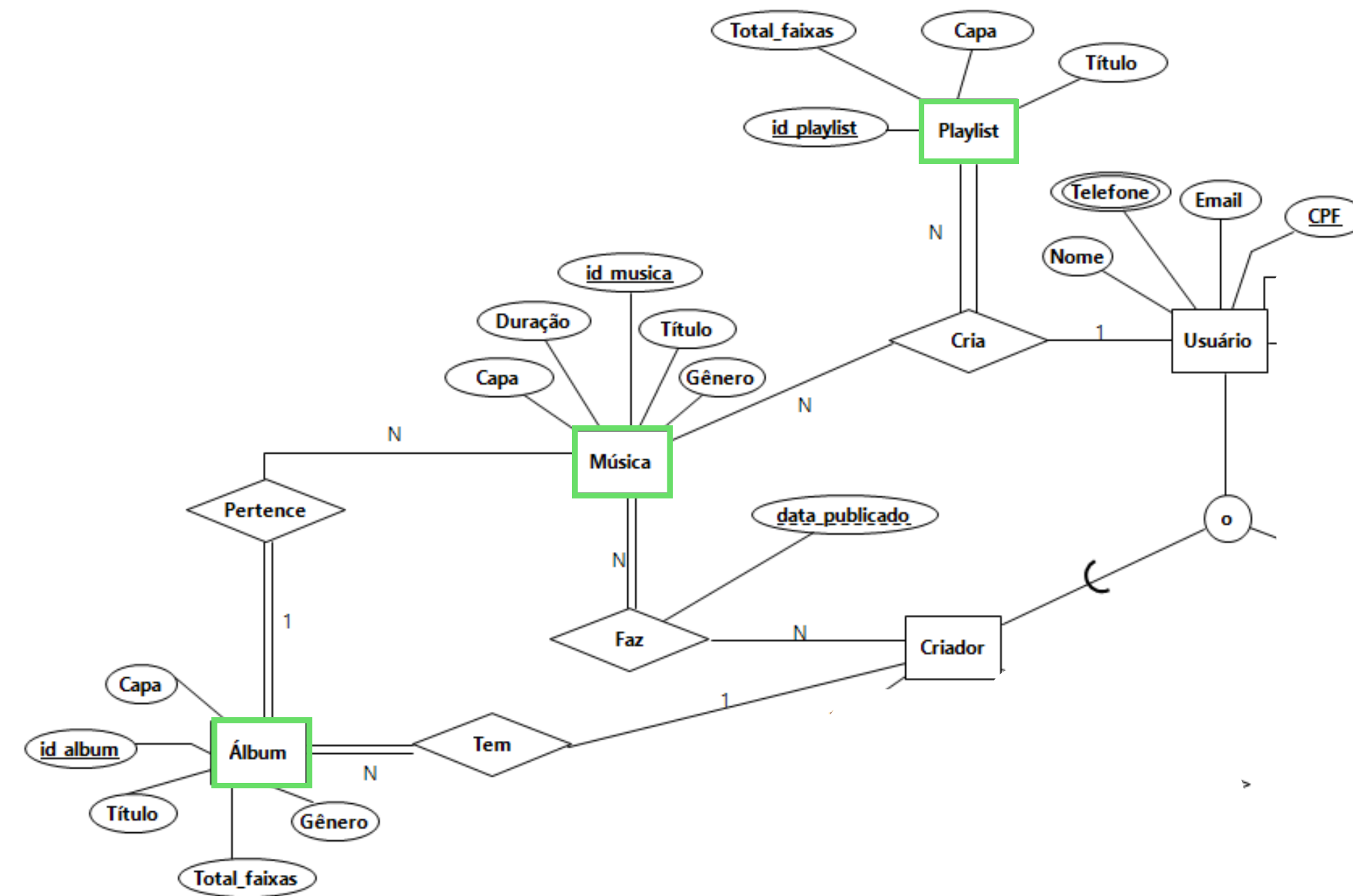
Criador(CPF, [agência]!, [conta]!)

CPF → Usuário(CPF) Agência → Conta Bancária(Agência) Conta → Conta Bancária(Conta)

Premium(CPF)

CPF → Usuário(CPF)

Música, Álbum e Playlist



Música(id música, id_álbum!, título, duração, capa, gênero)

id_álbum → Álbum(id_álbum)

Álbum(id álbum, [CPF]!, título, capa, gênero, Total_faixas)

CPF → Criador(CPF)

Playlist(id playlist, título, capa, Total_faixas)

Faz, Cria e Segue



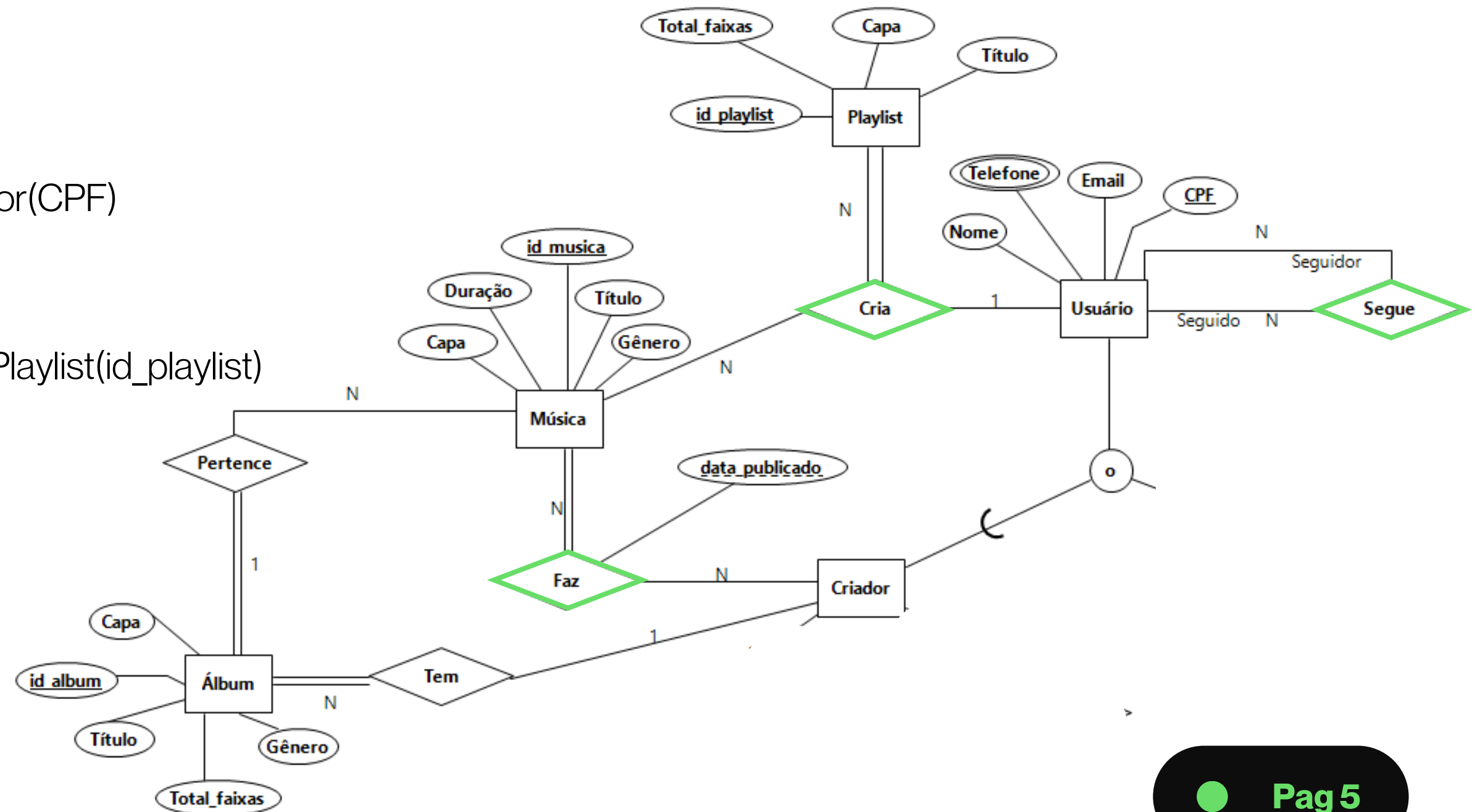
faz(id música, CPF, data publicada)

id_música → Música(id_música) CPF → Criador(CPF)

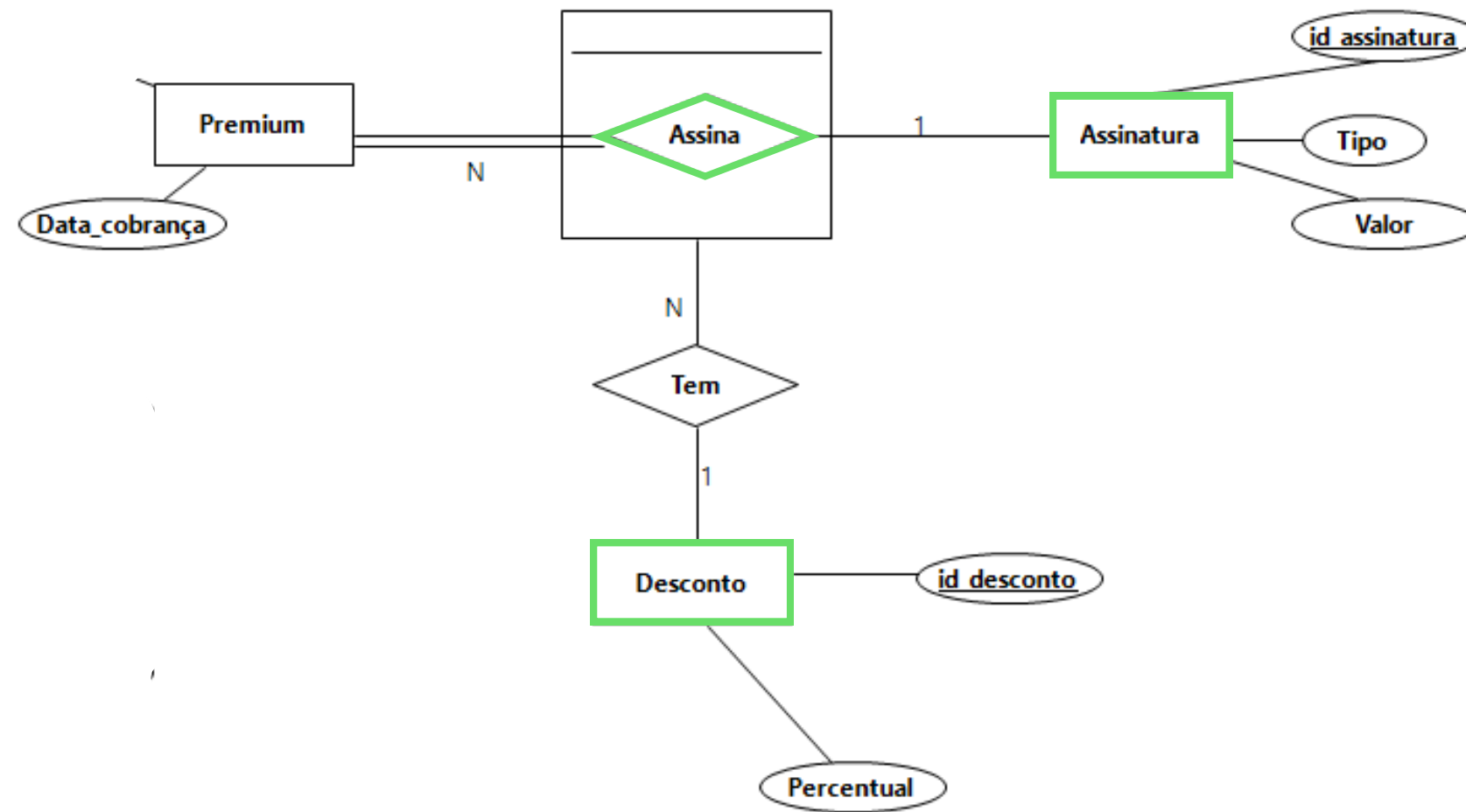
cria(id música, id_playlist, [CPF]!)

id_música → Música(id_música) id_playlist → Playlist(id_playlist)

segue(seguido, seguidor)



Desconto, Assina e Assinatura



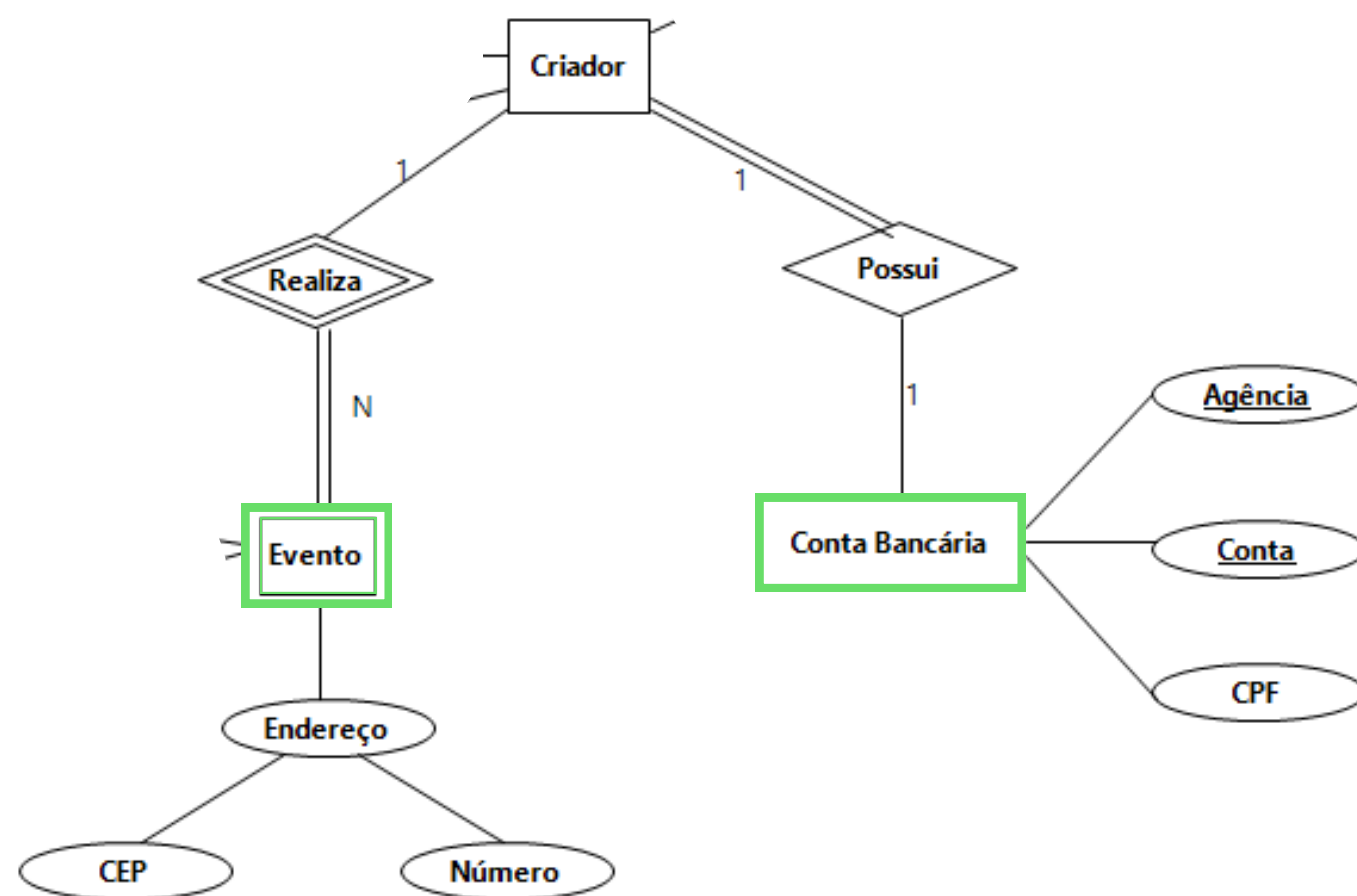
Desconto(id desconto, percentual)

Assina(CPF, id assinatura, id_desconto)

CPF → Usuário(CPF) id_assinatura → Assinatura(id_assinatura) id_desconto → Desconto(id_desconto)

Assinatura(id assinatura, tipo, valor, data_cobrança)

Evento e Conta Bancária



Evento(CPF, Num, data, endereço_Cep, endereço_Número)

CPF → Usuário(CPF)

Conta Bancária(agência, conta, CPF)

Projeto Lógico

Usuário(CPF, nome, email, telefone1, telefone2)

Criador(CPF, [agência]!, [conta]!)

CPF → Usuário(CPF) Agência → Conta Bancária(Agência)

Conta → Conta Bancária(Conta)

Premium(CPF)

CPF → Usuário(CPF)

Evento(CPF, Num, data, endereço_Cep, endereço_Número)

CPF → Usuário(CPF)

Música(id_música, id_álbum!, título, duração, capa, gênero)

id_álbum → Álbum(id_álbum)

Álbum(id_álbum, [CPF]!, título, capa, gênero, Total_faixas)

CPF → Criador(CPF)

Playlist(id_playlist, título, capa, Total_faixas)

Assinatura(id_assinatura, tipo, valor, data_cobranca)

Conta Bancária(agência, conta, CPF)

Desconto(id_desconto, percentual)

faz(id_música, CPF, data_publicada)

id_música → Música(id_música) CPF → Criador(CPF)

cria(id_música, id_playlist, [CPF]!)

id_música → Música(id_música) id_playlist → Playlist(id_playlist)

segue(seguido, seguidor)

seguido → Usuário(CPF)

seguidor → Usuário(CPF)

assina(CPF, id_assinatura, id_desconto)

CPF → Usuário(CPF) id_assinatura → Assinatura(id_assinatura) id_desconto → Desconto(id_desconto)



Criação das Tabelas

Criação das Tabelas

```
CREATE TABLE USUARIO(  
    CPF VARCHAR(20) PRIMARY KEY,  
    nome VARCHAR(45),  
    email VARCHAR(45),  
    telefone1 VARCHAR(14),  
    telefone2 VARCHAR(14)  
);  
  
CREATE TABLE CRIADOR (  
    CPF VARCHAR(20) PRIMARY KEY,  
    agencia VARCHAR(5) NOT NULL,  
    conta VARCHAR(8) NOT NULL,  
    CONSTRAINT criador_fk FOREIGN KEY (CPF) REFERENCES USUARIO(CPF),  
    CONSTRAINT agencia_fk FOREIGN KEY (agencia, conta) REFERENCES CONTA_BANCARIA(agencia, conta)  
);  
  
CREATE TABLE PREMIUM (  
    CPF VARCHAR(20) PRIMARY KEY,  
    data_cobranca DATE,  
    CONSTRAINT premium_fk FOREIGN KEY (CPF) REFERENCES USUARIO(CPF)  
);
```

Criação das Tabelas

```
CREATE TABLE ALBUM (  
  id_album NUMBER PRIMARY KEY,  
  CPF VARCHAR(20) NOT NULL,  
  titulo VARCHAR(45),  
  capa VARCHAR(45),  
  genero VARCHAR(45),  
  total_faixas NUMBER,  
  CONSTRAINT CPF_criador_album_fk FOREIGN KEY (CPF) REFERENCES CRIADOR(CPF)  
);
```

```
CREATE TABLE PLAYLIST(  
  id_playlist NUMBER PRIMARY KEY,  
  titulo VARCHAR(45),  
  capa VARCHAR(45),  
  genero VARCHAR(45),  
  total_faixas NUMBER  
);
```

```
CREATE TABLE CONTA_BANCARIA (  
  agencia VARCHAR(5),  
  conta VARCHAR(8),  
  CPF VARCHAR(20),  
  PRIMARY KEY (agencia, conta)  
);
```

Criação das Tabelas

```
CREATE TABLE FAZ(  
    id_musica NUMBER,  
    CPF VARCHAR(20),  
    data_publicada DATE,  
    PRIMARY KEY (id_musica, CPF, data_publicada),  
    CONSTRAINT id_musica_faz_fk FOREIGN KEY (id_musica) REFERENCES MUSICA(id_musica),  
    CONSTRAINT CPF_criador_faz_fk FOREIGN KEY (CPF) REFERENCES CRIADOR(CPF)  
);  
  
CREATE TABLE CRIA(  
    id_musica NUMBER,  
    id_playlist NUMBER,  
    CPF VARCHAR(20),  
    PRIMARY KEY (id_musica, id_playlist),  
    CONSTRAINT id_musica_cria_fk FOREIGN KEY (id_musica) REFERENCES MUSICA(id_musica),  
    CONSTRAINT id_playlist_fk FOREIGN KEY (id_playlist) REFERENCES PLAYLIST(id_playlist)  
);  
  
CREATE TABLE SEGUE(  
    seguido VARCHAR(20),  
    seguidor VARCHAR(20),  
    PRIMARY KEY (seguido, seguidor),  
    CONSTRAINT seguido_fk FOREIGN KEY (seguido) REFERENCES USUARIO(CPF),  
    CONSTRAINT seguidor_fk FOREIGN KEY (seguidor) REFERENCES USUARIO(CPF)  
);
```


Criação das Tabelas

```
CREATE TABLE DESCONTO(  
    id_desconto NUMBER PRIMARY KEY,  
    percentual NUMBER(5,2)  
);
```

```
CREATE TABLE ASSINA(  
    CPF VARCHAR(20),  
    idassinatura NUMBER,  
    id_desconto NUMBER,  
    PRIMARY KEY (CPF, idassinatura),  
    CONSTRAINT CPF_usuario_assina_fk FOREIGN KEY (CPF) REFERENCES PREMIUM(CPF),  
    CONSTRAINT idassinatura_fk FOREIGN KEY (idassinatura) REFERENCES ASSINATURA(idassinatura),  
    CONSTRAINT id_desconto_fk FOREIGN KEY (id_desconto) REFERENCES DESCONTO(id_desconto)  
);
```

```
CREATE TABLE ASSINATURA(  
    idassinatura NUMBER PRIMARY KEY,  
    tipo VARCHAR(45),  
    valor NUMBER  
);
```

Criação das Tabelas

```
CREATE TABLE EVENTO (  
  CPF VARCHAR(20),  
  numEvento NUMBER,  
  data DATE,  
  CEP VARCHAR(9),  
  numero NUMBER,  
  PRIMARY KEY (CPF, numEvento),  
  CONSTRAINT CPF_artista_evento_fk FOREIGN KEY (CPF) REFERENCES CRIADOR(CPF)  
);
```

```
CREATE TABLE CONTA_BANCARIA (  
  agencia VARCHAR(5),  
  conta VARCHAR(8),  
  CPF VARCHAR(20),  
  PRIMARY KEY (agencia, conta)  
);
```




Consultas

Consultas

```
SELECT A.CPF AS CPF_CRIADOR,  
       COUNT(*) AS TOTAL_ALBUNS  
FROM ALBUM A  
GROUP BY A.CPF  
HAVING COUNT(*) > 1;
```

Group By/Having

```
SELECT U.NOME          AS NOME_CRIADOR,  
       AL.TITULO       AS TITULO_ALBUM  
FROM CRIADOR C  
       LEFT OUTER JOIN ALBUM AL ON C.CPF = AL.CPF  
       INNER JOIN USUARIO U    ON C.CPF = U.CPF  
ORDER BY U.NOME;
```

Junção
Interna/Externa

Consultas

```
SELECT C.CPF  
FROM CRIADOR C  
WHERE EXISTS (  
  SELECT 1  
  FROM FAZ F  
  WHERE F.CPF = C.CPF  
);
```



Semi-Join

```
SELECT C.CPF  
FROM CRIADOR C  
WHERE NOT EXISTS (  
  SELECT 1  
  FROM FAZ F  
  WHERE F.CPF = C.CPF  
);
```



Anti-Join

Consultas

```
SELECT U.NOME,  
       A.TIPO,  
       A.VALOR  
FROM USUARIO U  
INNER JOIN PREMIUM PR ON U.CPF = PR.CPF  
INNER JOIN ASSINA ASI ON ASI.CPF = PR.CPF  
INNER JOIN ASSINATURA A ON A.ID_ASSINATURA = ASI.ID_ASSINATURA  
WHERE A.VALOR = (  
    SELECT MAX(V valor)  
    FROM ASSINATURA  
);
```

Subconsulta Escalar

```
SELECT CR.CPF,  
       U.NOME  
FROM CRIADOR CR  
INNER JOIN USUARIO U ON CR.CPF = U.CPF  
WHERE (CR.AGENCIA, CR.CONTA) = (  
    SELECT AGENCIA, CONTA  
    FROM CRIADOR  
    WHERE CPF = '111.222.333-44'  
);
```

**Subsoncult de
Linha**

Consultas

```
SELECT M.TITULO AS MUSICA,  
       M.ID_ALBUM  
FROM MUSICA M  
WHERE M.ID_ALBUM IN (  
    SELECT AL.ID_ALBUM  
    FROM ALBUM AL  
    WHERE AL.GENERO = 'Rock'  
);
```

**Subconsulta de
Tabela**

```
SELECT seguidor AS CPF  
FROM SEGUE
```

UNION

```
SELECT seguido AS CPF  
FROM SEGUE;
```

**Operação de
Conjunto**

Consultas

```
SELECT COUNT(*) AS TOTAL_GMAIL_USERS  
FROM USUARIO  
WHERE email LIKE '%@gmail.com';
```

Count(*)

```
SELECT U.CPF,  
       U.EMAIL  
FROM USUARIO U  
WHERE U.CPF NOT IN (  
    SELECT SEGUIDO  
    FROM SEQUE  
);
```

Anti-Join



Obrigado!