

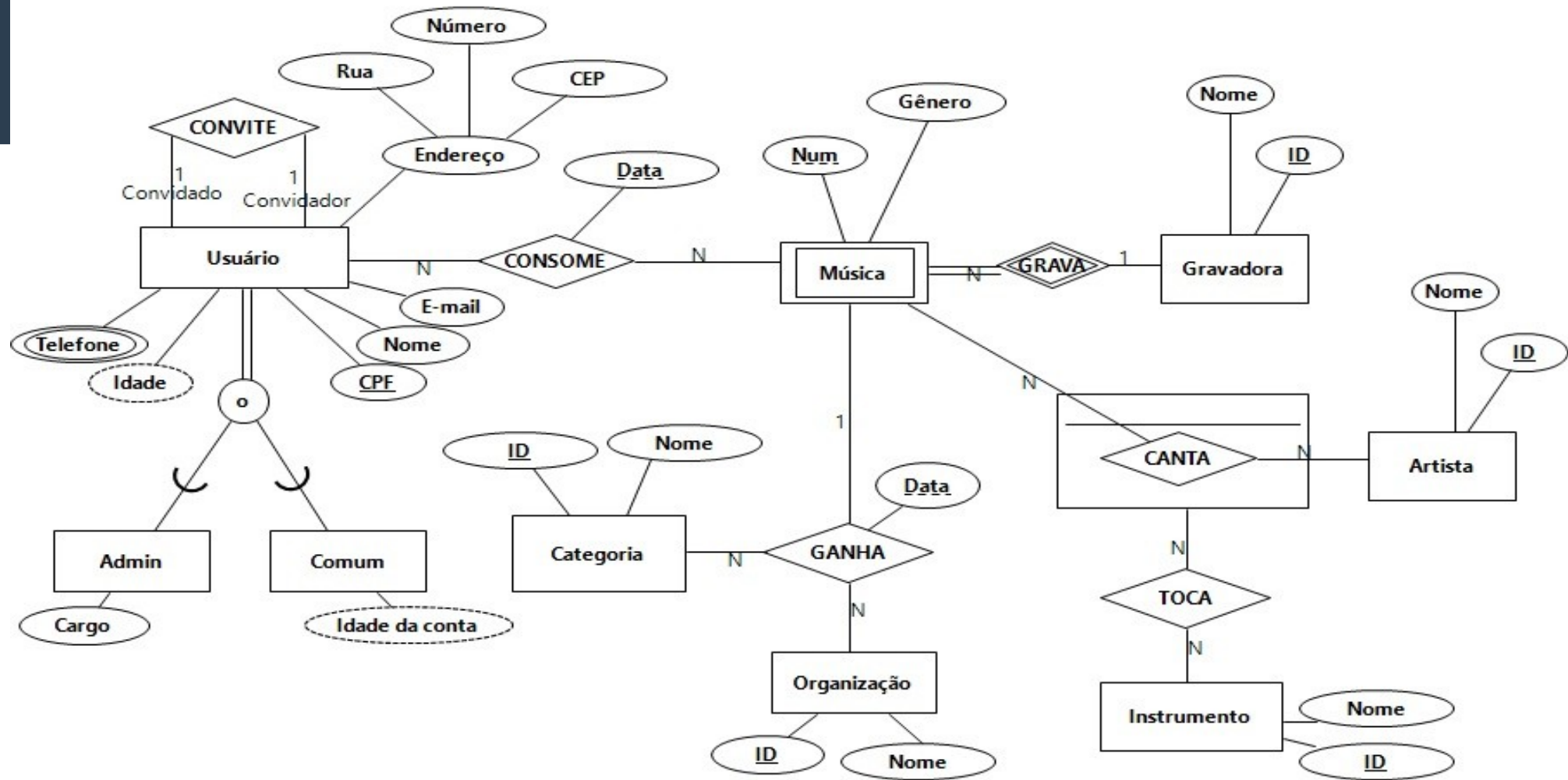
Gerenciamento de Dados e Informação

Apresentação do Projeto

Lucas Emmanuel
Gabriel Ferreira
Nicolas Brito
Mateus Viégas
Pedro Domingues

Apresentação

- **O tema escolhido foi uma plataforma de bancos de dados para músicas.**
- **As entidades modelam agentes envolvidos com produção e consumo de musica como usuários, artistas, gravadora.**
- **Bem como abstrações como categorias de música, performances e tipos de usuários.**



Projeto Lógico

ENTIDADES REGULARES

Usuário(CPF, Nome, E-mail, end_rua, end_num, end_CEP, telefone_1, telefone_2)

Artista(ID, Nome)

Gravadora(ID, Nome)

Instrumento(ID, Nome)

Organização(ID, Nome)

Categoria(ID, Nome)

Projeto Lógico

ENTIDADE FRACA

Música(ID, num, gênero)
ID → Gravadora(ID)

HERANÇAS

Admin(ID, Cargo)
ID → Usuário(CPF)

Comum(ID)
ID → Usuário(CPF)

ENTIDADE ASSOCIATIVA

Canta(ID_musica, ID_artista)
ID_musica → Musica(ID, num)
ID_artista → Artista(ID)

Projeto Lógico

Convite

Usuário.convidado-Usuário.convidador, 1:1, parcial-parcial, adição de chave estrangeira

Usuário(CPF, [CPF_convidado], Nome, E-mail, end_rua, end_num, end_CEP, telefone_1, telefone_2)

CPF_convidado → Usuário(CPF)

Consome

Usuário-Música, N:N, parcial-parcial, criação de relação

Consome(CPF, ID_musica, data)

CPF → Usuário(CPF)

ID_musica → Música(ID, num)

Projeto Físico

- O projeto físico foi implementado na linguagem PL/SQL.
- O físico foi criado e povoado seguindo as especificações do projeto lógico.

Projeto Físico

```
INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('38272418083','Tate Huber','neque@yahoo.edu','73775-5557','567-6538 Bilbendum Road','84676-195','175','82245-3899');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('66568769263','Lillith Mercer','et.lacinia@google.net','68473-8015','Ap #432-560 Et Av.','58227-477','923','18977-7273');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('98773832628','Blake Cotton','ridiculus.mus.proin@outlook.net','56776-5171','Ap #414-5894 Egestas. Ave','91545-177','758','13788-8486');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('14658923297','Dieter Tanner','urna.justo@google.net','37117-2716','P.O. Box 511, 3676 Nonummy Rd.','06068-482','682','27655-9683');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('51672289875','Jared Santana','nunc.sed@icloud.edu','15465-1248','325-7455 Lorem. St.','18128-642','372','45185-1656');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('59317743778','Herman Payne','pellentesque.ultrices@protonmail.org','80935-4525','Ap #517-3982 Pellentesque Street','65874-494','931','88361-1171');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('28453274274','Emerson Ochoa','phasellus.nulla@hotmail.net','48653-9933','Ap #682-2149 Eget Av.','25042-822','811','06393-1832');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('97588314932','Garrett Fitzpatrick','eros.nec@yahoo.co.uk','96207-7365','2801 Consequat St.','13400-767','682','57442-1573');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('48491585487','Carl Chapman','donec.tincidunt.donec@protonmail.ca','63552-1428','Ap #888-8432 Urna. Rd.','45668-482','627','13338-2432');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('58648837498','Fleur Church','phasellus.libero.mauris@hotmail.com','63768-5880','Ap #321-9418 Eget Rd.','61534-223','613','12878-8008');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('27565652844','Aaron Johnston','vulputate.dui.nec@hotmail.ca','96227-7320','8338 Tristique Rd.','65457-837','412','45225-4053');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('97237695877','Gretchen Lawrence','suspendisse.tristique@protonmail.edu','87688-1951','371-382 Aptent Rd.','92184-804','313','14790-8251');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('94147467348','Brittany Pollard','nunc@yahoo.org','47696-6513','Ap #621-2127 Tempor. Rd.','86242-272','774','33128-1163');

INSERT INTO USUARIO (CPF,NOME,EMAIL,TELEFONE1,END_RUA,END_CEP,END_NUM,TELEFONE2)
VALUES ('61272925396','Leandra Newman','integer.tincidunt@icloud.com','92123-5214','991-7002 Eu Rd.','66684-242','478','23423-8703');
```


Projeto Físico

```
CREATE OR REPLACE PROCEDURE INSERIR_COMUM AS
  v_cpf VARCHAR(15);
  contagem numeric := 17;
BEGIN
  FOR i IN 1..contagem LOOP
    -- Busca um CPF que não esteja presente nas tabelas ADMIN ou COMUM
    SELECT CPF INTO v_cpf
    FROM USUARIO
    WHERE CPF NOT IN (SELECT ID FROM ADMIN_ UNION SELECT ID FROM COMUM)
    FETCH FIRST 1 ROW ONLY;

    -- Se um CPF foi encontrado, faz o insert na tabela COMUM
    IF v_cpf IS NOT NULL THEN
      INSERT INTO COMUM(ID) VALUES (v_cpf);
      DBMS_OUTPUT.PUT_LINE('CPF ' || v_cpf || ' inserido com sucesso na tabela COMUM.');
```

Projeto Físico

-

```
CREATE OR REPLACE TRIGGER TRG_INSERE_MUSICA
BEFORE INSERT ON MUSICA
FOR EACH ROW
DECLARE
    V_GRAVADORA_ID GRAVADORA.ID%TYPE;
BEGIN
    SELECT ID INTO V_GRAVADORA_ID FROM GRAVADORA ORDER BY DBMS_RANDOM.VALUE FETCH FIRST 1 ROW ONLY;
    :NEW.ID := V_GRAVADORA_ID;
END;
/
```

Projeto Físico

- **As consultas conforme especificadas no projetos**

Projeto Físico

- **1) GROUP BY / HAVING**
- **1.A) Agrupar os gêneros musicais que tiveram mais de 1 música inseridas no BD, e suas quantidades**

```
-- 1.A) Agrupar os generos musicais que tiveram mais de 1 música inseridas no BD, e suas quantidades  
SELECT MUSICA.GENERO, COUNT(MUSICA.ID) AS QUANTIDADE  
FROM MUSICA  
GROUP BY GENERO  
HAVING COUNT(MUSICA.ID) > 1;
```

GENERO	QUANTIDADE
Eletrônica	2
Pop	3

Projeto Físico

- 2) JUNCTÃO INTERNA
- Agrupar quantos prêmios as organizações ganharam (dentre as que ganharam mais de 1 prêmio), com as quantidades na ordem decrescente

```
SELECT ORGANIZACAO.NOME, COUNT(GANHA.ID_ORGANIZACAO)
FROM ORGANIZACAO INNER JOIN GANHA
ON ORGANIZACAO.ID = GANHA.ID_ORGANIZACAO
GROUP BY ORGANIZACAO.NOME
HAVING COUNT(GANHA.ID_ORGANIZACAO) > 1
ORDER BY COUNT(GANHA.ID_ORGANIZACAO) DESC;
```

Projeto Físico

-

NOME	COUNT(GANHA.ID_ORGANIZACAO)
Billboard Music Awards	4
GDI Music Awards	3
Billboard Women in Music	3
People's Choice Awards	2
MTV Europe Video Awards	2
Brit Awards	2

Projeto Físico

- 3) JUNÇÃO EXTERNA
- 3.A) Nomes de todos os usuários, e seus cargos (se eles tiverem)

```
SELECT U.NOME, A.CARGO  
FROM USUARIO U LEFT OUTER JOIN ADMIN_ A ON U.CPF = A.ID  
ORDER BY U.NOME ASC;
```

Projeto Físico

-

NOME	CARGO
Aaron Johnston	Analista
Amal Marsh	-
Blake Cotton	Designer
Brittany Pollard	-
Carl Chapman	-
Colby Fox	Operador
Dieter Tanner	-
Emerson Ochoa	-

Fleur Church	-
Garrett Fitzpatrick	-
George Fischer	-
Gray Adams	Estudante
Gretchen Lawrence	-
Herman Payne	-
Ina Weeks	Arquiteto
Jared Santana	-
Leandra Newman	Médico

Projeto Físico

- 4) SEMI JUNÇÃO
- 4.A) Artistas que cantaram

```
SELECT ARTISTA.NOME  
FROM ARTISTA  
WHERE EXISTS  
    (SELECT * FROM CANTA WHERE ARTISTA.ID = CANTA.ID_ART);
```

Projeto Físico

•

NOME	
Garth Petty	Amos Boyd
Amos Boyd	Marvin Berg
Marvin Berg	Timon Willis
Timon Willis	Rigel Graves
Rigel Graves	Aspen Flowers
Aspen Flowers	Maisie Blevins
Maisie Blevins	Thaddeus Mcguire
Thaddeus Mcguire	Curran Franco
Curran Franco	Aidan Carrillo
Aidan Carrillo	Roanna Roth
Roanna Roth	Alice King
	Lyle Stanley

Projeto Físico

- 5) ANTI-JUNÇÃO
- 5.A) Usuários que não são administradores

```
SELECT USUARIO.NOME
FROM USUARIO
WHERE NOT EXISTS (
    SELECT 1
    FROM ADMIN_
    WHERE ADMIN_.ID = USUARIO.CPF
);
```

Projeto Físico

NOME
George Fischer
Tate Huber
Dieter Tanner
Jared Santana
Herman Payne
Emerson Ochoa
Garrett Fitzpatrick
Carl Chapman
Fleur Church
Gretchen Lawrence
Brittany Pollard

Emerson Ochoa
Garrett Fitzpatrick
Carl Chapman
Fleur Church
Gretchen Lawrence
Brittany Pollard
Talon Jenkins
Morgan Saunders
Molly Sharp
Lyle Sargent
Zelda Nelson
Amal Marsh

Projeto Físico

- 5) ANTI-JUNÇÃO
- 5.B) Usuários que não são comuns

```
SELECT USUARIO.NOME
FROM USUARIO
WHERE NOT EXISTS (
    SELECT 1
    FROM COMUM
    WHERE COMUM.ID = USUARIO.CPF
);
```

Projeto Físico

NOME
Lillith Mercer
Blake Cotton
Aaron Johnston
Leandra Newman
Leslie French
Colby Fox
Gray Adams
Ina Weeks

Projeto Físico

- 6) SUBCONSULTA DO TIPO ESCALAR
- 6.A) Número de músicas que cada artista canta

```
SELECT ARTISTA.NOME,  
    (  
        SELECT COUNT(*)  
        FROM CANTA  
        WHERE CANTA.ID_ART = ARTISTA.ID  
    ) AS NUMERO_DE_MUSICAS  
FROM ARTISTA;
```

Projeto Físico

NOME	NUMERO_DE_MUSICAS
Maisie Blevins	2
Timon Willis	3
Plato Diaz	0
Marvin Berg	1
Lyle Stanley	1
Roanna Roth	1
Garth Petty	1
Aidan Carrillo	1
Hedy Melton	0
Thaddeus Mcguire	1
Amos Boyd	1

Joan Eaton	0
Alice King	2
Lionel Rivas	0
Hamish Chan	0
Upton Schwartz	1
Christian Chapman	0
Curran Franco	1
Dominique Harding	0
Chiquita Patrick	0
Linus Ramos	0
Aspen Flowers	2

Projeto Físico

- **7) SUBCONSULTA DO TIPO LINHA**
- **7.A) Selecionar usuários que são comuns**

```
SELECT USUARIO.NOME, USUARIO.CPF
FROM USUARIO
WHERE USUARIO.CPF IN (
    SELECT COMUM.ID
    FROM COMUM
)
```

Projeto Físico

NOME	CPF
George Fischer	96728388178
Tate Huber	38272418683
Dieter Tanner	14658923297
Jared Santana	51672289875
Herman Payne	59317743778
Emerson Ochoa	28453274274
Garrett Fitzpatrick	97588314932
Carl Chapman	48491585487
Fleur Church	58648837498
Gretchen Lawrence	97237695877
Brittany Pollard	94147467348

Herman Payne	59317743778
Emerson Ochoa	28453274274
Garrett Fitzpatrick	97588314932
Carl Chapman	48491585487
Fleur Church	58648837498
Gretchen Lawrence	97237695877
Brittany Pollard	94147467348
Talon Jenkins	77843428477
Morgan Saunders	76543844622
Molly Sharp	45782635435
Lyle Sargent	23279179463
Zelda Nelson	46575854216

Projeto Físico

- 8) SUBCONSULTA DO TIPO TABELA
- 8.A) Nome e gênero das músicas tocadas por um artista

```
SELECT MUSICA.NOME, MUSICA.GENERO
FROM MUSICA
WHERE (MUSICA.ID, MUSICA.NUM) IN (
    SELECT CANTA.ID, CANTA.NUM
    FROM CANTA
    INNER JOIN ARTISTA ON CANTA.ID_ART = ARTISTA.ID
    WHERE ARTISTA.NOME = 'Timon Willis'
);
```

Projeto Físico

NOME_MUSICA	GENERO
Câmeras	Pop
Ela	Forró
E chorou	Pop

Projeto Físico

- 9) OPERAÇÃO DE CONJUNTO
- 9.A) Selecionar artistas que tocam e cantam

```
SELECT A.NOME
FROM ARTISTA A
WHERE A.ID IN (
    SELECT T.ID_MUS
    FROM TOCA T INNER JOIN MUSICA M ON T.ID_CANTA = M.ID AND T.ID_NUM = M.NUM
    INTERSECT
    SELECT C.ID_ART
    FROM CANTA C INNER JOIN ARTISTA A ON C.ID_ART = A.ID
)
```

Projeto Físico

NOME
Timon Willis