

Trabalho Prático de Banco de Dados – CSI440

Aluno: Luan Pereira Barbosa – 17.2.8122

Nome do aplicativo: Gerador de Carteirinhas Estudantis

Descrição Geral: O trabalho foi desenvolvido em Python utilizando o Pyqt5 e Qt designer para implementação da interface gráfica. Como SGBD foi utilizado o SQLite. Realizei modificações no projeto em relação ao que foi enviado nas etapas anteriores, neste documento está contido as duas versões.

Entregas Anteriores (Versão 1)

Sobre o Projeto (Etapa 1):

Este é um sistema que a partir de um banco de dados de alunos gera para cada um uma carteirinha estudantil com validade de cinco anos, o sistema também é capaz de realizar a renovação de uma carteirinha já existente.

Requisitos:

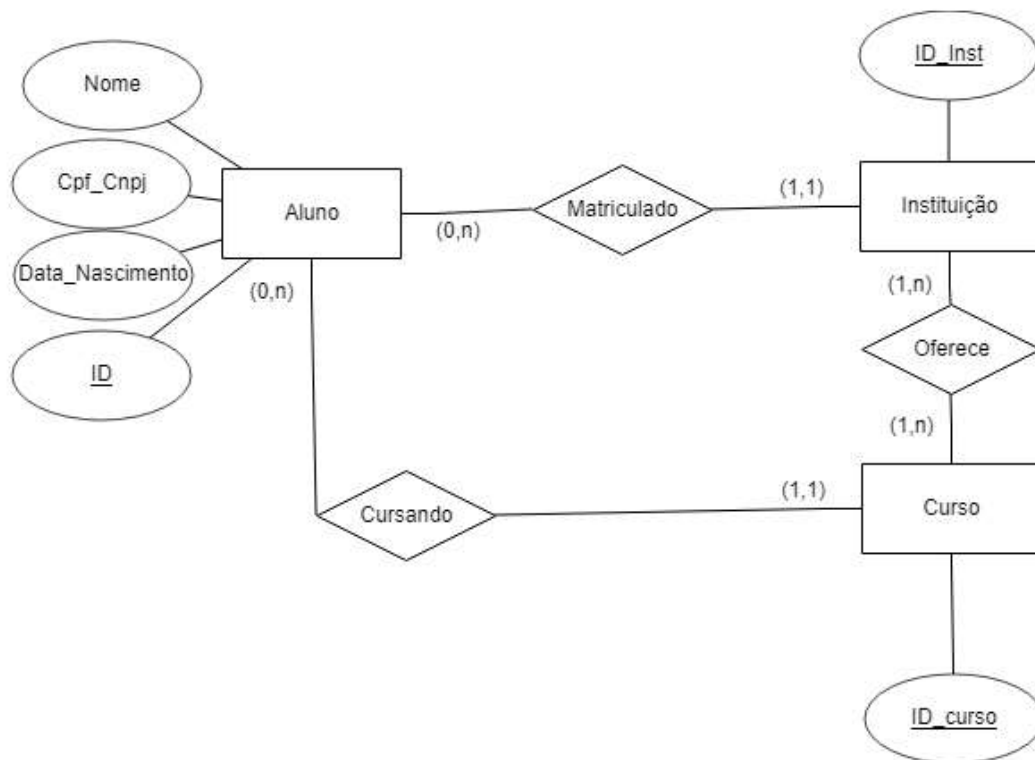
Deve-se armazenar os dados de cada aluno, como: Nome, CPF/CNPJ, data de nascimento, número identificador.

Dados da instituição: ID da instituição, ID do curso.

A partir destes dados verificar que o aluno está devidamente matriculado em um curso e gerar seu documento de identificação.

Para cada aluno é permitida apenas uma carteirinha que pode ser renovada a qualquer momento. O número de matrícula deve ser gerado a partir de um ID de instituição, ID de curso e um ID para o aluno.

Modelagem Conceitual do Banco de Dados (Etapa 2):



Mapeamento Lógico (Etapa 3):

Aluno

Nome	<u>CPF/CNPJ</u>	Data_Nascimento	ID
------	-----------------	-----------------	----

Instituição

<u>ID</u>	<u>ACPF</u>	<u>CID</u>
-----------	-------------	------------

*ACPF referencia Aluno (CPF/CNPJ)

*CID referencia Curso (ID)

Curso

<u>ID</u>	<u>ACPF</u>
-----------	-------------

*ACPF referencia Aluno (CPF/CNPJ)

Oferece

<u>IID</u>	<u>CID</u>
------------	------------

*IID referencia Instituição (ID)

*CID referencia Curso (ID)

Scripts SQL (Etapa 4):

```
CREATE DATABASE "Trabalho_BD"  
  WITH  
  OWNER = postgres  
  ENCODING = 'UTF8'  
  LC_COLLATE = 'Portuguese_Brazil.1252'  
  LC_CTYPE = 'Portuguese_Brazil.1252'  
  TABLESPACE = pg_default  
  CONNECTION LIMIT = -1;
```

```
CREATE SCHEMA public  
  AUTHORIZATION postgres;
```

```
COMMENT ON SCHEMA public  
  IS 'standard public schema';
```

```
GRANT ALL ON SCHEMA public TO PUBLIC;
```

```
GRANT ALL ON SCHEMA public TO postgres;
```

```
CREATE TABLE public."Aluno"  
(  
  "Nome" character varying(40)[] COLLATE pg_catalog."default" NOT NULL,  
  "CPF" character varying(15)[] COLLATE pg_catalog."default" NOT NULL,  
  "Data_Nasc" date NOT NULL,  
  "ID" integer NOT NULL,  
  CONSTRAINT "Aluno_pkey" PRIMARY KEY ("CPF")  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Aluno"  
  OWNER to postgres;
```

```
CREATE TABLE public."Curso"  
(  
  "ID" integer[] NOT NULL,  
  "ACPF" character varying(15)[] COLLATE pg_catalog."default" NOT NULL,  
  CONSTRAINT "Curso_pkey" PRIMARY KEY ("ID"),  
  CONSTRAINT "ACPF" FOREIGN KEY ("ACPF")  
    REFERENCES public."Aluno" ("CPF") MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
  NOT VALID  
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public."Curso"  
  OWNER to postgres;
```

```
CREATE TABLE public."Instituto"  
(  
  "ACPF" character varying(15)[] COLLATE pg_catalog."default" NOT NULL,  
  "ID" integer[] NOT NULL,  
  "CID" integer[] NOT NULL,  
  CONSTRAINT "Instituto_pkey" PRIMARY KEY ("ID"),  
  CONSTRAINT "ACPF" FOREIGN KEY ("ACPF")  
    REFERENCES public."Aluno" ("CPF") MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
  CONSTRAINT "CID" FOREIGN KEY ("CID")  
    REFERENCES public."Curso" ("ID") MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
  NOT VALID  
)  
  
TABLESPACE pg_default;
```

```
ALTER TABLE public."Instituto"  
  OWNER to postgres;
```

```
CREATE TABLE public."Oferece"  
(  
  "IID" integer[] NOT NULL,  
  "CID" integer[],  
  CONSTRAINT "Oferece_pkey" PRIMARY KEY ("IID"),  
  CONSTRAINT "CID" FOREIGN KEY ("IID")  
    REFERENCES public."Curso" ("ID") MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
  NOT VALID,  
  CONSTRAINT "IID" FOREIGN KEY ("IID")  
    REFERENCES public."Instituto" ("ID") MATCH SIMPLE  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
)  
  
TABLESPACE pg_default;
```

```
ALTER TABLE public."Oferece"  
  OWNER to postgres;
```

Protótipos da Interface (Etapa 5):

Infelizmente não consegui enviar os protótipos a tempo para a atividade.

Segunda Versão dos Envios das Etapas

Aqui serão apresentados as novas versões das etapas anteriores a fim de evidenciar as modificações no projeto.

Etapa 1: Descrição do projeto e requisitos básicos.

O projeto consiste em realizar o cadastro de alunos, cursos e instituições de ensino para que a partir destas informações seja criada uma carteirinha estudantil.

Cada carteirinha conta com um código único de identificação gerado a partir da função random com range entre 1 e 100.000. Este código pode ser utilizado para consulta da carteirinha existente.

Dados do aluno:

Nome (character varying)

CPF (int)

Data de nascimento (date)

E-mail (character varying)

Curso (character varying) Chave estrangeira para Curso: código de indentificação.

Instituto (character varying) Chave estrangeira para Instituto: CNPJ

Ano de início (int)

Status, que indica se o aluno está em processo de graduação ou já é graduado. (character varying)

Dados do Instituto:

Nome (character varying)

CNPJ (int)

Endereço (character varying)

Telefone (int)

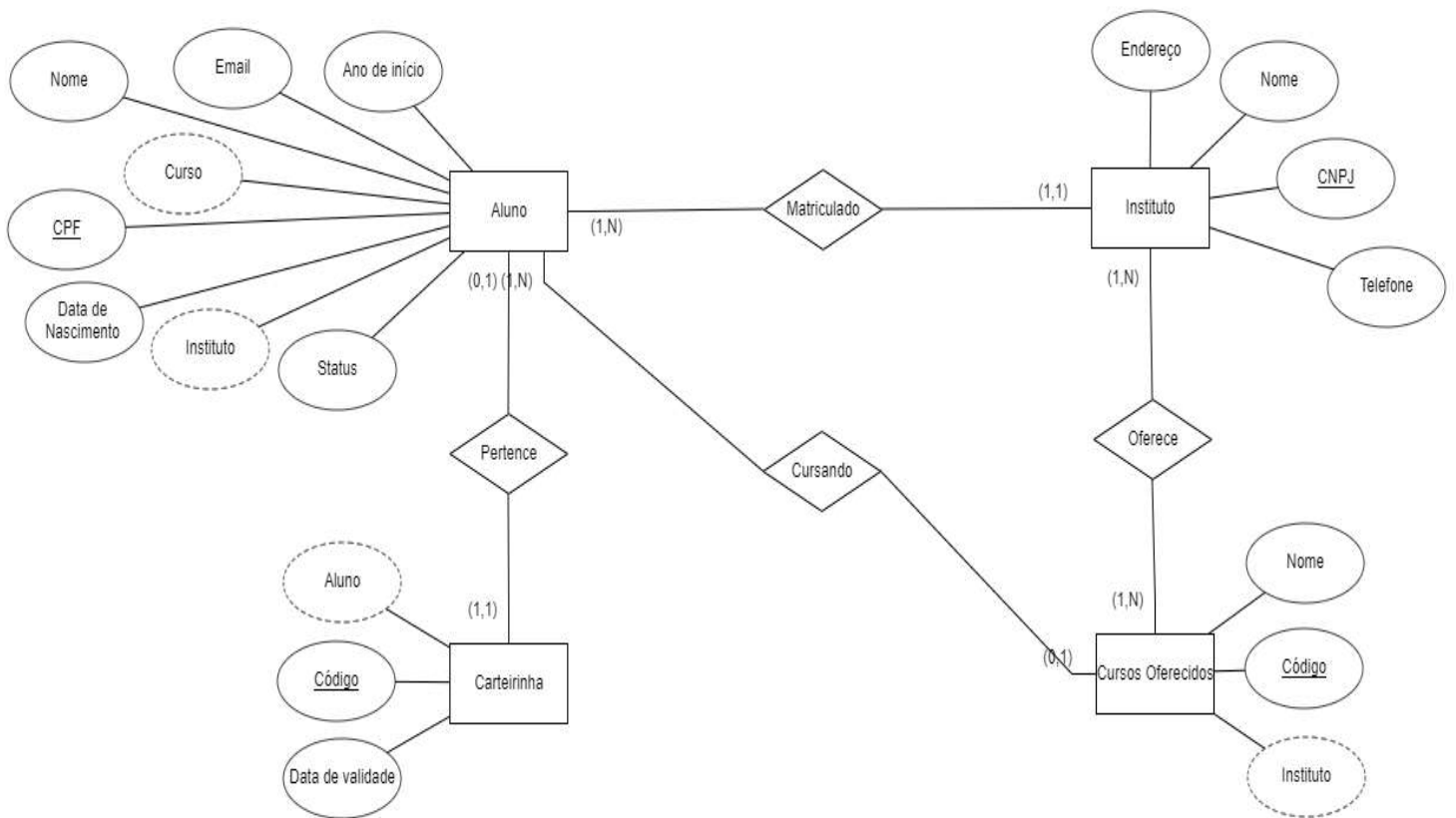
Dados do Curso:

Nome (character varying)

instituto que o oferece (int) Chave estrangeira para Instituto: CNPJ

código de identificação (int)

Etapa 2: Modelagem Conceitual:



Etapa 3: Mapeamento Lógico:

Aluno

Nome	<u>CPF</u>	Data_Nascimento	E-mail	<u>Curso</u>	<u>Instituto</u>	Ano de Início	Status
------	------------	-----------------	--------	--------------	------------------	---------------	--------

*Curso referencia Curso: (Código)

*Instituto referencia Instituto: (CNPJ)

Instituto

Nome	<u>CNPJ</u>	Endereço	Telefone
------	-------------	----------	----------

Cursos Oferecidos

Nome	<u>Código</u>	<u>Instituto</u>
------	---------------	------------------

*Instituto referencia Instituto: (CNPJ)

Carteirinhas

<u>Aluno</u>	<u>Código</u>	Data de Validade
--------------	---------------	------------------

*Aluno referencia Aluno: (CPF)

Etapa 4: Scripts SQL

```
create table instituto
(
    cnpj    int not null
        constraint instituto_pk
            primary key,
    nome    character varying [40],
    endereco character varying [40],
    telefone int
);

create table aluno
(
    nome        character varying [30],
    cpf         int
        constraint aluno_pk
            primary key,
    data_nascimento date,
    email       character varying [30],
    curso       character varying [20],
    instituto   int
        references instituto,
    ano_inicio  int,
    status      character varying [10]
);

create unique index aluno_cpf_uindex
    on aluno (cpf);

create table carteirinhas
(
    aluno    int
        references aluno,
    codigo   int
        constraint carteirinhas_pk
            primary key,
    data_validade date
);

create table cursos_oferecidos
(
    nome        character varying,
    codigo_curso int
        constraint cursos_oferecidos_pk
            primary key,
    instituto   int
        references instituto
);

create unique index cursos_oferecidos_codigo_curso_uindex
    on cursos_oferecidos (codigo_curso);

create unique index instituto_cnpj_uindex
    on instituto (cnpj);
```

*Gerado pelo Pycharm utilizando SQLite.

Etapa 5: Protótipos de Tela

Como eu não havia enviado na atividade, também não inserirei aqui.

Considerações:

Era minha intenção incluir a funcionalidade de inserir uma foto para cada aluno e também gerar um PDF contendo a carteirinha, em ambos os casos não fui capaz de implementar.