

## UC 2 - Implementar Banco de Dados

### Atividades de Classe e Casa

1. Diagramas do slide Modelagem de Dados (Aula 3) -  
<https://drive.google.com/file/d/1eu9lfXeY0muitC4o5TOhaoofsFe0jBC7/view?usp=sharing>
2. Exercícios do slide Introdução à Banco de Dados (Aula 6) -  
[https://drive.google.com/file/d/1qo\\_gXZ\\_m6UsG3oauamnkiAfeqtMZz\\_3S/view?usp=sharing](https://drive.google.com/file/d/1qo_gXZ_m6UsG3oauamnkiAfeqtMZz_3S/view?usp=sharing)  
\*Só é necessário os códigos SQL destes exercícios
3. Resultado final do projeto Biblioteca com base nas instruções da [aula 10](#) presente em nosso github.

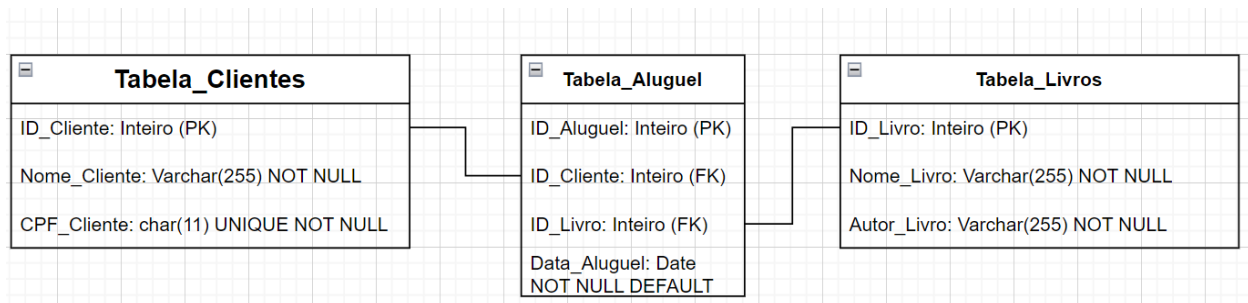
### Atividade Final

Utilizando as técnicas e conhecimentos adquiridos ao longo da unidade você deverá desenvolver uma aplicação de gerenciamento com integração à um banco de dados relacional.

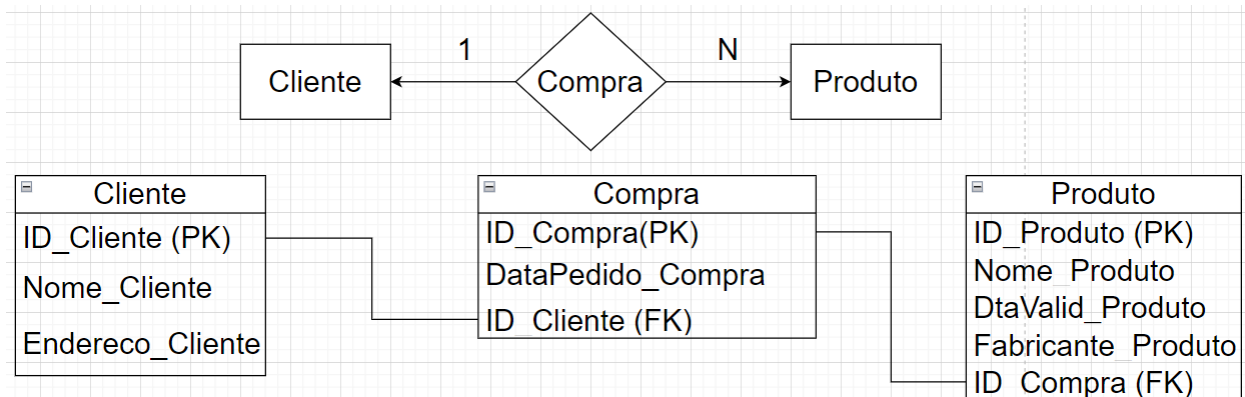
Seguem os requisitos dessa aplicação:

#### Obrigatórios:

1. Utilizar um SGBD para gerenciar os dados de sua aplicação. Sugestões:  
Postgres, MySQL, SQLite
2. Criar um Modelo Físico de Dados da sua aplicação. Deve conter pelo menos 2 entidades e 1 relacionamento. Exemplos:



MFD de aplicação de aluguel de livros



MLD de aplicação de compra de produtos

3. Criar tabelas no SGBD com base na modelagem e planejamento.
4. As tabelas podem ser criadas utilizando a interface gráfica de seus SGBD, mas será necessário fornecer os códigos SQL de criação das tabelas. Exemplo:

```
#Criação de Tabela Cliente
DROP TABLE IF EXISTS "Cliente";
CREATE TABLE "Cliente"(
    "ID" int GENERATED ALWAYS AS IDENTITY,
    "Nome" varchar(255) NOT NULL,
    "CPF" char(11) NOT NULL UNIQUE,
    Primary Key("ID"));
```

5. Criar um programa em Python, ou qualquer linguagem de programação da sua escolha, que se conecte com o banco de dados criado.
6. O programa deve possuir as seguintes funcionalidades:
  - a. Menu de navegação
  - b. Visualizar os dados de cada tabela criada no banco de dados
  - c. Inserção de dados em todas as tabelas
  - d. Atualização de dados em todas as tabelas
  - e. Remoção de dados em todas as tabelas

- f. Inserção em alguma tabela de relacionamento. Exemplo: Usuário da aplicação cadastra o aluguel de um livro. Ele seleciona o Cliente pelo ID, seleciona o Livro alugado e cadastra uma nova entrada na tabela Aluguel.
- 7. Exportar o banco de dados criado e disponibilizar o arquivo sql resultante. No postgres esse arquivo tem a terminalção .pgsql
- 8. Se sua aplicação utilizar algo além das seguintes tecnologias, providenciar instruções para rodar seu projeto. Exemplo::
  - a. Python
  - b. Postgres
  - c. Bibliotecas: psycopg2, faker

### **Sugestões:**

- 9. Utilizar o modelo de organização Modelo, Visualização, Controle
- 10. Utilizar Programação Orientada a Objetos para modelar as classes de Conexão e os tipos de Entidades(Funcionário, Cliente, Livro, etc) do seu programa. Uma vantagem facilmente obtida é poder criar métodos para gerar SQL facilmente e poder fazer validação de dados dentro do objeto.
- 11. Utilizar uma biblioteca de interface gráfica. Sugestão: PySimpleGui

### **12. Comentar o código. Opcional porém ideal.**

### **Considerações:**

É permitido o uso de qualquer ferramenta que considerar adequada.

Desde que os requisitos obrigatórios sejam atendidos, qualquer funcionalidade adicional é considerado como um extra.

A atividade deverá ser feita em duplas.

É permitido consultar os colegas, compartilhar código e pedir ajuda ao professor.

Entrega no dia 22/05/2023

A entrega será feita pelo gitHub como nas atividades anteriores.

### **Recursos:**

Aplicações feitas em sala com requisitos parcialmente atendidos:

<https://github.com/TarikPonciano/Programador-de-Sistema-SENAC/tree/main/Programador%20de%20Sistemas%20-%20UC%202/Reposit%C3%B3rio%20de%20Exemplos%20-%20Turma%201/Aula%2006-02/Biblioteca>

Cursos SQL:

[https://drive.google.com/drive/folders/1GEtDe0Y50MzPBhSPbZYKDHQ8JOyNsVgs?usp=share\\_link](https://drive.google.com/drive/folders/1GEtDe0Y50MzPBhSPbZYKDHQ8JOyNsVgs?usp=share_link)

Diagramas desenvolvidos em sala:

<https://drive.google.com/drive/u/0/folders/19i1gNBv8slldlHD7ljc6RIPxFqbSdjo>