



UNIVERSIDADE FEDERAL DE SANTA CATARINA CAMPUS DE ARARANGUÁ

Engenharia da Computação

(DEC7129) Banco de Dados

Professor Alexandre Leopoldo Gonçalves

Eduardo Pires

Enzo Accioly Adrião

João Pedro Moraes Ribeiro

RELATÓRIO

Acervo Digital

Araranguá 2025

Descrição do Objetivo Geral	3
Descrição Detalhada do Sistema	4
Requisitos Funcionais:	4
Requisitos Não Funcionais:	5
Modelagem Conceitual	6
Domínio de entidades:	6
Modelagem Lógica	7
Domínio de tabelas e chaves:	7
Script DDL	8
Consultas e Análise de Dados	10
Consulta 1: Total de Produções por Editora	10
Consulta 2: Média de Financiamento por Financiador	10
Consulta 3: Produtividade Acadêmica por Pessoa	10

Descrição do Objetivo Geral

O projeto tem como objetivo desenvolver um acervo digital voltado à organização e consulta de trabalhos acadêmicos, artigos científicos e produções universitárias em geral.

A aplicação permitirá o armazenamento estruturado de informações sobre autores, orientadores, cursos, áreas temáticas e instituições, possibilitando buscas eficientes e consultas analíticas sobre os dados cadastrados. Além de servir como ferramenta de apoio à gestão do conhecimento acadêmico, o sistema integrará recursos de inteligência artificial generativa para auxiliar o usuário na contextualização e resumo de trabalhos.

Assim, o projeto visa oferecer uma plataforma modular, extensível e útil para os membros da comunidade acadêmica, promovendo o acesso e a organização do conteúdo científico institucional.

Descrição Detalhada do Sistema

O sistema foi projetado para atender às necessidades de departamentos universitários e grupos de pesquisa que necessitam gerenciar um volume crescente de produções intelectuais. Para a fase de modelagem e implementação, foram levantados os seguintes requisitos:

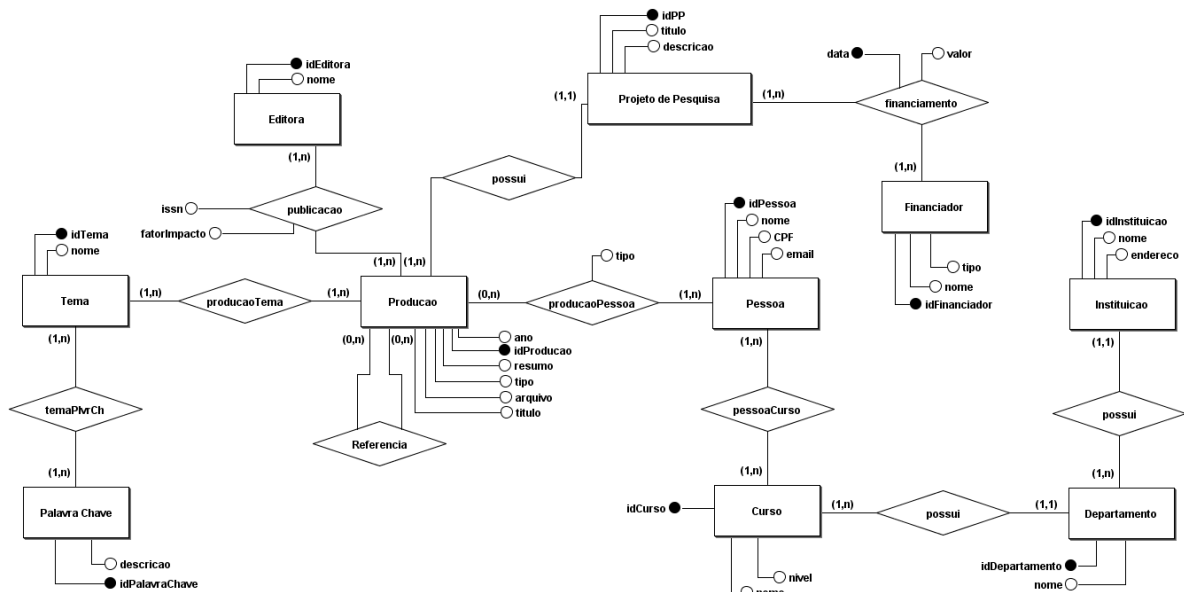
Requisitos Funcionais:

- **Gestão de Produções Acadêmicas:** O sistema deve permitir o cadastro, leitura, atualização e exclusão (CRUD) de diferentes tipos de trabalhos (TCCs, Dissertações, Teses e Artigos). Deve-se armazenar metadados essenciais como título, resumo, ano de publicação e o arquivo digital (PDF) da produção em formato binário (*BYTEA*).
- **Controle de Autoria e Orientação:** O sistema deve registrar as pessoas envolvidas nas produções, classificando-as conforme seu papel (Professor, Aluno ou Técnico). Deve ser possível vincular múltiplas pessoas a uma única produção.
- **Estrutura Institucional e Acadêmica:** O banco de dados deve refletir a hierarquia universitária, permitindo gerenciar Instituições, seus Departamentos e os Cursos vinculados. As pessoas devem estar associadas a estes cursos.
- **Gestão de Financiamentos:** O sistema deve rastrear a origem de recursos financeiros, permitindo o cadastro de Financiadores (Públicos ou Privados) e o registro de aportes financeiros (valores e datas) destinados a Projetos de Pesquisa específicos.
- **Classificação e Indexação:** As produções devem ser classificáveis por Temas e Palavras-chave, além de estarem vinculadas a Editoras (via publicações).
- **Geração de Relatórios Analíticos:** O sistema deve prover consultas pré-definidas que sumarizam as informações do banco, especificamente:
 - **Contagem de produções pela Editora.**
 - **Média de valores de financiamento por entidade financiadora.**
 - **Produtividade individual (número de produções por pessoa).**
- **Administração da Base de Dados:** O sistema deve possuir funcionalidades para "Reiniciar" o banco de dados, executando scripts que apagam todas as tabelas, recriam a estrutura e carregam dados iniciais de teste automaticamente.
- **Consulta em Linguagem Natural (IA):** O sistema deve integrar um módulo de Inteligência Artificial que permita ao usuário interagir com os dados presentes no banco.
- **Análise de Conteúdo de PDF (IA):** O sistema deve ser capaz de ler o conteúdo textual dos arquivos PDF armazenados no banco e, utilizando IA Generativa, responder perguntas ou gerar resumos sobre o conteúdo desses documentos.

Requisitos Não Funcionais:

- **Sistema Gerenciador de Banco de Dados:** A persistência dos dados deve ser realizada utilizando o SGBD PostgreSQL, garantindo conformidade com ACID e integridade referencial.
- **Linguagem de Programação:** A aplicação deve ser desenvolvida em linguagem Python, utilizando drivers nativos (psycopg2) para conexão com o banco de dados.
- **Interface de Usuário:** A interação com o usuário será realizada através de uma interface de linha de comando (Console), estruturada em menus de navegação numéricos.
- **Visualização de Dados:** A geração de gráficos para os relatórios deve utilizar bibliotecas de ciência de dados, como *Pandas* (para manipulação tabular), *Matplotlib* e *Seaborn* (para plotagem gráfica).
- **Integração com LLMs:** Para as funcionalidades de IA, o sistema deve conectar-se a APIs de modelos de linguagem de grande escala (como OpenAI GPT ou Google Gemini) para processamento de linguagem natural.
- **Segurança e Integridade:** O banco de dados deve impor restrições de integridade (*Foreign Keys*, *Unique*, *Check Constraints*) para impedir a inserção de dados inválidos.

Modelagem Conceitual



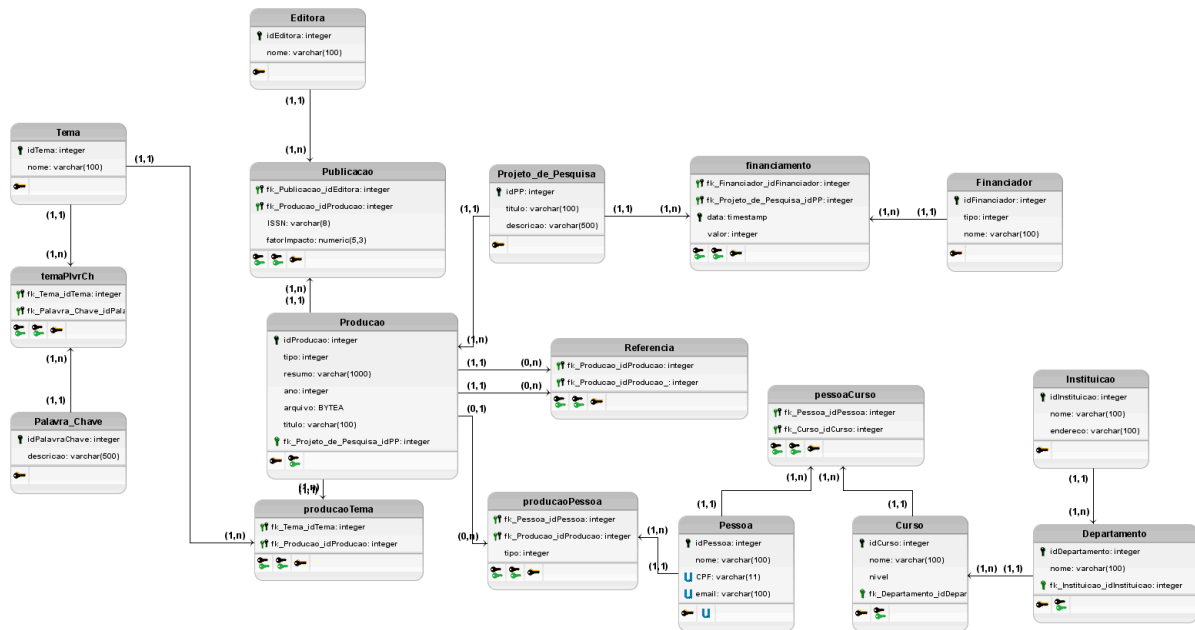
Domínio de entidades:

- **EDITORIA:** Responsável pela publicação dos trabalhos.
- **PROJETO_DE_PESQUISA:** O projeto ao qual uma produção está vinculada.
- **PRODUCAO:** Entidade central que representa o trabalho acadêmico.
- **FINANCIADOR:** Entidade que provê os fundos.
- **TEMA:** Área de conhecimento da produção.
- **PALAVRA_CHAVE:** Termos específicos para indexação.
- **PESSOA:** Indivíduos envolvidos (autores, orientadores).
- **INSTITUICAO:** Organização de ensino ou pesquisa.
- **DEPARTAMENTO:** Subdivisão da instituição.
- **CURSO:** Curso vinculado a um departamento.

Domínio de relacionamentos:

- **PUBLICACAO:** Vincula a produção à editora responsável.
- **FINANCIAMENTO:** Registra o aporte financeiro concedido a um projeto de pesquisa.
- **PRODUCAO_TEMA:** Classifica a produção dentro de uma área de conhecimento.
- **TEMA_PLVRCH:** Associa palavras-chave aos temas.
- **PRODUCAO_PESSOA:** Define a participação (autor, orientador) de uma pessoa na produção.
- **PESSOA_CURSO:** Vincula indivíduos (alunos/professores) aos seus respectivos cursos.
- **REFERENCIA:** Auto-relacionamento que indica as citações bibliográficas entre produções.
- **POSSUI:** Indica pertencimento ou hierarquia entre entidades (Projeto-Produção, Curso-Departamento, Departamento-Instituição).

Modelagem Lógica



Domínio de chaves:

- **EDITORA:** (id_editora [PK], nome)
- **PROJETO_DE_PESQUISA:** (id_projeto_pesq [PK], titulo, descricao)
- **PRODUCAO:** (id_producao [PK], id_projeto_pesq [FK], titulo, tipo, resumo, ano, arquivo)
- **PUBLICACAO:** (id_editora [PK/FK], id_producao [PK/FK], fator_impacto, ISSN)
- **REFERENCIA:** (id_producao_referente [PK/FK], id_producao_referenciada [PK/FK])
- **FINANCIADOR:** (id_financiador [PK], tipo, nome)
- **FINANCIAMENTO:** (id_financiador [PK/FK], id_projeto_pesq [PK/FK], data [PK], valor)
- **TEMA:** (id_tema [PK], nome)
- **PRODUCAO_TEMA:** (id_tema [PK/FK], id_producao [PK/FK])
- **PALAVRA_CHAVE:** (id_palavra_chave [PK], descricao)
- **TEMA_PALAVRA_CHAVE:** (id_tema [PK/FK], id_palavra_chave [PK/FK])
- **PESSOA:** (id_pessoa [PK], nome, CPF, email)
- **PRODUCAO_PESSOA:** (id_pessoa [PK/FK], id_producao [PK/FK], tipo)
- **INSTITUICAO:** (id_instituicao [PK], nome, endereco)
- **DEPARTAMENTO:** (id_departamento [PK], id_instituicao [FK], nome)
- **CURSO:** (id_curso [PK], id_departamento [FK], nome, nivel)
- **PESSOA_CURSO:** (id_pessoa [PK/FK], id_curso [PK/FK])

Script DDL

```
CREATE TABLE EDITORA (  
    id_editora INTEGER PRIMARY KEY NOT NULL,  
    nome VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE PROJETO_DE_PESQUISA (  
    id_projeto_pesq INTEGER PRIMARY KEY NOT NULL,  
    titulo VARCHAR(100) NOT NULL,  
    descricao VARCHAR(500) NOT NULL  
);  
  
CREATE TABLE PRODUCAO (  
    id_producao INTEGER PRIMARY KEY NOT NULL,  
    id_projeto_pesq INTEGER NOT NULL,  
    titulo VARCHAR(100) NOT NULL,  
    tipo VARCHAR(50) NOT NULL,  
    resumo VARCHAR(1000) NOT NULL,  
    ano INTEGER NOT NULL,  
    arquivo BLOB,  
    FOREIGN KEY (id_projeto_pesq) REFERENCES PROJETO_DE_PESQUISA (id_projeto_pesq),  
    CHECK (tipo IN ('TCC', 'Dissertação', 'Artigo', 'Tese'))  
);  
  
CREATE TABLE PUBLICACAO (  
    id_editora INTEGER NOT NULL,  
    id_producao INTEGER NOT NULL,  
    ISSN VARCHAR(9) NOT NULL,  
    fatorImpacto NUMERIC(5, 3),  
    PRIMARY KEY (id_editora, id_producao),  
    FOREIGN KEY (id_editora) REFERENCES EDITORA (id_editora),  
    FOREIGN KEY (id_producao) REFERENCES PRODUCAO (id_producao)  
);  
  
CREATE TABLE REFERENCIA (  
    id_producao_referente INTEGER NOT NULL,  
    id_producao_referenciada INTEGER NOT NULL,  
    PRIMARY KEY (id_producao_referente, id_producao_referenciada),  
    FOREIGN KEY (id_producao_referente) REFERENCES PRODUCAO (id_producao),  
    FOREIGN KEY (id_producao_referenciada) REFERENCES PRODUCAO (id_producao)  
);  
  
CREATE TABLE FINANCIADOR (  
    id_financiador INTEGER PRIMARY KEY NOT NULL,  
    tipo VARCHAR(50) NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    CHECK (tipo IN ('Público', 'Privado'))  
);  
  
CREATE TABLE FINANCIAMENTO (  
    id_financiador INTEGER NOT NULL,  
    id_projeto_pesq INTEGER NOT NULL,  
    valor INTEGER NOT NULL,  
    data_financiamento DATE NOT NULL,  
    PRIMARY KEY (id_financiador, id_projeto_pesq, data_financiamento),  
    FOREIGN KEY (id_financiador) REFERENCES FINANCIADOR (id_financiador),  
    FOREIGN KEY (id_projeto_pesq) REFERENCES PROJETO_DE_PESQUISA (id_projeto_pesq)  
);  
  
CREATE TABLE TEMA (  
    id_tema INTEGER PRIMARY KEY NOT NULL,  
    nome VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE PRODUCAO_TEMA (  
    id_tema INTEGER NOT NULL,  
    id_producao INTEGER NOT NULL,  
    PRIMARY KEY (id_tema, id_producao),  
    FOREIGN KEY (id_tema) REFERENCES TEMA (id_tema),  
    FOREIGN KEY (id_producao) REFERENCES PRODUCAO (id_producao)
```



```

);

CREATE TABLE PALAVRA_CHAVE (
    id_palavra_chave INTEGER PRIMARY KEY NOT NULL,
    descricao VARCHAR(50) NOT NULL
);

CREATE TABLE TEMA_PALAVRA_CHAVE (
    id_tema INTEGER NOT NULL,
    id_palavra_chave INTEGER NOT NULL,
    PRIMARY KEY (id_tema, id_palavra_chave),
    FOREIGN KEY (id_tema) REFERENCES TEMA (id_tema),
    FOREIGN KEY (id_palavra_chave) REFERENCES PALAVRA_CHAVE (id_palavra_chave)
);

CREATE TABLE PESSOA (
    id_pessoa INTEGER PRIMARY KEY NOT NULL,
    nome VARCHAR(100) NOT NULL,
    CPF VARCHAR(11) NOT NULL,
    email VARCHAR(100) NOT NULL,
    UNIQUE (email),
    UNIQUE (CPF)
);

CREATE TABLE PRODUCAO_PESSOA (
    id_pessoa INTEGER NOT NULL,
    id_producao INTEGER NOT NULL,
    tipo VARCHAR(50) NOT NULL,
    PRIMARY KEY (id_pessoa, id_producao),
    FOREIGN KEY (id_pessoa) REFERENCES PESSOA (id_pessoa),
    FOREIGN KEY (id_producao) REFERENCES PRODUCAO (id_producao),
    CHECK (tipo IN ('Professor', 'Aluno', 'Técnico'))
);

CREATE TABLE INSTITUICAO (
    id_instituicao INTEGER PRIMARY KEY NOT NULL,
    nome VARCHAR(100) NOT NULL,
    endereco VARCHAR(100) NOT NULL
);

CREATE TABLE DEPARTAMENTO (
    id_departamento INTEGER PRIMARY KEY NOT NULL,
    id_instituicao INTEGER NOT NULL,
    nome VARCHAR(100) NOT NULL,
    FOREIGN KEY (id_instituicao) REFERENCES INSTITUICAO (id_instituicao)
);

CREATE TABLE CURSO (
    id_curso INTEGER PRIMARY KEY NOT NULL,
    id_departamento INTEGER NOT NULL,
    nome VARCHAR(100) NOT NULL,
    nivel VARCHAR (30) NOT NULL,
    FOREIGN KEY (id_departamento) REFERENCES DEPARTAMENTO (id_departamento),
    CHECK (nivel IN ('Graduação', 'Mestrado', 'Doutorado'))
);

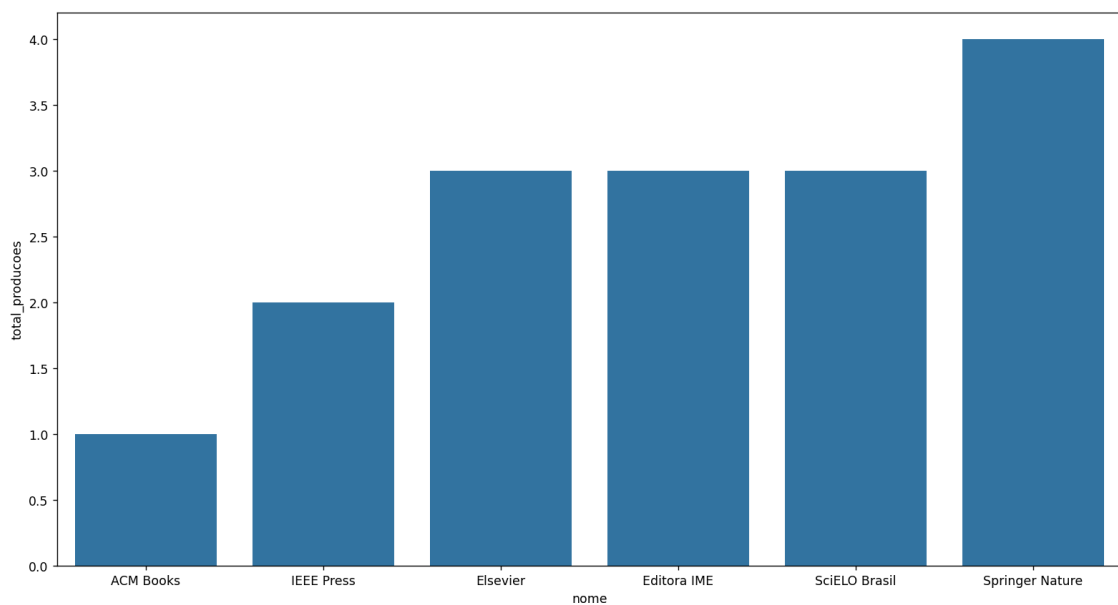
CREATE TABLE PESSOA_CURSO (
    id_pessoa INTEGER NOT NULL,
    id_curso INTEGER NOT NULL,
    PRIMARY KEY (id_pessoa, id_curso),
    FOREIGN KEY (id_pessoa) REFERENCES PESSOA (id_pessoa),
    FOREIGN KEY (id_curso) REFERENCES CURSO (id_curso));

```

Consultas e Análise de Dados

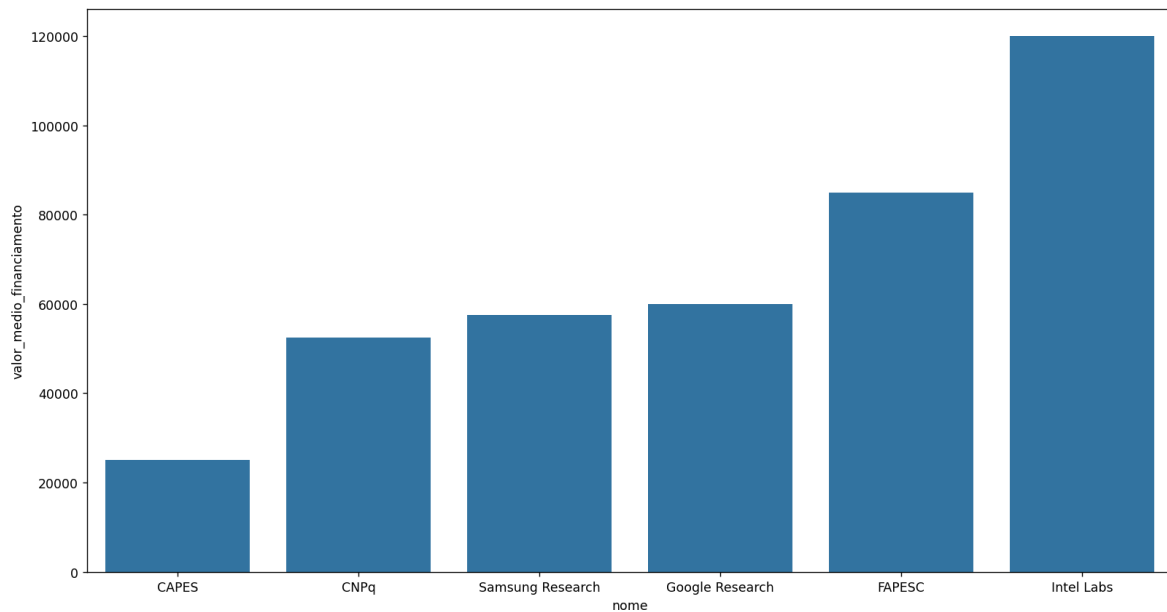
Consulta 1: Total de Publicações por Editora

```
SELECT E.nome, COUNT(P.id_producao) AS total_producoes
FROM PRODUCAO P
JOIN PUBLICACAO PU ON P.id_producao = PU.id_producao
JOIN EDITORA E ON PU.id_editora = E.id_editora
GROUP BY E.nome
ORDER BY total_producoes;
```



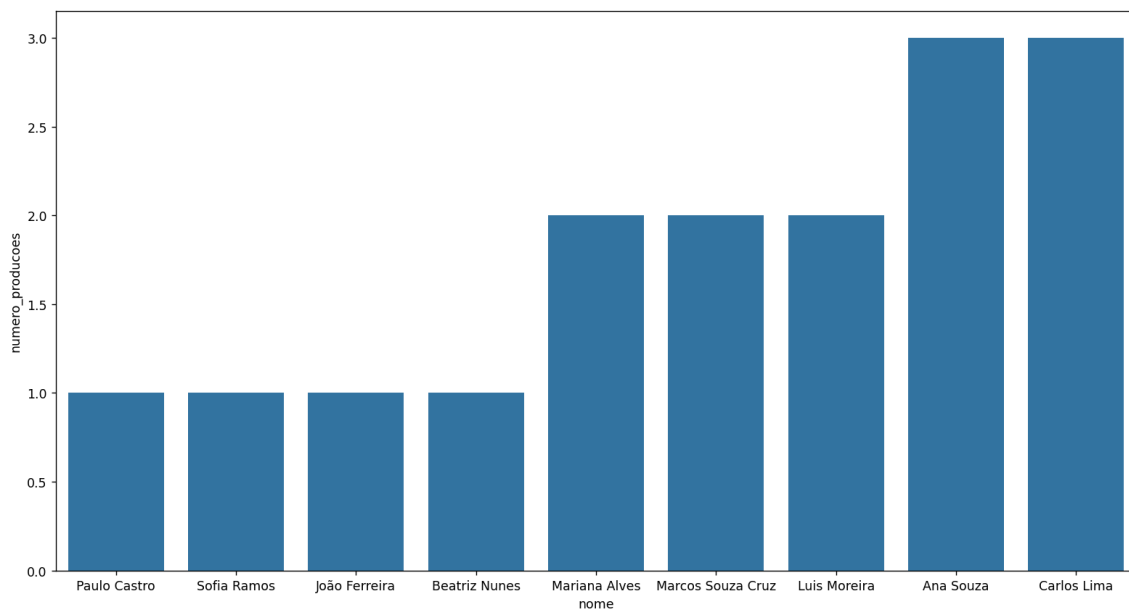
Consulta 2: Média de Financiamento por Financiador

```
SELECT F.nome, AVG(FI.valor) AS valor_medio_financiamento
FROM FINANCIAMENTO FI
JOIN FINANCIADOR F ON FI.id_financiador = F.id_financiador
JOIN PROJETO_DE_PESQUISA PP ON FI.id_projeto_pesq = PP.id_projeto_pesq
GROUP BY F.nome
ORDER BY valor_medio_financiamento;
```



Consulta 3: Produtividade Acadêmica por Pessoa

```
SELECT P.nome, COUNT(PP.id_producao) AS numero_producoes
FROM PRODUCAO_PESSOA PP
JOIN PESSOA P ON PP.id_pessoa = P.id_pessoa
JOIN PRODUCAO PR ON PP.id_producao = PR.id_producao
GROUP BY P.nome
ORDER BY numero_producoes;
```



Link do projeto no github: <https://github.com/agni000/Acervo-Digital/tree/main>