

## RELATÓRIO

**1. Regra do Negócio:** Com base no artigo, podemos abstrair algumas regras de negócios importantes que podem ser usadas para elaborar um banco de dados eficiente para gerenciar o protocolo de uma empresa.

- **Pastas de Documentos:** Cada documento protocolado está associado a uma pasta de documentos. As pastas são codificadas sequencialmente (número de ordem/ano) e arquivadas. As pastas contêm documentos relacionados a um determinado assunto, dispostos em ordem cronológica.
- **Fichas de Circulação:** Cada pasta ou documento deve ter uma ficha de circulação associada a ela. Essas fichas devem conter informações sobre a origem, destino, motivo da circulação, data e assinaturas de origem e assinaturas de destino. As fichas de circulação acompanham as pastas ou documentos em sua circulação.
- **Folhas de Indexação:** Cada documento é indexado com base em seu assunto principal e outros descritores relevantes. A indexação é realizada por pessoal qualificado. As Folhas de Indexação formam um arquivo importante para recuperação de informações.
- **Livro de Registro das Pastas:** Este livro tem por finalidade associar o número de cada pasta aos documentos nela contidos e ao assunto da pasta. É um registro sequencial por número da pasta, e onde está prevista a possibilidade de anexar e desanexar documentos.
- **Livro de Registro de Documentos:** Possui a mesma finalidade que o livro anterior, mas aplicado a cada documento individualmente.

Agora seguiremos para as Rotinas de Arquivamentos, onde os processos a serem seguidos para o registro de informações no Protocolo.

- **Rotinas de Expedição e Recepção:** Devem existir rotinas claras para a expedição e recepção de pastas e documentos, garantindo que as informações sejam entregues aos destinatários corretos e que sejam registradas adequadamente.
- **Rotina de Anexação/Desanexação:** Deve haver uma rotina para anexar ou desanexar documentos de pastas, mantendo o controle adequado sobre quais documentos estão associados a quais pastas.
- **Rotinas de Consulta:** O sistema de banco de dados deve permitir consultas eficientes para responder a perguntas comuns, como a localização de pastas, a data da última circulação de uma pasta, a lista de documentos relacionados a um determinado assunto, entre outras.

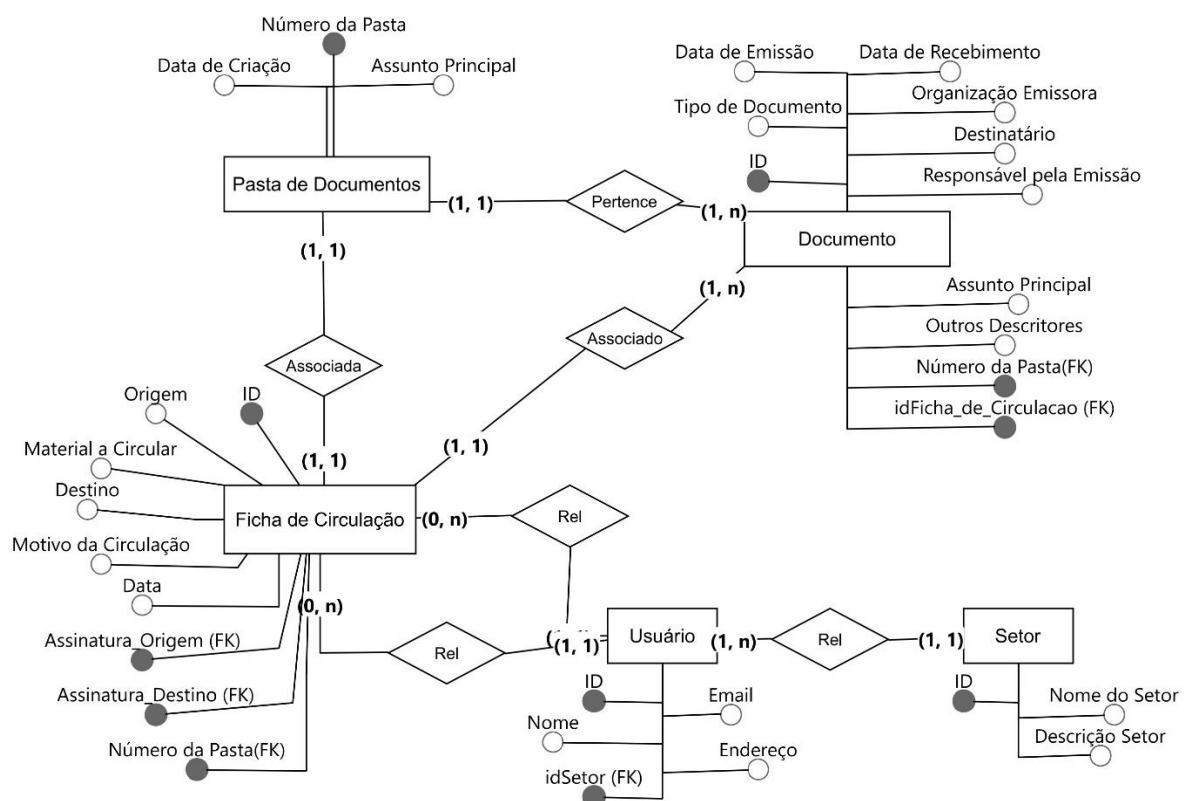
Finalizando essas rotinas, às vezes é preciso fazer a atualização ou recuperação rápida das informações no sistema de banco de dados. Essas etapas devem seguir algumas regras como:

- **Atualização de Registros:** Sempre que houver uma mudança, como a desanexação ou anexação de documentos, as informações devem ser atualizadas nos registros apropriados.
- **Recuperação Rápida de Informações:** O sistema de banco de dados deve ser projetado para fornecer respostas rápidas a consultas, garantindo que o protocolo funcione de forma eficiente.

**2. Modelagem Conceitual:** Um modelo conceitual de banco de dados é uma representação abstrata dos dados e das relações entre eles em um domínio de aplicação. Ele serve como um guia fundamental para o projeto e desenvolvimento de um banco de dados, ajudando a garantir que os dados sejam organizados e armazenados de maneira eficaz e atendendo aos requisitos da aplicação.

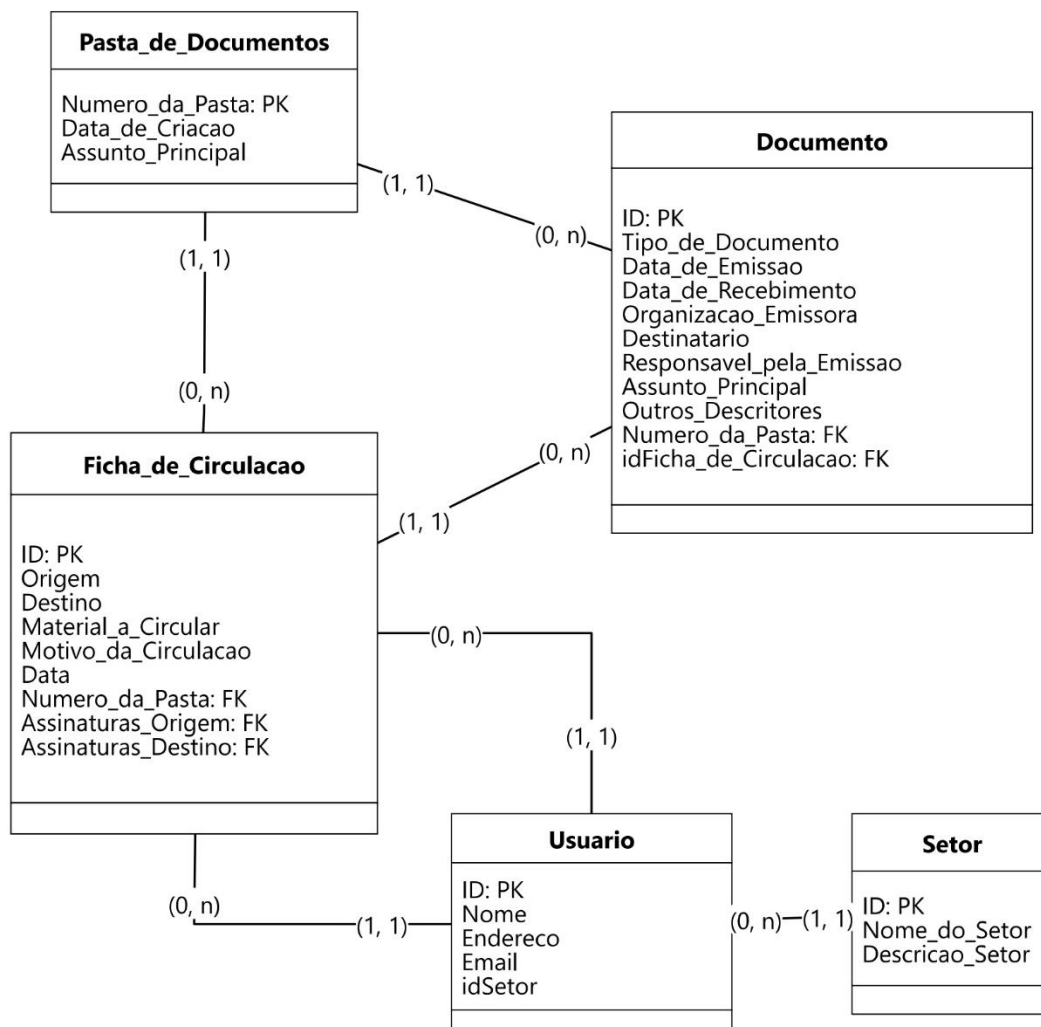
A técnica mais difundida de modelagem conceitual é a abordagem entidade-relacionamento (ER). Nesta técnica, um modelo conceitual é usualmente representado através de um diagrama, chamado diagrama entidade-relacionamento (DER) (Heuser, 1998).

No nosso exemplo utilizaremos cinco entidades (Pasta de Documentos, Documentos, Ficha de Circulação, Usuário e Setor) como mostra na figura abaixo:



**3. Modelagem Lógica:** Um modelo lógico de banco de dados é uma representação mais detalhada e estruturada dos dados e de suas relações em comparação com o modelo conceitual. É importante observar que o modelo lógico é mais próximo da implementação física do banco de dados do que o modelo conceitual, mas ainda é independente de um SGBD específico. Ele fornece uma estrutura de dados abstrata que pode ser usada como base para a criação de um esquema de banco de dados real em um SGBD específico, como MySQL, Oracle, SQL Server, entre outros.

No nosso exemplo continuaremos utilizando as cinco entidades para gerar o modelo lógico de nosso sistema, como mostra abaixo na imagem.



**4. Implementação Física:** A implementação física de um banco de dados refere-se à criação concreta e à configuração detalhada do banco de dados com base no modelo lógico ou conceitual previamente definido. É a fase em que o design abstrato do banco de dados é transformado em uma estrutura física real que será armazenada em dispositivos de armazenamento, como discos rígidos, memória RAM e outros meios de armazenamento.

A implementação física de nosso exemplo seguirá uma ordem cronológica das criações das tabelas, alterações das chaves estrangeiras, inserção dos dados nas tabelas criadas e algumas consultas.

O script abaixo mostra como foi feita a criação das tabelas para o nosso exemplo.

```
CREATE TABLE Pasta_de_Documentos
(
  Numero_da_Pasta INT PRIMARY KEY,
  Data_de_Criacao DATE NOT NULL,
  Assunto_Principal VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Documento
(
  ID INT PRIMARY KEY,
  Tipo_de_Documento VARCHAR(255) NOT NULL,
  Data_de_Emissao DATE NOT NULL,
  Data_de_Recebimento DATE NOT NULL,
  Organizacao_Emissora VARCHAR(255) NOT NULL,
  Destinatario VARCHAR(255) NOT NULL,
  Responsavel_pela_Emissao VARCHAR(255) NOT NULL,
  Assunto_Principal VARCHAR(255) NOT NULL,
  Outros_Descriptores VARCHAR(255) NOT NULL,
  Numero_da_Pasta INT NOT NULL,
  idFicha_de_Circulacao INT NOT NULL
);
```

```
CREATE TABLE Ficha_de_Circulacao
(
  ID INT PRIMARY KEY,
  Origem VARCHAR(255) NOT NULL,
  Destino VARCHAR(255) NOT NULL,
  Material_a_Circular VARCHAR(255) NOT NULL,
  Motivo_da_Circulacao VARCHAR(255) NOT NULL,
  Data DATE NOT NULL,
  Numero_da_Pasta INT NOT NULL,
  Assinaturas_Origem INT NOT NULL,
  Assinaturas_Destino INT
);
```

```
CREATE TABLE Usuario
(
  ID INT PRIMARY KEY,
  Nome VARCHAR(255) NOT NULL,
  Endereco VARCHAR(255) NOT NULL,
  Email VARCHAR(255) NOT NULL,
  idSetor INT NOT NULL
);
```

```
CREATE TABLE Setor
(
  ID INT PRIMARY KEY,
  Nome_do_Setor VARCHAR(255) NOT NULL,
  Descricao_Setor VARCHAR(255) NOT NULL
);
```

Após as criações das tabelas, criaremos as chaves estrangeiras para fazer os relacionamentos entre elas. O script abaixo mostra como eles foram incrementadas.

```
ALTER TABLE Documento ADD FOREIGN KEY(idFicha_de_Circulacao) REFERENCES
Ficha_de_Circulacao (ID)
```

```
ALTER TABLE Documento ADD FOREIGN KEY(Numero_da_Pasta) REFERENCES Pasta_de_Documentos
(Numero_da_Pasta)
```

```
ALTER TABLE Ficha_de_Circulacao ADD FOREIGN KEY(Numero_da_Pasta) REFERENCES
Pasta_de_Documentos (Numero_da_Pasta)
```

```
ALTER TABLE Ficha_de_Circulacao ADD FOREIGN KEY(Assinaturas_Origem) REFERENCES Usuario
(ID)
```

```
ALTER TABLE Ficha_de_Circulacao ADD FOREIGN KEY(Assinaturas_Destino) REFERENCES Usuario
(ID)
```

```
ALTER TABLE Usuario ADD FOREIGN KEY(idSetor) REFERENCES Setor (ID)
```

Após as alterações das chaves estrangeiras às tabelas, que é um passo importante para manter a integridade referencial dos dados. Agora iremos inserir os dados fictícios nas tabelas, utilizaremos 8 incrementos em cada, como mostra no script abaixo

#### **-- Inserir dados na tabela Setor**

```
INSERT INTO Setor (ID, Nome_do_Setor, Descricao_Setor)
VALUES
  (1, 'Recursos Humanos', 'Setor de Recursos Humanos'),
  (2, 'Tecnologia da Informação', 'Setor de TI'),
  (3, 'Financeiro', 'Departamento Financeiro'),
  (4, 'Vendas', 'Equipe de Vendas'),
  (5, 'Produção', 'Setor de Produção'),
  (6, 'Logística', 'Logística e Transporte'),
```

(7, 'Marketing', 'Departamento de Marketing'),  
(8, 'Jurídico', 'Departamento Jurídico');

**-- Inserir dados na tabela Usuario**

INSERT INTO Usuario (ID, Nome, Endereco, Email, idSetor)

VALUES

(1, 'João Silva', 'Rua A, 123', 'joao@email.com', 1),  
(2, 'Maria Santos', 'Av. B, 456', 'maria@email.com', 2),  
(3, 'Carlos Souza', 'Rua C, 789', 'carlos@email.com', 3),  
(4, 'Ana Pereira', 'Av. D, 101', 'ana@email.com', 1),  
(5, 'Pedro Ferreira', 'Rua E, 202', 'pedro@email.com', 4),  
(6, 'Lucia Lima', 'Av. F, 303', 'lucia@email.com', 5),  
(7, 'Marcos Oliveira', 'Rua G, 404', 'marcos@email.com', 3),  
(8, 'Cristina Alves', 'Av. H, 505', 'cristina@email.com', 2);

**-- Inserir dados na tabela Pasta\_de\_Documentos**

INSERT INTO Pasta\_de\_Documentos (Numero\_da\_Pasta, Data\_de\_Criacao, Assunto\_Principal)

VALUES

(1, '2023-01-01', 'Documentos de RH'),  
(2, '2023-02-15', 'Documentos de Vendas'),  
(3, '2023-03-10', 'Documentos Financeiros'),  
(4, '2023-04-20', 'Documentos de TI'),  
(5, '2023-05-05', 'Documentos de Marketing'),  
(6, '2023-06-30', 'Documentos de Produção'),  
(7, '2023-07-22', 'Documentos Jurídicos'),  
(8, '2023-08-18', 'Documentos de Logística');

**-- Inserir dados na tabela Ficha\_de\_Circulacao**

INSERT INTO Ficha\_de\_Circulacao (ID, Origem, Destino, Material\_a\_Circular, Motivo\_da\_Circulacao, Data, Numero\_da\_Pasta, Assinaturas\_Origem, Assinaturas\_Destino)

VALUES

(1, 'Setor A', 'Setor B', 'Relatório', 'Revisão', '2023-01-10', 1, 2, 3),  
(2, 'Setor B', 'Setor C', 'Contrato', 'Aprovação', '2023-02-20', 2, 3, 1),  
(3, 'Setor C', 'Setor D', 'Apresentação', 'Análise', '2023-03-15', 3, 1, 2),  
(4, 'Setor D', 'Setor A', 'Proposta', 'Revisão', '2023-04-25', 4, 2, 3),  
(5, 'Setor E', 'Setor F', 'Orçamento', 'Aprovação', '2023-05-15', 5, 3, 1),  
(6, 'Setor F', 'Setor G', 'Relatório', 'Análise', '2023-06-05', 6, 1, 2),  
(7, 'Setor G', 'Setor H', 'Contrato', 'Revisão', '2023-07-30', 7, 2, 3),  
(8, 'Setor H', 'Setor A', 'Apresentação', 'Aprovação', '2023-08-28', 8, 3, 1);

**-- Inserir dados na tabela Documento**

INSERT INTO Documento (ID, Tipo\_de\_Documento, Data\_de\_Emissao, Data\_de\_Recebimento, Organizacao\_Emissora, Destinatario, Responsavel\_pela\_Emissao, Assunto\_Principal, Outros\_Descritores, Numero\_da\_Pasta, idFicha\_de\_Circulacao)

VALUES

(1, 'Ofício', '2023-01-02', '2023-01-12', 'Empresa X', 'Setor A', 'João Silva', 'Ofício de RH', 'RH', 1, 1),  
(2, 'Contrato', '2023-02-16', '2023-02-25', 'Empresa Y', 'Setor B', 'Maria Santos', 'Contrato de Vendas', 'Vendas', 2, 2),  
(3, 'Nota Fiscal', '2023-03-11', '2023-03-20', 'Empresa Z', 'Setor C', 'Carlos Souza', 'Nota Fiscal', 'Financeiro', 3, 3),  
(4, 'Relatório', '2023-04-21', '2023-04-30', 'Empresa A', 'Setor D', 'Ana Pereira', 'Relatório de TI', 'TI', 4, 4),

(5, 'Apresentação', '2023-05-06', '2023-05-16', 'Empresa B', 'Setor E', 'Pedro Ferreira', 'Apresentação de Marketing', 'Marketing', 5, 5),

(6, 'Proposta', '2023-06-01', '2023-06-10', 'Empresa C', 'Setor F', 'Lucia Lima', 'Proposta de Produção', 'Produção', 6, 6),

(7, 'Contrato', '2023-07-23', '2023-08-01', 'Empresa D', 'Setor G', 'Marcos Oliveira', 'Contrato Jurídico', 'Jurídico', 7, 7),

(8, 'Ofício', '2023-08-19', '2023-08-29', 'Empresa E', 'Setor H', 'Cristina Alves', 'Ofício de Logística', 'Logística', 8, 8);

**5. SQL:** Após a implementação dos dados vamos procurar responder alguns questionamento da página inicial do artigo, bem como alguns que achar relevante para a atividade.

1. Que pasta trata de determinado assunto?

Nessa primeira pergunta vamos procurar responder através de consultas que pasta trata do assunto “**Documentos Financeiros**”. Para responder a esta consulta iremos recuperar o número da pasta e o assunto principal da **tabela Pasta\_de\_Documentos** utilizando o script abaixo:

```
SELECT Numero_da_Pasta, Assunto_Principal
FROM Pasta_de_Documentos
WHERE Assunto_Principal = 'Documentos Financeiros';
```

O resultado da consulta mostrará o número da pasta e o assunto principal correspondente, se houver algum.

2. Onde está essa pasta?

Nessa pergunta iremos juntar a pergunta da primeira questão com o questionamento de onde está a pasta referente a este assunto. Para isso a consulta irá utilizar a junção(JOIN) de duas tabelas a **Pasta\_de\_Documentos(pd)** e **Ficha\_de\_Circulacao(fc)**, irá selecionar o número da pasta, o assunto principal e a origem da ficha de circulação onde o assunto principal é "Documentos Financeiros".

```
SELECT pd.Numero_da_Pasta, pd.Assunto_Principal, fc.Origem
FROM Pasta_de_Documentos pd
INNER JOIN FichadeCirculacao fc ON pd.Numero_da_Pasta = fc.Numero_da_Pasta
WHERE pd.Assunto_Principal = 'Documentos Financeiros';
```

O resultado da consulta mostrará o número da pasta, o assunto principal e a origem da ficha de circulação relacionada a essa pasta. Isso lhe dirá qual pasta trata do assunto "Documentos Financeiros" e de onde essa pasta está circulando.

3. Há quanto tempo essa pasta está nesse lugar?

A pergunta agora é a quanto tempo a pasta com assunto **Documentos Financeiros** está no **Setor C** a consulta novamente faz uma junção das tabelas **Pasta\_de\_Documentos(pd)** e **Ficha\_de\_Circulacao(fc)** usando o campo “Numero\_da\_Pasta” em seguida filtra o assunto para aquele que o usuário bem como o Setor. O script utilizado para essa consulta é este:

```

SELECT
    pd.Numero_da_Pasta,
    pd.Assunto_Principal,
    fc.Origem AS Local_Especifico,
    CURRENT_DATE - MAX(fc.Data) AS Tempo_em_Dias
FROM Pasta_de_Documentos pd
INNER JOIN Ficha_de_Circulacao fc ON pd.Numero_da_Pasta = fc.Numero_da_Pasta
WHERE pd.Assunto_Principal = 'Documentos Financeiros'
    AND fc.Origem = 'Setor C'
GROUP BY pd.Numero_da_Pasta, pd.Assunto_Principal, fc.Origem;

```

O resultado da consulta mostrará o número da pasta, o assunto principal, o "Setor C" e o tempo (em dias) desde a última entrada da pasta nesse local.

#### 4. Qual foi o usuário que assinou o Ofício para a organização emissora X ?

Nessa pergunta utilizarem as tabelas **Usuario (u)**, **Ficha\_de\_Circulacao (fc)**, e **Documento (d)** usando as chaves estrangeiras **Assinaturas\_Origem** e **idFicha\_de\_Circulacao** para conectar as tabelas. Assim pegar apenas os documentos do tipo "Ofício" com a organização emissora "Empresa X" como mostra o código abaixo.

```

SELECT u.Nome AS Nome_do_Usuario
FROM Usuario u
INNER JOIN Ficha_de_Circulacao fc ON u.ID = fc.Assinaturas_Origem
INNER JOIN Documento d ON fc.ID = d.idFicha_de_Circulacao
WHERE d.Tipo_de_Documento = 'Ofício'
    AND d.Organizacao_Emissora = 'Empresa X';

```

O resultado da consulta mostrará o nome do usuário que assinou a origem do documento do tipo "Ofício" da organização emissora "Empresa X".

#### 5. Quais são os documentos que o responsável Pedro Ferreira assinou?

Fazemos uma junção entre as tabelas **Documento (d)** e **Usuario (u)** usando o campo **Responsavel\_pela\_Emissao**, que é o campo onde o nome do usuário que assinou o documento é armazenado. Filtramos os resultados para pegar apenas os documentos para os quais o **Responsavel\_pela\_Emissao** é igual a "**Pedro Ferreira**".

```

SELECT d.Tipo_de_Documento AS Nome_do_Documento,
    d.Assunto_Principal AS Assunto_do_Documento,
    d.Data_de_Emissao AS Data_de_Emissao
FROM Documento d
INNER JOIN Usuario u ON d.Responsavel_pela_Emissao = u.Nome
WHERE u.Nome = 'Pedro Ferreira';

```

O resultado da consulta mostrará os nomes dos documentos, os assuntos e as datas de emissão dos documentos que Pedro Ferreira assinou.

#### 6. Quais são os documentos que cada um dos usuários assinou e para qual setor foi designado?

As tabelas utilizadas nessa pergunta são **Documento (d)**, **Usuario (u)** e **Setor(s)**. O resultado desta consulta mostrará uma lista de documentos assinados por cada usuário,



incluindo o nome do usuário, o nome do documento, o assunto do documento, a data de emissão do documento e o nome do setor ao qual o usuário está associado.

```
SELECT u.Nome AS Nome_do_Usuario,  
       d.Tipo_de_Documento AS Nome_do_Documento,  
       d.Assunto_Principal AS Assunto_do_Documento,  
       d.Data_de_Emissao AS Data_de_Emissao,  
       s.Nome_do_Setor AS Setor_Designado  
FROM Documento d  
INNER JOIN Usuario u ON d.Responsavel_pela_Emissao = u.Nome  
INNER JOIN Setor s ON u.idSetor = s.ID;
```

Após a realização de todas as consultas e testes no PostgreSQL, podemos concluir que as operações de consulta e manipulação de dados foram executadas com sucesso, demonstrando a capacidade do sistema de gerenciamento de banco de dados para recuperar informações de acordo com os relacionamentos entre as tabelas e as condições especificadas nas consultas. Foi possível identificar documentos, usuários e setores envolvidos, bem como rastrear o histórico de movimentação de pastas e documentos. Para maiores detalhes o script total desde a criação das tabelas, como a inserção das chaves estrangeiras e também dos dados e concluindo com as consultas estão no Github nesse endereço: <https://github.com/TadeuAnderson/EPDMD/tree/9c12414ef026d3c500a6d7b8056f0faca19cf3da/Banco%20de%20Dados>