

# Roteiro de Trabalho Prático – Banco de Dados e Aplicação Conectada

## Objetivo Geral

Desenvolver um **sistema completo** composto por:

- um **modelo de dados relacional**, criado pelo grupo,
  - um **banco de dados implementado em SQL**, e
  - uma **aplicação conectada** (desktop, web ou mobile) que permita **consultas e manipulação de dados complexos** diretamente a partir da interface da aplicação.
- 

## 1. Etapas do Trabalho

### Etapa 1 – Definição do Domínio (Tema Livre)

Cada grupo deve escolher um **domínio de aplicação** real ou fictício, por exemplo:

- Sistema de biblioteca
- Gestão de pedidos de um restaurante
- Clínica médica
- Controle de eventos e inscrições
- Plataforma de cursos on-line
- Sistema de streaming de música
- Gestão de estoque e vendas

#### Entrega esperada:

Descrição breve (até 1 página) do contexto do sistema e principais entidades envolvidas.

---

### Etapa 2 – Modelagem Conceitual e Lógica

Crie um **Modelo Entidade-Relacionamento (MER)** do domínio escolhido.

Converta o MER em um **Modelo Relacional**, normalizado até a 3<sup>a</sup> Forma Normal.

Identifique chaves primárias, estrangeiras e restrições de integridade.

#### Ferramentas sugeridas:

draw.io, Lucidchart, MySQL Workbench, DBDesigner, ou similar.

#### Entrega esperada:

Diagrama ER + Esquema relacional (tabelas e relacionamentos).

---

### **Etapa 3 – Implementação do Banco de Dados**

Utilize um **SGBD relacional** (MySQL, PostgreSQL, SQL Server ou SQLite).

Crie o banco de dados e suas tabelas via **script SQL DDL**.

Insira dados de teste suficientes (mínimo: 10 registros por tabela).

#### **Entrega esperada:**

Script SQL completo (create + insert) do banco.

---

### **Etapa 4 – Desenvolvimento da Aplicação Conectada**

Desenvolva uma aplicação (em **Java, Python, PHP, Node.js, C#, etc.**) que se conecte ao banco de dados.

A aplicação deve conter pelo menos:

**Tela de login ou acesso inicial**

**Tela de cadastro ou atualização de dados**

**Tela de consultas personalizadas** (complexas, combinando múltiplas tabelas)

As consultas devem ser ativadas **a partir da interface** (por botões, menus, campos de busca, etc.).

#### **Entrega esperada:**

Código-fonte da aplicação + instruções de execução.

---

### **Etapa 5 – Consultas SQL Complexas**

Implemente **no mínimo 5 consultas SQL complexas**, que devem ser chamadas pela aplicação, de cada um dos tipos abaixo:

1. **Junções múltiplas (JOINS)** entre 3 ou mais tabelas;
2. **Consultas com subconsultas (subqueries)**;
3. **Consultas agregadas** (com GROUP BY, HAVING, AVG, COUNT, etc.);
4. **Consultas com operações de comparação de strings e multiconjunto** (some, any);
5. **Consultas com ordenação e limitação de resultados** (ORDER BY, LIMIT).

#### **Exemplo (ilustrativo):**

```
SELECT u.nome, COUNT(p.id_pedido) AS total_pedidos
FROM usuarios u
JOIN pedidos p ON u.id_usuario = p.id_usuario
WHERE p.data_pedido >= '2025-01-01'
GROUP BY u.nome
HAVING COUNT(p.id_pedido) > 3
```

ORDER BY total\_pedidos DESC;

---

## **Etapa 6 – Entrega e Demonstração**

Cada grupo deve apresentar:

**O modelo conceitual e lógico,**

**O banco de dados funcional,**

**A aplicação rodando com as consultas acionadas via interface.**

### **Forma de entrega:**

Repositório (GitHub, GitLab, etc.) com todo o código e scripts SQL.

Apresentação para o professor de todo o funcionamento e código das consultas

Observação: **Na avaliação serão analisadas a complexidade do banco e se as operações criadas e mostradas são de fato relevantes para o domínio em questão.**