

Projeto Interdisciplinar IV – Implementação de Aplicação Desktop em C# com MySQL

1. Introdução

O Projeto Interdisciplinar III dá continuidade ao trabalho realizado no semestre passado, em que foram levantados requisitos, elaboradas histórias de usuário, diagramas UML e modelos de banco de dados para diferentes problemas reais.

Nesta etapa, o foco será a **implementação prática de uma aplicação desktop em C#, integrada a um SGBDR MySQL**, transformando a modelagem em um sistema funcional que atenda às necessidades específicas de cada projeto.

2. Objetivo Geral

- Implementar uma aplicação desktop em **C#**, utilizando **MySQL** como banco de dados, baseada nos requisitos já modelados.
- Cada grupo desenvolverá um sistema independente, de acordo com seu levantamento anterior, consolidando os conhecimentos de **Banco de Dados II** e **Programação II**.

3. Objetivos Específicos

- Construir o **banco de dados físico** no MySQL a partir do DER modelado no semestre passado.
- Implementar **procedures e triggers** para automatizar regras de negócio.
- Desenvolver a **aplicação desktop em C# (Windows Forms ou WPF)**.
- Implementar as **funcionalidades principais** de cada sistema, com operações de cadastro, consulta, atualização e exclusão.
- Criar **relatórios gerenciais** de apoio à tomada de decisão.
- Validar o sistema com dados de teste e elaborar relatório técnico final.

4. Etapas do Projeto

4.1 Revisão dos Requisitos

Cada grupo deverá retomar os requisitos levantados no semestre passado, ajustando se necessário:

- **Biblioteca Escolar (BiblioKopke)**
 - ✓ Cadastro de livros, reservas e empréstimos.
 - ✓ Integração futura com o SIMADE.
 - ✓ Relatórios de empréstimos e acervo.
- **Clínica Veterinária São Camilo**
 - ✓ Cadastro de tutores, animais, veterinários e produtos.
 - ✓ Agendamento de consultas e serviços.
 - ✓ Controle de estoque de medicamentos.
 - ✓ Prontuários médicos.
 - ✓ Relatórios gerenciais.
- **Buffet Encanto**
 - ✓ Gestão de eventos e contratos.
 - ✓ Cadastro de clientes, funcionários e produtos.
 - ✓ Movimentações de caixa e relatórios financeiros.
 - ✓ Controle de estoque.
- **Hamburgueria EmBrasa**
 - ✓ Cadastro de cardápio e insumos.
 - ✓ Registro de vendas.
 - ✓ Controle de estoque com baixa automática.
 - ✓ Relatórios de vendas e consumo.
- **Marketing e Publicidade (Duda Costa)**
 - ✓ Gestão de clientes.
 - ✓ Controle de postagens (Instagram, TikTok, etc.).
 - ✓ Organização de eventos promocionais.
 - ✓ Gestão financeira vinculada a clientes e campanhas.

4.2 Banco de Dados (MySQL)

- Criar o banco físico com base no DER.

- Implementar **DDL (CREATE TABLE)** para todas as entidades.
- Inserir dados de teste com **DML (INSERT)**.
- Criar **Procedures e Triggers** específicas:
 - ✓ Biblioteca: atualização automática de disponibilidade de livros após empréstimo.
 - ✓ Veterinária: baixa de medicamentos após consulta.
 - ✓ Buffet: registro financeiro automático após contrato.
 - ✓ Hamburgueria: baixa de insumos após venda.
 - ✓ Marketing: vinculação obrigatória entre postagem e cliente.

4.3 Desenvolvimento Back-End (C#)

- Implementação orientada a objetos com classes correspondentes às entidades do banco.
- Métodos de negócio para cada caso de uso.
- Integração com o banco via **ADO.NET ou Entity Framework**.

4.4 Desenvolvimento Front-End (C# – Desktop)

- Aplicação desktop em **Windows Forms ou WPF**.
- Telas obrigatórias:
 - ✓ Login e autenticação.
 - ✓ Cadastro de entidades principais.
 - ✓ Movimentações específicas (emprestimos, consultas, vendas, agendamentos, postagens).
 - ✓ Relatórios de apoio à gestão.

4.5 Testes e Validação

- Execução dos casos de uso definidos.
- Testes de integração entre C# e MySQL.
- Documentação das evidências em relatório final.

5. Entregáveis

1. **Script SQL completo (MySQL)** com tabelas, procedures, triggers e dados de teste.
2. **Código-fonte da aplicação desktop em C#.**
3. **Documentação revisada** (requisitos, DER, UML, regras de negócio implementadas).
4. **Relatório de testes** com evidências.
5. **Apresentação final com demonstração funcional.**

6. Considerações Finais

Este projeto consolida a passagem da **modelagem conceitual para a implementação prática**. Cada grupo terá um sistema com identidade própria, refletindo o problema real levantado no semestre passado.

A aplicação em **C# com banco de dados MySQL** garante o aprendizado prático de programação orientada a objetos, integração com banco de dados e desenvolvimento de software completo, do levantamento até a entrega funcional.

Entregáveis (C# Desktop + MySQL), por quinzena

22/set (D0) — Kickoff + Base do Projeto

- **Entrega:**
 - ✓ Documento revisado de requisitos do grupo (domínio específico: biblioteca, clínica, buffet, hamburgueria, marketing etc.).
 - ✓ DER final e dicionário de dados.
 - ✓ Repositório criado (estrutura de solução C# — WinForms ou WPF) + guia de build.
- **Critérios de Aceitação:**
 - ✓ Requisitos priorizados (MoSCoW) e casos de uso principais confirmados.

- ✓ DER consistente (chaves PK/FK, cardinalidades).
- ✓ Projeto compila “Hello World” com camadas vazias (UI / Domain / Data).
- **Evidências:** PDF dos requisitos/DER + link do repositório + print da compilação.

07/out (D15) — Modelo Físico + DDL + Conexão C#↔MySQL

- **Entrega:**
 - ✓ Script DDL (MySQL) completo (CREATE DATABASE/TABLE/INDEX/constraints).
 - ✓ Semente inicial de dados (DML básica) para testes.
 - ✓ Camada de acesso a dados (ADO.NET ou EF) configurada e testada.
- **Critérios de Aceitação:**
 - ✓ Banco sobe “limpo” do zero (DROP/CREATE) e popula dados de exemplo.
 - ✓ App C# conecta no banco e executa pelo menos 1 SELECT real.
- **Evidências:** .sql versionado + vídeo curto mostrando app listar dados.

22/out (D30) — Regras de Negócio no BD + CRUDs Essenciais

- **Entrega:**
 - ✓ Triggers/Procedures específicas do domínio (p.ex., baixa de estoque, disponibilidade de livro, lançamento financeiro).
 - ✓ CRUDs funcionais das entidades centrais (p.ex., Clientes/Produtos/Livros/Eventos).
 - ✓ Testes de integração cobrindo regras (happy path + 1 caso de erro).
- **Critérios de Aceitação:**
 - ✓ Trigger/Procedure executa automaticamente a regra crítica do domínio.
 - ✓ CRUDs com validação mínima (campos obrigatórios, formatos).
- **Evidências:** scripts SQL, prints do app executando as ações, relatório de testes.

06/nov (D45) — Fluxos Operacionais + Autenticação/Perfis

- **Entrega:**

- ✓ Implementação dos **fluxos principais** (ex.: empréstimo/devolução; venda; agendamento/consulta; postagem/campanha).
- ✓ **Login e controle de acesso por perfil** (admin/operador/gerência).
- ✓ Logs (mínimo: quem fez o quê e quando) e tratamento básico de exceções.
- **Critérios de Aceitação:**
 - ✓ Fluxo ponta-a-ponta executável com dados reais do grupo.
 - ✓ Perfis restringem telas/ações conforme regras.
- **Evidências:** vídeo curto de navegação completa + tabela de perfis/permite.

21/nov (D60) — Relatórios + UX de uso diário + Qualidade

- **Entrega:**
 - ✓ **Relatórios** (mín. 2) relevantes ao domínio (ex.: estoque crítico, vendas por período, fluxo financeiro, empréstimos por aluno, desempenho de campanha).
 - ✓ Melhorias de **UX** (navegação, feedback visual, estados de erro, atalhos).
 - ✓ **Testes:** cobertura de cenários críticos e checklist de revisão (acessibilidade básica e consistência visual).
- **Critérios de Aceitação:**
 - ✓ Relatórios exportáveis (PDF/CSV ou impressão) e filtros funcionais.
 - ✓ Checklist de UX atendido (mensagens claras, campos com máscara/placeholder, foco/teclas rápidas onde couber).
- **Evidências:** PDFs/CSVs gerados, prints comparativos “antes/depois”, relatório de testes.

30/nov (Encerramento) — Release Candidate + Documentação Final + Apresentação

- **Entrega:**
 - ✓ **Release Candidate (RC)**: executável instalável/portável + scripts SQL finais.
 - ✓ **Manual do Usuário** (telas e fluxos) + **Manual Técnico** (arquitetura, camadas, conexão, scripts, como subir o ambiente).

- ✓ **Relatório Final** (objetivos, decisões, limitações, próximos passos) e **slides de apresentação** com demo guiada.
- **Critérios de Aceitação:**
 - ✓ App inicia do zero (instalação simples), conecta ao MySQL e executa os fluxos-chave.
 - ✓ Documentos completos, coerentes com o que foi implementado.
- **Evidências:** pacote RC, PDFs dos manuais, relatório final, slides e vídeo de 3–5 min de demonstração.