

# Relatório do Projeto - Gerenciamento de Biblioteca

Gabriel da Silva Reboli

December 1, 2024

## 1 Introdução

O objetivo deste projeto é consolidar o aprendizado da integração com bancos de dados usando C# e Windows Forms. O projeto implementa um sistema de gerenciamento de biblioteca, permitindo a criação, leitura, atualização e exclusão de registros relacionados a livros, membros e empréstimos. Além disso, o padrão de projeto Singleton foi utilizado para gerenciar a conexão com o banco de dados de forma eficiente.

## 2 Modelo de Dados

O modelo de dados é composto por três tabelas principais:

- **Livros:** Contém informações sobre os livros disponíveis na biblioteca.
- **Membros:** Representa os usuários cadastrados.
- **Empréstimos:** Registra os empréstimos realizados, associando membros a livros.

### Diagrama de Relacionamento

Livros (1) --- (N) Empréstimos (N) --- (1) Membros

### Estrutura das Tabelas

#### Tabela Livros

- LivroID (INT, PK, AUTO\_INCREMENT)
- Titulo (VARCHAR)
- Autor (VARCHAR)
- DataPublicacao (DATE)
- Categoria (VARCHAR)

### **Tabela Membros**

- MembroID (INT, PK, AUTO\_INCREMENT)
- Nome (VARCHAR)
- Email (VARCHAR)
- Telefone (VARCHAR)

### **Tabela Empréstimos**

- EmprestimoID (INT, PK, AUTO\_INCREMENT)
- LivroID (INT, FK para Livros)
- MembroID (INT, FK para Membros)
- DataEmprestimo (DATE)
- DataDevolucao (DATE)

## **3 Funcionalidades da Aplicação**

A aplicação foi desenvolvida em C# com Windows Forms e implementa as seguintes funcionalidades para cada tabela:

### **CRUD (Criar, Ler, Atualizar e Excluir)**

- **Livros:** Adicionar, visualizar, editar e excluir livros.
- **Membros:** Adicionar, visualizar, editar e excluir membros.
- **Empréstimos:** Registrar novos empréstimos e gerenciá-los.

### **Validações de Entrada**

- Campos obrigatórios não podem ser deixados em branco.
- Mensagens de erro são exibidas para entradas inválidas.

### **Interface Amigável**

- Uso de `DataGridView` para exibir registros.
- Botões para cada funcionalidade de CRUD.
- Campos organizados para facilitar a entrada de dados.

## 4 Uso do Padrão Singleton

O padrão Singleton foi utilizado para gerenciar a conexão com o banco de dados. A classe `DatabaseConnection` garante que apenas uma instância de conexão seja criada, reduzindo a sobrecarga e melhorando a eficiência.

### Implementação

```
public static class DatabaseConnection
{
    private static MySqlConnection _connection;
    private static readonly string _connectionString =
        "Server=localhost;Database=Biblioteca;User ID=root;Password=123456;";

    public static MySqlConnection GetConnection()
    {
        if (_connection == null || _connection.State == System.Data.ConnectionState.C
        {
            _connection = new MySqlConnection(_connectionString);
        }
        return _connection;
    }
}
```

## 5 Conclusão

Este projeto permitiu o aprendizado prático sobre integração com bancos de dados, desenvolvimento de interfaces gráficas e aplicação de padrões de projeto. As funcionalidades foram implementadas de forma robusta e amigável, atendendo aos requisitos estabelecidos.

### Melhorias Futuras

- Adicionar autenticação para diferentes tipos de usuários.
- Incluir relatórios analíticos sobre os empréstimos realizados.
- Melhorar o design da interface utilizando frameworks modernos.

## 6 Anexos

1. Código-fonte completo do projeto.
2. Script SQL para criação do banco de dados.