

Universidade Federal de Minas Gerais

Bruno Carneiro Mazzeiro  
Daniel Lopes Andrade  
Davi Braga Padilha Azevedo  
João Duque da Silva Filho  
Júlia Cesário Camelo  
Lauro Emmanuel Assunção Rafael

# **TRABALHO PRÁTICO DE PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II**

Belo Horizonte - MG  
2024

Universidade Federal de Minas Gerais

## ÍNDICE

1. APRESENTAÇÃO .....	3
2. MODELAGEM .....	3
2.1. USER STORY .....	3
2.2. CARTÕES CRC .....	4
3. DOCUMENTAÇÃO .....	6
3.1. README .....	6
4. PADRONIZAÇÃO DE NOMES.....	8
5. CONSIDERAÇÕES FINAIS .....	8

# 1. APRESENTAÇÃO

O trabalho foi desenvolvido com o intuito de apresentar uma solução prática para um Sistema de Biblioteca Avançado. É importante ressaltar que, está nas competências do time da programação, o uso da linguagem C++11 bem como das técnicas de: Boas Práticas de Programação, Programação Orientada a Objeto, Abstração, Encapsulamento, Herança, Polimorfismo e Modularização.

## 2. MODELAGEM

A modelagem do trabalho é muito importante para que as tarefas sejam realizadas de forma eficiente e que o trabalho atenda exatamente ao que foi solicitado. Por meio dessa modelagem, pode-se alcançar uma participação mais ampla de todos os envolvidos no projeto.

A estratégia utilizada de modelagem é a “User Story”, que trata de uma boa prática de documentação que visa descrever as funcionalidades e objetivos de uma solução que está sendo criada. A “User Story” tem enfoque na comunicação e prioriza atender as necessidades e dores do cliente.

### 2.1. USER STORY

**Operação:** Sistema de Biblioteca Avançado

**Descrição:** Como um assíduo leitor, desejo encontrar uma biblioteca que me dê um feedback constante sobre chegada de novos livros, status de um empréstimo e

gerenciamentos de multas. Assim, geraria mais comodidade e melhoraria minha comunicação com a biblioteca como leitor.

**Critérios de Aceitação:**

- Posso olhar o status de empréstimo de cada livro;
- Devo receber notificação com a chegada de novos livros;
- Posso olhar o acervo de livros disponíveis na biblioteca;
- Posso gerar boletos para o pagamento de multas;
- Posso emprestar meus livros à biblioteca;
- Posso agendar o empréstimo de um livro.

## **2.2. CARTÕES CRC**

Os cartões ou modelagem CRC é uma técnica de orientação a objetos que visa determinar as necessidades do usuário. Nesta técnica, os cartões são divididos em três áreas: classe, responsabilidades e colaboração.

Em classe, é definido um nome. Escolhido, ele deve transmitir clara e sucintamente o que cada tipo de objeto representa no mundo real.

Em responsabilidades, é definido os papéis que cada objeto desempenha no modelo. Nesta abordagem, cada objeto é visto como uma entidade ativa dentro do modelo, executando responsabilidades específicas de modo a cumprir seu papel.

Em colaborações, é definido os nomes dos colaboradores – os objetos que são chamados para os serviços – na mesma linha das responsabilidades que eles apoiam.

Abaixo, encontra-se a modelagem do projeto com base nos cartões CRC.

<b>Classe: Cliente</b>	
<b>Responsabilidades</b>	<b>Colaborações</b>
Saber identificador, nome, endereço, telefone, e-mail; Pagar multa; Reservar livro; Devolver livro; Receber notificação; Saber status de empréstimo; Fazer login; Localizar livro.	Livro; Biblioteca.

<b>Classe: Livro</b>	
<b>Responsabilidades</b>	<b>Colaborações</b>
Saber título, autor, ano de publicação, gênero, condição; Mostrar o status; Mostrar localização	Biblioteca; Cliente.

<b>Classe: Biblioteca</b>	
<b>Responsabilidades</b>	<b>Colaborações</b>
Localizar livro; Aplicar multa; Editar propriedades dos livros; Enviar notificações; Exibir o acervo; Cadastrar livro; Receber livro; Cadastrar clientes; Receber pagamento; Fazer cobranças.	Cliente; Livro.

### 3. DOCUMENTAÇÃO

A documentação é imprescindível para o desenvolvimento e manutenção de um software. Com ela é possível manter um registro claro do projeto e garantir que outros desenvolvedores possam compreender e contribuir com o código, seja na fase de desenvolvimento ou de manutenção. Portanto, dedicar tempo à criação e manutenção de uma documentação abrangente é uma prática muito importante para o desenvolvimento de um software de qualidade.

Para o proposto trabalho, foi elaborado um README, que trata de uma ferramenta de comunicação entre a equipe de desenvolvimento do software. O objetivo principal é documentar todos os requisitos do projeto e alguns outros pontos relevantes.

#### 3.1. README

**Apresentação do Problema:** o Sistema de Biblioteca Avançado propõe-se a resolver os desafios enfrentados pelas bibliotecas no gerenciamento eficiente de seus recursos, incluindo o controle de estoque de livros, o acompanhamento de empréstimos e devoluções, notificações de devolução atrasada, notícias sobre a chegada de novos livros e o gerenciamento de multas. A melhoria nesses processos se tornará um diferencial de mercado, aumentando a comodidade dos leitores e aproximando cliente e estabelecimento comercial.

**Visão Geral da Solução:**

A solução proposta é um programa em C++11 que prestaria uma interface para gerenciar uma biblioteca. Este programa ofereceria as seguintes funcionalidades:

- Adicionar e remover livros do acervo;
- Cadastrar e remover usuários;
- Realizar empréstimos e devoluções de livros;

- Aplicar e cobrar multas;
- Gerar boletos;
- Notificar clientes sobre a chegada de novos livros;
- Receber empréstimos de usuários;
- Agendar empréstimos.

### **Estrutura do Projeto:**

- Sistema.cpp: contém os arquivos-fonte do programa em C++11;
- Sistema.hpp: possui os cabeçalhos das classes utilizadas no programa;
- main.cpp: diretório onde são gerados os arquivos executáveis.
- TP.pdf: este arquivo, que fornece uma visão geral do projeto e instruções de uso.

### **Instruções de Instalação:**

- Certifique-se de ter um compilador C++11 instalado em seu sistema, como o GCC ou o Visual Studio;
- Baixe os arquivos do projeto em seu ambiente de desenvolvimento;
- Compile os arquivos-fonte utilizando o compilador C++11.

### **Instruções de Uso:**

- Execute o arquivo executável;
- Utilize o menu interativo para realizar as operações desejadas como: adicionar livros, cadastrar usuários, emprestar e devolver livros, etc;
- Siga as instruções fornecidas pelo programa para interagir com ele.

### **Principais Dificuldades:**

- Implementação eficiente das notificações de devolução atrasada e gerenciamento de multas;
- Lidar com casos complexos de interações entre usuários e livros em um ambiente competitivo;
- Garantir a robustez do sistema e a correta manipulação de erros e exceções.

## 4. PADRONIZAÇÃO DE NOMES

As nomenclaturas do projeto segue o seguinte padrão:

- **Nomes de Pastas / Arquivos:** UpperCamelCase
- **Branches:** UpperCamelCase → NomeDaTarefa
- **Variáveis:** lowerCamelCase
- **Funções:** lowerCamelCase
- **Commits:** a intenção do commit deve ser esclarecida no início da mensagem, seguindo o padrão abaixo, sendo diferente no caso de adições ou correções: i) feat: breve explicação da adição; ii) fix: breve explicação da correção.

## 5. CONSIDERAÇÕES FINAIS

Este documento fornece uma visão geral do projeto de implementação de um Sistema de Biblioteca Avançado. Ademais, inclui: sua estratégia de modelagem e documentação, seu propósito, estrutura, instruções de instalação e uso, bem como as principais dificuldades encontradas durante o desenvolvimento.



Bruno Carneiro Mazzeiro  
Daniel Lopes Andrade  
Davi Braga Padilha Azevedo  
João Duque da Silva Filho  
Júlia Cesário Camelo  
Lauro Emmanuel Assunção Rafael

## **TRABALHO PRÁTICO DE PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II**

Trabalho apresentado para a avaliação  
da disciplina de Programação e  
Desenvolvimento de Software II.  
Universidade Federal de Minas Gerais  
(UFMG).

**Professor: Evellyn Soares Cavalcante**

Belo Horizonte - MG  
2024