

**Relatório de Monitoramento e Manutenção de Sistema**

***Implementação de Melhorias***

Biblioteca Municipal do Bairro B

**Autor: Jacir Gaio**



Sumário

[Introdução 3](#_Toc143517934)

[Metodologia para monitoramento e manutenção do sistema 4](#_Toc143517935)

[Metodologia de implementação das melhorias e manutenção do sistema 4](#_Toc143517936)

[Apontamentos e melhorias 5](#_Toc143517937)

[**Número 0000001 - Pagina Cadastro de Livro** 5](#_Toc143517938)

[**Número 0000002 - Campos obrigatórios - Página Cadastro de Livros** 6](#_Toc143517939)

[**Número 0000003 - Mensagem de Nenhum registro encontrado** 7](#_Toc143517940)

[**Número 0000004 - Link para página de listagem de livros** 8](#_Toc143517941)

[**Número 0000005 - Campos obrigatórios - empréstimo de livros** 9](#_Toc143517942)

[**Número 0000006 - Listagem livros ordenados por data** 10](#_Toc143517943)

[**Número 0000007 Filtro de empréstimos de livros** 11](#_Toc143517944)

[**Número 0000008 - Tabela de empréstimo não formatada** 12](#_Toc143517945)

[**Número 0000009 - Campo de telefone editável** 13](#_Toc143517946)

[**Número 0000011 - Link para usuário deslogar** 15](#_Toc143517947)

[**Número 0000012 - Ênfase em empréstimos atrasados** 16](#_Toc143517948)

[**Número 0000013 – Paginação da lista de empréstimos de livros** 17](#_Toc143517949)

[**Número 0000014 - Cadastramento de usuários** 18](#_Toc143517950)

[Versionamento do projeto com Git 19](#_Toc143517951)

[GitHub como Repositório: 19](#_Toc143517952)

[Evidencias do repositório GitHub: 20](#_Toc143517953)

# Introdução

O projeto para a biblioteca municipal do bairro B que tem objetivo de gerenciar empréstimos de livros por meio de uma plataforma web e está atualmente na fase beta.

O foco deste relatório está na resolução de questões críticas, aprimoramento de funcionalidades e garantia da usabilidade. Com enfoque em resolução de problemas e otimizações, abrange requisitos, testes, avaliações de desempenho, feedback dos usuários e estratégias para enfrentar desafios.

# Metodologia para monitoramento e manutenção do sistema

A metodologia empregada para o monitoramento e manutenção do sistema, utilizando a ferramenta Mantis BT, envolve uma abordagem estruturada para rastrear e gerenciar problemas, erros e melhorias no sistema.

Inicialmente foi realizado a identificação e categorização de problemas, sejam eles erros, bugs ou oportunidades de melhoria. Cada problema é registrado na ferramenta Mantis BT, onde são fornecidas informações essenciais, como descrição do problema, contexto, passos para reproduzir e detalhes adicionais.

A ferramenta Mantis BT também permite o rastreamento de progresso, com atualizações regulares sobre o status de cada problema. À medida que as soluções são implementadas, são conduzidos testes para validar as correções e melhorias. Caso o problema seja confirmado como resolvido, ele é fechado na ferramenta, acompanhado de notas sobre a solução adotada. Além disso, também oferece a capacidade de gerar relatórios sobre os problemas identificados, status das correções, métricas de resolução e áreas de melhoria contínua.

# Metodologia de implementação das melhorias e manutenção do sistema

Após a identificação de problemas ou oportunidades de aprimoramento no sistema por meio da ferramenta Mantis BT, o processo de implementação das melhorias e correções foi conduzido em estrita conformidade com as especificações definidas no processo anterior. Para isso, foi adotado um desenvolvimento controlado e versionado utilizando a ferramenta GIT, e sincronização por meio do repositório na plataforma GitHub.

Essa metodologia tem como principal objetivo assegurar que as melhorias e a manutenção ocorram de maneira altamente organizada, com a finalidade de minimizar riscos e otimizar o valor proporcionado aos usuários e às operações do sistema.

# Apontamentos e melhorias

## **Número 0000001 - Pagina Cadastro de Livro**

**Descrição:**

Na página de cadastro de livro, há um erro ortográfico no campo "Ator" onde deveria ser "Autor".

**Solução:**

Corrigido o erro no código HTML da View /Livro/Cadastro.cshtml.

## **Número 0000002 - Campos obrigatórios - Página Cadastro de Livros**

**Descrição:**

No formulário da página de cadastro de livros, todos os campos não estão como obrigatórios, conforme especificação nos requisitos do projeto.

**Solução:**

Foram efetuadas as adaptações necessárias para tornar os campos obrigatórios, acrescentando o atributo "required" aos campos na View \_FormCadastro.cshtml.

## **Número 0000003 - Mensagem de Nenhum registro encontrado**

**Descrição:**

Na página de listagem de livros, quando a pesquisa não trouxer nenhum elemento, não está aparecendo na página a mensagem “Nenhum registro encontrado”.

**Solução:**

Foi aplicada uma lógica por meio de uma estrutura condicional "if" na view Livro/Listagem.cshtml, de forma a exibir a mensagem "Nenhum registro encontrado" na página quando a pesquisa não retornar nenhum resultado.

## **Número 0000004 - Link para página de listagem de livros**

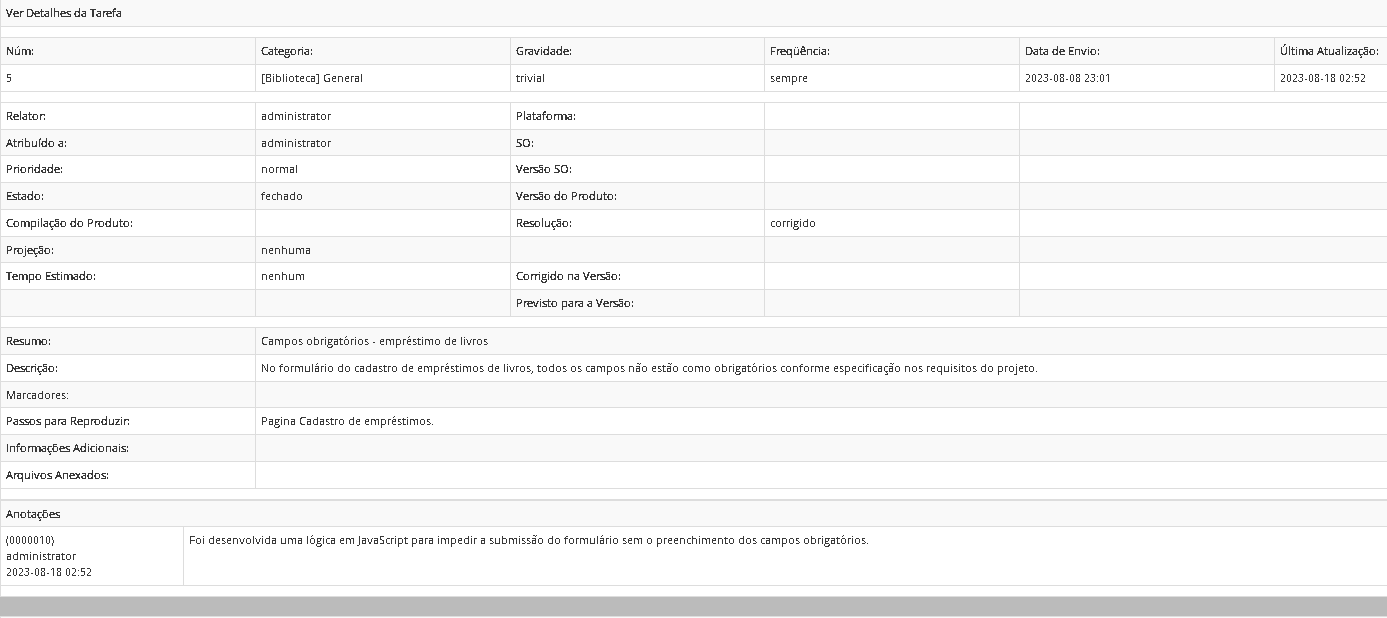
**Descrição:**

No menu, não há um link de atalho para a página de listagem de livros.  
A tabela somente é possível ser acessada após o cadastro de um livro ou se digitar o endereço diretamente no navegador.

**Solução:**

Implementei uma atualização no menu, criando um elemento dropdown que exibe dois links quando expandido: "Cadastro" e "Listagem". Isso permite aos usuários acessarem facilmente essas duas funcionalidades do sistema de maneira organizada e intuitiva.

## **Número 0000005 - Campos obrigatórios - empréstimo de livros**



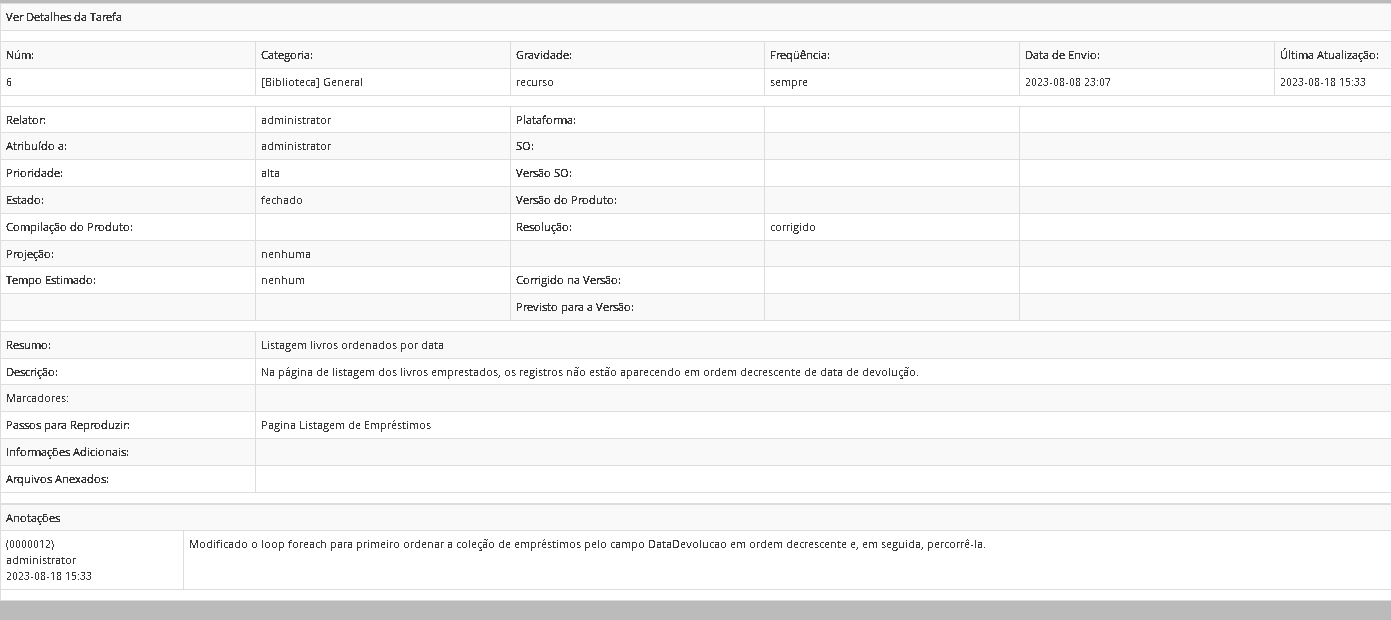
**Descrição:**

No formulário do cadastro de empréstimos de livros, todos os campos não estão como obrigatórios conforme especificação nos requisitos do projeto.

**Solução:**

Foi desenvolvida uma lógica em JavaScript para impedir a submissão do formulário sem o preenchimento dos campos obrigatórios.

## **Número 0000006 - Listagem livros ordenados por data**



**Descrição:**

Na página de listagem dos livros emprestados, os registros não estão aparecendo em ordem decrescente de data de devolução.

**Solução:**

Modificado o loop foreach para primeiro ordenar a coleção de empréstimos pelo campo DataDevolucao em ordem decrescente e, em seguida, percorrê-la.

## **Número 0000007 Filtro de empréstimos de livros**

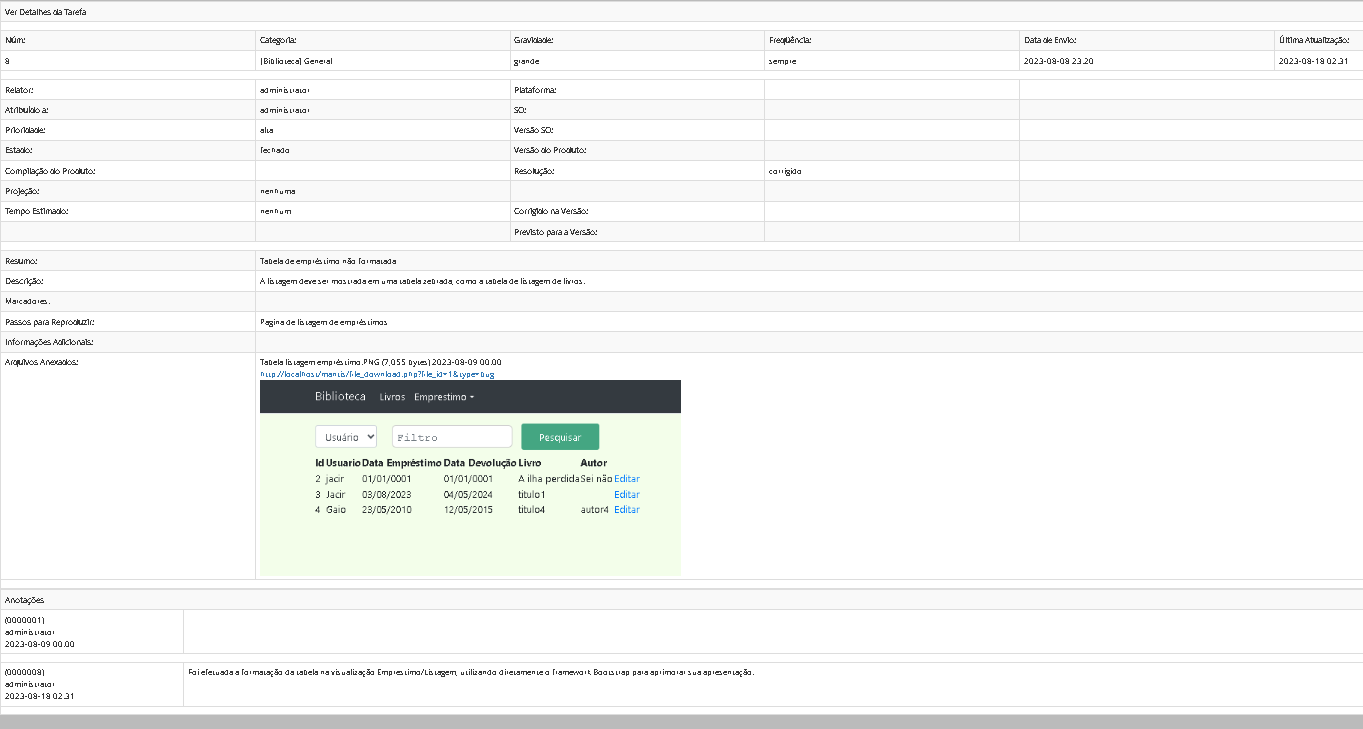
**Descrição:**

O filtro de empréstimo por usuário e título não está funcional. Os filtros devem considerar texto inteiro (título completo de um livro, por exemplo) ou textos parciais (parte do título de um livro, por exemplo), tanto para título quanto para autor.

**Solução:**

Realizado as adequações em EmprestimoController na action "Listagem";  
E em "EmprestimoService no método "ListarTodos"

## **Número 0000008 - Tabela de empréstimo não formatada**



**Descrição:**

A listagem deve ser mostrada em uma tabela zebrada, como a tabela de listagem de livros.

**Solução:**

Foi efetuada a formatação da tabela na visualização Emprestimo/Listagem, utilizando diretamente o framework Bootstrap para aprimorar sua apresentação.

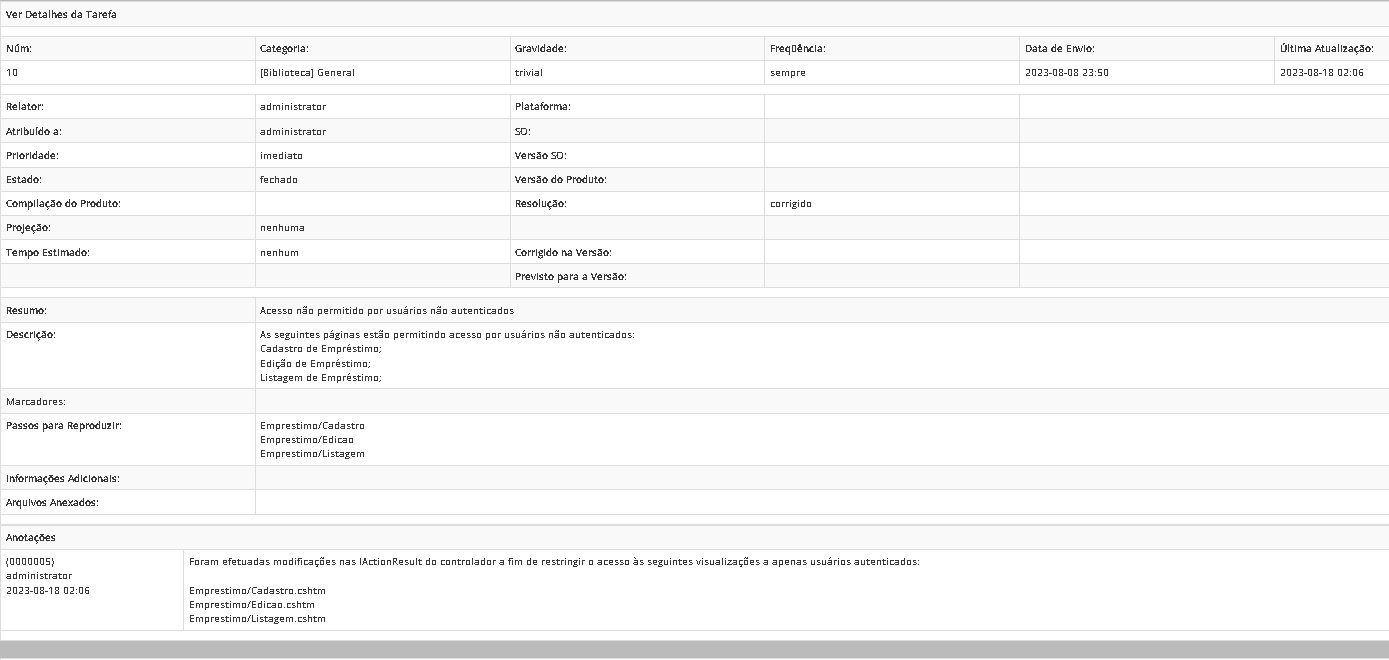
## **Número 0000009 - Campo de telefone editável**

**Descrição:**

A atualização do empréstimo deve permitir apenas a edição do campo de livro e dos campos de data, porém está permitindo a edição do campo telefone. Deve ser corrigido.

**Solução:**

Realizada uma modificação na visualização Emprestimo/Edicao.cshtml, restringindo a edição exclusivamente aos campos de livro e de data.

A edição foi desabilitada incluindo o atributo readonly="readonly" no input do campo telefone, garantindo que não seja possível realizar alterações nesse campo.**Número 0000010 - Acesso permitido por usuários não autenticados**

**Descrição:**

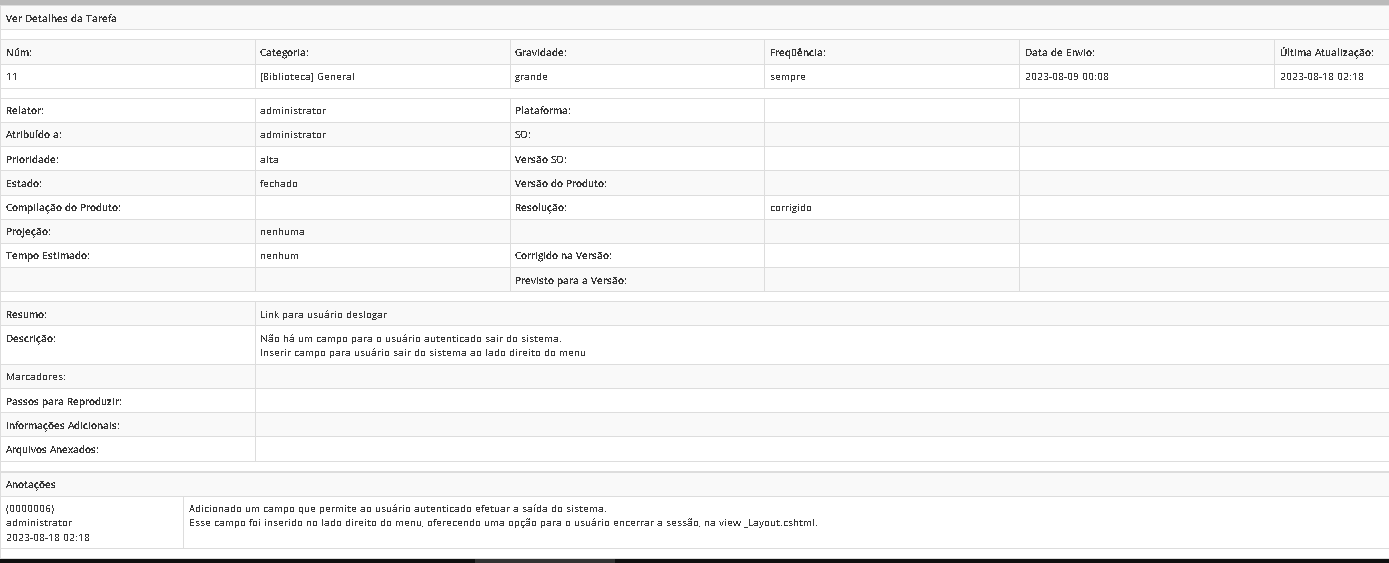
As seguintes páginas estão permitindo acesso por usuários não autenticados:  
 Cadastro de Empréstimo;  
 Edição de Empréstimo;  
 Listagem de Empréstimo.

**Solução:**

Foram efetuadas modificações nas IActionResult do controlador a fim de restringir o acesso às seguintes visualizações a apenas usuários autenticados:

Emprestimo/Cadastro.cshtm  
Emprestimo/Edicao.cshtm  
Emprestimo/Listagem.cshtm

## **Número 0000011 - Link para usuário deslogar**



**Descrição:**

Não há um campo para o usuário autenticado sair do sistema.  
Inserir campo para usuário sair do sistema ao lado direito do menu.

**Solução:**

Adicionado um campo que permite ao usuário autenticado efetuar a saída do sistema.

O campo foi inserido no lado direito do menu, oferecendo uma opção para o usuário encerrar a sessão.

## **Número 0000012 - Ênfase em empréstimos atrasados**

**Descrição:**

• Na listagem de empréstimos, deve-se incluir uma estilização que enfatize as linhas com registro de empréstimo atrasado.  
• Sugere-se a aplicação de classes de ênfase do Boostrap.

**Solução:**

Adicionado uma classe condicional table-danger à linha da tabela onde o empréstimo está atrasado, destacando visualmente essas entradas.  
Além disso, é verificado se a data de devolução é menor que a data atual para aplicar essa classe. Isso cria um destaque visível nas linhas dos empréstimos atrasados.

## **Número 0000013 – Paginação da lista de empréstimos de livros**

**Descrição:**

É preciso incluir funcionalidade de paginação nas listagens de empréstimo e de livros; mostrar dez livros por página; e utilizar Boostrap para a estilização.

**Solução:**

Realizado as adequações para incluir paginação na listagem dos livros e empréstimos.

Alterado as ações "Listagem" nos respectivos Controller;

Realizado as alterações nas respectivas Views "Listagem";

Utilizado estilização Bootstrap para melhorar a visualização.

## **Número 0000014 - Cadastramento de usuários**

**Descrição:**

- Na versão atual, existe apenas um usuário, cujo log in está definido diretamente via código.

- É necessário implementar uma estrutura de banco de dados que possibilite o cadastro de usuários.

- Também é preciso criar páginas de inclusão, edição, listagem e exclusão de usuário. Apenas o usuário de log in “admin” deve ter acesso a essas funcionalidades.

- A funcionalidade de log in deve ser alterada para que, em vez de usar dados fixos, use informações do banco de dados para validar o usuário.

- Preferencialmente, deve-se utilizar encriptação MD5 para armazenamento das senhas.

**Solução:**

- Implementado uma estrutura de banco de dados que possibilite o cadastro de usuários em níveis de acesso diferenciados (ADMIN e PADRÂO)

- Alterado a funcionalidade de log in para utilizar as informações do banco de dados para validar o usuário.

- Criado as páginas de inclusão, edição, listagem e exclusão de usuário. Implementado lógica para permitir que apenas o usuário de log in “ADMIN” tenha acesso a essas funcionalidades.

- Implementado encriptação MD5 para armazenamento das senhas.

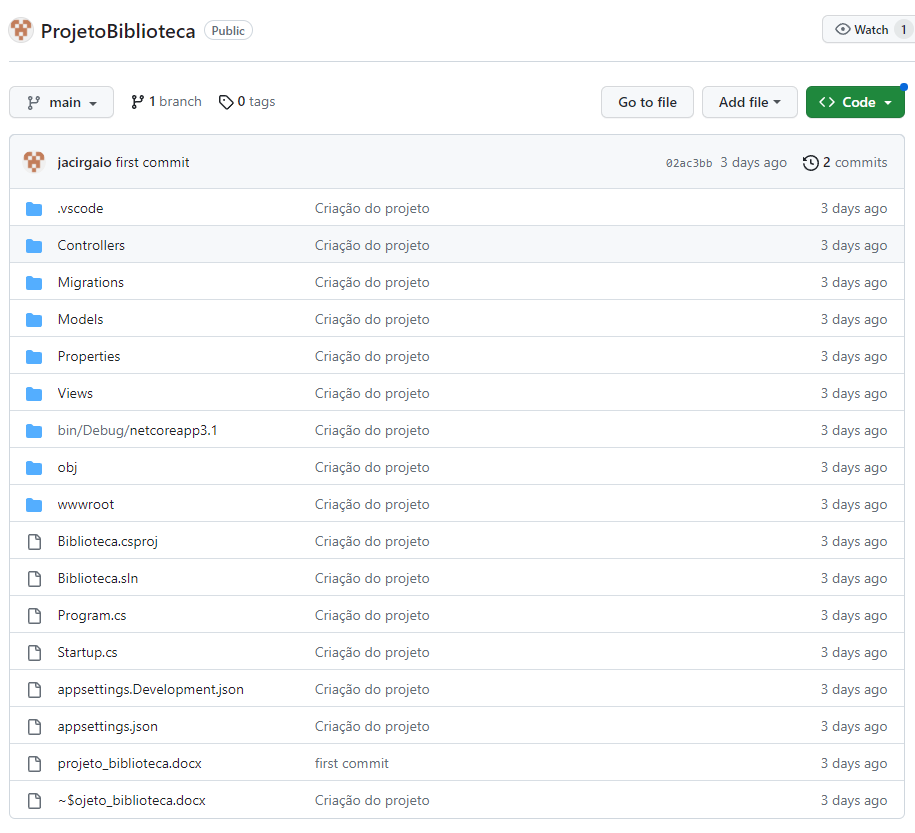
# Versionamento do projeto com Git

Para versionamento do projeto foi utilizado o Git, um sistema de controle de versões que tem a capacidade de rastrear alterações no código-fonte de maneira eficiente. Através do conceito de commits, as alterações foram documentadas, possibilitando a reversão de mudanças indesejadas ou a identificação precisa de onde ocorreram problemas.

# GitHub como Repositório:

O GitHub foi adotado como plataforma de hospedagem e colaboração para o projeto. Além de armazenar o código-fonte, ele também proporciona recursos para revisão de código, gerenciamento de problemas e integração contínua. As funcionalidades de controle de acesso permitem que diferentes membros de uma equipe colaborem de maneira eficiente, mantendo a segurança das informações.

<https://github.com/jacirgaio/ProjetoBiblioteca>

[](https://github.com/jacirgaio/ProjetoBiblioteca)

## Evidencias do repositório GitHub: