

# Sistema de Gerenciamento de Biblioteca

## 1. Introdução

### 1.1. Objetivo

Organizar a gestão de livros e usuários por meio de uma plataforma online, permitindo o cadastro, consulta, empréstimo e realizar devolução de obras. Facilitar o controle do acervo para os administradores e oferecer aos clientes acesso rápido e prático às informações sobre a disponibilidade de livros a partir de desktop e dispositivos móveis conectados à internet.

### 1.2. Escopo

O sistema abrangerá as funcionalidades essenciais para o gerenciamento de uma biblioteca, incluindo:

- **Cadastro e manutenção de acervo:** inserir, atualizar, excluir e consultar livros.
- **Gestão de usuários:** cadastro de clientes e funcionários.
- **Processos de empréstimo e devolução:** registrar retiradas e devoluções de livros, controlando prazos e aplicando multas em caso de atraso ou perda.
- **Consulta ao catálogo:** permitir que clientes pesquisem livros disponíveis por título, autor ou categoria.
- **Controle de acesso:** diferenciação de acesso entre administradores, funcionários e clientes.

#### Fora do escopo

- Suporte a e-books ou acervo digital.
- Reserva antecipada online de livros.

### 1.3. Visão Geral

A especificação está organizada da seguinte forma:

- Especificação de Requisitos – descreve as funcionalidades do sistema e os papéis de usuário.
- Diagrama de Casos de Uso – apresenta as interações entre os usuários e o sistema.
- Diagrama de Classes – detalha as principais classes, atributos, métodos e relacionamentos.

- Descrição dos Métodos – explica a finalidade dos principais métodos que serão implementados no sistema.
- Plano de Testes – define os cenários de teste para validar as funcionalidades.
- Fluxograma ou Diagrama de Sequência – ilustra os fluxos de interação nos processos mais relevantes, como cadastro de usuários e empréstimos.
- Fluxos de Telas - ilustra o fluxo de telas para as diferentes funcionalidades do sistema.
- Esquema Relacional - ilustra como as entidades se relacionam no banco de dados.

## 2. Especificação de Requisitos

### Funcionalidades Principais

- **Cadastro de livros:** inclusão, alteração, exclusão e listagem de livros.
- **Consulta de livros:** busca por título, autor ou categoria.
- **Cadastro de usuários:** clientes podem se cadastrar na plataforma e funcionários são inseridos no sistema pelo administrador.
- **Empréstimo de livros:** registro do empréstimo, data de retirada e data de devolução prevista.
- **Devolução de livros:** atualização do status do livro e aplicação e cálculo de multa, caso tenha ocorrido atraso.
- **Controle de acesso:** permissões diferenciadas entre administradores, funcionários e clientes.

### Papéis de Usuário

- **Cliente (Leitor)**
  - Se cadastrar e logar na plataforma.
  - Consultar livros.
  - Fazer reservas de livros.
  - Consultar empréstimos.
  - Pagar multa.
  - Alterar dados pessoais da conta.
- **Bibliotecário (Funcionário)**
  - Realizar login na plataforma.
  - Cadastrar, consultar e atualizar livros.
  - Consultar e atualizar os clientes.
  - Alterar dados pessoais (exceto a função).
  - Aprovar ou rejeitar cadastro de cliente.
  - Consultar e cadastrar empréstimos.
  - Registrar a separação do livro para retirada.
  - Registrar a retirada e devolução de livros.
  - Registrar a perda do livro.
  - Solicitar exclusão de livros.
  - Solicitar exclusão de clientes.

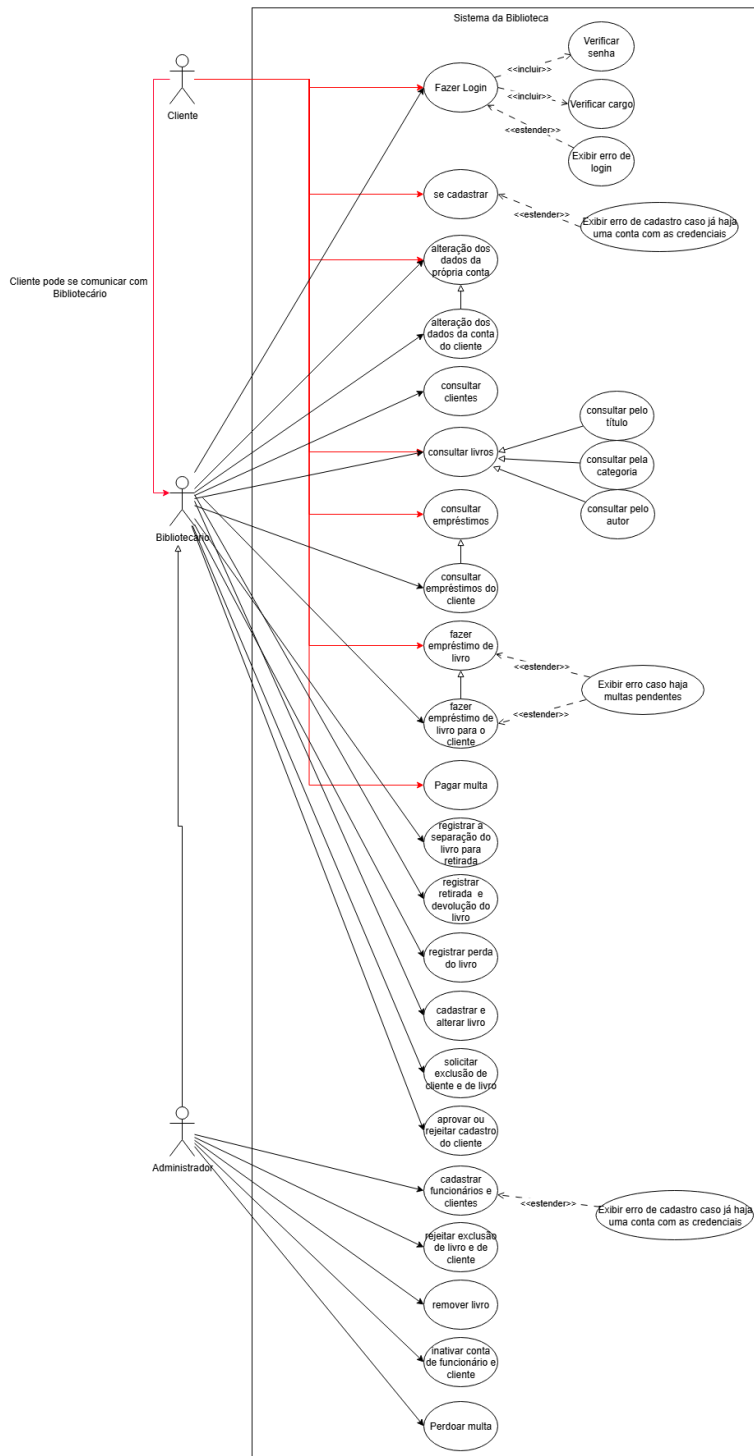
- **Administrador**

- Realizar login na plataforma.
- Cadastrar, consultar, atualizar e remover livros.
- Cadastrar, consultar, remover e atualizar qualquer usuário (exceto remover a própria conta).
- Alterar dados pessoais (exceto a função).
- Aprovar ou rejeitar cadastro de cliente.
- Consultar e cadastrar empréstimos.
- Registrar a separação do livro para retirada.
- Registrar a retirada e devolução de livros.
- Registrar a perda do livro.
- Perdoar multa.

### 3. Diagrama de Casos de Uso

O diagrama de casos de uso abaixo ilustra os atores que interagem com o sistema de gerenciamento de biblioteca, sendo eles: cliente, bibliotecário e administrador. Cada ator está associado com um conjunto específico de funcionalidades, como “consultar livros”, “fazer empréstimo do livro” e “perdoar multa”. O cliente, além de interagir com o sistema, pode se comunicar com o bibliotecário, caso necessite de algum auxílio. Já o administrador possui acesso a todas as funcionalidades disponíveis ao bibliotecário, além das suas funcionalidades exclusivas.

## CASOS DE USO



Link para visualização completa dos casos de uso:

- [diagrama de casos de uso](#)

## 4. Arquitetura do Sistema

A arquitetura do Sistema de Biblioteca Online é organizada em camadas, visando modularidade. Cada camada é responsável por um conjunto específico de funções, garantindo a separação de responsabilidades e facilitando futuras expansões.

### 4.1 Visão Geral

O sistema segue uma arquitetura **em três camadas (3-tier)**:

- **Camada de Apresentação (Front-end)**
  - Responsável pela interação com o usuário (clientes, funcionários e administradores).
  - Fornece interfaces gráficas acessíveis via navegador.
  - Principais funções:
    - Cadastro e login de usuários.
    - Consulta de livros disponíveis.
    - Solicitação de empréstimos e devoluções.
    - Painel administrativo para gestão do acervo.
- **Camada de Lógica de Negócio (Back-end)**
  - Implementa as regras de negócio do sistema.
  - Faz a validação de operações (ex.: verificar disponibilidade de livros, controlar prazos de devolução, calcular multas).
  - Principais módulos:
    - **Gestão de Usuários:** autenticação, controle de permissões e perfis.
    - **Gestão de Acervo:** cadastro, atualização e exclusão de livros.
    - **Gestão de Empréstimos:** registro de empréstimos e devoluções, cálculo de multas.
- **Camada de Dados (Banco de Dados)**
  - Responsável pelo armazenamento persistente das informações.
  - Contém as tabelas de usuários, livros, empréstimos e histórico.

### 4.2 Integração entre os Módulos

- A **Camada de Apresentação** comunica-se com a **Lógica de Negócio** através de APIs REST.
- A **Lógica de Negócio** acessa a **Camada de Dados** por meio de Mapeamento Objeto-Relacional (ORM) que permite a interação entre a aplicação e o banco de dados sem precisar escrever código SQL diretamente.

## 5. Diagrama de Classes

O diagrama de classes abaixo demonstra as classes que representam as entidades no sistema, além dos seus atributos, métodos e relacionamentos entre si.

### Classes

**Pessoa:** representa o usuário do sistema, guardando os seus dados e diferenciando o nível de acesso através do atributo função. Ela possui endereço e pode fazer reservas.

**Endereço:** representa o endereço do usuário. Possui um estado (ex: São Paulo, Rio de Janeiro, etc.).

**Estado:** representa o estado de um determinado país, tal como São Paulo, Rio de Janeiro, Espírito Santo, etc. Ele possui um país.

**País:** representa o país, tal como Brasil, Portugal, França, Chile, etc.

**Empréstimo:** representa a reserva feita pelo cliente de um livro, guardando dados importantes, tal como data de início do empréstimo, data de retirada do exemplar, status do empréstimo, etc. Ele possui multa e exemplar.

**Multa:** representa a multa, sendo aplicada caso haja perda ou atraso do exemplar, mudando seu status conforme as mudanças na situação do empréstimo.

**Exemplar:** representa a unidade física de um livro da biblioteca. Possui uma edição.

**Edicao:** representa a edição de um livro, como a edição de colecionador. Ela é composta por idioma, editora e título.

**Idioma:** representa a linguagem do livro, como o português, inglês, espanhol, etc.

**Editora:** representa a organização que publica e distribui o livro, tal como a Companhia das Letras.

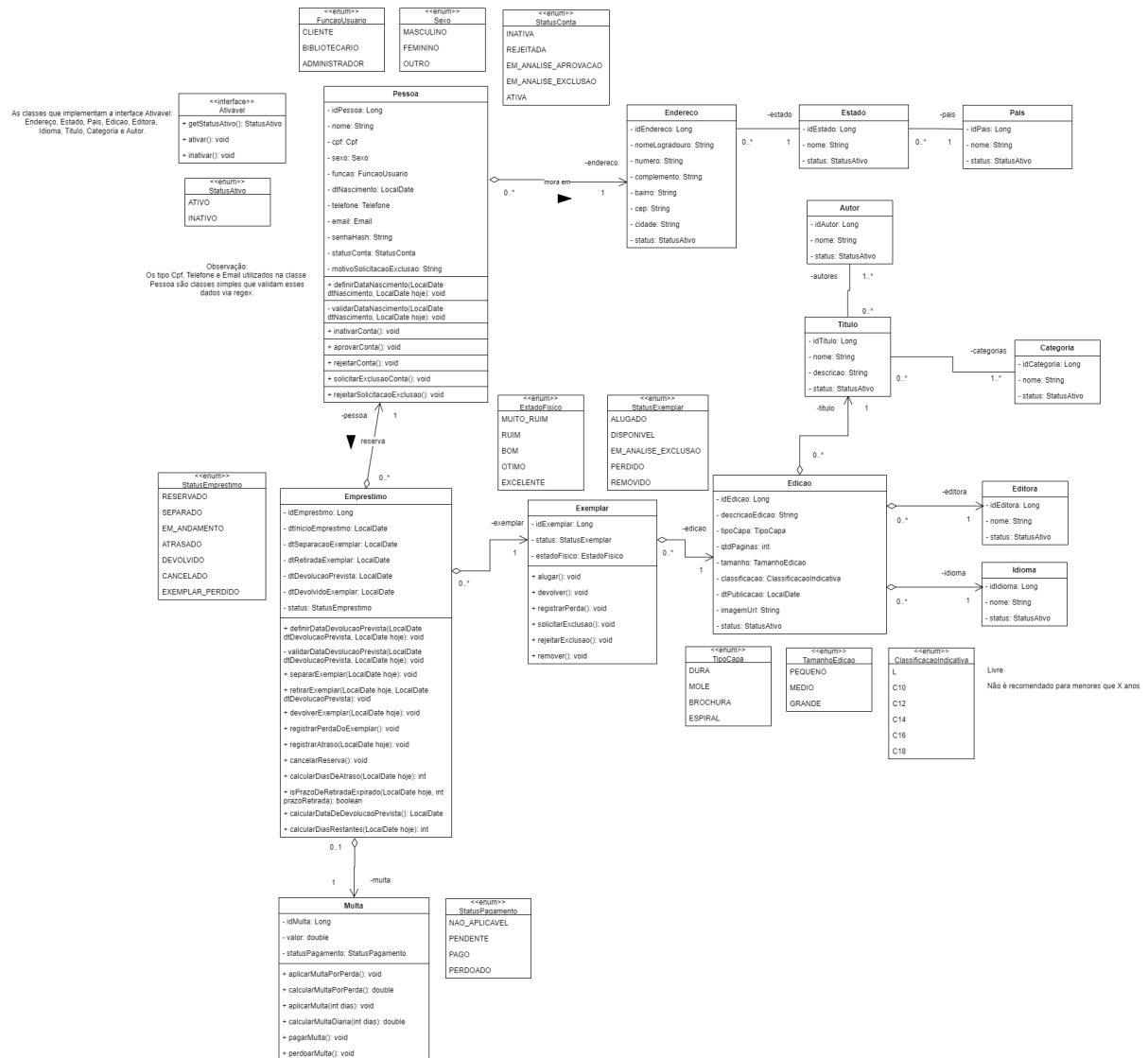
**Título:** representa o nome oficial de um livro, além de conter uma descrição sobre o que o livro aborda. Possui um ou mais autores e uma ou mais categorias.

**Autor:** representa o escritor do livro, tal como J. R. R. Tolkien, George R. R. Martin, Isaac Asimov, etc.



**Categoria:** representa o gênero que um livro pertence, servindo para classificar e organizar obras baseado na categoria que elas pertencem. Exemplos: ficção, drama, romance, terror, etc.

DIAGRAMA UML



## 6. Descrição dos Métodos

### Classe Pessoa

- **definirDataNascimento()**: insere a data de nascimento do usuário e aciona a validação da data.
- **validarDataNascimento()**: verifica se a data de nascimento está dentro dos limites de idade aceitos pelo sistema, utilizando como base a data atual.
- **inativarConta()**: altera o status da conta de “em análise exclusão” ou “ativa” para “inativa”.
- **aprovarConta()**: altera o status da conta de “em análise aprovação” para “ativa”.
- **rejeitarConta()**: altera o status da conta de “em análise aprovação” para “rejeitada”.
- **solicitarExclusaoConta()**: altera o status da conta de “ativa” para “em análise exclusão”.
- **rejeitarSolicitacaoExclusao()**: altera o status da conta de “em análise exclusão” para “ativa”.

### Classe Emprestimo

- **definirDataDevolucaoPrevista()**: insere a data de devolução prevista e aciona a validação da data.
- **validarDataDevolucaoPrevista()**: verifica se a data de devolução prevista respeita o limite máximo de prazo definido pelo sistema. Caso exista uma data anterior registrada, valida também se a nova data é posterior. A validação utiliza a data atual para verificar se a data de devolução prevista é no futuro.
- **separarExemplar()**: altera o status de “reservado” para “separado”.
- **retirarExemplar()**: altera o status de “separado” para “em andamento” e define a data de devolução prevista. Caso a data não seja informada, o sistema calcula automaticamente baseado na quantidade de páginas da edição do exemplar.
- **devolverExemplar()**: altera o status de “em andamento” ou “atrasado” para “devolvido”.
- **registrarPerdaDoExemplar()**: altera o status de “em andamento” ou “atrasado” para “exemplar perdido” e aciona a aplicação de multa por perda.
- **registrarAtraso()**: altera o status de “em andamento” para “atrasado” baseado na data atual e aciona a aplicação de multa por atraso.
- **cancelarReserva()**: altera o status de “separado” para “cancelado”.
- **calcularDiasDeAtraso()**: calcula a quantidade de dias de atraso em relação à data de devolução prevista, com base na data atual.

- **isPrazoDeRetiradaExpirado()**: verifica se o prazo de retirada expirou, considerando a data atual em relação ao prazo definido.
- **calcularDataDeDevolucaoPrevista()**: calcula a data de devolução prevista baseado na quantidade de páginas da edição do exemplar.
- **calcularDiasRestantes()**: calcula a quantidade de dias restantes até a data de devolução prevista, com base na data atual.

### Classe Multa

- **aplicarMultaPorPerda()**: altera o status de “não aplicável” para “pendente” e aciona o cálculo do valor da multa por perda.
- **calcularMultaPorPerda()**: calcula o valor fixo da multa aplicada por perda do exemplar.
- **aplicarMulta()**: altera o status de “não aplicável” para “pendente” e aciona o cálculo do valor diário por atraso.
- **calcularMultaDiaria()**: calcula o valor da multa diária aplicada por atraso.
- **pagarMulta()**: altera o status de “pendente” para “pago”.
- **perdoarMulta()**: altera o status de “pendente” para “perdoado”.

### Classe Exemplar

- **alugar()**: altera o status de “disponível” para “alugado”.
- **devolver()**: altera o status de “alugado” para “disponível”.
- **registrarPerda()**: altera o status do exemplar para “perdido”, normalmente a partir de “alugado”, mas também pode ser acionado no status “em análise exclusão” e “disponível”.
- **solicitarExclusao()**: altera o status de “disponível” para “em análise exclusão”.
- **rejeitarExclusao()**: altera o status de “em análise exclusão” para “disponível”.
- **remover()**: altera o status de “disponível” ou “em análise exclusão” para “removido”.

## 7. Plano de Testes

- **Testes Automatizados**

- Serão feitos testes automatizados com o objetivo de garantir que as classes e métodos do sistema estejam funcionando corretamente.

Para isso, serão desenvolvidos:

- **Testes unitários:** verificam partes específicas do código.
- **Testes de integração:** verificam a interação entre os componentes que formam o sistema, como as classes Pessoa, Emprestimo, Multa, etc.
- **Testes ponta a ponta:** verificam o fluxo completo do sistema, do front-end até o back-end.

- **Testes Manuais**

- Além dos testes automatizados, também serão feitos testes manuais para conferir se o sistema está funcionando bem como um todo. Isso inclui verificar a parte visual que o usuário vê (front-end), a troca de informações com o servidor (back-end) e outros pontos importantes da aplicação para garantir que tudo esteja funcionando corretamente.

- **Escopo de testes:**

- **Autenticação.**
- **Autorização** (se o usuário tem acesso para fazer uma determinada operação).
- **Validação de entrada** (como o formato do email, CPF e telefone).
- **Gerenciamento do cliente** (como a aprovação do cadastro, inativar uma conta, entre outros).
- **Gerenciamento de funcionários** (como cadastrar novos funcionários, alterar dados, entre outros).
- **Gerenciamento de livros** (como cadastrar exemplar, cadastrar edição, cadastrar título, entre outros).
- **Reserva de livros** (como a verificação se o cliente reservou mais livros do que a quantidade permitida, verificação se tem exemplares disponíveis para reserva, entre outros).
- **Gerenciamento de empréstimos** (como o cadastro de empréstimos, o fluxo de status do empréstimo, cálculo dos dias restantes, entre outros).
- **Gerenciamento de multas** (como aplicar multa por perda ou por atraso, cálculo do valor da multa, perdão da multa, entre outros).

- **Exemplos de casos de teste:**

- **Teste 1:** o usuário insere email, CPF e/ou telefone no formato inválido durante o cadastro.  
Resultado esperado: retornar um erro mostrando que o formato daquele dado está incorreto.
- **Teste 2:** o usuário tenta cadastrar com email já existente no sistema.  
Resultado esperado: retornar um erro mostrando que o email já existe no sistema.
- **Teste 3:** o usuário tenta reservar livros acima da quantidade permitida.  
Resultado esperado: retornar um erro mostrando o limite máximo de reservas.
- **Teste 4:** o usuário tenta reservar livros enquanto possui multa ativa.  
Resultado esperado: o sistema impede dele fazer a reserva e retorna falando que existe multa ativa.
- **Teste 5:** o usuário não insere todos os dados obrigatórios durante o cadastro.  
Resultado esperado: o sistema impede o cadastro enquanto ele não colocar todos os dados obrigatórios.

## 9. Diagrama de Sequência

Os quatro diagramas de sequência abaixo demonstram as principais funcionalidades do sistema, além da interação do usuário com o sistema e do sistema com o banco de dados.

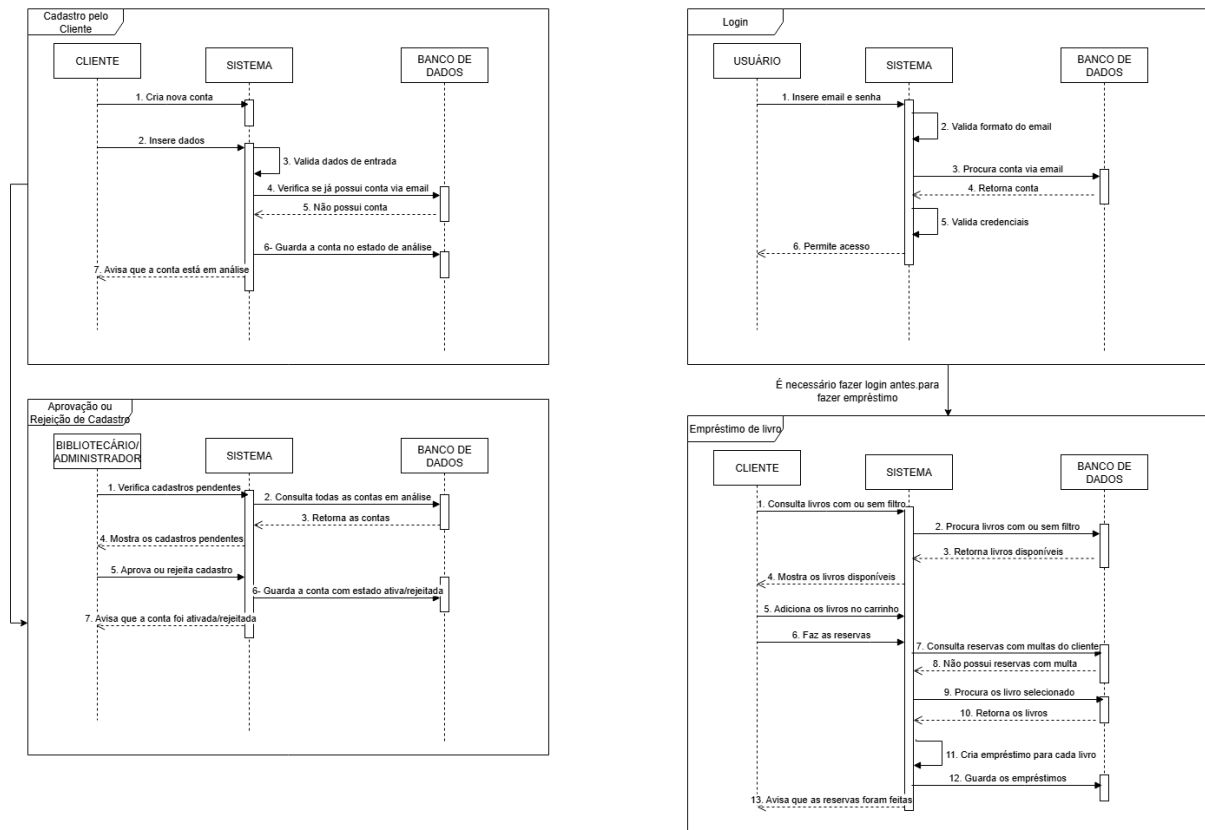
**Cadastro pelo Cliente:** ilustra o processo de cadastro que é feito pelo cliente no sistema, demonstrando as interações do cliente, a validação dos dados pelo sistema e o armazenamento dos dados no banco de dados.

**Aprovação ou Rejeição de Cadastro:** ilustra o processo do funcionário ou administrador para aprovar ou rejeitar o cadastro de conta do cliente. Sempre que o cliente se cadastrar, é necessário ter uma aprovação da conta para poder logar no sistema.

**Login:** ilustra o processo de login no sistema, ou seja, o processo de autenticação, validando as credenciais inseridas pelo usuário para permitir o acesso.

**Empréstimo de Livro:** ilustra o processo de reserva de um livro feito pelo cliente, desde a consulta dos livros disponíveis até a criação do empréstimo. É necessário o cliente fazer login no sistema para fazer o empréstimo.

## DIAGRAMAS DE SEQUÊNCIA



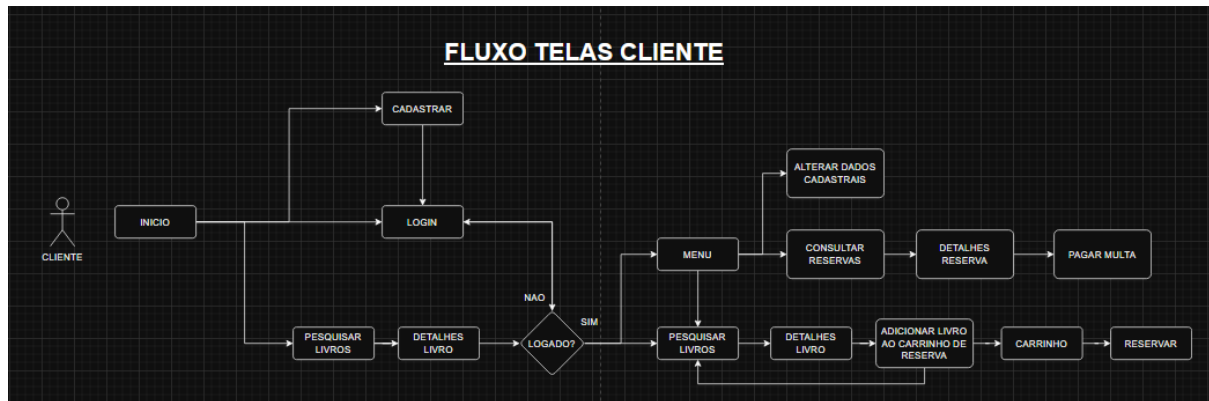
Observação: O funcionário vai poder fazer empréstimo para o cliente também

Link para visualização completa do diagrama de sequência:

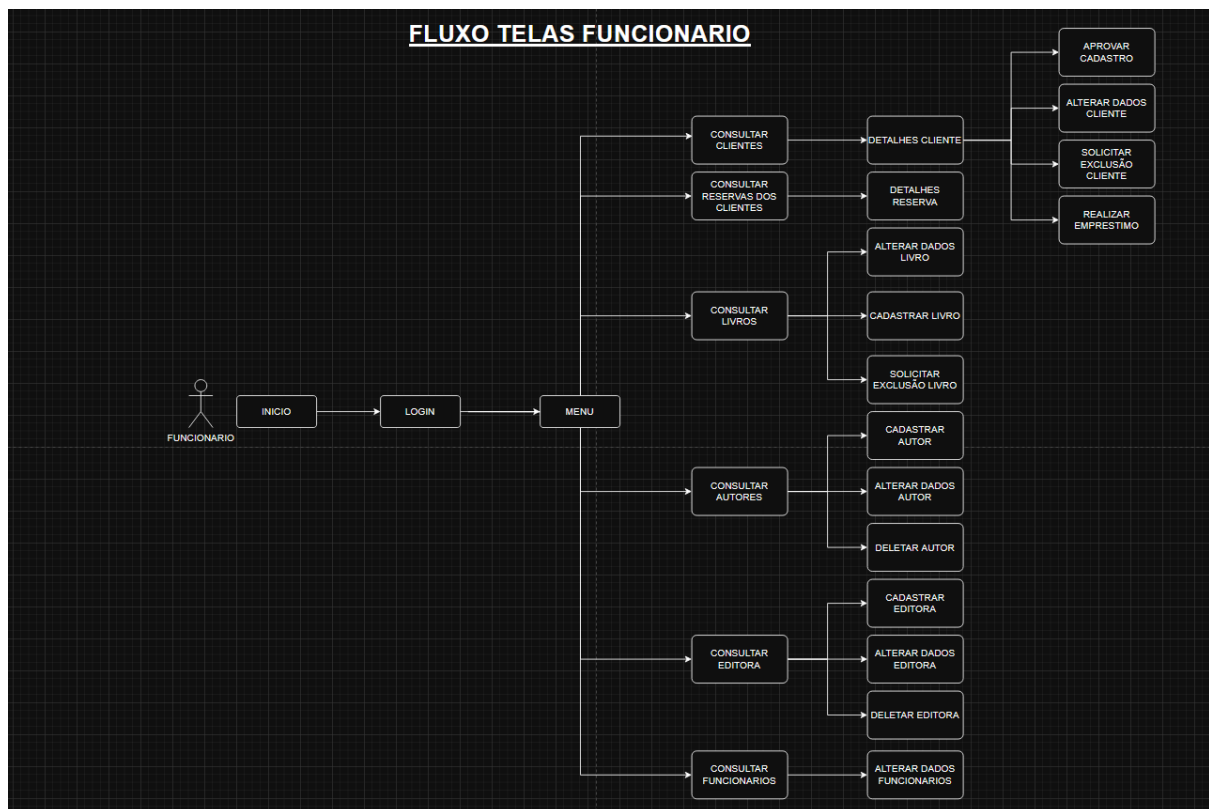
- [diagrama de sequência](#)

## 10. Fluxo de Telas

Demonstra por onde cada funcionalidade será acessada no front-end, por diferentes pontos de vista: cliente, funcionário e administrador.

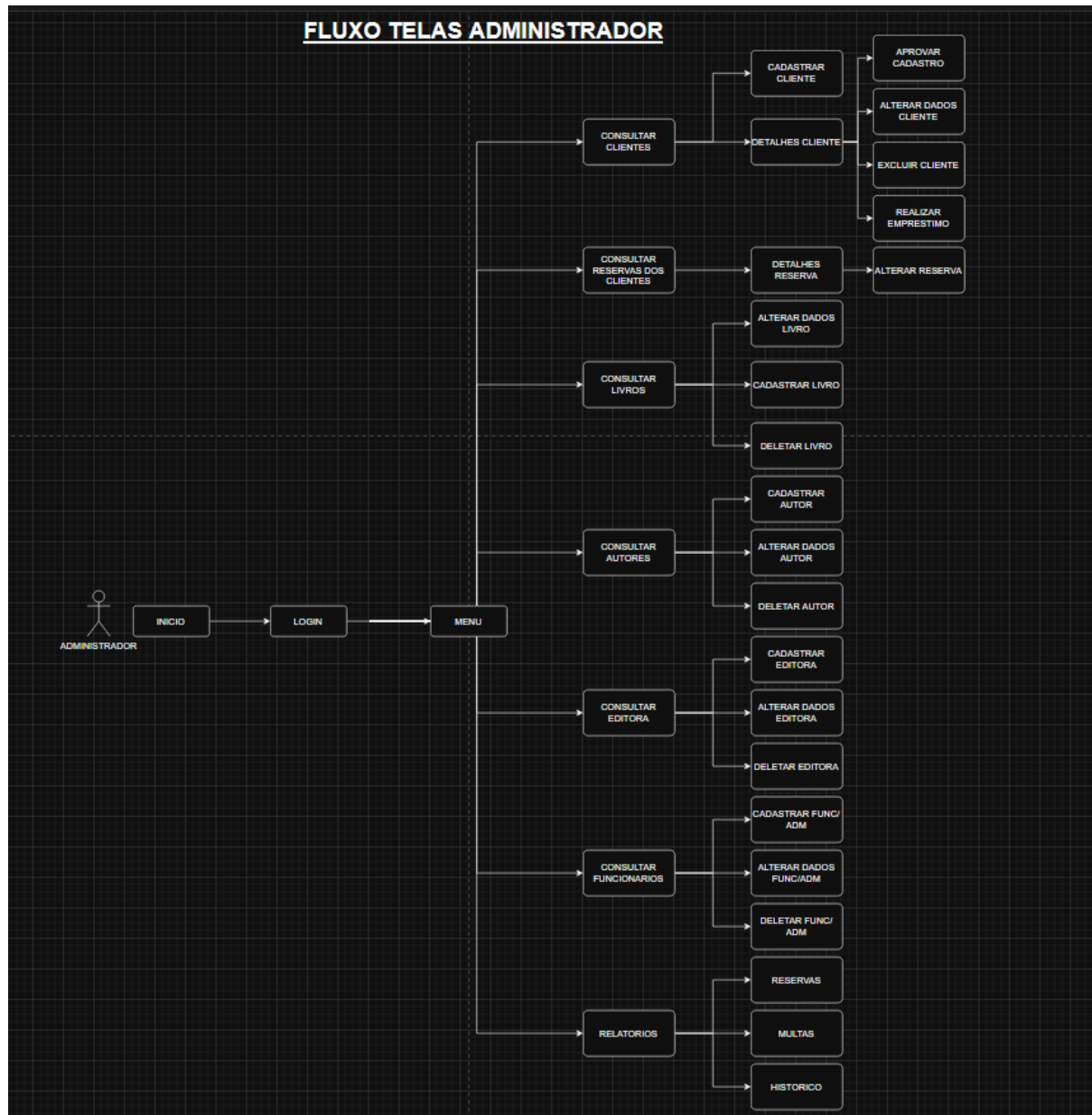


**Diagrama do cliente:** ponto de vista do cliente. O cliente entra pelo tela inicial onde pode pesquisar livros disponíveis, porém se tentar alugar um livro e não estiver logado será redirecionado para a tela de login, após se autenticar no sistema é liberado as funcionalidades extras como reservar, acessar detalhes de reservas e pagamentos de multa.





**Diagrama do funcionário:** ponto de vista do funcionário. O funcionário só terá acesso a suas funcionalidades após se autenticar no sistema, através de um menu que somente o funcionário poderá ver. Pode acessar as funcionalidades exclusivas do mesmo.



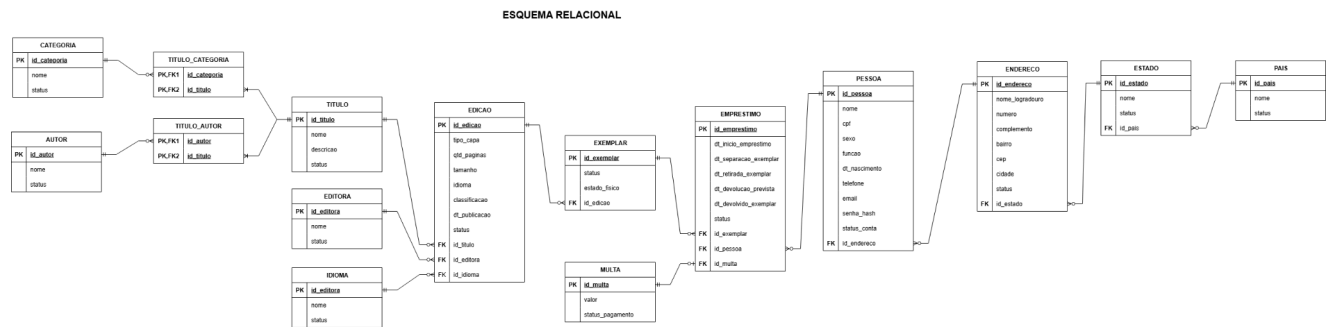
**Diagrama do administrador:** ponto de vista do administrador. O administrador só terá acesso a suas funcionalidades após se autenticar no sistema, através de um menu que somente o administrador poderá ver. Pode acessar as funcionalidades exclusivas do mesmo.

Link para visualização completa do fluxo de telas:

- [fluxo de telas](#)

## 11. Esquema Relacional

O diagrama abaixo mostra as entidades com os seus atributos e como se relacionam entre si no banco de dados. Essa representação visual facilita a compreensão de como os dados do sistema serão armazenados e como eles se relacionam.



Link para visualização completa do esquema relacional:

- [esquema relacional](#)

## INTEGRANTES

- Caio Felix
- Vinicius Sousa
- Diego Fonseca
- Gabriel Vicente