

Capa

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO**  
**CAMPUS SÃO PAULO**

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**GUSTAVO ALMEIDA DE OLIVEIRA**  
**MATHEUS FOGAÇA SARRETO**  
**MATHEUS VICTOR FELICIANO JUPI**

**SISTEMA DE BIBLIOTECA**

**SÃO PAULO**  
**2025**

Contracapa

**GUSTAVO ALMEIDA DE OLIVEIRA  
MATHEUS FOGAÇA SARRETO  
MATHEUS VICTOR FELICIANO JUPI**

**SISTEMA DE BIBLIOTECA**

Trabalho apresentado ao Programa do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus de São Paulo, como requisito parcial para aprovação na disciplina SPOBDD2 – Banco de Dados 2, sob orientação do Professor Francisco.

**SÃO PAULO**

**2025**

## Sumário

Introdução.....	1
Requisitos .....	2
Análise de requisitos.....	2
Requisitos Funcionais.....	3
Requisitos Não Funcionais .....	3
Cronograma de Atividades .....	5
Visão Espacial da Aplicação .....	7
Prototipação de Página no Figma.....	11
Considerações Finais.....	17
Referências Bibliográficas .....	18

## **Introdução**

Este trabalho aborda a estrutura de um sistema de gestão de bibliotecas, destacando os principais atores – como usuários comuns, funcionários da biblioteca e administradores – e os casos de uso que orientam suas interações. Dentre as funcionalidades analisadas, incluem-se o cadastro de livros, o gerenciamento de usuários, o controle de empréstimos e devoluções. A proposta visa demonstrar como um sistema bem planejado pode melhorar a eficiência operacional e proporcionar uma experiência mais prática e intuitiva para todos os envolvidos.

# Requisitos

## Análise de requisitos

### I. Identificação dos Atores

- **Funcionário da Biblioteca:** Pessoa responsável por cadastrar livros, gerenciar usuários, controlar empréstimos e devoluções.

### II. Casos de Uso e Descrição de Funcionalidades

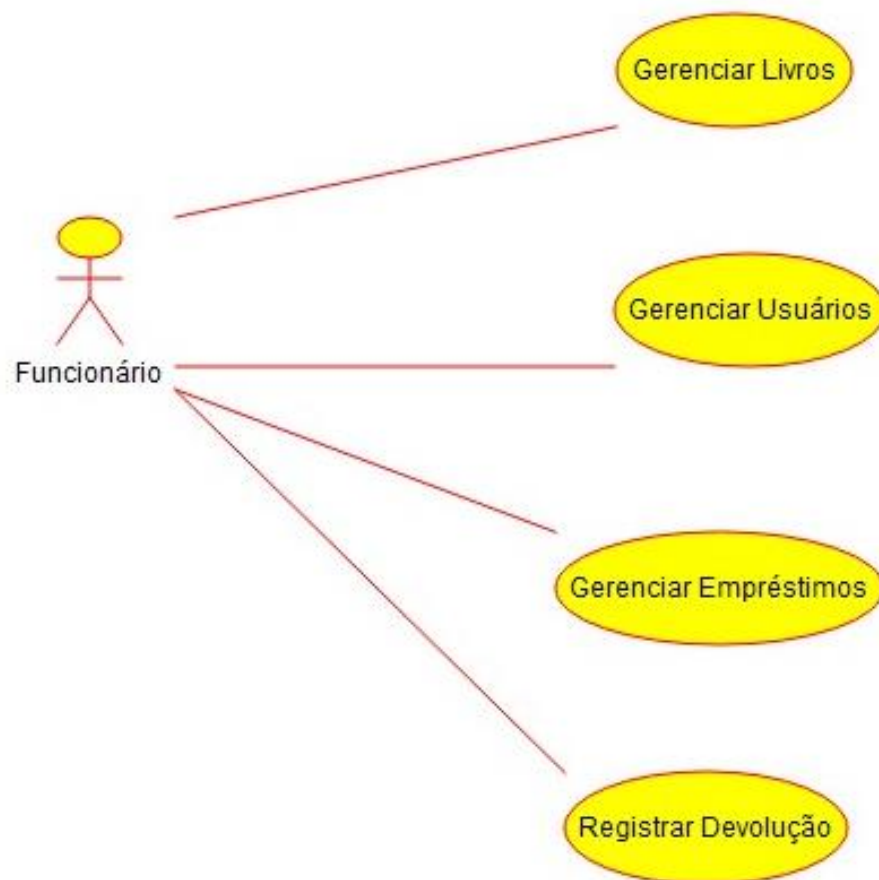
Cada caso de uso representa uma interação entre o sistema e os atores. Aqui estão alguns dos casos de uso principais:

- **Cadastrar e Gerenciar Livro**
  - **Descrição:** Permite que o funcionário da biblioteca registre novos livros, preenchendo informações como título, autor, editora, gênero, ano de publicação e ISBN.
  - **Ator:** Funcionário da Biblioteca
  - **Pré-condição:** O funcionário está autenticado no sistema.
  - **Pós-condição:** O livro está registrado no banco de dados.
- **Gerenciar Usuários**
  - **Descrição:** Permite que o funcionário registre novos usuários, atualize ou remova usuários existentes.
  - **Ator:** Funcionário da Biblioteca
  - **Pré-condição:** O funcionário está autenticado.
  - **Pós-condição:** O usuário é adicionado, atualizado ou removido da lista de usuários.
- **Realizar Empréstimo**
  - **Descrição:** Permite o registro de um empréstimo, associando um usuário a um livro, com data de retirada e data de devolução prevista.
  - **Ator:** Funcionário da Biblioteca
  - **Pré-condição:** O funcionário está autenticado e o usuário está cadastrado no sistema.
  - **Pós-condição:** O livro é marcado como emprestado no sistema.
- **Registrar Devolução**
  - **Descrição:** Registra a devolução de um livro, liberando-o para novos empréstimos.

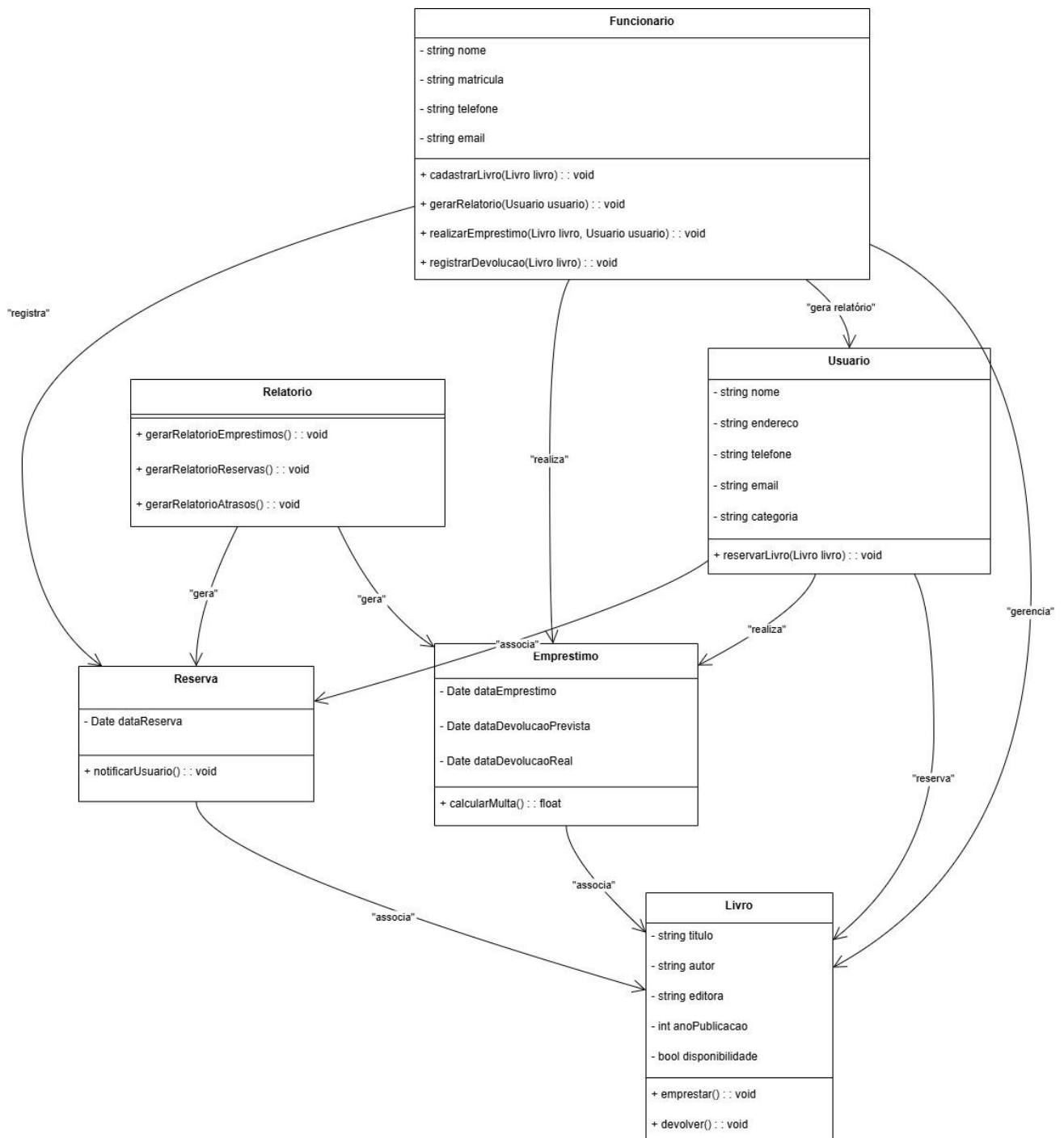
- **Ator:** Funcionário da Biblioteca
- **Pré-condição:** O livro foi emprestado e tem um registro no sistema.

### III. Diagramas

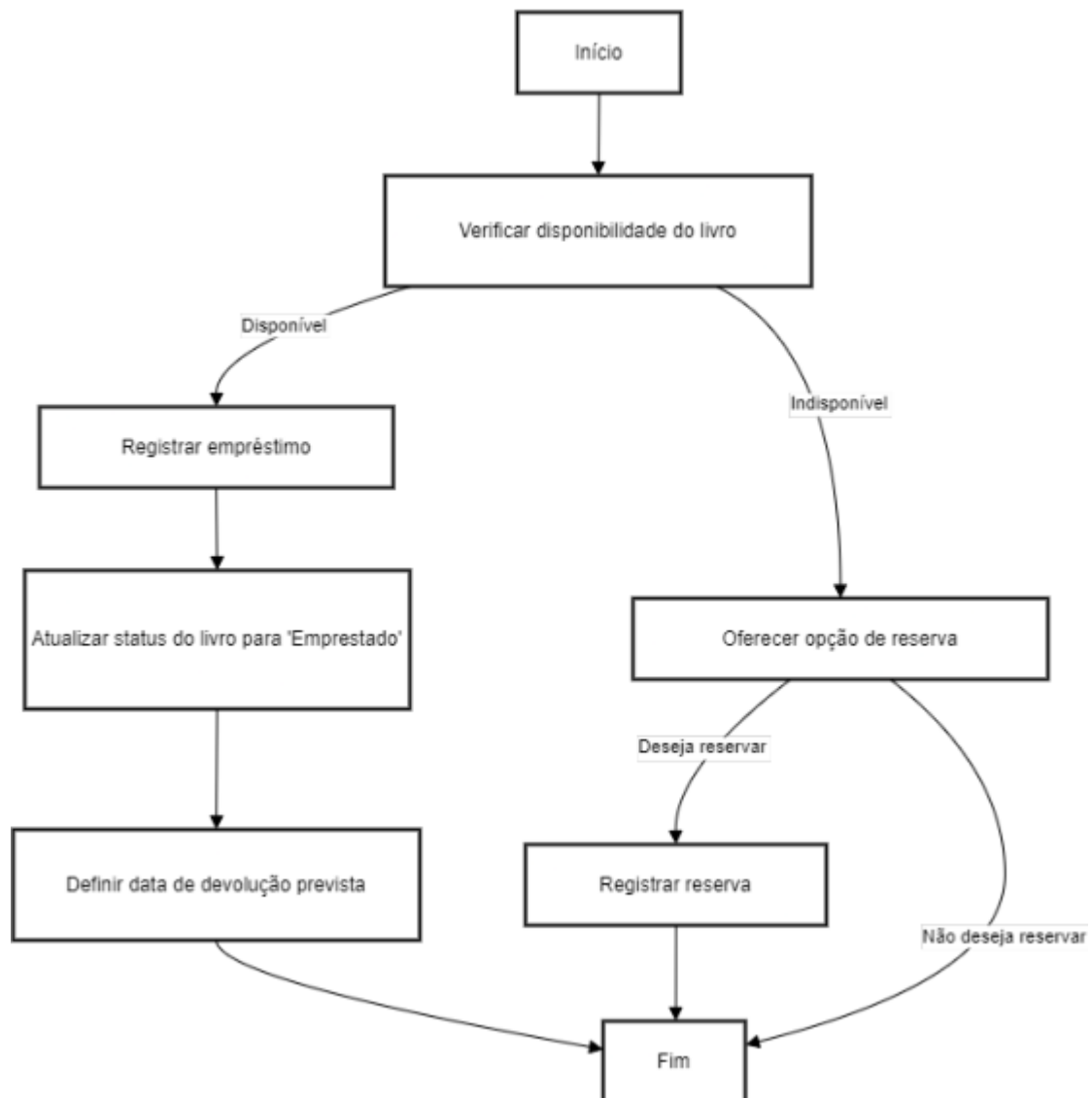
#### 1. Diagrama Caso de Uso



## 2. Diagrama de Classes



### 3. Diagrama de Atividade



## IV. Requisitos Funcionais

### Cadastro de Livros:

O sistema deve permitir o cadastro de livros com informações como título, autor, editora, ano de publicação, gênero, e ISBN.

Deve ser possível editar, visualizar e excluir dados dos livros cadastrados.

### Gestão de Usuários:

O sistema deve permitir o cadastro de usuários da biblioteca, incluindo informações como nome, endereço, telefone, e e-mail.

Os usuários podem ser classificados em categorias, como estudantes, professores ou visitantes.

### Controle de Empréstimos:



**Devoluções e Renovação de Empréstimos:**

O sistema deve permitir a devolução de livros e registrar a data de devolução.

O sistema deve possibilitar a renovação de um empréstimo, alterando a data de devolução.

**Consulta e Pesquisa:**

O sistema deve permitir a busca de livros por diferentes critérios (título, autor, gênero, ISBN). O sistema deve possibilitar a busca por usuários cadastrados.

**V. Requisitos Não Funcionais****1. Usabilidade:**

- A interface do sistema deve ser amigável e intuitiva, facilitando o uso por pessoas sem conhecimentos técnicos.
- O sistema deve permitir fácil navegação entre as funcionalidades de cadastro, consulta, empréstimos e devoluções.

**2. Desempenho:**

- O sistema deve responder rapidamente às consultas, mesmo quando houver um grande número de registros de livros e usuários.
- O tempo de resposta para buscas e operações de empréstimo deve ser inferior a 3 segundos.

**3. Segurança:**

- O sistema deve exigir autenticação para acessar as funções de gestão (ex.: cadastro de livros, gerenciamento de empréstimos).
- Os dados dos usuários devem ser protegidos, e o sistema deve oferecer controle de acesso baseado em níveis de permissão.

**4. Disponibilidade:**

- O sistema deve estar disponível para acesso durante o horário de funcionamento da biblioteca.
- Em caso de uso em ambiente web, deve garantir pelo menos 99% de disponibilidade durante o horário de pico.
- Escalabilidade:
- O sistema deve ser projetado para suportar o aumento do número de usuários e registros de livros sem perda de desempenho significativo.
- Deve ser possível adaptar o sistema para novas funcionalidades, como integração com outros sistemas de bibliotecas.

- Compatibilidade:
- O sistema deve ser compatível com diferentes navegadores e dispositivos (se for uma aplicação web).
- O sistema deve permitir a exportação de dados em formatos padrão, como CSV ou PDF, para integração com outros sistemas de relatórios.
- **Pós-condição:** O livro é marcado como disponível.

## **Cronograma de atividades**

### **1ª Passo: Levantamento de Requisitos**

- Reunião com o cliente para entender as necessidades do sistema.
- Identificar requisitos funcionais e não funcionais.

### **2ª Passo: Análise de Requisitos**

- Detalhar e validar requisitos levantados.
- Definir casos de uso principais.
- Criar o diagrama de casos de uso.

### **3ª Passo: Projeto de Sistema**

- Desenvolver o diagrama de classes e o diagrama de atividades.
- Definir a arquitetura do sistema: tecnologias para frontend, backend e banco de dados.

### **4ª Passo: Projeto de Interface**

- Esboçar protótipos das interfaces principais:
  - Tela de login.
  - Tela de busca de livros.
  - Tela de cadastro de usuário.
  - Tela de gerenciamento de empréstimos.

### **5ª Passo: Projeto de Banco de Dados**

- Definir o modelo de dados (lógico e físico).
- Criar o esquema do banco de dados (tabelas para livros, usuários, empréstimos e reservas).

### **6ª Passo: Configuração do Ambiente**

- Configurar o ambiente de desenvolvimento com ferramentas e frameworks definidos (ex.: Node.js, Express, MySQL).

### 7ª Passo: Implementação – Backend (Parte 1)

- Desenvolver a API para funcionalidades principais:
  - Cadastro de livros e usuários.
  - Gerenciamento de empréstimos e devoluções.
  - Registro de reservas.

### 8ª Passo: Implementação – Backend (Parte 2)

- Continuar o desenvolvimento do backend.
- Implementar endpoints para geração de relatórios e consulta de disponibilidade de livros.

### 9ª Passo: Implementação – Frontend

- Desenvolver interfaces para:
  - Busca de livros.
  - Cadastro de usuários.
  - Empréstimos e reservas.

### 10ª Passo: Integração

- Integrar o frontend com o backend.
- Testar a comunicação entre a interface de usuário e as funcionalidades da API.

### 11ª Passo: Testes e Ajustes

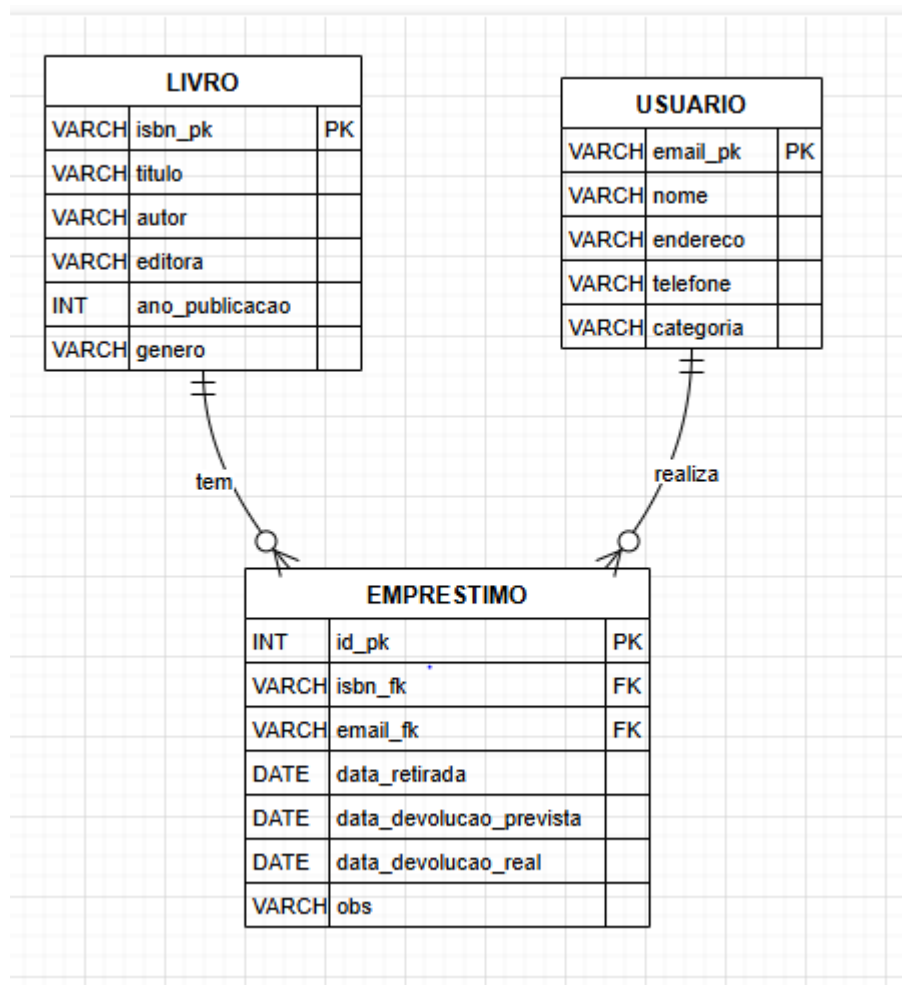
- Realizar testes de unidade, integração e usabilidade.
- Corrigir erros e fazer ajustes conforme feedback dos testes.

### 12ª Passo: Implantação e Treinamento

- Preparar o ambiente de produção e implantar o sistema.
- Realizar treinamento dos usuários finais (funcionários e administradores da biblioteca)

## Visão Espacial da Aplicação

### Projeto Lógico



#### LIVRO

A entidade LIVRO representa os livros disponíveis na biblioteca. Cada livro é identificado de maneira única pelo atributo isbn\_pk.

##### Atributos:

- isbn\_pk (VARCHAR) - Identificador único do livro (chave primária).
- titulo (VARCHAR) - Nome do livro.
- autor (VARCHAR) - Nome do autor do livro.
- editora (VARCHAR) - Nome da editora responsável pela publicação.
- ano\_publicacao (INT) - Ano de publicação do livro.
- genero (VARCHAR) - Gênero literário do livro.

##### Relacionamentos:

Um livro pode estar associado a vários empréstimos (1:N).

A relação "tem" indica que um livro pode ser emprestado diversas vezes, mas cada empréstimo está vinculado a apenas um livro.

## **USUARIO**

A entidade USUARIO representa os indivíduos cadastrados na biblioteca e autorizados a realizar empréstimos de livros. Cada usuário é identificado unicamente pelo atributo email\_pk.

### **Atributos:**

- email\_pk (VARCHAR) - Identificador único do usuário (chave primária).
- nome (VARCHAR) - Nome do usuário.
- endereco (VARCHAR) - Endereço residencial do usuário.
- telefone (VARCHAR) - Número de telefone para contato.
- categoria (VARCHAR) - Categoria do usuário (exemplo: estudante, professor, visitante).

### **Relacionamentos:**

- Um usuário pode realizar vários empréstimos (1:N).
- A relação "realiza" indica que um usuário pode emprestar diversos livros ao longo do tempo.

## **EMPRESTIMO**

A entidade EMPRESTIMO registra os empréstimos realizados pelos usuários da biblioteca. Cada empréstimo está vinculado a um livro específico e a um usuário.

### **Atributos:**

- id\_pk (INT) - Identificador único do empréstimo (chave primária).
- isbn\_fk (VARCHAR) - Chave estrangeira que referencia o livro emprestado.
- email\_fk (VARCHAR) - Chave estrangeira que referencia o usuário que realizou o empréstimo.

- data\_retirada (DATE) - Data em que o livro foi retirado da biblioteca.
- data\_devolucao\_prevista (DATE) - Data prevista para a devolução do livro.
- data\_devolucao\_real (DATE) - Data efetiva da devolução do livro (pode ser nula se o livro ainda não foi devolvido).
- obs (VARCHAR) - Observações adicionais sobre o empréstimo.

#### **Relacionamentos:**

Cada empréstimo está associado a um único livro e a um único usuário (N:1).

As chaves estrangeiras isbn\_fk e email\_fk garantem a integridade referencial, impedindo empréstimos de livros inexistentes ou por usuários não cadastrados.

### **INTEGRAÇÃO ENTRE AS ENTIDADES**

O modelo proposto estabelece a seguinte estrutura de relacionamento:

**LIVRO e EMPRESTIMO (1:N)** → Um livro pode ser emprestado diversas vezes ao longo do tempo.

**USUARIO e EMPRESTIMO (1:N)** → Um usuário pode realizar múltiplos empréstimos.

As chaves estrangeiras em EMPRESTIMO garantem que:

Apenas livros existentes no sistema podem ser emprestados.

Somente usuários cadastrados podem realizar empréstimos.

### **Projeto Físico**

```
CREATE TABLE LIVRO (
    isbn_pk VARCHAR(20) PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    autor VARCHAR(255) NOT NULL,
    editora VARCHAR(255) NOT NULL,
    ano_publicacao INT NOT NULL,
    genero VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE USUARIO (
    email_pk VARCHAR(255) PRIMARY KEY,
```

```
nome VARCHAR(255) NOT NULL,  
endereco VARCHAR(255) NOT NULL,  
telefone VARCHAR(20),  
categoria VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE EMPRESTIMO (  
    id_pk INT AUTO_INCREMENT PRIMARY KEY,  
    isbn_fk VARCHAR(20) NOT NULL,  
    email_fk VARCHAR(255) NOT NULL,  
    data_retirada DATE NOT NULL,  
    data_devolucao_prevista DATE NOT NULL,  
    data_devolucao_real DATE,  
    obs VARCHAR(255),  
    FOREIGN KEY (isbn_fk) REFERENCES LIVRO(isbn_pk) ON DELETE  
CASCADE,  
    FOREIGN KEY (email_fk) REFERENCES USUARIO(email_pk) ON DELETE  
CASCADE  
);
```

# Prototipação de Página no Figma

## Gerenciar e Cadastrar Livros:

The image shows a Figma prototype of a book management system interface. At the top, a dark blue notification box displays the text "localhost:3000 diz" and "Livro cadastrado com sucesso!" with an "OK" button. Below this is a form for adding a new book. The form fields are: ISBN (9786555320388), Título (Duna), Autor (Frank Herbert), Editora (Aleph), Ano de Publicação (1965), and Gênero (Ficção Científica). A red "Adicionar Livro" button is at the bottom of the form. Below the form is a table titled "Lista de Livros" with columns: ISBN, Título, Autor, Editora, Ano, Gênero, and Ações. The table contains two rows of book data. The first row is for "Neuromancer" by William Gibson, published in 1984 by Aleph, categorized as Cyberpunk. The second row is for "O Problema dos Três Corpos" by Liu Cixin, published in 2008 by Suma, categorized as Ficção Científica. Each row has "Editar" and "Deletar" buttons in the Ações column.

ISBN	Título	Autor	Editora	Ano	Gênero	Ações
9786555320052	Neuromancer	William Gibson	Aleph	1984	Cyberpunk	<button>Editar</button> <button>Deletar</button>
9786555600413	O Problema dos Três Corpos	Liu Cixin	Suma	2008	Ficção Científica	<button>Editar</button> <button>Deletar</button>

O processo principal é permitir que o usuário (bibliotecário) cadastre novos livros no sistema.

O formulário de cadastro deve coletar informações essenciais sobre o livro, como título, autor, editora, ano de publicação e gênero.

### Validação de Dados:

Antes de cadastrar o livro, o sistema deve validar se todos os campos obrigatórios foram preenchidos corretamente.

Por exemplo, o campo "Ano de Publicação" deve ser um número válido, e o campo "Título" não pode estar vazio.

## Funções Necessárias

### Função de Cadastro:

Uma função para capturar os dados do formulário e adicionar, deletar ou atualizar o livro à lista de livros cadastrados.



**Função de Exibição:**

Uma função para exibir a lista de livros cadastrados nos dados do cliente.

**Informações Necessárias****Campos do Formulário:**

- **ISBN:** Código do livro.
- **Título:** Nome do livro.
- **Autor:** Nome do autor do livro.
- **Editores:** Nome da editora que publicou o livro.
- **Ano de Publicação:** Ano em que o livro foi publicado.
- **Gênero:** Categoria ou gênero literário do livro.

**Dados Armazenados:**

Cada livro cadastrado deve ser armazenado com os seguintes atributos:

- Título
- Autor
- Editora
- Ano de Publicação
- Gênero

## Gerenciar Usuários:

**Gerenciar Usuários**

[Voltar](#)

**Adicionar Usuário**

E-mail:

E-mail

Nome:

Nome

Endereço:

Endereço

Telefone:

Telefone

Categoria:

Categoria

**Adicionar Usuário**

**Lista de Usuários**

Email	Nome	Endereço	Telefone	Categoria	Ações
adriana.lima@email.com	Adriana Lima	Travessa dos Pássaros, 753	11981236547	Estudante	<a href="#">Editar</a> <a href="#">Deletar</a>
aline.cavalcante@email.com	Aline Cavalcante	Rua Azul, 951	11998765489	Estudante	<a href="#">Editar</a> <a href="#">Deletar</a>
					<a href="#">Editar</a>

O formulário de cadastro deve coletar informações essenciais sobre o usuário, como email, nome, endereço, telefone e categoria.

### Armazenamento de Dados:

Após a validação, os dados do usuário devem ser armazenados em uma lista ou banco de dados para uso futuro.

## Funções Necessárias

### Função de Cadastro:

Uma função para capturar os dados do formulário e adicionar o cliente à lista de usuários cadastrados.

### Função de Exibição:

Uma função para exibir a lista de clientes cadastrados.

## **Informações Necessárias**

### **Campos do Formulário:**

- **Email:** Endereço de email único do usuário (chave primária).
- **Nome:** Nome completo do usuário.
- **Endereço:** Endereço residencial do usuário.
- **Telefone:** Número de telefone do usuário.
- **Categoria:** Categoria do usuário (por exemplo, estudante, professor, funcionário).

### **Dados Armazenados:**

**Cada usuário cadastrado deve ser armazenado com os seguintes atributos:**

- Email.
- Nome.
- Endereço.
- Telefone.
- Categoria.

# Gerenciar Empréstimos

## Gerenciar Empréstimos

[Voltar](#)

### Realizar Empréstimo

ISBN:

E-mail do Usuário:

Data de Retirada:

Data de Devolução Prevista:

Registrar Empréstimo

### Lista de Empréstimos

ID	ISBN	E-mail do Usuário	Data de Retirada	Data de Devolução Prevista	Data de Devolução	Ações
4	9788582850853	maria.santos@email.com	05-01-2024	19-01-2024	18-01-2024	<div><p>Devolver</p><p>Excluir</p></div>
5	9788582850853	maria.santos@email.com	05-01-2024	19-01-2024	17-01-2024	<div><p>Devolver</p><p>Excluir</p></div>
6	9788582850853	maria.santos@email.com	05-01-2024	19-01-2024	19-01-2024	<div><p>Devolver</p><p>Excluir</p></div>

Permite o registro de empréstimos de livros para os clientes.

## Processos:

- Seleção do livro pelo cliente.
- Registro das datas de retirada e devolução.
- Armazenamento dos dados do empréstimo.

## Informações Necessárias

- **ISBN do Livro:** Identificação do livro emprestado.
- **Email do Usuário:** Identificação do cliente que está alugando o livro.
- **Data de Retirada:** Data em que o livro foi retirado.
- **Data de Devolução Prevista:** Data prevista para a devolução do livro.

## Funções

- Captura dos dados do formulário.
- Armazenamento dos dados.
- Exibição da lista de empréstimos

## Tecnologias Utilizadas

### Frontend:

**HTML, CSS e JavaScript:** Para criar a interface de usuário.

**GitHub :** Para hospedar a aplicação frontend.

### Backend:

**Node.js:** Para implementar o servidor e as APIs REST que interagem com o banco de dados.

**Microsoft SQL Server:** Para armazenar as informações sobre livros, usuários e empréstimos.

**Azure:** Para hospedar o banco de dados e o backend.

### Controle de Versão:

**Git:** Para controle de versão do código-fonte.

**GitHub:** Para repositório de código(<https://github.com/Gustavo01010101/projeto-bd>)

## Considerações Finais

O sistema desenvolvido oferece uma solução eficiente para o gerenciamento de empréstimos de livros, permitindo que os usuários e administradores realizem operações de forma simples e intuitiva. A integração do frontend com o backend via API e a utilização de um banco de dados robusto garantem a escalabilidade e segurança do sistema.

Embora o sistema tenha sido projetado para atender a um conjunto específico de requisitos, ele pode ser expandido com funcionalidades adicionais, como notificações de devolução, controle de multas, demonstrando a alta estabilidade do sistema.

## **Referências Bibliográficas**

- I. Microsoft, "SQL Server Documentation," Microsoft Docs. Disponível em:  
<https://docs.microsoft.com/en-us/sql/sql-server/>.
- II. "Node.js Documentation," Node.js Foundation, 2021. Disponível em:  
<https://nodejs.org/en/docs/>.
- III. "Azure SQL Database Documentation" Disponível em:  
<https://learn.microsoft.com/en-us/azure/azure-sql/database/>.