

INSTITUTO FEDERAL DO PIAUÍ - Teresina Central  
CURSO DE TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS  
DISCIPLINA: Engenharia de Software  
Profº. Duany Dreyton Bezerra Sousa  
Aluno: David Sousa da Silva

### **Atividade Prática: Modelagem de Sistemas com UML**

Descrição do Software pelo Cliente: "Preciso de um sistema um pouco mais completo para gerenciar a minha biblioteca pessoal. Quero poder cadastrar os livros que tenho, com título, autor, ano de publicação e se está emprestado ou não. Gostaria de registrar para quem emprestei cada livro e quando foi emprestado. O sistema deve me permitir buscar livros por título ou autor e também gerar uma lista dos livros que estão emprestados no momento, com a informação de quem pegou emprestado. Além disso, quero ter um cadastro de clientes, com nome, telefone e endereço, para facilitar o registro dos empréstimos. Gostaria de poder buscar clientes por nome e também listar todos os clientes cadastrados. É importante que o sistema seja fácil de usar, tanto para mim quanto para os clientes que pegam livros emprestados, e que seja acessível de qualquer lugar com internet."

#### **Parte 1: Identificação de Elementos**

##### **Atores:**

Identifique os possíveis atores envolvidos no sistema.  
Dono da biblioteca.  
Clientes.

##### **Requisitos:**

Liste os requisitos funcionais e não-funcionais do sistema, com base na descrição do cliente.

##### **Requisitos Funcionais:**

###### **RF001 – Cadastro de Livros**

O sistema deve permitir ao dono da biblioteca<sup>1.1</sup> cadastrar livros, com as seguintes informações:

Título  
Autor  
Ano de publicação  
Status de empréstimo (emprestado ou disponível)

###### **RF002 – Cadastro de Clientes**

O sistema deve permitir ao dono da biblioteca cadastrar novos clientes, armazenando as seguintes informações:

Nome  
Email  
Senha  
Telefone  
Endereço

#### **RF003 – Registro de Empréstimos**

O sistema deve permitir ao dono da biblioteca registrar o empréstimo de um livro, associando-o a um cliente e armazenando:

Livro emprestado  
Cliente que pegou o livro  
Data de empréstimo  
Data limite de devolução

#### **RF004 – Busca de Livros por Título, Autor e Ano**

O sistema deve permitir ao dono da biblioteca ou ao cliente buscar livros filtrando pelo título, autor e ano.

#### **RF005 – Busca de Clientes por Nome**

O sistema deve permitir ao dono da biblioteca buscar clientes pelo nome.

#### **RF006 – Listagem de Livros Emprestados**

O sistema deve gerar uma lista dos livros que estão emprestados no momento, incluindo:

Nome do cliente que pegou o livro  
Nome do Livro  
Data de empréstimo  
Data limite de devolução

#### **RF007 – Listagem de Clientes**

O sistema deve listar todos os clientes cadastrados, exibindo:

Nome  
Telefone  
Endereço

#### **RF008 – Acessibilidade pela Internet**

O sistema deve ser acessível de qualquer local com conexão à internet.

Requisitos Não Funcionais:

#### **RNF001 – Usabilidade**

O sistema deve ser fácil de usar, com uma interface intuitiva, para que o dono da biblioteca e os clientes possam interagir com ele sem necessidade de treinamento técnico avançado.

**RNF002 – Acessibilidade via Internet**

O sistema deve ser acessível remotamente de qualquer lugar com conexão à internet, permitindo que o dono e os clientes utilizem o sistema sem estarem fisicamente no local da biblioteca.

**RNF003 – Segurança de Dados**

O sistema deve garantir a segurança dos dados, permitindo que apenas o dono da biblioteca tenha acesso a funcionalidades de cadastro, edição e remoção de dados (livros, clientes, e empréstimos).

Os dados pessoais dos clientes devem ser protegidos conforme as boas práticas de segurança.

**RNF004 – Tempo de Resposta**

O tempo de resposta do sistema para buscar livros, clientes ou registrar empréstimos deve ser inferior a 2 segundos em condições normais de uso.

**RNF005 – Integridade dos Dados**

O sistema deve garantir a integridade dos dados, impedindo que livros que estão emprestados possam ser emprestados novamente antes de serem devolvidos.

**RNF006 – Disponibilidade**

O sistema deve estar disponível para uso 24 horas por dia, 7 dias por semana, com um tempo de inatividade (downtime) máximo de 1% por mês, incluindo atualizações e manutenções.

**RNF007 – Compatibilidade de Navegadores**

O sistema deve ser compatível com os navegadores mais populares (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari), garantindo a mesma experiência de uso em todos eles.

**RNF008 – Escalabilidade**

O sistema deve ser escalável, podendo acomodar um número crescente de livros, clientes e transações (empréstimos) sem perda significativa de desempenho.

**RNF009 – Backup de Dados**

O sistema deve realizar backups automáticos dos dados pelo menos uma vez por dia, para garantir que os registros de livros, clientes e empréstimos não sejam perdidos em caso de falha.

**RNF010 – Suporte a Múltiplos Dispositivos**

O sistema deve ser responsivo, adaptando-se para ser acessado em dispositivos móveis, tablets e desktops.

### **Casos de Uso:**

Identifique os principais casos de uso do sistema.

#### **Cadastrar Livros**

Ator: Dono da biblioteca

Descrição: O sistema deve permitir que o dono da biblioteca cadastre novos livros no sistema, informando o título, autor, ano de publicação e status de empréstimo (emprestado ou disponível).

Requisito relacionado: RF001

#### **Cadastrar Clientes**

Ator: Dono da biblioteca

Descrição: O sistema deve permitir que o dono da biblioteca cadastre novos clientes, armazenando nome, telefone e endereço.

Requisito relacionado: RF002

#### **Registrar Empréstimo de Livro**

Ator: Dono da biblioteca

Descrição: O sistema deve permitir que o dono da biblioteca registre o empréstimo de um livro, associando o livro a um cliente, e registrando a data do empréstimo e a data limite de devolução.

Requisito relacionado: RF003

#### **Buscar Livros por Título, Autor e Ano**

Ator: Dono da biblioteca ou Cliente

Descrição: O sistema deve permitir que o dono da biblioteca ou cliente busque livros filtrando pelo título, autor ou ano de publicação.

Requisito relacionado: RF004

#### **Buscar Clientes por Nome**

Ator: Dono da biblioteca

Descrição: O sistema deve permitir que o dono da biblioteca busque clientes cadastrados pelo nome.

Requisito relacionado: RF005

#### **Listar Livros Emprestados**

Ator: Dono da biblioteca

Descrição: O sistema deve gerar uma lista dos livros que estão emprestados no momento, incluindo o nome do cliente que pegou o livro, o nome do livro, a data de empréstimo e a data limite de devolução.

Requisito relacionado: RF006

### **Listar Clientes Cadastrados**

Ator: Dono da biblioteca

Descrição: O sistema deve listar todos os clientes cadastrados, exibindo nome, telefone e endereço.

Requisito relacionado: RF007

### **Acessar o Sistema via Internet**

Ator: Dono da biblioteca ou Cliente

Descrição: O sistema deve ser acessível de qualquer local com conexão à internet, permitindo que os usuários utilizem as funcionalidades independentemente de sua localização.

Requisito relacionado: RF008

### **Entidades para o Banco de Dados**

Defina as entidades necessárias para armazenar os dados do sistema, incluindo seus atributos.

#### **Livro**

##### **Atributos:**

**id\_livro:** Identificador único do livro (PK)

**titulo:** Título do livro

**autor:** Autor do livro

**ano\_publicacao:** Ano de publicação do livro

**status\_emprestimo:** Status do livro (emprestado ou disponível)

#### **Cliente**

##### **Atributos:**

**id\_cliente:** Identificador único do cliente (PK)

**nome:** Nome do cliente

**email:** Email do cliente

**senha:** Senha do cliente

**telefone:** Telefone do cliente

**endereco:** Endereço do cliente

#### **Emprestimo**

##### **Atributos:**

**id\_emprestimo:** Identificador único do empréstimo (PK)

**id\_livro:** Referência ao livro emprestado (FK para Livro)

**id\_cliente:** Referência ao cliente que pegou o livro (FK para Cliente)

**data\_emprestimo:** Data em que o livro foi emprestado

**data\_limite\_devolucao:** Data limite para devolução do livro

Requisitos Funcionais:

Classifique os requisitos identificados na Parte 1 como funcionais ou não-funcionais.

**Requisitos Funcionais:**

**RF001** – Cadastro de Livros: Permitir o cadastro de livros.

**RF002** – Cadastro de Clientes: Permitir o cadastro de clientes.

**RF003** – Registro de Empréstimos: Permitir registrar empréstimos de livros.

**RF004** – Busca de Livros por Título, Autor e Ano: Permitir busca de livros.

**RF005** – Busca de Clientes por Nome: Permitir busca de clientes.

**RF006** – Listagem de Livros Emprestados: Gerar lista de livros emprestados.

**RF007** – Listagem de Clientes: Listar todos os clientes cadastrados.

**RF008** – Acessibilidade pela Internet: Permitir acesso ao sistema pela internet.

Requisitos Não-Funcionais:

Classifique os requisitos identificados na Parte 1 como funcionais ou não-funcionais.

**Requisitos Não-Funcionais:**

**RNF001** – Usabilidade: O sistema deve ser fácil de usar, com uma interface intuitiva.

**RNF002** – Acessibilidade via Internet: Deve ser acessível remotamente.

**RNF003** – Segurança de Dados: Garantir a segurança de dados, com acesso restrito a funções de administração.

**RNF004** – Tempo de Resposta: O tempo de resposta deve ser inferior a 2 segundos.

**RNF005** – Integridade dos Dados: Garantir que livros emprestados não possam ser emprestados novamente antes da devolução.

**RNF006** – Disponibilidade: Sistema deve estar disponível 24/7, com downtime máximo de 1% por mês.

**RNF007** – Compatibilidade de Navegadores: O sistema deve ser compatível com os principais navegadores.

**RNF008** – Escalabilidade: Sistema deve ser escalável para mais registros e transações.

**RNF009** – Backup de Dados: Backup automático deve ocorrer diariamente.

**RNF010** – Suporte a Múltiplos Dispositivos: O sistema deve ser responsivo.

Regras de Negócio:

Liste as regras de negócio que devem ser seguidas pelo sistema.

**Regras de Negócio:**

**RB001** – Um livro emprestado não pode ser emprestado novamente enquanto não for devolvido.

**RB002** – Clientes devem ter nome, telefone e endereço para serem cadastrados.

**RB003** – Um empréstimo só pode ser registrado se o cliente estiver cadastrado.

**RB004** – O dono da biblioteca tem acesso exclusivo ao cadastro e edição de livros, clientes e empréstimos.

**RB005** – O status do livro deve ser alterado automaticamente para "disponível" ou "emprestado" com base no registro ou devolução do empréstimo.

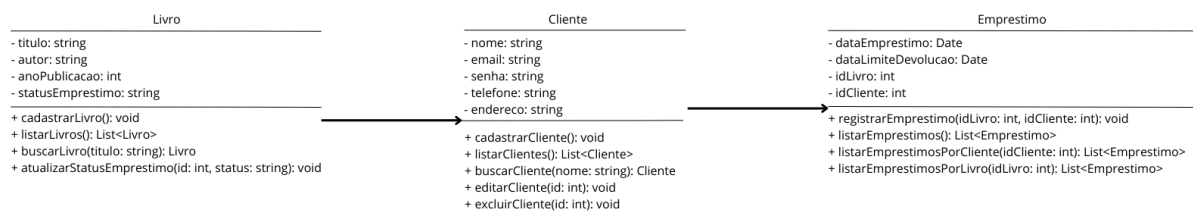
**RB006** – Um cliente não pode pegar mais de um exemplar do mesmo livro simultaneamente.

**RB007** – Apenas livros disponíveis podem ser emprestados.

### Parte 3: Modelagem com UML

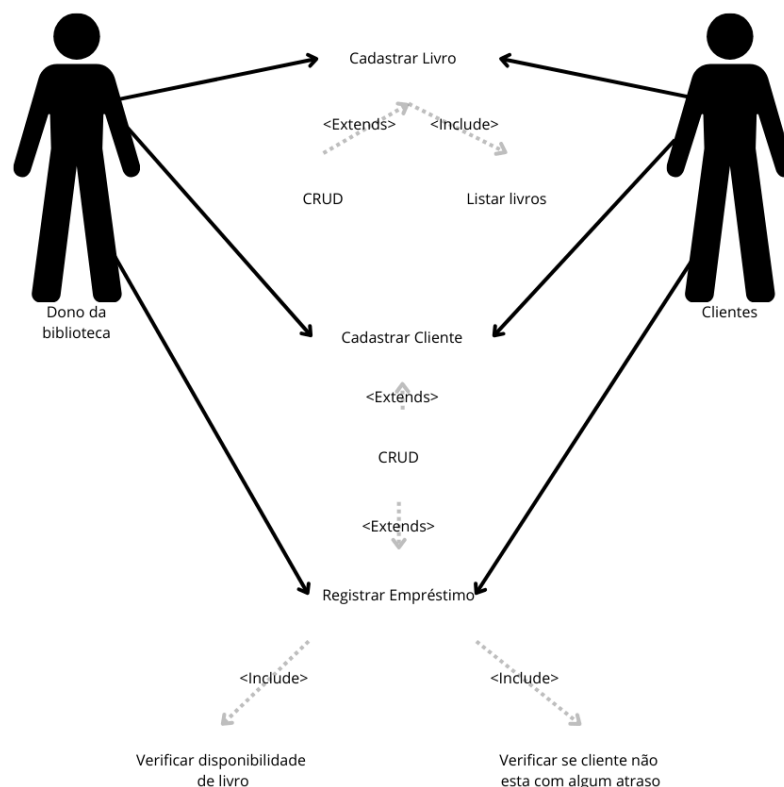
#### Diagrama de Classe:

Crie um diagrama de classe representando as entidades e seus relacionamentos.



#### Diagrama de Casos de Uso:

Crie um diagrama de casos de uso representando os atores e os casos de uso do sistema.



## Parte 4: Escolha da Arquitetura

### Arquitetura:

Com base nos requisitos do sistema, escolha a arquitetura mais adequada: standalone, web/cloud ou mobile. Justifique sua escolha.

### **Arquitetura Web/Cloud**

A escolha de uma arquitetura web/cloud para o sistema se justifica pela necessidade de acessibilidade remota, conforme os requisitos funcionais e não-funcionais. Isso significa que o sistema estará disponível de qualquer lugar com conexão à internet, o que atende diretamente ao RF008 e ao RNF002.

**React:** Utilizando React para a camada de interface do usuário (frontend), temos uma biblioteca moderna, eficiente e amplamente adotada para criar interfaces dinâmicas e responsivas. Isso atende ao requisito de usabilidade (RNF001), oferecendo uma interface amigável e fácil de usar.

**Node.js:** No backend, o Node.js é utilizado para implementar a API REST que fornecerá os serviços necessários ao sistema, como cadastro de livros, clientes, e registro de empréstimos. O Node.js permite escalabilidade e boa performance, além de facilitar o desenvolvimento com JavaScript tanto no frontend quanto no backend.

**Firebase:** Para o banco de dados e autenticação, a escolha de Firebase traz benefícios como facilidade de integração, escalabilidade e suporte a operações em tempo real. Além disso, o Firebase oferece ferramentas de backup e segurança, alinhando-se ao RNF003 (segurança dos dados) e RNF009 (backup de dados).

### **Arquitetura Mobile**

A escolha de desenvolver a versão mobile do sistema usando React Native segue a lógica de reutilização de código e compatibilidade com várias plataformas, atendendo ao RNF010 (suporte a múltiplos dispositivos).

**React Native:** Com essa tecnologia, é possível criar uma aplicação mobile nativa para iOS e Android, mantendo uma experiência de uso consistente. React Native permite compartilhar a maior parte do código com o frontend web, garantindo eficiência no desenvolvimento e mantendo a mesma interface intuitiva, conforme exigido pelo RNF001 (usabilidade).

### **Justificativa para a Arquitetura Web e Mobile**

**Acessibilidade e Usabilidade:** O sistema poderá ser acessado de qualquer lugar e de qualquer dispositivo (desktop, tablet, mobile), garantindo flexibilidade para o dono da biblioteca e seus clientes, conforme os requisitos funcionais e não-funcionais descritos (RF008 e RNF002).

**Escalabilidade e Performance:** A combinação de Node.js e Firebase oferece uma solução escalável que pode crescer com o número de usuários, clientes, livros e



empréstimos (RNF008). A performance em tempo real e a baixa latência no acesso e registro de dados proporcionam uma experiência de uso eficiente (RNF004).

**Segurança:** A integração de Firebase com sua autenticação e segurança de dados garante que apenas usuários autorizados possam realizar as operações de cadastro, empréstimo e devolução (RNF003).

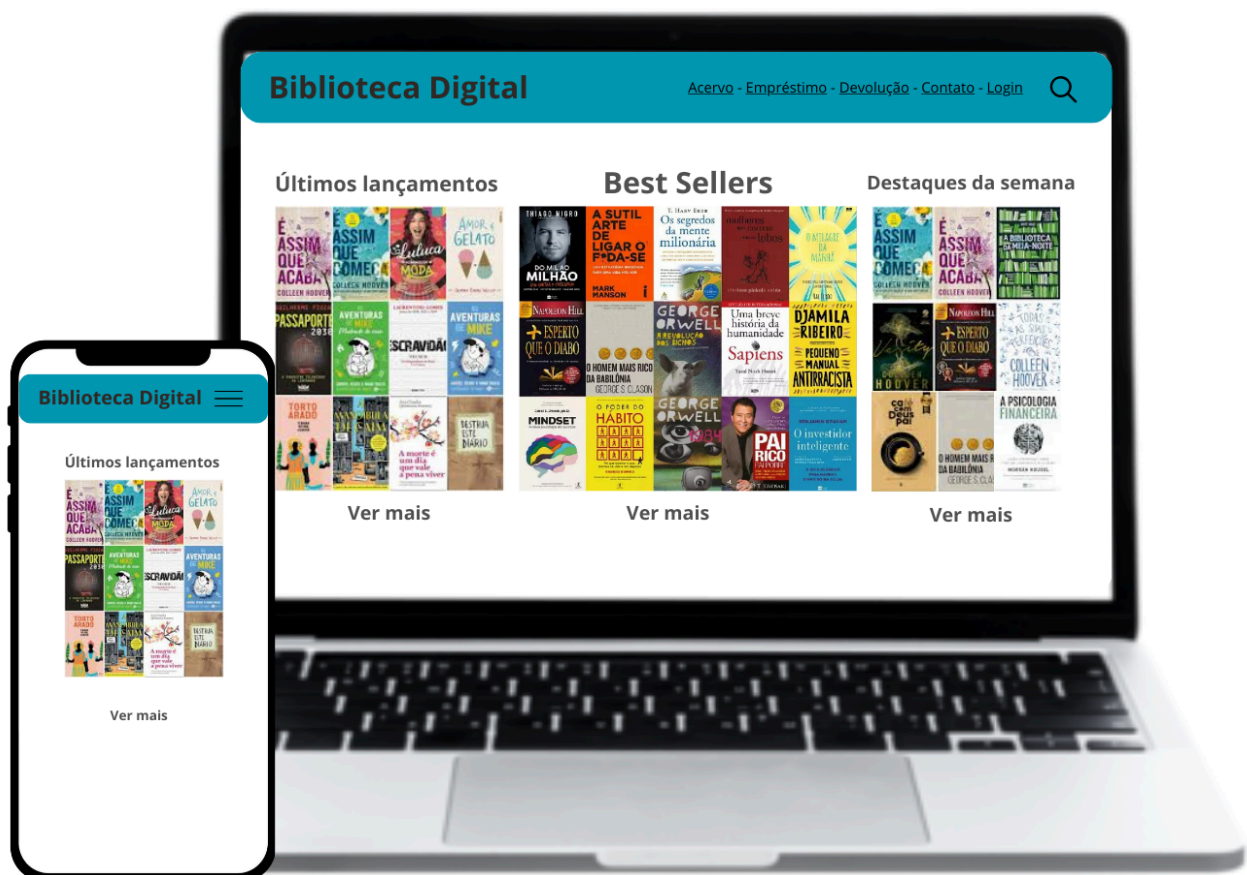
**Compatibilidade e Experiência Consistente:** Com React e React Native, o sistema garante uma experiência de uso semelhante em navegadores e dispositivos móveis, além de garantir compatibilidade com os principais navegadores (RNF007 e RNF010).

## Parte 5: Construção de Protótipos

### Protótipos:

Crie protótipos de baixa ou alta fidelidade, representando o layout e as funcionalidades do sistema.

### Protótipos:







## Contato

### Formulário de Contato

Nome:

Email:

Mensagem:

Enviar



## Login

Seu e-mail

contato@hmlcsapro.com

Sua senha

1234

☐ Manter-me logado

Logar

Ainda não tem conta?

[Cadastre-se](#)



## Parte 6: Consultas SQL

### Consultas:

Escreva consultas SQL para realizar as seguintes operações no banco de dados:

**1. Listar todos os livros emprestados, com a informação de quem pegou emprestado.**

```
SELECT
    l.titulo AS "Título do Livro",
    l.autor AS "Autor",
    c.nome AS "Nome do Cliente",
    e.data_emprestimo AS "Data de Empréstimo",
    e.data_limite_devolucao AS "Data Limite para Devolução"
FROM
    Emprestimo e
JOIN
    Livro l ON e.id_livro = l.id_livro
JOIN
    Cliente c ON e.id_cliente = c.id_cliente
WHERE
    l.status_emprestimo = 'emprestado';
```

## 2. Buscar livros por título.

```
SELECT
    id_livro,
    titulo,
    autor,
    ano_publicacao,
    status_emprestimo
FROM
    Livro
WHERE
    titulo LIKE '%[livroxxx]%';
```

## 3. Listar os livros emprestados para uma determinada pessoa.

```
SELECT
    l.titulo AS "Título do Livro",
    l.autor AS "Autor",
    e.data_emprestimo AS "Data de Empréstimo",
    e.data_limite_devolucao AS "Data Limite para Devolução"
FROM
    Emprestimo e
JOIN
    Livro l ON e.id_livro = l.id_livro
JOIN
    Cliente c ON e.id_cliente = c.id_cliente
WHERE
    c.nome = '[clientexxx]';
```

## 4. Listar todos os clientes cadastrados.

```
SELECT
    id_cliente,
    nome,
    email,
    telefone,
    endereco
FROM
    Cliente;
```

## 5. Buscar clientes por nome.

```
SELECT
    id_cliente,
    nome,
    email,
    telefone,
    endereco
FROM
    Cliente
WHERE
    nome LIKE '%[parte_do_nome]%';
```

Observação: A complexidade das consultas SQL e dos diagramas UML foram intencionalmente mantidas baixa. O foco principal é a compreensão dos conceitos de modelagem e sua aplicação prática em um contexto um pouco mais elaborado.