**SENAI ETTORE ZANINI**

**TÉCNICO DESENVOLVIMENTO DE SISTEMAS**

**Luís Otávio Bordin Pereira**

**Spacebook**Gerenciamento   
de  
Biblioteca

**Sertãozinho  
2024**

LISTA DE ILUSTRAÇÕES SE HOUVER (GERAR AUTOMATICO)

TABELAS SE HOUVER (GERAR AUTOMATICO)

**SUMÁRIO**

**1 INTRODUÇÃO**

A Spacebook, biblioteca localizada em uma escola, sempre foi um ponto de encontro para alunos ávidos por conhecimento. No entanto, com o tempo, a equipe começou a enfrentar grandes desafios na administração do acervo. Os empréstimos de livros muitas vezes eram mal controlados, resultando em devoluções atrasadas e extravios. Além disso, encontrar livros específicos em meio a tantas prateleiras tornou-se uma tarefa difícil tanto para os alunos quanto para os bibliotecários.

Diante desses problemas, surgiu a necessidade de criar um sistema eficiente para gerenciar melhor o acervo, controlar os empréstimos e facilitar a busca por títulos. Esse sistema não só trouxe organização à biblioteca, mas também transformou a experiência de todos, tornando o processo de pesquisa e empréstimo de livros mais ágil e eficiente.

**2 REQUISITOS DO SISTEMA**

Na aplicação Spacebook os Requisitos de sistema são as especificações técnicas e funcionais que definem o que o sistema deve atender para ser desenvolvido e funcionar corretamente. Esses requisitos servem como base para o desenvolvimento e implementação da Spacebook, garantindo que todas as expectativas e necessidades sejam contempladas no produto final.

2.1 Requisitos funcionais

Os requisitos funcionais do sistema Spacebook estão identificados na Tabela 1, os requisitos funcionais são as funcionalidades especificas e que são necessária para um funcionamento correto e garantindo o atendimento das expectativas e necessidades

|  |  |
| --- | --- |
| Requisito Funcional | |
| RF001 | Cadastrar Livro |
| RF002 | Editar Livro |
| RF003 | Excluir Livro |
| RF004 | Cadastrar Aluno |
| RF005 | Editar Aluno |
| RF006 | Remover Aluno |
| RF007 | Empréstimo Livro |
| RF008 | Registrar Devolução |
| RF009 | Consultar Empréstimo |

Tabela 1 requisitos funcionais

**2.2.1 RF001 – Cadastrar Livro**

Descrição: O sistema deve permitir que o usuário cadastre novos livros no acervo da biblioteca.

Prioridade: Alta

Entrada:

* Título do livro (obrigatório).
* Autor(es) (obrigatório).
* Editora (obrigatório).
* Ano de publicação (não obrigatório).
* Isbn (não obrigatório).
* Quantidade total de livros presentes na biblioteca (obrigatório).
* Quantidade de livros disponíveis (obrigatório).
* Valor aquisição (não obrigatório).
* Status do empréstimo (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

Mensagem de sucesso ao cadastrar o livro.

Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O livro cadastrado deve ser exibido na lista de acervo disponível.

**2.2.2 RF002 – Editar Livro**

Descrição: O sistema deve permitir que o usuário edite as informações de um livro previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do livro a ser editado (ID ou ISBN).
* Campos a serem editados (autor(es),editora, ano de publicação, isbn, quantidade total, quantidade disponível, valor aquisição, status do livro ).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de acervo e nos registros de empréstimos associados ao livro.

**2.2.3 RF003 – Excluir Livro**

Descrição: O sistema deve permitir que o usuário exclua um livro do acervo.

Prioridade: Média

Entrada:

* Identificação do livro a ser excluído (ID).

Processamento:

* O sistema deve verificar se há empréstimos ativos ou pendentes associados ao livro.
* Se houver empréstimos associados, o sistema deve impedir a exclusão e informar o usuário
* Saída:

Mensagem de sucesso ao excluir o livro.

Mensagem de erro caso existam empréstimos associados impedindo a exclusão.

Pós-condição:

O livro deve ser removido da lista de acervo e não deve aparecer nas buscas e consultas.

**2.2.1 RF004 – Cadastrar Aluno**

Descrição: O sistema deve permitir que o usuário cadastre novos alunos no acervo da biblioteca.

Prioridade: Alta

Entrada:

* RA (obrigatório).
* Nome (obrigatório).
* Sobrenome (obrigatório).
* Data de nascimento (obrigatório).
* Endereço (não obrigatório).
* Email (não obrigatório).
* Celular (obrigatório).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao cadastrar o livro.
* Mensagem de erro em caso de campos obrigatórios não preenchidos ou duplicidade de cadastro.

Pós-condição:

* O livro cadastrado deve ser exibido na lista de acervo disponível.

**2.2.5 RF005 – Editar Aluno**

Descrição: O sistema deve permitir que o usuário edite as informações de um aluno previamente cadastrado.

Prioridade: Alta

Entrada:

* Identificação do aluno a ser editado (ID ou RA).
* Campos a serem editados (Nome, Sobrenome, Data de nascimento, Endereço, Email, Celular).

Processamento:

* O sistema deve validar os campos obrigatórios.

Saída:

* Mensagem de sucesso ao salvar as alterações.
* Mensagem de erro caso haja duplicidade ou campos obrigatórios não preenchidos.

Pós-condição:

* As alterações devem ser refletidas na lista de acervo e nos registros de empréstimos associados ao livro.

**2.2.6 RF006 – Remover Aluno**

Descrição: O sistema deve permitir que o usuário exclua um aluno do acervo.

Prioridade: Média

Entrada:

* Identificação do aluno a ser excluído (RA ou ID).

Processamento:

* O sistema deve verificar se há empréstimos ativos ou pendentes associados ao aluno.
* Se houver empréstimos associados a esse aluno, o sistema deve impedir a exclusão e informar o usuário.

Saída:

* Mensagem de sucesso ao excluir o aluno.
* Mensagem de erro caso existam empréstimos associados impedindo a exclusão.

Pós-condição:

* O aluno deve ser removido da lista de acervo e não deve aparecer nas buscas e consultas.

**RF007 – Empréstimo de Livro**

Descrição: O sistema deve permitir que o usuário registre o empréstimo de um ou mais livros para um aluno cadastrado.

Prioridade: Alta

Entrada:

* Identificação do aluno (número de matrícula).
* Identificação do(s) livro(s) a ser(em) emprestado(s) (ID do livro).
* Data de início do empréstimo (obrigatória).
* Data prevista de devolução.

Processamento:

* O sistema deve verificar a disponibilidade dos livros no acervo.
* O sistema deve reduzir a quantidade disponível do livro emprestado.
* O sistema deve associar o empréstimo ao aluno e ao(s) livro(s) selecionado(s).

Saída:

* Mensagem de sucesso ao registrar o empréstimo.
* Mensagem de erro caso algum dos livros não esteja disponível.

Pós-condição:

* O empréstimo deve ser registrado com status "Ativo".
* A quantidade disponível do livro deve ser atualizada no sistema.

**3 DIAGRAMAS DO SISTEMA**

3.1 Diagrama de Classe

O diagrama de classe é fundamental para modelar a estrutura do sistema da spacebook orientados a objetos, representando as classes, atributos e seus relacionamentos. Ele auxilia o programador a entender a organização do sistema, proporcionando clareza no desenvolvimento. Isso resulta em um design mais eficiente e na correta implementação dos conceitos planejados. Podemos observar na figura 1

3.2 Banco de Dados

O PostgreSQL é um dos sistemas de gerenciamento de banco de dados relacionais mais avançados e robustos disponíveis no mercado, amplamente utilizado por sua confiabilidade, desempenho e compatibilidade com padrões SQL. Ele oferece suporte a uma vasta gama de tipos de dados, índices e extensões, tornando-o extremamente versátil para diferentes aplicações. Uma ferramenta essencial para trabalhar com o PostgreSQL é o pgAdmin, uma interface gráfica que facilita a administração e o monitoramento de bancos de dados. Com ele, é possível criar e gerenciar esquemas, realizar consultas SQL, monitorar desempenho, entre muitas outras tarefas, tudo de forma intuitiva e acessível, permitindo que tanto desenvolvedores quanto administradores trabalhem de maneira mais eficiente.

O Diagrama Entidade-Relacionamento (DER) é uma representação visual que orienta a modelagem de dados, facilitando o entendimento das relações entre diferentes entidades em um sistema. Ele destaca atributos, entidades e os relacionamentos entre elas, sendo um guia essencial no planejamento e construção de bases de dados estruturadas. A figura 2 inserida ilustra como as entidades e seus relacionamentos podem ser dispostos, ajudando a visualizar, por exemplo, a forma como diferentes tabelas em um banco de dados PostgreSQL se conectam e interagem, refletindo diretamente na implementação através de ferramentas como o pgAdmin.

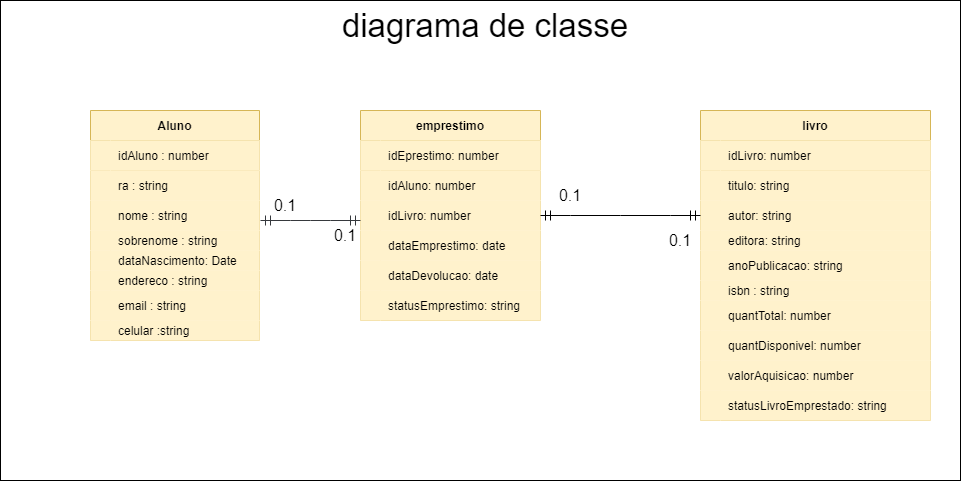
Scripts desempenham um papel fundamental no desenvolvimento e na administração de sistemas de banco de dados, como o PostgreSQL, sendo utilizados para automatizar tarefas, executar consultas, criar e modificar estruturas de banco de dados, e muito mais.

figura 1 diagrama de classe spacebook

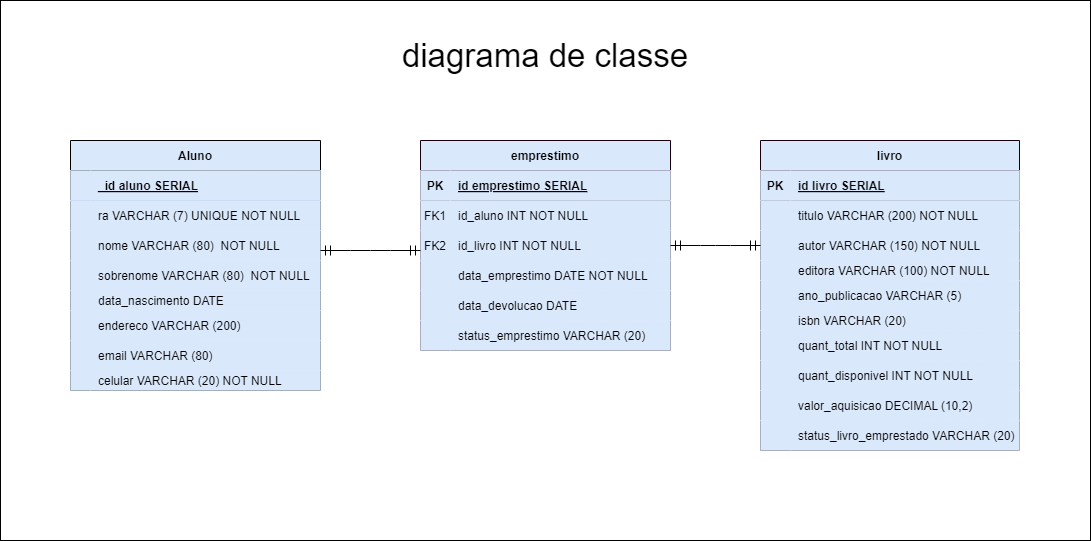


figura 2 DER spacebook

**4 ROTAS DA APLICAÇÃO – BACK-END**

Uma rota no backend é basicamente um endereço (uma URL) que o servidor usa para identificar o que precisa ser feito quando o usuário acessa determinada parte da aplicação. Quando falamos de MVC (Model-View-Controller) e CRUD (Create, Read, Update, Delete), a rota ajuda a conectar o usuário com essas operações em diferentes partes do sistema: no banco de dados, na interface e na lógica de negócios. As rotas tornam-se pontos de acesso para interagir com o banco de dados e manipular dados na interface de forma simples e organizada, usando a estrutura MVC para dividir as responsabilidades e facilitar a manutenção do sistema.

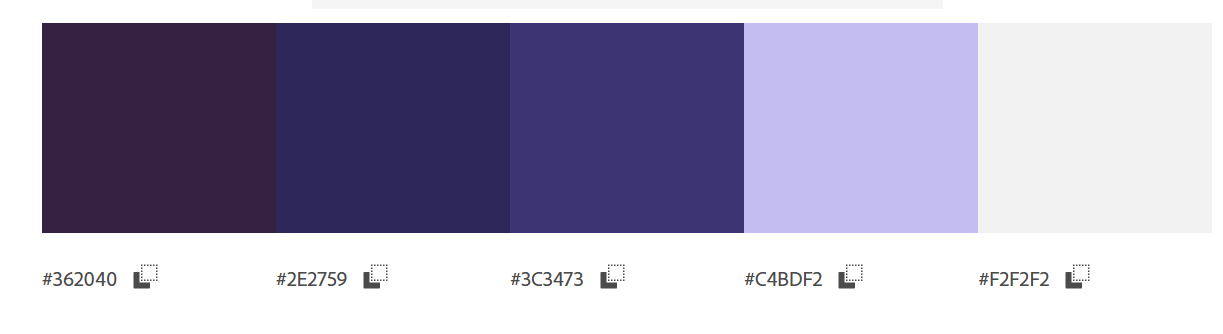
|  |  |  |
| --- | --- | --- |
| get | /lista/livros | Rota para listar os livros |
| post | /novo/livro | Rota para cadastrar novo livro |
| get | /lista/alunos | Rota para listar alunos |
| post | /novo/aluno | Rota para cadastrar novo aluno |
| get | /lista/emprestimo | Rota para listar empréstimos |
| post | /novo/emprestimo | Rota para cadastrar um novo emprestimo |

5 INTERFACE E EXPERIÊNCIA DO USUÁRIO

O ponto de interação entre o usuário e o sistema é uma interface, seja ele um site, aplicativo, software ou dispositivo. É composta por todos os elementos visuais, interativos e funcionais que permitem ao usuário comunicar-se com a tecnologia e realizar ações. Exemplos de interfaces incluem botões, menus, formulários, janelas e ícones, assim como sons ou até mesmo gestos em dispositivos móveis.

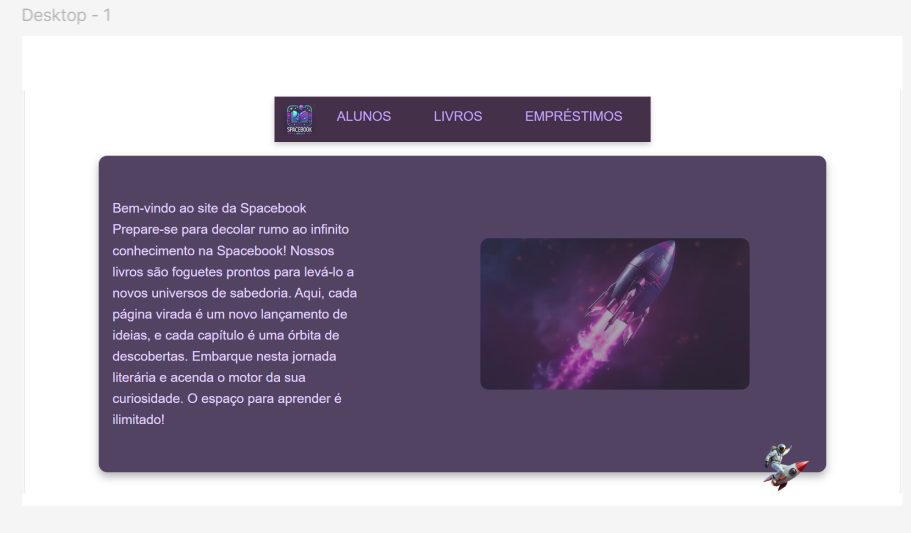
5.2 Paleta de Cores

O nome da biblioteca é um trocadilho com a palavra space que significa espaço e a cor que faz mais sentido é o roxo assim usando também tons que se aproximam do roxo, pois além de combinar com o tema galáxia na psicologia das cores, roxo, malva, violeta e lilás simbolizam a fantasia (da bruxa a fada), a busca anímica, a vontade de tornar possível o impossível assim como descobrir o espaço inteiro de conhecimento já que é tão vasto. O roxo, por exemplo, ativa o lobo frontal, área responsável pelo planejamento de ações e imaginação, como atividades de criação.



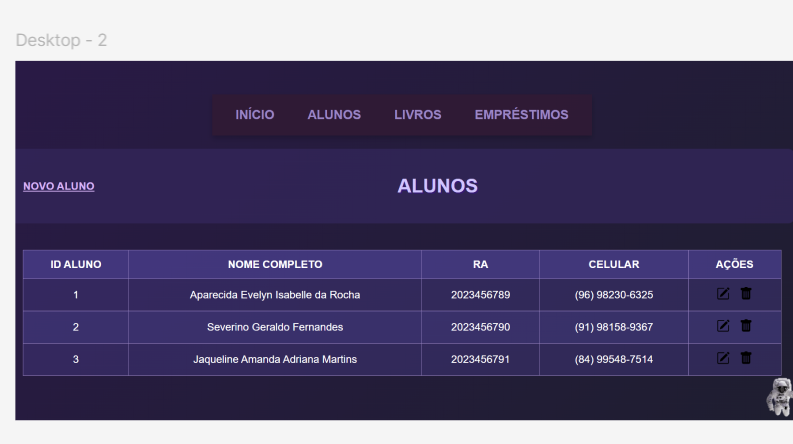
5.3 Mockup

Essa parte da página é o início onde apresentamos a proposta da pagina



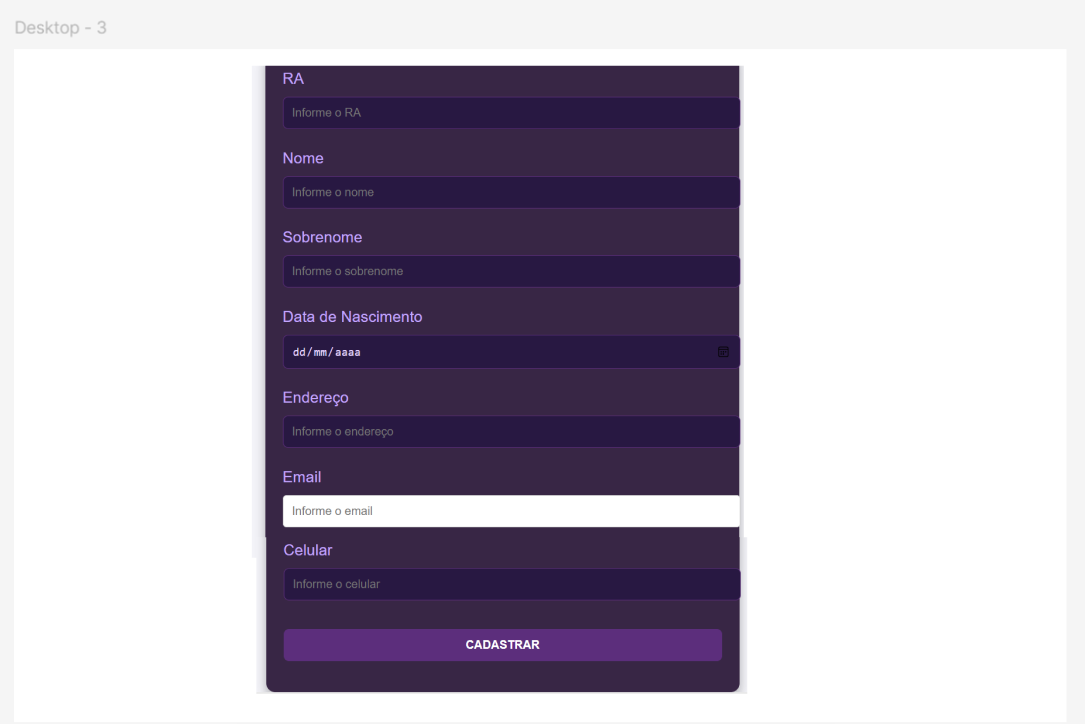
interface 1

Nesta segunda interface o sistema abre quando se clica no botão aluno e mostra as listas de alunos já cadastrados e no canto superior tem um botão escrito novo aluno que encaminha para outra interface.



interface 2

após clicar em novo aluno no canto superior da tela você entra na 3 interface onde existe um formulário com dados para preencher e cadastrar um novo aluno



interface 3

Voltando para a página inicial e clicando no botão livros você é encaminhado para uma interface que lista os livros cadastrados e que fazem parte do acervo



interface 4

Seguindo a lógica do sistema clicando no botão de empréstimos podemos ver os empréstimos já cadastrados.



interface 5

BIBLIOGRAFIA

Link das documentações utilizadas e livros consultados

1. Psicologia das Cores

- Psicologia das Cores - Como as cores afetam as emoções e o comportamento:

https://www.99designs.com.br/blog/tips/psicologia-das-cores/

- Artigos sobre o impacto das cores na UX e UI:

https://uxdesign.cc/color-psychology-in-ui-design-6f2360ef6c39

2. Documentações de CSS e HTML

- W3Schools - Referência completa de CSS e HTML:

https://www.w3schools.com/

- MDN Web Docs - Documentação oficial de CSS:

https://developer.mozilla.org/pt-BR/docs/Web/CSS

3. Ferramentas e Boas Práticas de Desenvolvimento

- GitHub - Para controle de versão e colaboração:

https://github.com/

- Visual Studio Code - Editor de código recomendado:

https://code.visualstudio.com/

4. Contribuição do ChatGPT

O ChatGPT foi utilizado como uma ferramenta de suporte durante o desenvolvimento do projeto. Sua capacidade de responder dúvidas técnicas e fornecer exemplos foi essencial para resolver problemas e otimizar o código.

- Link para interações com o ChatGPT:

https://openai.com/chatgpt