

**CENTRO PAULA SOUZA**  
**Etec PROFESSOR ADHEMAR BATISTA HEMÉRITAS**  
**Curso Técnico em Programação de Jogos Digitais**

**Arthur Gonçalves Andrade**  
**Isabella Florentino Nunes**  
**Guilherme Ferreira Calixto**  
**Mônica Sousa Carvalho**

**SISTEMA DE ADMINISTRAÇÃO PARA A BIBLIOTECA**

**SÃO PAULO**  
**2024**

**Arthur Gonçalves Andrade**  
**Isabella Florentino Nunes**  
**Guilherme Ferreira Calixto**  
**Mônica Sousa Carvalho**

## **SISTEMA DE ADMINISTRAÇÃO PARA A BIBLIOTECA**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Programação de Jogos Digitais da Etec Prof. Adhemar Batista Heméritas, orientado pelo Prof. Alexandre Aguiar como requisito para obtenção do título de Curso Técnico em Programação de Jogos Digitais.

**SÃO PAULO**  
**2024**

## **RESUMO**

O sistema de gestão de empréstimos nas bibliotecas de escolas públicas enfrenta desafios devido ao uso de planilhas, dificultando o controle de prazos e a organização. Para solucionar esses problemas, propomos a criação de uma aplicação web integrada a um banco de dados, que automatiza e otimiza o processo de empréstimos e devoluções de forma eficiente e lúdica. A interface será desenvolvida utilizando HTML e CSS, garantindo uma experiência intuitiva e responsiva. O banco de dados será estruturado com Java, proporcionando uma base sólida para o armazenamento e gerenciamento dos dados. A integração entre o front-end e o banco de dados será realizada por meio de JavaScript, permitindo comunicação eficiente entre as partes. O objetivo é melhorar a organização e o controle dos empréstimos, facilitar o acesso aos livros e oferecer uma experiência agradável aos usuários, contribuindo para a modernização e a eficiência das bibliotecas escolares.

**Palavras-Chave:** Bibliotecas. Banco de Dados. Web.

## **ABSTRACT**

The loan management system in public school libraries faces challenges due to the use of spreadsheets, making it difficult to keep track of deadlines and stay organized. To solve these problems, we propose the creation of a web application integrated with a database, which automates and optimizes the loans and returns process in an efficient and entertaining way. The interface will be developed using HTML and CSS, ensuring an intuitive and responsive experience. The database will be structured with Java, providing a solid basis for data storage and management. Integration between the front-end and the database will be carried out using JavaScript, enabling efficient communication between the parties. The aim is to improve the organization and control of loans, facilitate access to books and offer a pleasant experience to users, contributing to the modernization and efficiency of school libraries.

Keywords: Libraries. Database. Web.

## LISTA DE ABREVIATURAS E SIGLAS

HTML (Linguagem de Marcação de Hipertexto)

CSS (Folhas de Estilo em Cascatas)

JPA (Java Persistence API)

HTTP (Protocolo de Transferência de Hipertexto)

DIVS (Divisões)

CRUD (Criar, Ler, Atualizar e Deletar)

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>8</b>
1.1 Objetivo Geral .....	9
1.2 Objetivos Específicos.....	9
1.3 Justificativa .....	9
1.4 Metodologia .....	10
1.4.1 Pesquisa Online:.....	10
1.4.2 Pesquisa de Campo Qualitativa: .....	11
1.4.3 Público Alvo.....	12
1.4.4 Cronograma .....	12
1.4.5 Pesquisa qualitativa .....	14
<b>2 DESENVOLVIMENTO CRIATIVO.....</b>	<b>16</b>
2.1 Desenvolvimento do mascote .....	16
2.2 Rework.....	16
2.2.1 Situação Inicial .....	16
2.2.2 Processo Criativo e Resultados .....	16
2.2.3 Resultado Final .....	17
2.3 Desenvolvimento do logotipo .....	18
2.3.1 Software Utilizado .....	18
2.3.2 Processo Criativo .....	19
2.4 Estilização do Site.....	19
<b>3 DESENVOLVIMENTO TÉCNICO .....</b>	<b>22</b>
3.1 Abstração dos dados .....	22
3.2 Linguagem de Programação.....	22
3.2.1 Desenvolvimento do Banco de Dados .....	22
3.2.2 Desenvolvimento Web .....	24
3.2.3 Conexão das Tabelas.....	25
3.3 Bibliotecas Utilizadas .....	25

3.3.1	Anotações e classes usadas para definir a entidade: .....	26
3.3.2	Anotações e classes usadas para definir o controlador: .....	26
3.3.3	As anotações e classes usadas para definir o repositório: .	26
3.4	Framework Utilizada .....	27
4	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>29</b>
5	<b>REFERÊNCIAS</b> .....	<b>30</b>

## 1 INTRODUÇÃO

O sistema atual de gestão de empréstimos nas bibliotecas de escolas públicas enfrenta uma série de dificuldades, principalmente devido ao uso de planilhas, que dificultam a organização e o controle de prazos, além de estarem sujeitas a erros. Essa metodologia manual resulta em uma perda de tempo significativa e sobrecarga de trabalho para os funcionários, que precisam lidar com processos repetitivos e imprecisos. Além disso, o excesso de livros compromete o espaço físico da biblioteca e torna a gestão do acervo ainda mais desafiadora.

Para resolver esses problemas, propomos a criação de uma aplicação web integrada com um banco de dados, que automatize o processo de empréstimos e devoluções. Com isso, será possível centralizar todas as informações, controlar os prazos de forma precisa e reduzir a sobrecarga operacional. A automação permitirá alertas e notificações em tempo real, garantindo que os livros sejam devolvidos dentro do prazo e que o atendimento seja mais ágil e eficiente.

A solução também inclui a implementação de elementos lúdicos, como um sistema de pontuação e recompensas, para incentivar o engajamento dos alunos na leitura e na devolução dos livros dentro do prazo estipulado. Os alunos poderão acumular pontos a cada empréstimo e trocar por novos livros ou outros benefícios, tornando o processo mais atrativo e motivador. Essa gamificação ajudará a criar uma cultura de responsabilidade e estimula o envolvimento dos estudantes com a biblioteca.

Além disso, a aplicação ajudará na organização do acervo, permitindo uma gestão mais eficaz do espaço físico disponível. Será possível catalogar e classificar os livros de forma intuitiva, facilitando tanto o processo de empréstimo quanto a localização dos itens. Com isso, esperamos transformar a gestão das bibliotecas escolares, oferecendo um serviço mais ágil, eficiente e motivador para os alunos, ao mesmo tempo em que otimiza a administração dos recursos da biblioteca.

## 1.1 Objetivo Geral

Nossa solução visa aprimorar e otimizar os processos de empréstimos, devoluções e a organização dos livros, proporcionando uma experiência mais eficiente e dinâmica para os usuários.

## 1.2 Objetivos Específicos

- Desenvolver a aplicação web para gestão de empréstimos e devoluções, usando SpringBoot e PostgreSQL/MySQL.
- Implementar controle de prazos e multas, otimizando a gestão das devoluções.
- Criar sistema de pontuação e recompensas para incentivar a devolução no prazo e promover a leitura.
- Integrar a nova plataforma com o sistema existente, importando dados de livros e usuários.
- Garantir usabilidade e acessibilidade da interface para facilitar o uso por alunos, professores e funcionários.
- Realizar testes de usabilidade para validar a eficiência do sistema junto aos funcionários da biblioteca.
- Documentar o código e criar manual de uso, facilitando a administração do sistema pelos funcionários da biblioteca.

## 1.3 Justificativa

Atualmente, a metodologia de gestão de empréstimos na biblioteca é totalmente manual, com o controle sendo realizado por meio de registros em cadernos ou planilhas. Esse processo, além de ser ineficiente, está sujeito a erros humanos, o que impacta diretamente a qualidade do serviço prestado aos usuários. O aumento no volume de empréstimos, sem a devida automação, acentua ainda mais essas falhas, tornando o processo cada vez mais difícil de gerenciar.

A falta de ferramentas tecnológicas gera uma sobrecarga de trabalho para os funcionários, que precisam lidar com tarefas repetitivas, demoradas e propensas a falhas, como a busca manual de informações sobre os livros emprestados e o controle de prazos de devolução.

Além disso, a ausência de automatização resulta em uma significativa perda de tempo e energia, que poderiam ser direcionados para atividades mais estratégicas, como o atendimento ao público ou a organização do acervo. A baixa usabilidade dos sistemas de gestão agrava ainda mais a situação, dificultando a busca de dados precisos e a tomada de decisões rápidas. Esse quadro de ineficiência operacional não só prejudica o desempenho da biblioteca, mas também impacta diretamente a experiência dos usuários, que enfrentam longas esperas e problemas no acompanhamento de seus empréstimos, como a dificuldade para obter informações sobre prazos e disponibilidade de livros.

A falta de um sistema automatizado não só compromete a produtividade interna, como também contribui para um atendimento insatisfatório e frustrante para os usuários, que se veem diante de um processo lento e desorganizado. Dessa forma, a biblioteca perde a oportunidade de se modernizar e oferecer um serviço mais ágil, eficiente e amigável, alinhado às expectativas de seus frequentadores e às exigências de um ambiente cada vez mais digital e conectado. As principais falhas desse modelo atual são a falta de automação, a baixa usabilidade dos sistemas utilizados, a ineficiência na gestão de processos e a sobrecarga de trabalho dos funcionários, o que resulta em uma experiência insatisfatória tanto para a equipe quanto para os usuários.

## **1.4 Metodologia**

A metodologia adotada para o desenvolvimento do projeto envolveu uma abordagem prática e de imersão, combinando o uso de ferramentas tecnológicas e pesquisa qualitativa para entender as necessidades reais do ambiente em que o sistema seria implementado.

### **1.4.1 Pesquisa Online:**

Para o desenvolvimento do front-end do nosso projeto, utilizamos as plataformas HomeHost e Rockcontent, que nos forneceram as ferramentas e recursos necessários para criar uma interface interativa e funcional. Essas plataformas permitiram que focássemos na usabilidade e na estética do projeto, facilitando a construção de uma experiência do usuário intuitiva e agradável.

Além disso, para aprimorar nosso conhecimento técnico, complementamos o aprendizado por meio de cursos online em plataformas especializadas, como Amigoscode e Udemy. Os cursos foram fundamentais, especialmente no que diz respeito à modelagem de banco de dados, capacitando nossa equipe a desenvolver uma estrutura robusta e eficiente para o armazenamento e manipulação das informações necessárias para o projeto.

#### 1.4.2 Pesquisa de Campo Qualitativa:

Realizamos uma pesquisa de campo qualitativa por meio de entrevistas com dois funcionários da biblioteca, Maurício Fonseca da Conceição e Priscilla Martins Diniz. Essas entrevistas nos proporcionaram uma visão aprofundada sobre o funcionamento cotidiano da biblioteca, especialmente no que diz respeito ao processo de empréstimo de livros e materiais.

As conversas com os profissionais nos ajudaram a entender não apenas os procedimentos operacionais, mas também as dificuldades e desafios enfrentados pelos funcionários na gestão dos empréstimos. Essa interação direta com o público-alvo do projeto foi essencial para identificar áreas de melhoria e necessidades específicas, permitindo que adaptássemos nosso sistema para atender de forma mais eficaz às demandas da biblioteca.

A pesquisa qualitativa também nos possibilitou explorar aspectos humanos e contextuais do processo de empréstimo, como as expectativas dos usuários, os fluxos de trabalho internos e as limitações tecnológicas da biblioteca, o que contribuiu para um planejamento mais alinhado com a realidade do ambiente em que nosso sistema será implementado.

### 1.4.3 Público Alvo

Nosso projeto é voltado para a comunidade acadêmica da Etec Adhemar Batista Heméritas, abrangendo os alunos, funcionários e professores que frequentam a biblioteca da instituição. Esse público representa usuários com necessidades distintas, o que torna nosso sistema essencial para otimizar o processo de empréstimo e gestão dos materiais.

**Alunos:** Estudantes de diversos cursos da Etec, que buscam acesso ágil a livros e materiais didáticos para complementar seus estudos e pesquisas. A solução visa oferecer uma experiência mais dinâmica e acessível, permitindo que os alunos encontrem e reservem materiais de forma rápida, facilitando o seu processo de aprendizado e desenvolvimento acadêmico.

**Professores:** Docentes que frequentemente utilizam a biblioteca para buscar referências, livros técnicos e materiais de apoio para suas aulas e projetos de pesquisa. Para este grupo, a nossa plataforma visa não apenas simplificar o empréstimo de materiais, mas também aprimorar o gerenciamento e a organização dos recursos acadêmicos que são essenciais para o desenvolvimento de suas atividades pedagógicas.

**Funcionários:** A equipe da biblioteca, composta por profissionais responsáveis pela organização, controle e manutenção do acervo, que desempenham um papel crucial na gestão dos empréstimos, devoluções e renovação de materiais. A nossa solução se destina a facilitar o dia a dia desses funcionários, proporcionando uma maneira mais eficiente de monitorar e registrar as transações, além de melhorar o atendimento aos usuários da biblioteca.

### 1.4.4 Cronograma

Imagem 1 - Cronograma

Fevereiro		Março		Abril	
Dia	Atividade	Dia	Atividade	Dia	Atividade
21	Formação do grupo de TCC e divisão de tarefas	3	Introdução da documentação do TCC	20	Início do site
				20	Pesquisa de campo com os funcionários da biblioteca
24	Pesquisa de possibilidade de realização	17	Início dos testes para a preparação do desenvolvimento do trabalho		

Fonte: Do próprio autor, 2024.

Imagem 2 – Cronograma

Maio		Junho	
Dia	Atividade	Dia	Atividade
			Início da repaginação do mascote
22	Estruturação do projeto principal e criação da primeira tabela e entidade		
		5	Mudança de cor do site para a paleta de cor da escola
29	Geração do formulário em HTML e CSS para a conexão com o SQL		

Fonte: Do próprio autor, 2024.

Imagem 3 – Cronograma

Julho		Agosto		Setembro	
Dia	Atividade	Dia	Atividade	Dia	Atividade
27	Início da criação dos desenhos do mascote para o site	10	Desenvolvimento do front-end do formulário de cadastro	15	Anexação das imagens do mascote no site
		22	Desenvolvimento do front-end do formulário de consulta		

Fonte: Do próprio autor, 2024.

Imagem 4 - Cronograma

Outubro		Novembro		Dezembro	
Dia	Atividade	Dia	Atividade	Dia	Atividade
20	Finalização do banco de dados	15	Início do desenvolvimento do vídeo de apresentação	5	Finalização da documentação
31	Elaboração de ideias para o vídeo de apresentação	26	Finalização do site		
		27	Finalização do vídeo de apresentação		
		28	Apresentação de TCC		

Fonte: Do próprio autor, 2024

#### 1.4.5 Pesquisa qualitativa

Uma pesquisa qualitativa é um método sistemático de investigação utilizado em diversas áreas acadêmicas e profissionais para compreender fenômenos complexos e multifacetados. Ao contrário da pesquisa quantitativa, que se concentra em números e estatísticas, a pesquisa qualitativa busca explorar significados, experiências e perspectivas por meio de uma análise detalhada e interpretativa dos dados. Utilizando técnicas como entrevistas, observações e análise de documentos, os pesquisadores buscam capturar as experiências humanas.

Durante o primeiro semestre, o grupo desenvolveu uma pesquisa qualitativa, entrevistando a responsável pela biblioteca e professora Priscilla Martins Diniz, e o estagiário da biblioteca, Mauricio Fonseca da Conceição.

Questões abordadas para o ex-funcionário da biblioteca Mauricio:

1. Você acha que um sistema manual de empréstimos e devolução de livros se tornar cansativo ou repetitivo quando é em grande escala?

**Resposta:** Sim, nós não temos muita demanda de empréstimos em alta

escala, porém quando temos se torna cansativo.

2. Você acredita que seria mais fácil se o sistema de empréstimos fosse digital?

**Resposta:** Acredito que sim.

3. Você acredita que o nosso projeto seria viável e poderia otimizar seu tempo de trabalho?

**Resposta:** Sim, acho que ajudaria a termos mais tempo para organizar outros detalhes da biblioteca.

Questões abordadas para a Priscila:

1. Você acha que um sistema manual de empréstimos e devolução de livros se tornar cansativo ou repetitivo quando é em grande escala?

**Resposta:** Sim, o sistema de empréstimo manual quando é em grande escala se torna difícil no controle de devoluções por exemplo, ficar buscando os nomes das pessoas em listas manuais muitas vezes demora.

2. Você acredita que seria mais fácil se o sistema de empréstimos fosse digital?

**Resposta:** Sim, o sistema digitalizado pode facilitar muito a gestão dos empréstimos já que pode se fazer uma pesquisa rápida pelo nome do usuário ou pelo título do livro.

3. Você acredita que o nosso projeto seria viável e poderia otimizar seu tempo de trabalho?

**Resposta:** Sim, acredito que um sistema informatizado será muito útil, não só para gerenciar os empréstimos dos livros como também para aumentar a busca pelos exemplares.

## 2 DESENVOLVIMENTO CRIATIVO

Para garantir que nosso sistema web fosse intuitivo e estilizado, passamos por alguns processos de desenvolvimento criativo como a reformulação do mascote e a criação de um logotipo marcante e impactante, o que representa grande parte da estilização do projeto.

### 2.1 Desenvolvimento do mascote

O desenvolvimento do mascote consistiu em uma reformulação do design para torná-lo mais cativante e representativo da marca. Foram utilizadas formas arredondadas, cores vibrantes e expressões mais alegres para criar uma personalidade acolhedora e energética, fortalecendo a conexão com o público.

### 2.2 Rework

O processo de *rework* do mascote foi guiado por uma transformação profunda que trouxe uma nova vida e personalidade ao personagem, mantendo sua essência, mas adaptando-o ao contexto atual.

#### 2.2.1 Situação Inicial

O mascote original não transmitia carisma ou identidade própria, o que dificultava a conexão emocional com o público. A ausência de cores vibrantes deixava sua aparência apagada e pouco atraente, o que passava uma impressão de melancolia e falta de energia. Além disso, o design sem estilização resultava em um visual pouco impactante e desinteressante.

#### 2.2.2 Processo Criativo e Resultados

A reformulação teve como foco criar um visual moderno, vibrante, acolhedor e que ressoasse com o público-alvo de maneira mais envolvente.

1. **Construção utilizando Formas Definidas e Circulares:**  
A estrutura do mascote foi redesenhada com formas arredondadas e bem definidas. As linhas suaves e circulares foram escolhidas intencionalmente para evocar um sentimento de fofura e proximidade, qualidades essenciais para fortalecer o vínculo com o público.
2. **Paleta de cores contento cores fortes e vivas:**  
Uma das mudanças mais notáveis foi a escolha de uma nova paleta de cores. As cores sem vida foram substituídas por tons fortes e vibrantes, especialmente roxo e amarelo, que são complementares. Essa combinação não apenas destaca o mascote, mas também reflete uma energia positiva e calorosa, tornando-o mais chamativo e agradável aos olhos.
3. **Aparência Alegre e Acolhedora:** Para contrapor a antiga aparência, o mascote recebeu expressões mais animadas e acolhedoras, projetadas para passar alegria e simpatia. A idealização era de um mascote que fosse amigável e acessível, transmitindo confiança e bem-estar.
4. **Criação de Personalidade:** Por fim, um dos objetivos principais do rework foi dar personalidade ao mascote, tornando-o mais do que uma imagem, mas sim um “personagem” com traços e atitudes reconhecíveis. Essa mudança faz com que ele ganhe identidade e fortalece seu papel como um símbolo representativo da marca.

### 2.2.3 Resultado Final

O novo mascote é vibrante, amigável e cheio de personalidade, combinando um visual atraente com um caráter acolhedor. A combinação de formas arredondadas, cores vibrantes e uma expressão positiva trouxe um frescor ao personagem, possibilitando uma conexão emocional mais forte com o público e reforçando os valores da marca que representa, melhorando

extremamente o visual da coruja.

Imagem 5 - Mascote



Fonte: Do próprio autor, 2024

## 2.3 Desenvolvimento do logotipo

O processo criativo do logotipo teve como foco principal destacar os elementos mais marcantes do mascote, para refletir sua personalidade de forma clara e impactante. O objetivo era desenvolver um logotipo facilmente reconhecível, que combinasse a essência do personagem com uma estética moderna e retrô ao mesmo tempo.

### 2.3.1 Software Utilizado

Para o desenvolvimento do logotipo, escolhemos o Adobe Photoshop, um software bastante utilizado para edição de imagens e design gráfico. O Photoshop nos ofereceu diversas ferramentas que facilitaram a criação do nosso logotipo, como o trabalho com camadas, que nos permitiu editar diferentes elementos de forma independente, e as ferramentas de seleção e máscara, que nos ajudaram a definir com precisão as formas. Além disso, recursos como aplicação de gradientes e efeitos de texto, garantiram um melhor controle sobre

o design, tornando possível criarmos um logotipo atraente e profissional.

### 2.3.2 Processo Criativo

O processo criativo do logotipo foi direto e focado, buscando um design robusto e minimalista com uma estética inspirada nos anos 80. A ideia central foi destacar os elementos mais marcantes do nosso mascote: seus olhos e a cartola. Esses elementos foram escolhidos por sua forte identidade visual e por representarem de forma clara a personalidade do personagem. O objetivo foi criar um logotipo simples, mas impactante, que fosse facilmente reconhecível e transmitisse a essência do mascote de maneira moderna, mas com um toque retrô característico da década de 1980.

Imagem 6 - Logotipo



Fonte: Do próprio Autor, 2024.

## 2.4 Estilização do Site

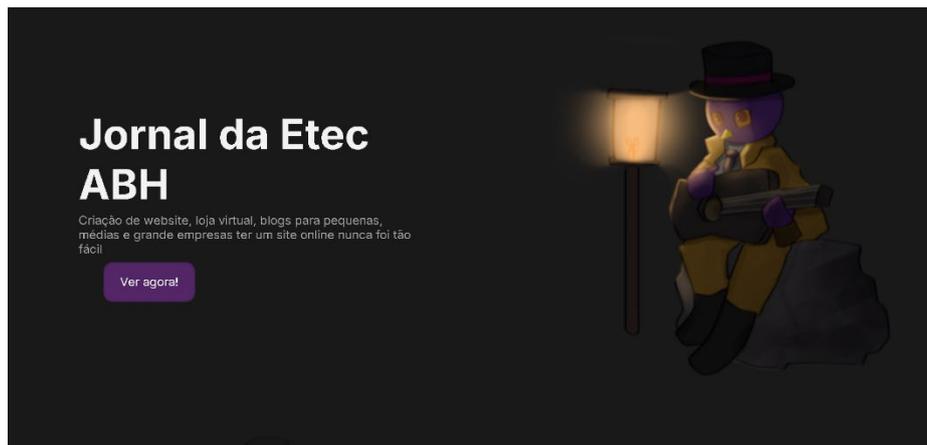
Após definirmos a paleta de cores, redesenharmos o mascote e criarmos o logotipo desenvolvemos outras representações do mascote em diferentes poses e contextos. Essas variações foram criadas para enriquecer a estilização do nosso site, oferecendo flexibilidade visual e garantindo que a identidade do personagem fosse atraente em diversas seções e elementos da interface.

Imagem 7 - Página Inicial



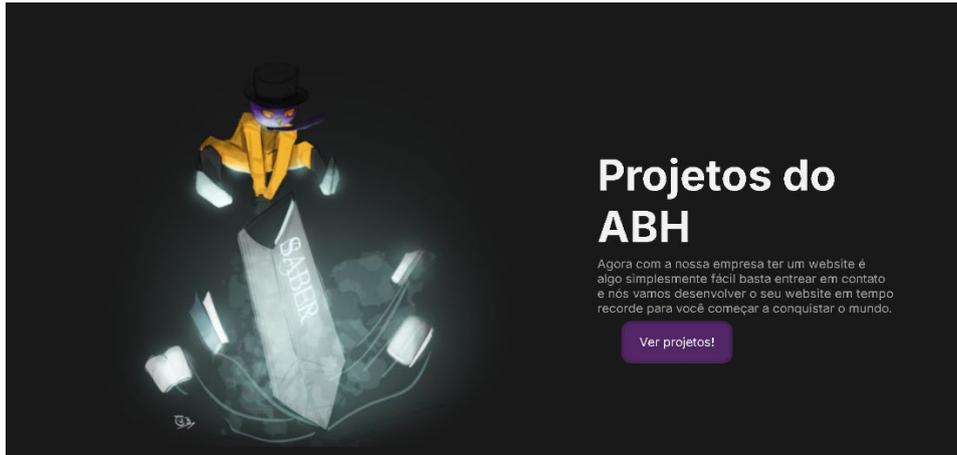
Fonte: Do próprio Autor, 2024.

Imagem 8 - Banco de Dados



Fonte: Do próprio Autor, 2024.

Imagem 9 - Banco de Dados



Fonte: Do próprio Autor, 2024.

### 3 DESENVOLVIMENTO TÉCNICO

O desenvolvimento técnico do projeto envolveu a abstração dos dados da biblioteca usando a ferramenta Diaportable, organizando a estrutura do banco de dados. Utilizamos Java com Spring Boot para o back-end, criando entidades, repositórios e *controllers*, enquanto HTML, CSS e JavaScript foram usados no front-end para estruturar, estilizar e conectar os formulários ao banco de dados. O Spring Boot facilitou a criação de tabelas e a comunicação entre o front-end e o back-end.

#### 3.1 Abstração dos dados

Para melhorar a visualização dos dados que seriam incluídos nas tabelas do banco de dados, foi necessário realizar uma abstração dos processos de cadastro utilizados na biblioteca da instituição. Essa abstração foi realizada por meio da ferramenta Diaportable, que permitiu representar de forma mais clara e estruturada os dados envolvidos no processo de cadastro. Essa abordagem facilitou a organização e o entendimento dos dados, possibilitando a criação de um modelo de banco de dados mais eficiente e alinhado com as necessidades da instituição.

#### 3.2 Linguagem de Programação

Utilizamos no nosso projeto Java para o desenvolvimento do back-end, incluindo entidades JPA, repositórios e controladores, HTML para estruturar a página, CSS para estilizar a interface e garantir o layout responsivo, e JavaScript para conectar o front-end e back-end, permitindo a criação de dados no banco através de requisições HTTP.

##### 3.2.1 Desenvolvimento do Banco de Dados

Para o desenvolvimento do banco de dados do nosso projeto, optamos pela linguagem Java, com a qual já possuímos familiaridade. Utilizamos o

*framework* Spring Boot, que, apesar de compartilhar a mesma sintaxe básica do Java, oferece uma estrutura simplificada e eficiente para o desenvolvimento de aplicações *web*. Para a criação das tabelas no banco de dados, foi necessário implementar três componentes principais: a entidade, o repositório e o *controller*. A entidade definiu a estrutura dos dados a serem armazenados, o repositório gerencia a comunicação com o banco, e o *controller* controla as interações e a lógica de acesso das informações, permitindo a criação, leitura, atualização e exclusão de dados de forma organizada e funcional.

Imagem 11 - Entidades

```
1 package com.example.demo.entities;
2
3
4
5 > import org
6
7 7 usages
8
9 @Entity
10 public class Aluno {
11
12     7 usages
13     @Id
14     @SequenceGenerator
15     (
16         name = "aluno_id",
17         sequenceName = "aluno_id"
18     )
19     @GeneratedValue
20     (
21         generator = "aluno_id",
22         strategy = GenerationType.SEQUENCE
23     )
24     private Long aluno_id;
25     7 usages
26     private String nome;
27     7 usages
28     private String livro;
29     7 usages
30     private Boolean checarDevol;
31     7 usages
32     private String curso;
33     7 usages
34     private String serie;
35     7 usages
36     private String periodo;
37     7 usages
38     private Date data_entrada;
39     7 usages
40     private Date data_devol;
41
42 }
```

Fonte: Do próprio Autor, 2024.

## Imagem 12 - Controllers

```

1 package com.example.demo.controllers;
2
3 > import ...
4
5 no usages
6 @Controller
7 @RequestMapping("/aluno")
8 public class AlunoController {
9
10     @Autowired
11     private AlunoRepository alunoRepository;
12
13     no usages
14     @PostMapping("/addAluno")
15     public String addAluno(@RequestParam String nome,
16                           @RequestParam String livro,
17                           @RequestParam Boolean checarDevol,
18                           @RequestParam String curso,
19                           @RequestParam String serie,
20                           @RequestParam String periodo,
21                           @RequestParam("data_entrada") String dataEntre,
22                           @RequestParam("data_devol") String dataDevol,
23                           Model model) {
24         // Formato da data esperado do formulário (ex: "yyyy-MM-dd")
25         SimpleDateFormat dateFormat = new SimpleDateFormat(pattern: "yyyy-MM-dd");
26
27         Aluno aluno = new Aluno();
28         aluno.setName(nome);
29         aluno.setLivro(livro);
30         aluno.setChecarDevol(checarDevol);
31         aluno.setCurso(curso);
32         aluno.setSerie(serie);
33         aluno.setPeriodo(periodo);
34     }
35 }

```

Fonte: Do próprio Autor, 2024.

## Imagem 13 - Repositórios

```

1 package com.example.demo.repository;
2
3 > import ...
4
5 no usages
6 @Repository
7 public interface AlunoRepository
8     extends JpaRepository<Aluno, Long> {
9 }
10
11

```

Fonte: Do próprio Autor, 2024.

### 3.2.2 Desenvolvimento Web

No desenvolvimento do site, foram utilizadas duas linguagens principais, o HTML e o CSS, cada uma com funções específicas que contribuíram para a construção da página de maneira funcional e visualmente atraente.

O HTML foi utilizado para a definição da estrutura básica do site. Inserimos os elementos essenciais, como divs, links e tabelas, com o objetivo de organizar o conteúdo e facilitar a visualização da disposição da página. Essa estruturação foi fundamental para que tivéssemos uma visão clara de como os componentes da interface seriam organizados e interligados. A partir disso, planejamos e ajustamos a hierarquia das informações e garantimos que o conteúdo estivesse

acessível e funcional.

Depois de definirmos a estrutura, partimos para a estilização, utilizando o CSS. O que foi essencial para personalizar a parte visual do site, nos permitindo adicionar estilos como tamanho e tipo de fontes, cores, espaçamentos, bordas e alinhamentos, e nos proporcionou um maior controle sobre o *layout* responsivo da página. Também foi usado para inserir animações interativas, como as transições de *hover* nos botões e efeitos de interação na navegação. Aplicamos o CSS na criação da barra de navegação, onde foram definidas as cores, o espaçamento entre os itens de menu e as transições suaves durante a interação do usuário.

### 3.2.3 Conexão das Tabelas

A conexão entre as tabelas do banco de dados e os formulários *web* foi realizada utilizando JavaScript. O código responsável por essa conexão foi implementado no *controller* de cada entidade, que contém as informações necessárias, como o endereço do banco de dados, o número da porta e o tipo de operação a ser executada. No caso específico do formulário de cadastro, a operação realizada foi do tipo *POST*, destinada à criação de novos dados no banco. O que nos garantiu a comunicação eficiente entre o front-end (formulários) e o back-end (banco de dados), permitindo o envio e armazenamento adequado das informações inseridas pelos usuários.

## 3.3 Bibliotecas Utilizadas

Jakarta Persistence (JPA):

- jakarta.persistence-api (para as anotações de entidade e mapeamento)

Spring Framework:

- spring-context (para a injeção de dependências)

- spring-web (para o Spring MVC e a manipulação de requisições HTTP)

Spring Data JPA:

- spring-data-jpa (para repositórios baseados em JPA)

Java Standard Library:

java.util (para manipulação de coleções e datas)

java.text (para formatação e parsing de datas)

### 3.3.1 Anotações e classes usadas para definir a entidade:

jakarta.persistence.Entity;

jakarta.persistence.GeneratedValue;

jakarta.persistence.GenerationType;

jakarta.persistence.Id;

jakarta.persistence.SequenceGenerator;

java.util.Date; (Biblioteca padrão do Java).

java.util.Objects; (Biblioteca padrão do Java).

Essas bibliotecas fazem parte da Jakarta Persistence API.

### 3.3.2 Anotações e classes usadas para definir o controlador:

com.example.demo.entities.Aluno; (Referência à classe de entidade "Aluno" definida no seu projeto).

com.example.demo.repository.AlunoRepository; (Referência à interface de repositório JPA para a entidade "Aluno").

org.springframework.beans.factory.annotation.Autowired;

org.springframework.stereotype.Controller;

org.springframework.ui.Model;

org.springframework.web.bind.annotation.PostMapping;

org.springframework.web.bind.annotation.RequestMapping;

org.springframework.web.bind.annotation.RequestParam;

java.text.ParseException; (Biblioteca padrão do Java).

java.text.SimpleDateFormat; (Biblioteca padrão do Java).

Essas bibliotecas são parte do Spring Framework.

### 3.3.3 As anotações e classes usadas para definir o repositório:

org.springframework.data.jpa.repository.JpaRepository;

`org.springframework.stereotype.Repository;`

Essas bibliotecas são parte do Spring Data JPA, que facilita a interação com bancos de dados relacionais no contexto do Spring Framework.

### 3.4 Framework Utilizada

Para o desenvolvimento do nosso projeto utilizamos o Spring Boot, que facilitou a criação e estruturação da aplicação de forma rápida e eficiente. Ele é um framework baseado em Java, muito utilizado para o desenvolvimento de aplicações web, e foi crucial para a organização e escalabilidade do nosso sistema através das funcionalidades listadas abaixo.

#### 1. Estrutura Simplificada:

O Spring Boot nos forneceu uma estrutura de desenvolvimento pronta para uso, que nos permitiu focar na lógica de negócios sem precisar configurar manualmente componentes como servidores ou dependências.

#### 2. Criação e Manipulação de Entidades:

O Spring Boot utilizou sua integração com o Spring Data JPA, o que permitiu que as entidades fossem mapeadas diretamente para as tabelas do banco de dados. Simplificando a criação das tabelas, já que as classes Java funcionaram como representações das tabelas no banco de dados.

#### 3. Repositórios:

O Spring Data também auxiliou na criação dos repositórios, que são responsáveis por realizar as operações de *CRUD* no banco de dados de forma automática e simplificada. Através da criação de interfaces de repositório, pudemos realizar as operações sem a necessidade de escrever SQL manualmente, deixando o código mais limpo e organizado.

#### 4. Controllers e Manipulação de Requisições:

Também nos proporcionou a implementação fácil dos controllers, que são responsáveis por gerenciar as requisições HTTP e orquestrar a interação entre

o frontend (formulários) e o backend (banco de dados). No formulário de cadastro, o Spring Boot gerenciou a operação do tipo POST para criar dados, associando diretamente os dados recebidos via JavaScript com o banco de dados.

O Spring Boot nos forneceu uma plataforma robusta e de fácil configuração, o que otimizou o desenvolvimento do back-end e permitiu que a comunicação entre o front-end e o banco de dados fosse eficiente e tornando o processo mais ágil e organizado.

#### **4 CONSIDERAÇÕES FINAIS**

Após identificarmos falhas significativas no gerenciamento das bibliotecas das escolas públicas brasileiras, desenvolvemos um sistema web integrado a um banco de dados com o objetivo de minimizar os erros nos processos de empréstimos, devoluções e cadastramentos de livros. Além disso, buscamos otimizar o tempo de execução dessas tarefas e garantir uma gestão mais eficiente dos recursos da biblioteca. Para tornar a utilização do sistema mais agradável e intuitiva, investimos na criação de uma interface responsiva e de fácil navegação, incorporando elementos gráficos que tornam a experiência mais fluida e visualmente atrativa. Essa abordagem não só acelera os processos dentro das bibliotecas das escolas públicas, mas também contribui para um desempenho mais eficaz dos funcionários, melhorando a qualidade do serviço prestado aos alunos e à comunidade escolar.

## 5 REFERÊNCIAS

AmigosCode. SpringBoot Tutorial| Full Course. Youtube, 2023. Disponível em: [[https://youtu.be/9SGDpanrc8U?si=\\_VZeugVe-AVjwtYh](https://youtu.be/9SGDpanrc8U?si=_VZeugVe-AVjwtYh)]. Acesso em: 13 de maio de 2024.

AmigosCode. SpringBoot – Learn Spring Boot 3. Youtube, 2022. Disponível em: [<https://youtu.be/-mwpoE0x0JQ?si=AsbK9uDEfFWISpg6>]. Acesso em: 17 de junho de 2024.

Lourenço, Ivan. Aprenda HTML e CSS e Crie o seu Portifólio Online. Udemy, 2018. Disponível em: [<https://www.udemy.com/share/101zna/>]. Acesso em: 20 de agosto de 2024.