

<h1>Desde o Zero</h1>



Como Criamos um Sistema de Acesso Escolar em 2 Dias

Sumário

- 1.0 Introdução..... 1**
 - 1.1 Contexto do projeto..... 1
 - 1.2 Objetivo do projeto 1
- 2.0 Planejamento e Divisão de Tarefas 2**
 - 2.1 Equipe e responsabilidades 2
 - 2.2 Cronograma e tempo disponível 2
- 3.0 Importação de Dados 2**
 - 3.1 Preparação dos arquivos CSV 3
 - 3.2 Estrutura do banco de dados MySQL 3
- 4.0 Desenvolvimento do Projeto 4**
- 5.0 Explicação Técnica do Código 4**
 - 5.1 Estrutura HTML 5
 - 5.2 Lógica em JavaScript 6
 - 5.3 Conexão e requisições em PHP 6
- 6.0 Execução Local do Projeto 7**
- 7.0 Conclusão 8**
- 8.0 Anexo 8**

1.0 Introdução

Durante o desenvolvimento de um projeto acadêmico em tempo limitado, surgiu a necessidade de criar uma solução simples, funcional e rápida para a geração de dados de acesso dos alunos a uma plataforma educacional. O sistema, desenvolvido em poucos dias, teve como principal objetivo automatizar a criação de **PDF's personalizados** com os dados de login de cada estudante, organizados por turma.

Para isso, foi desenvolvido um site simples utilizando **HTML**, **CSS**, **JavaScript**, **PHP** e **MySQL**, capaz de importar dados a partir de planilhas, gerar os acessos, e disponibilizá-los de forma visual e prática para impressão ou envio digital. Todo o processo, da estruturação do banco de dados até a exportação do PDF, foi feito pensando em **otimizar tempo** e **facilitar o uso** por educadores e gestores escolares.

Este e-Book apresenta todo o passo a passo da criação do projeto, incluindo trechos de código comentados, explicações técnicas e dicas práticas para quem deseja reutilizar ou adaptar a ideia em outro contexto escolar. Ao final, o leitor encontrará também o link para o repositório no GitHub, além de instruções para execução local.

1.1 Contexto do Projeto

Trabalhando na área de TI e suporte ao usuário em uma escola chamada **Colégio Caetano**, enfrentei, junto com um colega, um desafio recorrente no início do ano letivo: distribuir os acessos dos alunos à plataforma educacional **Bernoulli 4.0**.

Embora o Bernoulli fosse funcional no que diz respeito à criação de logins, ele **não oferecia suporte nativo para exportar ou compartilhar esses acessos** com os alunos — não havia a opção de envio por e-mail, nem a geração de um PDF personalizado por usuário. Esse processo acabava sendo manual, moroso e sujeito a erros.

Foi então que, inspirada por uma funcionalidade semelhante em outro sistema educacional parceiro, tivemos a ideia de **criar uma ferramenta própria** para suprir essa carência. A solução precisava ser:

- **Rápida de desenvolver** (o prazo era de dois dias);
- **Fácil de usar por pessoas com pouca familiaridade com sistemas**;
- E, principalmente, **eficiente na geração em massa de PDF's personalizados**.

O projeto foi nomeado como **“Matrículas”** e, embora tenha nascido como uma solução emergencial, se mostrou extremamente funcional e prático — tanto que decidi transformá-lo neste e-Book para compartilhar o conhecimento com outros profissionais da área educacional e de tecnologia.

1.2 Objetivo do Projeto

O principal objetivo deste projeto é proporcionar uma solução prática, ágil e funcional para a geração de credenciais de acesso dos alunos à plataforma Bernoulli 4.0, utilizada pelo Colégio Caetano como ferramenta de apoio pedagógico.

Diante da inexistência de um recurso nativo no sistema da plataforma que permitisse o envio automatizado ou em lote dos dados de login aos estudantes, surgiu a necessidade de criar um mecanismo interno que facilitasse essa entrega. O projeto “Geração de PDF's de Matrículas” foi

idealizado justamente para suprir essa lacuna, possibilitando a criação de arquivos personalizados contendo os dados de acesso de cada aluno, organizados por turma.

Além disso, o sistema buscou otimizar o processo de distribuição das credenciais, evitando erros manuais, otimizando o tempo de entrega e reduzindo o retrabalho por parte da equipe de TI e suporte. Ao permitir a seleção de turmas e alunos e a geração automática de um documento em PDF com orientações de acesso, a ferramenta torna o processo mais eficiente e acessível tanto para os profissionais envolvidos quanto para os próprios estudantes.

2.0 Planejamento e Divisão de Tarefas

Este capítulo apresenta a etapa de organização inicial do projeto, detalhando como foi feito o planejamento estratégico para sua execução. Nele, abordamos a composição da equipe e a distribuição de responsabilidades entre os envolvidos, além da definição do cronograma e do tempo disponível para a realização das tarefas. Essa etapa foi essencial para garantir que o desenvolvimento ocorresse de forma ordenada, produtiva e dentro dos prazos estabelecidos.

2.1 Equipe e Responsabilidades

Para desenvolver esse projeto, contei com a ajuda de um colega, e juntos dividimos as tarefas de forma que aproveitássemos melhor o tempo e as habilidades de cada um.

Fiquei responsável por toda a parte técnica: desde a estruturação do banco de dados no MySQL Workbench até a criação do site utilizando HTML, CSS, JavaScript e PHP. Também fui eu quem preparou o ambiente de desenvolvimento local no VSCode, configurei as conexões com o banco, fiz os testes e desenvolvi a funcionalidade de geração de PDF's com os dados de acesso dos alunos. Além disso, organizei o repositório do projeto e deixei tudo documentado para facilitar futuras consultas.

Já o meu colega ficou responsável por organizar os dados. Ele criou os logins de todos os alunos, separou por turma e gerou planilhas em formato CSV, que foram fundamentais para importar os dados de forma eficiente para o banco de dados.

2.2 Cronograma e Tempo Disponível

O tempo foi um dos maiores desafios deste projeto. Desde o início, ficou claro que não se tratava de algo com prazo flexível ou extenso. Pelo contrário, tínhamos apenas **dois dias** para entregar todos os cadastros prontos e funcionando corretamente.

Essa urgência exigiu foco absoluto e decisões ágeis. Não houve espaço para longos planejamentos ou etapas prototipadas — eu precisei colocar a mão na massa desde o primeiro momento. Minha prioridade foi garantir que o sistema conseguisse importar os dados dos alunos de forma organizada e, principalmente, que gerasse os PDF's de maneira funcional e padronizada.

3.0 Importação de Dados

Neste capítulo, apresento como foi realizada a etapa de importação dos dados dos alunos, um dos pontos centrais do projeto. Explico como preparei os arquivos CSV que continham as informações fornecidas pela instituição, detalho a estrutura adotada no banco de dados MySQL e mostro a lógica

por trás da criação de tabelas separadas para cada turma. Essas etapas foram fundamentais para garantir a organização, integridade e facilidade de acesso aos dados durante a geração dos PDF's personalizados.

3.1 Preparação dos arquivos CSV

Meu colega ficou responsável por cadastrar manualmente todos os alunos. Ele organizou as informações em planilhas do Excel, separando os dados por turma, um a um. Em cada tabela foram inseridos os campos essenciais como nome completo, matrícula, login e senha. Ao final, cada planilha foi salva individualmente no formato .csv, com o nome correspondente à turma, facilitando a importação direta para o banco de dados MySQL.

3.2 Estrutura do banco de dados MySQL

O banco de dados utilizado no projeto foi o **MySQL**, escolhido por sua facilidade de uso, integração com PHP e confiabilidade. Para manter os dados organizados, criamos uma tabela separada para cada turma, com o mesmo formato e estrutura.

Cada tabela possui os seguintes campos:

- `N` (chave primária): um número inteiro usado como identificador único.
- `matricula`: número de matrícula do aluno.
- `nome`: nome completo do aluno.
- `login`: login gerado para acesso à plataforma.
- `senha`: senha de acesso do aluno.

A estrutura foi padronizada em todas as tabelas, o que facilitou o processo de leitura e manipulação dos dados no PHP. Os arquivos CSV gerados anteriormente foram importados diretamente para suas respectivas tabelas, mantendo a consistência e integridade das informações.

```
1 CREATE TABLE `1_ano2` (  
2   `N` int NOT NULL,  
3   `matricula` int DEFAULT NULL,  
4   `nome` varchar(50) DEFAULT NULL,  
5   `login` varchar(50) DEFAULT NULL,  
6   `senha` varchar(50) DEFAULT NULL,  
7   PRIMARY KEY (`N`)  
8 )
```

4.0 Desenvolvimento do Projeto

Utilizei tecnologias acessíveis e compatíveis com execução local, como HTML, CSS e JavaScript no frontend, além de PHP e MySQL no backend. Essa escolha permitiu um desenvolvimento ágil, com boa integração entre interface e banco de dados, atendendo à necessidade de entrega rápida e garantindo um sistema funcional e de fácil manutenção.

Organizei o repositório de forma objetiva, separando claramente os arquivos do frontend, backend e banco de dados. Toda a estrutura do projeto, incluindo os diretórios, arquivos utilizados e orientações para execução, está descrita detalhadamente no **Anexo – Capítulo 8.0**, localizado na **página 8** deste documento. O repositório completo também está disponível via link direto para o GitHub.

Pensei no layout com foco na praticidade. Criei um seletor de turmas que, ao ser alterado, envia uma requisição ao servidor solicitando os nomes dos alunos daquela turma. O resultado preenche dinamicamente um segundo seletor com os nomes dos alunos.

Quando um aluno é selecionado, é feita outra requisição ao servidor, que retorna os dados de matrícula, login e senha daquele aluno. Esses dados são exibidos na tela e utilizados para gerar um PDF personalizado, contendo todas as informações necessárias de acesso à plataforma educacional.

A geração do PDF é feita no próprio navegador, utilizando a biblioteca **html2pdf.js**, o que elimina a necessidade de softwares adicionais e torna o processo ágil e intuitivo.

5.0 Explicação Técnica do Código

Neste capítulo, explico a lógica por trás das principais partes do código do projeto, desde a manipulação dos eventos no frontend até a comunicação com o backend e a geração dos PDF's com os dados dos alunos.

5.1 Estrutura HTML

O formulário principal conta com dois campos `<select>`: um para a escolha da turma e outro para a seleção do aluno. A partir dessas seleções, as ações subsequentes são disparadas via JavaScript. Além disso, incluí uma div chamada `.section_dados`, inicialmente oculta, que exibe os dados retornados do backend e que serão utilizados para a geração do PDF.

```
1 <select name="Turmas" id="turmas">
2   <option value="0">Selecione uma turma</option>
3   <option value="11">1º ano 1</option>
4   <option value="12">1º ano 2</option>
5   <option value="13">1º ano 3</option>
6   <option value="21">2º ano 1</option>
7   <option value="22">2º ano 2</option>
8   <option value="31">3º ano 1</option>
9 </select>
```

Também utilizei IDs e classes bem definidos para facilitar a manipulação via JavaScript e estilização com CSS. O container `#content`, por exemplo, é o elemento central que contém os dados a serem renderizados no PDF com a biblioteca `html2pdf`.

```
1 <script
2   src="https://cdnjs.cloudflare.com/ajax/libs/
3   html2pdf.js/0.10.3/html2pdf.bundle.min.js"
4   integrity="sha512-yu5WG6ewBNKx8svICzUA01vozhmiQCV
5   fzjzW40eCHJdsDRaOifh9hPLwBDex5b32gWCzawTp1F3FJz60ps6TnQ=="
6   crossorigin="anonymous"
7   referrerpolicy="no-referrer">
8 </script>
```

5.2 Lógica em JavaScript

A primeira parte do script é responsável por escutar alterações no campo de seleção de turmas (#turmas). Quando o usuário escolhe uma turma, o código converte o valor selecionado em um nome de tabela correspondente no banco de dados (como 1_ano1, 2_ano2, etc.). Esse nome é então enviado via fetch() para o backend (index.php) utilizando o método POST e o corpo da requisição em JSON. O backend retorna a lista de alunos, que é então usada para popular o campo de seleção de alunos (#alunos) dinamicamente.

```
1 switch(turma){
2   case 11: table = "1_ano1"; break;
3   case 12: table = "1_ano2"; break;
```

```
1 fetch('http://localhost:8000/lib/index.php',{
2   method: 'POST',
3   headers: {'Content-Type': 'application/json'},
4   body: JSON.stringify({table: table})
5 })
```

A segunda parte da lógica entra em ação quando o botão de gerar PDF é acionado. O JavaScript pega o nome do aluno selecionado e a tabela da turma, envia essas informações para o backend (download.php), e recebe de volta os dados do aluno: matrícula, nome, login e senha.

```
1 document.querySelector(".section_dados").style.display = "block";
2 document.querySelector("#db_nome").textContent = db_nome;
3 document.querySelector("#table").textContent = document.querySelector("#turmas")
4   .options[document.querySelector("#turmas").selectedIndex].textContent;
5 document.querySelector("#db_matricula").textContent = db_matricula;
6 document.querySelector("#db_senha").textContent = db_senha;
7 document.querySelector("#db_login").textContent = db_login;
```

Com esses dados recebidos, a interface exibe dinamicamente essas informações na seção oculta .section_dados, que é então revelada. Por fim, utilizo a biblioteca html2pdf.js para gerar um PDF diretamente no navegador com os dados exibidos.

```
1 // Gerar e baixar PDF
2 html2pdf().set(options).from(content).save();
```

5.3 Conexão e Requisições em PHP

No lado do servidor, implementei dois arquivos PHP principais: index.php e download.php. Ambos são responsáveis por receber as requisições feitas pelo JavaScript e retornar os dados apropriados em formato JSON.

5.3.1 index.php

Esse arquivo recebe o nome da tabela via POST e executa uma consulta SQL para buscar os nomes dos alunos daquela turma específica. Para evitar problemas de segurança, como **SQL Injection**, implementei uma verificação que permite apenas nomes de tabelas previamente definidos em um array (`$allowed_tables`). Se a tabela recebida for válida, a consulta `SELECT nome FROM tabela` é realizada e os nomes dos alunos são retornados para preencher dinamicamente o campo de seleção no front-end.

```
1 // **Proteção contra SQL Injection** para o nome da tabela
2 $allowed_tables = ['1_ano1', '1_ano2', '1_ano3'];
3 if (!in_array($table, $allowed_tables)) {
4     echo json_encode(['success' => false, 'message' => 'Tabela inválida']);
5     exit;
6 }
```

A conexão com o banco de dados MySQL é feita utilizando `mysqli`, com as credenciais locais do ambiente de desenvolvimento. Também adicionei cabeçalhos para permitir requisições de diferentes origens (CORS), o que é útil em contextos de desenvolvimento local.

```
1 header('Content-Type: application/json');
2 header('Access-Control-Allow-Origin: *');
3 header('Access-Control-Allow-Methods: POST, GET, OPTIONS');
4 header('Access-Control-Allow-Headers: Content-Type');
5
6 $host = "localhost";
7 $user = "root";
8 $password = "148663";
9 $database = "lista_login";
```

5.3.2 download.php

Esse arquivo é responsável por receber o nome do aluno e a tabela correspondente, e retornar todas as informações daquele aluno específico (matrícula, nome, login e senha). Ele realiza uma consulta com a cláusula `WHERE nome = '$aluno'`, buscando o registro exato.

```
1 $query = "SELECT * FROM `{$table}` WHERE `nome` = '{$aluno}'";
2 $result = $conn->query($query);
```


6.0 Execução Local do Projeto

Este capítulo traz todas as instruções para executar o sistema de geração de dados de login localmente, exatamente como foi feito durante o desenvolvimento. O objetivo é permitir que qualquer pessoa que realize um **fork** do repositório consiga configurar e testar o projeto em seu próprio ambiente de forma simples e eficiente.

Antes de iniciar, é necessário garantir que os seguintes softwares estejam instalados:

- **XAMPP** (ou similar, como WampServer, Laragon): Servidor local com Apache, PHP e MySQL.
- **Navegador web** (Google Chrome, Firefox, Edge, etc.).
- **Editor de código** (VS Code ou outro de sua preferência).
- **Git** (opcional, para clonar o repositório).

1. Clonar ou baixar o repositório

Você pode clonar o repositório com o Git:



```
1 git clone https://github.com/BiancaADJ/matriculas.git
```

Ou simplesmente fazer o download como .zip diretamente pelo GitHub e extrair a pasta.

2. Mover o projeto para o diretório do servidor

- Mova a pasta do projeto para o diretório htdocs do XAMPP:
 - Caminho padrão: C:\xampp\htdocs\matriculas

3. Iniciar o servidor local

- Abra o **XAMPP Control Panel**
- Inicie os serviços **Apache** e **MySQL**

4. Importar o banco de dados

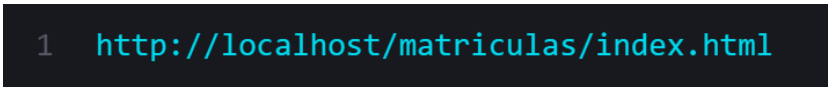
- Acesse <http://localhost/phpmyadmin>
- Crie um banco de dados chamado `lista_login`
- Importe o arquivo de backup com as tabelas e dados das turmas (disponível no anexo ou na pasta database/backup.sql do projeto)

5. Ajustar permissões (se necessário)

- Verifique se o arquivo `index.php` e `download.php` estão com permissões adequadas para leitura e execução.
- Certifique-se de que os nomes das tabelas e colunas estão conforme o esperado no MySQL.

6. Acessar o sistema

Abra o navegador e digite:



```
1 http://localhost/matriculas/index.html
```

A interface será carregada, permitindo selecionar a turma e gerar os dados de login em PDF.

7.0 Conclusão

O desenvolvimento deste projeto representou uma oportunidade de aplicar na prática diversos conhecimentos técnicos, desde a organização de dados em arquivos CSV até a criação de uma aplicação web funcional com geração automatizada de PDFs. Com um prazo extremamente curto, conseguimos entregar um sistema eficiente, capaz de gerar e apresentar rapidamente os dados de login para centenas de alunos organizados por turma.

A experiência foi desafiadora, mas ao mesmo tempo gratificante. Trabalhar com prazos apertados exigiu foco, disciplina e domínio das ferramentas utilizadas. Fiquei responsável por toda a parte técnica do sistema – desde a lógica de backend até a interface visual – enquanto meu colega ficou responsável por organizar e gerar os arquivos com os dados das matrículas.

No fim, entregamos um sistema leve, funcional e intuitivo, que pode ser facilmente replicado por qualquer escola ou instituição que precise de uma solução semelhante. Mais do que cumprir um objetivo pontual, este projeto serviu como uma prova de conceito e como uma base para soluções futuras.

8.0 Anexos

Repositório Git: <https://github.com/BiancaADJ/matriculas>

XAMPP: https://www.apachefriends.org/pt_br/index.html

VSCode: <https://code.visualstudio.com>