**CENTRO PAULA SOUZA**

**ETEC JOSÉ SANT’ANA DE CASTRO**

**Documentação Cadastro de Livros**

**Júlia Carla do Carmo Júlio**

**Maria Cecília da Conceição Pinto**

**Pietra Massarroti**

**Link do repositório:**

https://github.com/PietraMassarotti/CadastroLivros

**OBJETIVO DA APLICAÇÃO**

Permitir que usuários cadastrem, visualizem, editem e excluam livros, com cada livro vinculado a um usuário específico por meio de login.

**Partes Principais da Aplicação**

Sistema de Login e Cadastro e Cadastro de Livros.

O usuário se cadastra com nome, e-mail e senha (cadastro.php).

Os dados são salvos no banco de dados (criar\_user.php).

**Área de Gerenciamento de Livros**

Página principal (index.php) lista os livros do usuário logado.

É possível:

Criar livro (form\_criar.php e criar\_livro.php)

Editar livro (form\_editar.php e editar\_livro.php)

Excluir livro (deletar\_livro.php)

**Banco de Dados**

Usa MySQL.

Tabelas principais:

usuarios (armazenar nome, e-mail, senha)

livros (titulo, autor, ano, id do usuário)

Organização em MVC simples

Models: definem as classes Livro e Usuario.

DAOs: controlam a comunicação com o banco (LivroDAO, UsuarioDAO).

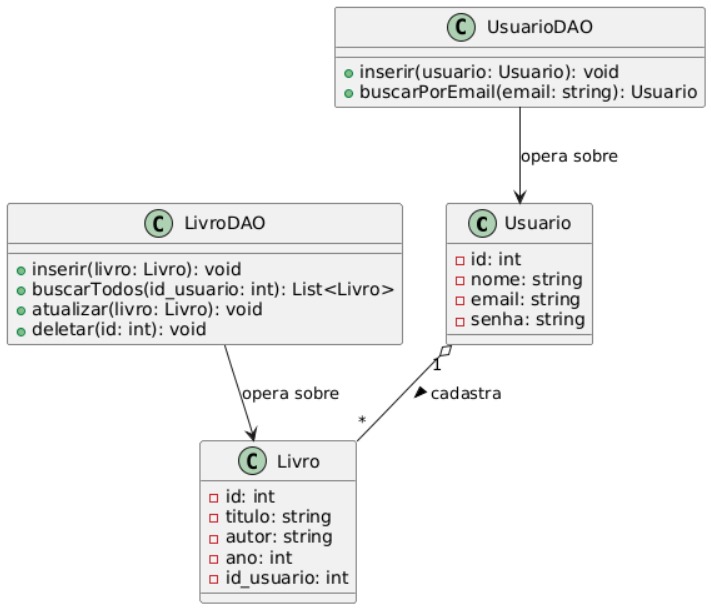
Views: formulários e tabelas HTML.

Controllers: scripts que processam os dados (como criar\_livro.php).

Objetivo Final

Um sistema web em PHP com MySQL, simples e funcional, que permite que cada usuário gerencie sua própria lista de livros. A estrutura segue um padrão básico de MVC e utiliza sessões para manter o controle de usuário.

**Diagrama UML**

****

**Prompts Utilizados**

1. Adicione uma tabela de livros no meu banco de dados tendo como atributos por exemplo autor, ano, isbn, titulo, id etc.

– Cria o banco de dados

CREATE DATABASE sistema\_login;

USE sistema\_login;

-- Cria a tabela de usuários

CREATE TABLE usuarios (

Id INT PRIMARY KEY AUTO\_INCREMENT,

Nome VARCHAR(255) NOT NULL,

Email VARCHAR(255) NOT NULL UNIQUE,

Senha\_hash VARCHAR(255) NOT NULL,

Created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Resposta**:

Claro! Aqui está o SQL para adicionar uma tabela de livros ao seu banco de dados sistema\_login, com os atributos que você mencionou:

Sql

Copiar

Editar

-- Cria a tabela de livros

CREATE TABLE livros (

Id INT PRIMARY KEY AUTO\_INCREMENT,

Titulo VARCHAR(255) NOT NULL,

Autor VARCHAR(255) NOT NULL,

Ano INT,

Isbn VARCHAR(20) UNIQUE,

Created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

Explicação dos campos:

Id: identificador único do livro.

Titulo: título do livro.

Autor: nome do autor.

Ano: ano de publicação (formato inteiro).

Isbn: número padrão internacional de livro (único).

Created\_at: data e hora de inserção no banco.

Se desejar relacionar os livros com os usuários (por exemplo, para indicar quem cadastrou o livro), posso adicionar uma chave estrangeira. Deseja isso também?

1. Adapte o código a baixo para a tabela de livros: Atributos da tabela livro: id, titulo, autor, ano, isbn, create\_at, update\_at, usuario\_id Código:<?php class Usuario { // Propriedades privadas para encapsular os dados do usuário private $id; // Identificador único do usuário (geralmente autoincrementado no banco) private $nome; private $email; // E-mail do usuário (usado como login/identificador único) private $senha\_hash; // Senha do usuário armazenada de forma segura (hash) private $created\_at; // Data/hora de criação do usuário (geralmente definida pelo banco) public function \_\_construct($data) { $this->id = $data[‘id’]; $this->nome = $data[‘nome’]; $this->email = $data[‘email’]; $this->senha\_hash = $data[‘senha\_hash’]; $this->created\_at = $data[‘created\_at’]; } // Getters (métodos para acessar os valores das propriedades) public function getId() { return $this->id; } public function getNome() { return $this->nome; } public function getEmail() { return $this->email; } public function getSenhaHash() { return $this->senha\_hash; } public function getCreatedAt() { return $this->created\_at; } // Setters (métodos para definir/modificar os valores das propriedades) public function setNome() { $this->nome = $nome; } public function setEmail($email) { $this->email = $email; } public function setSenhaHash($senha\_hash) { $this->senha\_hash = $senha\_hash; } } ?>

**Resposta**:

Claro! Abaixo está o código adaptado para uma classe Livro, com base na estrutura da sua tabela de livros e seguindo o mesmo estilo da classe Usuario:

Php

CopiarEditar

<?php

Class Livro {

// Propriedades privadas do livro

Private $id;

Private $titulo;

Private $autor;

Private $ano;

Private $isbn;

Private $created\_at;

Private $updated\_at;

Private $usuario\_id; // ID do usuário que cadastrou o livro

// Construtor: inicializa os dados com um array associativo

Public function \_\_construct($data) {

$this->id = $data[‘id’];

$this->titulo = $data[‘titulo’];

$this->autor = $data[‘autor’];

$this->ano = $data[‘ano’];

$this->isbn = $data[‘isbn’];

$this->created\_at = $data[‘created\_at’];

$this->updated\_at = $data[‘updated\_at’];

$this->usuario\_id = $data[‘usuario\_id’];

}

// Getters

Public function getId() {

Return $this->id;

}

Public function getTitulo() {

Return $this->titulo;

}

Public function getAutor() {

Return $this->autor;

}

Public function getAno() {

Return $this->ano;

}

Public function getIsbn() {

Return $this->isbn;

}

Public function getCreatedAt() {

Return $this->created\_at;

}

Public function getUpdatedAt() {

Return $this->updated\_at;

}

Public function getUsuarioId() {

Return $this->usuario\_id;

}

// Setters

Public function setTitulo($titulo) {

$this->titulo = $titulo;

}

Public function setAutor($autor) {

$this->autor = $autor;

}

Public function setAno($ano) {

$this->ano = $ano

}

Public function setIsbn($isbn) {

$this->isbn = $isbn;

}

Public function setUpdatedAt($updated\_at) {

$this->updated\_at = $updated\_at;

}

Public function setUsuarioId($usuario\_id) {

$this->usuario\_id = $usuario\_id;

}

}

?>

1. Prompts: Apartir da imagem enviada faça a estrutura das pastas para um documento Read.me Chatgpt: #

📁 Estrutura de Pastas

CadastroDeLivros/

├── config/

│ └── Database.php

│

├── models/

│ ├── Livro.php

│ ├── LivroDAO.php

│ ├── Usuario.php

│ └── UsuarioDAO.php

│

├── public/

│ ├── livros/

│ │ ├── css/

│ │ │ ├── form.css

│ │ │ └── tabela.css

│ │ ├── criar\_livro.php

│ │ ├── deletar\_livro.php

│ │ ├── editar\_livro.php

│ │ ├── form\_criar.php

│ │ ├── form\_editar.php

│ │ ├── index.php

│ │ └── logout.php

│ │

│ ├── login/

│ │ ├── css/

│ │ │ ├── form.css

│ │ │ └── tabela.css

│ │ ├── cadastro.php

│ │ ├── criar\_user.php

│ │ ├── index.php

│ │ ├── login.php

│ │ └── process\_login.php

│

├── utils/

│ ├── Sanitizacao.php

│ ├── SenhaValidacao.php

│ └── ver\_senha.php

│

├── Desafio-IA.pdf

└── README.md (sugestão de adicionar)

**Reflexão sobre uso de IA:**

Durante o projeto, ficou evidente que a IA acelera o desenvolvimento e melhora os resultados. Mais do que uma ferramenta de “cópia e cola”, ela atua como uma assistente que potencializa a criatividade e a eficiência, sempre com supervisão humana. Seu uso é especialmente útil na programação, automatizando tarefas repetitivas. No entanto, é preciso ter clareza nas perguntas, já que a IA não substitui a orientação humana e pode seguir caminhos menos eficazes. A troca de experiências entre pessoas continua sendo essencial para uma aprendizagem mais dinâmica e completa.