****

|  |  |
| --- | --- |
| **Curso** | Técnico em Desenvolvimento de Sistemas |
| **Aluno** | Carina de Paula Mendes |
| **Turma** | EUV - Módulo 3 |
| **Agenda 9** | Desenvolvimento de Sistemas III |

**PHP - ORIENTADO A OBJETO**

1. **Desenvolver um módulo em PHP Orientado a Objetos, aplicado a um recorte do seu Trabalho de Conclusão de Curso (TCC).**
2. **Aplicar os conceitos de classe base e ao menos duas classes filhas, utilizando atributos, métodos, encapsulamento (getters e setters) e herança.**
3. **Usar a Inteligência Artificial como parceira para estruturar e revisar o código.**

**VARAL DOS SONHOS: Plataforma Web para Gestão de Doações e Conexão Solidária**

**1. Introdução**

Este relatório apresenta o desenvolvimento de um módulo em PHP Orientado a Objetos, aplicado ao projeto Varal dos Sonhos, que consiste em uma plataforma digital de gerenciamento de doações.

O sistema conecta crianças em situação de vulnerabilidade social, autoras de cartinhas com seus sonhos (brinquedos), a doadores solidários dispostos a apadrinhá-las. A atividade contou com o apoio da Inteligência Artificial (IA), que auxiliou na definição da estrutura das classes, na implementação do código e na revisão dos principais conceitos de Programação Orientada a Objetos (POO).

**Link do Projeto:**

<https://github.com/CarinaMendesDev/PH_orientado_objeto>

1. **Co-criação com a IA**

O processo de desenvolvimento ocorreu em etapas:

* O recorte do TCC foi apresentado à IA (Varal dos Sonhos).
* A IA sugeriu a criação de uma classe base (Pessoa) e duas classes filhas (Crianca e Doador).
* O código foi organizado em arquivos separados, utilizando require\_once para inclusão.
* Foram implementados atributos privados/protegidos, getters e setters (encapsulamento) e herança.

Por fim, foram instanciados objetos no index.php e exibidos no navegador.

1. **Módulo Desenvolvido: Cadastro e Vinculação de Criança e Doador**

O módulo escolhido para esta atividade foi o Cadastro e Vinculação de Criança e Doador, um recorte do projeto Varal dos Sonhos.

Neste módulo, o objetivo é representar, de forma simples, como as crianças cadastradas com suas cartinhas podem ser vinculadas a doadores solidários, responsáveis por apadrinhar seus sonhos.

Para isso, foram implementadas:

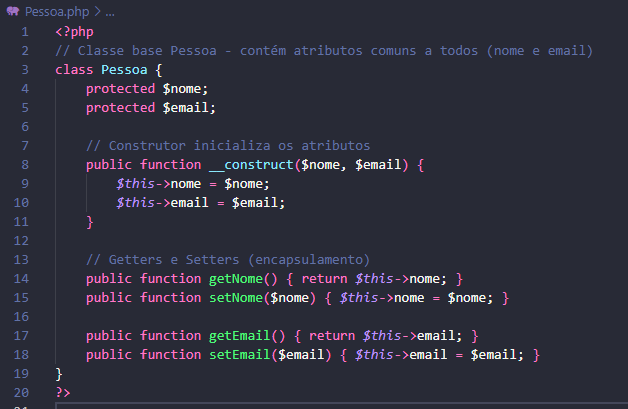
* **Classe base (Pessoa):** contém atributos e métodos comuns (nome e e-mail), aplicando encapsulamento e reutilização de código.
* **Classe filha (Crianca)**: herda de Pessoa e possui atributos específicos como idade e sonho.
* **Classe filha (Doador):** também herda de Pessoa e possui atributos próprios como telefone e presente apadrinhado.
* **Arquivo principal (index.php):** instancia os objetos e exibe os dados no navegador, simulando o vínculo entre criança e doador.

Com esse recorte, é possível observar na prática como os conceitos de POO foram aplicados: herança, encapsulamento, getters e setters, instanciação de objetos e organização do código em arquivos separados.

Este módulo ilustra um dos principais fluxos do projeto Varal dos Sonhos: a ligação entre a cartinha da criança e o cadastro do doador, que representa o núcleo da proposta solidária do sistema.

1. **Código Desenvolvido e Conceitos Aplicados (Pessoa.php, Crianca.php, Doador.php e index.php)**

* **Pessoa.php**



**Conceitos aplicados da apostila:**

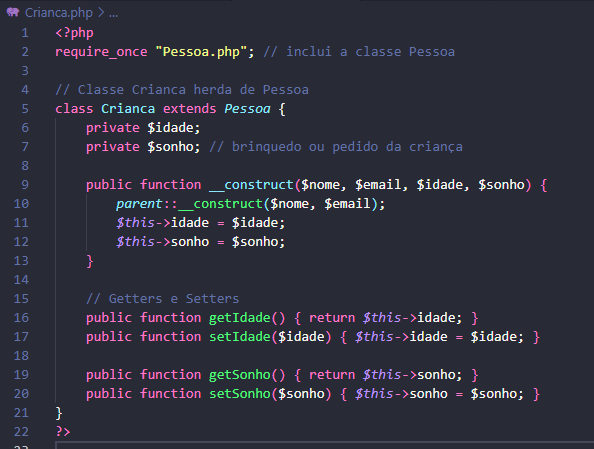
Classe e atributos: definição da classe Pessoa com variáveis internas.

Construtor: usado para instanciar objetos já com valores.

Encapsulamento: atributos protected não podem ser acessados diretamente fora da classe.

Getters e Setters: usados para acessar/modificar atributos privados ou protegidos.

* **Crianca.php**



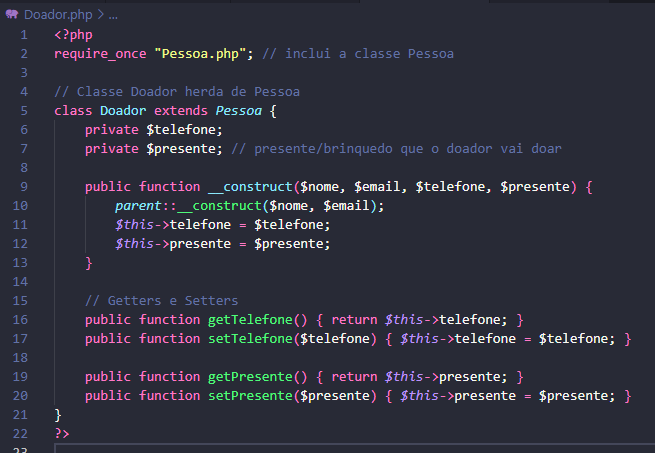
**Conceitos aplicados da apostila:**

Herança: extends Pessoa (a classe filha herda atributos e métodos da classe base).

Inclusão de Arquivos: require\_once garante que a classe Pessoa seja carregada apenas uma vez.

Reutilização de código: parent::\_\_construct() chama o construtor da classe pai.

* **Doador.php**



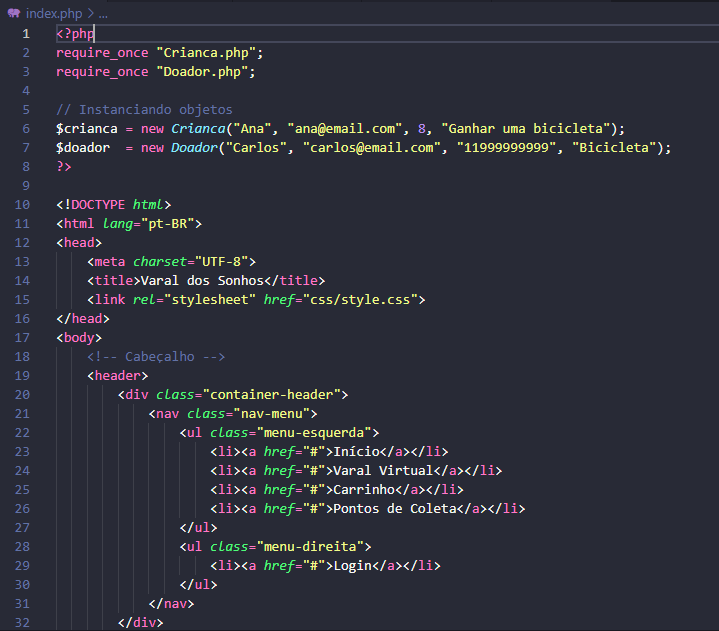
**Conceitos aplicados da apostila:**

Herança novamente aplicada (extends Pessoa).

Encapsulamento com atributos private.

Getters e Setters controlam o acesso seguro às informações.

* **index.php**



**Conceitos aplicados da apostila:**

Inclusão de Arquivos: carregamento das classes.

Instanciação de Objetos: criação de $crianca e $doador com new.

Uso de métodos getters: acesso seguro aos atributos.

Saída no navegador: impressão com echo.

**4. Print da Execução**



**5. Conceitos POO aplicados**

**Classe e Objeto:** A classe Pessoa serviu como molde, e os objetos foram instanciados no index.php. - Classe: Pessoa, Crianca, Doador.- Objeto: $crianca e $doador no index.php.

**Encapsulamento:** atributos private/protected + getters e setters.

**Herança:** Crianca e Doador estendem Pessoa, herdando seus atributos.

**Inclusão de Arquivos**: Foi usado require\_once para carregar as classes.

**Instanciação**: new Crianca(...), new Doador(...).

**6. Conclusão**

A realização desta atividade permitiu consolidar, na prática, os principais conceitos estudados na apostila de Programação Orientada a Objetos (POO), evidenciando como essa abordagem contribui para organizar o código, facilitar o reuso e aumentar a segurança dos dados.

A co-criação com a Inteligência Artificial (IA) trouxe benefícios relevantes ao processo de aprendizagem, tais como:

- Reforçou o entendimento de POO em PHP

- Estruturou as classes,

- Mostrou a importância de organizar o código em arquivos separados.

- Permitiu aplicar o conteúdo diretamente no contexto do TCC Varal dos Sonhos.

- Revisou erros e aplicou os conceitos passo a passo, simulando um trabalho colaborativo e interativo.

Assim, a experiência se mostrou produtiva e enriquecedora, ao simular um trabalho em dupla.