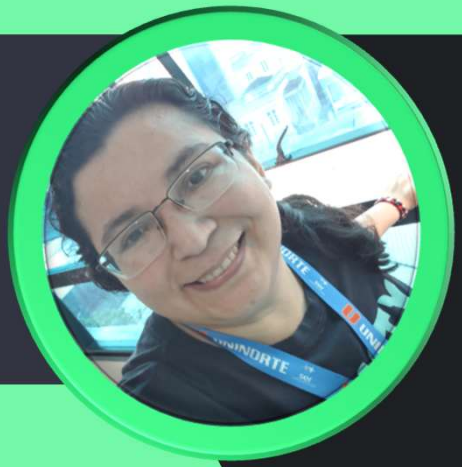
BACK-END FRAMEWORKS

PROFESSORA MÁRCIA SILVA

Hello!



Profa.: Márcia Silva

Graduação:

- Análise e Desenvolvimento de Sistemas
- Pedagogia

Especialização:

- Docência do Ensino Superior
- Projetos de Aplicativos Móveis Multiplataformas
- MBA em Segurança da Informação

Conteúdo Programático

01 UNIDADE

- INTRODUÇÃO À FRAMEWORKS
- O QUE SÃO FRAMEWORKS
- PRINCIPAIS FRAMEWORKS BACK-END
- ARQUITETURA EM CAMADAS

02 UNIDADE

- CLASSIFICAÇÃO POR LINGUAGEM
- COMPARAÇÃO ENTRE FRAMEWORKS

Conteúdo Programático

03 UNIDADE

- REQUISITOS DO DESENVOLVIMENTO DE UM FRAMEWORK
- ESPECIFICAÇÃO DE APLICAÇÃO
- DESENVOLVIMENTO DE APLICAÇÃO

04 UNIDADE

- DEPLOY DE APLICAÇÃO
- CRIAÇÃO DE API

O QUE VAMOS USAR?

- **Navegador:** Chrome, Edge, etc;
- **Editor HTML:** Visual Studio Code;
- **Banco:** Mysql;
- **Linguagens de programação:** PHP.
- **Framework:** Laravel.

PLANO DE ENTREGAS - AVALIAÇÕES

AV1 - Trabalho (Total: 2,0 pontos) + Prova (Total: 8,0 pontos)

Parte 1 - Escrita do Trabalho (Valendo 1,0 pontos)

Parte 2 - Apresentação do Pitch (Valendo 1,0 pontos)

Critérios de Avaliação:

- Clareza na escrita e organização do conteúdo.
- Adequação ao tema e profundidade do conteúdo.
- Coerência entre objetivos, metodologia e resultados apresentados.
- Qualidade da redação (gramática, estrutura e clareza).

Pesos das Avaliações: AV1 (Total: 10,0 pontos)

PLANO DE ENTREGAS - AVALIAÇÕES

AV2 - Projeto Final (Total: 5,0 pontos) + Prova (Total: 5,0 pontos)

Parte 1 - Relatório Escrito do Projeto: 2,5 pontos

Parte 2 - Apresentação do Projeto já desenvolvido: 2,5 pontos

Critérios de Avaliação:

- Complexidade e profundidade do projeto.
- Qualidade técnica e inovação.
- Documentação e justificativa das escolhas feitas durante o desenvolvimento.
- Funcionamento e apresentação do projeto.

Pesos das Avaliações: AV2 (Total: 10,0 pontos)

DATA DAS PROVAS

Avaliação de 2ª Chamada – 20 questões (10,0 pontos).

Avaliação Final – 20 questões (10,0 pontos).

AVALIAÇÃO 1
08/04

AVALIAÇÃO 2
03/06

AVALIAÇÃO 2ª CHADA
17/06

AVALIAÇÃO FINAL
24/06

PLANO DE ENTREGAS - AVALIAÇÕES

Observações:

- O projeto final será uma aplicação web funcional que deve ser entregue com o código-fonte completo e documentação.
- Os alunos devem demonstrar a aplicação ao vivo durante a apresentação ou fornecer um link de acesso para a avaliação da aplicação.
- O relatório escrito deve ser entregue antes da apresentação, e a aplicação web deve estar funcionando corretamente no momento da apresentação.
- A avaliação levará em conta tanto os aspectos técnicos quanto a clareza na comunicação do trabalho realizado.

01 UNIDADE

1. INTRODUÇÃO À FRAMEWORKS
2. O QUE SÃO FRAMEWORKS
3. PRINCIPAIS FRAMEWORKS BACK-END
4. ARQUITETURA EM CAMADAS

01

INTRODUÇÃO À FRAMEWORKS

INTRODUÇÃO À FRAMEWORKS

01

- O que vocês sabem sobre frameworks?
- Já ouviram falar de algum?
- Qual sua expectativa para a disciplina?

INTRODUÇÃO À FRAMEWORKS

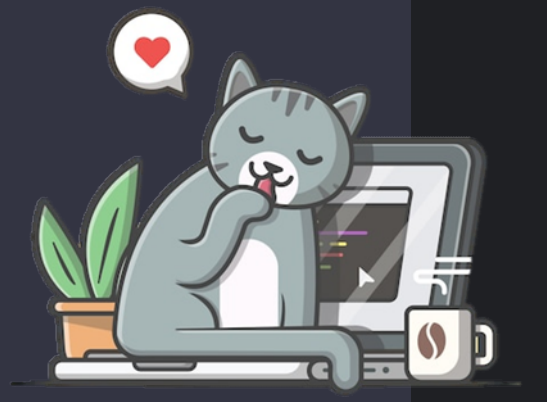
01

Nos últimos anos, a programação web tem passado por uma série de mudanças. Algumas linguagens aparecem, outras se tornam menos usadas, ferramentas são criadas, tudo com o objetivo de tornar a web cada vez mais dinâmica e fluida para o usuário final e, também, mais prática e poderosa (com novos recursos para os desenvolvedores).

INTRODUÇÃO À FRAMEWORKS

01

Usar frameworks é algo que já faz parte do dia a dia da maioria dos desenvolvedores, especialmente de quem trabalha com um grande número de projetos que usam funções similares. Afinal, a possibilidade de reutilizar códigos com poucas alterações ajuda a poupar tempo.



INTRODUÇÃO À FRAMEWORKS

01

O framework dá uma caixa de ferramentas para o programador, que vai além do que é oferecido pela linguagem. Seu conceito, porém, pode ser confuso em relação a outras formas de aproveitar códigos em vários projetos, como a orientação a objetos e às classes.

O QUE SÃO FRAMEWORKS ?

01

O framework nada mais é do que uma ferramenta que vai te ajudar a ter como único objetivo focar em desenvolver o projeto, não em detalhes de configurações.

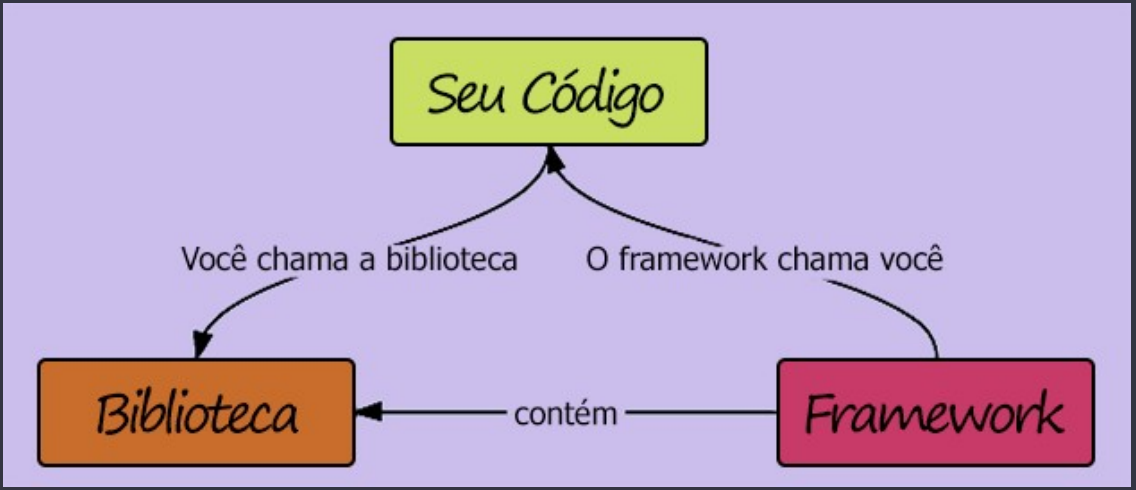
FRAME (a estrutura de uma casa)

WORK (o desenvolvimento)

O QUE SÃO FRAMEWORKS ?

01

O framework trouxe a prática de evitar que tenhamos que fazer tarefas repetitivas, automatizando parte do trabalho.



O QUE SÃO FRAMEWORKS ?

01

Por que usar um Framework?

- Otimizar o tempo de desenvolvimento
- Definir e padronizar as melhores práticas de programação
- Oferecer maior segurança
- Evitar códigos duplicados
- Diminuir a ocorrência de bugs
- Gerar maior consistência no processo de desenvolvimento e nas aplicações criadas

O QUE SÃO FRAMEWORKS ?

01

Por que usar um Framework?

- Reduzir as chances de erro no código
- Possibilitar que os desenvolvedores se dediquem aos elementos específicos do projeto
- Simplificar e encurtar a curva de aprendizado do time de desenvolvimento
- Poupar custos

INTRODUÇÃO À FRAMEWORKS

01

Como um framework funciona?

- Basicamente, é um template com diversas funções que podem ser usadas pelo desenvolvedor. Com ele, é desnecessário gastar tempo para reproduzir a mesma função em diferentes projetos, auxiliando em um gerenciamento ágil de projetos.
- Possui um conjunto de bibliotecas, que permitem aos desenvolvedores trabalharem sobre eles para operações maiores.

O QUE SÃO FRAMEWORKS ?

01

Diferença entre Bibliotecas e Frameworks

Um framework representa a estrutura dentro da qual você desenvolverá um software. Assim, seu código deve — desde o princípio — seguir os padrões estabelecidos pelo framework.

Já uma biblioteca representa recursos que poderão ser utilizados por você no decorrer do desenvolvimento, fornecendo elementos para completar uma etapa do desenvolvimento ou otimizá-lo.

PRINCIPAIS FRAMEWORKS

01

Front - End

- Angular
- Bootstrap
- Vue.js
- React Js

Back - End

- Django (Python)
- Laravel (PHP)
- Ruby on Rails (Ruby)
- ExpressJS (NodeJS)

Introdução ao PHP e Configuração do Ambiente

01

O que é PHP?

- PHP (Hypertext Preprocessor) é uma linguagem de programação interpretada, usada principalmente no desenvolvimento de sites dinâmicos e aplicações web.
- É uma linguagem server-side, ou seja, o código PHP é executado no servidor antes de ser enviado ao navegador.
- Comumente usada em conjunto com o MySQL para manipulação de bancos de dados.

Introdução ao PHP e Configuração do Ambiente

01

Aplicações do PHP:

- Criação de sites dinâmicos.
- Desenvolvimento de sistemas de gerenciamento de conteúdo (CMS).
- Integração com bancos de dados.
- Criação de APIs e sistemas de autenticação.

Introdução ao PHP e Configuração do Ambiente

01

Instalação do XAMPP:

XAMPP é um pacote de software que inclui Apache (servidor web), MariaDB/MySQL (banco de dados), e intérpretes para PHP e Perl. Passos para instalar o XAMPP:

1. Baixe o instalador do XAMPP no site oficial (https://www.apachefriends.org).
2. Siga as etapas do instalador para instalar o XAMPP.
3. Após a instalação, abra o painel de controle do XAMPP e inicie os serviços Apache e MySQL.

Introdução ao PHP e Configuração do Ambiente

01

Estrutura Básica de um Script PHP:

Todo código PHP é escrito entre as tags:

```
<?php
// Código PHP vai aqui
?>
```

Arquivos PHP têm a extensão .php.

Introdução ao PHP e Configuração do Ambiente

01

Estrutura Básica de um Script PHP:

Criar o Primeiro Script: Crie um arquivo PHP chamado teste.php dentro da pasta htdocs.

```
<?php

echo "Bem-vindo ao PHP!";

?>
```

Para executar, acesse no navegador: http://localhost/teste.php.

Introdução ao PHP e Configuração do Ambiente

01

Variáveis em PHP:

Uma variável é um contêiner (recipiente) para armazenar dados que podem ser utilizados e manipulados no programa. As variáveis permitem que o valor armazenado seja alterado durante a execução do script.

No PHP, as variáveis são identificadas com o símbolo $ antes do nome e podem armazenar diferentes tipos de dados, como números, textos ou valores lógicos.

Introdução ao PHP e Configuração do Ambiente

01

Exemplo de uso básico de variáveis:

```
<?php
  $nome = "Márcia"; // Variável que armazena um texto (string)
  $idade = 18;      // Variável que armazena um número inteiro (integer)
  $salario = 2500.50; // Variável que armazena um número decimal (float)
  echo "Nome: $nome, Idade: $idade, Salário: R$ $salario.";
?>
```

Resultado no navegador: Nome: Márcia, Idade: 18, Salário: R$ 2500.5.

Introdução ao PHP e Configuração do Ambiente

01

Regras para Nomes de Variáveis:

- No PHP, todas as variáveis devem começar com um cifrão ($), seguido por uma letra ou um sublinhado (_).

```php
<?php
    $nome = "Márcia";
    $_idade = 18;
?>
```

- Podem conter letras, números e sublinhados após o primeiro caractere:

```php
<?php
    $nome1 = "João";
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Regras para Nomes de Variáveis:

- Não podem conter espaços ou caracteres especiais:

```
<?php
    $meu nome = "Márcia"; // Espaços não são permitidos
    $nome@idade = 18; // Caracteres especiais não são permitidos
?>
```

- São case-sensitive (diferenciam maiúsculas de minúsculas):

  Ex: $Nome e $nome são variáveis diferentes.

Introdução ao PHP e Configuração do Ambiente

01

Regras para Nomes de Variáveis:

- São case-sensitive (diferenciam maiúsculas de minúsculas):

```
<?php
$Nome = "Márcia";
$nome = "Silva";

echo $Nome; // Resultado: Márcia
echo $nome; // Resultado: Silva
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

1. String: Texto entre aspas ("texto" ou 'texto').
2. Integer: Números inteiros (34, -5).
3. Float: Números decimais (3.14, -0.99).
4. Boolean: Verdadeiro (true) ou falso (false).
5. Array: Armazena vários valores em uma única variável.
6. Object: Representa objetos instanciados a partir de uma classe.
7. NULL: Representa a ausência de valor ou uma variável sem valor.
8. Resource: Um identificador especial para recursos externos

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

1. String:

- Representa texto (uma sequência de caracteres).
- Envolvida por aspas simples (') ou duplas (").

```php
<?php
    $nome = "Márcia Silva"; // String
    echo $nome;
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

2. Integer:

- Representa números inteiros (positivos ou negativos).

```php
<?php
    $idade = 18; // Integer
    echo $idade;
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

3. Float (ou Double):

- Representa números decimais ou de ponto flutuante.

```php
<?php
    $preco = 99.99; // Float
    echo $preco;
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

4. Boolean:

- Representa um valor lógico: true (verdadeiro) ou false (falso).

```
<?php
    $ativo = true; // Boolean
    echo $ativo; // Exibe 1 para true, vazio para false
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

5. Array:

- Armazena vários valores em uma única variável.
- Pode ser indexado ou associativo.

Exemplo (array indexado):

```
<?php
    $frutas = ["Maçã", "Banana", "Laranja"];
    echo $frutas[0]; // Exibe "Maçã"
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

5. Array:

Exemplo (array associativo):

```
<?php
    $pessoa = ["nome" => "Márcia", "idade" => 18];
    echo $pessoa["nome"]; // Exibe "Márcia"
?>
```

- O símbolo => no PHP é chamado de operador de associação. Ele é usado principalmente em arrays associativos para indicar a relação entre uma chave e um valor.

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

6. Object:

- Representa objetos instanciados a partir de uma classe.

```
<?php
class Pessoa {
    public $nome;
    public $idade;
}
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

6. Object:

- Representa objetos instanciados a partir de uma classe.

```
$usuario = new Pessoa();
$usuario->nome = "Márcia";
$usuario->idade = 18;
echo $usuario->nome; // Exibe "Márcia"
?>
```

- O símbolo -> no PHP é chamado de operador de objeto (ou "object operator"). Ele é usado para acessar propriedades ou métodos de um objeto instanciado a partir de uma classe.

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

7. NULL:

- Representa a ausência de valor ou uma variável sem valor.

```
<?php
    $vazia = NULL;
?>
```

Introdução ao PHP e Configuração do Ambiente

01

Tipos de Dados:

8. Resource:

- Um identificador especial para recursos externos, como conexão a banco de dados ou manipulação de arquivos.

```
<?php
    $arquivo = fopen("texto.txt", "r"); // Resource
?>
```

Exercício 1: Tipos de Variáveis em PHP

Criar o arquivo exercicio1.php:

- Abra o editor de HTML (Visual Studio Code).
- Crie um novo arquivo dentro da pasta htdocs e salve-o com o nome exercicio1.php.

Definir variáveis de diferentes tipos:

- String: Para armazenar texto.
- Integer: Para armazenar números inteiros (sem casas decimais).
- Float: Para armazenar números com casas decimais.
- Array: Para armazenar múltiplos valores em uma única variável.

Exercício 1: Tipos de Variáveis em PHP

```php
<?php
// 1. String - Armazena uma sequência de caracteres
$nome = "Márcia Silva";
echo "Nome: " . $nome . "<br>";

// 2. Integer - Armazena números inteiros
$idade = 18;
echo "Idade: " . $idade . "<br>";

// 3. Float - Armazena números com casas decimais
$altura = 1.60;
echo "Altura: " . $altura . "m<br>";

// 4. Array - Armazena múltiplos valores
$frutas = ["Maçã", "Banana", "Laranja"];
echo "Fruta 1: " . $frutas[0] . "<br>";
?>
```

EXEMPLO

Até a próxima aula!