

# Desenvolvimento WEB II

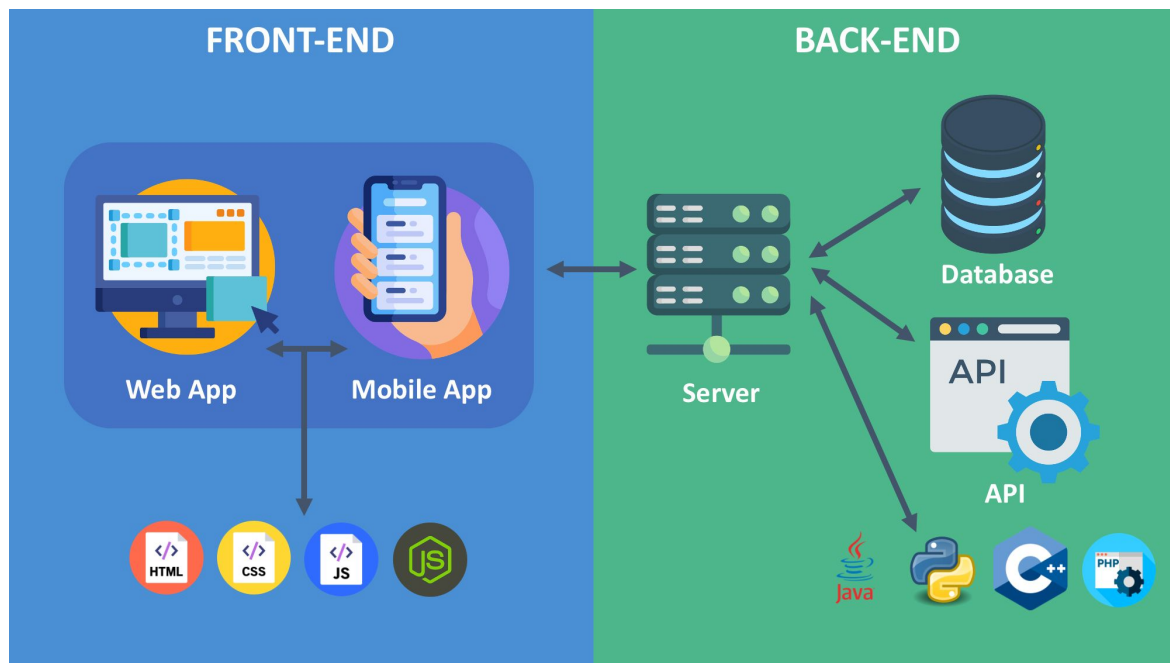
## Prof. Ernani Gottardo

PHP



# PHP

- Front-end
  - é o que seus usuários vêem e interagem, incluindo elementos visuais
- Back-end
  - consiste nos dados e na infraestrutura que fazem a aplicação funcionar



# PHP

- PHP é uma linguagem de script open source que executa no **lado do servidor**
- PHP é executado em várias plataformas (Windows, Linux, Unix, Mac OS X, etc.)
- PHP é compatível com quase todos os servidores web usados atualmente (Apache, IIS, etc.)
- PHP suporta uma ampla gama de bancos de dados

# Funcionalidades PHP

- Gerar conteúdo de página dinâmico
- Adicionar, excluir, modificar dados em seu **banco de dados**
- Criar, abrir, ler, escrever, excluir e fechar **arquivos no servidor**
- **Coletar dados** de formulário
- Enviar e receber **cookies**

# Arquitetura PHP



# Ambiente PHP



## WampServer

Apache, MariaDB, MySQL, PHP, PhpMyadmin,  
MySQL Workbench



# WampServer

- A pasta "**www**" é criado automaticamente (geralmente **c:\wamp64\www**)
- Crie uma **subpasta** em "www" e coloque os arquivos .php dentro (colocar também HTML, js, css).
- Clique no link "**localhost**" no menu do WampSever ou abra o navegador de internet e acesse a URL: `http:// localhost`

# Configurar Visual Studio

- Criar arquivo launch.json (Run -> Add Configuration...)

```
{  
  "name": "Launch Chrome",  
  "request": "launch",  
  "type": "chrome",  
  "url": "http://localhost/${workspaceFolderBasename}/${relativeFile}",  
  "webRoot": "${workspaceFolder}"  
},  
{  
  "type": "msedge",  
  "request": "launch",  
  "name": "Launch Edege",  
  "url": "http://localhost/${workspaceFolderBasename}/${relativeFile}",  
  "webRoot": "${workspaceFolder}"  
}
```



# Conceitos Básicos PHP

- Um script PHP pode ser colocado em **qualquer parte** do documento HTML.
- Um script PHP começa com **<?php** e termina com **?>** .
- A extensão de arquivo padrão para arquivos PHP é **.php**(como .html não funciona).

# Conceitos Básicos PHP

- Criar arquivo launch.json (Run -> Add Configuration...)

```
<!DOCTYPE html>
<head></head>
<html>
<body>
    <h1>My first PHP page</h1>

    <?php
    //comentários em PHP são iguais a javascript
    //início do código PHP
    echo "<h2>PHP is Fun!</h2>";
    echo "Hello World PHP! <br>";
    ?>

</body>
</html>
```

# Variáveis

- No PHP, uma variável começa com o sinal \$, seguido pelo nome da variável.
- As variáveis em PHP são case-sensitive.
- PHP é uma linguagem fracamente tipada

```
<?php
```

```
$txt = "IFRS Erechim";
```

```
$x = 5;
```

```
$y = 10.5;
```

```
echo "I love " . $txt . "!" . "<br>"; //"."->concatenar string
```

```
echo "I love $txt! <br>"; //aspas duplas permite colocar a variável diretamente
```

```
?>
```

# Array

- Indexados

```
$cars = [];  
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";  
$arrlength = count($cars); //número de elementos do array
```

- Associativos

```
$age = [];  
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";  
echo "Peter is " . $age['Peter'] . " years old." . '<br/>';
```

# Array

- Multidimensional

```
$arr = array (array())//ou $arr = [[]];
```

```
$arr[0][0]="Maria"; //nome
```

```
$arr[0][1]=42; //idade
```

```
$arr[1][0]="Paulo";
```

```
$arr[1][1]=45;
```

```
echo $arr[1][1] . '<br/>';
```

- Ordenação

- `sort()` - ordena arrays em ordem crescente
- `rsort()` - ordenar ordena por ordem decendente

```
sort($cars);
```

```
var_dump($x);
```

- `var_dump($arr)`: mostra informações gerais sobre variáveis/arrays

# Operadores Aritméticos

Operador	Nome	Exemplo
+	Adição	$\$x + \$y$
-	Subtração	$\$x - \$y$
*	Multiplicação	$\$x * \$y$
/	Divisão	$\$x / \$y$
%	Módulo	$\$x \% \$y$
**	Exponenciação	$\$x ** \$y$

# Operadores Comparação

Operador	Nome	Exemplo
==	Igual	\$x == \$y
!=	Diferente	\$x != \$y
<>	Diferente	\$x <> \$y
>	Maior que	\$x > \$y
<	Menor que	\$x < \$y
>=	Maior que ou igual a	\$x >= \$y
<=	Menor que ou igual a	\$x <= \$y

# Operadores Lógicos

Operador	Nome	Exemplo
&&	E	\$x && \$y
	OU	\$x    \$y
!	NEGAÇÃO	!\$x



# Funções de String

Função	Descrição
<a href="#"><u>str_replace()</u></a>	Substitui alguns caracteres em uma string (diferencia maiúsculas de minúsculas)
<a href="#"><u>strlen(string)</u></a>	Retorna o comprimento de uma string
<a href="#"><u>strpos(string, localizar)</u></a>	Retorna a posição da primeira ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas). Se não encontrar retorno "False"
<a href="#"><u>strrchr()</u></a>	Encontra a última ocorrência de uma string dentro de outra string
<a href="#"><u>strripos(string, localizar)</u></a>	Encontra a posição da última ocorrência de uma string dentro de outra string (sem distinção entre maiúsculas e minúsculas)
<a href="#"><u>strrpos(string, localizar)</u></a>	Localiza a posição da <b>última</b> ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas)
<a href="#"><u>strstr()</u></a>	Encontra a <b>primeira</b> ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas)
<a href="#"><u>strtolower()</u></a>	Converte uma string em letras minúsculas
<a href="#"><u>strtoupper()</u></a>	Converte uma string em letras maiúsculas
<a href="#"><u>substr(string, start,length)</u></a>	Retorna uma parte de uma string
<a href="#"><u>trim()</u></a>	Remove espaços em branco ou outros caracteres de ambos os lados de uma string

# Funções de String

```
<?php  
echo strlen("Hello world!")."<br/>"; // resultado 12  
$pos= strpos("Hello world!", "world"); // resultado 6, se não encontrar  
retorna 0 ou false  
if($pos===false){  
    echo "String não encontrada";  
} else {  
    echo "Posição da string ".$pos;  
}  
echo str_replace("world", "Dolly", "Hello world!"); // resultado Hello  
Dolly!  
echo substr("Hello world!",2,4);  
?>
```

# Funções matemáticas

```
echo(round(4.96754,2) . "<br>"); //arredondar 2 decimais
```

```
echo(rand(10,100) . "<br>"); // número randômico entre 10 e 100
```

```
echo(ceil(0.60) . "<br>"); // número inteiro acima
```

```
echo(floor(0.60) . "<br>"); // número inteiro abaixo
```

```
echo(pow(2,4) . "<br>"); //potência
```

```
echo(sqrt(9) . "<br>"); // raiz quadrada
```

# Funções de data

- A função PHP **date ()** formata um timestamp para uma data e hora.
- A sintaxe da função date() é a seguinte.
  - date(format,timestamp)
- Setar o fuso horário:

```
date_default_timezone_set('America/Sao_Paulo');
```

- Se não for informado timestamp será retornada a data e hora atual do **servidor**.

```
<?php
```

```
echo "Data de Hoje " . date("d/m/Y") . "<br>";
```

```
echo "Dia do mês " . date("m") . "<br>";
```

```
echo "Ano " . date("Y") . "<br>";
```

```
echo "Dia da semana " . date("w") . "<br>";
```

```
echo "Data/Hr atual " . date("d/m/Y H:i:s") . "<br>";
```

```
?>
```

# Formatação data

- Opções para o parâmetro format de data.
  - d - Representa o dia do mês (01 a 31)
  - m - representa um mês (01 a 12)
  - Y - representa um ano (em quatro dígitos)
  - l - representa o dia da semana como texto
  - w - representa o dia da semana como número (0-domingo a 6-sábado)
  - t- último dia do mês

# Formatação hora

- Opções para o parâmetro **format** de hora.
  - H - formato de 24 horas de uma hora (00 a 23)
  - h - formato de 12 horas de uma hora com zeros à esquerda (01 a 12)
  - i - Minutos com zeros à esquerda (00 a 59)
  - s - segundos com zeros à esquerda (00 a 59)
  - a - Em minúsculas Ante meridiem e Post meridiem (am ou pm)

# Criar variável tipo data

- A função PHP **date\_create(*time*, *timezone*)** pode ser usada para criar uma variável do tipo data com a data atual do servidor
- A função PHP **date\_format(*object*, *format*)** pode ser utilizada para formatar uma variável tipo data
- A função PHP **date\_create\_from\_format(*format*, *time*, *timezone*)** pode ser usada para converter uma string legível para um timestamp.

```
$hj=date_create();
```

```
echo date_format($hj,"d/m/Y H:i:s");
```

```
$dt1=date_create_from_format("d/m/Y H:i:s","02/05/2024 00:00:00");
```

```
echo date_format($dt1,"d/m/Y H:i:s");
```

# Checar validade de data

- A função PHP **checkdate**(*month*, *day*, *year*) verifica se uma data é válida

```
$month = 02;
```

```
$day = 28;
```

```
$year = 2024;
```

```
// retorna um boolean após a checagem da data
```

```
if (checkdate($month,$day,$year)) {/
```

```
    echo "data válida " . "<br>";
```

```
} else {
```

```
    echo "data inválida " . "<br>";
```

```
}
```



# Diferença entre duas datas

- A função PHP **datediff(*datetime1*, *datetime2*)** calcula a diferença entre duas datas e retorna um objeto PHP **DateInterval**.

```
$date1=date_create_from_format("d/m/Y", "15/02/2024");
```

```
$date2=date_create_from_format("d/m/Y", "15/03/2024");
```

```
$diff=date_diff($date1,$date2);
```

```
echo $diff->format("%a"); // número de dias
```

# Adicionar valores a uma data

- A função **date\_add (object, interval)** pode adicionar dias, meses, anos, horas, minutos e segundos a uma data.

```
$date=date_create_from_format("d/m/Y","22/03/2024");
```

```
$interval = new DateInterval('P5D'); //5 dias 'P1Y': 1 ano 'P30D': 30 dias
```

```
date_add($date,$interval);
```

```
echo date_format($date,"d/m/Y") . '<br>';
```

# Repetição

- while

```
$x = 1;  
while($x <= 5) {  
    echo " O número é: $x <br>";  
    $x++;  
}
```

- for

```
for ($x = 0; $x <= 10; $x++) {  
    echo "O número é: $x <br>";  
}
```

# Repetição

- Foreach (arrays)

```
<?php
```

```
$nums = [10, 20, 30, 40];
```

```
foreach ($nums as $n) {
```

```
    echo "Número atual: " . $n . "<br>";
```

```
}
```

```
?>
```

# End