



# Programação para a WEB - servidor (server-side)

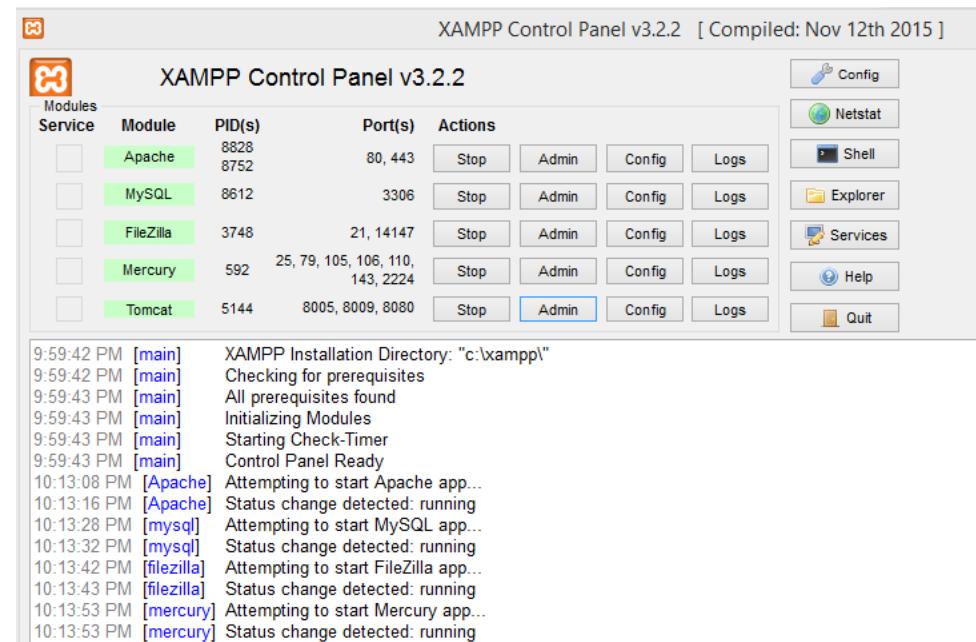
Sérgio da Silva Nogueira

# AMP STACK

- Para utilização do PHP no Windows é necessária a utilização de uma aplicação com AMP STACK
  - Apache
  - MySQL
  - PHP
- Exemplos:
  - AMPPS;
  - WAMP;
  - XAMP;
  - Laragon;
  - ...



# XAMPP



# COMPOSER

- Gestor de dependências de PHP.
- Disponível em: <https://getcomposer.org/>

# COMPOSER



A Dependency Manager for PHP

Latest: 1.9.3

[Getting Started](#)

[Download](#)

# COMPOSER

 [Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

[Download Composer](#) Latest: v1.9.3

## Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

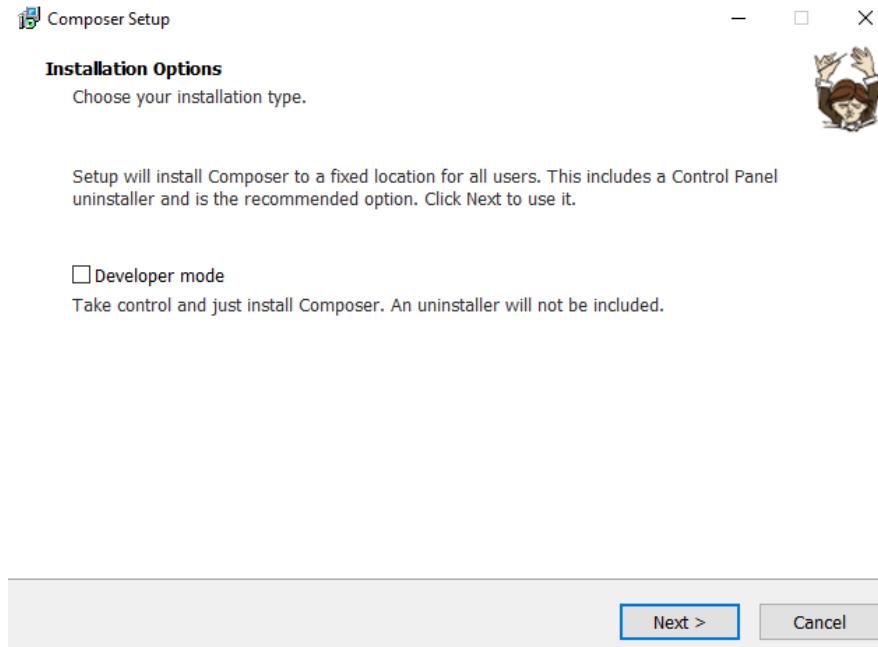
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

## Command-line installation

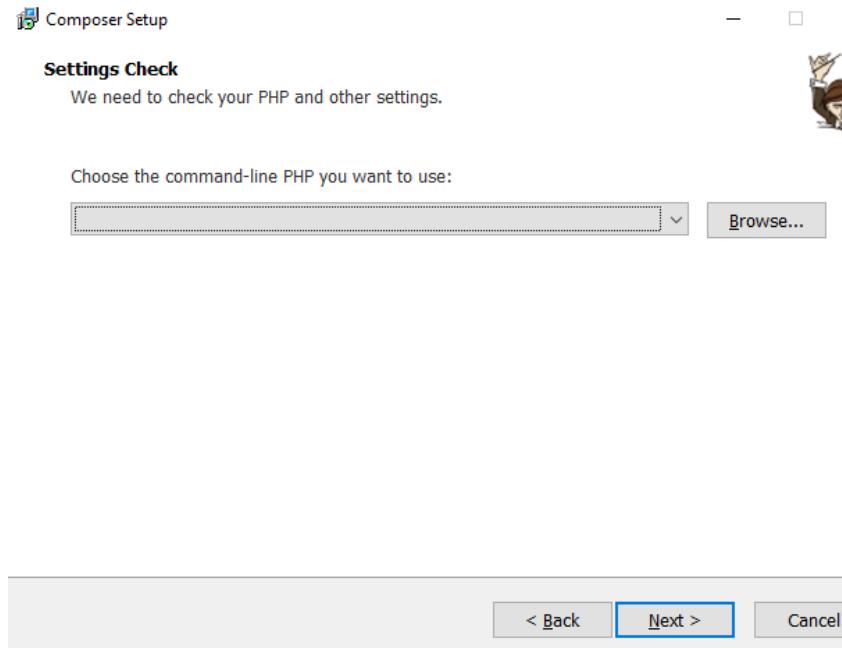
To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"  
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e0012edf3e80b6978849f5eff0d4b4e4c79ff1609dd1e613307e'  
php composer-setup.php  
php -r "unlink('composer-setup.php');"
```

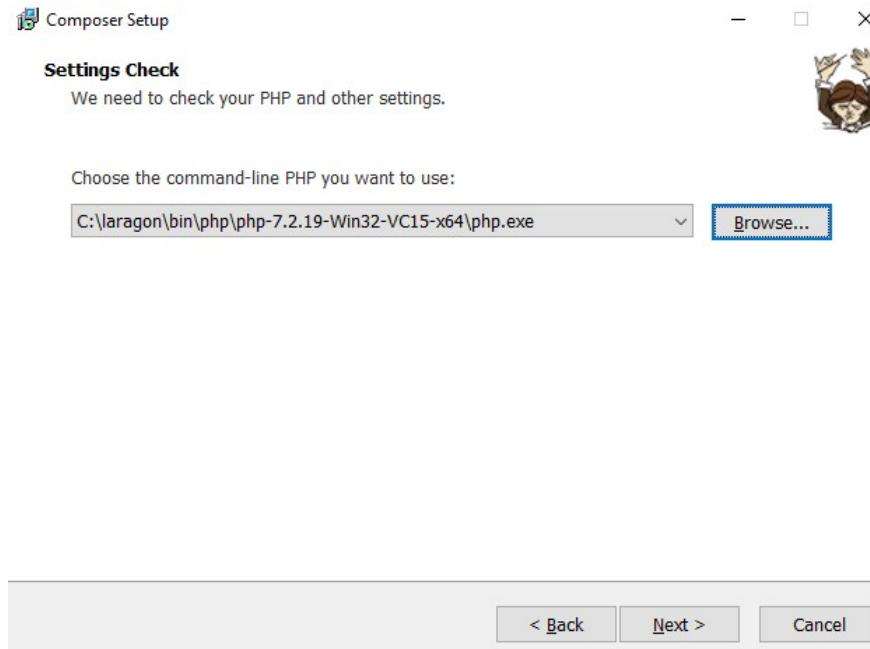
# COMPOSER



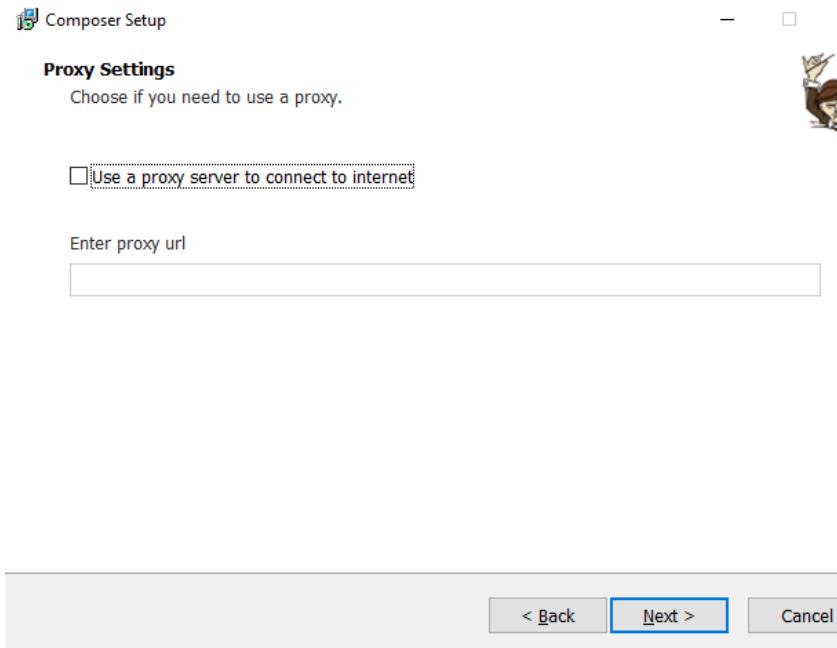
# COMPOSER



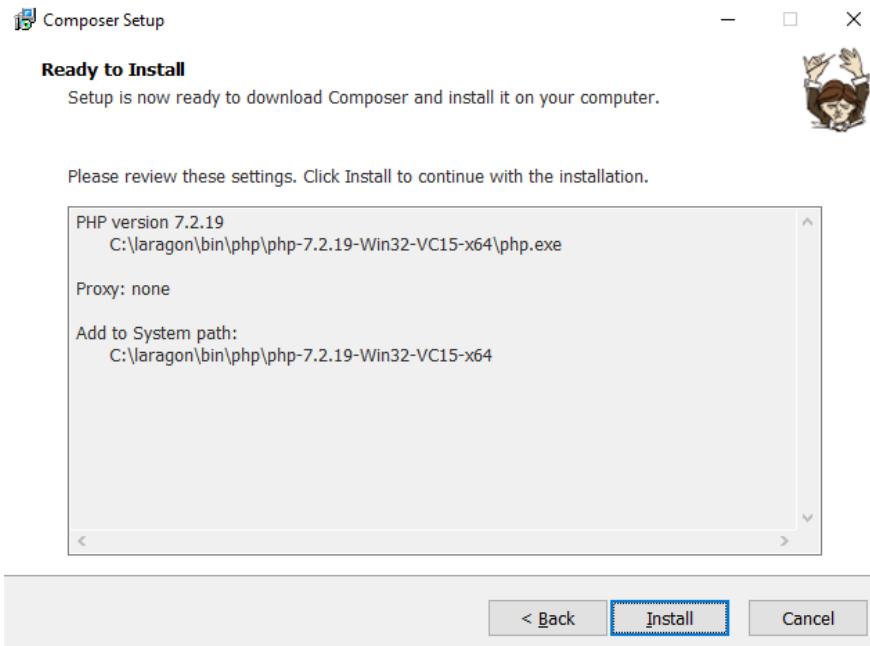
# COMPOSER



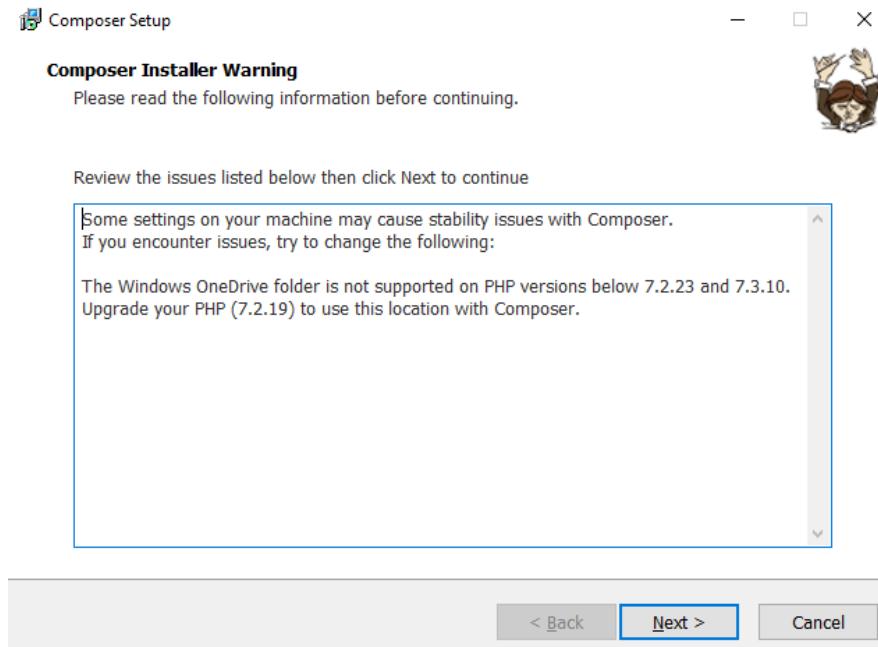
# COMPOSER



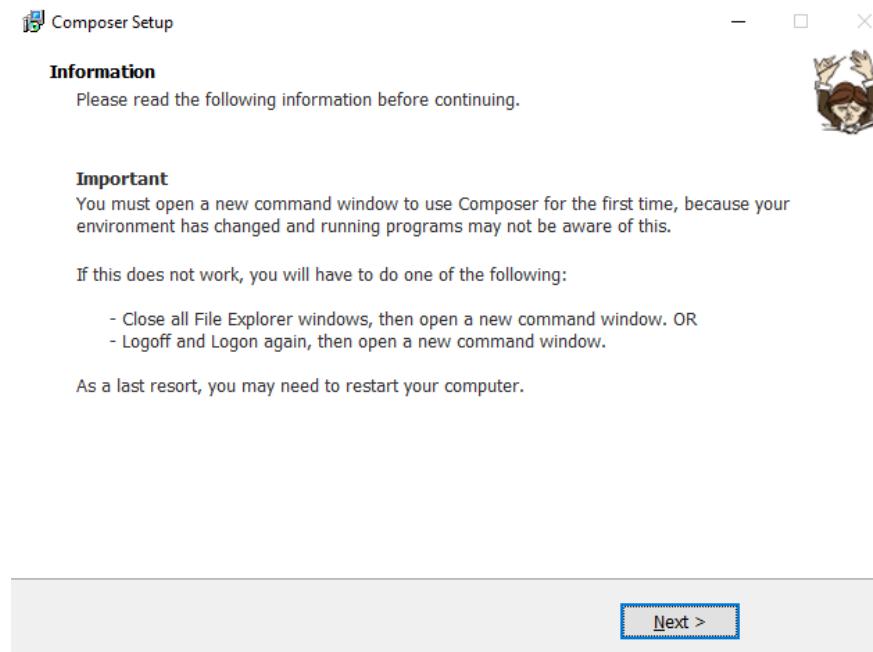
# COMPOSER



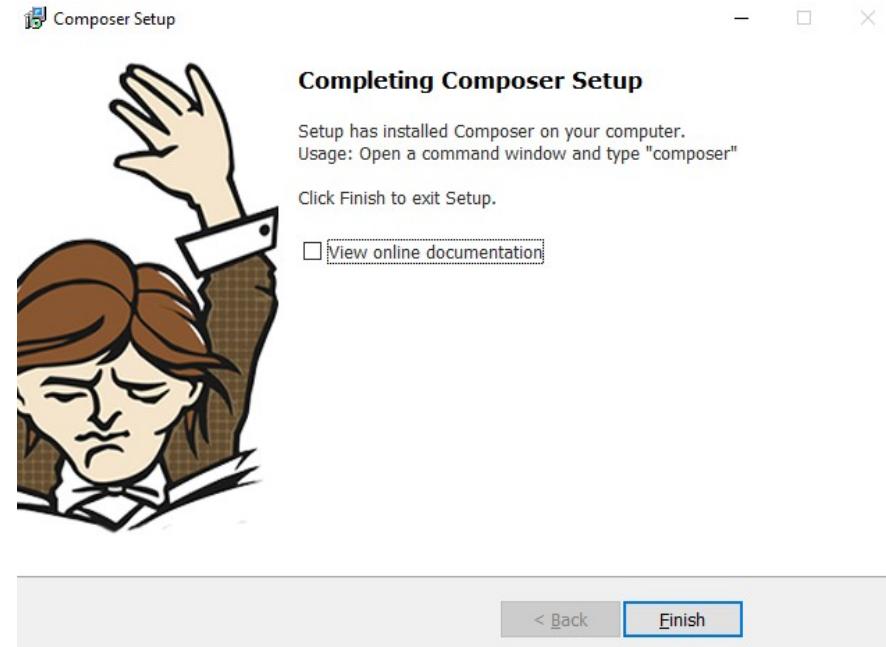
# COMPOSER



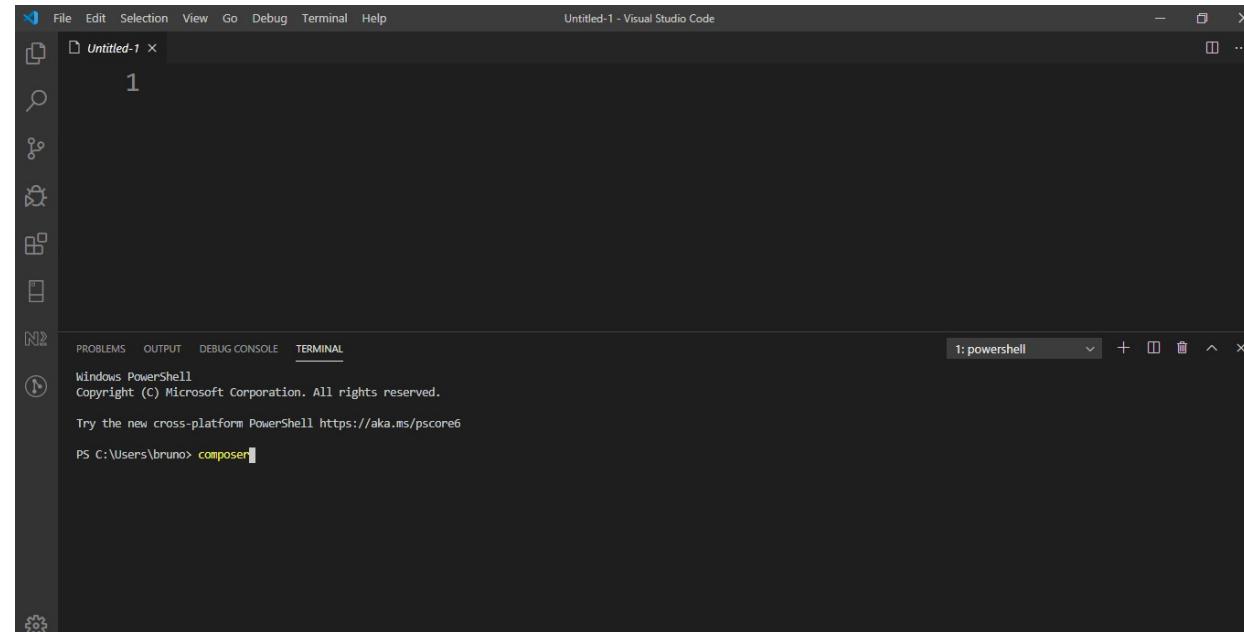
# COMPOSER



# COMPOSER



# COMPOSER



A screenshot of the Visual Studio Code interface. The title bar says "Untitled-1 - Visual Studio Code". The main area shows a dark-themed code editor with a single character '1' on the first line. On the left is a sidebar with various icons. At the bottom is a terminal window titled "1: powershell". The terminal output shows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\bruno> composer
```

Abrir terminal no VS Code e escrever composer, caso sejam apresentados os comandos do composer significa que o mesmo está corretamente instalado

# PHP

- Rasmus Lerdorf – engenheiro de software, membro da equipe Apache e o homem misterioso do ano – é o criador e a força motriz original por trás do PHP. A primeira parte do PHP foi desenvolvida para utilização pessoal no final de 1994. Tratava-se de um wrapper de PerlCGI que o auxiliava a monitorar as pessoas que acessavam o seu site pessoal. No ano seguinte, ele montou um pacote chamado de Personal Home Page Tools (também conhecido como PHP Construction Kit) em resposta aos utilizadores que por acaso ou por relatos falados depararam-se com o seu trabalho. A versão 2 foi logo lançada sob o título de PHP/FI e incluía o Form Interpreter, uma ferramenta para analisar sintaticamente consultas de SQL. Em meados de 1997, o PHP estava sendo utilizado mundialmente em aproximadamente 50.000 sites.

# PHP

- Obviamente estava se tornando muito grande para uma única pessoa administrar, mesmo para alguém concentrado e cheio de energia como Rasmus. Agora uma pequena equipe central de desenvolvimento mantinha o projeto sobre o modelo de “junta benevolente” do código-fonte aberto, com contribuições de desenvolvedores e usuários em todo o mundo. Zeev Suraski e Andi Gutmans, dois programadores israelenses que desenvolveram os analisadores de sintaxe PHP3 e PHP4, também generalizaram e estenderam seus trabalhos sob a rubrica de Zend.com (Zeev, Andi, Zend, entendeu?). O quarto trimestre de 1998 iniciou um período de crescimento explosivo para o PHP, quando todas as tecnologias de código-fonte aberto ganharam uma publicidade intensa. Em outubro de 1998, de acordo com a melhor suposição, mais de 100.000 domínios únicos utilizavam PHP de alguma maneira. Um ano depois, o PHP quebrou a marca de um milhão de domínios. Enquanto escrevo esta apostila, o número explodiu para cerca de dois milhões de domínios.

## O que pode ser feito com PHP?

- Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feita também com PHP, como guardar dados de um formulário, gerar páginas dinamicamente ou enviar e receber cookies. PHP também tem como uma das características mais importantes o suporte a um grande número de base de dados, como dBase, Interbase, mSQL, mySQL, Oracle, Sybase, PostgreSQL e vários outros. Construir uma página baseada em uma base de dados torna-se uma tarefa extremamente simples com PHP. Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP. Ainda é possível abrir sockets e interagir com outros protocolos.

# PHP

- PHP é uma linguagem de programação muito versátil
- O código PHP é escrito entre as tags: <?php e ?>
- Com PHP é possível criar páginas com conteúdo dinâmico, por exemplo, a página do Facebook irá variar consoante o utilizador que se autenticar.
- O código PHP funciona do lado do servidor, assim não é visível para o utilizador.
- Documentação oficial disponível em: <https://www.php.net/manual/en/>

# PHP

- linguagem de propósito geral
- server-side
- desenvolvimento web (originalmente)
- páginas dinâmicas
- influências: C, c++, Perl, Java, Tcl

# PHP

- Velocidade e robustez
- Estruturado e orientação a objetos
- Portabilidade - independência de plataforma - escreva uma vez, execute em qualquer lugar
- Tipagem dinâmica e fracamente tipada
- Sintaxe similar a C/C++ e o Perl • Open-source

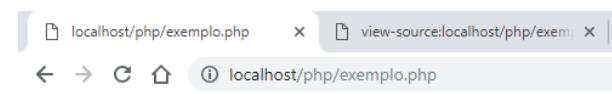
# PHP

- Todos os ficheiros PHP têm a extensão .php
- Mesmo em páginas apenas com conteúdo HTML deve ser utilizada a extensão .php
- Os projetos devem ser guardados na pasta específica do software instalado: htdocs, www ou wwwroot. Outros software AMP Stack podem utilizar uma pasta diferente.
- Para apresentar a página deve ser acedido o link <http://localhost>

## PHP - EXEMPLO c:\xampp\htdocs\atec

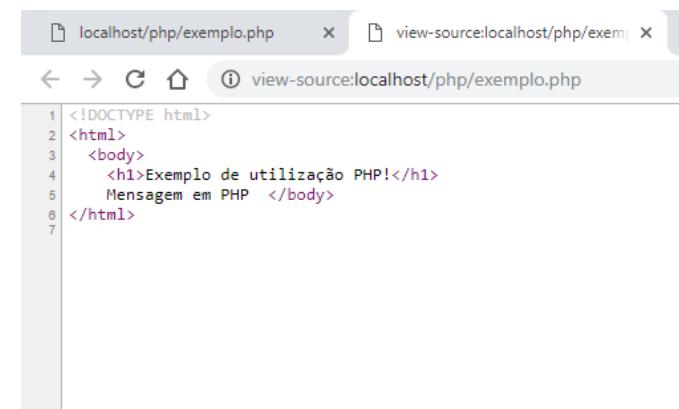
```
<!DOCTYPE html>
<html>
  <body>
    <h1>Exemplo de utilização PHP!</h1>
    <?php
      echo "Mensagem em PHP";
    ?>
  </body>
</html>
```

# PHP - EXEMPLO



**Exemplo de utilização PHP!**

Mensagem em PHP



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>Exemplo de utilização PHP!</h1>
5     Mensagem em PHP  </body>
6   </html>
7 
```

# CONVENÇÕES

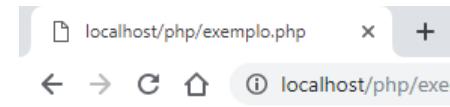
- camelCase

# COMENTÁRIOS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
    echo "<p>Olá, eu serei visto no ecrã</p>";
    // Eu não! Sou apenas um comentário.

    echo "<p>Olá, eu também serei visto</p>";
    # Já eu não serei!

    echo "<p>E eu aqui novamente no ecrã</p>";
    /* Eu não aparecerei no ecrã novamente
       pois sou um comentário */
    ?>
  </body>
</html>
```



Olá, eu serei visto no ecrã

Olá, eu também serei visto

E eu aqui novamente no ecrã

# PHP tipos primitivos

- Escalares
  - Booleano
  - Inteiro
  - Vírgula flutuante
  - String
- Compostos
  - Array
  - Object

# PHP tipos primitivos

- Booleano (case-insensitive)

```
1 True , False , TRUE , FALSE , TrUe , FALSE
```

- Inteiro (bases 2, 8, 10 e 16)

```
1 0b0101 , 0123 , 1234 , -123 , 0x1A
```

- Ponto flutuante

```
1 1.234 , 1.2e3 , 7E-10 , NAN
```

- String

```
1 'Hello World' , "Hello World"
```

## READLINE

- Para ler do teclado em console deve ser utilizado o comando readline
- EX:
- `$name = readline("Please insert your name: ");`

## READLINE

- Para ler do teclado em console deve ser utilizado o comando fgets(STDIN)
- EX:
  - `echo "Please insert your name: ";`  
`$name = fgets(STDIN);`

# VARIÁVEIS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $name = "Guilherme";
      $age = 20;

      echo $name; // Guilherme
      echo "</br>";
      echo $age; // 20
    ?>

  </body>
</html>
```

Guilherme  
20

# VARIÁVEIS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $a = "mundo!";
      echo "Olá, $a <br>"; // Olá,
      mundo!
      echo 'Olá, $a <br>'; // Olá, $a
    ?>
  </body>
</html>
```

Olá, mundo!  
Olá, \$a

# VARIÁVEIS

```
<?php  
    $nome="José";  
?  
  
<!DOCTYPE html>  
<html>  
    <body>  
        <?php  
            echo $nome;  
        ?>  
    </body>  
</html>
```

José

# CONSTANTES

- As constantes são em tudo semelhantes às variáveis, com a particularidade de não ser possível alterar o seu valor após serem definidas.

Para definir uma constante é preciso utilizar a função *define* como a seguir:

```
define ("EXEMPLO", "Olá mundo.");
```

Ou utilizar a palavra reservada *const*, esta função está disponível a partir do *PHP 5.3.0*

```
const EXEMPLO2 = 'Exemplo de constante';
```

## NOMES RESERVADOS

- Alguns nomes são reservados pela própria linguagem não sendo possível a sua utilização para variáveis definidas pelo utilizador.
- Lista de variáveis: <https://www.php.net/manual/en/reserved.keywords.php>

# STRINGS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      echo "Olá," . " mundo!";
    | ?>
    </body>
</html>
```

Olá, mundo!

# STRINGS

```
<?php
    $nome = "José";
    $sobrenome = "Silva";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $nome . " " . $sobrenome;
        ?>
    </body>
</html>
```

José Silva

## STRINGS INTERPOLATION

```
<?php  
  
$name = "Sérgio";  
echo "<p>We need more {$name}s to help us!</p>";
```

# FUNÇÕES DE STRINGS

```
<?php
    $nome = "Teste de PHP";
?>

<!DOCTYPE html>
<html>
<body>
    <?php
        // strlen - Retorna nº de caracteres
        echo strlen($nome) . "<br>";

        // strpos - Retorna primeira ocorrência de um caractere
        echo strpos($nome,"e") . "<br>";

        // stripos - Retorna última ocorrência de um caractere
        echo stripos($nome,"e") . "<br>";

        // strtolower - converte para letras min.
        echo strtolower($nome) . "<br>";

        // strtoupper - converte para letras maiusculas
        echo strtoupper($nome) . "<br>";
    ?>
</body>
</html>
```

12

1

7

teste de php  
TESTE DE PHP

# FUNÇÕES DE STRINGS

```
<?php
    $nome = "Teste DE PHP";
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // SUBSTR_COUNT - Conta quantas ocorreram
            // de um texto ou string
            // Faz diferença Maiusculas e minusculas

            echo substr_count($nome, "e") . "<br>";
            echo substr_count($nome, "E") . "<br>";
        ?>
    </body>
</html>
```

2

1

# FUNÇÕES DE NÚMEROS

```
<?php
    $num1 = 10;
    $num2 = 4;
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            // Exponencial
            echo pow(8,2) . "<br>";

            // Raiz Quadrada
            echo sqrt(25) . "<br>";
        ?>
    </body>
</html>
```

64

5

# FUNÇÕES DE NÚMEROS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Valor absoluto
      echo abs(-10);
    ?>
  </body>
</html>
```

10

# R A N D O M

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Random
      echo rand() . "<br>";

      // Random num intervalo
      echo rand(1,10) . "<br>";
    ?>
  </body>
</html>
```

1989777520  
2

# IS\_NUMERIC

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_numeric($n1) . "<br>";
            echo "N2: ". is_numeric($n2) . "<br>";
            echo "N3: ". is_numeric($n3) . "<br>";
        ?>
    </body>
</html>
```

N1: 1

N2: 1

N3:

# IS\_INT

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_int($n1) . "<br>";
            echo "N2: ". is_int($n2) . "<br>";
        ?>
    </body>
</html>
```

N1: 1

N2:

# IS\_FLOAT

```
<?php
    $n1 = 10;
    $n2 = 1.2;
    $n3 = "Olá";
?>
<!DOCTYPE html>
<html>
    <body>
        <?php
            echo "N1: ". is_float($n1) . "<br>";
            echo "N2: ". is_float($n2) . "<br>";
        ?>
    </body>
</html>
```

N1:

N2: 1

# ROUND

```
<?php  
    $n1 = 1.2;  
    $n2 = 1.7;  
?>  
<!DOCTYPE html>  
<html>  
    <body>  
        <?php  
            echo round($n1) . "<br>";  
            echo round($n2) . "<br>";  
  
            //arredonda sempre para baixo  
            echo floor($n1) . "<br>";  
            echo floor($n2) . "<br>";  
  
            //arredonda sempre para cima  
            echo ceil($n1) . "<br>";  
            echo ceil($n2) . "<br>";  
?>  
    </body>  
</html>
```

1

2

1

1

2

2

# CONVERSÕES DE TIPOS

```
<!DOCTYPE html>
<html>
<body>
<?php
    $var = 100;
    $type_casting = (bool) $var;      // torna-se booleano
    $type_casting = (int) $var;       // torna-se inteiro
    $type_casting = (float) $var;     // torna-se float
    $type_casting = (string) $var;    // torna-se string
    $type_casting = (array) $var;     // torna-se array
    echo $type_casting = (bool)$var; // 1
?
</body>
</html>
```

# OPERADORES ARITMÉTICOS

| <i>Operador</i> | <i>Nome</i>   | <i>Exemplo</i> |
|-----------------|---------------|----------------|
| +               | Adição        | $\$a + \$b$    |
| -               | Subtração     | $\$a - \$b$    |
| -               | Negação       | $-\$a$         |
| *               | Multiplicação | $\$a * \$b$    |
| /               | Divisão       | $\$a / \$b$    |
| %               | Módulo        | $\$a \% \$b$   |

# OPERADORES ARITMÉTICOS

```
<?php  
    $a = 15;  
    $b = 5;  
    $c = $a + $b; // $c = 20  
    $d = $c - $b; // $d = 15  
    $e = $b * 2; // $e = 10  
    $f = $c / 10; // $e = 2  
    $g = $e / 3; // $e = 1  
?  
?
```

# OPERADORES DE ATRIBUIÇÃO

| <i>Operador</i> | <i>Nome</i>                  | <i>Exemplo</i> |
|-----------------|------------------------------|----------------|
| =               | Atribuição simples           | $\$a = \$b$    |
| + =             | Atribuição com adição        | $\$a + = \$b$  |
| - =             | Atribuição com subtração     | $\$a - = \$b$  |
| * =             | Atribuição com multiplicação | $\$a * = \$b$  |
| / =             | Atribuição com divisão       | $\$a / = \$b$  |
| % =             | Atribuição com módulo        | $\$a \% = \$b$ |
| . =             | Atribuição com concatenação  | $\$a . = \$b$  |

# OPERADORES DE ATRIBUIÇÃO

```
<?php  
    $a = 1+1;          // $a = 2  
    $a += 2;           // $a = $a + 2;   ($a = 4)  
    $a -= 1;           // $a = $a - 1;   ($a = 3)  
    $a *= 2;           // $a = $a * 2;   ($a = 6)  
    $a /= 6;           // $a = $a / 6;   ($a = 1)  
    $a %= 3;           // $a = $a % 3;   ($a = 1)  
    echo $a;  
?  
?
```

# OPERADORES DE STRING

| <i>Operador</i> | <i>Nome</i>                 | <i>Exemplo</i> |
|-----------------|-----------------------------|----------------|
| =               | Concatenação                | $\$a = \$b$    |
| . =             | Atribuição com concatenação | $\$a. = \$b$   |

# OPERADORES DE INCREMENTO E DECREMENTO

| <i>Operador</i> | <i>Nome</i>    | <i>Descrição</i>                          |
|-----------------|----------------|---|
| $++ \$a$        | Pré-incremento | Incrementa \$a em um, e então retorna \$a |
| $-- \$a$        | Pré-decremento | Decrementa \$a em um, e então retorna \$a |
| $\$a ++$        | Pós-incremento | Retorna \$a, e então incrementa \$a em um |
| $\$a --$        | Pós-decremento | Retorna \$a, e então decrementa \$a em um |

# OPERADORES DE INCREMENTO E DECREMENTO

```
<?php  
    $a = 10;  
    echo ++$a;      // Saída: 11  
    echo $a++;      // Saída: 11  
    $b = 5;  
    echo $b--;      // Saída: 5  
    echo --$b;      // Saída: 3  
?>
```

# OPERADORES DE COMPARAÇÃO

| <i>Operador</i>       | <i>Nome</i>        | <i>Exemplo</i>                |
|-----------------------|--------------------|-------------------------------|
| <code>==</code>       | Igual              | <code>\$a == \$b</code>       |
| <code>!=</code>       | Diferente          | <code>\$a != \$b</code>       |
| <code>&lt;&gt;</code> | Diferente          | <code>\$a &lt;&gt; \$b</code> |
| <code>====</code>     | Idêntico           | <code>\$a === \$b</code>      |
| <code>!==</code>      | Não idêntico       | <code>\$a !== \$b</code>      |
| <code>&gt;</code>     | Maior que          | <code>\$a &gt; \$b</code>     |
| <code>&lt;</code>     | Menor que          | <code>\$a &lt; \$b</code>     |
| <code>&gt;=</code>    | Maior ou igual que | <code>\$a &gt;= \$b</code>    |
| <code>&lt;=</code>    | Menor ou igual que | <code>\$a &lt;= \$b</code>    |

# OPERADORES DE COMPARAÇÃO

## Exemplo

```
<?php  
    $a=1;  $b="001";  
  
    $c = ($a == $b);          // true  
    $d = ($a === $b);         // false  
  
    $e = 2;  
    $f = ($e <= $a);          // false  
?>
```

# OPERADORES LÓGICOS

| <i>Operador</i>   | <i>Nome</i>  | <i>Exemplo</i>           |
|-------------------|--------------|--------------------------|
| <i>and</i>        | E lógico     | $(\$a \text{ and } \$b)$ |
| <i>&amp;&amp;</i> | E lógico     | $(\$a \&\& \$b)$         |
| <i>or</i>         | OU lógico    | $(\$a \text{ or } \$b)$  |
| <i>  </i>         | OU lógico    | $(\$a \parallel \$b)$    |
| <i>xor</i>        | OU exclusivo | $(\$a \text{ xor } \$b)$ |
| !                 | Não          | $!\$a$                   |

# OPERADORES LÓGICOS

```
<?php  
    $a=1;  
    $b="001";  
    $c=2;  
  
    if($a == $b) and ($a < $c)  
        $a += $c;  
  
    $d = (false || true); // true  
?>
```

# ESTRUTURAS DE DECISÃO

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $idade = 17;

      if($idade < 18) {
        echo "Não pode entrar aqui!";
      } else {
        echo "Bem-vindo!";
      }
    ?>
  </body>
</html>
```

Não pode entrar aqui!

# ESTRUTURAS DE DECISÃO

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $idade = 21;
      $identidade = true;

      if($idade > 18 && $identidade == true)
      {
        echo "Bem-vindo!";
      }
    ?>
  </body>
</html>
```

Bem-vindo!

# ESTRUTURAS DE DECISÃO

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome = "Till Lindemann";

      if($nome == "Richard Kruspe") {
        echo "Olá Richard Kruspe!";
      } elseif ($nome == "Oliver Riedel") {
        echo "Olá Oliver Riedel!";
      } elseif ($nome == "Till Lindemann") {
        echo "Olá Till Lindemann!";
      } else {
        echo "Olá $nome!";
      }
    ?>
  </body>
</html>
```

Olá Till Lindemann!

# ESTRUTURAS DE DECISÃO

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome = "Fulano";

      switch($nome) {
        case "Fulano":
          echo "Olá Fulano!";
          break;

        case "Sicrano":
          echo "Olá Sicrano!";
          break;

        case "Beltrano":
          echo "Olá Beltrano!";
          break;

        default:
          echo "Qual é o seu nome?";
          break;
      }
    ?>
  </body>
</html>
```

Olá Fulano!

# OPERADOR TERNÁRIO

- O operador ternário permite apresentar condições de forma mais simplificada

```
<?php  
  
$preco = 10;  
  
$resultado = ($preco > 0) ? 'Custa '. $preco : 'Gratis';  
  
echo $resultado;  
  
?>
```

# CICLOS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $num = 0;

      while($num < 10 ) {
        echo $num++;
      }
    ?>
  </body>
</html>
```

0123456789

# CICLOS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      for($a = 1; $a <= 10; $a++){
        $cubo = $a * $a * $a;
        echo "O cubo de $a é $cubo<br>";
      }
    ?>
  </body>
</html>
```

O cubo de 1 é 1  
O cubo de 2 é 8  
O cubo de 3 é 27  
O cubo de 4 é 64  
O cubo de 5 é 125  
O cubo de 6 é 216  
O cubo de 7 é 343  
O cubo de 8 é 512  
O cubo de 9 é 729  
O cubo de 10 é 1000

# CICLOS

```
<ul>
<?php
$nomes = ['Ana', 'Maria', 'Miguel', 'José'];

foreach($nomes as $nome) {
    echo "<li>".$nome."</li>";
}

?>

</ul>
```

- Ana
- Maria
- Miguel
- José

# DATAS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");
      print_r(getdate());
    ?>
  </body>
</html>
```

Array ( [seconds] => 49 [minutes] => 10 [hours] => 19  
[mday] => 1 [wday] => 3 [mon] => 5 [year] => 2019 [yday]  
=> 120 [weekday] => Wednesday [month] => May [0] =>  
1556734249 )

# DATAS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");

      $agora = getdate();
      $ano = $agora["year"];
      $mes = $agora["mon"];
      $dia = $agora["mday"];

      echo $dia."/".$mes."/".$ano;
    ?>
  </body>
</html>
```

1/5/2019

# DATAS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      date_default_timezone_set("Europe/Lisbon");
      setlocale(LC_TIME, "pt-PT");

      $ano = strftime('%G');
      $mes = strftime('%B');
      $dia = strftime('%d');

      echo $dia."/".$mes."/".$ano;
    ?>
  </body>
</html>
```

01/maio/2019

# DATAS

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $data1 = new DateTime('2019-02-10');
      $data2 = new DateTime('2019-02-14');

      $diferenca = $data2->diff($data1);
    ?>

    <pre>
      <?php print_r($diferenca); ?>
    </pre>
  </body>
</html>
```

```
          DateInterval Object
(
  [y] => 0
  [m] => 0
  [d] => 4
  [h] => 0
  [i] => 0
  [s] => 0
  [f] => 0
  [weekday] => 0
  [weekday_behavior] => 0
  [first_last_day_of] => 0
  [invert] => 1
  [days] => 4
  [special_type] => 0
  [special_amount] => 0
  [have_weekday_relative] => 0
  [have_special_relative] => 0
)
```

# FUNÇÕES PHP

- `phpversion();` → mostra a versão do PHP
- `strtolower($var);` → transforma string para minúsculas
- `ucfirst($var);` → transforma o 1º char em maiúsculas
- `strip_tags($input);` → retira as tags html de uma string

# FUNÇÕES

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario(){
        return 100*3;
      }

      echo calcularSalario();
    ?>
  </body>
</html>
```

300

# FUNÇÕES

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario($salario,$ndias){
        return $salario * $ndias;
      }

      echo calcularSalario(100,30);
    ?>
  </body>
</html>
```

3000

# FUNÇÕES

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      function calcularSalario($salario,$ndias){
        $euro = $salario * $ndias;
        $dolar = $salario * $ndias * 1.2;

        return array($euro,$dolar);
      }

      $salarioFinal = calcularSalario(100,30);

      echo $salarioFinal[0] . " em euros <br>";
      echo $salarioFinal[1] . " em dolares <br>";
    ?>
  </body>
</html>
```

3000 em euros  
3600 em dolares

# ARRAYS

```
<?php  
$array = array(  
    "foo" => "bar",  
    "bar" => "foo",  
);  
  
// a partir do PHP 5.4  
$array = [  
    "foo" => "bar",  
    "bar" => "foo",  
];  
?>
```

# ARRAYS

```
<?php  
    $salada=array("Alface", "Cenoura", "Cebola");  
?  
  
<!DOCTYPE html>  
<html>  
    <body>  
        <?php  
            echo $salada[0] . "<br>";  
            echo $salada[2] . "<br>";  
        ?>  
    </body>  
</html>
```

Alface  
Cebola

# ARRAYS

```
<?php  
$array[ ] = $var;  
?>
```

```
<?php  
    $salada=array("Alface", "Cenoura", "Cebola");  
?  
<!DOCTYPE html>  
<html>  
<body>  
<?php  
    echo $salada[0] . "<br>";  
    echo $salada[2] . "<br>";  
    echo count($salada) . "<br>";  
    $salada[]="Rúcula";  
    echo count($salada) . "<br>";  
    echo $salada[3] . "<br>";  
?  
</body>  
</html>
```

Alface  
Cebola  
3  
4  
Rúcula

# ARRAYS

```
<?php
    $salada=array("Alface", "Cenoura", "Cebola");
?>

<!DOCTYPE html>
<html>
    <body>
        <pre>
            <?php
                print_r($salada);
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[0] => Alface  
[1] => Cenoura  
[2] => Cebola  
)

# ARRAYS UNSET

```
<?php
$arr1 = array(
    'geeks', // [0]
    'for', // [1]
    'geeks' // [2]
);

// remove item at index 1 which is 'for'
unset($arr1[1]);

// Print modified array
var_dump($arr1);

// Re-index the array elements
$arr2 = array_values($arr1);

// Print re-indexed array
var_dump($arr1);
?>
```

# ARRAYS SPLICE

```
<?php
$arr1 = array(
    'geeks', // [0]
    'for', // [1]
    'geeks' // [2]
);

// remove item at index 1 which is 'for'
array_splice($arr1, 1, 1);

// Print modified array
var_dump($arr1);
?>
```

Edit Co

# ARRAYS ASSOCIATIVOS

```
<?php
    $agenda=array(
        "nome" => "José",
        "sobrenome" => "Silva",
        "telefone" => 222333444
    );
?>

<!DOCTYPE html>
<html>
    <body>
        <pre>
            <?php
                print_r($agenda);
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[ nome ] => José  
[ sobrenome ] => Silva  
[ telefone ] => 222333444  
)

# ARRAYS ASSOCIATIVOS

```
<?php
    $agenda=array(
        "nome" => "José",
        "sobrenome" => "Silva",
        "telefone" => 222333444
    );
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            echo $agenda["nome"];
        ?>
    </body>
</html>
```

José

# FUNÇÕES DE ARRAYS

```
<?php  
    $nums=array(10,5,3,44,5,88,7);  
?  
  
<!DOCTYPE html>  
<html>  
    <body>  
        <?php  
            echo min($nums) . "<br>";  
            echo max($nums) . "<br>";  
            echo array_sum($nums) . "<br>";  
        ?>  
    </body>  
</html>
```

3

88

162

# FUNÇÕES DE ARRAYS

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            sort($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[0] => 3  
[1] => 5  
[2] => 5  
[3] => 7  
[4] => 10  
[5] => 44  
[6] => 88  
)

# FUNÇÕES DE ARRAYS

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            rsort($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[0] => 88  
[1] => 44  
[2] => 10  
[3] => 7  
[4] => 5  
[5] => 5  
[6] => 3  
)

# FUNÇÕES DE ARRAYS

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            shuffle($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[0] => 5  
[1] => 10  
[2] => 44  
[3] => 7  
[4] => 88  
[5] => 3  
[6] => 5  
)

# FUNÇÕES DE ARRAYS

```
<?php
    $nums=array(10,5,3,44,5,88,7);
?>

<!DOCTYPE html>
<html>
    <body>
        <?php
            shuffle($nums);
        ?>

        <pre>
            <?php
                print_r($nums)
            ?>
        </pre>
    </body>
</html>
```

Array  
(  
[0] => 5  
[1] => 10  
[2] => 44  
[3] => 7  
[4] => 88  
[5] => 3  
[6] => 5  
)

# SERVER-SIDE INCLUDES

- Vantagens:
  - Ficheiros são unidos pelo lado Servidor
  - Ficheiros externos podem conter PHP, HTML ou texto
  - Reutilização de Código
  - Elementos comuns de uma página são atualizados apenas editando um ficheiro

# SERVER-SIDE INCLUDES

- Comandos:
  - include
  - include\_once
  - require
  - require\_once

## SERVER-SIDE INCLUDES

- include - tenta continuar o processamento da página mesmo se faltar o ficheiro
- require - para o processamento se o ficheiro externo não está disponível
- once - previne que o ficheiro seja incluído mais que uma vez

# SERVER-SIDE INCLUDES

## Exemplo

```
<?php  
    include "variaveis.php";  
    echo $a;  
    echo $b;  
?>
```

## Arquivo "variaveis.php"

```
<?php  
    echo "Arquivo contendo as variáveis";  
    $a = 10;  
    $b = 20;  
?>
```

# SUPER GLOBAIS

- São tipos de variáveis especiais em PHP que significam que estão sempre acessíveis, independentemente do scope do script, função, class ou ficheiro.
- Exemplos:
  - `$GLOBALS`
  - `$_SERVER`
  - `$_REQUEST`
  - `$_POST`
  - `$_GET`
  - `$_FILES`
  - `$_ENV`
  - `$_COOKIE`
  - `$_SESSION`

# SUPER GLOBAIS

- A variável `$GLOBALS` em php é uma superglobal que é usada para aceder a todas as variáveis globais.
- PHP armazena todas as variáveis globais num array `$GLOBALS[index]`. O index será o nome da variável.
- `print_r($GLOBALS); //apresenta todas as superglobals`

# SUPER GLOBAIS

- A variável `$_SERVER` em php é uma superglobal que é usada para aceder a informações sobre headers, paths e localizações de scripts

# SUPER GLOBAIS

- A variável `$_REQUEST` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona para os 2 métodos GET e POST.

# SUPER GLOBAIS

- A variável `$_POST` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona apenas para o método POST dos formulários

# SUPER GLOBAIS

- A variável `$_GET` em php é uma superglobal que é usada para aceder a informações depois de submeter um formulário.
- Funciona apenas para o método GET dos formulários.
- Pode ser usada para recolher informação de um URL

## SUPER GLOBAIS

- A variável `$_SESSION` em php é uma superglobal que é usada para aceder e armazenar informação entre várias páginas.
- Ao contrário de cookie, a informação não fica armazenada no computador cliente.
- Por defeito, duram até o utilizador fechar o browser.

# SUPER GLOBAIS

- Um sessão começa com a função session\_start();
- Variáveis de sessão são superglobais de php : \$\_SESSION
- Exemplo:

```
session_start();
$_SESSION['login'] = true;
```

# SUPER GLOBAIS

- Para remover todas as variáveis de sessão : `session_unset();`
- Para destruir a sessão : `session_destroy();`

## SUPER GLOBAIS

- A variável `$_COOKIE` em php é uma superglobal que é usada para identificar um utilizador
- Um cookie é um pequeno ficheiro que o servidor embebe no computador do utilizador.
- De cada vez que o mesmo computador faz um request da página via browser, também vai enviar uma cookie.

# SUPER GLOBAIS

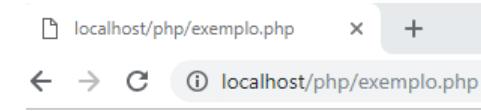
- Um cookie é criado com a função setcookie()
- Sintaxe
  - setcookie(name, value, expire, path, domain, secure, httponly);
- Apenas o parâmetro do nome é necessário, os restantes parâmetros são opcionais.

## GET / POST

- As variáveis GET e POST são consideradas superglobais, ou seja funcionam entre páginas permitindo a passagem de valores entre elas.
- No caso do GET as variáveis são enviadas no URL de destino.
- No caso do POST as variáveis não são visíveis.

# GET

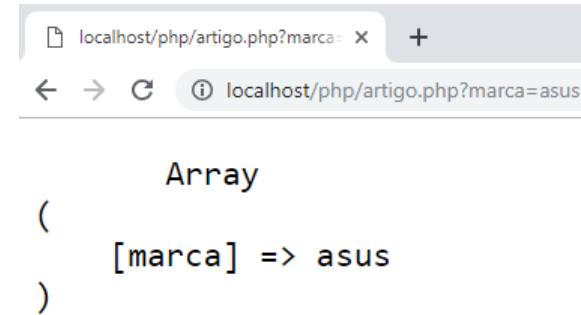
```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <ul>
      <li>
        <a href="artigo.php?marca=asus">Asus</a>
      </li>
      <li>
        <a href="artigo.php?marca=hp">HP</a>
      </li>
    </ul>
  </body>
</html>
```



- [Asus](#)
- [HP](#)

## GET

```
<!DOCTYPE html>
<html>
  <body>
    <pre>
      <?php
        print_r($_GET);
      ?>
    </pre>
  </body>
</html>
```

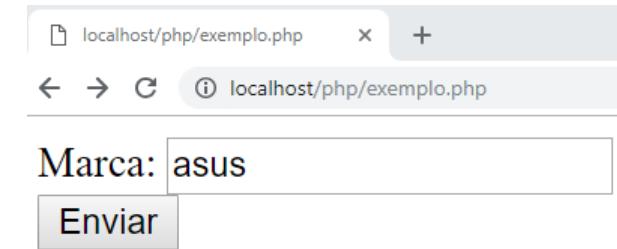


localhost/php/artigo.php?marca=asus

Array  
(  
 [marca] => asus  
)

# POST

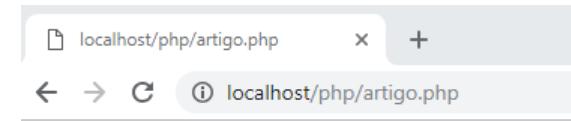
```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <form action="artigo.php" method="post">
      <label for="marca">Marca: </label>
      <input type="text" name="marca" id="marca">
      <br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```



A screenshot of a web browser window. The address bar shows "localhost/php/exemplo.php". The page content is a form with a label "Marca:" and a text input field containing "asus". Below the input field is a submit button labeled "Enviar".

# POST

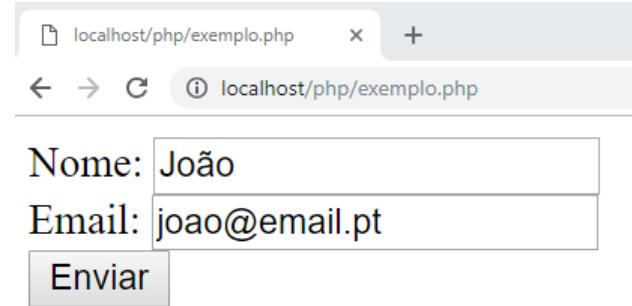
```
<!DOCTYPE html>
<html>
  <body>
    <pre>
      <?php
        print_r($_POST);
      ?>
    </pre>
  </body>
</html>
```



```
Array
(
    [marca] => asus
)
```

# FORM

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <form action="artigo.php" method="post">
      <label for="marca">Nome: </label>
      <input type="text" name="nome" id="nome">
      <br>
      <label for="marca">Email: </label>
      <input type="text" name="email" id="email">
      <br>
      <input type="submit" value="Enviar">
    </form>
  </body>
</html>
```

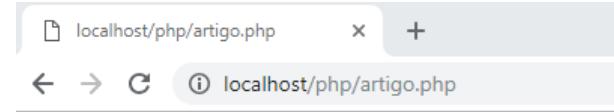


A screenshot of a web browser window titled "localhost/php/exemplo.php". The page displays a simple HTML form. The "Nome:" label has a red box around it, and the "Email:" label has a blue box around it. The "Enviar" button is also highlighted with a red box.

# FORM

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $nome=$_POST["nome"];
      $email=$_POST["email"];

      echo "Bem-vindo ".$nome." - ".$email;
    ?>
  </body>
</html>
```



Bem-vindo João - joao@email.pt

## LINKS DINÂMICOS

- Pretendemos agora criar uma página com uma lista de artigos. Ao clicar num artigo devemos ser redirecionados para a página com os detalhes do artigo. Isto obrigar-nos-ia a criar uma página para cada artigo... mas com PHP podemos usar apenas 1 página e preenche-la consoante o artigo clicado.

# LINKS DINÂMICOS

exemplo.php

```
<!DOCTYPE html>
<html>
  <body>
    <ul>
      <li>
        <a href="artigo.php?cod=1">Computador</a>
      </li>
      <li>
        <a href="artigo.php?cod=2">Smartphone</a>
      </li>
      <li>
        <a href="artigo.php?cod=3">Caderno</a>
      </li>
    </ul>
  </body>
</html>
```

- [Computador](#)
- [Smartphone](#)
- [Caderno](#)

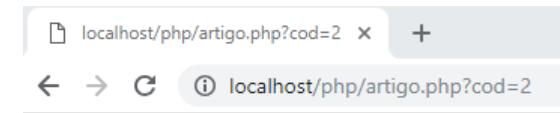
# LINKS DINÂMICOS

artigo.php

```
<!DOCTYPE html>
<html>
  <body>

  <?php
    $artigos = array("Computador", "Smartphone", "Caderno");
    $codigo = $_GET["cod"];
    echo $artigos[$codigo];
  ?>

  </body>
</html>
```



Caderno

## PHP MAILER

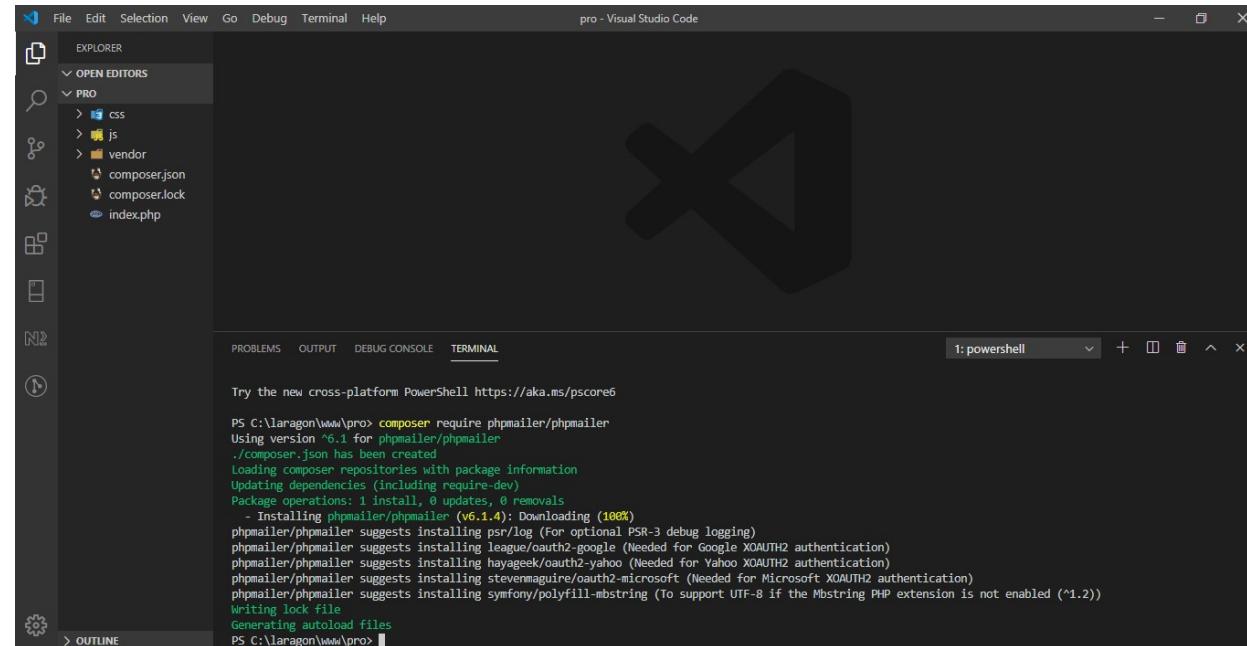
- Pacote de PHP que acrescenta a possibilidade de enviar emails a partir de um formulário PHP

<https://github.com/PHPMailer/PHPMailer>

- Abrindo a pasta do nosso projeto devemos agora no terminal executar o seguinte comando para instalar o PHPMailer:

composer require phpmailer/phpmailer

# PHPMAILER



The screenshot shows a Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project structure with files like index.php, composer.json, and composer.lock. The Terminal tab is active at the bottom, displaying the output of a Composer command:

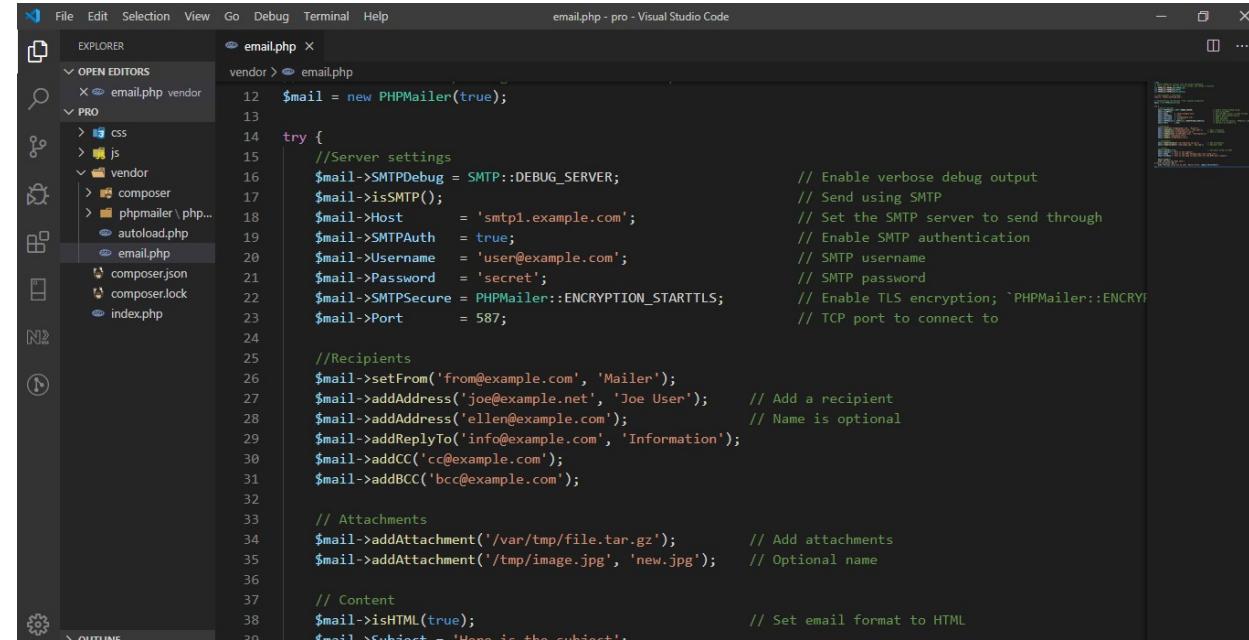
```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\laragon\www\pro> composer require phpmailer/phpmailer
Using version ^6.1 for phpmailer/phpmailer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing phpmailer/phpmailer (v6.1.4): Downloading (100%)
    phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
    phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 authentication)
    phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 authentication)
    phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microsoft XOAUTH2 authentication)
    phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2))
  Writing lock file
  Generating autoload files
PS C:\laragon\www\pro>
```

## PHP MAILER

- Dentro da pasta vendor vamos ter acesso à pasta com o PHP Mailer.
- Vamos agora voltar ao repositório do Github do PHP Mailer e copiar para um ficheiro o conteúdo de “A Simple Example”.

# PHP MAILER



A screenshot of the Visual Studio Code interface. The title bar reads "email.php - pro - Visual Studio Code". The left sidebar shows a file tree with "OPEN EDITORS" containing "email.php" and "vendor", and a "PRO" section containing "css", "js", "vendor", "composer", "phpmailer\php...", "autoload.php", "email.php", "composer.json", "composer.lock", and "index.php". The main editor area displays the following PHP code:

```
12 $mail = new PHPMailer(true);
13
14 try {
15     //Server settings
16     $mail->SMTPDebug = SMTP::DEBUG_SERVER;           // Enable verbose debug output
17     $mail->isSMTP();                                // Send using SMTP
18     $mail->Host = 'smtp1.example.com';               // Set the SMTP server to send through
19     $mail->SMTPAuth = true;                          // Enable SMTP authentication
20     $mail->Username = 'user@example.com';           // SMTP username
21     $mail->Password = 'secret';                     // SMTP password
22     $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; // Enable TLS encryption; `PHPMailer::ENCRYPTION_SMTPS` also accepted
23     $mail->Port = 587;                             // TCP port to connect to
24
25     //Recipients
26     $mail->setFrom('from@example.com', 'Mailer');
27     $mail->addAddress('joe@example.net', 'Joe User'); // Add a recipient
28     $mail->addAddress('ellen@example.com');           // Name is optional
29     $mail->addReplyTo('info@example.com', 'Information');
30     $mail->addCC('cc@example.com');
31     $mail->addBCC('bcc@example.com');
32
33     // Attachments
34     $mail->addAttachment('/var/tmp/file.tar.gz');      // Add attachments
35     $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name
36
37     // Content
38     $mail->isHTML(true);                            // Set email format to HTML
39     $mail->Subject = 'Here is the subject'.
```

## PHP MAILER

- Preenchemos agora os campos “host”, “username” e “password” para configuração da conta.
- Preencher os campos da zona “Receipts” (basta setFrom e setAddress)
- Neste exemplo vamos apagar a parte dos “Attachments” uma vez que não vamos anexar qualquer ficheiro ao email
- E finalmente alterar o “Content” do email (podemos retirar o AltBody).

## PHP MAILER

- Tomando em consideração que vamos usar uma conta Gmail temos de:
  - Ativar o IMAP
  - Permitir o acesso a aplicações externas não verificadas

# PHPMAILER

## Definições



Geral Marcadores Caixa de entrada Contas e importação Filtros e endereços bloqueados **Encaminhamento e POP/IMAP** Suplementos

Chat Avançadas Offline Temas

**Encaminhamento:**  
[Saiba mais](#)

Adicionar um endereço de encaminhamento

Sugestão: Pode também encaminhar apenas algumas das suas mensagens ao [criar um filtro!](#)

**Transferência via POP:**  
[Saiba mais](#)

1. **Estado:** O **POP foi ativado** para todo o correio que chegou desde 14/02/08

- Ativar POP para **todo o correio** (mesmo correio que já tenha sido transferido)
- A partir de agora, ativar o POP para **correio a receber**
- Desativar POP**

2. **Quando as mensagens são acedidas com POP** [Guarde uma cópia do Gmail na caixa de entrada](#) ▾

3. **Configure o seu cliente de email** (por exemplo, Outlook, Eudora, Netscape Mail)  
[Instruções de configuração](#)

**Acesso IMAP:**  
(acesso a Gmail por parte de outros clientes que utilizem o IMAP)

**Estado:** O **IMAP está ativado**

- Ativar IMAP**
- Desativar IMAP**

# PHPMAILER

Acesso a aplicações menos seguras

Para proteger a sua conta, as aplicações e os dispositivos que utilizam tecnologia de início de sessão menos segura estão bloqueados. Para manter a sua conta segura, a Google irá desativar automaticamente esta definição se não estiver a ser utilizada. [Saiba mais](#)



Desativado

[Ativar o acesso \(não recomendado\)](#)

<https://myaccount.google.com/security>

# PHP MAILER

```
$mail->SMTPDebug = SMTP::DEBUG_SERVER;
$mail->isSMTP();
$mail->Host      = 'smtp.gmail.com';
$mail->SMTPAuth  = true;
$mail->Username   = 'brunofrs7@gmail.com';
$mail->Password   = '';
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
$mail->Port       = 587;

//Recipients
$mail->setFrom('brunofrs7@gmail.com', 'Bruno');
$mail->addAddress('brunofrs7@gmail.com', 'Teste');

// Content
$mail->isHTML(true);
$mail->Subject = 'Assunto do Email';
$mail->Body    = 'Corpo do email';
```

# PHPMAILER

```
2020-02-16 18:09:00 SERVER -> CLIENT: 220 smtp.gmail.com ESMTP s139sm17624839wme.35 - gsmtp
2020-02-16 18:09:00 CLIENT -> SERVER: EHLO localhost
2020-02-16 18:09:00 SERVER -> CLIENT: 250-smtp gmail.com at your service, [176.79.185.60]250-SIZE 35882577250-8BITM1
2020-02-16 18:09:00 CLIENT -> SERVER: STARTTLS
2020-02-16 18:09:00 SERVER -> CLIENT: 220 2.0.0 Ready to start TLS
2020-02-16 18:09:00 CLIENT -> SERVER: EHLO localhost
2020-02-16 18:09:00 SERVER -> CLIENT: 250-smtp gmail.com at your service, [176.79.185.60]250-SIZE 35882577250-8BITM1
CHUNKING250 SMTPUTF8
2020-02-16 18:09:00 CLIENT -> SERVER: AUTH LOGIN
2020-02-16 18:09:00 SERVER -> CLIENT: 334 VXNlcmbWU6
2020-02-16 18:09:00 CLIENT -> SERVER: [credentials hidden]
2020-02-16 18:09:00 SERVER -> CLIENT: 334 UGFzc3dvcmQ6
2020-02-16 18:09:00 CLIENT -> SERVER: [credentials hidden]
2020-02-16 18:09:01 SERVER -> CLIENT: 235 2.7.0 Accepted
2020-02-16 18:09:01 CLIENT -> SERVER: MAIL FROM:<brunofrs7@gmail.com>
2020-02-16 18:09:01 SERVER -> CLIENT: 250 2.1.0 OK s139sm17624839wme.35 - gsmtp
2020-02-16 18:09:01 CLIENT -> SERVER: RCPT TO:<brunofrs7@gmail.com>
2020-02-16 18:09:01 SERVER -> CLIENT: 250 2.1.5 OK s139sm17624839wme.35 - gsmtp
2020-02-16 18:09:01 CLIENT -> SERVER: DATA
2020-02-16 18:09:01 SERVER -> CLIENT: 354 Go ahead s139sm17624839wme.35 - gsmtp
2020-02-16 18:09:01 CLIENT -> SERVER: Date: Sun, 16 Feb 2020 18:09:00 +0000
2020-02-16 18:09:01 CLIENT -> SERVER: To: Teste <brunofrs7@gmail.com>
2020-02-16 18:09:01 CLIENT -> SERVER: From: Bruno <brunofrs7@gmail.com>
2020-02-16 18:09:01 CLIENT -> SERVER: Subject: Assunto do Email
2020-02-16 18:09:01 CLIENT -> SERVER: Message-ID: <WSJf0xIl0OpT6QZ6Vj21XuTMVZ9JUhxFsief0ac@localhost>
2020-02-16 18:09:01 CLIENT -> SERVER: X-Mailer: PHPMailer 6.1.4 (https://github.com/PHPMailer/PHPMailer)
2020-02-16 18:09:01 CLIENT -> SERVER: MIME-Version: 1.0
2020-02-16 18:09:01 CLIENT -> SERVER: Content-Type: text/html; charset=iso-8859-1
2020-02-16 18:09:01 CLIENT -> SERVER:
2020-02-16 18:09:01 CLIENT -> SERVER: Corpo do email
2020-02-16 18:09:01 CLIENT -> SERVER:
2020-02-16 18:09:02 SERVER -> CLIENT: 250 2.0.0 OK 1581876541 s139sm17624839wme.35 - gsmtp
2020-02-16 18:09:02 CLIENT -> SERVER: QUIT
2020-02-16 18:09:02 SERVER -> CLIENT: 221 2.0.0 closing connection s139sm17624839wme.35 - gsmtp
Message has been sent
```