

FUNDAÇÃO VALEPARAIBANA DE ENSINO
COLÉGIOS UNIVAP – UNIDADE CENTRO

CURSO TÉCNICO EM INFORMÁTICA

RODRIGO RODA OLIVETO ALVES

LISTA DE EXERCÍCIOS 2º BIMESTRE
PROGRAMAÇÃO AVANÇADA PARA WEB

Lista apresentada ao Curso Técnico de informática
como composição de nota.
Prof. Me. Hélio Lourenço Esperidião Ferreira

SÃO JOSÉ DOS CAMPOS
2023

PROGRAMAÇÃO AVANÇADA PARA WEB

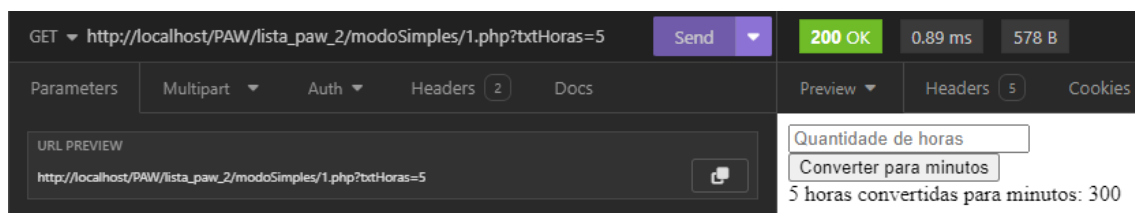
LISTA DE EXERCÍCIOS II – PHP BÁSICO

1 - Faça todos os exercícios com seus respectivos formulários e não utilize programação orientada a objetos.

1. Converter uma quantidade de horas digitadas pelo usuário em minutos. Informe o resultado em minutos.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX1</title>
</head>
<body>
  <form action="" method="get">
    <input type="number" name="txtHoras" min="1" placeholder="Quantidade de
horas"><br>
    <input type="submit" value="Converter para minutos"><br>

    <?php
      if(isset($_GET['txtHoras'])){
        $horas = $_GET['txtHoras'];
        $minutos = $horas * 60;
        if ($horas != "") {
          echo "$horas horas convertidas para minutos: $minutos";
        }
      }
    ?>
  </form>
</body>
</html>
```



2. Crie um programa que dada a idade de uma pessoa calcule quantos dias, horas, minutos e segundo essa pessoa já viveu

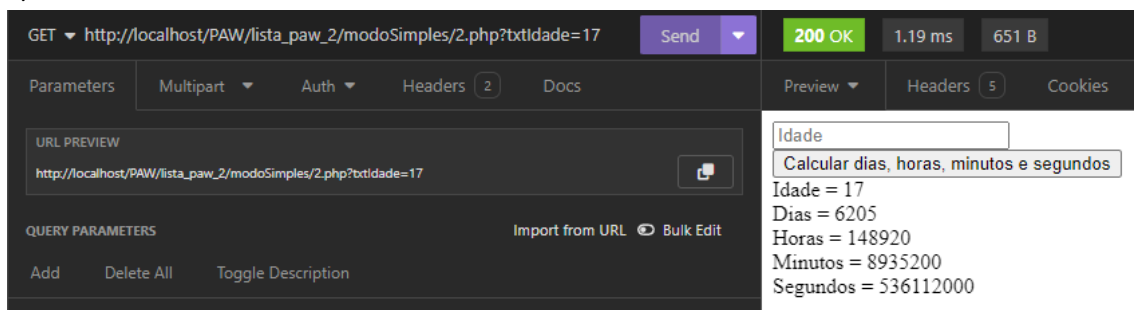
```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX2</title>
</head>
<body>
<form action="" method="get">
  <input type="number" name="txtIdade" min="1" placeholder="Idade"><br>
  <input type="submit" value="Calcular dias, horas, minutos e segundos"><br>

  <?php
    if (isset($_GET['txtIdade'])){
      $idade= $_GET['txtIdade'];
      $dias = $idade * 365;
      $horas = $dias * 24;
      $minutos = $horas * 60;
      $segundos = $minutos * 60;

      echo"Idade = $idade<br>";
      echo"Dias = $dias<br>";
      echo"Horas = $horas<br>";
      echo"Minutos = $minutos<br>";
      echo"Segundos = $segundos<br>";
    }
  ?>
</form>
</body>
</html>

```



3. Crie um programa que leia duas notas de um aluno e apresente a média

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

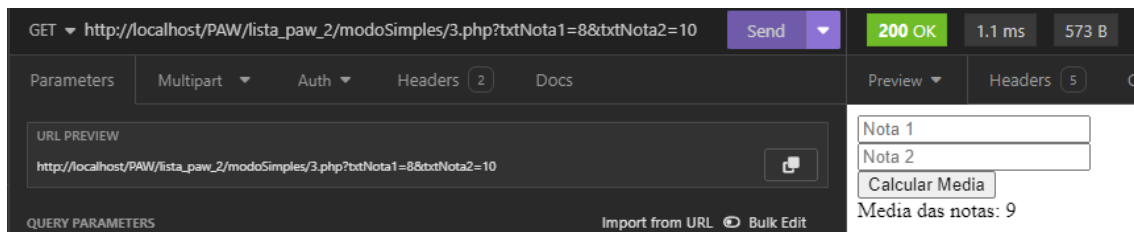
<title>EX3</title>
</head>
<body>
  <form action="" method="get">
    <input type="number" name="txtNota1" placeholder="Nota 1"><br>
    <input type="number" name="txtNota2" placeholder="Nota 2"><br>
    <input type="submit" value="Calcular Media"><br>

    <?php
      if(isset($_GET['txtNota1']) && isset($_GET['txtNota2'])){
        $nota1 = $_GET['txtNota1'];
        $nota2 = $_GET['txtNota2'];
        $media = ($nota1 + $nota2) / 2;

        echo "Media das notas: $media";
      }

    ?>
  </form>
</body>
</html>

```



4. Uma imobiliária vende terrenos retangulares. Faça um programa para ler as dimensões de um terreno e depois exibir a área e comprimento.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX4</title>
</head>

```

```

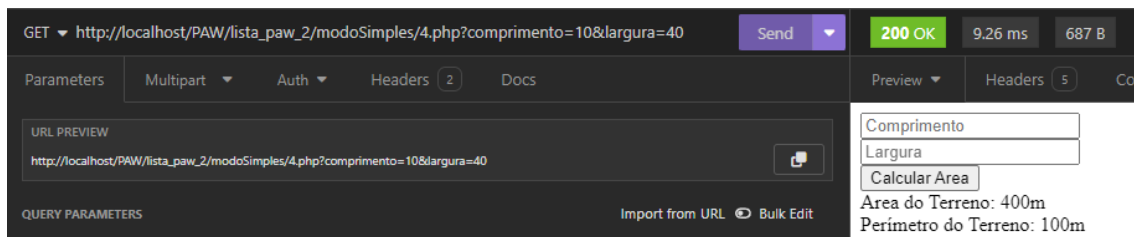
<body>
  <form action="" method="get">
    <input type="number" name="comprimento" placeholder="Comprimento"><br>
    <input type="number" name="largura" placeholder="Largura"><br>
    <input type="submit" value="Calcular Area"><br>

    <?php
      if(isset($_GET['comprimento']) && isset($_GET['largura'])){
        $comprimento = $_GET['comprimento'];
        $largura = $_GET['largura'];
        $area = $comprimento * $largura;
        $perimetro = $comprimento*2 + $largura*2;

        echo"Area do Terreno: $area".'.m<br>';
        echo"Perímetro do Terreno: $perimetro".'.m';
      }

    ?>
  </form>
</body>
</html>

```



5. Escreva um programa para ler o número total de eleitores de um município, o número de votos brancos, nulos e válidos. Calcular e escrever o percentual que cada um representa em relação ao total de eleitores.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX5</title>
</head>
<body>
  <form action="" method="get">
    <input type="number" name="totalEleitores" placeholder="Total de Eleitores"><br>
    <input type="number" name="votosBrancos" placeholder="Votos Brancos"><br>
    <input type="number" name="votosNulos" placeholder="Votos Nulos"><br>
    <input type="number" name="votosValidos" placeholder="Votos Validos"><br>
    <input type="submit" value="Calcular Percentual"><br><br>

```

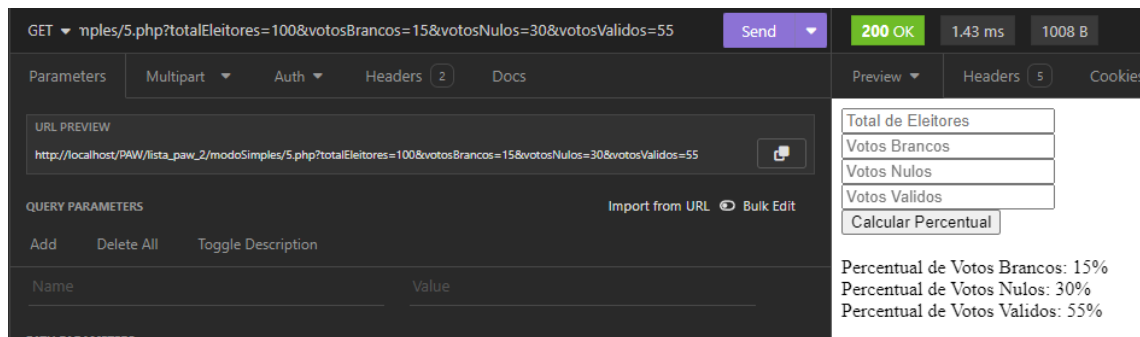
```

<?php
    if(isset($_GET['totalEleitores']) && isset($_GET['votosBranco']) &&
isset($_GET['votosNulos']) && isset($_GET['votosValidos'])) {
        $totalEleitores = $_GET['totalEleitores'];
        $pctBranco = ($_GET['votosBranco']/ $totalEleitores)*100;
        $pctNulos = ($_GET['votosNulos']/ $totalEleitores)*100;
        $pctValidos = ($_GET['votosValidos']/ $totalEleitores)*100;

        echo "Percentual de Votos Branco: $pctBranco<br>";
        echo "Percentual de Votos Nulos: $pctNulos<br>";
        echo "Percentual de Votos Validos: $pctValidos<br>";
    }

?>
</form>
</body>
</html>

```



6. Faça um programa que calcule e mostre valores de graus Celsius e graus Fahrenheit, cujos graus variem de 40 a 70 F de 1 em 1. A conversão de graus Fahrenheit (F) para graus Celsius (C) é dada por: $C = \frac{5}{9} * (F - 32)$.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX6</title>
</head>
<body>
    <table border="1">
        <?php
            for ($i = 40; $i <= 70 ; $i++){
                $far = $i;
                $cel = 5/9*($far-32);
            }
        </?php>
    </table>
</body>
</html>

```

```

        echo"<tr>";
        echo"<td>$far °F";
        echo"<td>$cel °C";
        echo"</tr>";
    }
    ?>
</table>
</body>
</html>

```

40 °F	4.444444444444 °C
41 °F	5 °C
42 °F	5.555555555556 °C
43 °F	6.111111111111 °C
44 °F	6.666666666667 °C
45 °F	7.222222222222 °C
46 °F	7.777777777778 °C
47 °F	8.333333333333 °C
48 °F	8.888888888889 °C
49 °F	9.444444444444 °C
50 °F	10 °C
51 °F	10.555555555556 °C
52 °F	11.111111111111 °C
53 °F	11.666666666667 °C
54 °F	12.222222222222 °C
55 °F	12.777777777778 °C
56 °F	13.333333333333 °C
57 °F	13.888888888889 °C
58 °F	14.444444444444 °C
59 °F	15 °C
60 °F	15.555555555556 °C
61 °F	16.111111111111 °C
62 °F	16.666666666667 °C
63 °F	17.222222222222 °C
64 °F	17.777777777778 °C
65 °F	18.333333333333 °C

7. Faça um programa que calcule e mostre a área de um trapézio. Sabe-se que: $A = (base\ maior + base\ menor) * altura / 2$;

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX7</title>
</head>
<body>
    <form action="" method="get">
        <input type="number" name="baseMenor" placeholder="Base Menor"><br>

```

```

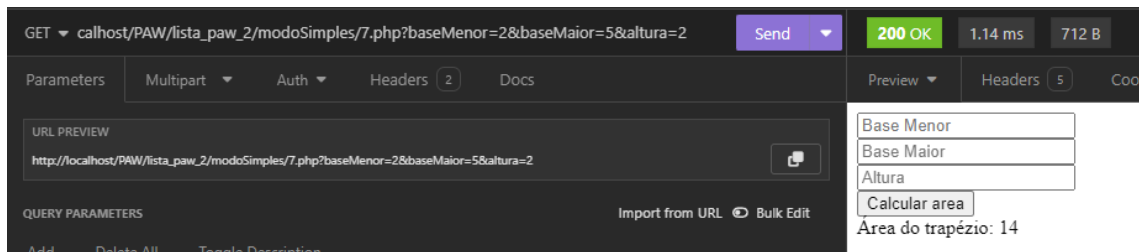
<input type="number" name="baseMaior" placeholder="Base Maior"><br>
<input type="number" name="altura" placeholder="Altura"><br>
<input type="submit" value="Calcular area">

```

```

<?php
    if(isset($_GET['baseMenor']) && isset($_GET['baseMaior']) && isset($_GET['altura'])) {
        $baseMenor = $_GET['baseMenor'];
        $baseMaior = $_GET['baseMaior'];
        $altura = $_GET['altura'];
        $area = ($baseMenor+$baseMaior)*$altura;
        echo"<br>Área do trapézio: $area";
    }
?>
</form>
</body>
</html>

```



8. Calcular o salário líquido do funcionário sabendo que este é constituído pelo salário bruto mais o valor das horas extras subtraindo 8% de INSS do total. Serão lidos nesse problema o salário bruto, o valor das horas extras e o número de horas extras. Apresentar ao final o salário líquido.

```

<html>
<head>
<title>EX8</title>
</head>
<body>
<form action="" method="get">
<input type="number" name="salarioBruto" placeholder="Digite seu salário bruto"><br>
<input type="number" name="qtdHorasExtras" placeholder="Digite quantidade de horas extras"><br>
<input type="number" name="valorHorasExtras" placeholder="Digite valor da hora extra"><br>
<input type="submit" value="Calcular salário líquido"><br><br>

<?php
    if(isset($_GET['salarioBruto']) && isset($_GET['qtdHorasExtras']) &&
isset($_GET['valorHorasExtras'])) {
        $salarioBruto = $_GET['salarioBruto'];
        $qtdHorasExtras = $_GET['qtdHorasExtras'];

```

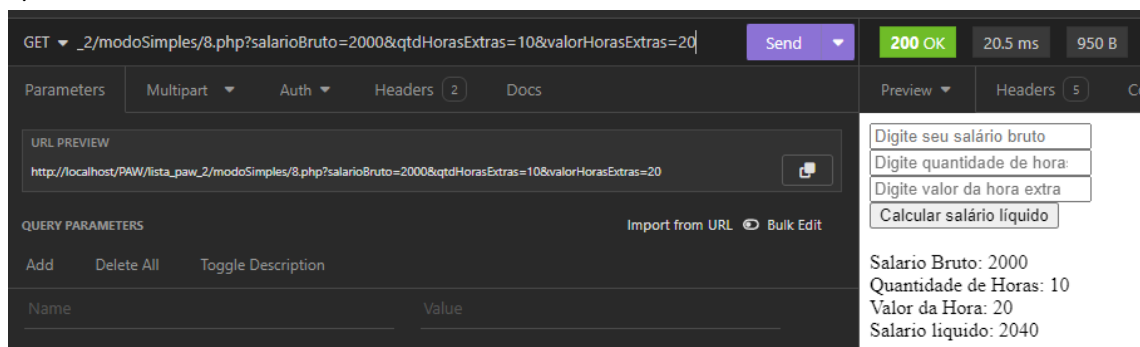


```

        $valorHorasExtras = $_GET['valorHorasExtras'];
        $INSS = 0.08;

        $salarioLiquido = $salarioBruto + ($qtdHorasExtras*$valorHorasExtras) -
($salarioBruto*$INSS);
        echo"Salario Bruto: $salarioBruto<br>";
        echo"Quantidade de Horas: $qtdHorasExtras<br>";
        echo"Valor da Hora: $valorHorasExtras<br>";
        echo"Salario liquido: $salarioLiquido<br>";
    }
    ?>
</form>
</body>
</html>

```



9. Efetuar a leitura do número de quilowatts consumido e calcular o valor a ser pago de energia elétrica, sabendo-se que o valor a pagar por quilowatt é de 0,12. Apresentar o valor total a ser pago pelo usuário acrescido de 18% de ICMS.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX9</title>
</head>
<body>
    <form action="" method="get">
        <input type="number" name="consumoKW" placeholder="Digite o consumo em
kW"><br>
        <input type="submit" value="Calcular valor a pagar"><br><br>

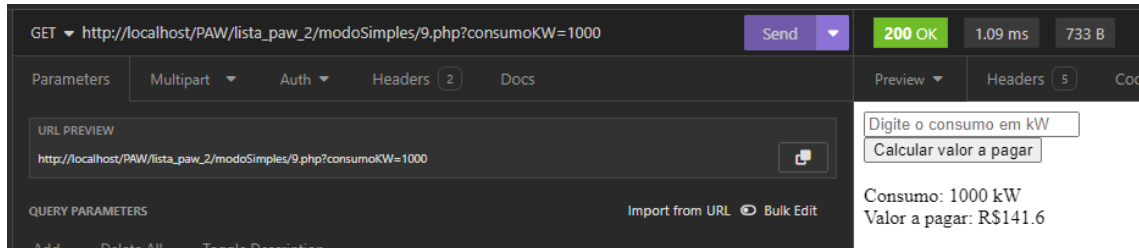
    <?php
        if(isset($_GET['consumoKW'])){
            $consumo = $_GET['consumoKW'];
            $valorPagar = $consumo * 0.12;
            $valorPagar = $valorPagar*1.18;

```

```

        echo"Consumo: $consumo kW<br>";
        echo"Valor a pagar: R$ $valorPagar";
    }
    ?>
</form>
</body>
</html>

```



10. Calcular a média de combustível gasto pelo usuário, sendo informado a quantidade de quilômetros rodados e a quantidade de combustível.

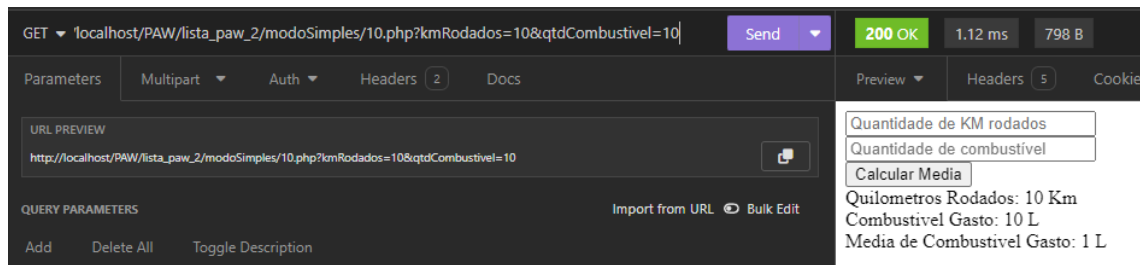
```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX10</title>
</head>
<body>
    <form action="" method="get">
        <input type="number" name="kmRodados" placeholder="Quantidade de KM rodados"
        style="width:200px"><br>
        <input type="number" name="qtdCombustivel" placeholder="Quantidade de
        combustível" style="width:200px"><br>
        <input type="submit" value="Calcular Media"><br>

    <?php
        if(isset($_GET['kmRodados']) && isset($_GET['qtdCombustivel'])) {
            $kmRodados = $_GET['kmRodados'];
            $qtdCombustivel = $_GET['qtdCombustivel'];
            $mediaCombustivelGasto = $kmRodados/$qtdCombustivel;

            echo"Quilometros Rodados: $kmRodados Km<br>";
            echo"Combustivel Gasto: $qtdCombustivel L<br>";
            echo"Media de Combustivel Gasto: $mediaCombustivelGasto L<br>";
        }
        ?>
    </form>
</body>
</html>

```

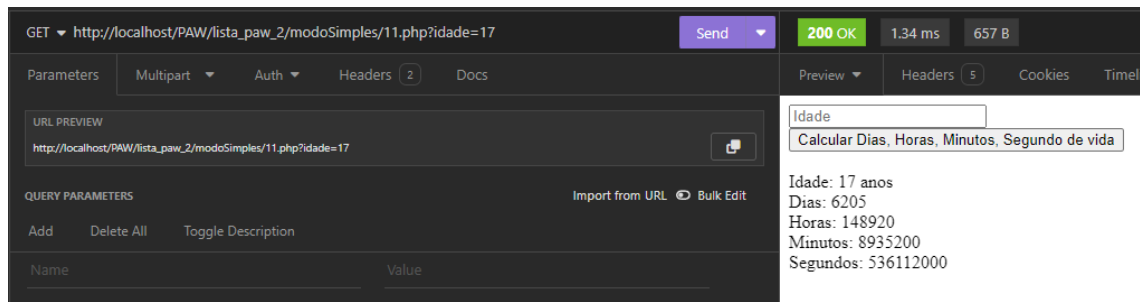


11. Crie um programa que dada a idade de uma pessoa calcule quantos dias, horas, minutos e segundo essa pessoa já viveu.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX11</title>
</head>
<body>
  <form action="" method="get">
    <input type="number" name="idade" placeholder="Idade"><br>
    <input type="submit" value="Calcular Dias, Horas, Minutos, Segundo de vida"><br><br>

    <?php
      if(isset($_GET['idade'])){
        $idade = $_GET['idade'];
        $dias = $idade*365;
        $horas = $dias*24;
        $minutos = $horas*60;
        $segundos = $minutos*60;

        echo"Idade: $idade anos<br>";
        echo"Dias: $dias<br>";
        echo"Horas: $horas<br>";
        echo"Minutos: $minutos<br>";
        echo"Segundos: $segundos<br>";
      }
    ?>
  </form>
</body>
</html>
```



12. Faça um algoritmo que calcule e mostre valores de graus Celsius e graus Fahrenheit, cujos graus variem de 40 a 70 F de 1 em 1. A conversão de graus Fahrenheit (F) para graus Celsius (C) é dada por: $C = \frac{5}{9} * (F - 32)$.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX6</title>
</head>
<body>
  <table border="1">
    <?php
      for ($i = 40; $i <= 70 ; $i ++){
        $far = $i;
        $cel = 5/9*($far-32);

        echo"<tr>";
        echo"<td>$far ºF";
        echo"<td>$cel ºC";
        echo"</tr>";
      }
    ?>
  </table>
</body>
</html>
```

GET

http://localhost/PAW/lista_paw_2/modoSimples/12.php

Send

200 OK

9.66 ms

1746 B

Parameters

Multipart

Auth

Headers2

Docs

Preview

Headers5

Cool

URL PREVIEW

http://localhost/PAW/lista_paw_2/modoSimples/12.php

QUERY PARAMETERS

Import from URL

Bulk Edit

Add

Delete All

Toggle Description

Name

Value

PATH PARAMETERS

Path parameters are url path segments that start with a colon ':' e.g. 'id'

40 °F	4.444444444444 °C
41 °F	5 °C
42 °F	5.555555555556 °C
43 °F	6.111111111111 °C
44 °F	6.666666666667 °C
45 °F	7.222222222222 °C
46 °F	7.777777777778 °C
47 °F	8.333333333333 °C
48 °F	8.888888888889 °C
49 °F	9.444444444444 °C
50 °F	10 °C
51 °F	10.555555555556 °C
52 °F	11.111111111111 °C
53 °F	11.666666666667 °C
54 °F	12.222222222222 °C
55 °F	12.777777777778 °C
56 °F	13.333333333333 °C
57 °F	13.888888888889 °C
58 °F	14.444444444444 °C
59 °F	15 °C
60 °F	15.555555555556 °C
61 °F	16.111111111111 °C
62 °F	16.666666666667 °C
63 °F	17.222222222222 °C
64 °F	17.777777777778 °C
65 °F	18.333333333333 °C
66 °F	18.888888888889 °C

13. Uma fábrica de refrigerantes Meia-Cola vende seu produto em três formatos: lata de 350 ml, garrafa de 600 ml e garrafa de 2 litros. Se um comerciante compra uma determinada quantidade de cada formato, faça um algoritmo para calcular quantos litros de refrigerante ele comprou.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX13</title>
</head>
<body>
  <form action="" method="get">
    Latas de 350ml:
    <input type="number" name="latas350"><br>
    Garrafas de 600ml:
    <input type="number" name="garrafas600"><br>
    Garrafas de 2L:
    <input type="number" name="garrafas2L"><br>
    <input type="submit" value="Calcular Litros"><br><br>

  <?php

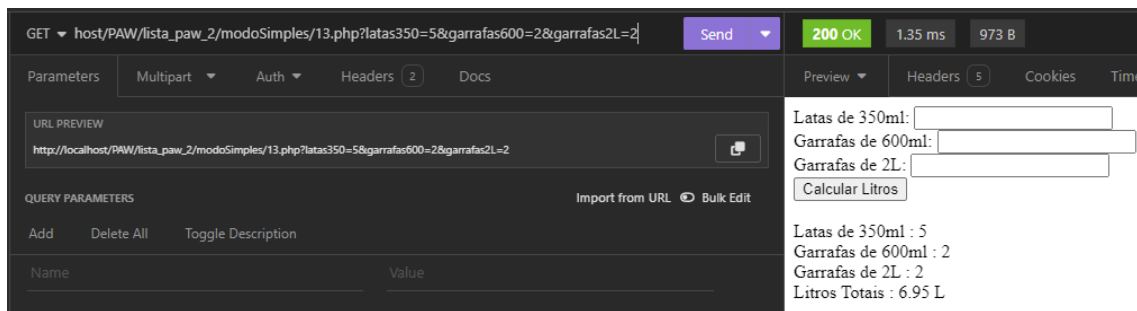
```

```

if(isset($_GET['latas350']) && isset($_GET['garrafas600']) && isset($_GET['garrafas2L']))){
    $latas350 = $_GET['latas350'];
    $garrafas600 = $_GET['garrafas600'];
    $garrafas2L = $_GET['garrafas2L'];

    $litrosTotais = ((($latas350*350) + ($garrafas600*600) + ($garrafas2L*2000))/1000;
    echo"Latas de 350ml : $latas350<br>";
    echo"Garrafas de 600ml : $garrafas600<br>";
    echo"Garrafas de 2L : $garrafas2L<br>";
    echo"Litros Totais : $litrosTotais L<br>";
}
?>
</form>
</body>
</html>

```



14. Construa um programa que leia três notas de um aluno, calcule a média obtida por este aluno e no final escreva o resultado indicando se o mesmo foi **aprovado** ou **reprovado** (considere que aluno aprovado obteve Média $\geq 7,0$ e aluno reprovado Média $< 7,0$).

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX14</title>
</head>
<body>
    <form action="" method="get">
        <input type="number" name="nota1" placeholder="Nota 1"><br>
        <input type="number" name="nota2" placeholder="Nota 2"><br>
        <input type="number" name="nota3" placeholder="Nota 2"><br>
        <input type="submit" value="Calcular Media"><br><br>

    <?php

```

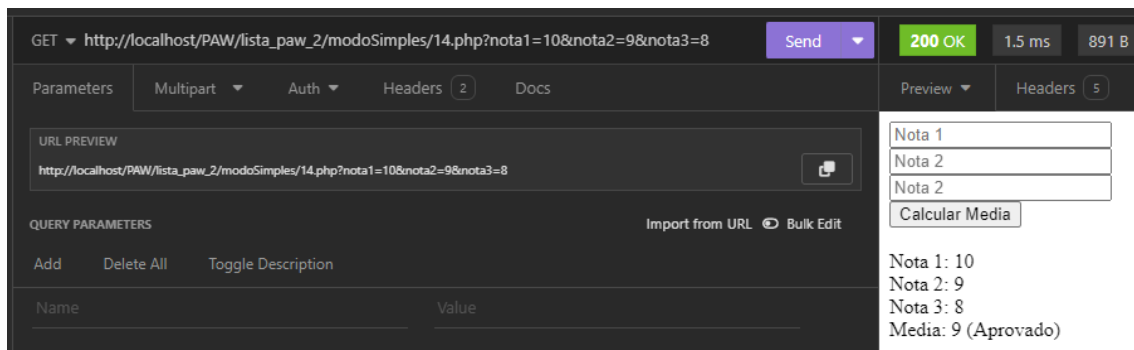
```

if(isset($_GET['nota1']) && isset($_GET['nota2']) && isset($_GET['nota3'])){
    $nota1 = $_GET['nota1'];
    $nota2 = $_GET['nota2'];
    $nota3 = $_GET['nota3'];
    $media = ($nota1 + $nota2 + $nota3)/3;

    if($media >= 7){
        $aprovacao = 'Aprovado';
    }else{
        $aprovacao = 'Reprovado';
    }

    echo"Nota 1: $nota1<br>";
    echo"Nota 2: $nota2<br>";
    echo"Nota 3: $nota3<br>";
    echo"Media: $media ($aprovacao)<br>";
}
?>
</form>
</body>
</html>

```



15. Elabore um algoritmo que dada a idade de um nadador classifique-o em uma das seguintes categorias:

Infantil A = 5 a 7 anos

Infantil B = 8 a 11 anos

Juvenil A = 12 a 13 anos

Juvenil B = 14 a 17 anos

Adultos = Maiores de 18 anos

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX15</title>
</head>

```

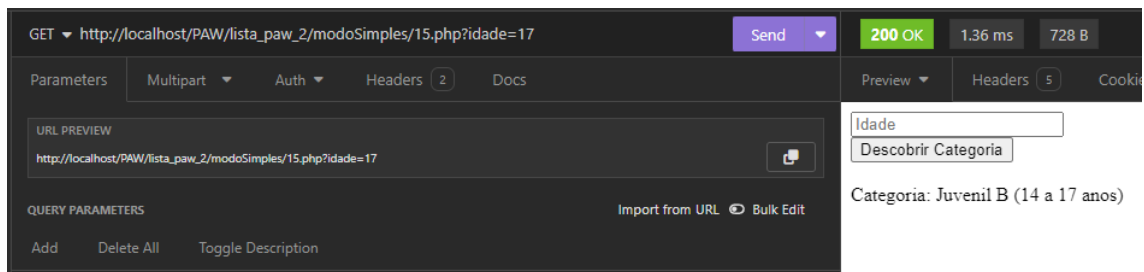
```

<body>
  <form action="" method="get">
    <input type="number" name="idade" placeholder="Idade"><br>
    <input type="submit" value="Descobrir Categoria"><br><br>
  </form>

  <?php
    if(isset($_GET['idade'])){
      $idade = $_GET['idade'];
      if($idade <= 0){
        $categoria = 'Digite valor adequado!';
      }elseif($idade > 0 && $idade < 4){
        $categoria = 'Fora dos Limites';
      }elseif($idade >= 5 && $idade <=7){
        $categoria = 'Infantil A (5 a 7 anos)<br>';
      }elseif($idade >= 8 && $idade <=11){
        $categoria = 'Infantil B (7 a 11 anos)<br>';
      }elseif($idade >= 12 && $idade <=13){
        $categoria = 'Juvenil A (12 a 13 anos)<br>';
      }elseif($idade >= 14 && $idade <=17){
        $categoria = 'Juvenil B (14 a 17 anos)<br>';
      }elseif($idade > 18){
        $categoria = 'Adultos (Maiores de 18)<br>';
      }

      echo"Categoria: $categoria ";
    }
  ?>
</body>
</html>

```



16. Elabore um algoritmo para calcular a equação do segundo grau e imprima as raízes da equação na tela, os valores a, b e c são inseridos pelo usuário: $ax^2 + bx + c = 0$

$$\Delta = b^2 - 4ac, x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

```

<!DOCTYPE html>
<html lang="en">
<head>

```



```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>EX16</title>
</head>
<body>
<form action="" method="get">
    Equacao Modelo:  $ax^2 + bx + c = 0$ <br>
    <input type="number" name="a" placeholder="a"><br>
    <input type="number" name="b" placeholder="b"><br>
    <input type="number" name="c" placeholder="c"><br>
    <input type="submit" value="Calcular"><br><br>

<?php
    if(isset($_GET['a']) && isset($_GET['b']) && isset($_GET['c'])){
        $a = $_GET['a'];
        $b = $_GET['b'];
        $c = $_GET['c'];
        $delta = ($b**2) - (4*$a*$c);
        if($delta < 0){
            echo"A equacao nao possui raizes<br>";
        }elseif($delta == 0){
            echo"A equacao possui apenas uma raiz<br>";
            $x = -$b/2*$a;
            echo"Raiz da equacao: $x";
        }elseif($delta > 0){
            echo"A equacao possui mais de uma raiz<br>";
            $x1 = (-$b + sqrt($delta))/2*$a;
            $x2 = (-$b - sqrt($delta))/2*$a;
            echo"Raizes da equacao: $x1 e $x2";
        }
    }
?>

</form>
</body>
</html>

```

GET http://localhost/PAW/lista_paw_2/modoSimples/16.php?a=1&b=3&c=2 Send 200 OK 1.42 ms 837 B

Parameters Multipart Auth Headers 2 Docs Preview Headers 5 Cookies

URL PREVIEW
http://localhost/PAW/lista_paw_2/modoSimples/16.php?a=1&b=3&c=2

QUERY PARAMETERS Import from URL Bulk Edit

Add	Delete All	Toggle Description	Name	Value

Equacao Modelo: $ax^2 + bx + c = 0$

a

b

c

A equacao possui mais de uma raiz
 Raizes da equacao: -1 e -2

17. Construa um programa que leia nome de um funcionário, o número de horas trabalhadas, o valor que recebe por horas trabalhadas e o número de filhos, com estas informações, calcular o salário deste funcionário, considerando que o mesmo terá uma gratificação de 3% sobre o salário bruto por cada filho, caso o mesmo possua acima de três filhos. Escreva ao final, o nome do funcionário, seu respectivo salário e o acréscimo de salário, caso ela tenha tido direito a esta gratificação.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX17</title>
</head>
<body>
  <form action="" method="get">
    <input type="text" name="nomeFuncionario" placeholder="Nome"><br>
    <input type="number" name="horasTrabalhadas" placeholder="Horas Trabalhadas"><br>
    <input type="number" name="valorHorasTrabalhadas" placeholder="Valor de Cada Hora
Trabalhada"><br>
    <input type="number" name="numeroDeFilhos" placeholder="Quantidade de
Filhos"><br>
    <input type="submit" value="Calcular Salário"><br><br>

  <?php
    if(isset($_GET['nomeFuncionario']) && isset($_GET['horasTrabalhadas']) &&
isset($_GET['valorHorasTrabalhadas']) && isset($_GET['numeroDeFilhos'])){
      $nome = $_GET['nomeFuncionario'];
      $horasTrabalhadas = $_GET['horasTrabalhadas'];
      $valorHorasTrabalhadas = $_GET['valorHorasTrabalhadas'];
      $numeroDeFilhos = $_GET['numeroDeFilhos'];

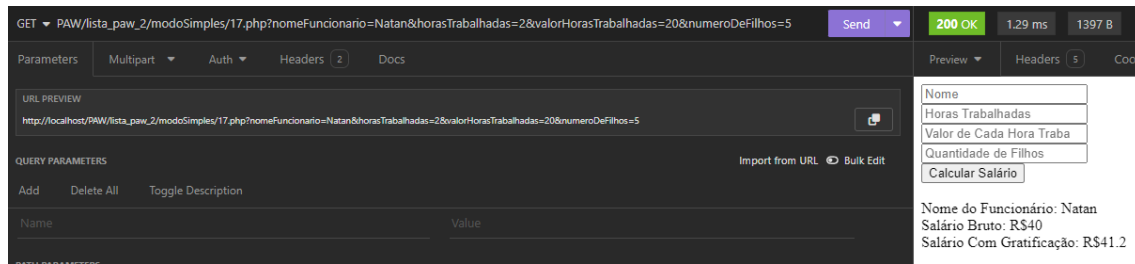
      $salario = $horasTrabalhadas*$valorHorasTrabalhadas;

      echo"
Nome do Funcionário: $nome<br>
Salário Bruto: R$$salario<br>
";
      if($numeroDeFilhos > 3){
        $salario = $salario*1.03;
        echo"Salário Com Gratificação: R$$salario";
      }
    }
  ?>
```

```

</form>
</body>
</html>

```



18. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:

- a. Para homens: $(72.7h * h) - 58$
- b. Para mulheres: $(62.1 * h) - 44.7$

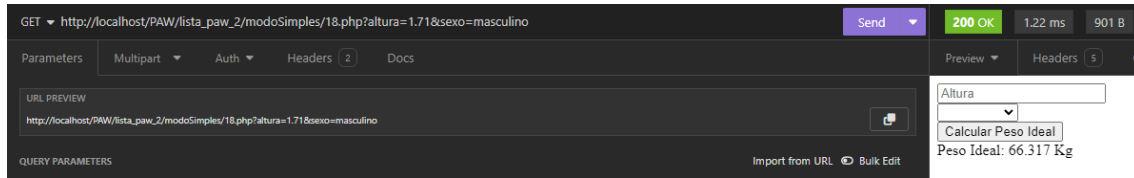
```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX18</title>
</head>
<body>
  <form action="" method="get">
    <input type="text" name="altura" placeholder="Altura"><br>
    <select name="sexo">
      <option value=""></option>
      <option value="masculino">Masculino</option><br>
      <option value="feminino">Feminino</option><br>
    </select><br>
    <input type="submit" value="Calcular Peso Ideal"><br>

    <?php
      if(isset($_GET['altura']) && isset($_GET['sexo'])){
        $sexo = $_GET['sexo'];
        $altura = $_GET['altura'];
        if($sexo == 'masculino'){
          $pesoIdeal = (72.7*$altura) - 58;
        }elseif($sexo == 'feminino'){
          $pesoIdeal = (62.1*$altura) - 44.7;
        }
        echo "Peso Ideal: $pesoIdeal Kg<br>";
      }
    ?>
  </form>
</body>

```

</html>



19. Construa programa que leia três lados de um triângulo, verifique e escreva que tipo de triângulo eles formam (considere **triângulo equilátero** com três lados iguais, **triângulo isósceles** com dois lados iguais e **triângulo escaleno** com todos os lados diferentes).

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>EX19</title>
```

```
</head>
```

```
<body>
```

```
  <form action="" method="get">
```

```
    Digite os lados de um Triangulo: <br>
```

```
    <input type="number" name="lado1"><br>
```

```
    <input type="number" name="lado2"><br>
```

```
    <input type="number" name="lado3"><br>
```

```
    <input type="submit" value="Classificar Triangulo"><br>
```

```
<?php
```

```
  if(isset($_GET['lado1']) && isset($_GET['lado2']) && isset($_GET['lado3'])){\
```

```
    $lado1 = $_GET['lado1'];
```

```
    $lado2 = $_GET['lado2'];
```

```
    $lado3 = $_GET['lado3'];
```

```
    if(($lado1 == $lado2 && $lado3 != $lado1) || ($lado1 == $lado3 && $lado1 != $lado2) || ($lado2 == $lado3 && $lado2 != $lado1)){
```

```
      echo"Triangulo Isocetes<br>";
```

```
    }elseif($lado1 != $lado2 && $lado1 != $lado3 && $lado2 != $lado3){
```

```
      echo"Triangulo Escaleno<br>";
```

```
    }elseif($lado1 == $lado2 && $lado1 == $lado3){
```

```
      echo"Trinagulo Equilatero<br>";
```

```
    }
```

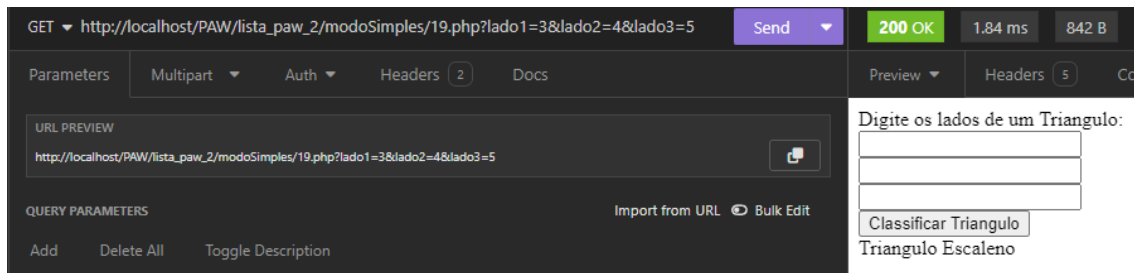
```
  }
```

```
?>
```

```
</form>
```

```
</body>
```

```
</html>
```



20. Construa programa para determinar se o indivíduo esta com um peso favorável. Essa situação é determinada através do IMC (Índice de Massa Corpórea), que é definida como sendo a relação entre o peso (PESO) e o quadrado da Altura (ALTURA) do indivíduo. Ou seja

$$IMC = \frac{PESO}{ALTURA^2}$$

Escreva na tela de acordo com as seguintes situações:

Condição	Situação
IMC abaixo de 20	Abaixo do peso
IMC de 20 até 25	Peso Normal
IMC de 25 até 30	Sobre Peso
IMC de 30 até 40	Obeso
IMC de 40 e acima	Obeso Mórbido

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>EX20</title>
</head>
<body>
  <form action="" method="get">
    <input type="number" name="peso" placeholder="Peso"><br>
    <input type="text" name="altura" placeholder="Altura"><br>
    <input type="submit" value="Calcular IMC"><br>

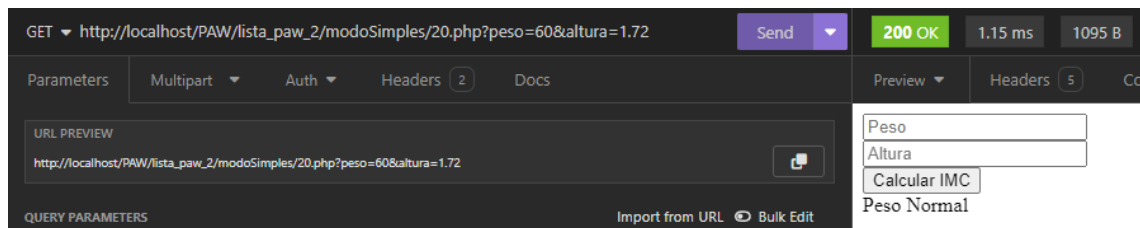
  <?php
    if(isset($_GET['peso']) && isset($_GET['altura'])){
      $peso = $_GET['peso'];
      $altura = $_GET['altura'];
      $imc = $peso/$altura**2;

      if($imc < 20){
        echo"Abaixo do Peso<br>";
      }elseif($imc >= 20 && $imc <= 25){
        echo"Peso Normal<br>";
      }
    }
  }
  </?php>
</body>
</html>
```

```

    }elseif($imc >= 25 && $imc <= 30){
        echo"Sobre Peso<br>";
    }elseif($imc >= 30 && $imc < 40){
        echo"Obeso<br>";
    }elseif($imc >= 40){
        echo"Obeso Morbido<br>";
    }
}
?>
</form>
</body>
</html>

```



21. Construa programa que leia três lados de um triângulo, verifique e escreva que tipo de triângulo eles formam (considere **triângulo equilátero** com três lados iguais, **triângulo isósceles** com dois lados iguais e **triângulo escaleno** com todos os lados diferentes).

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>EX21</title>
</head>
<body>
    <form action="" method="get">
        Digite os lados de um Triangulo: <br>
        <input type="number" name="lado1"><br>
        <input type="number" name="lado2"><br>
        <input type="number" name="lado3"><br>
        <input type="submit" value="Classificar Triangulo"><br>

    <?php
        if(isset($_GET['lado1']) && isset($_GET['lado2']) && isset($_GET['lado3'])){
            $lado1 = $_GET['lado1'];
            $lado2 = $_GET['lado2'];
            $lado3 = $_GET['lado3'];

```

```


        if(($lado1 == $lado2 && $lado3 != $lado1) || ($lado1 == $lado3 && $lado1 != $lado2)
        || ($lado2 == $lado3 && $lado2 != $lado1)){
            echo"Triangulo Isoceles<br>";
        }elseif($lado1 != $lado2 && $lado1 != $lado3 && $lado2 != $lado3){
            echo"Triangulo Escaleno<br>";
        }elseif($lado1 == $lado2 && $lado1 == $lado3){
            echo"Trinagulo Equilatero<br>";
        }
    }
    ?>
</form>
</body>
</html>


```

GET http://localhost/PAW/lista_paw_2/modoSimples/21.php?lado1=3&lado2=4&lado3=5 Send 200 OK 0.88 ms 842 B

Parameters Multipart Auth Headers 2 Docs Preview Headers 5 Code

URL PREVIEW

http://localhost/PAW/lista_paw_2/modoSimples/21.php?lado1=3&lado2=4&lado3=5 

QUERY PARAMETERS Import from URL  Bulk Edit

Add Delete All Toggle Description

Digite os lados de um Triangulo:

Classificar Triangulo

Triangulo Escaleno

**CONSTRUA APIS E UTILIZE ORIENTAÇÃO A OBJETOS
CASO O ALUNO NÃO UTILIZE OS PADRÕES DE ORIENTAÇÃO A OBJETOS, MVC
E REST A LISTA SERÁ DESCONSIDERADA.**

Apresente além do código fonte os prints do funcionamento no Insomnia.

Código 'index.php' para os exercícios do 22 ao 34:

```
<?php
$metodo = $_SERVER['REQUEST_METHOD'];
$rota = explode("/", $_SERVER['REQUEST_URI']);
// echo $_SERVER['REQUEST_URI'];
// print_r($rota)
if($metodo == "GET"){
    if ($rota[1] == "horas"){
        if(isset($rota[2])){
            if($rota[2] == "minutos"){
                if(isset($rota[3])){
                    if(is_numeric($rota[3])){
                        // echo "tem variavel";
                        require_once "control/controle_Horas_Minutos.php";
                    }
                }
            }elseif($rota[2] == "segundos"){
                if(isset($rota[3])){
                    if(is_numeric($rota[3])){
                        require_once "control/controle_Horas_Segundos.php";
                    }
                }
            }elseif(is_numeric($rota[2])){
                require_once "control/controle_Horas.php";
            }
        }
    }elseif($rota[1] == "idades"){
        if(isset($rota[2])){
            if($rota[2] == "dias"){
                if(isset($rota[3])){
                    if(is_numeric($rota[3])){
                        require_once "control/controle_Idade_Dias.php";
                    }
                }
            }elseif($rota[2] == "horas"){
                if(isset($rota[3])){
                    if(is_numeric($rota[3])){
                        require_once "control/controle_Idade_Horas.php";
                    }
                }
            }elseif($rota[2] == "minutos"){
```



```

        if(isset($rota[3])){
            if(is_numeric($rota[3])){
                require_once "control/control_Idade_Minutos.php";
            }
        }
    }elseif($rota[2] == "segundos"){
        if(isset($rota[3])){
            if(is_numeric($rota[3])){
                require_once "control/control_Idade_Segundos.php";
            }
        }
    }elseif(is_numeric($rota[2])){
        require_once "control/control_Idade.php";
    }
}

}elseif($rota[1] == "alunos"){
    if(isset($rota[2])){
        if($rota[2] == "notas"){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    if(isset($rota[4])){
                        if(is_numeric($rota[4])){
                            require_once "control/control_Alunos_Media.php";
                        }
                    }
                }else{
                    require_once "control/control_Alunos_Aprovacao.php";
                }
            }
        }
    }
}

}elseif($rota[1] == "terrenos"){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    require_once "control/control_Terrenos_Area.php";
                }
            }
        }
    }
}

}elseif($rota[1] == "graus"){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){

```

```

        if(isset($rota[3])){
            if(is_numeric($rota[3])){
                require_once"control/control_Graus_Tabela.php";
            }
        }else{
            require_once"control/control_Graus.php";
        }
    }
}
}elseif ($rota[1] == 'trapezios'){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    if(isset($rota[4])){
                        if(is_numeric($rota[4])){
                            require_once "control/control_Trapezio_Area.php";
                        }
                    }
                }
            }
        }
    }
}
}
}elseif ($rota[1] == 'quilowatts'){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            require_once "control/control_Quilowatts_ValorPagar.php";
        }
    }
}
}elseif ($rota[1] == 'classificacoes'){
    if(isset($rota[2])){
        if($rota[2] == 'nadadores'){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    require_once"control/control_Classificacoes_Categoria.php";
                }
            }
        }
    }
}
}elseif ($rota[1] == 'bhaskaras'){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    if(isset($rota[4])){

```

```

        if(is_numeric($rota[4])){
            require_once"control/control_e_Bhaskaras_Raizes.php";
        }
    }
}
}
}
}
}
}elseif($rota[1] == "imc"){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    require_once"control/control_e_Calcular_IMC.php";
                }
            }
        }
    }
}
}elseif($rota[1] == 'triangulos'){
    if(isset($rota[2])){
        if(is_numeric($rota[2])){
            if(isset($rota[3])){
                if(is_numeric($rota[3])){
                    if(isset($rota[4])){
                        if(is_numeric($rota[4])){
                            require_once"control/control_e_Triangulos_Classificar.php";
                        }
                    }
                }
            }
        }
    }
}
}
}
}

}elseif($metodo == "POST"){
    if($rota[1] == "terrenos"){
        require_once"control/control_e_Terrenos_POST.php";
    }elseif($rota[1] == 'funcionarios'){
        if(isset($rota[2])){
            if($rota[2] == 'salarios'){
                require_once "control/control_e_Funcionarios_Salario.php";
            }
        }
    }elseif($rota[1] == 'combustiveis'){
        if(isset($rota[2])){

```

```

        if($rota[2] == 'medias'){
            require_once "control/control_Combustiveis_Media.php";
        }
    }
}
}
}

?>

```

22. Converter uma quantidade de horas digitadas pelo usuário em minutos. Informe o resultado em minutos. Crie a classe Horas (Horas.php)

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/horas/minutos/n	/horas/minutos/5	Transforma horas em minutos	{ "horas":5, "minutos":300 }
GET	/horas/segundos/n	/horas/segundos/5	Transforma horas em segundos	{ "horas":5, "segundos":18000 }
GET	/horas/n	/horas/5	Retorna horas em minutos e segundos	{ "horas":5, "minutos":300, "segundos":18000 }

Modelo 'Horas.php':

```

<?php
class Horas{
    private $horas;

    public function setHoras($v){
        $this->horas = $v;
    }
    public function getHoras(){
        return $this->horas;
    }
    public function converterParaMinutos(){
        return $this->horas * 60;
    }
    public function converterParaSegundos(){
        return $this->converterParaMinutos() * 60;
    }
}

?>

```

Controle 'controle_Horas_Minutos.php':

```
<?php
require_once "modelo/Horas.php";
//caminho absoluto funciona, porem realtivo nao
$rota = explode("/",$_SERVER['REQUEST_URI']);
// $rota[3] = n de horas

$horas = new Horas();
$horas->setHoras($rota[3]);

$resposta = array();
$resposta['horas'] = (float)$horas->getHoras();
$resposta['minutos'] = $horas->converterParaMinutos();
echo json_encode($resposta);

?>
```

Controle 'controle_Horas_Segundos.php':

```
<?php
require_once "modelo/Horas.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

$horas = new Horas();
$horas->setHoras($rota[3]);

$resposta = array();
$resposta['horas'] = (float)$horas->getHoras();
$resposta['segundos'] = $horas->converterParaSegundos();
echo json_encode($resposta);

?>
```

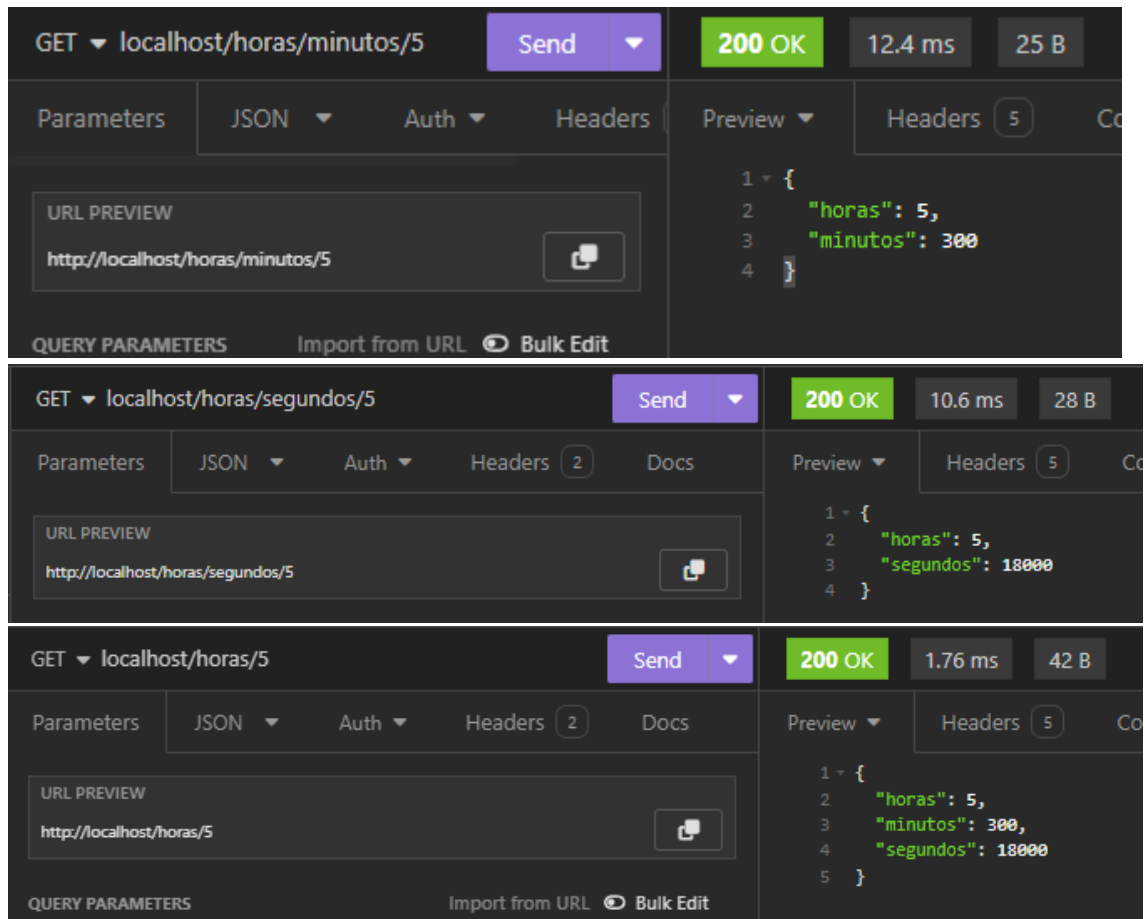
Controle 'controle_Horas.php':

```
<?php
require_once "modelo/Horas.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

$horas = new Horas();
$horas->setHoras($rota[2]);

$resposta = array();
$resposta['horas'] = (float)$horas->getHoras();
$resposta['minutos'] = $horas->converterParaMinutos();
$resposta['segundos'] = $horas->converterParaSegundos();
echo json_encode($resposta);

?>
```



23. Crie um programa que dada a idade de uma pessoa calcule quantos dias, horas, minutos e segundo essa pessoa já viveu

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/idades/dias/n	Idades/dias/1	Retorna quantos dias equivalem a 1 ano	{ "anos":1, "dias":365 }
GET	/idades/horas/n	/idades/horas/1	Retorna quantas horas equivalem a 1 ano	{ "anos":1, "horas":8760 }
GET	/idades/minutos/n	/idades/minutos/1	Retorna quantos minutos equivalem a 1 ano	{ "anos":1, "minutos":525600 }
GET	/idades/segundos/n	/idades/segundos/1	Retorna quantos segundos equivalem a 1 ano.	{ "anos":1, "segundos":31536000 }
GET	/idades/n	/idades/1	Retorna quantos dias, horas, minutos e segundos equivalem a 1 ano.	{ "anos":1, "dias":365, "horas":8760, "minutos":525600, "segundos":31536000 }

				<pre> “minutos”:525600, “segundos”:31536000 } </pre>
--	--	--	--	--

Modelo ‘Idades.php’:

```

<?php
class Idades{
    private $idade;

    public function setIdade($v){
        $this->idade = $v;
    }
    public function getIdade(){
        return $this->idade;
    }
    public function converterParaDias(){
        return $this->idade * 365;
    }
    public function converterParaHoras(){
        return $this->converterParaDias() * 24;
    }
    public function converterParaMinutos(){
        return $this->converterParaHoras() * 60;
    }
    public function converterParaSegundos(){
        return $this->converterParaMinutos() * 60;
    }
}
?>

```

Controle ‘controle_Idade_Dias.php’:

```

<?php
require_once "modelo/Idades.php";

$rota = explode("/", $_SERVER['REQUEST_URI']);
$idades = new Idades();
$idades -> setIdade($rota[3]);

$resposta = array();
$resposta['anos'] = (float)$idades->getIdade();
$resposta['dias'] = $idades->converterParaDias();
echo json_encode($resposta);
?>

```

Controle ‘controle_Idade_Horas.php’:

```

<?php
require_once "modelo/Idades.php";

```

```
$rota = explode("/", $_SERVER['REQUEST_URI']);
$idades = new Idades();
$idades -> setIdade($rota[3]);

$resposta = array();
$resposta['anos'] = (float)$idades->getIdade();
$resposta['horas'] = $idades->converterParaHoras();
echo json_encode($resposta);
```

?>

Controle 'controle_Idade_Minutos.php':

```
<?php
```

```
require_once "modelo/Idades.php";
```

```
$rota = explode("/", $_SERVER['REQUEST_URI']);
$idades = new Idades();
$idades -> setIdade($rota[3]);
```

```
$resposta = array();
$resposta['anos'] = (float)$idades->getIdade();
$resposta['minutos'] = $idades->converterParaMinutos();
echo json_encode($resposta);
```

?>

Controle 'controle_Idade_Segundos.php':

```
<?php
```

```
require_once "modelo/Idades.php";
```

```
$rota = explode("/", $_SERVER['REQUEST_URI']);
$idades = new Idades();
$idades -> setIdade($rota[3]);
```

```
$resposta = array();
$resposta['anos'] = (float)$idades->getIdade();
$resposta['segundos'] = $idades->converterParaSegundos();
echo json_encode($resposta);
```

?>

Controle 'controle_Idade.php':

```
<?php
```

```
require_once "modelo/Idades.php";
```

```
$rota = explode("/", $_SERVER['REQUEST_URI']);
```



```
$idades = new Idades();  
$idades -> setIdade($rota[2]);
```

```
$resposta = array();  
$resposta['anos'] = (float)$idades->getIdade();  
$resposta['dias'] = $idades->converterParaDias();  
$resposta['horas'] = $idades->converterParaHoras();  
$resposta['minutos'] = $idades->converterParaMinutos();  
$resposta['segundos'] = $idades->converterParaSegundos();  
echo json_encode($resposta);
```

?>

The image displays three sequential screenshots of a REST client interface, showing the results of GET requests to different endpoints. Each screenshot includes the method (GET), the URL, the response status (200 OK), the response time, and the response body in JSON format.

First Screenshot: GET localhost/idades/dias/1

- Status: 200 OK
- Time: 15.1 ms
- Size: 21 B
- Response Body:

```
{  
  "anos": 1,  
  "dias": 365  
}
```

Second Screenshot: GET localhost/idades/horas/1

- Status: 200 OK
- Time: 2.29 ms
- Size: 23 B
- Response Body:

```
{  
  "anos": 1,  
  "horas": 8760  
}
```

Third Screenshot: GET localhost/idades/minutos/1

- Status: 200 OK
- Time: 2.05 ms
- Size: 27 B
- Response Body:

```
{  
  "anos": 1,  
  "minutos": 525600  
}
```

GET localhost/idades/segundos/1

Send

200 OK1.81 ms30 B

ParametersJSONAuthHeaders2Docs

URL PREVIEW
http://localhost/idades/segundos/1

PreviewHeaders5

1 {
2 "anos": 1,
3 "segundos": 31536000
4 }

GET localhost/idades/1

Send

200 OK1.74 ms71 B

ParametersJSONAuthHeaders2Docs

URL PREVIEW
http://localhost/idades/1

QUERY PARAMETERS
Import from URL Bulk Edit

PreviewHeaders5

1 {
2 "anos": 1,
3 "días": 365,
4 "horas": 8760,
5 "minutos": 525600,
6 "segundos": 31536000
7 }

24. Crie um programa que leia duas notas de um aluno e apresente a média

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/alunos/nostas/n1/n2	/alunos/nostas/7.5/6.1	Retorna a média entre 7.5 e 6.1	{ "n1":7.5, "n2":6.1 "media":6.8, "conceito": "Aprovado" }
GET	/alunos/nostas/medias	/alunos/nostas/7.5	Retorna o conceito aprovado ou reprovado	{ "media":7.5, "conceito": "Aprovado" }

Modelo 'Alunos.php':

```
<?php
class Alunos{
    private $nota1;
    private $nota2;
    public function setNotas($v1,$v2){
        $this->nota1 = $v1;
        $this->nota2 = $v2;
    }

    public function setMedia($v1){
        $this->nota1 = $v1;
    }
    public function getNota1(){
        return $this->nota1;
    }
    public function getNota2(){
        return $this->nota2;
    }
    public function retornarMedia($v1,$v2){
        return ($v1+$v2)/2;
    }
    public function retornarAprovacao($v1,$v2){
        if($this->retornarMedia($v1,$v2) < 6){
            return "Reprovado";
        }else{
            return "Aprovado";
        }
    }
    public function retornarAprovacaoMedia($v1){
        if($v1 < 6){
            return "Reprovado";
        }else{
            return "Aprovado";
        }
    }
}
```

```
}  
?>
```

Controle 'controle_Alunos_Media.php':

```
<?php
```

```
require_once "modelo/Alunos.php";  
$rota = explode("/", $_SERVER['REQUEST_URI']);  
$alunos = new Alunos();  
$alunos->setNotas($rota[3], $rota[4]);  
  
$resposta = array();  
$resposta['n1'] = (float)$rota[3];  
$resposta['n2'] = (float)$rota[4];  
$resposta['media'] = $alunos->retornarMedia($resposta['n1'], $resposta['n2']);  
$resposta['conceito'] = $alunos->retornarAprovacao($resposta['n1'], $resposta['n2']);  
echo json_encode($resposta);  
?>
```

Controle 'controle_Alunos_Aprovacao.php':

```
<?php
```

```
require_once "modelo/Alunos.php";  
$rota = explode("/", $_SERVER['REQUEST_URI']);  
$alunos = new Alunos();  
$alunos->setMedia($rota[3]);  
  
$resposta = array();  
$resposta['media'] = (float)$rota[3];  
$resposta['conceito'] = $alunos->retornarAprovacaoMedia($rota[3]);  
echo json_encode($resposta);  
  
?>
```

The image displays two screenshots of a REST client interface, likely Postman, showing HTTP GET requests and their corresponding JSON responses.

Top Screenshot:

- Method: GET
- URL: localhost/alunos/notas/7.5/6.1
- Status: 200 OK
- Time: 1.95 ms
- Size: 53 B
- Response (JSON):

```
{  
  "n1": 7.5,  
  "n2": 6.1,  
  "media": 6.8,  
  "conceito": "Aprovado"  
}
```

Bottom Screenshot:

- Method: GET
- URL: localhost/alunos/notas/7.5
- Status: 200 OK
- Time: 2.44 ms
- Size: 35 B
- Response (JSON):

```
{  
  "media": 7.5,  
  "conceito": "Aprovado"  
}
```

25. Uma imobiliária vende terrenos retangulares. Faça um programa para ler as dimensões de um terreno e depois exibir a área e comprimento.

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	terrenos/base/altura	terrenos/10/20	Retorna área	{ "base": 10, "altura":20, "area":200 }
POST	terrenos/	{ "base":10 "altura":20 }	Retorna o conceito aprovado ou reprovado	{ "base": 10, "altura":20, "area":200 }

Modelo 'Terrenos.php':

```
<?php
class Terrenos{
    private $base;
    private $altura;
    public function setAltura($v1){
        $this->altura = $v1;
    }
    public function getAltura(){
        return $this->altura;
    }
    public function setBase($v1){
        $this->base = $v1;
    }
    public function getBase(){
        return $this->base;
    }
    public function calcularArea(){
        return $this->base * $this->altura;
    }
}
```

?>

Controle 'controle_Terrenos_Area.php':

```
<?php
require_once "modelo/Terrenos.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);
$terrenos = new Terrenos();
$terrenos ->setBase($rota[2]);
$terrenos ->setAltura($rota[3]);
```

```

$resposta = array();
$resposta['base'] = (float)$rota[2];
$resposta['altura'] = (float)$rota[3];
$resposta['area'] = $terrenos->calcularArea();
echo json_encode($resposta);
?>

```

Controle 'controle_Terrenos_POST.php':

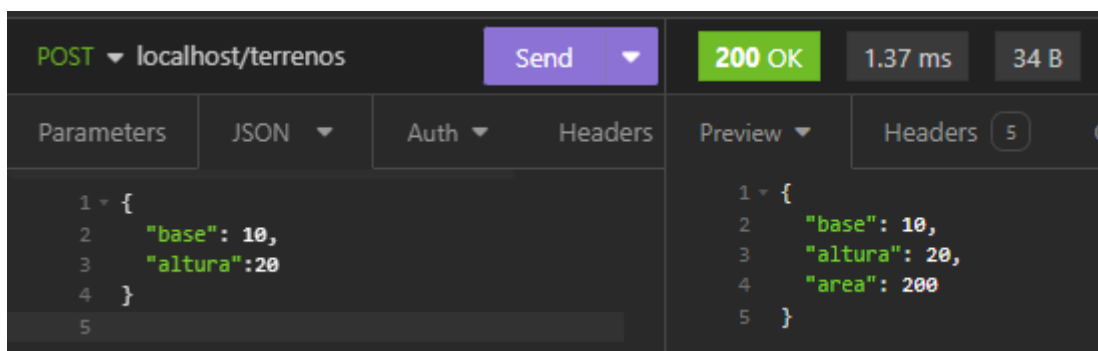
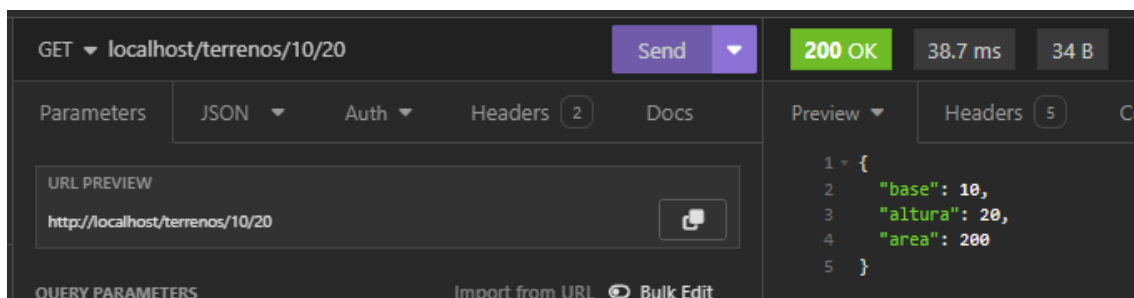
```

<?php
require_once"modelo/Terrenos.php";
$jsonRecebido = file_get_contents('php://input');
$dadosJson = json_decode($jsonRecebido);

$base = $dadosJson->base;
$altura = $dadosJson->altura;
$terrenos = new Terrenos();
$terrenos->setBase($base);
$terrenos->setAltura($altura);

$resposta = array();
$resposta['base'] = $base;
$resposta['altura'] = $altura;
$resposta['area'] = $terrenos->calcularArea();
header("HTTP/1.1 200 OK");
header('Content-Type: application/json; charset=utf-8');
echo json_encode($resposta);
?>

```



26. Faça um programa que calcule e mostre valores de graus Celsius e graus Fahrenheit, cujos graus variem de n1 a n2 F de 1 em 1. A conversão de graus Fahrenheit (F) para graus Celsius (C) é dada por: $C = 5/9 * (F - 32)$.

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/graus/n1/n2	/ graus /32/35	Retorna graus °C.	{ "32°F": "0°C", "33°F": "0,55°C", "34°F": "1,11°C", "35°F": "1,66°C" }
GET	/graus/n1	/ graus /32	Retorna o equivalente em °C	{ "32°F": "0°C" }

Modelo 'Gaus.php':

```
<?php
class Gaus{
    private $graus;
    private $inicio;
    private $fim;
    public function setGaus($v1){
        $this->graus = $v1;
    }
    public function getGaus(){
        return $this->graus;
    }
    public function converterParaCelsius($v){
        $celsius = 5/9*($v-32);
        return $celsius;
        //C * 9/5 + 32 = F
    }

    public function setInicioTabela($inicio){
        $this->inicio = $inicio;
    }
    public function getInicioTabela(){
        return $this->inicio;
    }
    public function setFimTabela($fim){
        $this->fim = $fim;
    }
    public function getFimTabela(){
        return $this->fim;
    }
}
```

```
}
```

```
?>
```

Controle 'controle_Graus_Tabela.php':

```
<?php
```

```
require_once"modelo/Graus.php";  
$rota = explode("/",$_SERVER['REQUEST_URI']);
```

```
$graus = new Graus();  
$graus ->setInicioTabela($rota[2]);  
$graus ->setFimTabela($rota[3]);
```

```
$resposta = array();
```

```
$inicio = $rota[2];  
$fim = $rota[3];  
for($i=$inicio;$i<=$fim;$i++){  
    $graus ->setGraus($i);  
    $nomePosicao = (string)$i.'°F';  
    $resposta[$nomePosicao] = $graus->converterParaCelsius($i).'°C';  
}  
echo json_encode($resposta);
```

```
?>
```

Controle 'controle_Graus.php':

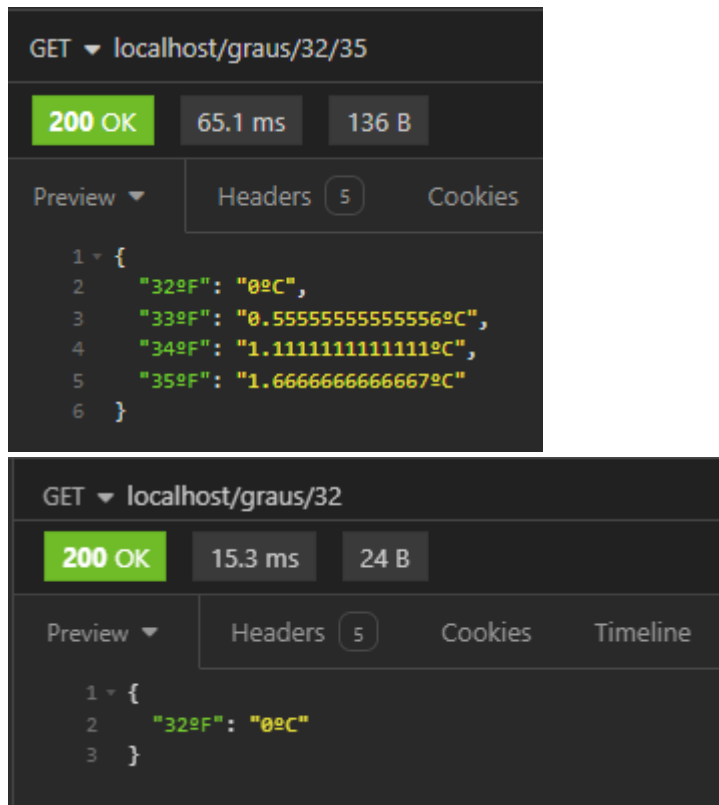
```
<?php
```

```
require_once"modelo/Graus.php";  
$rota = explode("/",$_SERVER['REQUEST_URI']);
```

```
$graus = new Graus();  
$graus ->setGraus($rota[2]);
```

```
$resposta = array();  
$nomePosicao = (string)$rota[2].'°F';  
$tempFar = floatval($rota[2]);  
$resposta[$nomePosicao] = $graus->converterParaCelsius($tempFar).'°C';  
echo json_encode($resposta);
```

```
?>
```

27. Faça um programa que calcule e mostre a área de um trapézio. Sabe-se que: $A = (base\ maior + base\ menor) * altura / 2$;

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/trapezios/b1/b2/h	/ trapezios /3/5/2	Calcula a área do trapézio	<pre>{ "baseMaior": "5", "BaseMenor": "3", "altura": "2", "area": "8" }</pre>

Modelo 'Trapezios.php':

```
<?php
```

```
class Trapezio{
```

```
    private $baseMaior;
```

```
    private $baseMenor;
```

```
    private $altura;
```

```
    public function setBaseMenor($v){
```

```
        $this->baseMenor = $v;
```

```
    }
```

```
    public function getBaseMenor(){
```

```
        return $this->baseMenor;
```

```
    }
```

```

    public function setBaseMaior($v){
        $this->baseMaior = $v;
    }
    public function getBaseMaior(){
        return $this->baseMaior;
    }

    public function setAltura($v){
        $this->altura = $v;
    }
    public function getAltura(){
        return $this->altura;
    }

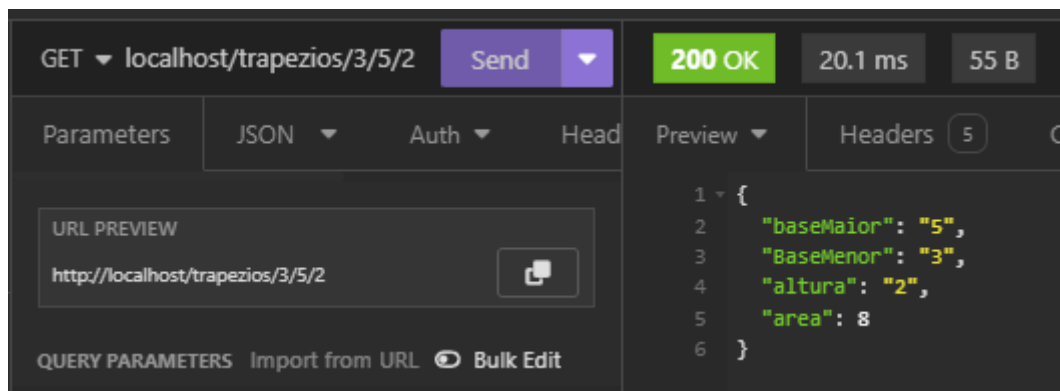
    public function calcularArea($v1,$v2,$v3){
        return (($v1+$v2)*$v3)/2;
    }
}
?>
Controle 'controle_Trapezios_Area.php':
<?php
    require_once "modelo/Trapezio.php";
    $rota = explode("/",$_SERVER['REQUEST_URI']);

    $trapezio = new Trapezio();
    $trapezio ->setBaseMenor($rota[2]);
    $trapezio ->setBaseMaior($rota[3]);
    $trapezio ->setAltura($rota[4]);

    $resposta = array();
    $resposta['baseMaior'] = $trapezio->getBaseMaior();
    $resposta['BaseMenor'] = $trapezio->getBaseMenor();
    $resposta['altura'] = $trapezio->getAltura();
    $resposta['area'] = $trapezio->calcularArea($rota[2],$rota[3],$rota[4]);
    echo json_encode($resposta);

?>

```



28. Calcular o salário líquido do funcionário sabendo que este é constituído pelo salário bruto mais o valor das horas extras subtraindo 8% de INSS do total. Serão lidos nesse problema o salário bruto, o valor das horas extras e o número de horas extras. Apresentar ao final o salário líquido.

Método	Rota	Envio de dados	Funcionalidade	Retorno
POST	/funcionarios/salarios	{ "salarioBruto":1300, "valorHoraExtra":15, "totalHorasExtras":8 }	Apresentar ao final o salário líquido	{ "salarioBruto":1300, "valorHoraExtra":15, "totalHorasExtras":8, "salarioLiquido":1306,40 }

Modelo 'Funcionarios.php':

```
<?php
```

```
class Funcionarios{
    private $salarioBruto;
    private $valorHoraExtra;
    private $totalHorasExtras;

    public function setSalarioBruto($v){
        $this->salarioBruto = $v;
    }
    public function getSalarioBruto(){
        return $this->salarioBruto;
    }
    public function setValorHoraExtra($v){
        $this->valorHoraExtra = $v;
    }
    public function getValorHoraExtra(){
        return $this->valorHoraExtra;
    }
    public function setTotalHorasExtras($v){
        $this->totalHorasExtras = $v;
    }
    public function getTotalHorasExtras(){
        return $this->totalHorasExtras;
    }
}
```

```

        public function calcularSalario(){
            return ($this->salarioBruto + ($this->totalHorasExtras*$this->valorHoraExtra)) * 0.92;
        }
    }
?>

```

Controle 'controle_Funcionarios_Salario.php':

```

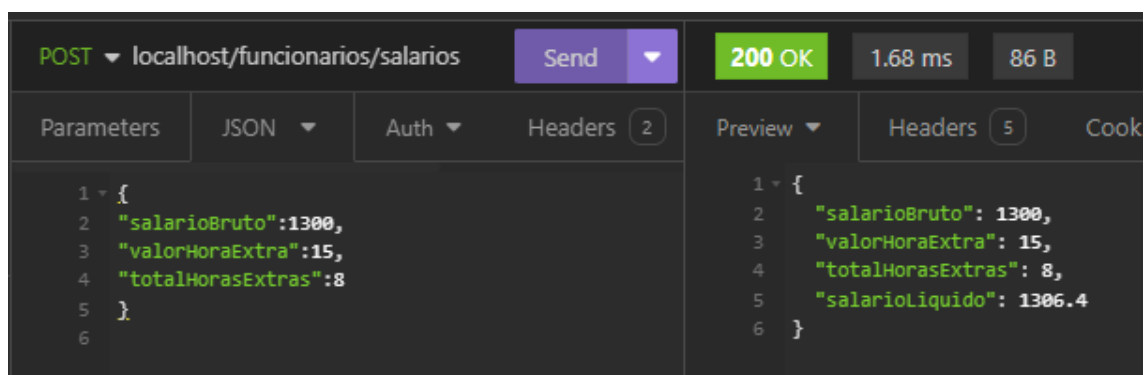
<?php
    require_once "modelo/Funcionarios.php";
    $jsonRecebido = file_get_contents('php://input');
    $obj = json_decode($jsonRecebido);
    $salarioBruto = $obj->salarioBruto;
    $valorHoraExtra = $obj->valorHoraExtra;
    $totalHorasExtras = $obj->totalHorasExtras;

    $funcionario = new Funcionarios();
    $funcionario->setSalarioBruto($salarioBruto);
    $funcionario->setValorHoraExtra($valorHoraExtra);
    $funcionario->setTotalHorasExtras($totalHorasExtras);

    $resposta = array();
    $resposta['salarioBruto'] = $funcionario->getSalarioBruto();
    $resposta['valorHoraExtra'] = $funcionario->getValorHoraExtra();
    $resposta['totalHorasExtras'] = $funcionario->getTotalHorasExtras();
    $resposta['salarioLiquido'] = $funcionario->calcularSalario();

    header("HTTP/1.1 200 OK");
    header('Content-Type: application/json; charset=utf-8');
    echo json_encode($resposta);
?>

```



29. Efetuar a leitura do número de quilowatts consumido e calcular o valor a ser pago de energia elétrica, sabendo-se que o valor a pagar por quilowatt é de 0,12. Apresentar o valor total a ser pago pelo usuário acrescido de 18% de ICMS.

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/quilowatts/n	/quilowatts/350	Calcular o valor a ser pago	{ "quilowatts":350, "valorQuilowatts":0,12, "imposto": 18%

				<pre> "totalPagar": } </pre>
--	--	--	--	--

Modelo 'Quilowatts.php':

```

<?php
class Quilowatts{
    private $Kwatts;

    public function setKwatts($v){
        $this->Kwatts = $v;
    }
    public function getKwatts(){
        return $this->Kwatts;
    }
    public function calcularValorPagar(){
        return ($this->Kwatts * 0.12) * 1.18;
        //resultado em Reais
    }
}

?>

```

Controle 'controle_Quilowatts_ValorPagar.php':

```

<?php
require_once"modelo/Quilowatts.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

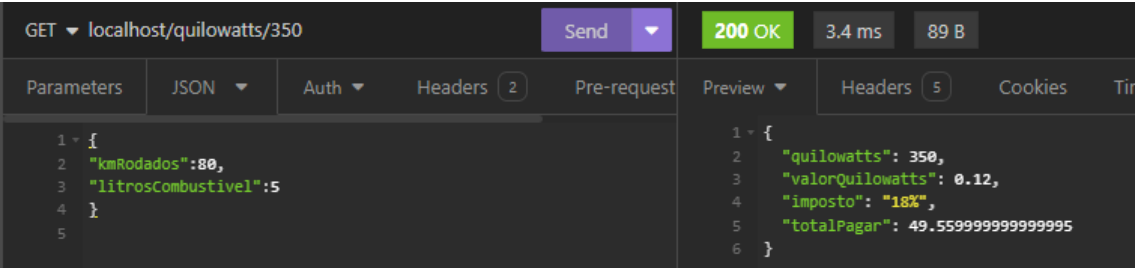
$quilowatts = new Quilowatts();
$quilowatts->setKwatts($rota[2]);

$resposta = array();
$resposta['quilowatts'] = (float)$quilowatts->getKwatts();
$resposta['valorQuilowatts'] = 0.12;
$resposta['imposto'] = "18%";
$resposta['totalPagar'] = (float)$quilowatts->calcularValorPagar();

echo json_encode($resposta);

?>

```



30. Calcular a média de combustível gasto pelo usuário, sendo informado a quantidade de quilômetros rodados e a quantidade de combustível.

Método	Rota	Envio de dados	Funcionalidade	Retorno
POST	/combustiveis/medias	{ "kmRodados":80, "litrosCombustivel":5 }	Calcular consumo	{ "kmRodados":80, "litrosCombustivel":5, "consumo": "16km/L"} }

Modelo 'Combustiveis.php':

```
<?php
class Combustiveis{
    private $kmRodados;
    private $litrosCombustivel;

    public function setKmRodados($v1){
        $this->kmRodados = $v1;
    }
    public function getKmRodados(){
        return $this->kmRodados;
    }
    public function setLitrosCombustivel($v1){
        $this->litrosCombustivel = $v1;
    }
    public function getLitrosCombustivel(){
        return $this->litrosCombustivel;
    }

    public function calcularMediaCombustivel(){
        return $this->kmRodados / $this->litrosCombustivel;
    }

}
?>
```

Controle 'controle_Funcionarios_Salario.php':

```
<?php
require_once "modelo/Funcionarios.php";
$jsonRecebido = file_get_contents('php://input');
$obj = json_decode($jsonRecebido);
$salarioBruto = $obj->salarioBruto;
$valorHoraExtra = $obj->valorHoraExtra;
$totalHorasExtras = $obj->totalHorasExtras;

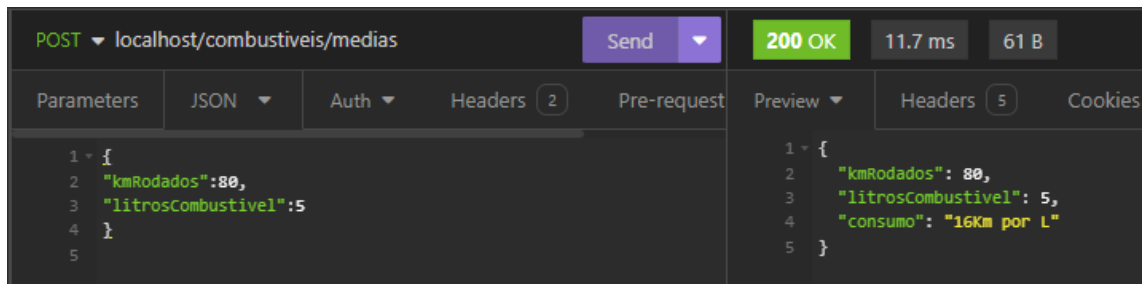
$funcionario = new Funcionarios();
$funcionario->setSalarioBruto($salarioBruto);
$funcionario->setValorHoraExtra($valorHoraExtra);
$funcionario->setTotalHorasExtras($totalHorasExtras);
```

```

$resposta = array();
$resposta['salarioBruto'] = $funcionario->getSalarioBruto();
$resposta['valorHoraExtra'] = $funcionario->getValorHoraExtra();
$resposta['totalHorasExtras'] = $funcionario->getTotalHorasExtras();
$resposta['salarioLiquido'] = $funcionario->calcularSalario();

header("HTTP/1.1 200 OK");
header('Content-Type: application/json; charset=utf-8');
echo json_encode($resposta);
?>

```



31. Elabore um algoritmo que dada a idade de um nadador classifique-o em uma das seguintes categorias:
- Infantil A = 5 a 7 anos
 - Infantil B = 8 a 11 anos
 - Juvenil A = 12 a 13 anos
 - Juvenil B = 14 a 17 anos
 - Adultos = Maiores de 18 anos

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/classificacoes/nadadores/idade	/classificacoes/nadadores/9	Retorna classificação	{ "idade":9, "categoria":Infantil B }

Modelo 'Classificacoes.php':

```

<?php
class Classificacoes{
    private $idade;

    public function setidade($v){
        $this->idade = $v;
    }
    public function getidade(){
        return $this->idade;
    }
    public function classificarCategoria(){
        if($this->idade <= 0){
            return 'Digite valor adequado!';
        }elseif($this->idade > 0 && $this->idade < 4){

```

```

        return 'Fora dos Limites';
    }elseif($this->idade >= 5 && $this->idade <=7){
        return 'Infantil A (5 a 7 anos)';
    }elseif($this->idade >= 8 && $this->idade <=11){
        return 'Infantil B (7 a 11 anos)';
    }elseif($this->idade >= 12 && $this->idade <=13){
        return 'Juvenil A (12 a 13 anos)';
    }elseif($this->idade >= 14 && $this->idade <=17){
        return 'Juvenil B (14 a 17 anos)';
    }elseif($this->idade > 18){
        return 'Adultos (Maiores de 18)';
    }
}

}

?>

```

Controle 'controle_Classificacoes_Categoria.php':

```

<?php
require_once"modelo/Classificacoes.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

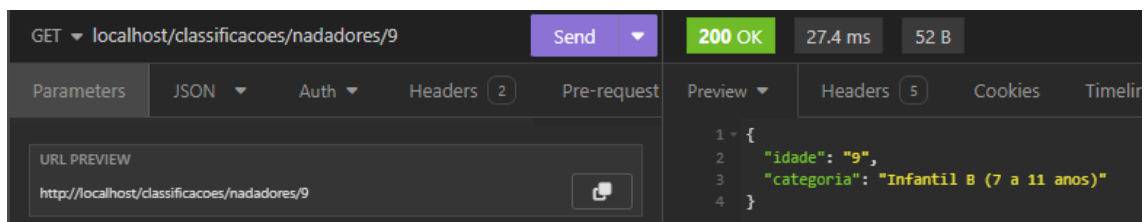
$classificacoes = new Classificacoes();
$classificacoes->setidade($rota[3]);

$resposta = array();
$resposta['idade'] = $classificacoes->getidade();
$resposta['categoria'] = $classificacoes->classificarCategoria();

echo json_encode($resposta);

?>

```



32. Elabore um algoritmo para calcular a equação do segundo grau e imprima as raízes da equação na tela, os valores a, b e c são inseridos pelo usuário: $ax^2 + bx + c = 0$, $\Delta = b^2$

$$- 4ac, x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Método	Rota	Envio de dados	Funcionalidade	Retorno
GET	/bhaskaras/a/b/c	/bhaskaras/3/15/12	Retorna as raízes da equação	{ "a":3, "b":15, "c":12, "delta":81, "x1":1, "x2":4 }

Modelo 'Bhaskaras.php':

```
<?php
class Bhaskaras{
    private $a;
    private $b;
    private $c;

    public function setA($v){
        $this->a = $v;
    }
    public function setB($v){
        $this->b = $v;
    }
    public function setC($v){
        $this->c = $v;
    }
    public function getA(){
        return $this->a;
    }
    public function getB(){
        return $this->b;
    }
    public function getC(){
        return $this->c;
    }

    public function calcularDelta(){
        $delta = ($this->b**2) - (4*$this->a*$this->c);
```

```

        return $delta;
    }

    public function calcularRaizesUnica(){
        $delta = $this->calcularDelta();
        if($delta < 0){
            return "Nao possui raizes";
        }elseif($delta == 0){
            return -$this->b/(2*$this->a);
        }
    }

    public function calcularesX1(){
        $delta = $this->calcularDelta();
        if($delta > 0){
            return (-$this->b + sqrt($delta))/(2*$this->a);
        }
    }

    public function calcularesX2(){
        $delta = $this->calcularDelta();
        if($delta > 0){
            return (-$this->b - sqrt($delta))/(2*$this->a);
        }
    }
}

```

?>

Controle 'controle_Bhaskaras_Raizes.php':

<?php

```

require_once "modelo/Bhaskaras.php";
$rota = explode("/", $_SERVER['REQUEST_URI']);

```

```

$bhaskara = new Bhaskaras();
$bhaskara->setA($rota[2]);
$bhaskara->setB($rota[3]);
$bhaskara->setC($rota[4]);
$bhaskara->calcularDelta();
$bhaskara->calcularRaizesUnica();

```

```

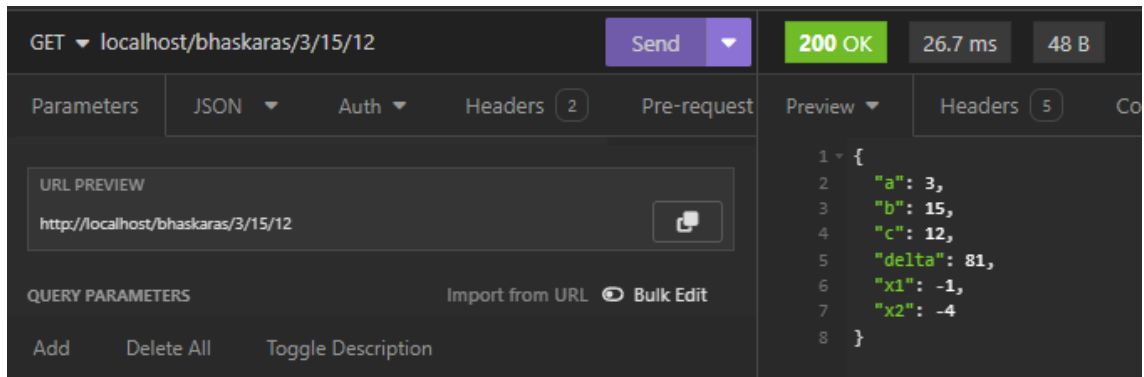
$resposta = array();
$resposta['a'] = (float)$bhaskara->getA();
$resposta['b'] = (float)$bhaskara->getB();
$resposta['c'] = (float)$bhaskara->getC();
$resposta['delta'] = $bhaskara->calcularDelta();
$resposta['x1'] = $bhaskara->calcularesX1();

```

```
$resposta['x2'] = $bhaskara->calcularesX2();
```

```
echo json_encode($resposta);
```

```
?>
```



***obs.: para os exercícios 33 e 34 monte você a tabela de rotas, entrada de dados e retornos**

33. Construa programa para determinar se o indivíduo esta com um peso favorável. Essa situação é determinada através do IMC (Índice de Massa Corpórea), que é definida como sendo a relação entre o peso (PESO) e o quadrado da Altura (ALTURA) do indivíduo. Ou seja

$$IMC = \frac{PESO}{ALTURA^2}$$

Escreva na tela de acordo com as seguintes situações:

| Condição | Situação |
|-------------------|----------------|
| IMC abaixo de 20 | Abaixo do peso |
| IMC de 20 até 25 | Peso Normal |
| IMC de 25 até 30 | Sobre Peso |
| IMC de 30 até 40 | Obeso |
| IMC de 40 e acima | Obeso Mórbido |

| Método | Rota | Envio de dados | Funcionalidade | Retorno |
|--------|------------------|----------------|----------------|---|
| GET | /imc/peso/altura | /imc/60/1.70 | Calcular IMC. | {
"peso": "60",
"altura": "1.70",
"imc":
20.761245674740486,
"observacao":
"Peso Normal"
} |

Modelo 'IMC.php':

```
<?php
class IMC{
    private $peso;
    private $altura;

    public function setPeso($v1){
        $this->peso = $v1;
    }
    public function getPeso(){
        return $this->peso;
    }
    public function setAltura($v1){
        $this->altura = $v1;
    }
    public function getAltura(){
        return $this->altura;
    }
    public function calcularIMC(){
        return $this->peso / ($this->altura ** 2);
    }

    public function observacao(){
        if($this->calcularIMC() < 20){
            return "Abaixo do Peso";
        }elseif($this->calcularIMC() >= 20 && $this->calcularIMC() <= 25){
            return "Peso Normal";
        }elseif($this->calcularIMC() >= 25 && $this->calcularIMC() <= 30){
            return "Sobre Peso";
        }elseif($this->calcularIMC() >= 30 && $this->calcularIMC() < 40){
            return "Obeso<br>";
        }elseif($this->calcularIMC() >= 40){
            return "Obeso Morbido";
        }
    }
}
?>
```

Controle 'controle_Calcular_IMC.php':

```
<?php
require_once"modelo/IMC.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

$imc = new IMC();
$imc->setPeso($rota[2]);
```

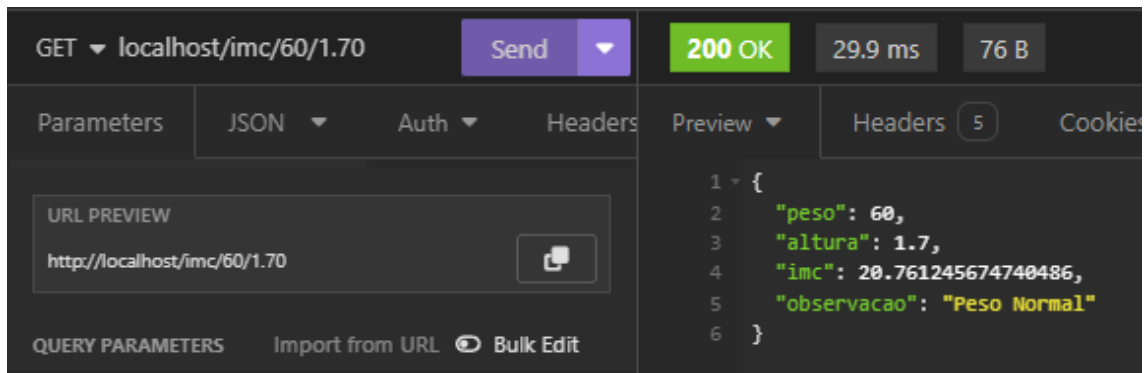
```

$imc->setAltura($rota[3]);

$resposta = array();
$resposta['peso'] = (float)$imc->getPeso();
$resposta['altura'] = (float)$imc->getAltura();
$resposta['imc'] = $imc->calcularIMC();
$resposta['observacao'] = $imc->observacao();

echo json_encode($resposta);
?>

```



34. Construa programa que leia três lados de um triângulo, verifique e escreva que tipo de triângulo eles formam (considere **triângulo equilátero** com três lados iguais, **triângulo isósceles** com dois lados iguais e **triângulo escaleno** com todos os lados diferentes).

| Método | Rota | Envio de dados | Funcionalidade | Retorno |
|--------|-------------------------------|-------------------|------------------------|---|
| GET | /triangulos/lado1/lado2/lado3 | /triangulos/3/3/3 | Classificar trinagulo. | {
"lado1": "3",
"lado2": "3",
"lado3": "3",
"classificacao":
"Trinagulo Equilatero"
} |

Modelo 'Triangulos.php':

```

<?php
class Triangulos{
    private $lado1;
    private $lado2;
    private $lado3;

    public function setLado1($v){
        $this->lado1 = $v;
    }
    public function setLado2($v){
        $this->lado2 = $v;
    }
    public function setLado3($v){

```

```

        $this->lado3 = $v;
    }

    public function getLado1(){
        return $this->lado1;
    }
    public function getLado2(){
        return $this->lado2;
    }
    public function getLado3(){
        return $this->lado3;
    }

    public function ClassificarTriangulo(){
        if(($this->lado1 == $this->lado2 && $this->lado3 != $this->lado1) || ($this->lado1 ==
$this->lado3 && $this->lado1 != $this->lado2) || ($this->lado2 == $this->lado3 && $this-
>lado2 != $this->lado1)){
            return "Triangulo Isoceles";
        }elseif($this->lado1 != $this->lado2 && $this->lado1 != $this->lado3 && $this->lado2 !=
$this->lado3){
            return "Triangulo Escaleno";
        }elseif($this->lado1 == $this->lado2 && $this->lado1 == $this->lado3){
            return "Trinagulo Equilatero";
        }
    }
}
?>

```

Controle 'controle_Triangulos_Classificar.php':

```

<?php
require_once "modelo/Triangulos.php";
$rota = explode("/",$_SERVER['REQUEST_URI']);

$classificacao = new Triangulos();
$classificacao->setLado1($rota[2]);
$classificacao->setLado2($rota[3]);
$classificacao->setLado3($rota[4]);

$resposta = array();
$resposta['lado1'] = (float)$classificacao->getLado1();
$resposta['lado2'] = (float)$classificacao->getLado2();
$resposta['lado3'] = (float)$classificacao->getLado3();
$resposta['classificacao'] = $classificacao->ClassificarTriangulo();

echo json_encode($resposta);

```

?>

GET localhost/triangulos/3/3/3

Send

200 OK

12.3 ms

70 B

Parameters

JSON

Auth

Headers

URL PREVIEW

http://localhost/triangulos/3/3/3

QUERY PARAMETERS

Import from URL

Bulk Edit

Preview

Headers

Cookies

Timeline

```
1 {
2   "lado1": 3,
3   "lado2": 3,
4   "lado3": 3,
5   "classificacao": "Trinagulo Equilatero"
6 }
```