

Exercicios

Exercicio 1: Xerador de secuencias "FizzBuzz" en HTML

Conceptos Integrados:

- Bucles (for)
- Operadores Aritméticos (especialmente o módulo %)
- Condicionais (if-elseif-else)
- Xeración de HTML dinámico

Descrición do Problema: "FizzBuzz" é un exercicio clásico de programación. A tarefa consiste en xerar unha lista ordenada de números do 1 ao 100. Pero con certas regras:

- Se un número é divisible por 3, en lugar do número, debe mostrarse a palabra "Fizz".
- Se un número é divisible por 5, debe mostrarse "Buzz".
- Se un número é divisible por 3 e por 5, debe mostrarse "FizzBuzz".
- En calquera outro caso, móstrase o número.

O teu script PHP debe xerar unha lista ordenada () en HTML con estes resultados.

Requisitos:

0. Usa un bucle for que conte do 1 ao 100.
1. Dentro do bucle, utiliza unha estrutura if-elseif-else para avaliar as condicións na orde correcta.
2. Usa o operador módulo (%) para comprobar se un número é divisible por outro (\$numero % 3 === 0).
3. Xera dinamicamente os elementos da lista () con echo. Os elementos especiais ("Fizz", "Buzz", "FizzBuzz") deben ter un estilo diferente (por exemplo, en negriña) usando a etiqueta .

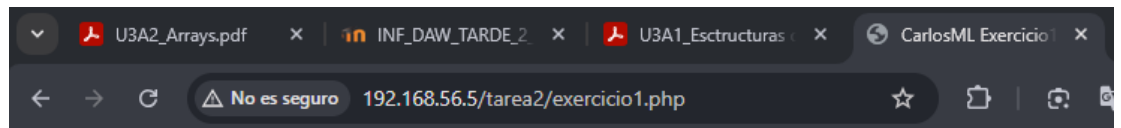
Exemplo do resultado esperado (fragmento):

```
<ol>
  <li>1</li>
  <li>2</li>
  <li><strong>Fizz</strong></li>
  <li>4</li>
  <li><strong>Buzz</strong></li>
  <li><strong>Fizz</strong></li>
  ...
  <li>14</li>
  <li><strong>FizzBuzz</strong></li>
  ...
</ol>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML Ejercicio1</title>
</head>
<body>
  <ol>
  <?php
    for($i=1;$i<=100;$i++){
      if($i%3===0 && $i%5===0){
        echo "<li>Fizz</li>";
      }elseif($i%3===0){
        echo "<li>Buzz</li>";
      }elseif($i%5===0){
        echo "<li>FizzBuzz</li>";
      }else{
        echo "<li>$i</li>";
      }
    }
  ?>
  </ol>
</body>
</html>

```



```

1. 1
2. 2
3. Buzz
4. 4
5. FizzBuzz
6. Buzz
7. 7
8. 8
9. Buzz
10. FizzBuzz
11. 11
12. Buzz
13. 13
14. 14
15. Fizz
16. 16
17. 17
18. Buzz
19. 19
20. FizzBuzz
21. Buzz
22. 22
23. 23
24. Buzz
25. FizzBuzz
26. 26
27. Buzz
28. 28
29. 29
30. Fizz
31. 31
32. 32
33. Buzz
34. 34
35. FizzBuzz
36. Buzz
37. 37
38. 38
39. Buzz
40. FizzBuzz

```

Exercicio 2: Cesta da compra dinámica

Conceptos á usar:

- Arrays (introdución conceptual)
- Bucles (foreach)
- Condicionais (if-else)
- Operadores aritméticos e de asignación
- Xeración dunha táboa HTML

Descrición do problema: Simula o resumo dunha cesta da compra. Tes unha lista de produtos nun array, onde cada produto ten un nome, un prezo e un indicador de se está en oferta. O teu script debe percorrer esta lista, mostrar cada produto nunha táboa, calcular o subtotal, aplicar un desconto aos produtos en oferta e, finalmente, mostrar o prezo total.

Requisitos:

0. Define un array de produtos. Cada elemento do array será outro array asociativo coas claves `nome`, `prezo` e `en_oferta` (un valor booleano `true/false`).
1. Define unha variable para o desconto (ex: `$desconto_oferta = 0.10; // 10%`).
2. Crea unha táboa HTML (`<table>`) coas cabeceiras: "Produto", "Prezo Orixinal", "Prezo Final".
3. Usa un bucle `foreach` para percorrer o array de produtos.
4. Dentro do bucle, usa un `if` para comprobar se `$produto['en_oferta']` é `true`.
 - Se o é, calcula o prezo final con desconto. Mostra o prezo final en cor verde e engade unha pequena nota "(En oferta!)".
 - Se non, o prezo final é o mesmo que o orixinal.
5. Acumula o prezo final de cada produto nunha variable `$total_cesta`.
6. Despois do bucle, fóra da táboa, mostra o total da cesta nunha cabeceira `<h2>`.

Exemplo do resultado esperado:

```
<!-- Táboa xerada por PHP -->
<table border="1">
  <thead>
    <tr><th>Produto</th><th>Prezo Orixinal</th><th>Prezo Final</th></tr>
  </thead>
  <tbody>
    <tr><td>Televisor 4K</td><td>800€</td><td>800€</td></tr>
    <tr><td>Barra de Son</td><td>150€</td><td style="color: green;">135€ (En oferta!)</td></tr>
    <tr><td>Consola</td><td>500€</td><td>500€</td></tr>
  </tbody>
</table>

<h2>Total da Cesta: 1435€</h2>
```

```
<!DOCTYPE html>
```

```

<html lang="gl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML_E2</title>
</head>
<body>
  <?php
    $produtos = [
      ["nome" => "Mazás", "prezo" => 2.5, "en_oferta" => true],
      ["nome" => "Leite", "prezo" => 1.2, "en_oferta" => false],
      ["nome" => "Pan", "prezo" => 1.0, "en_oferta" => true]
    ];
    $desconto_oferta=0.10;
    $total_cesta=0;
  ?>
  <table border="1">
    <thead>
      <tr><th>Produto</th><th>Prezo Orixinal</th><th>Prezo Final</th></tr>
    </thead>
    <tbody>
      <?php
        foreach ($produtos as $producto) {
          if($producto['en_oferta']==true){
            $prezoFinal=$producto['prezo']-($producto['prezo']*$desconto_oferta);
            echo '<tr><td>' . $producto['nome'] . '</td><td>' . $producto['prezo'] . '
€</td><td style="color:green;">' . $prezoFinal . ' (En oferta!)' . '</td></tr>';
            $total_cesta+=$prezoFinal;
          }else{
            echo '<tr><td>' . $producto['nome'] . '</td><td>' . $producto['prezo'] . '
€</td><td>' . $producto['prezo'] . '</td></tr>';
            $total_cesta+=$producto['prezo'];
          }
        }
      ?>
    </tbody>

  </table>

  <?php
    echo "<h2>Total da Cesta: $total_cesta</h2>"
  ?>
</body>
</html>

```

Producto	Prezo Orixinal	Prezo Final
Mazás	2.5 €	2.25 (En oferta!)
Leite	1.2 €	1.2
Pan	1 €	0.9 (En oferta!)

Total da Cesta: 4.35

Exercicio 3: Panel de control de usuario segundo o seu Rol

Conceptos integrados:

- Variables
- Estrutura condicional avanzada (match)
- Xeración de HTML dinámico

Descrición do problema: Unha aplicación web a miúdo mostra contido diferente dependendo do rol do usuario que iniciou sesión. O teu traballo é crear un script que, dada unha variable que contén o rol dun usuario, xere un panel de control (unha serie de "widgets" ou caixas) personalizado para ese rol.

Requisitos:

0. Define unha variable `$rol_usuario` e asígnalle un dos seguintes valores: 'admin', 'editor', 'rexistrado', ou 'convidado'.
1. Usa unha estrutura `match` sobre a variable `$rol_usuario`.
2. A expresión `match` debe devolver un array cos títulos dos widgets que cada rol pode ver.
 - 'admin': Pode ver "Xestión de Usuarios", "Estatísticas do Sitio", "Editor de Contido" e "Ver Sitio".
 - 'editor': Pode ver "Editor de Contido" e "Ver Sitio".
 - 'rexistrado': Pode ver "Ver Perfil" e "Ver Sitio".
 - default (para invitados e outros): Pode ver só "Ver Sitio".
3. O resultado do `match` (o array de widgets) gárdao nunha variable `$widgets_a_mostrar`.
4. Finalmente, usa un bucle `foreach` para percorrer `$widgets_a_mostrar` e xerar unha serie de `<div>` con estilo para cada widget.

Exemplo do resultado esperado (se `$rol_usuario = 'admin'`):

```
<h1>Panel de Administración</h1>

<!-- Widgets xerados por PHP -->
<div style="border: 1px solid black; padding: 10px; margin: 5px;">
  <h2>Xestión de Usuarios</h2>
</div>
<div style="border: 1px solid black; padding: 10px; margin: 5px;">
  <h2>Estatísticas do Sitio</h2>
</div>
<div style="border: 1px solid black; padding: 10px; margin: 5px;">
  <h2>Editor de Contido</h2>
</div>
<div style="border: 1px solid black; padding: 10px; margin: 5px;">
  <h2>Ver Sitio</h2>
</div>
```

```

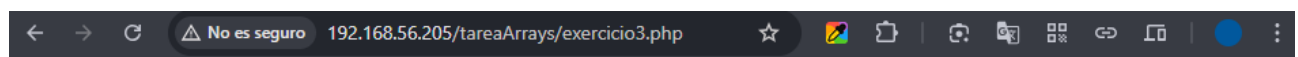
<!DOCTYPE html>
<html lang="gl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML_E3</title>
</head>
<body>
  <?php
    $rol_usuario='admin';

    $widgets_a_mostrar= match($rol_usuario){
      'admin'=> ["Xestión de Usuarios", "Estatísticas do Sitio", "Editor de Contido",
"Ver Sitio"],
      'editor'=> ["Editor de Contido", "Ver Sitio"],
      'rexistrado'=> [ "Ver Perfil", "Ver Sitio"],
      'default'=> [ "Ver Sitio"],
    };

    echo "<h1>Panel de Administración</h1>";

    foreach ($widgets_a_mostrar as $widget){
      echo ' <div style="border: 1px solid black; padding: 10px; margin: 5px;">';
      echo " <h2>$widget</h2>";
      echo "</div>";
    }
  ?>
</body>

```



Panel de Administración

Xestión de Usuarios

Estatísticas do Sitio

Editor de Contido

Ver Sitio

Exercicio 4: Selector de destinatarios para newsletter

Conceptos integrados:

- Arrays de arrays asociativos
- Bucles (foreach)
- Condicionais (if)
- Control de fluxo de bucle (continue)

Descrición do problema: Tes que programar a lóxica do servidor para seleccionar a que usuarios se lles enviará unha newsletter. Tes unha lista de usuarios nun array, pero só debes seleccionar aqueles que cumpran dúas condicións: que estean **activos** e que teñan a **subscripción á newsletter activada**. O teu script debe xerar unha lista dos nomes dos destinatarios finais.

Requisitos:

0. Define un array \$usuarios. Cada elemento será un array asociativo con nome, email, activo (booleano) e suscrito_newsletter (booleano).
1. Crea un array baleiro \$destinatarios = []; para gardar os nomes dos usuarios seleccionados.
2. Usa un bucle foreach para percorrer a lista de \$usuarios.
3. Dentro do bucle, usa un if cun operador lóxico || (OR) para comprobar se o usuario está inactivo (!\$usuario['activo']) ou non está suscrito (!\$usuario['suscrito_newsletter']).
 - Se algunha destas condicións é certa, usa continue para saltar inmediatamente ao seguinte usuario sen procesalo.
4. Se o usuario pasa o filtro, engade o seu nome ao array \$destinatarios.
5. Ao final, mostra unha mensaxe co número total de destinatarios e unha lista HTML con todos os nomes do array \$destinatarios.

Exemplo do resultado esperado:

```
<h3>Procesando lista de usuarios...</h3>
<h2>A newsletter enviarase a 2 destinatarios:</h2>
<ul>
  <li>Ana García</li>
  <li>Carlos Pérez</li>
</ul>
```



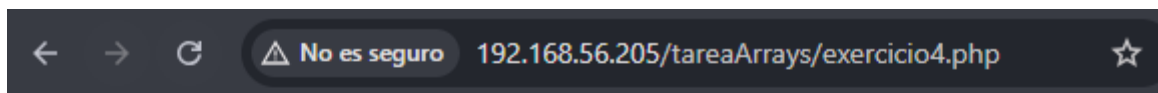
```

<!DOCTYPE html>
<html lang="gl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML_E4</title>
</head>
<body>
  <?php
    $usuarios=[
      ["nome"=> "Carlos", "email" => "carlos@gmail.com", "activo"=> true,
"subscrito_newsletter" => true],
      ["nome"=> "Ana", "email" => "ana@gmail.com", "activo"=> true,
"subscrito_newsletter" => true],
      ["nome"=> "Adrian", "email" => "adrian@gmail.com", "activo"=> false,
"subscrito_newsletter" => true],
      ["nome"=> "Hugo", "email" => "hugo@gmail.com", "activo"=> false,
"subscrito_newsletter" => false]
    ];
    $destinatarios=[];

    foreach($usuarios as $usuario){
      if(!$usuario['activo'] || !$usuario['subscrito_newsletter'] ){
        continue;
      }else{
        $destinatarios[]=$usuario["nome"];
      }
    }

    echo '<h3>Procesando lista de usuarios...</h3>';
    echo "<h2>A newsletter enviarase a " . count($destinatarios) . "
destinatarios:</h2>";
    echo "<ul>";
    foreach ($destinatarios as $destinatario){
      echo "<li>".$destinatario. "</li>";
    }
    echo "</ul>";
  ?>
</body>
</html>

```



Procesando lista de usuarios...

A newsletter enviarase a 2 destinatarios:

- Carlos
- Ana

Exercicio 5: Visualizador de datos simple (gráfico de barras)

Conceptos integrados:

- Arrays asociativos
- Bucles (foreach)
- Condicionais (if-else)
- Xeración de HTML e CSS dinámico

Descrición do problema: O servidor a miúdo necesita presentar datos de forma visual. A túa tarefa é xerar un gráfico de barras horizontal moi sinxelo usando HTML e CSS en liña. Recibirás os datos nun array asociativo onde as claves son as etiquetas (ex: meses) e os valores son as cantidades a representar.

Requisitos:

0. Define un array asociativo `$datos_vendas = ["Xaneiro" => 150, "Febreiro" => 220, "Marzo" => 180, "Abril" => 90];`.
1. Define unha variable `$valor_obxectivo = 175;`.
2. Usa un bucle `foreach` para percorrer os datos.
3. Dentro do bucle, por cada mes e valor:
 - Xera un `<div>` para a etiqueta do mes.
 - Xera outro `<div>` que representará a barra. O ancho desta barra debe ser o valor en píxeles (ex: `style="width: 220px; ..."`).
 - Usa un `if` para comprobar se o valor supera o `$valor_obxectivo`.
 - Se o supera, a cor de fondo da barra debe ser verde (`background-color: lightgreen;`).
 - Se non, debe ser laranxa (`background-color: lightsalmon;`).
4. Engade un pouco de estilo básico para que o resultado sexa visualmente claro.

Exemplo do resultado esperado:

```
<h1>Vendas do Primeiro Trimestre</h1>

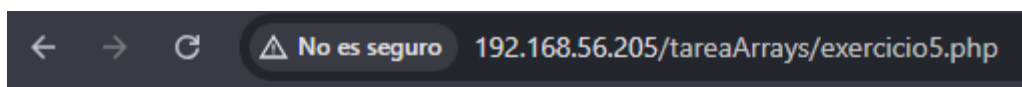
<!-- Gráfico xerado por PHP -->
<div style="margin-bottom: 10px;">
  <strong>Xaneiro:</strong>
  <div style="width: 150px; background-color: lightsalmon; color: white;
padding: 5px;">150</div>
</div>
<div style="margin-bottom: 10px;">
  <strong>Febreiro:</strong>
  <div style="width: 220px; background-color: lightgreen; color: white; padding:
5px;">220</div>
</div>
...
```

```

<!DOCTYPE html>
<html lang="gl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML_E5</title>
</head>
<body>
  <?php
    $datos_vendas = ["Xaneiro" => 150, "Febreiro" => 220, "Marzo" => 180, "Abril" =>
90];
    $valor_obxectivo = 175;

    foreach($datos_vendas as $mes => $valor){
      echo '<div style="margin-bottom: 10px;">';
      echo "<strong>$mes:</strong>";
      if ($valor>$valor_obxectivo){
        echo '<div style="width:'.$valor.'px; background-color: lightgreen; color:
white; padding: 5px;">'.$valor.'</div>';
      }else{
        echo '<div style="width:'.$valor.'px; background-color: lightsalmon; color:
white; padding: 5px;">'.$valor.'</div>';
      }
    }
    echo "</div>"
  ?>
</body>
</html>

```



Xaneiro:

150

Febreiro:

220

Marzo:

180

Abril:

90

Exercicio 6: Buscador dos primeiros números primos

Conceptos integrados:

- Bucles anidados (while e for)
- Control de fluxo (break)
- Operadores (% , ++)
- Lóxica algorítmica

Descrición do problema: Este é un desafío máis algorítmico. Un número primo é un número natural maior que 1 que non ten divisores positivos á parte de 1 e el mesmo. O teu script debe atopar os primeiros N números primos e mostralos nunha lista.

Requisitos:

0. Define unha variable `$cantidade_primos_a_atopar = 10;`
1. Crea un array baleiro `$primos_atopados = [];` e un contador `$numero_actual = 2;`.
2. Usa un bucle `while` que se execute mentres o número de elementos en `$primos_atopados` sexa menor que `$cantidade_primos_a_atopar`.
3. Dentro do `while`, debes comprobar se `$numero_actual` é primo. Para iso:
 - Asume que é primo cunha variable booleana: `$e_primo = true;`
 - Usa un bucle `for` que vaia dende 2 ata a raíz cadrada de `$numero_actual`. (Podes simplificalo indo ata `$numero_actual / 2`).
 - Dentro do `for`, comproba se `$numero_actual % $i === 0`. Se é así, o número non é primo.
 - Se non é primo, cambia `$e_primo` a `false` e usa `break` para saír do bucle `for` (non ten sentido seguir comprobando).
4. Despois do bucle `for`, se `$e_primo` segue sendo `true`, engade `$numero_actual` ao array `$primos_atopados`.
5. Incrementa `$numero_actual` para comprobar o seguinte número.
6. Ao final, fóra do bucle `while`, mostra unha lista `` cos números primos atopados.

Exemplo do resultado esperado (para 10 primos):

```
<h2>Os primeiros 10 números primos son:</h2>
<ul>
  <li>2</li>
  <li>3</li>
  <li>5</li>
  <li>7</li>
  <li>11</li>
  <li>13</li>
  <li>17</li>
  <li>19</li>
```

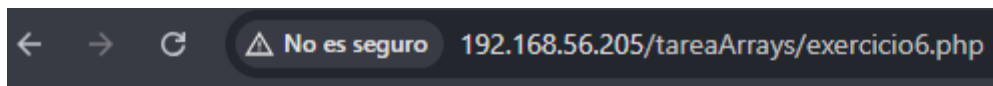
```
<li>23</li>
<li>29</li>
</ul>
```

```
<!DOCTYPE html>
<html lang="gl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CarlosML_E6</title>
</head>
<body>
  <?php
    $cantidad_primos_a_atopar = 10;

    $primos_atopados = [];
    $numero_actual = 2;

    while (count($primos_atopados) < $cantidad_primos_a_atopar) {
      $e_primo = true;
      for ($i = 2; $i <= $numero_actual / 2; $i++) {
        if ($numero_actual % $i === 0) {
          $e_primo = false;
          break;
        }
      }

      if ($e_primo) {
        $primos_atopados[] = $numero_actual;
      }
      $numero_actual++;
    }
    echo "<h2>Os primeiros $cantidad_primos_a_atopar números primos son:</h2>";
    echo "<ul>";
    foreach ($primos_atopados as $primo) {
      echo "<li>$primo</li>";
    }
    echo "</ul>";
  ?>
</body>
</html>
```



Os primeiros 10 números primos son:

- 2
- 3
- 5
- 7
- 11
- 13
- 17
- 19
- 23
- 29