

Exercicio 1: Procesamento e agrupación de datos dun ficheiro CSV

Este exercicio simula unha tarefa moi habitual: importar datos dende un ficheiro (por exemplo, un exportado dunha tenda online ou un sistema de xestión) e realizar cálculos e agrupacións con eles.

Obxectivo:

Ler un ficheiro CSV con datos de vendas de libros dunha librería galega, procesar eses datos usando arrays para extraer estatísticas e amosar un resumo agrupado.

Tarefas a realizar:

1. Preparar o ficheiro de datos:

- Crea un ficheiro de texto chamado `vendas.csv`.
- Cada liña do ficheiro representará unha venda e conterá os seguintes datos separados por comas: `id_libro,titulo,autor,categoria,prezo,provincia_venta`.
- Engade polo menos 10 liñas con datos variados. Utiliza autores, títulos e categorías que teñan sentido en Galicia.

Exemplo de contido para `vendas.csv`:

```
101,Follas Novas,Rosalía de Castro,Poesía,15.50,A Coruña
102,Memorias dun neno labrego,Xosé Neira Vilas,Narrativa,12.00,Pontevedra
103,O lapis do carpinteiro,Manuel Rivas,Narrativa,18.75,Lugo
104,Cantigas de Santa María,Afonso X o Sabio,Poesía,22.00,Ourense
105,Sempre en Galiza,Alfonso Daniel Rodríguez Castelao,Ensaio,25.00,A Coruña
106,A praia dos afogados,Domingo Villar,Novela Negra,19.95,Pontevedra
107,Cousas,Alfonso Daniel Rodríguez Castelao,Narrativa,14.50,Lugo
```

2. Ler e procesar o ficheiro en PHP:

- No teu script PHP, utiliza a función `fopen()` para abrir o ficheiro `vendas.csv` en modo lectura.
- Crea un bucle (`while`) e utiliza `fgetcsv()` para ler o ficheiro liña por liña. Esta función devolve cada liña como un array indexado.
- Dentro do bucle, converte cada array indexado nun array asociativo para que sexa máis doado traballar con el (ex: `['id_libro' => 101, 'titulo' => 'Follas Novas', ...]`). Almacena todos estes arrays asociativos nun único array principal chamado `$vendas`.

3. Análise e cálculo de datos:

- Unha vez teñas todos os datos no array `$vendas`, realiza as seguintes operacións:

- **Cálculo de ingresos totais:** Percorre o array e suma os prezos de todas as vendas para obter os ingresos totais. Podes facelo de forma máis eficiente usando `array_column()` para extraer todos os prezos e despois `array_sum()`.
- **Agrupación por provincia:** Crea un novo array chamado `$vendas_por_provincia`. Itera sobre o array `$vendas` e agrupa as vendas por provincia. O resultado debe ser un array asociativo onde as claves sexan as provincias ("A Coruña", "Lugo", "Ourense", "Pontevedra".) e os valores sexan arrays coas vendas correspondentes a cada unha.
- **Filtrado por categoría:** Utiliza a función `array_filter()` para crear un novo array que conteña unicamente os libros da categoría "Narrativa".

4. Mostrar os resultados:

- Presenta na páxina web un informe claro cos resultados obtidos:
 - Os ingresos totais.
 - Unha listaxe das vendas agrupadas por provincia, mostrando o título do libro e o prezo en cada unha.
 - Unha listaxe separada cos libros que pertencen á categoría "Narrativa".

Conceptos a utilizar:

- Lectura e procesamento de ficheiros (`fopen`, `fgetcsv`).
- Conversión de arrays indexados a asociativos.
- Funcións de manipulación de arrays de alto nivel (`array_column`, `array_sum`, `array_filter`).
- Lóxica de agrupación de datos para xerar estruturas de arrays máis complexas.
- Extracción e presentación de datos procesados.

```

<?php
$ficheiro = fopen("vendas.csv", "r");
$vendas = [];

while (($fila = fgetcsv($ficheiro, 1000, ",")) !== false) {
    $venda = [
        'id_libro' => $fila[0],
        'titulo' => $fila[1],
        'autor' => $fila[2],
        'categoria' => $fila[3],
        'prezo' => (float)$fila[4],
        'provincia_venta' => $fila[5]
    ];
    $vendas[] = $venda;
}
fclose($ficheiro);

$ingresos_totais = array_sum(array_column($vendas, 'prezo'));

$vendas_por_provincia = [];
foreach ($vendas as $venda) {
    $provincia = $venda['provincia_venta'];
    $vendas_por_provincia[$provincia][] = $venda;
}

$narrativa = array_filter($vendas, function ($v) {
    return $v['categoria'] === 'Narrativa';
});
?>

<!DOCTYPE html>
<html lang="gl">
<head>
    <meta charset="UTF-8">
    <title>CarlosML_Ejercicio1</title>
</head>
<body>

<h2>Ingresos totais</h2>
<p><?php echo number_format($ingresos_totais, 2, ',', '.'); ?> €</p>

<h2>Ventas agrupadas por provincia</h2>
<?php foreach ($vendas_por_provincia as $provincia => $vendas): ?>

```

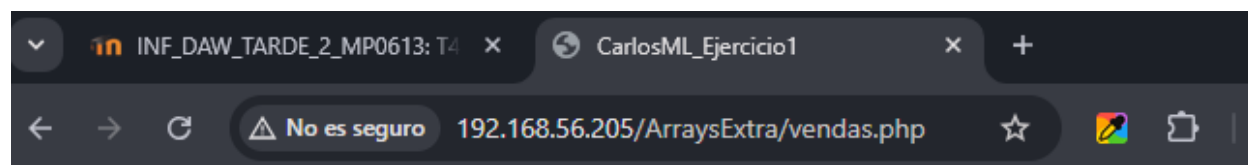
```

<h3><?php echo $provincia; ?></h3>
<ul>
    <?php foreach ($vendas as $v): ?>
        <li>
            <?php echo $v['titulo']; ?> -
            <?php echo $v['autor']; ?>
            (<?php echo $v['categoria']; ?>) -
            <?php echo number_format($v['prezo'], 2, ',', '.'); ?> €
        </li>
    <?php endforeach; ?>
</ul>
<?php endforeach; ?>

<h2>Libros da categoría "Narrativa"</h2>
<ul>
    <?php foreach ($narrativa as $v): ?>
        <li>
            <?php echo $v['titulo']; ?> -
            <?php echo $v['autor']; ?>
            (<?php echo $v['provincia_venta']; ?>) -
            <?php echo number_format($v['prezo'], 2, ',', '.'); ?> €
        </li>
    <?php endforeach; ?>
</ul>

</body>
</html>

```



Ingresos totais

181,75 €

Vendas agrupadas por provincia

A Coruña

- Follas Novas — Rosalia de Castro (Poesia) — 15,50 €
- Sempre en Galiza — Alfonso Daniel Rodriguez Castelao (Ensaio) — 25,00 €

Pontevedra

- Memorias dun neno labrego — Xose Neira Vilas (Narrativa) — 12,00 €
- A praia dos afogados — Domingo Villar (Novela Negra) — 19,95 €

Lugo

- O lapis do carpinteiro — Manuel Rivas (Narrativa) — 18,75 €
- Cousas — Alfonso Daniel Rodriguez Castelao (Narrativa) — 14,50 €
- Merlin e familia — Alvaro Cunqueiro (Fantasía) — 20,00 €

Ourense

- Cantigas de Santa Maria — Afonso X o Sabio (Poesia) — 22,00 €
- Os vellos non deben de namorarse — Alfonso Daniel Rodriguez Castelao (Teatro) — 17,25 €
- A esmorga — Eduardo Blanco Amor (Narrativa) — 16,80 €

Libros da categoría "Narrativa"

- Memorias dun neno labrego — Xose Neira Vilas (Pontevedra) — 12,00 €
- O lapis do carpinteiro — Manuel Rivas (Lugo) — 18,75 €
- Cousas — Alfonso Daniel Rodriguez Castelao (Lugo) — 14,50 €
- A esmorga — Eduardo Blanco Amor (Ourense) — 16,80 €

Exercicio 2: Xerador de equipos aleatorios para un torneo

Este exercicio céntrase na manipulación da estrutura dun array: barallar, dividir e reorganizar elementos para crear unha estrutura de datos nova, como os emparellamentos dun torneo.

Obxectivo:

Crear un script que, a partir dunha lista de participantes nun torneo, xere aleatoriamente os emparellamentos para a primeira rolda.

Tarefas a realizar:

1. Definir a lista de participantes:

- Crea un array indexado chamado `$participantes` que conteña os nomes dos inscritos no "Torneo de Chave de Fisterra".
- Para simplificar, comeza cun número de participantes que sexa unha potencia de 2 (por exemplo, 8).

Exemplo de lista de participantes:

```
$participantes = [  
    'Antía', 'Xoel', 'Sabela', 'Breixo',  
    'Iria', 'Anxo', 'Noa', 'Brais'  
];
```

2. Barallar e crear os emparellamentos:

- Usa a función `shuffle()` para desordenar aleatoriamente o array de `$participantes`. Isto asegura que os emparellamentos sexan totalmente ao chou.
- Unha vez barallado, divide o array de participantes en grupos de dous. A forma máis eficiente de facelo é coa función `array_chunk()`. O resultado será un array multidimensional onde cada elemento interno é un array con dous participantes (un emparellamento).
- Almacena este novo array de emparellamentos nunha variable chamada `$rolda1`.

3. Mostrar os resultados:

- Percorre o array `$rolda1` e mostra na páxina os enfrontamentos da primeira rolda de forma clara.

Exemplo de saída (variará en cada execución):

Emparellamentos da primeira rolda do Torneo de Pandeiretas de Fisterra:

- Enfrontamento 1: Sabela vs. Anxo
- Enfrontamento 2: Breixo vs. Antía
- Enfrontamento 3: Noa vs. Brais
- Enfrontamento 4: Iria vs. Xoel

4. Ampliación:

- Modifica o script para que funcione correctamente cando o número de participantes é impar (por exemplo, 9).
- Neste caso, un participante debe pasar directamente á seguinte rolda ("bye").
- A lóxica debería:
 1. Comprobar se o número de participantes é impar.
 2. Se o é, seleccionar aleatoriamente un participante do array barallado e gardalo como o que pasa de rolda.
 3. Eliminar ese participante do array principal.
 4. Crear os emparellamentos co resto dos participantes (que agora será un número par).
 5. Mostrar tanto os emparellamentos como o nome do participante que avanza directamente.

Conceptos a utilizar:

- Ordenación aleatoria de arrays (`shuffle`).
- Reestruturación de arrays planos en arrays multidimensionais (`array_chunk`).
- Lóxica algorítmica para manexar casos especiais e excepcións (número impar de participantes).
- Selección e eliminación de elementos aleatorios dun array (`array_rand`, `unset`).
- Presentación de datos estruturados de forma lexible.

```

<!DOCTYPE html>
<html lang="gl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CarlosML_Ejercicio2</title>
</head>
<body>
    <?php

        $participantes = [
            'Antía', 'Xoel', 'Sabela', 'Breixo',
            'Iria', 'Anxo', 'Noa', 'Brais'
        ];
        shuffle($participantes);
        $rolda1=array_chunk($participantes,2);

        echo "<h1>Ronda 1:</h1>";

        for($i=0;$i<count($rolda1);$i++){
            echo "<p>Enfrontamento ". ($i +1). ". ":". $rolda1[$i][0]. " vs. ". $rolda1[$i]
[1]. "</p>";
        }

        echo "<h1>Ampliación:</h1>";

        $participantesImpares = [
            'Antía', 'Xoel', 'Sabela', 'Breixo',
            'Iria', 'Anxo', 'Noa'
        ];

        shuffle($participantesImpares);

        if (count($participantesImpares) % 2 != 0) {
            $indiceAleatorio = array_rand($participantesImpares, 1);
            $pasaRonda = $participantesImpares[$indiceAleatorio];
            echo "<h4>$pasaRonda pasa directamente á seguinte rolda!</h4>";

            unset($participantesImpares[$indiceAleatorio]);
        }

        $rolda2 = array_chunk($participantesImpares, 2);
    </?php>

```

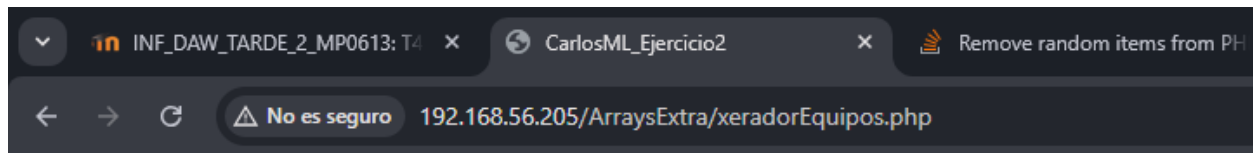


```

        echo "<h1>Ronda 2:</h1>";
        for ($i = 0; $i < count($rolda2); $i++) {
            echo "<p>Enfrentamento " . ($i + 1) . ": " . $rolda2[$i][0] . " vs. " .
$rolda2[$i][1] . "</p>";
        }
    ?>

</body>
</html>

```



Ronda 1:

Enfrentamento 1: Anxo vs. Iria

Enfrentamento 2: Antía vs. Breixo

Enfrentamento 3: Sabela vs. Xoel

Enfrentamento 4: Noa vs. Brais

Ampliación:

Antía pasa directamente á seguinte rolda!

Ronda 2:

Enfrentamento 1: Noa vs. Sabela

Enfrentamento 2: Iria vs. Breixo

Enfrentamento 3: Xoel vs. Anxo