

# Les dates

- [Les dates](#)
- [Exemple Calendari. Procedural](#)
- [Exemple Calendari. OOP](#)
- [Exemple Calendari. OOP + classe calendari](#)

# Les dates

El maneig de dates i hores en PHP ha evolucionat molt, passant de les funcions bàsiques a una orientació a objectes molt més robusta i fiable. Entendre aquest sistema és crucial per a qualsevol aplicació, especialment per a la correcta gestió de fusos horaris.

## El funcionament intern de les dates en PHP

PHP gestiona el temps i les dates basant-se en el concepte de **Timestamp Unix** (Marca de temps Unix), tot i que la majoria de les funcions modernes treballen internament amb objectes.

### El timestamp Unix

Un timestamp Unix és un valor sencer que representa el **nombre de segons** transcorreguts des de l'**Època Unix** (la mitjanit UTC de l'1 de gener de 1970).

- **Avantatge:** És universal, independent del fus horari i ideal per a emmagatzemar dates a bases de dades o fer càlculs de temps (sumar o restar segons).
- **Funció Clau:** La funció `time()` retorna el timestamp Unix actual.

### El fus Horari (Timezone)

El fus horari és la configuració més crítica. PHP necessita saber en quin fus horari ha d'interpretar i mostrar les dates, ja que un mateix timestamp pot representar una hora diferent segons la ubicació.

## Configuració (PHP.INI)

Dos paràmetres al fitxer `php.ini` (o configurats dinàmicament al codi) afecten directament el maneig de dates:

## date.timezone

Aquest és el paràmetre **obligatori** i més important. Estableix el fus horari per defecte que PHP utilitzarà per a totes les operacions de data i hora.

- **Configuració Recomanada:** S'ha d'establir a una cadena de fus horari vàlida (e.g., `Europe/Madrid`, `America/New_York`).
- **Problema Comú:** Si no es defineix, PHP genera un avís (`Warning`) i utilitza un fus horari per defecte insegur, cosa que pot causar errors de càlcul d'una hora.
- **Configuració Dinàmica:** Es pot sobrescriure al codi amb `date_default_timezone_set('Europe/Madrid')`.

## date.default\_latitude / date.default\_longitude

Aquests paràmetres, juntament amb `date.sunrise_color` i `date.sunset_color`, s'utilitzen per a funcions de càlcul d'efemèrides com ara `date_sunrise()` i `date_sunset()`. No afecten el format de data ni l'hora.

# Funcions disponibles (Procedural)

PHP ofereix un conjunt de funcions antigues però encara molt utilitzades, que converteixen entre timestamps i cadenes de text (strings):

## Lectura de temps

- `time()`: Retorna el timestamp Unix actual.
- `microtime(true)`: Retorna el timestamp Unix amb precisió de microsegons (com a `float`).

## Formatació de Dates

- `date(string $format, int $timestamp = time()): string`:
  - Converteix un timestamp Unix (o el temps actual per defecte) en una cadena de data/hora formatada.
  - El paràmetre `$format` utilitza caràcters especials (e.g., `Y` per a l'any complet, `m` per al mes numèric, `H` per a les hores).
  - Exemple: `date('d/m/Y H:i:s')` retorna `"26/09/2025 11:24:54"`.
- `gmdate()`: Igual que `date()`, però sempre formatat en el fus horari **GMT/UTC**.

# Conversió de cadena a Timestamp

- `strtotime(string $time, int $now = time()): int`:
  - La funció `strtotime()` de PHP és una de les funcions més potents i flexibles, tot i ser de l'enfocament procedural antic. La seva funció principal és analitzar (o interpretar) una cadena de text en anglès (human-readable) que descriu una data o hora i convertir-la al **Timestamp Unix** corresponent (el nombre de segons des de l'1 de gener de 1970 00:00:00 UTC). És popular per la seva capacitat d'interpretar formats absoluts (com "2025-12-31") i formats relatius (com "next Monday" o "+2 weeks").
  - Exemples: `strtotime('tomorrow')`, `strtotime('+1 week')`, `strtotime('2025-01-01')`.

<https://www.php.net/manual/es/function strtotime.php>

- `mktime()`: Retorna el timestamp Unix d'una data específica, permetent passar any, mes, dia, hora, minut, i segon com a arguments separats.

## Exemples. Formats Date

```
<?php
$diaAvui=date("d");
echo "formato: date(\"d\")->    ".$diaAvui."<br/>";
$diaAvui=date("D");
echo "formato: date(\"D\")->    ".$diaAvui."<br/>";
$diaAvui=date("j");
echo "formato: date(\"j\")->    ".$diaAvui."<br/>";
$diaAvui=date("l");
echo "formato: date(\"l\")->    ".$diaAvui."<br/>";
$diaAvui=date("N");
echo "formato: date(\"N\")->    ".$diaAvui."<br/>";
$diaAvui=date("S");
echo "formato: date(\"S\")->    ".$diaAvui."<br/>";
$diaAvui=date("d").date("S");
echo "formato: date(\"d\").date(\"S\")->    ".$diaAvui."<br/>";
$dia=date("d-m-Y");
echo "formato: ".$dia."<br/>";
$dia=date("d-m-Y",strtotime("15-12-2025"));
echo "formato: ".$dia."<br/>";
$dia=date("N",strtotime("8-11-2025"));
echo "formato: ".$dia."<br/>";
```

```
<?php
echo "Avui es: " . date ("j M Y G:i:s") . "<br/>";
```

```

echo date ("YmdHis") . "<br/>"; //202509281534
echo "Dia setmana: " . date ("w") . "<br/>";
echo "Dia setmana: " . date ("w", strtotime("09/11/2025")) . "<br/>";

echo "Mes actual: " . date("n") . "<br/>";
$mes=date("n")+1;
echo "Mes actual: " . $mes . "<br/>";

$darrerDiaMes = date("t")."-".date("n")."-".date("Y");
echo "Dia setmana darrer dia mes: " . date("w",strtotime($darrerDiaMes));

```

## Exemples. Date avui

```

<?php
$diaAvui=date("N");
switch ($diaAvui) {
    case 1:
        $diaSemana="Lunes";
        break;
    case 2:
        $diaSemana="Martes";
        break;
    case 3:
        $diaSemana="Miercoles";
        break;
    case 4:
        $diaSemana="Jueves";
        break;
    case 5:
        $diaSemana="Viernes";
        break;
    case 6:
        $diaSemana="Sabado";
        break;
    case 7:
        $diaSemana="Domingo";
        break;
}
$mesAvui=date("n");
switch ($mesAvui) {
    case 1:
        $Mes="Enero";
        break;
    case 2:

```

```

        $Mes="Febrero";
        break;
    case 3:
        $Mes="Marzo";
        break;
    case 4:
        $Mes="Abril";
        break;
    case 5:
        $Mes="Mayo";
        break;
    case 6:
        $Mes="Junio";
        break;
    case 7:
        $Mes="Julio";
        break;
    case 8:
        $Mes="Agosto";
        break;
    case 9:
        $Mes="Septiembre";
        break;
    case 10:
        $Mes="Octubre";
        break;
    case 11:
        $Mes="Noviembre";
        break;
    case 12:
        $Mes="Diciembre";
        break;
}

echo $diaSemana.", ".date("j")." de ".$Mes." de ".date("Y");

```

## Exemples. Operacions Date

```

<?php
$fecha_actual = date("d-m-Y");
//sumo 1 día
echo 'Ahora:           '. date('d-m-Y') ."<br/>";
echo 'Sumo 31 dia:      '.date("d-m-Y",strtotime($fecha_actual."+ 31
days"))."<br/><br/>";
//resto 1 día
echo 'Ahora:           '. date('d-m-Y') ."<br/>";
echo 'Resto 31 dia:      '.date("d-m-Y",strtotime($fecha_actual."- 31
days"))."<br/><br/>";

```

```

//sumo 1 mes
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Sumo 1 mes:          '.date("d-m-Y",strtotime($fecha_actual."+ 1 month"))."<br/>
<br/>";
//resto 1 mes
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Resto 1 mes:         '.date("d-m-Y",strtotime($fecha_actual."- 1
month"))."<br/><br/>";

//sumo 1 año
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Sumo 1 año:          '.date("d-m-Y",strtotime($fecha_actual."+ 1 year"))."<br/>
<br/>";
//resto 1 año
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Resto 1 año:         '.date("d-m-Y",strtotime($fecha_actual."- 1 year"))."<br/>
<br/>";

//sumo 1 semana
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Semana Siguiente:    '. date("d-m-Y",strtotime($fecha_actual."+ 1 week"))."<br/>
<br/>";
//resto 1 semana
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Semana Anterior:     '. date("d-m-Y",strtotime($fecha_actual."- 1 week"))."<br/>
<br/>";

$semanaSiguiente = time() + (7 * 24 * 60 * 60); // 7 días; 24 horas; 60 minutos;
60 segundos
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Semana Siguiente:     '. date('d-m-Y', $semanaSiguiente) . "<br/><br/>";

$semanaSiguiente = time() - (7 * 24 * 60 * 60); // 7 días; 24 horas; 60 minutos;
60 segundos
echo 'Ahora:                '. date('d-m-Y') . "<br/>";
echo 'Semana Anterior:      '. date('d-m-Y', $semanaSiguiente) . "<br/><br/>";

```

### **strtotime( STRING ).**

Retorna un object timestamp a partir d'una data expressada en la cadena de  
 ⚠ characters

- Si el separador es / llavors la data esta expressada MES/DIA/ANY
- Si el separador es - llavors la data està expressada DIA/MES/ANY

La funció `strtotime()` de PHP té un comportament diferent i potencialment confús quan s'utilitza la barra inclinada (/) com a separador entre els components numèrics de la data (mes, dia, any), en comparació amb el guió (-).

Aquesta diferència es basa en la manera com PHP, i en última instància, la biblioteca C subjacent, intenta inferir quin format de data s'està utilitzant.

## El Comportament de `strtotime()` segons el Separador

La clau de la interpretació de `strtotime()` rau en la distinció entre el format americà (**M/D/Y**) i el format europeu (**D/M/Y**).

### El Guió (-): Format ISO 8601 (Any-Mes-Dia)

Quan utilitzes el guió, `strtotime()` assumeix el format més estricte i universal: **Any-Mes-Dia (Y-M-D)**.

| Entrada (Guió) | Format Interpretar | Explicació                              |
|----------------|--------------------|-----------------------------------------|
| "2025-03-01"   | Y-M-D              | S'interpreta com l'1 de març de 2025.   |
| "03-01-2025"   | Y-M-D              | S'interpreta com el 3 de gener de 2025. |

**Conclusió amb -:** El guió indueix el format internacional (ISO 8601), que prioritza l'any al principi (o l'any al final si es presenta només amb 3 components, assumint M-D-Y).

### La Barra Inclorada (/): Format Americà (Mes/Dia/Any) US

Quan utilitzes la barra inclinada (/), `strtotime()` prioritza el format de data curt utilitzat comunament als Estats Units: **Mes/Dia/Any (M/D/Y)**.

| Entrada (Barra) | Format Interpretar | Explicació                                                                             |
|-----------------|--------------------|----------------------------------------------------------------------------------------|
| "03/01/2025"    | <b>M/D/Y</b>       | S'interpreta com l'1 de març de 2025.                                                  |
| "01/03/2025"    | <b>M/D/Y</b>       | S'interpreta com el 3 de gener de 2025.                                                |
| "31/01/2025"    | Error              | Si el primer número supera 12 (més d'un mes), l'anàlisi falla, ja que s'espera un mes. |

**Conclusió amb /:** La barra inclinada obliga a utilitzar la lògica **Mes/Dia/Any (M/D/Y)**, independentment de les preferències regionals del servidor.

## L'Ambivalència i el Risc d'Errors

El problema sorgeix quan un desenvolupador en una regió on s'utilitza el format Dia/Mes/Any (D/M/Y) assumeix que la barra inclinada mantindrà aquest ordre.

| Desenvolupador | Espera                                | PHP Llegeix (M/D/Y)   | Resultat (Error)       |
|----------------|---------------------------------------|-----------------------|------------------------|
| Europeu        | 12 de maig de 2025<br>(12/05/2025)    | 5 de desembre de 2025 | <b>Data Incorrecta</b> |
| Europeu        | 5 de desembre de 2025<br>(05/12/2025) | 12 de maig de 2025    | <b>Data Incorrecta</b> |

Això es coneix com a "**ambigüitat de format de data**" i pot provocar errors subtils i difícils de detectar en la lògica de l'aplicació.

## La solució recomanada: `DateTime` amb Format Explícit

Per evitar qualsevol ambigüitat, la pràctica recomanada és **evitar totalment l'ús de `strtotime()`** per analitzar dates d'entrada de l'usuari amb format D/M/Y o M/D/Y. En el seu lloc, s'ha d'utilitzar el mètode `DateTime::createFromFormat()`, ja que obliga a PHP a utilitzar el format de data que tu especifiques amb exactitud.

PHP

```
<?php
// Entrada de l'usuari (format català/europeu esperat: DD/MM/YYYY)
$data_entrada = "01/03/2025";

// 1. Ús ambigü (i erroni en context europeu):
// $timestamp_error = strtotime($data_entrada);
// S'interpreta com Març 1 (M/D) en lloc de l'1 de Març (D/M).

// 2. Ús Correcte (Orientat a Objectes):
// Indica explícitament a PHP que la cadena és Dia/Mes/Any
$data_objecte = DateTime::createFromFormat('d/m/Y', $data_entrada);

if ($data_objecte) {
    // La data és l'1 de març de 2025
    echo "Data analitzada correctament: " . $data_objecte->format('Y-m-d');
} else {
    echo "Error: El format de data no és vàlid.";
}
```

Amb `createFromFormat()`, elimines qualsevol conjectura de PHP sobre si estàs utilitzant un guió, una barra inclinada o un punt, ja que tu defineixes la màscara d'anàlisi exacta.

## Objectes per dates (Enfocament OO)

L'enfocament Orientat a Objectes (OOP) és l'estàndard modern en PHP, ja que resol problemes complexos de càlculs de dates i gestió de fusos horaris amb molta més fiabilitat.

### `DateTime` (o `DateTimeImmutable`)

Representa una data i hora concretes, sempre associada a un fus horari:

| Mètode Clau                                       | Descripció                                                                       | Exemple                                                                       |
|---------------------------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <b>Constructor</b>                                | Crea un nou objecte data.                                                        | <code>new DateTime()</code> (ara) o <code>new DateTime('2025-10-20')</code> . |
| <code>format(string \$format)</code>              | Formata l'objecte data en una cadena (similar a la funció <code>date()</code> ). | <code>\$data-&gt;format('Y-m-d')</code>                                       |
| <code>setTimezone(DateTimeZone \$timezone)</code> | Canvia el fus horari de l'objecte sense alterar el punt en el temps.             | <code>\$data-&gt;setTimezone(new DateTimeZone('America/New_York'))</code>     |
| <code>createFromFormat()</code>                   | Analitza una data d'una cadena amb un format conegut de forma segura.            | <code>DateTime::createFromFormat('d/m/Y', '26/09/2025')</code>                |

## DateTimeZone

Un objecte que encapsula un fus horari concret.

- **Ús:** Es fa servir com a argument en el constructor de `DateTime` o al mètode `setTimezone()`.
- Exemple: `new DateTimeZone('Asia/Tokyo')`.

## DateInterval

Representa un període de temps (e.g., 2 anys, 3 mesos i 5 dies).

- **Ús:** S'utilitza en operacions de data/hora.
- Exemple: `new DateInterval('P2Y3M5D')` (P de Període, 2 Anys, 3 Mesos, 5 Dies).

## DatePeriod

Representa una col·lecció de dates i hores dins d'un període, iterant segons un `DateInterval`.

- **Ús:** Generar una llista de totes les hores laborables entre dues dates.

## DateTime per càlculs (Immutabilitat)

La major fortalesa de l'OOP és el maneig de càlculs amb els mètodes `add()` i `sub()` (i `diff()` per calcular la diferència entre dues dates).

Per evitar modificar l'objecte original (un risc en programació), és molt recomanable utilitzar `DateTimeImmutable`, que sempre retorna una nova instància amb el canvi aplicat:

PHP

```
<?php
$ara = new DateTimeImmutable('now'); // Data original inalterable
```

```
// Sumem un període de temps (un mes i 5 dies)
$interval = new DateInterval('P1M5D');
$futur = $ara->add($interval); // Crea un nou objecte

echo "Ara: " . $ara->format('Y-m-d') . "\n";
echo "Futur: " . $futur->format('Y-m-d') . "\n";
```

# Exemple Calendari. Procedural

```
<html>
  <head>
    <title>Calendari <?php echo date("M") . "-" . date("Y"); ?></title>
  </head>
  <body>
    <h3> Calendari <?php echo date("M") . "-" . date("Y"); ?></h3>
    <table border="1" cellspacing="0" cellpadding="0">

      <tr>
        <td>Dl</td><td>Dm</td><td>Dc</td><td>Dj</td><td>Dv</td><td>Ds</td><td>Dg</td>
      </tr>
      <?php
        $dia=1;
        $nDiesMes= date("t");
        $iWDIni= date("w",strtotime( "1-".date("n")."-".date("Y") ));          //Dia
setmana primer dia
        $iWDFi= date("w",strtotime($nDiesMes."-".date("n")."-".date("Y"))); //Dia
setmana darrer dia mes

        echo "<tr>";
        for ($i=1 ; $i<$iWDIni ; $i++) {
          echo "<td>&nbsp;--</td>";
        }

        //pintar dies del mes
        while ($dia <= $nDiesMes)
        {
          //si es dilluns llavors echo "<tr>"

          echo "<td>$dia</td>";

          //si es diumenge llavors echo "</tr>"
          $dia++;
        }

      ?>
    </table>
  </body>
</html>
```



# Exemple Calendari. OOP

Calendari, utilitzant la potència dels objectes `DateTimeImmutable` i `DateInterval`. Aquesta aproximació manté el codi dins del bloc PHP principal, però utilitza l'OOP per a la manipulació de dates, la qual cosa el fa més segur i menys propens a errors de fus horari.

A continuació, tens el codi utilitzant només objectes de data.

PHP

```
<?php
// index.php

// -----
// 1. Configuració i Inicialització de Dades
// -----

// Configuració del fus horari (CRÍTIC per a la fiabilitat de dates)
date_default_timezone_set('Europe/Madrid');

// Data d'avui i objectes base (Immutable per seguretat)
$dataAvui = new DateTimeImmutable('today');

// Primer dia del mes que volem mostrar (Setembre 2025)
$dataMesActual = new DateTimeImmutable('first day of this month');
// Últim dia del mes
$darrerDiaMes = new DateTimeImmutable('last day of this month');

// -----
// 2. Càlcul de la Data d'Inici (Primer Dilluns Visible)
// -----

// Dia de la setmana del 1r dia del mes (N: 1=Dl, 7=Dg)
$diaSetmanaPrimerDia = (int)$dataMesActual->format('N');

// Dies a restar per arribar al Dilluns anterior (o el mateix Dilluns)
$diesARestar = $diaSetmanaPrimerDia - 1;
$dataIteracio = $dataMesActual->modify("-$diesARestar days");

// -----
// 3. Càlcul de la Data de Finalització (Últim Diumenge Visible)
// -----

// Dia de la setmana de l'últim dia del mes (N: 1=Dl, 7=Dg)
$diaSetmanaDarrerDia = (int)$darrerDiaMes->format('N');
```

```

// Dies a sumar per arribar al diumenge (dia 7)
// Si acaba en Diumenge (7), sumem 0 dies. Si acaba en Dissabte (6), sumem 1 dia.
$diesASumar = 7 - $diaSetmanaDarrerDia;

// Data de finalització de la graella (l'últim diumenge de la setmana del mes)
$dataFiGraella = $darrerDiaMes->modify("+$diesASumar days");

// Interval per avançar dia a dia dins del bucle
$intervalDia = new DateInterval('P1D');
?>

<!DOCTYPE html>
<html lang="ca">
<head>
    <meta charset="UTF-8">
    <title>Calendari Ajustat: <?php echo $dataMesActual->format("F - Y"); ?></title>
    <style>
        .calendari-taula {
            border-collapse: collapse;
            width: 400px;
            text-align: center;
        }
        .calendari-taula th, .calendari-taula td {
            padding: 8px;
            border: 1px solid #ccc;
        }
        .altre-mes {
            color: #aaa;
            background-color: #f7f7f7;
        }
        .avui {
            background-color: #ffe0b2;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h3> Calendari Ajustat: <?php echo $dataMesActual->format("F - Y"); ?></h3>
    <table class="calendari-taula" cellspacing="0" cellpadding="0">
        <tr>
            <th>Dl</th><th>Dm</th><th>Dc</th><th>Dj</th><th>Dv</th><th>Ds</th><th>Dg</
th>
            </tr>

            <?php
            // -----
            // 4. Bucle d'Iteració
            // -----
            while ($dataIteracio < $dataFiGraella) {

```

```

// Comença una nova fila cada dilluns (N=1)
if ($dataIteracio->format('N') == 1) {
    echo "<tr>";
}

// -----
// Determinació d'Estils
// -----
$classe = '';

// A) És el dia actual?
if ($dataIteracio->format('Y-m-d') === $dataAvui->format('Y-m-d')) {
    $classe = 'avui';
}

// B) És del mes que estem mostrant?
elseif ($dataIteracio->format('m') !== $dataMesActual->format('m')) {
    $classe = 'altre-mes';
}

// Pintar la cel·la (j: dia del mes sense zero inicial)
echo "<td class='$classe'>";
echo $dataIteracio->format('j');
echo "</td>";

// Tanca la fila cada diumenge (N=7)
if ($dataIteracio->format('N') == 7) {
    echo "</tr>";
}

// Avançar al dia següent
$dataIteracio = $dataIteracio->add($intervalDia);
}
?>
</table>
</body>
</html>

```

## Explicació OOP

1. **Fiabilitat amb `DateTimeImmutable`**: En lloc d'utilitzar múltiples crides a `date()` i la funció ambigua `strtotime()`, ara tens un objecte `$dataMesActual` que representa el mes, i un objecte `$dataIteracio` que es modifica utilitzant el mètode `->modify()` o `->add()` juntament amb `DateInterval`.
2. **Càlculs Clars**: El càlcul del primer dilluns (`$dataIteracio = $dataMesActual->modify("-$diesARestar days")`) és molt més explícit i fiable que les manipulacions de strings de l'exemple inicial.

3. **Iteració Sencilla:** El bucle `while` avança dia a dia utilitzant `$dataIteracio = $dataIteracio->add($intervalDia)`, aprofitant la lògica de dates de PHP per gestionar canvis de mes i anys automàticament.
4. **Consistència:** Es calcula la `$dataFiGraella` per garantir que la taula del calendari sempre tingui **sis setmanes completes** (6 \* 7 dies), evitant que la taula es redimensioni segons el dia de la setmana en què acabi el mes.

# Exemple Calendari. OOP + classe calendari

Un calendari, aplicant l'**Orientació a Objectes (OOP)** a PHP, utilitzant la classe `DateTimeImmutable` per a una manipulació de dates més fiable i encapsulant la lògica de generació del calendari en una classe `Calendari`. Aquesta solució separa la lògica (`Calendari.php`) de la presentació (`index.php`) i utilitza iteradors per generar les cel·les sense lògica complexa d'indexació.

## Implementació Orientada a Objectes (`Calendari.php`)

Crearem la classe `Calendari` que s'encarregarà de tots els càlculs de dates i de generar el HTML del calendari.

PHP

```
<?php

class Calendari
{
    /** @var DateTimeImmutable Data del mes que es vol mostrar */
    private DateTimeImmutable $dataMes;

    /** @var DateTimeImmutable Data del primer dia a iterar (el primer dilluns visible) */
    private DateTimeImmutable $dataIniciIteracio;

    /** @var DateTimeImmutable Data de l'últim dia a iterar (l'últim diumenge visible) */
    private DateTimeImmutable $dataFiIteracio;

    /**
     * Constructor
     * @param DateTimeImmutable|null $dataMes Si és null, s'utilitza la data actual.
     */
    public function __construct(?DateTimeImmutable $dataMes = null)
    {
        // Posicionem la data al primer dia del mes seleccionat (o el mes actual)
        $any = $dataMes?->format('Y') ?? date('Y');
        $mes = $dataMes?->format('m') ?? date('m');
```

```

        // Utilitzem createFromFormat per garantir el primer dia
        $this->dataMes = DateTimeImmutable::createFromFormat('Y-m-d', "$any-$mes-01");

        $this->calcularDatesIteracio();
    }

    /**
     * Calcula la data d'inici (primer dilluns) i fi (últim diumenge)
     * de la graella, ajustant-se exactament al mes.
     */
    private function calcularDatesIteracio(): void
    {
        $primerDiaMes = $this->dataMes;
        $darrerDiaMes = $this->dataMes->modify('last day of this month');

        // 1. Càlcul de la data d'inici de la iteració (el primer dilluns visible)
        $diaSetmanaPrimerDia = (int)$primerDiaMes->format('N'); // 1 (Dl) a 7 (Dg)
        $diesARestar = $diaSetmanaPrimerDia - 1;
        $this->dataIniciIteracio = $primerDiaMes->modify("-$diesARestar days");

        // 2. Càlcul de la data de finalització de la iteració (l'últim diumenge visible)

        // Busquem el dia de la setmana del darrer dia del mes
        $diaSetmanaDarrerDia = (int)$darrerDiaMes->format('N'); // 1 (Dl) a 7 (Dg)

        // Necessitem avançar fins al diumenge (dia 7)
        // Si el mes acaba un diumenge (7), sumem 0 dies.
        // Si el mes acaba un dissabte (6), sumem 1 dia.
        $diesASumar = 7 - $diaSetmanaDarrerDia;

        // L'últim dia real de la graella és el diumenge de l'última setmana
        $this->dataFiIteracio = $darrerDiaMes->modify("+$diesASumar days");
    }

    /**
     * Retorna el títol del calendari (Mes i Any)
     */
    public function getTitol(): string
    {
        // Exemple de formatació: Setembre - 2025
        return $this->dataMes->format('F') . ' - ' . $this->dataMes->format('Y');
    }

    /**
     * Mètode principal per generar el HTML del calendari
     */
    public function generaTaulaHTML(): string
    {

```

```

$html = '<table class="calendari-taula" border="1" cellspacing="0"
cellpadding="0">';

// Capçalera de dies de la setmana
$html .= '<tr>';
$diesSetmana = ['Dl', 'Dm', 'Dc', 'Dj', 'Dv', 'Ds', 'Dg'];
foreach ($diesSetmana as $dia) {
    $html .= "<th>$dia</th>";
}
$html .= '</tr>';

// Cos del calendari
$dataActual = $this->dataIniciIteracio;
$avui = new DateTimeImmutable('today');
$intervalDia = new DateInterval('P1D');

while ($dataActual < $this->dataFiIteracio) {

    // Nova fila (<tr>) cada dilluns (N=1)
    if ($dataActual->format('N') == 1) {
        $html .= '<tr>';
    }

    // Determinació d'estils
    $esDelMesActual = $dataActual->format('m') === $this->dataMes-
>format('m');
    $esAvui = $dataActual->format('Y-m-d') === $avui->format('Y-m-d');

    $classe = $esAvui ? 'avui' : ($esDelMesActual ? 'mes-actual' : 'altre-
mes');

    $html .= "<td class='$classe'>";
    $html .= $dataActual->format('j');
    $html .= '</td>';

    // Tancament de fila (</tr>) cada diumenge (N=7)
    if ($dataActual->format('N') == 7) {
        $html .= '</tr>';
    }

    // Avançar al dia següent amb l'interval
    $dataActual = $dataActual->add($intervalDia);
}

$html .= '</table>';
return $html;
}
}

```

## Ús de la Classe (`index.php`)

El fitxer principal es simplifica molt, ja que tota la lògica es troba a la classe `Calendari`.

PHP

```
<?php
// index.php

require_once 'Calendari.php';

// Opcional: Establir el fus horari per consistència
date_default_timezone_set('Europe/Madrid');

// Per a provar un mes amb menys setmanes (Octubre 2025 comença Dimecres)
// $calendari = new Calendari(new DateTimeImmutable('2025-10-01'));

// Crear l'objecte Calendari per al mes actual (Setembre 2025)
$calendari = new Calendari();

$titol = $calendari->getTitol();
$taula_html = $calendari->generaTaulaHTML();
?>

<!DOCTYPE html>
<html lang="ca">
<head>
    <meta charset="UTF-8">
    <title>Calendari Ajustat: <?php echo $titol; ?></title>
    <style>
        .calendari-taula {
            width: 100%;
            max-width: 500px;
            border-collapse: collapse;
            text-align: center;
        }
        .calendari-taula th, .calendari-taula td {
            padding: 8px;
            border: 1px solid #ccc;
        }
        .calendari-taula .avui {
            background-color: #ffe0b2;
            font-weight: bold;
        }
        .calendari-taula .altre-mes {
            color: #aaa;
        }
    </style>
</head>
<body>
    <div class="calendari">
        <div class="titol"><?php echo $titol; ?></div>
        <table class="taula">
            <tr>
                <td colspan="7"><div class="avui"><?php echo $avui; ?></div></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $març; ?></td>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $febrer; ?></td>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $gener; ?></td>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $desembre; ?></td>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $novembre; ?></td>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $octubre; ?></td>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $setembre; ?></td>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $agost; ?></td>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes"><?php echo $maig; ?></td>
                <td class="altre-mes"><?php echo $abril; ?></td>
                <td class="altre-mes"><?php echo $març; ?></td>
                <td class="altre-mes"><?php echo $febrer; ?></td>
            </tr>
            <tr>
                <td class="altre-mes"><?php echo $juliol; ?></td>
                <td class="altre-mes"><?php echo $juny; ?></td>
                <td class="altre-mes">&lt
```

```
        background-color: #f7f7f7;
    }
</style>
</head>
<body>
    <h3> Calendari Ajustat: <?php echo $titol; ?></h3>

    <?php echo $taula_html; ?>

</body>
</html>
```

# Avantatges de l'Orientació a Objectes

1. **Fiabilitat de Dates:** L'ús de `DateTimeImmutable` (en lloc de constants com `date("n")` o `date("Y")` i funcions com `strtotime()`) elimina l'ambigüitat en els càlculs i evita errors per manipulació del mateix objecte (immutabilitat).
2. **Encapsulament:** Tota la lògica de càlculs (com saber el primer dilluns, els dies a restar, etc.) està encapsulada dins la classe `Calendari`, fent que el codi sigui molt més net i fàcil de mantenir.
3. **Flexibilitat:** El constructor ara pot acceptar qualsevol data, permetent generar calendaris per a qualsevol mes sense haver de modificar la lògica interna del codi.
4. **Separació de Responsabilitats:** El fitxer `index.php` només s'encarrega de crear l'objecte i mostrar el seu resultat (`generaTaulaHTML()`), separant la **Lògica** (la classe) de la **Presentació** (el HTML).