

Folla 5.2. POO 2. Propiedades e métodos estáticos. Serialización en ficheiros.

1.- Imos facer unha páxina web **flotaVehiculos.php** que nos permita xestionar unha flota de vehículos, que iremos gardando nun array, e poderemos gardar nun ficheiro empregando serialize.

Define unha clase **vehiculo.php** que teñan como propiedades private a *matrícula*, *modelo*, *kms*. A clase **vehiculo** deberá ter os seguintes métodos:

- x un **construtor** que reciba a *matrícula*, *modelo* e *kms* como argumentos e garde eses valores.
- x **getMatricula()**, **getModelo()**, **getKms()**.
- x **mostraEnTR()**: Debe devolver unha cadea de texto poñendo cada propiedade dentro dun <td>, formando unha fila de táboa en HTML (empezando por <tr> e rematando en </tr>).

Na nosa páxina iremos teremos un formulario para introducir vehículos novos: matrícula, modelo, kms, que chamará á mesma páxina **flotaVehiculos.php** para crear un vehículo novo, e mostrará debaixo do formulario todos os vehículos do array nunha táboa. Gardaremos todos os vehículos creados nunha variable **\$_SESSION['flota']** que será un array de vehículos.

Engadiremos un botón **GARDAR en FICHEIRO** que permita gardar nun ficheiro flota.txt o array completo.

Engadiremos un botón **RECUPERAR de FICHEIRO** que permita recuperar o array de obxectos desde o ficheiro **flota.txt**.

Teremos tamén un botón **Pechar Sesión**, que pechará a sesión, e fará que o array (e tamén a táboa) non teña ningún vehículo.

OLLO: Comproba que xusto despois de GARDAR en ficheiro o array, o que almacena en disco o servidor (na carpeta /tmp do servidor docker) e o que se garda no ficheiro coincide

OPCIONAL: EXEMPLO DE HORAS E DATAS

2. Modifica a clase Data que se amosa máis abaixo, para que a propiedade calendario non sexa accesible dende o exterior da clase. Debes engadir os seguintes métodos estáticos, e modificar o existente **getData()** :

- **getCalendario()** : que devolverá o valor desta propiedade
- **getHora()** : que devolvera a hora co seguinte formato HH:MM:SS
- **getDataHora()** : que chamará aos métodos **getData()** e **getHora()** para amosar tanto a data como a hora
- A saída que debe amosar é a seguinte, dependendo do día no que estamos:

```
Usamos o calendario: Calendario gregoriano
Hoxe é xoves 6 de Febreiro do 2025 e son as 09:48:31
```

A clase da que debes modificar é Data:

```
class Data
{
    public static $calendario = "Calendario gregoriano";

    public static function getData()
    {
        $ano = date('Y'); //LEMBRA QUE NOS DÁ O ANO ACTUAL
        $mes = date('m');
        $dia = date('d');
        return $dia . '/' . $mes . '/' . $ano;
    }
}
```

Para converter o date('d') a texto terás que definir un array, que logo empregarás na mensaxe da clase. E o mesmo cos meses:

```
private static dias = ['Domingo', 'Luns', ... ];
```