Folla 5.3. POO 3. Autocarga. Métodos máxicos. Interfaces. Clases abstractas. Lanzamento de excepcións.

1. **Métodos máxicos**. Crea unha clase **Artigo** con 2 propiedades (id e nome), non accesibles desde o exterior
a) Define un construtor que estableza o valor das propiedades que reciba
   b) Cando se clone un artigo o id increméntase en 1
   c) Emprega métodos máxicos para establecer e obter os valores da súas propiedades.
   d) Se mostramos o obxecto cun echo, chamarase a un método máxico **__toString()**, que á súa vez chamará a un método mostraArtigo() no que mostre os valores das súas propiedades
   e) Comproba que funciona todo, creando un articulo 'pantalla', clónao, asigna o nome a 'rato', e móstrao
2. **Métodos e clases abstractas**. Define unha clase abstracta de nome **Calculo** que teña como atributos $operando1 $operando2 e $resultado e que defina os métodos setOperando1, setOperando2, getResultado e un método abstracto calcular. A continuación define tres subclases desta clase que teñen como obxectivo realizar as operacións de suma, resta e multiplicación. Para isto:
   a) Antes de realizar a operación debes comprobar que os operandos teñen algún valor.
   b) As clases **Calculo** e subclases **Suma**, **Resta** e **Multiplica** que crees deben estar nunha carpeta de nome clases.
   c) No script onde comprobes o funcionamento destes cálculos debes facer que se carguen automaticamente todas as clases que se atopen nesa carpeta:

```
<?php
...
$operacion=new Suma( );
$operacion->setOperando1(5);
$operacion->setOperando2(7);
$operacion->calcular( );
```

   d) Engade un construtor a Calculo para que reciba os valores cando se defina o obxecto.

3. Engade ao exercicio anterior u formulario web que permita introducir 2 valores, e un select para a operación. A páxina mostrará debaixo do formulario a operación e o seu resultado.
4. **Polimorfismo**. Comproba que funciona o exemplo dos apuntes de polimorfismo. Engade unha clase CocheHidroxeno que estenda a clase Coche, e comproba que nos obriga a definir o método consumir()
5. **Traits**. Crea un trait **DatosPersoa**, que garde nome, apelidos e idade, e os seus métodos de entrada e saída. Crea agora 2 clases Vendedor e Cliente, e empreguen ese trait, e comproba que podes empregar eses métodos para gardar e mostrar información nesas clases. Define o método __toString( ), que chama a mostrar Valores.
6. **Lanzamento de excepcións**. **Interface**. Modifica o exercicio 1 para que empregue a seguinte interface:

```
<?php
interface Comparar {
    public function comparar($value);
}
?>
```

Engade nos artigos unha propiedade prezo, que será o que comparemos. Cando mostramos un artigo queremos ver tamén este valor. O método ***comparar( )*** comprobará que o obxecto recibido é de tipo Artigo, se non lanzará unha excepción cunha mensaxe

Fai un exemplo que compare 2 artigos con este método ***comparar( )***