



BASE DE DATOS EN PHP

## DESARROLLO WEB EN ENTORNO

### EJERCICIO CLASE ABSTRACTA

1º. Vamos a crear una clase denominada Figura de la cual no se podrán crear objetos. Esta clase debe tener lo siguiente:

- 2 variables de tipo double denominadas x e y.
- Constructor con argumentos que inicialicen x e y
- Método abstracto denominado área que retorne un double.

2º. A continuación crearemos dos clases publicas denominadas Cuadrado y

Circulo, las cuales heredarán de la clase Figura.

La clase Circulo tendrá lo siguiente:

- Una variable de clase de tipo double denominada radio.
- Tendrá el constructor con argumentos.
- Desarrollará la lógica necesaria del método área de la clase padre.

La clase Cuadrado tendrá lo siguiente:

- Una variable de clase de tipo double denominada lado.
- Tendrá el constructor con argumentos.
- Desarrollará la lógica necesaria del método área de la clase padre.

## EJERCICIOS INTERFACES

Creemos un proyecto llamado **EjercicioInterfaces** crearemos una interface llamada **Vehiculo** que deberán implementar nuestras clases y que contendrá:

☐ Dos métodos que retornarán un String llamados frenar y acelerar. Cada uno de los métodos tendrá un argumento de tipo entero llamado distancia.

Después crearemos dos clases llamadas Coche y Moto que implementaran la interfaz creada.

Las clases se describen a continuación:

### Clase Coche

Esta clase implementará la interfaz Vehiculo y contará con una propiedad de clase denominada velocidad, de tipo entero, inicializada a cero.

El método **frenar** tiene que retornarnos un mensaje como este **return "El coche ha frenado ya y va a "+velocidad+"km/hora";**

El método **acelerar** tiene que comprobar que no superamos la velocidad máxima fijada en la interfaz y nos tiene que retornar al final el siguiente mensaje **return "El coche ha acelerado y va a "+velocidad+"km/hora";**

### **Clase Moto**

Esta clase implementará la interfaz Vehiculo y contará con una variable de clase denominada velocidad, de tipo entero, inicializada a cero.

El método **frenar** tiene que retornarnos un mensaje como este **return "La moto ha frenado ya y va a " + velocidad + "km/hora";**

El método **acelerar** tiene que comprobar que no superamos la velocidad máxima fijada en la interfaz y nos tiene que retornar al final el siguiente mensaje **return "La moto ha acelerado y va a " + velocidad + "km/hora";**

**En el método main haremos lo siguiente:**

Creamos una matriz de tipo Vehiculo con dos cajones.

En cada cajón metemos un objeto (Coche y Moto)

Recorremos la matriz de tipo Vehiculo mostrando por consola los datos de los métodos frenar y acelerar que hemos introducido a mano