

## PHP: Trabajo práctico N°5

Los siguientes ejercicios se deben realizar en una web que permita ingresar a index.php y mediante un menú seleccionar cual de ellos visualizar. Todas las pantallas deben tener el mismo estilo visual:

1. **Insta-gramo:** Realizar una web que muestre todas las imágenes que contiene en la carpeta /imagenes con su respectivo nombre. Al final de dicha web debe haber un formulario que permita subir una imagen con un nombre. (Entregar con 3 imágenes cargadas)
2. **Insta-reciclando (para cuidar el medio ambiente):** Reutilizando el ejercicio anterior, Realizar una web que liste todos los nombres de imagenes que contiene en la carpeta /imagenes y al lado de dicho nombre un link que indique “mostrar imagen” que lleve a una segunda pantalla donde efectivamente se muestre dicha imagen. (Entregar con 3 imágenes cargadas)
3. **Lanzar dados:** Realizar una web que permita tocar un botón “lanzar dados” que lanza 2 dados sin cargar (misma probabilidad de tocar cualquiera de sus lados) y muestre en pantalla el resultado. Dicho resultado consta de un número que es la suma obtenida de ambos dados, y dos imágenes de dados, con los valores resultantes en cada uno (Utilizar función rand() o similar)
4. **Contador de visitas... extraterrestres:** Crear una web que permita ingresar en un formulario nombre y planeta perteneciente. Luego del envío del formulario, muestre listado de valores cargados previamente y un contador de cuantas visitas fueron extraterrestres. (Almacenar dichos valores en archivos del lado del servidor)
5. **Mostrando configuraciones:** Mostrar en pantalla las siguientes 5 configuraciones almacenadas en un archivo config.ini (Utilizar función parse\_ini\_file):
  - a. host de base de datos
  - b. puerto de base de datos
  - c. nombre de base de datos
  - d. usuario de base de datos
  - e. clave de base de datos
6. **La Matrix... digo, matriz:** Dada una matriz de 10 por 10, cargada con valores de 1 a 100:
  - a. Recorrer la matriz con un sólo FOR y mostrar en pantalla los valores que componen la diagonal principal (1,12,23,etc)
  - b. Recorrer la matriz con un sólo FOR y mostrar en pantalla los valores que componen la diagonal secundaria (10,19,28,etc)
  - c. Recorra la matriz (Ahora si con 2 for) y sume todos los valores que contiene. Muestre el resultado

7. **Buscando una palabra en un pajar, más bien en un String:** Crear una función que reciba una palabra a buscar y un texto, y devuelva cuantas ocurrencias hubo de la misma. Recorrer el texto String caracter a caracter sin utilizar funciones de String, la firma de la función será similar a: buscar(clave, texto)

Los siguientes tres ejercicios tienen la intención de conocer el paradigma de objetos. Donde no debemos valernos de estructuras de control if-while-for para modelar la solución a los problemas, si no, de familias de objetos polimórficos que envían mensajes e interactúan entre sí. El mismo enunciado dará los pasos a seguir, y en la próxima clase se trabajará en conjunto explicando los motivos de este modo de trabajo y despejando dudas.

8. **Piedra, papel o tijera:** Crear las clases Piedra, Papel y Tijera. Todas ellas deberán tener los siguientes métodos
- constructor(nombreJugador)
  - competirContra( otro )
  - teCompiteTijera( otro )
  - teCompitePiedra( otro ),
  - teCompitePapel( otro ),
  - ganador() -> devuelve el nombre del jugador "o la palabra Empate, si lo hubo"

las funciones competirContra() buscan implementar el patrón **double dispatch**, y las otras tres se limitan a responder si ganan ellos, el contrincante o hay Empate (\$this, \$otro o new Empate() ). Al finalizar, se le indicará que indique el ganador(). Algunos ejemplos para ayudar:

```
Tijera{
    ...

    funcion competirContra(otro){
        return otro.teCompiteTijera( this )
    }

    ...
}
```

```
-----

$jugadaFacu = new Tijera("Facu");
$jugadaAle = new Piedra("Ale");

$resultado = $jugadaFacu.compiteContra($jugadaAle);
echo $resultado.ganador(); // Imprime Ale

-----
```

## 9. Calcula Doris - No olvidemos imprimir el resultado final!

Crear los objetos Operando, Sumar, Restar. Todos deben responder al método resolverEcuacion() de modo tal de poder escribir:

```
$resultado = new Sumar(
    new Restar( new Operando(5), new Operando(3)) ,
    new Sumar( new Operando(3), new Operando(4))
).resolverEcuacion();
```

y resultado debe ser 9. Crear ahora el objeto Multiplicar, agregarlo a la ecuación y multiplicando todo por 2, ver que sucede ¿Tuvimos que modificar algo de lo existente?

## 10. Packman come tutti

Crear los objetos packman, fantasma , pildora y fantasma comestible.

Packman debe responder al método: chocaContra( elemento ) y lanzar una excepción si se queda sin vidas. Dicha Excepcion debe llamarse FinPartida, y debe indicar la cantidad de puntos obtenidos.

Por defecto packman comienza con 3 vidas y cero puntos.

Si packman choca un fantasma, resta una vida

Si packman choca un fantasma comestible, suma 100 puntos

Si packman choca una pildora, suma 10 puntos

Si packman se queda sin vidas, lanza excepcion FinPartida.

Imprimir en pantalla los puntos obtenidos.

Ejemplo clarificador (se supone):

```
$packman = new Packman();
$pildora = new Pildora();
$fantasma = new Fantasma();
$fantasmaComestible = new FantasmaComestible();

try(){
    packman.chocaContra($fantasmaComestible);
    packman.chocaContra($fantasmaComestible);
    packman.chocaContra($fantasmaComestible);
    packman.chocaContra($pildora);
    packman.chocaContra($pildora);
    packman.chocaContra($pildora);
    packman.chocaContra($fantasma);
    packman.chocaContra($fantasma)
    packman.chocaContra($fantasma)
} catch ( $finPartida ){
    echo $excepcion->mensaje() //imprime 330 puntos
}
```