

TRABAJO PRÁCTICO OBLIGATORIO

- Cumplir con las siguientes condiciones es parte de la evaluación del Trabajo Final:

En las clases previas a la defensa del trabajo práctico deben mostrar avances de la resolución y realizar consultas a los docentes. El día de la defensa, el programa debe funcionar y no producir errores de sintaxis, **todos los integrantes del grupo deben estar presentes y tener micrófono para poder exponer y responder las preguntas de los docentes.**

- El trabajo final se puede desarrollar en grupo de máximo 3 (tres) alumnos.

Habrán Dos tareas en PEDCO para subir un archivo nombrado con el/los apellido/s del/os alumno/s **“apellidos.zip”** con:


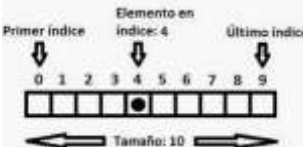






- Un archivo **“juegoApellidos.php”** con el **código fuente**, donde cada función implementada se encuentra debidamente documentada (como fue visto en los apuntes de teoría utilizando `/** */` y especificando la descripción de la función, tipo y nombre de parámetros, tipo de retorno), el nombre de la función y de los parámetros formales deben ser significativos de la funcionalidad que resuelven. (no se aceptan nombres del estilo: `function opcion5(...)`)
- Un archivo **“disenioEstructuras.pdf”** con apellido, nombre, legajo, mail, carrera, fecha de entrega del alumno, la **representación de las estructuras de datos utilizadas (con TODOS los datos de precarga utilizados en el programa).**

La primera tarea de PEDCO tendrá el nombre PreDefensa, donde se subirán los archivos antes de realizar la defensa oral del trabajo. Y la segunda tarea tendrá el nombre PostDefensa, donde uno de los integrantes del grupo deberá subir la reentrega en caso que el docente solicite modificaciones.

Consideraciones:

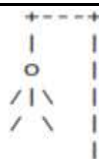
- En el archivo **disenioEstructuras.pdf** debe representar las estructuras de datos¹ indicando nombre de la estructura, índices y valores. Esto le ayudará a identificar cómo crear, acceder y recorrer las estructuras de datos. Este diseño debe ser el primer producto a desarrollar por el grupo.
- Deberá descargar el archivo **juegoApellidos.php** con la plantilla del programa a desarrollar, y renombrar el archivo reemplazando la palabra Apellidos por los del grupo. Este archivo contiene la estructura general del programa y deberá ser completada por los alumnos.
En el código encontrará comentario del tipo `/*>>>` indicación `<<<*/`, con alguna indicación que debe reemplazar. Puede buscar las ocurrencias de `/*>>>` en el archivo (ctrl + f)
- El trabajo será desarrollado íntegramente en PHP. Pero no pierda de vista tener papel y lápiz para realizar cualquier análisis / diseño.
- Recuerde que es un Trabajo Final que será evaluado, utilice un vocabulario apropiado tanto en el código fuente como en los mensajes que mostrará en pantalla.
- Antes de comenzar a codificar, leer el enunciado completamente con el fin de hacer un buen análisis del problema a resolver: datos de entrada, objetivos, estructuras de datos que debe utilizar, posibles funciones a implementar, etc.
- Si implementa un módulo, debe asegurarse que funciona correctamente de forma individual (test o prueba unitaria), es decir, testear cada módulo con distintos valores de parámetros y determinar si la salida es la esperada.
- El día de la defensa debe explicar la ejecución del programa y el código fuente, además de responder las preguntas que realizará el docente.

Resumen:

 <p>Leer atentamente el enunciado del problema resolver</p>	 <p>Identifique las estructuras de datos diseñadas para resolver el problema, qué tipos de datos almacenan.</p>  <p>Represéntelas en papel según lo visto en la teoría, ¡Le ayudará a entender para qué se usan y cómo recorrerlas y accederlas! Genere el archivo “disenioEstructuras”</p>	 <p>Descargar el archivo juegoApellidos.php donde otro programador comenzó a resolver el problema. Renombrar el archivo a juegoXYZ.php, siendo XYZ su apellido</p>	 <p>Relacionar el enunciado con lo programado en juegoXYZ.php, para determinar qué está programado y qué falta. Esto le ayudará a reusar funciones y tener que codificar menos</p> 	 <p>¡Manos a la obra! Comenzar a completar el código</p>  <p>Los símbolos <code>/*>>></code> le ayudarán a identificar qué completar en el código fuente</p>
--	---	---	---	---

¹ Arreglos/colecciones

ENUNCIADO DEL PROBLEMA A RESOLVER



El ahorcado es un juego de adivinanza que se desea implementar en lenguaje PHP para hacer una competencia. El juego consiste en adivinar una palabra de la que se conoce una pista y la cantidad de letras de la palabra. En cada partida, jugador adivinador indica una secuencia de letras:

Por cada letra:

- Si la letra pertenece a la palabra, se descubre dicha letra en todas las posiciones donde se encuentra
- Si la letra no pertenece a la palabra, el jugador descuenta un intento de los 6 (valor constante) intentos con los que se configura el juego.



¿Cuándo termina cada partida y cuál es el puntaje?:

- La partida termina cuando el jugador adivinador completa la palabra y gana. En este caso el jugador obtiene el **puntaje** asociado a la palabra más la cantidad de intentos que le quedaron.
- O bien, la partida termina cuando el jugador adivinador descuenta todos los intentos y pierde. Es decir, queda ahorcado. En este caso el **puntaje** es cero.

Menú de Usuario:

El usuario ejecuta el programa, y el juego se presenta como un menú de opciones con el que usuario puede interactuar:

- Jugar con una palabra aleatoria
- Jugar con una palabra elegida
- Agregar una palabra al listado
- Mostrar la información completa de un número de juego
- Mostrar la información completa del primer juego con más puntaje.
- Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
- Mostrar la lista de palabras ordenada por orden alfabético.
- Salir del juego

Descripción de las Estructuras de datos:

(Momento de tomar papel y lápiz, y dibujar la representación!...)

Para desarrollar el programa debe utilizar las siguientes estructuras de datos:

Estructura a) Una estructura indexada de arreglos asociativos que almacene información de las palabras disponibles para jugar. Cada arreglo asociativo tendrá el siguiente formato: ("**palabra**"=> "**palabra**", "**pista**"=> "**frase que ayuda a descubrir la palabra**", "**ganaPuntos**"=> **puntosSiGana**)

Por ejemplo:

```
coleccionPalabras[0] = ("palabra"=> "papa", "pista"=> "se cultiva bajo tierra", "ganaPuntos"=>7 )
coleccionPalabras[1] = ("palabra"=> "hepatitis", "pista"=> "enfermedad que inflama el hígado", "ganaPuntos"=> 7)
coleccionPalabras[2] = ("palabra"=> "volkswagen", "pista"=> "marca de vehiculo", "ganaPuntos"=> 10)
...
```

Estructura b) Una estructura indexada de arreglos asociativos que almacenará información de las letras de la palabra seleccionada para jugar. Esta estructura se utilizará sólo para poder jugar y sólo se cargará cuando se determine la palabra.

Cada arreglo asociativo tendrá el siguiente formato: ("**letra**"=> "**letra**", "**descubierta**"=> **true/false**)

Por ejemplo si la palabra elegida es "papa", entonces la estructura para comenzar a jugar es la siguiente:

```
coleccionLetras[0] = ("letra"=> "p", "descubierta"=> false)
coleccionLetras[1] = ("letra"=> "a", "descubierta"=> false)
coleccionLetras[2] = ("letra"=> "p", "descubierta"=> false)
coleccionLetras[3] = ("letra"=> "a", "descubierta"=> false)
```

El índice de la estructura indica el orden de las letras para formar la palabra.

(Para esta estructura en particular, tome alguna palabra que usted haya agregado a la colección de palabras, y represéntela!...)



Estructura c) una estructura indexada con los juegos realizados indicando el puntaje obtenido y la índice de la estructura del inciso a) con la que se jugó.

coleccionJuegos[0] = ("**puntos**"=> 0, "**indicePalabra**" => 1) significa que en la primer jugada obtuvo 0 puntos y adivinó la palabra 1 (hepatitis)

coleccionJuegos [1] = ("**puntos**"=> 10, " **indicePalabra** " => 2) significa que en la segunda jugada obtuvo 10 puntos de la palabra 2 (volkswagen)

coleccionJuegos [2] = ("**puntos**"=> 0, " **indicePalabra** " => 1) significa que en la tercer jugada obtuvo 0 puntos de la palabra 1 (hepatitis)

coleccionJuegos [3] = ("**puntos**"=> 8, " **indicePalabra** " => 0) significa que en la cuarta jugada obtuvo 8 puntos de la palabra 0 (papa)

etc.

Requerimientos del Menú de Usuario:

Cada opción del menú de opciones permite ejecutar una serie de funciones que implementan la siguiente funcionalidad *(por opción se hace una descripción de la funcionalidad, luego el alumno usará distintas funciones por opción):*

1. **Jugar con una palabra aleatoria:** utilizando la función **rand** de PHP seleccionará el índice aleatorio entre 0 y la cantidad de palabras. Con la palabra y la cantidad de intentos deberá jugar un juego y guardar los puntos y el índice de la palabra en la colección de juegos.
Cuando comience a jugar deberá mostrar la pista de la palabra.
2. **Jugar con una palabra elegida:** Deberá solicitar al usuario el índice válido entre 0 y la cantidad palabras. Con la palabra y la cantidad de intentos deberá jugar un juego y guardar los puntos y el índice de la palabra en la colección de juegos.
Cuando comience a jugar deberá mostrar la pista de la palabra.
3. **Agregar una palabra al listado:** Deberá solicitar una palabra (que convertirá a minúsculas utilizando función predefinida de PHP). Si la palabra No existe en la colección de palabras, solicitará la pista y el puntaje y la misma será agregada a la colección de palabras. Observación: La nueva palabra debe estar disponible para jugar en las opciones 1 y 2 del menú.
4. **Mostrar la información completa de un número de juego:** Deberá solicitar al usuario un nro de juego entre 0 y la cantidad de juegos existentes en la colección de juegos y mostrar la información del juego.
5. **Mostrar la información completa del primer juego con más puntaje:** Deberá buscar el juego con más puntaje y mostrar la información del juego.
6. **Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario:** Deberá solicitar al usuario un puntaje, buscar el primer juego que supere dicho puntaje y mostrar el juego. Considere que el usuario tiene la posibilidad de ingresar un puntaje muy alto.
7. **Mostrar la lista de palabras ordenada por orden alfabético.** Utilizar el método de ordenamiento visto en teoría que sea más adecuado para ordenar el arreglo de **Palabras** de mayor a menor por orden alfabético. Utilizar la función predefinida de PHP "print_r" para imprimir la estructura de datos. Cuando utilice la función "print_r" escriba un comentario en el código fuente explicando qué hace esta función. (Queda a cargo del alumno investigar qué hace print_r utilizando el manual de PHP disponible en internet)
8. **Salir:** termina el programa principal.

Requerimientos para la programación del juego:

(Algunos de los siguientes requerimientos ya están desarrollados en la plantilla entregada por la cátedra. Tiene que poder identificarlos en la plantilla y vincularlos con estos requerimientos):

1. Implementar una función de precarga de palabras que retorne un arreglo de palabras, según la estructura indicada en el inciso a) (Estructura a)) de la sección de estructuras.
2. Implementar una función de precarga que retorne un arreglo de juegos jugados según la estructura indicada en c) (Estructura c)) de la sección de estructuras.
3. Implementar una función que a partir de una palabra, retorne la estructura indicada en el inciso b) (Estructura b)) de la sección de estructuras. Es decir una función que divida la palabra en letras.

4. Para visualizar el menú de opciones (que siempre es el mismo), implementar una función **seleccionarOpcion** que muestre las opciones del menú en la pantalla, le solicite al usuario una opción válida (si la opción no es válida vuelva a solicitarla en la misma función hasta que la opción sea válida), y retorne **el número** de la opción. La última opción del menú debe ser "Salir".
5. Implementar una función que a partir de la colección de palabras (Estructura a)) y una palabra, la función retorne true si la palabra existe en la colección de palabras.
6. Implementar una función que a partir de la colección de letra (Estructura b)) y una letra, la función retorne true si la letra existe en la colección de letras.
7. Implementar una función que cargue una nueva palabra (palabra, pista y puntaje) en la colección de palabras (Estructura a)).
8. Implementar una función que retorne un índice aleatorio (utilice la función rand).
9. Implementar una función que solicite al usuario un índice entre dos valores. La función debe validar el dato ingresado por el usuario.
10. Implementar una función que determine si una colección de letras (Estructura b)) fue descubierta. Es decir todas sus posiciones indican descubierta == true.
11. Implementar una función que solicite una letra al usuario, y que valide que el string ingresado tenga un único carácter.
12. Implementar una función que modifique la colección de letras (Estructura b)), destapando o descubriendo todas las ocurrencias de una letra.
13. Implementar una función que a partir de la colección de letras (Estructura b)), genere un string con las letras descubiertas y con * en las posiciones donde la letra aún no fue descubierta.
14. Implementar la función jugar.
15. Implementar la función agregarJuego (es decir, los datos de la partida) en la colección de juegos (Estructura c)).
16. Implementar la función mostrarPalabra (datos de la Estructura a)).
17. Implementar la función mostrarJuego (datos de la Estructura b)).
18. Implementar una función que a partir de la colección de juegos obtenga el índice de la partida con mayor puntaje (Estructura c))
19. Implementar una función que a partir de la colección de juegos y un puntaje, la función obtenga el índice de la primer partida que supere el puntaje ingresado por parámetro. Si no existe la función debe retornar -1.
20. Implementar una función sin retorno que ingrese la colección de palabras (Estructura a)) y muestre la colección de palabras ordenada por orden alfabético utilizando la función print_r.
21. Implementación del Programa Principal:
 - a. Precargar las estructuras de palabras y de juegos.
 - b. Inicializar la cantidad de intentos en 6 (seis)
 - c. Repetir el menú de opciones mientras la opción seleccionada no sea la opción Salir.
 - d. Cuando el usuario selecciona la opción del menú, debe invocar a la/s función/es necesarias. Salvo algunas excepciones, debe contar con funciones con parámetros formales y retorno. Asesorarse con la Cátedra para implementar las funciones correctamente de modo que los resultados de las funciones puedan ser reusado.
 - e. Investigar la instrucción **switch** en el manual de PHP. ¿a qué tipo de estructura de control vista en teoría corresponde? Escriba un comentario sobre la instrucción en el código fuente.
22. En el desarrollo utilizar funciones de String predefinida por PHP (strtolower, strtoupper, strlen, etc.)

Algunas capturas de pantalla de lo que espera el usuario:

```
Simbolo del sistema - php: ahorcado.php
D:\misphp>php ahorcado.php

( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:20

Indique una opción válida:█
```

```
Símbolo del sistema - php. ahorcado.php
( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:20
Indique una opción válida:2
-----
Seleccione un índice entre 0 y 2: 0
Pista: se cultiva bajo tierra

-----
Ingrese una letra: p
La letra 'p' PERTENECE a la palabra.
Palabra a Descubrir: p*p*
-----
Ingrese una letra: e
La letra 'e' NO pertenece a la palabra. Quedan 5 Intentos.
Palabra a Descubrir: p*p*
-----
Ingrese una letra:
```

```
Símbolo del sistema - php. ahorcado.php
Ingrese una letra: x
La letra 'x' NO pertenece a la palabra. Quedan 3 Intentos.
Palabra a Descubrir: p*p*
-----
Ingrese una letra: x
La letra 'x' NO pertenece a la palabra. Quedan 2 Intentos.
Palabra a Descubrir: p*p*
-----
Ingrese una letra: x
La letra 'x' NO pertenece a la palabra. Quedan 1 Intentos.
Palabra a Descubrir: p*p*
-----
Ingrese una letra: x
La letra 'x' NO pertenece a la palabra. Quedan 0 Intentos.
Palabra a Descubrir: p*p*
|||||AHORCADO AHORCADO|||||
-----
( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:
```

```
Símbolo del sistema - php. ahorcado.php
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:3
-----
Ingrese la palabra: HEPATITIS
La palabra ya existe
Ingrese la palabra: PAPA
La palabra ya existe
Ingrese la palabra: Matematica
Ingrese pista: Es una materia del secundario
Ingrese puntaje: 6
-----
( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:
```



```
Simbolo del sistema - php_ahorcado.php
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:4
-----
Seleccione un índice entre 0 y 4: 4

<-<-< Juego 4 >->->
Puntos ganados: 8
Información de la palabra:
palabra: papa
pista: se cultiva bajo tierra
puntosPalabra: 7
-----

( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:.
```

```
Simbolo del sistema - php_ahorcado.php
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:5
-----

<-<-< Juego 1 >->->
Puntos ganados: 10
Información de la palabra:
palabra: volkswagen
pista: marca de vehiculo
puntosPalabra: 10
-----

( 1 ) Jugar con una palabra aleatoria
( 2 ) Jugar con una palabra elegida
( 3 ) Agregar una palabra al listado
( 4 ) Mostrar la información completa de un número de juego
( 5 ) Mostrar la información completa del primer juego con más puntaje
( 6 ) Mostrar la información completa del primer juego que supere un puntaje indicado por el usuario
( 7 ) Mostrar la lista de palabras ordenada por puntaje
( 8 ) Salir
Indique una opción válida:
```