

## **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 arrojar errores de sintaxis.

**El trabajo final es de a pares o individual.** Las fechas de entrega se publicaran en Pedco y Discord.

Habrà una tarea en PEDCO para subir la resolución antes de la defensa y otra tarea luego de la defensa si es necesario corregir errores. La entrega consta de dos archivos:

- i) Un archivo "programaGrupoXXX.php" con el **código fuente** php
- ii) Un archivo "Estructuras.pdf" con apellido, nombre, legajo, mail, carrera, fecha de entrega del alumno y la **representación de las estructuras de datos utilizadas, con los datos almacenados en las estructuras.**

### **Consideraciones:**

- Debe representar las estructuras de datos indicando índices/claves y valores. Esto le ayudará a identificar cómo crear, acceder y recorrer las estructuras.
- El trabajo será desarrollado en PHP.
- 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.
- Cada función implementada debe ser debidamente documentada (como fue visto en los apuntes de teoría utilizando `/** */`, especificando descripción de la función, tipo y nombre de los parámetros, tipo del retorno). Los nombres de las funciones y de los parámetros formales deben ser acorde a la funcionalidad que resuelven.
- Si implementa una función, 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.

## ENUNCIADO DEL PROBLEMA A RESOLVER:

La cuarentena se extendió demasiado y un profesor de matemáticas necesita hacer un programa para mostrarles a sus alumnos varios conceptos matemáticos sobre matrices de números.

El programa mostrará un menú de opciones con el que interactúa el usuario:

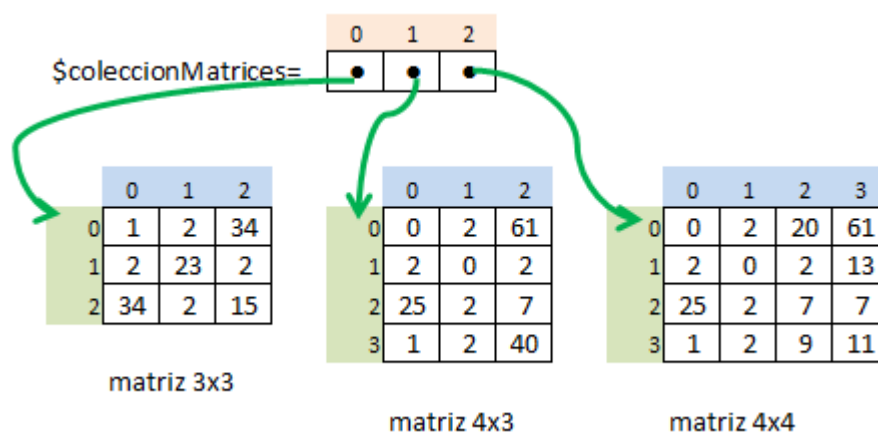
### Menú de opciones que se mostrará en pantalla:

- 1) Mostrar cantidad de matrices del programa
- 2) Mostrar una matriz
- 3) Ingresar una matriz NxM
- 4) Mostrar una matriz en números Romanos
- 5) Mostrar el resumen de una matriz
- 6) salir

Para la realización de estas opciones es necesario desarrollar ciertas funciones que consideramos los ayudarán a resolver todos los problemas.

**Funciones mínimas a desarrollar:** (relea las consideraciones a tener en cuenta para el desarrollo del trabajo)

- a) **una función** que muestre en pantalla las opciones del menú, solicite un número válido (si no lo es tiene que volver a pedirlo) y retorne el valor de la opción seleccionada.
- b) **una función** cuyo nombre es obtenerColeccionMatrices sin parámetros de entrada y que retorna la siguiente colección de matrices



### **Donde:**

La matriz NxM en la posición 0 de la colección es de dimensión 3x3 (3filas y 3columnas)

La matriz NxM en la posición 1 de la colección es de dimensión 4x3 (4filas y 3columnas)

La matriz NxM en la posición 2 de la colección es de dimensión 4x4 (4filas y 4columnas)

Queda **a cargo del alumno**, agregar una matriz simétrica en la posición 3 de la colección.

- c) una función que dado un número entre 0 y 100 lo transforme en número romano. En caso que el número es 0 deberá retornar un guión "-".
- d) una función que dada una matriz NxM retorna true si es cuadrada, false caso contrario
- e) una función que dada una matriz NxM determina si es simétrica o no. Una matriz es simétrica si es cuadrada y cumple con las siguientes condiciones: si cada valor  $(i,j) = (j,i)$
- f) una función que dada una colección de matrices, una dimensión N y una dimensión M, retorna la colección de matrices con una nueva matriz NxM (agregar una posición más al arreglo). Internamente la función debe utilizar la función predefinida de php para obtener valores aleatorios entre 0 y 100. (queda a cargo del alumno investigar la función rand en el manual de php).
- g) una función que dada una matriz NxM la muestre en pantalla con el siguiente formato de filas y columnas:

0	2	61
2	0	2
25	2	7
1	2	40

- h) una función que dada una matriz de NxM retorne la suma de todos sus números.
- i) una función que dada una matriz de NxM retorne el promedio todos sus números.
- j) una función que dada una matriz de NxM retorne un arreglo asociativo con el resumen de la matriz con las siguientes claves: "dimension1", "dimension2", "suma", "promedio", "cuadrada" (valor "si"/"no"), "simetrica" (valor "si"/"no")

### **Explicación PROGRAMA PRINCIPAL y MENU de opciones:**

Debe desarrollar un programa principal que realice los siguientes pasos:

- Inicializa una variable con la colección de matrices previamente definidas en la función b).
- Pida opciones al usuario mientras la opción no sea salir del programa
- Por cada opción de menú debe realizar lo siguiente
  - **Opción 1-** Mostrar cantidad de matrices del programa: Mostrar cuántas matrices hay cargadas en la colección de matrices del programa.
  - **Opción 2:** Solicitar un nro de matriz al usuario y mostrarla en pantalla en el formato indicado en la opción g) de funciones mínimas a desarrollar.
  - **Opción 3:** Solicitar al usuario la dimensión N y la dimensión M, y agregar una matriz con número aleatorios a la colección de matrices

- **Opción 4:** Solicitar un nro de matriz al usuario y mostrar en pantalla con el formato indicado en la opción g), la matriz de números y la matriz transformada, donde cada número se convirtió a nro romano.
- **Opción 5:** Solicitar un nro de matriz al usuario, mostrar la matriz y mostrar el resumen de la matriz (en este caso utilice la función print\_r de php para mostrar el resumen)
- **Opción 6:** salir del programa, Termina la ejecución.

Aclaraciones sobre el práctico que surgen de las consultas de los alumnos:

### 1) ¿Por qué definir una función de precarga de la colección de matrices en lugar de cargar el arreglo en el principal?

En la teoría de la unidad 4 estudiamos los beneficios de la modularización:

1. Aumenta la legibilidad y comprensión
2. Construcción divide y vencerás permite que los equipos de programadores trabajen en módulos independientes
3. La modularización aísla las modificaciones, hace más manejable los cambios.
4. La modularización aísla errores
5. operación aparecerá una sola vez, es decir, reuso

Por ahí para ampliar la respuesta:

1. Aumenta la legibilidad y comprensión para otros programadores: Al inicio del programa principal invocar una función obtenerColeccionMatrices, que retorne la colección de matrices, da mucha más claridad/expresividad al código. Distinto sería ver varias líneas asignando matrices a una colección, y luego entender que en ese punto sólo se está generando un conjunto de matrices precargadas que se utilizarán en el programa.

3. Aísla las modificaciones. Cuando se necesite agregar más matrices en la precarga, se modifica la función agregando más líneas de código. De esta manera también se evita quitar legibilidad al programa principal. Además el día de mañana, se agregan herramientas de programación para manipular archivos o bases de datos, y podría reemplazar el código de la función para que acceder a un archivo/Base de Datos y que automáticamente cargue las matrices. Para el programa principal el cambio es transparente.

4. aísla errores: Si se detecta que las estructuras de datos precargadas son erróneas, se sabe cuál es la función que hay que modificar. Se puede llevar la función a otro archivo .php y testearla por separado.

2 y 5. Divide y vencerás, Reuso: se puede llevar esta función de precarga a una librería y que distintos programadores la utilicen.

### 2) Dimensiones de matrices de dimensión NxM.

Las dimensiones indican que N es la cantidad de filas, y M es la cantidad de columnas. En este ejercicio siempre estamos hablando de matrices, es decir hablamos de 2 dimensiones.

Ejemplos:

```
$matriz1x3 = [ [1,2,3] ];  
$matriz3x1 = [ [1] , [2] , [3] ];
```

### 3) ¿Cómo puedo formatear la matriz para mostrarla en pantalla?

Se puede recurrir a funciones de string <https://www.php.net/ref.strings>. Por ejemplo existe la función `str_pad`

### 4) Al mostrar la matriz, también hay que mostrar el resumen?

En este caso son opciones de menú diferentes.