

TRABAJO PRÁCTICO OBLIGATORIO: Estadísticas MENSUALES de Productos

Alumno: SAWICKI, MAURICIO Legajo: FAI-2256

Asignación del TP: 17/08/2022

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

El programa debe funcionar acorde a lo solicitado y no arrojar errores de sintaxis.

El alumno debe entregar:

Un archivo “estadisticasProductos.php” con el **código fuente**, donde cada función implementada se encuentra debidamente documentada (como fue visto en los apuntes de teoría utilizando `/** */` y especificando descripción de la función, tipo y nombre de parámetros, tipo de retorno), los nombres de la función y de los parámetros formales deben ser acorde a la funcionalidad que resuelven.

- **Consideraciones:**

- Debe representar las estructuras de datos (arreglos) indicando índices y valores. Esto le ayudará a identificar cómo crear, acceder y recorrer las estructuras.
- Antes de comenzar a codificar, leer el enunciado completamente (sobre todo los requerimientos obligatorios donde encontrará pautas para resolver el ejercicio.) 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. Por ejemplo puede crear un nuevo archivo .php con el módulo y un programa principal para testear si el resultado es correcto.
- 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 empresa de electrodomésticos Fàbregas requiere un sistema para evaluar estadísticas mensuales de ventas de sus productos. Para implementar el sistema se utilizarán dos estructuras de datos: La primera colección almacenará el monto (\$) total de ventas de cada mes (enero a diciembre) y, la segunda colección (estructura de datos paralela a la anterior), almacenará información del Electrodoméstico que obtuvo el mayor monto de venta(\$) en el mes.

Descripción de las Estructuras de datos:

Para el desarrollo debe utilizar las siguientes estructuras:

a) Una estructura de datos donde cada elemento de la misma contiene el monto total de ventas de productos de cada mes. De esta manera:

ventas[0] se encuentra el monto total de venta de productos del mes **enero**

ventas [1] se encuentra el monto total de venta de productos del mes **febrero**

ventas [2] se encuentra el monto total de venta de productos del mes **marzo**

...

ventas [11] se encuentra el monto total de venta de productos del mes **diciembre**

b) Una estructura de arreglos asociativos que sólo almacena información del Producto con mayor monto de venta del mes (el monto de venta se obtiene como 'precioProducto * cantProductos'). Cada arreglo asociativo tendrá el siguiente formato: (**"prod"=> 'nombreProducto' , "precioProd" => elPrecioUnitario, "cantProd"=> cantProdVendidos**)

Por ejemplo:

prodMasVendido[0] = (**"prod"=> 'Heladera Gama' , "precioProd"=>25600, "cantProd" =>3**) es la información del producto con mayor monto de venta del mes enero

prodMasVendido[1] = (**"prod"=> 'Microondas Samsung' , " precioProd " =>5300.50, " cantProd" =>20**) es la información del producto con mayor monto de venta del mes febrero

...

prodMasVendido[11] = (**"prod"=> 'TV Smart Samsung 49' , "precioProd " =>58500, " cantProd" =>10**) es la información del producto con mayor monto de venta del mes diciembre

Implementación del Menú de Usuario:

El menú de opciones es el siguiente:

- 1) Ingresar una venta
- 2) Mes con mayor monto de ventas
- 3) Primer mes que supera un monto de ventas
- 4) Información de un mes
- 5) Productos más vendidos Ordenados

Explicación de la funcionalidad:

1 - Ingresar una venta de un producto en un mes determinado (Mes, producto, precio Unitario y cantidad de productos vendidos). Este menú debe realizar lo siguiente:

- a. solicitar los datos al usuario de la venta mensual del producto
- b. incrementar el monto total de las ventas de productos en el mes ingresado
- c. actualizar la información en la estructura **prodMasVendido** sólo si la venta ingresada tiene mayor monto de venta que el producto ya existente en ese mes.

2 - Mostrar la información completa del mes con mayor monto de ventas.

3 - Cuál es el primer mes que supera un monto total de venta ingresado por un usuario.

4- Imprimir la información completa de un mes (el mes es elegido por el usuario).

Ejemplo:

<diciembre>

El producto con mayor monto de venta: TV Smart Samsung 49

Cantidad de Productos Vendidos: 10

Precio Unitario: \$58500

Monto de venta del producto: \$585000

Monto acumulado de ventas del mes diciembre: \$4115468

5- Utilizar el método de ordenamiento que sea más adecuado para ordenar el arreglo de **prodMasVendidos** de mayor a menor por monto de venta. 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)

Requirimientos obligatorios para la implementación:

- a) Las estructuras de datos deben ser precargadas con datos, de manera que el programa ya contenga datos cuando sea ejecutado:
 - i. La estructura "prodMasVendido" debe ser creada e inicializada utilizando una función de precarga que retorne la estructura con datos.
 - ii. La estructura "ventas" debe ser creada en otra función que reciba por parámetro la estructura "prodMasVendido" y almacene en cada mes "precioProd * cantProd".
 - iii. **las funciones de a.i y a.ii serán invocadas en el programa principal.**
 - iv. las estructuras de datos deberán ser pasadas por parámetro a las funciones implementadas para resolver cada menú de opciones.
- b) Para visualizar el menú de opciones (que siempre es el mismo), implementar una función **menuOpciones** que muestre las opciones 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".
- c) 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.
- d) Utilizar funciones de String predefinida por PHP (strtolower, strtoupper, etc.).
- e) Crear una función solicitar mes que solicite el mes al usuario y retorne un mes válido.
- f) Crear una función que dado un número entre 0 y 11 devuelva el nombre del mes que le corresponde. Y otra función que dado un string que representa el mes devuelva el número que le corresponde. Utilizarla convenientemente.
- g) Crear una función solicitarMes que pida al usuario un nombre de mes válido y retorne el índice que le corresponde.
- h) En los recorridos de las estructuras de datos no "hardcode" la cantidad de elementos a recorrer.

GITHUB:

Práctica Básica de git/github

Las siguientes pautas son introductorias para comenzar a utilizar git/github.

Como el objetivo es que comiencen a conocer la herramienta, los alumnos que recién están comenzando, cuando terminen el desarrollo del tp final deben seguir los siguientes pasos:

- 1- Debe **crear un usuario en github.com** (El usuario se utilizará en próximas materias de la Tecnicatura, elija un nombre de usuario que lo identifique, usar nombre y apellido).
- 2- Debe **instalar git-scm.com** en su pc, y luego con git bash (o la terminal del IDE VSCode) configurar git con el mail y nombre de usuario utilizado para crear la cuenta github:
 - > git config --global user.email "xyz@est.fi.uncoma.edu.ar"
 - > git config --global user.name "nickusuario"
- 3- Debe crear un **repositorio Privado** con su cuenta, con un archivo Readme. El nombre del repositorio debe ser **TPProductos**.
- 4- Agregar como colaborador al profesor (**Nick: majorotter**). (En la página del repositorio github, settings/configuración, Manage access/Administrar Acceso, Invite a collaborator).

5- En vscode, abrir la terminal. En caso de no contar con vscode, abrir git bash.

En la terminal situarse en una carpeta/directorio donde trabaje habitualmente (ejemplo: `cd ../misPhp`)

Ejecutar el comando:

> `git cloneurl del repo de github.....` // La url copiarla del repositorio en github

Al ejecutar el clone, se debería haber clonado el repositorio **TPProductos** con el archivo `readme.md`

6- Agregar el archivo php con el que trabajó. Modificar el archivo `readme.md` (El `readme`, debe modificarlo indicando materia, año, una descripción del trabajo)

7- Luego en `git.bash` o en un terminal ejecutar los siguientes comandos:

a) `git status` //verá los archivos que trackea git y que fueron modificados, y los archivos que aún no son trackeados por git.

b) `git add README.md` // para aceptar los cambios en el `readme`

c) `git status` // ver los cambios

d) `git add .` // el punto sirve para agregar todos los cambios y archivos

e) `git commit -m "Readme modificado"` //confirmamos los cambios

f) `git status` //debe indicar que no hay nada para realizar commit.

g) `git log` //Ver el historial de commits. Debería ver el commit inicial y el realizado

h) `git remote -v` //ver el origen del proyecto

i) `git push` //enviar y actualizar la rama main al origen

opcional: instalar en vscode las extensiones: Git, Git Graph

opcional: instalar Gitkraken (otras extensiones que permitan el uso de git)