

Materia: PROGRAMACIÓN III

Apellido:					Fecha:					
Nombre:					Docente <sup>(2)</sup> :	Friadenrich / Bongioanni				
División:					Nota <sup>(2)</sup> :					
Legajo:					Firma <sup>(2)</sup> :					
Instancia <sup>(1)</sup> :	PP	x	RPP		SP		RSP		FIN	

Se debe realizar una aplicación para dar de ingreso con **imagen del ítem**.

Se deben respetar los nombres de los archivos y de las clases.

Se debe crear **una clase en PHP** por cada entidad y los archivos PHP solo deben llamar a métodos de las clases.

1era parte

1-

**A- (1 pt.) index.php:** Recibe todas las peticiones que realiza el postman y administra a qué archivo se debe incluir.

**B- (1 pt.) TiendaAlta.php:** (por POST) se ingresa Nombre, Precio, Tipo ("Smartphone" o "Tablet"), Marca, Stock (unidades).

Se guardan los datos en el archivo de texto **tienda.json**, tomando un **id autoincremental** como identificador (emulado). Si el nombre y tipo ya existen, se actualiza el precio y se suma al stock existente.

Completar el alta con imagen del producto, guardando la imagen con el nombre y tipo como identificación en la carpeta **/ImágenesDeProductos/2024**.

2-

(1pt.) **ProductoConsultar.php:** (por POST) Se ingresa Nombre, Tipo y Marca, si coincide con algún registro del archivo **tienda.json**, retornar **"existe"**. De lo contrario, informar si no existe ningún producto con ese tipo o ese nombre.

3-

**A- (1 pt.) AltaVenta.php:** (por POST) se recibe el email del usuario y el Nombre, Tipo, Marca y Stock, si el ítem existe en **tienda.json** y hay stock, guardar en **ventas.json** (los datos ingresados junto con la fecha, número de pedido, id autoincremental y precio total de la venta). **Se debe descontar la cantidad vendida del stock.**

**B- (1 pt)** Completar el alta de la venta con imagen de la venta (ej: una imagen del usuario), guardando la imagen con el nombre+tipo+marca+email(solo usuario hasta el @) y fecha de la venta en la carpeta **/ImágenesDeVenta/2024**.

## 2da parte

### 4- (2 pts.) **ConsultasVentas.php**: (por GET)

Datos a consultar:

- a- La cantidad de productos vendidos en un día en particular (se envía por parámetro), si no se pasa fecha, se muestran los del día de ayer.
- b- El listado de **ventas** de un usuario ingresado (por mail).
- c- El listado de **ventas** por tipo de producto.
- d- El listado de **ventas** cuyo precio total esté entre dos números ingresados.
- e- El listado de cantidad de dinero / **ganancia por día** de una **fecha ingresada**. Si no se ingresa una fecha, se muestran las ganancias totales.
- f- **Mostrar el producto más vendido.**

### 5- (1 pts.) **ModificarVenta.php** (por PUT)

Debe recibir el número de pedido, el email del usuario, el nombre, tipo, marca y cantidad. Si existe (por numero de pedido) se modifican el resto de los datos, de lo contrario, informar que no existe ese número de pedido.

## 3ra parte

### 6- (2 pts.) **AltaConjuntos.php** (por POST)

Debe permitir ingresar los detalles del smartphone (nombre, tipo), los detalles de la tablet (nombre, tipo) y una imagen del conjunto.

Ambos productos deben existir como productos simples en el archivo **tienda.json**.

Una vez revisado que los productos existan, el sistema debe combinarlos para formar un objeto conjunto (**nombre\_smartphone, nombre\_tablet, marca\_smartphone, marca\_tablet, precio\_total, tipo="conjunto"**) y almacenarlo como un único registro en el archivo **tienda.json**. Este nuevo elemento **NO debe modificar los registros existentes, funciona como otra opción de compra.**

Se debe poder diferenciar un producto simple de un conjunto dentro de la lógica de la aplicación a través del campo **tipo** que comparten ambas entidades.

Completar el alta con **imagen del conjunto**, guardando la imagen con nombre del smartphone + nombre de la tablet como identificación en la carpeta **/ImágenesDeConjuntos/2024**.

Ej. de tienda.json

```
[
  {id: 10, tipo: "smartphone", nombre: "iPhone 12", stock: 10, precio: 800, ...}, // prod. simple
  {id: 11, tipo: "tablet", nombre: "iPad Pro", stock: 10, precio: 1000...}, //prod. simple
  {id: 12, tipo: "conjunto", nombre_smartphone: "iPhone 12", nombre_tablet: "iPad Pro",
    precio_total: 1620, ...}, //conjunto
  ...
]
```

7- (1 pts.) **BorrarVenta.php** (por DELETE), debe recibir un número de pedido, se borra la venta (soft-delete, no físicamente) y la foto relacionada a esa venta debe moverse a la carpeta **/ImágenesBackupVentas/2024**.

**Código obsoleto, copiado y pegado que no tenga utilidad (-1 punto).**

***Se pueden bajar plantillas de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.***

***Se tiene que sumar 7 (siete) puntos para lograr un cuatro (4).***

***Ocho puntos equivalen a un cinco (5),***

***Nueve puntos equivalen a seis (6),***

***Diez puntos: ocho (8),***

***Y si se suman los once puntos la nota será de diez (10).***