

Materia: PROGRAMACIÓN III

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

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

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.

Se deben guardar todos los datos en MYSQL

1era parte (10pts)

1-

A- (3 pt.) index.php: Recibe todas las peticiones que realiza el Postman y posee todas las rutas de la aplicación a través de SLIM.

B- (1 pt.) ruta: “/tienda/alta”: (por POST) se ingresa Nombre, Precio, Tipo (“Smartphone” o “Tablet”), Marca, Stock (unidades). Se guardan los datos de la **tabla** de la base de datos **MySQL tienda**, tomando un **id autoincremental** como identificador. Si la **marca 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-

(1 pt.) ruta: “/tienda/consultar”:

 (por POST) Se ingresa Nombre, Tipo y Marca. Si coincide con algún registro de la tabla de la base de datos MySQL **tienda**, retornar **"existe"**. De lo contrario, informar si no existe el **tipo** o la **marca**. (No hay productos de la marca x / no hay productos del tipo x).

3-

A- (1 pt.) ruta: “/ventas/alta”: (por POST) se recibe el email del usuario y el Nombre, Tipo, Marca y Stock, si el ítem existe en la **tabla tienda** y **hay stock**, guardar en la **tabla ventas** (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**. **EN LA BASE DE DATOS SE GUARDA SOLO LA RUTA DE LA IMAGEN**.

4-

(2 pts.) ruta principal: **“/ventas/consultar”** (por GET)

Datos a consultar:

A- ruta: **“/productos/vendidos”** 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- ruta: **“/ventas/porUsuario”** El listado de **ventas** de un usuario ingresado (por mail).

C- ruta: **“/ventas/porProducto”** El listado de **ventas** por tipo de producto.

D- ruta: **“/productos/entreValores”** El listado de **ventas** cuyo precio total esté entre dos números ingresados.

E- ruta: **“/ventas/ingresos”** 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- ruta: **“/productos/masVendido”** **Mostrar el producto más vendido.**

5- (1 pt.) ruta: **“/ventas/modificar”** (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.

2da parte (4pts)

6-

A- (1pts) Crear en la base de datos la tabla usuarios con los datos **id, mail, usuario, contraseña, perfil (cliente, empleado, admin), foto, fecha_de_alta, fecha_de_baja**.

C- (1pts) ruta **“/registro”** (por POST)

Recibe los datos de mail, usuario, contraseña, perfil y foto de un usuario y lo agrega a la tabla **usuarios**.

La imagen se guarda en la carpeta **“ImagenesDeUsuarios/2024/”** con el nombre del usuario + perfil + fecha.

B- (2pts) ruta: **“/login”** (por POST)

Se envían los datos usuario y contraseña de un usuario. Se realiza el login y se devuelve un token **JWT que verifique a ese usuario junto a su perfil**.

7-

A-(2pts) Crear la clase **ConfirmarPerfil** que en su método **__invoke** tendrá un **middleware** que toma el **JWT del header** de la petición y confirma que el **perfil** del token **sea el correcto**. La clase recibe el **perfil o perfiles** va a confirmar por su método **__construct**.

B- (1pts) Las rutas:

/tienda/alta

/tienda/consultar/ventas/ingresos

/ventas/modificar

deben estar limitadas para solo ser accedidas por usuarios **“admin”**

Las rutas:

/tienda/consultar/productos/vendidos

/tienda/consultar/ventas/porUsuario

/tienda/consultar/ventas/porProducto

/tienda/consultar/productos/entreValores

/tienda/consultar/productos/masVendido

/ventas/alta

/ventas/consultar

deben estar limitadas para solo ser accedidas por usuarios **“admin”** y usuarios **“empleado”**

C- (1pts) Crear uno o varios middlewares para las rutas de **/tienda/consultar** que revisen que los datos necesarios para realizar las consultas estén presentes.

4ta parte (3pts)

8- (3pts) ruta: “/ventas/descargar” (por GET, solo admin).

Descargar (NO guardar) un CSV del listado de ventas.

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

Se puede 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 13 (trece) puntos para lograr un cuatro (4).

Quince (15) puntos equivalen a un cinco (5)

Diecisiete (17) puntos equivalen a un seis (6),

Dieciocho (18) puntos equivalen a un (7),

Diecinueve (19): ocho (8),

Veinte (20): nueve (9)

Y si se suman los veintiún (21) puntos la nota será de diez (10).