

## Parte 7 - (MySQL)

### TP 01 SQL (ABM con listado)

#### Tabla: usuario

(id autoincremental ,nombre,apellido, clave,mail,fecha\_de\_registro )

#### Tabla:producto

(id autoincremental,código\_de\_barra (6 cifras ),nombre ,tipo, stock, precio,fecha\_de\_creación,fecha\_de\_modificación )

#### Tabla:venta

(id autoincremental,id\_producto ,id\_usuario, cantidad,fecha\_de\_venta, )

Hacer todos los script necesarios y entregarlos por GDB

NOTA:insertar los siguientes datos en sus consultas SQL correspondientes.

Productos:

id	código_d e_barra	nombre	tipo	stock	precio	fecha_de_creación	fecha_de_modificación
1001	77900361	Westmacot t	liquido	33	\$15.87	2/9/2021	9/26/2020
1002	77900362	Spirit	solido	45	\$69.74	9/18/2020	4/14/2020
1003	77900363	Newgrosh	polvo	14	\$68.19	11/29/2020	2/11/2021
1004	77900364	McNickle	polvo	19	\$53.51	11/28/2020	4/17/2020
1005	77900365	Hudd	solido	68	\$26.56	12/19/2020	6/19/2020
1006	77900366	Schrader	polvo	17	\$96.54	8/2/2020	4/18/2020
1007	77900367	Bachellier	solido	59	\$69.17	1/30/2021	6/7/2020
1008	77900368	Fleming	solido	38	\$66.77	10/26/2020	10/3/2020
1009	77900369	Hurry	solido	44	\$43.01	7/4/2020	5/30/2020
1010	77900310	Krauss	polvo	73	\$35.73	3/3/2021	8/30/2020

Usuarios:

nombre	apellido	clave	mail	fecha_de_registro	localidad
Esteban	Madou	2345	dkantor0@example.com	1/7/2021	Quilmes
German	Gerram	1234	ggerram1@hud.gov	5/8/2020	Berazategui
Deloris	Fosis	5678	bsharpe2@wisc.edu	11/28/2020	Avellaneda
Brok	Neiner	4567	bblazic3@desdev.cn	12/8/2020	Quilmes
Garrick	Brent	6789	gbrent4@theguardian.com	12/17/2020	Moron
Bili	Baus	0123	bhoff5@addthis.com	11/27/2020	Moreno

Ventas:

id_producto	id_usuario	cantidad	fecha_de_venta
1001	101	2	7/19/2020
1008	102	3	8/16/2020
1007	102	4	1/24/2021
1006	103	5	1/14/2021
1003	104	6	3/20/2021
1005	105	7	2/22/2021
1003	104	6	12/2/2020
1003	106	6	6/10/2020
1002	106	6	2/4/2021
1001	106	1	5/17/2020

3.- Realizar las siguientes consultas.

1. Obtener los detalles completos de todos los usuarios, ordenados alfabéticamente.
2. Obtener los detalles completos de todos los productos líquidos.
3. Obtener todas las compras en los cuales la cantidad esté entre 6 y 10 inclusive.
4. Obtener la cantidad total de todos los productos vendidos.
5. Mostrar los primeros 3 números de productos que se han enviado.
6. Mostrar los nombres del usuario y los nombres de los productos de cada venta.
7. Indicar el monto (cantidad \* precio) por cada una de las ventas.
8. Obtener la cantidad total del producto 1003 vendido por el usuario 104.
9. Obtener todos los números de los productos vendidos por algún usuario de

'Avellaneda'.

10. Obtener los datos completos de los usuarios cuyos nombres contengan la letra 'u'.
11. Traer las ventas entre junio del 2020 y febrero 2021.
12. Obtener los usuarios registrados antes del 2021.
13. Agregar el producto llamado 'Chocolate', de tipo Sólido y con un precio de 25,35.
14. Insertar un nuevo usuario .
15. Cambiar los precios de los productos de tipo sólido a 66,60.
16. Cambiar el stock a 0 de todos los productos cuyas cantidades de stock sean menores a 20 inclusive.
17. Eliminar el producto número 1010.
18. Eliminar a todos los usuarios que no han vendido productos.

## Parte 9 - Ejercicios PDO

### Aplicación N° 27 (Registro BD)

**Archivo:** registro.php

**método:**POST

Recibe los datos del usuario( nombre,apellido, clave,mail,localidad )por POST , crear un objeto con la fecha de registro y utilizar sus métodos para poder hacer el alta, guardando los datos la base de datos  
retorna si se pudo agregar o no.

### Aplicación N° 28 ( Listado BD)

**Archivo:** listado.php

**método:**GET

Recibe qué listado va a retornar(ej:usuarios,productos,ventas)  
cada objeto o clase tendrán los métodos para responder a la petición devolviendo un listado en JSON

### Aplicación N° 29( Login con bd)

**Archivo:** Login.php

**método:**POST

Recibe los datos del usuario(clave,mail )por POST ,  
crear un objeto y utilizar sus métodos para poder verificar si es un usuario registrado en la base de datos,

Retorna un :

"Verificado" si el usuario existe y coincide la clave también.

"Error en los datos" si esta mal la clave.

"Usuario no registrado si no coincide el mail"

Hacer los métodos necesarios en la clase usuario.

### Aplicación N° 30 ( AltaProducto BD)

**Archivo:** altaProducto.php

**método:**POST

Recibe los datos del producto(código de barra (6 cifras ),nombre ,tipo, stock, precio )por POST , carga la fecha de creación y crear un objeto ,se debe utilizar sus métodos para poder verificar si es un producto existente,

si ya existe el producto se le suma el stock , de lo contrario se agrega .

Retorna un :

"Ingresado" si es un producto nuevo

"Actualizado" si ya existía y se actualiza el stock.

"no se pudo hacer"si no se pudo hacer

Hacer los métodos necesarios en la clase

### **Aplicación N° 31 (RealizarVenta BD )**

**Archivo: RealizarVenta.php**

**método:POST**

Recibe los datos del producto(código de barra), del usuario (el id )y la cantidad de ítems ,por POST .

Verificar que el usuario y el producto exista y tenga stock.

Retorna un :

"venta realizada"Se hizo una venta

"no se pudo hacer"si no se pudo hacer

Hacer los métodos necesarios en las clases

### **Aplicación N° 32(Modificacion BD)**

**Archivo: ModificacionUsuario.php**

**método:POST**

Recibe los datos del usuario(nombre, clavenueva, clavevieja,mail )por POST , crear un objeto y utilizar sus métodos para poder hacer la modificación,

guardando los datos la base de datos

retorna si se pudo agregar o no.

**Solo pueden cambiar la clave**

### **Aplicación N° 33 ( ModificacionProducto BD)**

**Archivo: modificacionproducto.php**

**método:POST**

Recibe los datos del producto(código de barra (6 cifras ),nombre ,tipo, stock, precio )por POST ,

crear un objeto y utilizar sus métodos para poder verificar si es un producto existente, si ya existe el producto el stock se sobrescribe y se cambian todos los datos excepto: el código de barras .

Retorna un :

"Actualizado" si ya existía y se actualiza

"no se pudo hacer"si no se pudo hacer

Hacer los métodos necesarios en la clase

## Parte 10- Ejercicios PDO listados

Funciones de filtrado:

**se deben realizar la funciones que reciban datos por parámetros y puedan ejecutar la consulta para responder a los siguientes requerimientos**

- A. Obtener los detalles completos de todos los usuarios y poder ordenarlos alfabéticamente de forma ascendente o descendente.
- B. Obtener los detalles completos de todos los productos y poder ordenarlos alfabéticamente de forma ascendente y descendente.
- C. Obtener todas las compras filtradas entre dos cantidades.
- D. Obtener la cantidad total de todos los productos vendidos entre dos fechas. E. Mostrar los primeros "N" números de productos que se han enviado. F. Mostrar los nombres del usuario y los nombres de los productos de cada venta. G. Indicar el monto (cantidad \* precio) por cada una de las ventas.
- H. Obtener la cantidad total de un producto (ejemplo:1003) vendido por un usuario (ejemplo: 104).
- I. Obtener todos los números de los productos vendidos por algún usuario filtrado por localidad (ejemplo: 'Avellaneda').
- J. Obtener los datos completos de los usuarios filtrando por letras en su nombre o apellido.
- K. Mostrar las ventas entre dos fechas del año.

**Nota: Organice el código de estas consultas de la manera que quiera , pero debe tener las funcionalidades en clases y las llamadas desde un archivo .php**

Respuestas:

las respuestas de GDB deberían tener una primer hoja con índice indicando el archivo y el renglón que resuelve la consigna ej:

a :{ archivo :usuarios.php, renglón :66}

b:{ archivo :productos.php, renglón :666}

...etc.