

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 o tabla de html <table>

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.