

## Actividad 7.2: APLICACIÓN PARA REPRESENTANTES.

### CONSIDERACIONES PREVIAS

- En todo el proyecto será obligatorio respetar la estructura del **MODELO VISTA CONTROLADOR**.
- Todas las URL's utilizadas serán con un formato amigable.
- Para acceder a las bases de datos se utilizará PDO, excepciones, y consultas preparadas.

### SERVICIOS

Se implementarán dos servicios uno para CLIENTES y el otro para PRODUCTOS, cada servicio lo pondremos en una carpeta independiente dentro de la carpeta del ejercicio, ambos tendrán las mismas características.

- Solo tendrán implementadas las solicitudes por el método Get que servirán para devolver la información que se solicite.
- La devolución será en formato JSON e incluirá todos los campos de la tabla en cuestión.
- Hay que utilizar un modelo como el que se describe en este enunciado **para cada servicio**.
- Los dos servicios devolverán un registro concreto si se especifica un Id (dniCliente o IdProductos) y todos los registros si no se especifica ninguno.

Llamaremos al servicio como si fuera una API (utilizando `file_get_contents`) y el resultado se decodificará con `json_decode`.

#### **json\_decode**

Decodifica una variable en formato JSON de dos formas distintas

```
$ver=json_decode($result,true);
```

`$ver` será un array asociativo de forma que si `$result` solo lleva un registro será de una dimensión pero si es un grupo de registros será un array asociativo de dos dimensiones.

```
$ver=json_decode($result);
```

`$ver` tendrá objetos de forma que si `$result` solo lleva un registro será un objeto pero si es un grupo de registros será un array de objetos.

### MODELO

Crearemos un modelo **para la aplicación principal y otro para cada servicio**.

El diseño de este modelo es libre, pero ha de cumplir los siguientes requisitos:

- Las propiedades han de ser **privadas**
- Si es necesario se crearan **set** y **get** para acceder a algunas de ellas, utilizando los métodos mágicos.
- Todos aquellos métodos que sean necesarios para relacionarse con la base de datos, recibirán **la conexión como parámetro** y si necesitan algún valor para un campo **se obtendrán de las propiedades** que antes se habrán cargado con ellos. Nunca estos datos se pasarán como parámetros.
- Todos los métodos que involucren a más de un registro (del tipo `getAll`) serán **estáticos**.

## CONTROLADORES

Serán obligatorios los siguientes controladores.

### 1.- index.php.

Se mostrará una pantalla con un formulario con los datos principales del PEDIDO:

idPedido: Se obtendrá sumando uno al número máximo que se haya introducido. No se podrá modificar en este formulario

Fecha: La fecha de hoy y si que se podrá modificar.

DniCliente: El dniCliente se rellenará con una lista desplegable que obtendrá los valores con una petición al servicio CLIENTE, para recuperar todos los clientes.

Una vez pulsado el botón del formulario, y en el mismo archivo se procesará el formulario:

- Se almacenan los tres campos en sendas variables de sesión.
- Se crea una variable de sesión llamada **totalLineas** que almacenará el número de líneas introducidas y que inicialmente valdrá 0.
- A continuación se salta al siguiente controlador (lineas.php).

### 3.- lineas.php.

Pedido	Nlinea	producto	cantidad
10	1	1	5
10	2	3	10
10	3	3	45
10	4	2	8
10	5	<input type="text" value="Macarrones"/>	<input type="text"/>

[terminar](#)

Este controlador servirá para rellenar las líneas de pedidos. (las iremos almacenando en variables de sesión que se muestran a continuación)

En pantalla aparecerán todas las líneas de este mismo pedido que ya se hayan introducido y al final un formulario con las siguientes características:

- Se rellenará solo los campos de *producto* y *cantidad*
- Los productos se rellenarán con una lista desplegable que obtendrá los valores usando el servicio PRODUCTOS
- En el botón del formulario debe de poner "continuar"

Si no se hubiese rellenado ninguna línea se mostrará solo el formulario con el número de pedido y el número de línea 1.

Después del formulario aparecerá un vínculo con el texto "terminar" para ir al tercer controlador (terminar.php).

Cuando se pulsa el botón "continuar" del formulario, se almacenan los datos introducidos en variables de sesión:

- En primer lugar se incrementará la variable de sesión totalLineas
- Usaremos un array de variables de sesión, para las nlineas, idProductos, y cantidad.

Se tienen que seguir introduciendo líneas hasta que se pulse el vínculo terminar.

### 4.- terminar.php.

Este controlador es el encargado de terminar todo el proceso, es decir, insertar las variables de sesión en la base de datos y destruirlas. Mostrando al final un mensaje con todo lo que se ha introducido (será una variable string, que se va componiendo mientras se almacena el pedido). Es importante recordar que todo el controlador debe de ser una única transacción.

En la tabla lineaspedidos.

- Alta en pedidos

idpedido: Calcularemos el máximo de los valores que tiene en la tabla pedidos el campo idPedido y le sumaremos uno.

fecha: La de hoy.

Dnicliente : El dni de la variable de sesión

Alta en lineaspedidos:

Idpedido es el que hemos calculado antes,

nlinea es un contador desde 1 hasta el número de productos a añadir para ese pedido.

El resto de campos los sacamos del carrito.

En esta tabla tendremos que dar de alta tantos registros como productos haya en el carrito.

Se creará un string que contendrá las sentencias en html necesarias para visualizar el pedido completo, y al final se mostrará por pantalla.

El pedido completo debe de contener:

- Todos los datos del cliente. Se obtendrán llamando al servicio con el dni del clientes
- Las líneas del pedido. Para cada línea hay que incluir el nombre y el precio del producto, obtenidos llamando al servicio de PRODUCTOS con su id. Con el precio y la cantidad se tiene que mostrar el importe.
- Al final hay que mostrar también la suma de importes.

Se borrarán todas las cookies y todos los registros del carrito.

#### **FUNCION:**

Tendremos un fichero de funciones (funciones.php) con una sola función que será para crear los select de los formularios como ya hemos hecho otras veces. En este caso estos SELECT obtendrán los datos de los servicios por lo tanto cambia la estructura de la función de la siguiente manera:

```
function lista ($url,$tabla, $nomid, $mostrar)
```

-**\$url**: Es la dirección donde se encuentra el servicio.

- **\$tabla**: Se utilizará com nombre para el SELECT

- **\$momid** y **\$mostrar**: Son los nombres de los campos que se reciben en el json, y que son el id y el campo a mostrar en el SELECT

#### **AMPLIACIONES:**

1.- Modificarlo para que tenga autocargas de clases y la estructura vistas en clase.