

PEC 2 – Desarrollo Web en Entorno Servidor

EJERCICIO 1 – PAGINACIÓN

REQUISITOS PREVIOS

- Descarga el archivo *ficheros_pec2_1.zip*. Descomprime su contenido en el directorio htdocs.
- Ejecuta las instrucciones SQL contenidas en el documento *pec2_1.sql*
- Comprueba que se ha creado la base de datos *pec2_1* y que contiene las tablas *categorías*, *pedidos*, *pedidosproductos*, *productos* y *restaurantes*. Además, añadirá 40 productos nuevos a la tabla *productos* y 5 pedidos a los restaurantes.

ENUNCIADO

Cuando accedemos a los productos de la categoría *Comida*, estos se muestran todos seguidos. Vamos a hacer que los productos se muestren de forma paginada mediante enlaces html compuestos de números, como se muestra en la imagen.

Usuario: admin@empresa.com Administración Home Ver carrito Preferencias Cerrar sesión					
<h2>Comida</h2>					
Platos e ingredientes					
1 2 3 4 5					
Nombre	Descripción	Peso Stock		Comprar	
Harina	8 paquetes de 1kg de harina cada uno	8	100	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Azúcar	20 paquetes de 1kg cada uno	20	3	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Naranjas	Dulces y jugosas naranjas recién cosechadas	0.3	80	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Sandía	Sandías grandes y jugosas de verano	5	20	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Melón	Melones dulces y refrescantes	2	30	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Uvas	Uvas rojas y verdes de la región	0.5	70	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Tomates	Tomates frescos y maduros	0.3	90	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Lechuga	Lechugas frescas y crujientes	0.2	120	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Espinacas	Espinacas frescas y saludables	0.25	80	<input type="text" value="1"/>	<input type="button" value="Comprar"/>
Pepino	Pepinos crujientes y saludables	0.25	70	<input type="text" value="1"/>	<input type="button" value="Comprar"/>

Para lograrlo, añade a *bd.php* la función:

```
function productos_en_categoria($codCat){
    $res = leer_config(dirname(__FILE__) . "/configuracion.xml", dirname(__FILE__) .
"/configuracion.xsd");
    $bd = new PDO($res[0], $res[1], $res[2]);
    $sql = "SELECT count(CodProd) FROM productos WHERE CodCat = ?";
    $preparada=$bd->prepare($sql);
    $preparada->execute(array($codCat));
    $datos=$preparada->fetchALL();
    if ($preparada) {
        return $datos[0][0];
    }else{
        return False;
    }
}
```

Esta función recibe el código de una categoría y devuelve el número de productos que existen en dicha categoría.

Con ese número, podemos calcular el número de páginas que tendrá nuestra paginación, es decir, el número de enlaces que hemos de tener. Veamos cómo.

Supongamos que tenemos 42 productos en la categoría de comidas y decidimos mostrar los productos de 10 en 10 ¿Cuántas páginas se deberían mostrar?

$$\frac{42}{10} = 4.2 \xrightarrow{\text{redondeamos hacia arriba}} 5$$

Tendríamos que mostrar 5 páginas, y, por tanto, crear 5 enlaces numéricos, de forma que:

- El enlace 1 muestre los productos del 0 al 9 incluidos
- El enlace 2 muestre los productos del 10 al 19 incluidos
- El enlace 3 muestre los productos del 20 al 29 incluidos
- El enlace 4 muestre los productos del 30 al 39 incluidos
- El enlace 5 muestre los productos del 40 al 49 incluidos

La función `ceil` de PHP nos permite redondear hacia arriba un número decimal.

Por otra parte, sustituye la función `cargar_productos_categoria` de `bd.php` por:

```
function cargar_productos_categoria($codCat, $comienzo=0, $limite=100){
    $res = leer_config(dirname(__FILE__) . "/configuracion.xml", dirname(__FILE__) .
"/configuracion.xsd");
    $bd = new PDO($res[0], $res[1], $res[2]);
    $sql = "SELECT * FROM productos WHERE Codcat = $codCat AND Stock >= " .
leer_cookie('stock', 0). " LIMIT ". $comienzo . ",". $limite;
    $resul = $bd->query($sql);
    if (!$resul) {
        return FALSE;
    }
    if ($resul->rowCount() === 0) {
        return FALSE;
    }
    return $resul;
}
```

Esta función recibe tres valores:

1. **\$cod_cat**: el código de la categoría de la que queremos cargar los productos
2. **\$comienzo**: registro en el que empezamos a buscar los datos.
3. **\$limite**: número máximo de datos que queremos buscar en la consulta SELECT

De esta forma,

```
cargar_productos_categoria($_GET['categoria'], 0, 10);
```

devolverá, empezando por el primer producto, 10 productos en total.

Para obtener los 10 productos siguientes, usaríamos

```
cargar_productos_categoria($_GET['categoria'], 10, 10);
```

Si en la categoría tenemos 18 productos y escribimos

```
cargar_productos_categoria($_GET['categoria'], 20, 10);
```

La función devuelve *Falso* puesto que la consulta se ha podido ejecutar, pero el número de filas encontradas es de 0, ya que el producto número 20 no existe.

Por último, en la línea 28 de `categorias.php`, sustituye la línea 28 por

```
$url = "productos.php?categoria=".$_cat['CodCat']. "&numProducto=0";
```

De esta forma, cada vez que accedamos a la página productos, veremos la categoría que hayamos escogido partiendo desde el primer producto.

Tu trabajo consiste en modificar la página productos.php realizando los cambios necesarios para que el restaurante pueda navegar entre las diferentes páginas de productos mediante los enlaces numéricos, mostrando los productos de 10 en 10.

Por ejemplo, el enlace 3 deberá redirigirnos a la página de los productos de la categoría en la que nos encontramos, mostrándonos 10 productos, comenzando por el número de producto 20.

El único fichero a entregar es productos.php

EJERCICIO 2 – AJAX DESDE EL SERVIDOR

REQUISITOS PREVIOS

- Descarga el archivo *ficheros_pec2_2.zip*. Descomprime su contenido en el directorio htdocs.
- Comprueba que tienes el servidor de apache y la base de datos levantados.
- Ejecuta las instrucciones SQL contenidas en el documento *pec2_2.sql*
- Comprueba que se ha creado la base de datos *pec2_2* y que contiene las tablas *categorías*, *envíos*, *pedidos*, *pedidosproductos*, *productos* y *restaurantes*.

La tabla envíos tiene tres campos:

- ✓ CodEnv: Código del envío.
- ✓ CodPed: Código de pedido
- ✓ FechaEnv: Fecha en la que se ha enviado el pedido.

ENUNCIADO

Accediendo con el usuario admin@empresa.com y clave 1234 serás redirigido a la zona de administración.



Vamos a dotar al administrador de una nueva funcionalidad que le permita encontrar productos

Para ello, accede a **Buscador de productos** en la **Zona de administración (segundo enlace)**. En ella, el administrador podrá escribir una cadena de caracteres y deberá encontrar en la base de datos todos los productos que contengan dicha cadena de caracteres.

Tu trabajo consiste en:

[1] Crear en *bd_pec2_2.php* la función: *buscar_productos()* que de funcionalidad a la página *Buscador de productos*.

La función *buscar_productos* tiene la siguiente cabecera:

```
array: buscar_productos(string $cadena_busqueda)
```

La función devuelve un array cuyas claves y valores son los siguientes:

Clave	Valor
“encontrado”	Número de veces que ha aparecido la cadena
“productos”	Array con los nombres de los productos que contienen dicha cadena

A la hora de ejecutar la consulta preparada, haz uso de la expresión:

```
$preparada->execute(array('%'.$cadena_busqueda.'%'));
```

[2] Crear el documento *buscador_json.php*. Este documento devolverá el resultado de la llamada a la función ***buscar_productos()*** de php a JavaScript en formato JSON.

[3] Escribe la función *buscarProducto(valor)* en *pec2_2.js*. Esta función realizará una petición al servidor mediante **el método GET** de la página *buscador_json.php*.

La función ha de enviar a *buscador_json.php* la cadena de texto introducida en el formulario.

Muestra en un alert el resultado en JSON obtenido de la petición al servidor de *buscador_json.php*.

Entrega los ficheros:

- **bd_pec2_2.php**
- **buscador_json.php**
- **pec2_2.js**