upna

4'25 1/4

Sistemas de Información Web
Junio de 2023

Nombre y Apellidos:	IKER URDIROZ BARRIO	DNI:	73466243J

(Nota para todos los ejercicios)
Servidor base de datos: localhost
Base de datos: universidad
Usuario: admin
Contraseña: Examen

0'25

1.- Subir imágenes y escalar ficheros (3 puntos)
Hacer un formulario que recoja un fichero de tipo imagen jpg y lo almacene en una carpeta llamada "originales".
A la vez que hacemos el proceso de subida de esa imagen la vamos a escalar a un tamaño de 500 píxeles de alto o ancho (según si la imagen sea vertical u horizontal) y la guardamos en la carpeta medianas.
Luego volvemos a hacer lo mismo a un tamaño de 150 pixeles y la guardamos en la carpeta peques.

2.- Desplegables con AJAX (2 puntos)
Tenemos en la base de datos una tabla de **familias** con los campos:
- Idfamilia
- familia.

Otra tabla **subfamilias** con los campos:
- Idsubfamilia
- Subfamilia
- Idfamilia

1

Finalmente una tercera tabla con los campos:
- Idproducto
- Producto
- Precio
- UnidadesEnStock
- Idsubfamilia

Vamos a crear una página html en la que aparezca un desplegable con las familias. Cuando el usuario selecciona una familia a través de AJAX cargaremos un desplegable con las subfamilias asociadas. Y cuando se selecciones una subfamilia, cargaremos por AJAX un desplegable con los productos asociados a dicha subfamilia.

1

3.- Listado con búsqueda (3 puntos)
En la tabla de productos tenemos 10.000 productos. Queremos hacer un html que muestre una tabla con un listado de productos con las siguientes columnas:

upna

ID Producto	Producto	Precio	Unidades en stock

En la parte superior tendremos un cuadro de texto y un botón para poder buscar. Esta búsqueda hará que el listado que tenemos abajo se filtre acorde al campo de Producto.
El listado mostrará sólo los 100 primeros productos. Y luego en la parte inferior tendremos un par de botones para ir a los siguientes 100 o los anteriores 100 productos.
Para ello usaremos en la query de sql la propiedad limit. Esta parte de sql nos permite hacer por ejemplo:

Select * from..... Limit 0, 100 → Nos da los 100 primeros
Select * from ..... limit 100, 100 → Nos da los 100 siguientes

Y así sucesivamente.

4.- Usar una API (2 puntos)
(Nota: Tenemos estas funciones ya implementadas, no hay que programarlas, sólo utilizarlas)
Las funciones son:

Cargarpaginas() → Nos devuelve un json con la variable páginas que contiene el número de páginas para recoger

Cargarpagina(número página) → Nos devuelve un json con el listado de datos de la página solicitada. (Nota: estos datos son los datos de productos que hemos utilizado en el ejercicio 2)

Tenemos que hacer un proceso que lea todas las páginas una a una. Para cada página tiene que parsear el json recibido y meter los datos en la tabla de productos de la base de datos.

SISTEMA DE INFORMACIÓN WEB:

2023

conexion.php ⟶ para los php siguientes

```php
<? php
  function conexion(){
    $servidor = "localhost";
    $bd = "universidad";
    $user = "admin";
    $password = "Examen";

    $con = mysqli_connect($servidor, $user, $password, $bd);

    if(!$con){
      echo "Error en la conexión a la base de datos";
    }
    return $con;
  }
?>
```

② 3 tablas

Familia
idFamilia
Familia

Subfamilia
idSubfamilia
Subfamilia
idFamilia

Producto
id producto
producto
precio
unidadENSTOCK
idsubfamilia

- HTML con 3 desplejables

1. Familias ⟶ cuando se elige familia → 2. Subfamilias asociadas ⟶ cuando se elige subfamilia → 3. Productos asociados a la subfamilia

TODO CON AJAX

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta>
    <htle></htle>
    <scnpt></scnpt>  → libreria AJAX
</head>
<body>
  <h1> Despiegables dinámicos </h1>
  <label for="Familia">Familia: </label>
  <select id="Familia" onchange="cargarSubFamilias();">
      <ophon value="">Seleccione una Familia</ophon>
  </select>
  <br>
  <label for="subfamilia">Subfamilia: </label>
  <slect id="subfamilia" onchange="cargar Productos();">
      <ophon value="">seleccione una subfamilia</ophon>
  </select>
  <br>
  <label for="producto">Producto:</label>
  <select id="producto">
      <ophon value="">Seleccione un producto</ophon>
  </select>
  <script>
  Funchon cargarSubFamilias(){
    var familiaId = $('#Familia').val();
     $.ajax({
        url: 'get-subfamilias.php',
        type: 'GET',
        data: {idFamilia: familiaId},
        success: Funchon(data){
           $('#subFamilia').html('<ophon value="">selecciona una
              subfamilia</ophon>' + data);
        }
     });
  }
  Funchon cargaProductos(){
     var subfamiliaId = $('#subFamilia').val();
     $.ajax({
        url: get_productos.php',
        type: 'GET',
        data: {idsubFamilias: subfamiliaId},
        success: Funchon(data){
           $('#producto').html('<ophon value="">seleccione un producto
              </ophon>' + data);
        }
     });
  }
  $(document).ready(Funchon(){
     $.ajax({
        url: 'get-Familias.php')
        type: 'GET'
        success: Funchon(data){
           $('#Familia').append(data);
```

Notas:

- onchange → valor del campo cambia. cuando el usuario selecciona una opción diferente en los despiegabies.
- for → asociar etiqueta con elemento de formulario.
- function cargarSubFamilias() → Definimos función. NO toma parámetros. se ejecuta cuando el usuario elige una opción en el despiegable ⇒ [onchange]
- $('#Familia').val() → obtenemos el valor seleccionado en el despiegable (idFamilia de la tabla)
- $.ajax → PETICIÓN AJAX AL SERVIDOR
- url → quién procesa la petición (con la Familia se coge subFamilia)
- data → datos enviados al servidor
- success → Función de éxito
- $('#subFamilia').html(...) → Actualizamos contenido HTML del despiegable de subfamilias con los nuevos datos recibidos.
- $(document).ready → cargar los datos familia en el despiegable cuando la página se cargue inicialmente
- success → Agrega datos recibidos al elemento con ID 'Familia'

get-familias.php

```
<? php
   include 'conexion.php';        → Archivo de conexión.
                                   → $con = conexion();
   $sql = "SELECT idFamilia, Familia FROM Familias"; → consulta para obtener todas las familias
   $result = $con → query($sql); → Ej. consulta
   $ophons = " ";  → vacío pq suponemos que si o sí habrá familias
   IF ($result → num-rows > 0) {  → verifica si la consulta devolvió algún resultado
      while($row = $result → fetch-assoc()) {  → obtiene segunda fila del conjunto de resultados. ??
         $ophons . = "<ophon value = '". $row["idFamilia"]. "'>. $row["Familia"]. "<ophon>";
      }  → Iteramos sobre cada fila para construir las opciones del desplegable.
   }else{
      $ophons = "<ophon> No hay familias disponibles </ophon>";
   }
   echo $ophons; → Imprimir opciones para que JS las pueda usar.
   $con → close(); → cerrar conexión
?>
```

get_subfamilias.php

```
<? php
   include 'conexion.php'   → $idFamilia = isset($GET['idFamilia']) ?
                                $_GET['idFamilia'] : '';
   $sql = "SELECT idsubfamiliassubfamilia FROM subfamilias WHERE idfamilia =
         = '$idFamilia'";
   $result = $con → query($sql);               → iniciar desplegable
   $ophons = "<ophon value = ''> Seleccione una subfamilia </ophon>";
   IF ($result → num-rows > 0) {
      while($row = $result → fetch-assoc()) {
         $ophons . = "<ophon value ='" . $row ["idsubfamilia"]. "'>". $row
               ["subfamilia"]. "</ophon>";
      }
   }else{
      $ophons . = "<ophon> No hay subfamilias disponibles </ophon>";
   }
   echo $ophons,
   $con → close(),
?>
```

```php
<?php
    include "conexion.php";

    $idSubFamilia = isset($_GET['idSubFamilia']) ? $_GET['idSubFamilia'] : '';

    $sql = "SELECT idProducto, producto FROM producto WHERE idSubFamilia = '$idSubFamilia'";
    $result = $conn->query($sql);

    $options = "<option value=''> seleccione un producto </option>";
    IF ($result -> num_rows > 0){
        while ($row = $result -> fetch_assoc()) {
            $options .= "<option value='". $row["idproducto"]. "'>". row["producto"].
                "</option>";
        }
    }else{
        $options .= "<option>NO hay productos disponibles</option>";
    }
    echo $options;
    $con -> close();
?>
```

③ Tabla productos ⟶ 10000 productos

HTML ⟶ Muestra tabla con un listado de productos con las sig. columnas:

ID producto	Producto	Precio	U. en stock

Filtrar

* cuadro de texto y botón (para buscar)
* Mostrar los 100 (primeros) productos ⟶ Abajo un par de botones para los siguientes 100 o los anteriores 100

4.1

```php
<? php
include 'conexion.php'

Funchan cargarPaginas(){
{ return 10;

Funchon cargarPagina($numeroPagina){
    return json-encode($productos);
}

Funchon guardarProductosEnBD($productos){
  $con = conexion();
  For each ($productos as $producto){
    $stmt = $con -> prepare ("INSERT INTO Productos
      (ID-Producto, Producto, Precio, Unidades-en-stock) VALUES (?,?,?,?)");
    $stmt -> and-param("isii, $producto[ID-Producto], $producto[Producto]
    $smtmt -> execute();
  }
  $con -> close();
}

$totalPaginas = cargarPaginas();

For( $i = 1; $i <= $totalPaginas ,$i++){
  $jsonProductos = cargarPagina($i);
  $productos = json-decode($jsonProductos ,true);
  guardarProductosEnBD( $productos);
}

echo "Hecho";
?>
```