

## Sistemas de Información Web Mayo de 2022

Nombre y Apellidos:		DNI:	
---------------------	--	------	--

### (Nota para todos los ejercicios)

Servidor base de datos: localhost

Base de datos: universidad

Usuario: admin

Contraseña: Examen

### 1.- Lectura y almacenamiento de datos (4 puntos)

Hemos programado una llamada a una API de comunicaciones. Esta se encuentra en un fichero llamado: "llamadas\_api.php" (Ya está implementada, no hay que implementarla)

En este fichero php tenemos dos funciones que se llaman:

- `numero_paginas()` → Nos devuelve un número con todas las páginas de datos que tenemos
- `carga_datos(numero_pagina)` → Nos devuelve el contenido del número de página que proporcionamos

La carga de datos es bastante larga, por eso el sistema nos divide la información en páginas. El número de páginas es una información que nos devuelve la función: `numero_paginas()`. Cada página nos devuelve 200 registros en formato json que debemos parsear para poder procesarlos.

La función: `carga_datos(numero_pagina)`, nos devuelve un json que contiene la información que debemos de recoger y almacenar en una base de datos.

La estructura del json que recibimos es:

```
{
  "array": [
    {
      "nombre": "Diego",
      "apellido1": "Perez",
      "apellido2": "Randez",
      "poblacion": "Cintruenigo",
      "provincia": "Navarra"
    },
    {
      "nombre": "Carmen",
      "apellido1": "Poveda",
      "apellido2": "Murillo",
      "poblacion": "Zaragoza",
      "provincia": "Aragón"
    },
    {
      "nombre": "Paula",
      "apellido1": "Murillo",
      "apellido2": "Arana",
      "poblacion": "Madrid",
      "provincia": "Madrid"
    },
    .....--> Así hasta los 200 registros
  ]
}
```

En el ejercicio debemos de programar un php que realice todas las llamadas necesarias al api para recoger todos los registros y meterlos en una tabla de la

base de datos. (la tabla de la base de datos ya existe). La estructura de la tabla es:

personas: nombre, apellido1, apellido2, poblacion, provincia

## 2.- Creación de registro de usuarios (3 puntos)

Para una página que tenemos vamos a crear un apartado para que los usuarios se puedan registrar.

Los datos para recoger en el formulario son:

Nombre  
Primer apellido  
Segundo apellido  
Usuario  
Contraseña  
Repetir contraseña  
Fotografía

Debemos crear por un lado el formulario html que se encargará de recoger la información.

Por otro lado, debemos crear el fichero php que procese la información y la almacene en la base de datos.

La tabla en la base de datos ya está creada y tiene esta estructura:

usuarios: nombre, apellido1, apellido2, user, passwd, foto

(Nota: en el campo foto sólo se guarda el nombre del fichero. La imagen se guarda en el directorio: imagenes)

## 3.- Herramienta de búsqueda con AJAX (3 puntos)

En la base de dato tenemos la tabla de usuarios que hemos usado en el ejercicio número 2.

Ahora en una página html vamos a colocar un campo de texto para buscar, un botón con la palabra buscar y una capa (div) donde poner el resultado de la búsqueda.

Cuando el usuario pinche sobre el botón buscar, vamos a generar por AJAX un listado de los productos que, en su nombre, primer apellido o segundo apellido, contienen los datos introducidos. Todo el listado de usuarios se mostrará en el div ordenado por apellido1, apellido2 y nombre.

La información se mostrará en una tabla. Y en dicha tabla aparecerá en una columna la imagen que se ha cargado en el ejercicio anterior.

## Ejercicio1 SIW 2022

```
<?php
require_once 'llamadas_api.php';

// Database connection parameters
$servername = "localhost";
$username = "admin";
$password = "Examen";
$dbname = "universidad";

//Creamos la nueva conexión a la BD
$conn = new mysqli($servername, $username, $password, $dbname);

//Comprobamos la conexión(No se si es necesario en el examen, yo la pondría por si acaso.
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

//Esto lo que hace es preparar la inserción a la base de datos, mediante "prepare", como aun no sabemos los valores que vamos a introducir ponemos "?"".
$stmt = $conn->prepare("INSERT INTO personas (nombre, apellido1, apellido2, poblacion, provincia) VALUES (?, ?, ?, ?, ?)");

//Verificamos si la preparación ha sido correcta
if (!$stmt) {
    die("Prepare failed: " . $conn->error);
}

//Aquí definimos las variables y su tipo de los valores que van a ser insertados, en este caso 5 s ya que s->String, i->Int, d->Float.
$stmt->bind_param("sssss", $nombre, $apellido1, $apellido2, $poblacion, $provincia);

try {
    //Llamamos a la función que nos ha dado el profesor para obtener en la variable totalPages el número de páginas.
    $totalPages = numero_paginas();

    //Bucle para poder operar con todas las páginas.
    for ($i = 1; $i <= $totalPages; $i++) {
        // Llamamos a la función que nos ha dado el profe, la cual nos devuelve una respuesta json de el número de la página que le hemos dado
        $jsonData = carga_datos($i);

        // Decodeamos la respuesta en la variable data
        $data = json_decode($jsonData, true);
    }
}
```

```

//Verificar si el decode ha sido correcto.
if ($data === null) {
    throw new Exception('Error decoding JSON on page ' . $i);
}

//Recorremos el array de data(recordar que la primera posición
del array será todo esto->

{ "nombre": "Diego", "apellido1": "Perez", "apellido2": "Randez",
  "poblacion": "Cintruenigo", "provincia": "Navarra" }

)
//Almacenamos cada valor de la posición del array en el que
estamos operando en las variables declaradas anteriormente.
foreach ($data['array'] as $person) {
    $nombre = $person['nombre'];
    $apellido1 = $person['apellido1'];
    $apellido2 = $person['apellido2'];
    $poblacion = $person['poblacion'];
    $provincia = $person['provincia'];

    // Ejecutamos la inserción y comprobamos que haya sido
    correcta
    if (!$stmt->execute()) {
        throw new Exception("Execute failed: (" . $stmt->errno
        . ") " . $stmt->error);
    }
}

}

//Si os acordáis y/o quereis, poneis esto.
} catch (Exception $e) {
    echo 'Error: ' . $e->getMessage();
}

//Cerramos la inserción y la conexión a la BD
$stmt->close();
$conn->close();
?>

```

## EJERCICIO 2 SIW

```

<html>
<head>
    <title>Registro de Usuarios</title>
</head>
<body>
    <h2>Registro de Usuarios</h2>

```

```

<form action="procesar_registro.php" method="POST"
enctype="multipart/form-data">
    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre" name="nombre"
required><br>

    <label for="apellido1">Primer Apellido:</label><br>
    <input type="text" id="apellido1" name="apellido1"
required><br>

    <label for="apellido2">Segundo Apellido:</label><br>
    <input type="text" id="apellido2" name="apellido2"
required><br>

    <label for="usuario">Usuario:</label><br>
    <input type="text" id="usuario" name="usuario"
required><br>

    <label for="password">Contraseña:</label><br>
    <input type="password" id="password" name="password"
required><br>

    <label for="confirm_password">Repetir
Contraseña:</label><br>
    <input type="password" id="confirm_password"
name="confirm_password" required><br>

    <label for="foto">Fotografía:</label><br>
    <input type="file" id="foto" name="foto" accept="image/*"
required><br>

    <input type="submit" value="Registrarse">
</form>
</body>
</html>

<?php
// Verificamos si se ha enviado el formulario
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Verificar si las contraseñas coinciden
    if ($_POST['password'] !== $_POST['confirm_password']) {
        die("Las contraseñas no coinciden.");
    }

    // Recogida de datos del formulario
    $nombre = $_POST['nombre'];
    $apellido1 = $_POST['apellido1'];
    $apellido2 = $_POST['apellido2'];
    $usuario = $_POST['usuario'];
    $password = $_POST['password'];

```

```

        // Procesar la fotografía, hacemos un $_FILES['foto']['name']
        para obtener el nombre del archivo subido en el apartado de la
        subida de archivos con nombre 'foto'.
        $foto = $_FILES['foto']['name'];
        //Obtenemos la ruta del archivo (cuando se sube un archivo este
        se almacena en una carpeta temporal del servidor)
        $foto_temporal = $_FILES['foto']['tmp_name'];
        //Ahora declaramos cual queremos que sea la ruta donde se va a
        guardar, como en el enunciado nos dicen que existe un directorio
        con el nombre imagenes, concatenamos un string con el nombre del
        directorio y el nombre del archivo.
        $ruta_foto = 'imagenes/' . $foto;
        //Ahora mediante la funcion move_uploaded_file movemos el archivo
        desde la ruta temporal a la ruta donde queremos almacenar el
        archivo
        move_uploaded_file($foto_temporal, $ruta_foto);

        // Conexión a la base de datos(poner los datos del enunciado)
        $servername = "localhost";
        $username = "username";
        $password_db = "password";
        $dbname = "database_name";
        //Conexion
        $conn = new mysqli($servername, $username, $password_db, $dbname);
        //Comprobar la conexion
        if ($conn->connect_error) {
            die("Error de conexión: " . $conn->connect_error);
        }

        // Preparar la consulta SQL(mismo que el ejercicio1)
        $stmt = $conn->prepare("INSERT INTO usuarios (nombre, apellido1,
        apellido2, user, passwd, foto) VALUES (?, ?, ?, ?, ?, ?)");
        $stmt->bind_param("ssssss", $nombre, $apellido1, $apellido2,
        $usuario, $password, $foto);

        // Ejecutar la consulta
        if (!$stmt->execute()) {
            echo "Error al registrar usuario: " . $stmt->error;
        }

        // Cerrar la conexión y la declaración
        $stmt->close();
        $conn->close();
    } else {
        die("Acceso no permitido.");
    }
    ?>

```

## EJERCICIO 3

```
<html>
<head>
  <script>
    function buscarUsuarios() {
      var query = $('#query').val();
      $.ajax({
        url: 'buscar_usuarios.php',
        method: 'GET',
        data: { query: query },
        success: function(resultado) {
          $('#resultado').html(resultado);
        },
        error: function() {
          alert("Se ha producido un error");
        }
      });
    }
  </script>
</head>
<body>
  <h1>Búsqueda de Usuarios</h1>
  <input type="text" id="query" placeholder="Buscar...">
  <button onclick="buscarUsuarios()">Buscar</button>
  <div id="resultado"></div>
</body>
</html>

<?php
// Conexión a la base de datos (Con los datos del examen)
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "mi_base_de_datos";
//Ejecutar la conexión
$conn = new mysqli($servername, $username, $password, $dbname);

//Comprobando éxito de la conexión
if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}
```

```

// Obtener el término de búsqueda
$query = $_GET['query'];
$query = "%$query%";

// Preparar y ejecutar la consulta SQL
$sql = "SELECT nombre, apellido1, apellido2, user, foto FROM usuarios
WHERE nombre LIKE ? OR apellido1 LIKE ? OR apellido2 LIKE ? ORDER BY
apellido1, apellido2, nombre";
//Hacemos el prepare de la consulta
$stmt = $conn->prepare($sql);
//Le damos los parametros a la consulta
$stmt->bind_param("sss", $query, $query, $query);
//Ejecutamos
$stmt->execute();
$result = $stmt->get_result();

// Generar la tabla de resultados
echo "<table border='1'>";
//Le damos estructura
echo "<tr><th>Nombre</th><th>Primer Apellido</th><th>Segundo
Apellido</th><th>Usuario</th><th>Foto</th></tr>";
//Bucle para sacar el resultado
while ($row = $result->fetch_assoc()) {
    echo "<tr>";
    echo "<td>" . $row['nombre'] . "</td>";
    echo "<td>" . $row['apellido1'] . "</td>";
    echo "<td>" . $row['apellido2'] . "</td>";
    echo "<td>" . $row['user'] . "</td>";
    echo "<td><img src='imagenes/" . $row['foto'] . "'></td>";
    echo "</tr>";
}
//Cerramos tabla
echo "</table>";
//Cerramos consulta y conexion de la BD
$stmt->close();
$conn->close();
?>

```