



**CÓDIGO: DSUP12-GP-04**

**1. TEMA:** Algoritmos y programas (Sistema de ingresos).

**2. TIEMPO DE DURACIÓN:** 120 minutos

**3. FUNDAMENTOS:**

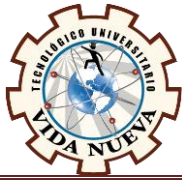
Las variables de sesión en PHP son variables que se almacenan en el servidor y se asocian con un usuario específico a través de un identificador de sesión único. Estas variables se pueden utilizar para almacenar información temporal que se necesita en diferentes páginas o scripts de un sitio web mientras el usuario está en sesión.

Para utilizar variables de sesión en PHP, se debe llamar a la función "session\_start()" al comienzo de cada script de PHP donde se desee utilizar variables de sesión. Esto inicializa la sesión del usuario y crea un identificador de sesión único que se utiliza para asociar las variables de sesión con el usuario.

A continuación, se pueden crear variables de sesión utilizando la sintaxis "\$\_SESSION['nombre\_de\_variable'] = valor;". Estas variables se pueden utilizar y actualizar en diferentes páginas y scripts dentro de la misma sesión.

Al finalizar la sesión del usuario, se puede llamar a la función "session\_destroy()" para eliminar todas las variables de sesión asociadas con el usuario y cerrar la sesión.

Es importante tener en cuenta que las variables de sesión pueden ser sensibles a la seguridad, ya que pueden ser utilizadas para almacenar información confidencial del usuario. Por lo tanto, es importante utilizar medidas de seguridad



adicionales, como cifrado de datos, para proteger la información almacenada en variables de sesión.

#### **4. OBJETIVO(S)**

- Desarrollar un sistema de login para gestión de seguridad de un sistema web.
- Crear una interfaz intuitiva para el login.
- Validar el ingreso de usuarios mediante con el uso de PHP.

#### **5. MATERIALES**

- Software (Sublime Text 3)
- Software (XAMPP)

#### **6. EQUIPOS**

- PC

#### **7. NORMAS DE SEGURIDAD**

- Mantener el área limpia: Es importante mantener el laboratorio limpio y ordenado. Se deben recoger los residuos en recipientes adecuados y limpiar regularmente las superficies de trabajo.
- No dejar los equipos encendidos y sin supervisión: Los equipos deben ser apagados y desconectados cuando no estén en uso. Nunca se deben dejar encendidos y sin supervisión.
- Evitar tocar los componentes electrónicos: Se deben evitar tocar los componentes electrónicos, como tarjetas de circuitos o cables, sin antes haber descargado la electricidad estática del cuerpo.



- Conectar y desconectar los equipos adecuadamente: Al conectar o desconectar equipos, se debe asegurarse de que estén apagados y desconectados de la fuente de alimentación antes de manipularlos.

## 8. PREPARACIÓN PREVIA/ CONOCIMIENTOS PREVIOS

Elemento faltante	Cantidad	Elemento deteriorado	Estado
Sistema operativo windows 10/11	1	Windows 7 Windows xp	
Sublime text	1	V2.0	
Xamp	1	V1.4	

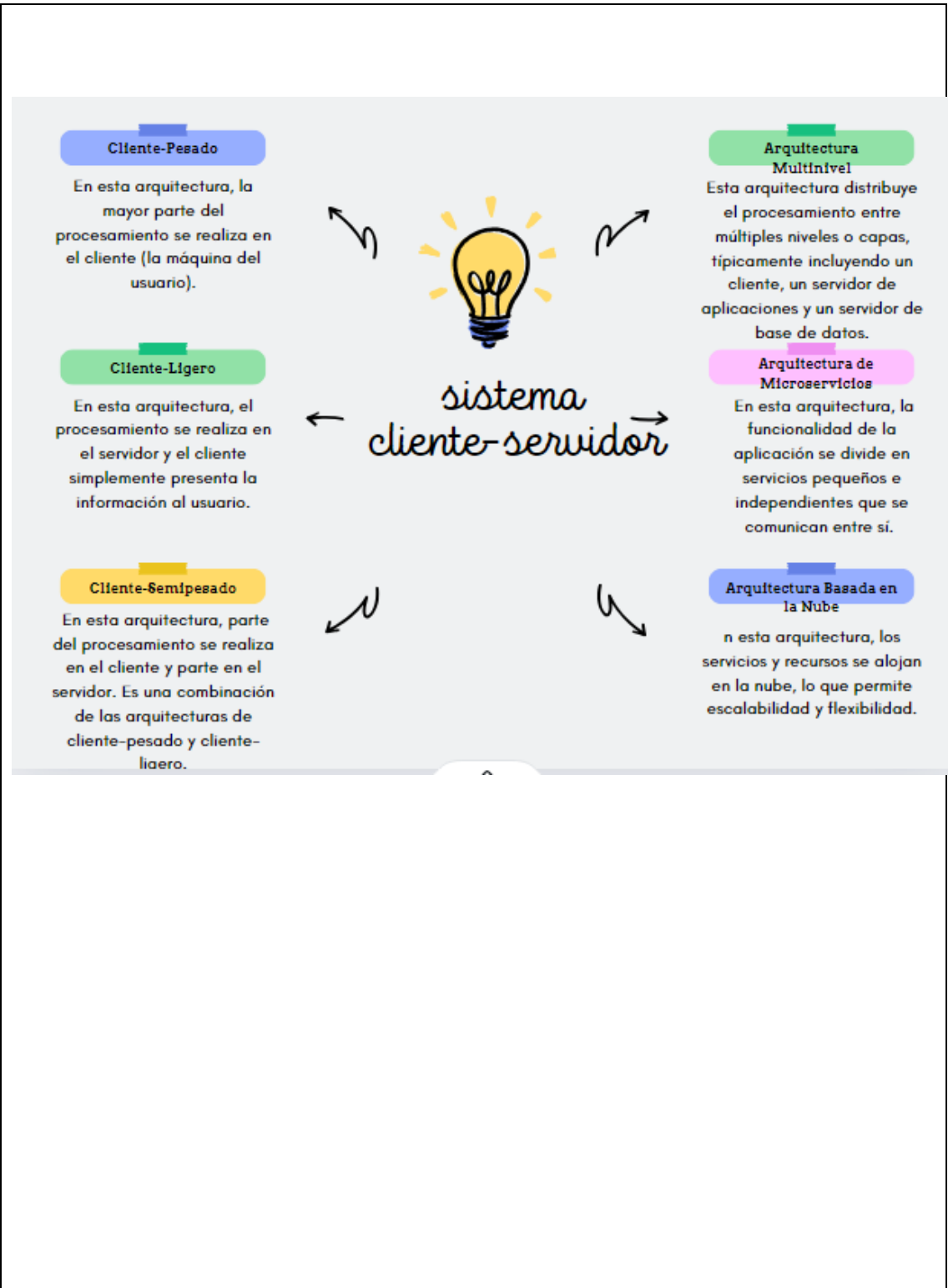
## 9. PROCEDIMIENTO

1. Diseñar una interfaz web con HTML y Javascript según el Anexo 1.
2. Diseñar la interfaz gráfica mediante el desarrollo de hojas de estilo en casada, JavaScript, JQuery y Ajax.
3. Crear la programación del lado del servidor con PHP para leer los usuarios y contraseñas de un archivo de texto plano según el Anexo 2 y conceder acceso al sistema en el caso de encontrar coincidencias.
4. Responder las preguntas planteadas.

## 10. ORGANIZADOR GRÁFICO DE LAS IDEAS PRINCIPALES

**Figura 1:**

*Realizar un organizador gráfico sobre los tipos de sistemas cliente - servidor.*





## 11 ANÁLISIS COMPARATIVO-RESOLUCIÓN DE EJERCICIOS-RESOLUCIÓN DE CASOS-ENSAYO-OTROS.

**Figura 2:**

*Detalle de los métodos de intercambio de datos.*

Métodos de intercambio de datos	
HTTP GET	Es utilizado para solicitar datos de un servidor. En este método, los datos se envían a través de la URL del navegador, lo que lo hace adecuado para peticiones potentes y seguras.
HTTP POST	Es utilizado para enviar datos al servidor para su procesamiento. Los datos se incluyen en el cuerpo de la solicitud, lo que permite enviar grandes cantidades de datos y mantener la URL limpia.
AJAX	Es una técnica que permite a las aplicaciones web enviar y recibir datos de manera asíncrona sin recargar la página completa. Utiliza varios métodos HTTP como GET y POST.
PUT	Es utilizado para actualizar o crear un recurso en el servidor. Los datos se envían en el cuerpo de la solicitud, similar a POST, pero PUT es idempotente, lo que significa que múltiples solicitudes idénticas producirán el mismo resultado.

*Nota: Elaboración propia.*

## 12. PREGUNTAS DE REFUERZO

¿Por qué es importante validar la recepción de datos en el lado del servidor?

Validar la recepción de datos en el lado del servidor es crucial para garantizar la integridad y seguridad de un sistema. Al hacerlo, se previenen ataques de seguridad como la inyección de código malicioso o la manipulación de datos.



¿Cómo funciona el JSON y AJAX en productos de software web?

JSON (JavaScript Object Notation) se utiliza en productos de software web como un formato ligero para intercambiar datos entre el servidor y el cliente. Permite estructurar la información de manera legible y fácilmente procesable por JavaScript.

¿Qué son las capas e programación y como funciona cada una de ellas? De un ejemplo de la capa de negocios

En programación, las capas son niveles de abstracción que organizan el código en una aplicación. La capa de presentación se encarga de la interfaz de usuario, la de negocios maneja la lógica empresarial y la de datos administra la persistencia de datos. Por ejemplo, en la capa de negocios de un sistema de gestión de pedidos, se implementaría la lógica para procesar órdenes, verificar inventario y calcular precios.

### 13. CONCLUSIONES

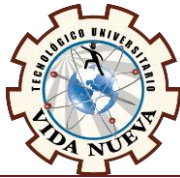
(Relacionadas a los objetivos)

- Se desarrollo un sistema que incluye un login para su acceso.
- Se creó una interfaz bastante cómoda para el usuario.
- Se validó las credenciales para el acceso al sistema usando PHP y Ajax.

### 14. EVALUACIÓN DEL APRENDIZAJE

**Rúbrica:**

CONSIDERACIONES	0,10 MUY BUENO	0, 05 BUENO	0, 01 REGULAR
<b>Elaboración y ejecución de la práctica</b>	La aplicación cumple con el propósito para el que fue diseñada	Desarrolla de parcial con el propósito para el que fue diseñada	Desarrolla de forma incompleta la aplicación diseñada
<b>Interfaz de usuario</b>	La interfaz es estéticamente atractiva y coherente en todo momento	Demuestra un buen diseño y coherente	Demuestra un limitado conocimiento en los temas tratados
<b>Análisis de resultados obtenidos</b>	La aplicación se ejecuta sin problemas y sin retrasos	Los resultados obtenidos de la aplicación contienen errores	Los resultados obtenidos no corresponden a la aplicación propuesta



<b>Recursos didácticos y prácticos</b>	Todas las funcionalidades de la aplicación funcionan correctamente	Tiene limitada las funciones de la aplicación presentada	No contiene funcionalidad en el proyecto presentado
----------------------------------------	--------------------------------------------------------------------	----------------------------------------------------------	-----------------------------------------------------

## 15. BIBLIOGRAFÍA

Gauchat, J. (2012). *El gran libro de HTML5, CSS3 & JAVASCRIPT*. España.  
Prado, M. (2014). *Creación y diseño web edición 2014*. España. Lasso, M. (2013). *Programación java script técnicas esenciales*. España

## 16. ANEXOS

### Figura 3:

*Formulario de login.*

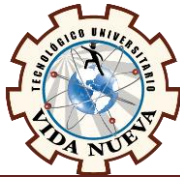
### Login

Nombre de Usuario:

Contraseña:

Login

*Nota: Elaboración propia.*



**Figura 4:**

*Archivo de usuarios y contraseñas.*

## Login

Nombre de Usuario:

isma@gmail.com

Contraseña:

.....

Login

Login exitoso!

Nombre de Usuario: isma@gmail.com

Contraseña: Lavacalola1212

*Nota: Elaboración propia.*

**Figura 5:**

*Index.php*

```
index.php
2  <html Lang="es">
9  <body>
29
30  <script>
31      $(document).ready(function() {
32          $('#loginForm').submit(function(event) {
33              event.preventDefault(); // Prevenir el envío del formulario
34
35              // Obtener los datos del formulario
36              var formData = {
37                  username: $('#username').val(),
38                  password: $('#password').val()
39              };
40
41              // Enviar los datos mediante AJAX
42              $.ajax({
43                  type: 'POST',
44                  url: 'login.php',
45                  data: formData,
46                  dataType: 'json',
47                  encode: true
48              })
49              .done(function(data) {
50                  if (data.success) {
51                      $('#result').html('<div class="alert alert-success">Login exitoso!</div>');
52                      $('#result').append('<p>Nombre de Usuario: ' + data.username + '</p>');
53                      $('#result').append('<p>Contraseña: ' + data.password + '</p>');
54                  } else {
55                      $('#result').html('<div class="alert alert-danger">Error: ' + data.message + '</div>');
56                  }
57              });
58          });
59      });
60  </script>
61  </body>
62  </html>
```



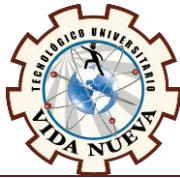


Figura 6:

*Login.php*

```
login.php
1  <?php
2  header('Content-Type: application/json');
3
4  $response = array();
5
6  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
7      // Obtener Los datos enviados por AJAX
8      $username = isset($_POST['username']) ? $_POST['username'] : '';
9      $password = isset($_POST['password']) ? $_POST['password'] : '';
10
11     // Conectar a la base de datos
12     $servername = "localhost"; // Cambiar si es necesario
13     $dbname = "base";
14     $dbuser = "root"; // Cambiar por tu nombre de usuario de la BD
15     $dbpass = ""; // Cambiar por tu contraseña de la BD
16
17     $conn = new mysqli($servername, $dbuser, $dbpass, $dbname);
18
19     // Verificar la conexión
20     if ($conn->connect_error) {
21         $response['success'] = false;
22         $response['message'] = "Error de conexión a la base de datos: " . $conn->connect_error;
23     } else {
24         // Preparar y ejecutar la consulta SQL
25         $stmt = $conn->prepare("SELECT * FROM user WHERE NOMBRE = ? AND CONTRA = ?");
26         $stmt->bind_param("ss", $username, $password);
27         $stmt->execute();
28         $result = $stmt->get_result();
29
30         // Verificar si hay un usuario que coincide
31         if ($result->num_rows > 0) {
```

*Nota: Elaboración propia.*