

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

23-2-2024

CGI BBDD Python

Implantación de Aplicaciones Web 2º ASIR

Several thin, curved lines in dark blue and light grey that originate from the bottom left and curve upwards and to the right.

Andres Catalina Blázquez

Contenido De La Memoria

| | |
|---|----|
| 1. Creación de BBDD | 2 |
| 2. Creación de formularios | 2 |
| a) Registrarse | 2 |
| b) Login | 3 |
| 3. Python | 4 |
| a) Registrarse | 4 |
| I. Iniciar página y recoger datos | 4 |
| II. Conexión a la BBDD y encriptar contraseña | 5 |
| III. Insertar datos a la BBDD..... | 5 |
| IV. Excepciones..... | 6 |
| b) Login | 7 |
| 4. Conclusión y explicaciones..... | 9 |
| a) Modificaciones en httpd.conf de Apache | 9 |
| I. Módulo de CGI | 9 |
| II. Crear Directorio | 10 |
| I. Crear Directorio | 10 |
| 5. Pruebas de que todo funciona correctamente y errores | 11 |

1. Creación de BBDD

```
1  -- CREAR BASE DE DATOS
2  • CREATE DATABASE empresa;
3
4  -- PONER EN USO BASE DE DATOS
5  • USE empresa;
6
7  -- CREAR TABLA USUARIOS
8  • CREATE TABLE usuarios (
9      id INT AUTO_INCREMENT PRIMARY KEY,
10     usuario VARCHAR(25) NOT NULL UNIQUE,
11     contrasena VARCHAR(256) NOT NULL,
12     email VARCHAR(25) NOT NULL
13 )Engine=InnoDB;
```

Output

| # | Time | Action | Message |
|-----|----------|---|-------------------|
| ✓ 1 | 10:41:30 | CREATE DATABASE empresa | 1 row(s) affected |
| ✓ 2 | 10:41:30 | USE empresa | 0 row(s) affected |
| ✓ 3 | 10:41:30 | CREATE TABLE usuarios (id INT AUTO_INCREMENT PRIMARY KEY, usuario VARCHAR(25) NOT NULL ... | 0 row(s) affected |

2. Creación de formularios

En mi caso he optado por 2 formularios (Registrarse y Login) para poder registrar usuarios en la BBDD y además luego poder comprobar que los usuarios pueden acceder a una pagina que se ha creado previamente. (La pagina esta desarrollada de otros años anteriores y mejorada con ayuda de inteligencia artificial)

a) Registrarse

Código de la pagina

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Formulario de Registro</title>
7      <link rel="stylesheet" href=" ../css/index.css">
8  </head>
9  <body>
10     <div class="container">
11         <h2>Registro</h2>
12         <form action=" ../cgi/register.py" method="POST">
13             <div class="form-group">
14                 <label for="usuario">Usuario:</label>
15                 <input type="text" id="usuario" name="usuario" required>
16             </div>
17             <div class="form-group">
18                 <label for="contrasena">Contraseña:</label>
19                 <input type="password" id="contrasena" name="contrasena" required>
20             </div>
21             <div class="form-group">
22                 <label for="email">Email:</label>
23                 <input type="email" id="email" name="email" required>
24             </div>
25             <button type="submit">Registrarse</button>
26         </form>
27     </div>
28 </body>
29 </html>
```

Vista desde forma visual para el cliente

Registro

Usuario:

Contraseña:

Email:

Registrarse

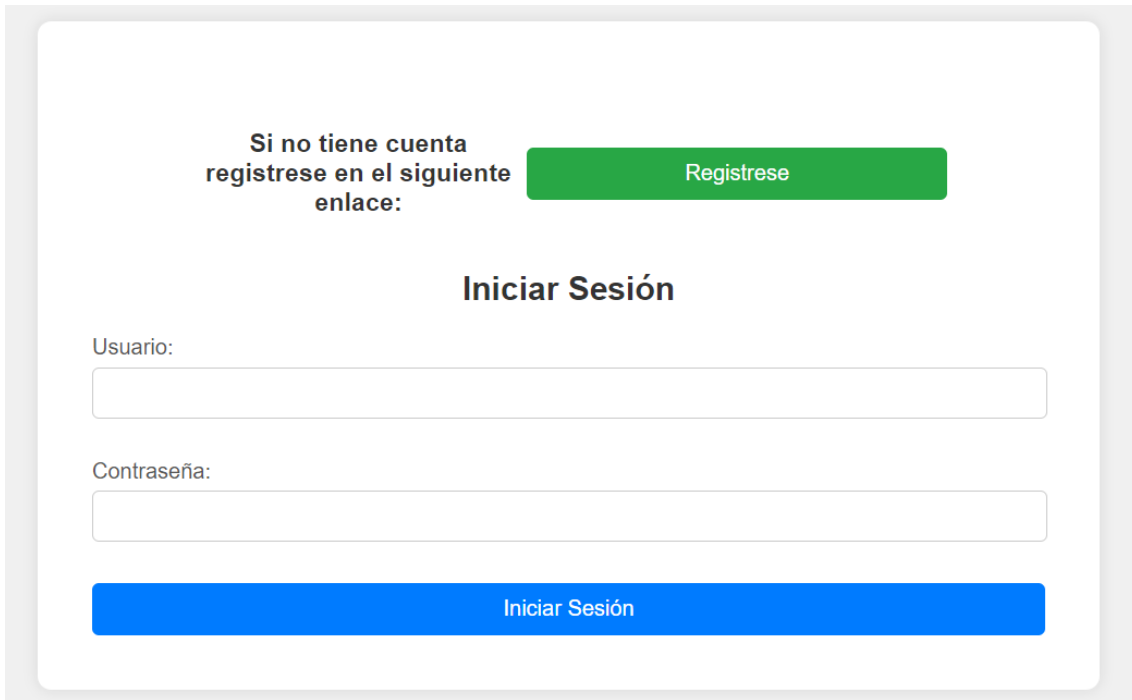
b) Login

Código de la pagina

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Inicio de Sesión</title>
7      <link rel="stylesheet" href="./css/index.css">
8  </head>
9  <body>
10     <div class="container">
11         <div class="login-btn">
12             <h3>Si no tiene cuenta regístrese en el siguiente enlace:</h3>
13             <button type="button" onclick="location.href='./pages/register.html'">Regístrese</button>
14         </div>
15         <h2>Iniciar Sesión</h2>
16         <form action="./cgi/index.py" method="POST">
17             <div class="form-group">
18                 <label for="usuario">Usuario:</label>
19                 <input type="text" id="usuario" name="usuario" required>
20             </div>
21             <div class="form-group">
22                 <label for="contrasena">Contraseña:</label>
23                 <input type="password" id="contrasena" name="contrasena" required>
24             </div>
25             <button type="submit">Iniciar Sesión</button>
26         </form>
27     </div>
28 </body>
29 </html>
```

Vista desde forma visual para el cliente.

(Hemos añadido un botón que redirige a la página de Registrarse por si no tienes cuenta ya que la página de inicio de sesión será la página principal index.html)

El formulario está contenido en un div blanco con un borde gris. En la parte superior, hay un texto "Si no tiene cuenta regístrese en el siguiente enlace:" a la izquierda de un botón verde que dice "Regístrese". Debajo de esto, el título "Iniciar Sesión" está centrado. A continuación, hay dos campos de entrada: "Usuario:" y "Contraseña:", cada uno con un input rectangular. Al final, hay un botón azul que dice "Iniciar Sesión".

Si no tiene cuenta
regístrese en el siguiente
enlace:

Regístrese

Iniciar Sesión

Usuario:

Contraseña:

Iniciar Sesión

3. Python

a) Registrarse

I. Iniciar página y recoger datos

En el archivo de Python importamos las librerías (`import cgi/ import hashlib/ import mysql.connector`), iniciamos el HTML con un `print` (`print("Content-type: text/html\n")`). Nos traemos los datos del HTML con `getvalue()`, habiendo cogido las librerías `cgi`.

```
1  #!C:\Program Files\Python\python.exe
2
3  import cgi
4  import hashlib
5  import mysql.connector
6
7  print("Content-type: text/html\n")
8  print("<html><head><title>Formulario Procesado</title></head><body>")
9  form = cgi.FieldStorage()
10
11  usuario = form.getvalue('usuario', '')
12  contrasena = form.getvalue('contrasena', '')
13  email = form.getvalue('email', '')
```

II. Conexión a la BBDD y encriptar contraseña

En este paso indicamos que, si existe “usuario”, “contraseña” y “email” abra un cursor que realice la conexión a la BBDD y poder comenzar a realizar operaciones en la misma.

Además, como opción de seguridad he puesto que las contraseñas se encripten en formato hexadecimal antes de guardarlas en la BBDD, así no se guardan en texto plano y en caso de que alguien acceda a la BBDD no pueda visualizar las contraseñas de forma clara. Para realizar ese paso tenemos que importar previamente la librería hashlib.

```
if usuario and contrasena and email:
    # Database connection
    try:
        connection = mysql.connector.connect(
            host='localhost',
            database='empresa',
            user='root',
            password='')

        cursor = connection.cursor()

        # Convert the password to bytes
        passwd_bytes = contrasena.encode('utf-8')

        # Create an SHA-256 hash object
        passwd_hash = hashlib.sha256()

        # Update the hash with the password bytes
        passwd_hash.update(passwd_bytes)

        # Get the hexadecimal representation of the hash
        passwd_hash_hex = passwd_hash.hexdigest()
```

III. Insertar datos a la BBDD

Una vez hemos recogido todos los parámetros, generamos una variable llamada query y metemos la inserción, además, en otra variable llamada data introducimos los datos que hemos recogido previamente y lanzamos el cursor para que ejecute la inserción de los datos facilitados por el cliente

```
# Insert data into the database using a parameterized query
query = "INSERT INTO usuarios (usuario, contrasena, email) VALUES (%s, %s, %s)"
data = (usuario, passwd_hash_hex ,email)
cursor.execute(query, data)
```

En caso de que sea correcta la inserción hemos puesto que se muestre una interfaz que le indique al cliente que se ha registrado correctamente.

```
# Commit the changes to the database
connection.commit()
print('<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pagina inicio</title>
  <link rel="stylesheet" href="../../css/welcome.css">
</head>
<body>
  <div class="container">
    <h1>Bienvenido!</h1>
    <p>Se ha registrado correctamente</p>
    <p>Por favor inicie sesion a traves del siguiente boton para poder disfrutar de la pagina</p>
    <a href="../../index.html" class="logout-btn">Iniciar Sesion</a>
  </div>

  <footer>
    <p>&copy; 2024 Your Website. Todos los derechos reservados.</p>
  </footer>
</body>
</html>')
```

IV. Excepciones

Capturamos excepciones con `except`.

Nos ha interesado capturar los errores de MySQL y mostrarlos por pantalla

Con el `finally`, se haya ejecutado el `try` o el `except`, cerramos el cursor y la conexión a MySQL.

```
except mysql.connector.Error as err:
    print("<h2>Error en la base de datos: {}".format(err))
finally:
    # Close the database connection
    if connection.is_connected():
        cursor.close()
        connection.close()
else:
    print("<h2>Faltan datos en el formulario.</h2>")

print("</body></html>")
```

b) Login

Al igual que en anterior este archivo de Python lleva una estructura similar.

Primero recogemos los datos, si existen esos datos abre un cursor, realiza la conexión a la BBDD, realiza la SELECT de los usuarios con el nombre que se ingresa en el formulario y se guarda en una variable. Si esa variable tiene algún valor significa que ese usuario existe entonces pasamos a comprobar la contraseña.

```
1  #!C:\Program Files\Python\python.exe
2
3  import cgi
4  import hashlib
5  import mysql.connector
6
7  print("Content-type: text/html\n")
8
9  form = cgi.FieldStorage()
10 usuario = form.getvalue('usuario', '')
11 contrasena = form.getvalue('contrasena', '')
12
13 if usuario and contrasena:
14     hash_passwd = hashlib.sha256(contrasena.encode()).hexdigest()
15
16     try:
17         connection = mysql.connector.connect(
18             host='localhost',
19             database='empresa',
20             user='root',
21             password='')
22
```



```
23 cursor = connection.cursor()
24
25 query = "SELECT usuario, contrasena FROM usuarios WHERE usuario = %s"
26 cursor.execute(query, (usuario,))
27 result = cursor.fetchone()
28
29 if result:
30     _, hash_contrasena = result
31     if hash_contrasena == hash_passwd:
32         print('<!DOCTYPE html>
33         <html lang="en">
34         <head>
35             <meta charset="utf-8">
36             <meta name="viewport" content="width=device-width, initial-scale=1.0">
37             <title>Pagina inicio</title>
38             <link rel="stylesheet" href="../../css/welcome.css">
39         </head>
40         <body>
41             <header>
42                 <nav>
43                     <ul>
44                         <li><a href="#">Inicio</a></li>
45                         <li><a href="#">Acerca De</a></li>
46                         <li><a href="#">Servicios</a></li>
47                         <li><a href="#">Contactos</a></li>
48                     </ul>
49                 </nav>
50             </header>
51
52             <div class="container">
53                 <h1>Bienvenido!</h1>
54                 <p>Ha iniciado sesion correctamente</p>
55                 <a href="../../index.html" class="logout-btn">Cerrar Sesion</a>
56             </div>
57
58             <footer>
```

* (_, hash_contrasena = result) → Con esto lo que realizamos es la desestructuración de tuplas, es decir, result es una tupla que contiene dos elementos: el primero corresponde al nombre de usuario que se asigna a _ y se descarta y el segundo corresponde a la contraseña que se asigna a hash_contrasena (En Python la barra baja '_' se utiliza para indicar que el valor no se utilizará) *

```
_, hash_contrasena = result
```

Posteriormente para comprobar la contraseña hemos comprobado que la contraseña que esta almacenada en la BBDD sea el mismo hash que el que realizamos mas arriba hasheando la contraseña que nos facilita el usuario.

Si ambas comprobaciones son correctas, muestra una página que será la pagina principal, pero en el caso de que la contraseña no sea correcta lanza por pantalla un mensaje de ERROR sin especificar que la contraseña es incorrecta ya que lo veo un fallo de seguridad. Además, en caso de que la SELECT no encuentre nada al realizar la búsqueda en la BBDD le vamos a indicar que saque por pantalla un mensaje que indique error de usuario.

En este archivo también tenemos las mismas excepciones que antes y finalizamos de la misma forma.

```
        else:
            print("<h2>ERROR!</h2>")
        else:
            print("<h2>Error de usuario!</h2>")
    except mysql.connector.Error as err:
        print("<h2>Error en la base de datos: {}</h2>".format(err))
    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()
    else:
        print("<h2>Por favor, ingrese usuario y contraseña.</h2>")

print("</body></html>")
```

4. Conclusión y explicaciones

Esta practica me ha parecido muy interesante ya que hemos comprobado a través de una librería de Python como se pueden tratar los datos que se ingresan por el cliente a través de una interfaz gráfica.

Con esta practica hemos tenido que modificar y preparar el archivo de configuración de Apache.

a) Modificaciones en httpd.conf de Apache

I. Módulo de CGI

Cargar el módulo responsable de manejar la ejecución de scripts CGI en el servidor web Apache.

```
LoadModule cgi_module modules/mod_cgi.so
```

II. Crear Directorio

Creamos la configuración para el directorio donde va a estar alojada nuestra página

+ExecCGI: indica que los scripts CGI en este directorio pueden ser ejecutados

+Indexes: permite la generación de listas de directorios si no se encuentra un archivo de índice.

AddHandler cgi-script .cgi .pl .py: cuando un cliente solicite un archivo con una de estas extensiones en este directorio, el servidor tratará el archivo como un script CGI y lo ejecutará en lugar de servirlo como un archivo estático.

```
<Directory "${INSTALL_DIR}/www/cgiidos">
  Options +ExecCGI +Indexes
  AddHandler cgi-script .cgi .pl .py
</Directory>
```

I. Crear Directorio

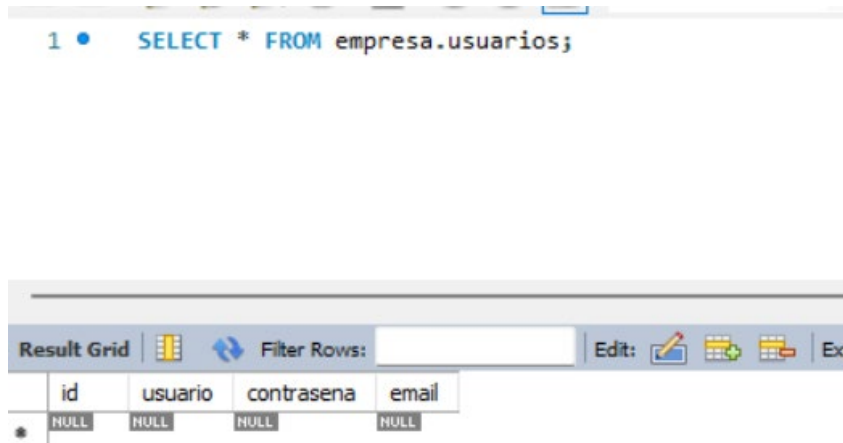
Configurar para que cuando un cliente solicite uno de estos archivos sean manejados como scripts CGI por el servidor web Apache.

```
# AddHandler cgi-script .cgi
AddHandler cgi-script .cgi .pl .py
# For type map (negotiated resources)
```

Para esta práctica me han tenido que ayudar varios compañeros además de buscar información por internet y pedir información y ayuda de configuración a inteligencia artificial además de preguntar a personas que se dedican al desarrollo de aplicaciones web.

5. Pruebas de que todo funciona correctamente y errores

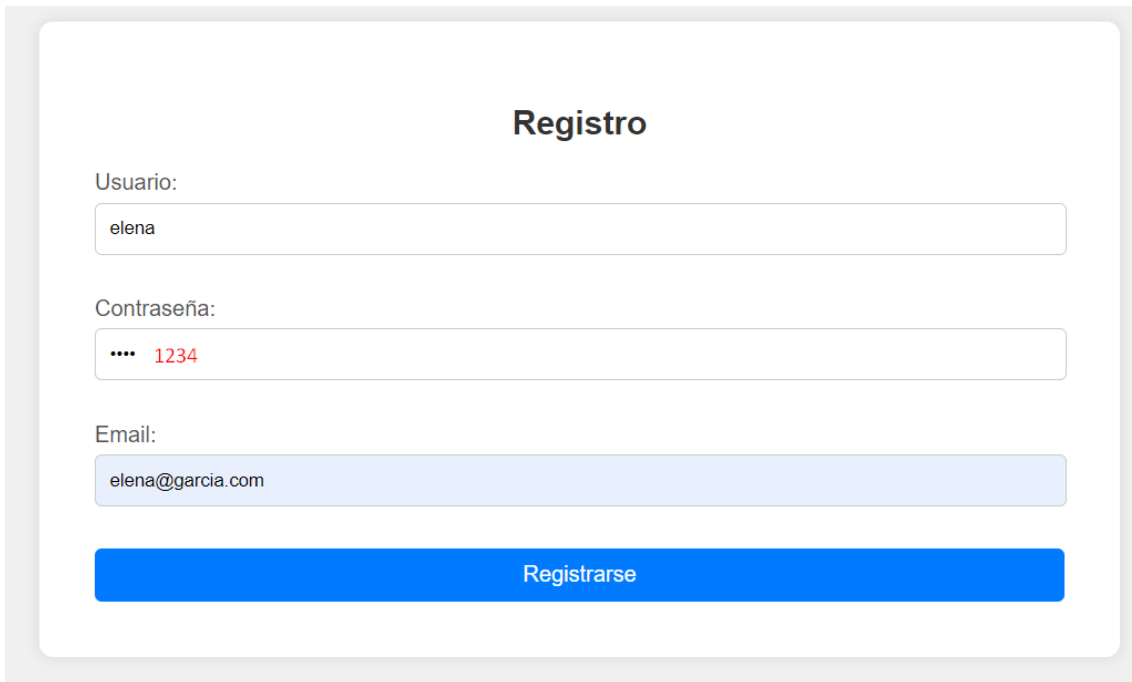
La base de datos no tiene ningún usuario, vamos a registrar uno



Accedemos a la pagina y como no tenemos cuenta pulsamos en Registre

The image shows a web form for login and registration. At the top, there is a red-bordered box containing the text 'Si no tiene cuenta regístrese en el siguiente enlace:' and a green button labeled 'Regístrese'. Three red arrows point from the text to the button. Below this box is the heading 'Iniciar Sesión'. Under the heading, there are two input fields: 'Usuario:' and 'Contraseña:'. At the bottom of the form is a blue button labeled 'Iniciar Sesión'.

Rellenamos los datos que nos solicitan y pulsamos en Registrarse

A registration form titled "Registro" with three input fields: "Usuario:" containing "elena", "Contraseña:" containing masked characters and "1234", and "Email:" containing "elena@garcia.com". A blue "Registrarse" button is at the bottom.

Registro

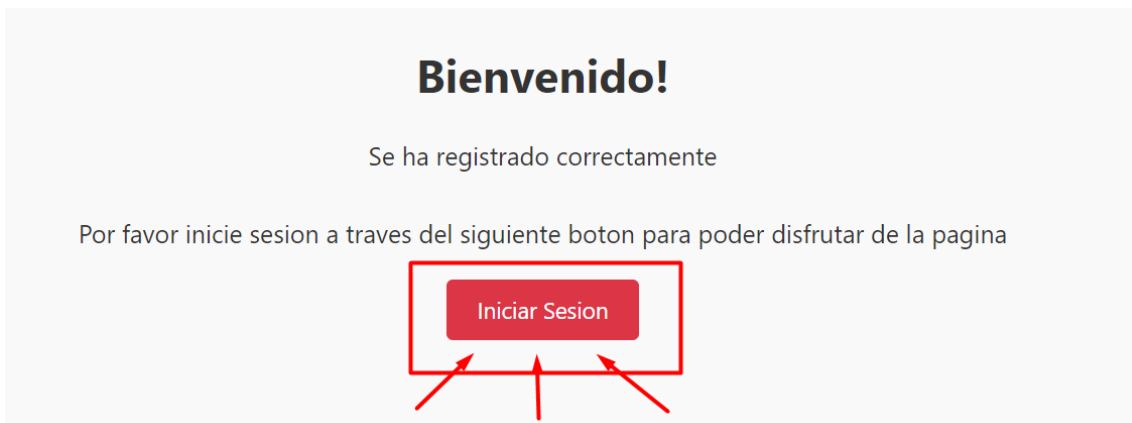
Usuario:
elena

Contraseña:
.... 1234

Email:
elena@garcia.com

Registrarse

Cuando le damos nos sale una interfaz con un texto y un botón, pulsamos en iniciar sesión y ese botón redirige de nuevo a index.html

A welcome screen titled "Bienvenido!" with the text "Se ha registrado correctamente" and "Por favor inicie sesion a traves del siguiente boton para poder disfrutar de la pagina". A red button labeled "Iniciar Sesion" is highlighted with a red box and three red arrows pointing to it from below.

Bienvenido!

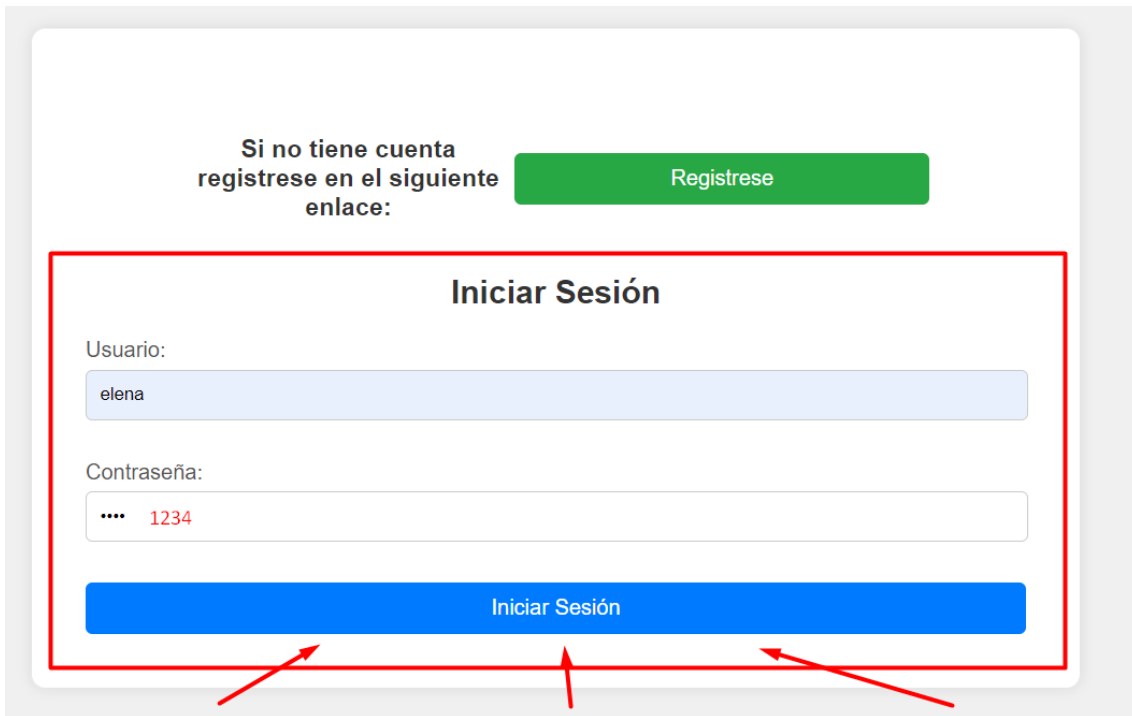
Se ha registrado correctamente

Por favor inicie sesion a traves del siguiente boton para poder disfrutar de la pagina

Iniciar Sesion

Rellenamos los datos (usuario y contraseña) y pulsamos en iniciar sesión.

Automáticamente cargara la pagina



Si no tiene cuenta
registre en el siguiente
enlace:

Registrese

Iniciar Sesión

Usuario:

elena

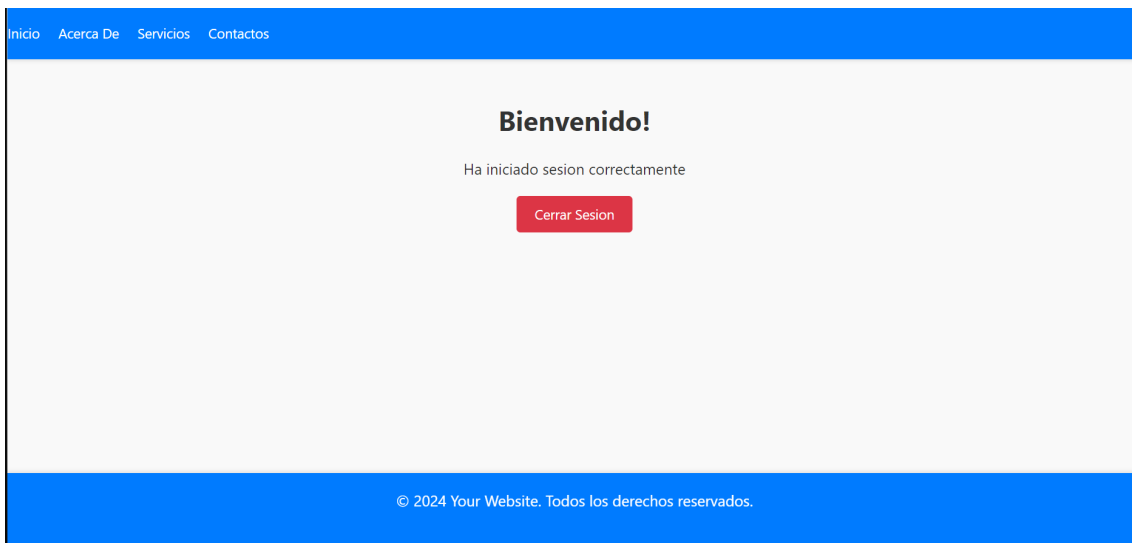
Contraseña:

.... 1234

Iniciar Sesión

Three red arrows point from the bottom of the login form box to the text 'Esta es la página que carga' below.

Esta es la página que carga



Inicio Acerca De Servicios Contactos

Bienvenido!

Ha iniciado sesion correctamente

Cerrar Sesion

© 2024 Your Website. Todos los derechos reservados.

AHORA VAMOS A PROBOCAR FALLOS PARA VER LOS ERRORES QUE TENEMOS

El primer error es de usuario y contraseña incorrectos al iniciar sesión

Error de usuario**Iniciar Sesión**

Usuario:

sajdkhdja

Contraseña:

....

Iniciar Sesión**Error de usuario!**Error de contraseña**Iniciar Sesión**

Usuario:

elena

Contraseña:

.....

Iniciar Sesión**ERROR!**

El segundo error es de intentar registrar el mismo usuario

Registro

Usuario:

Contraseña:

Email:

Registrarse

Error en la base de datos: 1062 (23000): Duplicate entry 'elena' for key 'usuarios.usuario'