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

20-2-2024

CGI BBDD PYTHON

Several thin, curved lines in shades of blue and grey originate from the bottom left and sweep upwards and to the right.

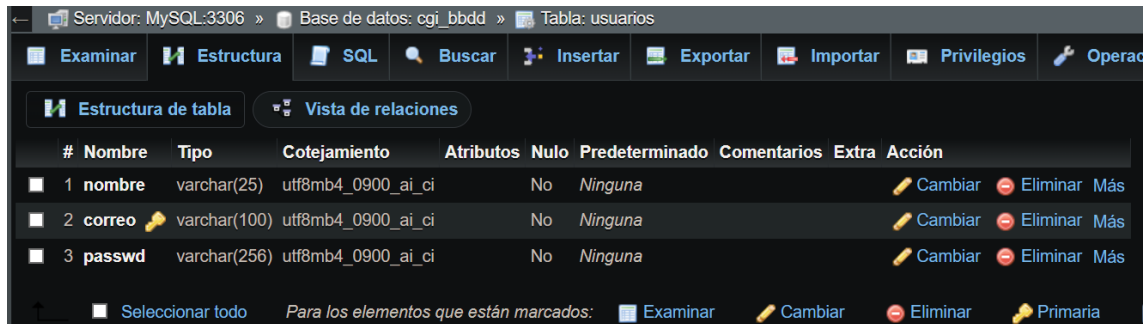
Adrian Delgado
LA ARBOLEDA

Contenido

BASE DE DATOS	1
HTML – INDEX.HTML.....	1
PYTHON – FORMULARIO.PY.....	2
INICIAR PAGINA Y RECOGER DATOS	2
CONEXIÓN A LA BBDD	2
ENCRIPtar CONTRASEÑAS	2
INSERTAR DATOS A MySQL	3
EXCEPCIONES	3
CIERRE PÁGINA.....	3

Base de datos

Creamos una base de datos con estas características: ponemos de primary key el correo, agregamos un nombre y luego añadimos una password que va a ser varchar(256) ya que va encriptada en SHA-255.



The screenshot shows a MySQL database interface with the following table structure:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	nombre	varchar(25)	utf8mb4_0900_ai_ci		No	Ninguna			Cambiar Eliminar Más
2	correo	varchar(100)	utf8mb4_0900_ai_ci		No	Ninguna			Cambiar Eliminar Más
3	passwd	varchar(256)	utf8mb4_0900_ai_ci		No	Ninguna			Cambiar Eliminar Más

Below the table, there are options to 'Seleccionar todo' and 'Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria'.

HTML – index.html

HTML con el formulario, en cada input se pone “name” con el que luego en el archivo Python llamaremos a esos datos por el método POST.

```

2  <html lang="en">
56 <body>
58 <h2>Formulario de Registro</h2>
59
60 <form action="./formulario.py" method="post">
61   <label for="nombre">Nombre:</label>
62   <input type="text" id="nombre" name="nombre" required>
63
64   <label for="email">Email:</label>
65   <input type="email" id="email" name="email" required>
66
67   <label for="passwd">Contraseña:</label>
68   <input type="password" id="passwd" name="passwd" required>
69
70   <input type="submit" value="Registrarse">
71 </form>
72
73 </body>
74 </html>

```

PYTHON – formulario.py

Importamos las librerías necesarias, iniciamos el HTML con un print en Python. Nos traemos los datos del HTML con `getvalue()`, habiendo cogido las librerías `cgi`.

Iniciar pagina y recoger datos

```
import cgi
import hashlib
import mysql.connector

print("Content-type: text/html\n")
print("<html><head><title>Formulario Procesado</title></head><body>")
form = cgi.FieldStorage()

nombre = form.getvalue('nombre', '')
email = form.getvalue('email', '')
passwd = form.getvalue('passwd', '')
```

Conexión a la BBDD

Si existen nombre, email y passwd, realiza una conexión con la base de datos, abre el cursor para poder hacer operaciones con la base de datos.

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

        cursor = connection.cursor()
```

Encriptar contraseñas

Encriptamos la contraseña.

```
# Convert the password to bytes
passwd_bytes = passwd.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()
```

Insertar datos a MySQL

Una vez tenemos todos los parámetros, los hacemos un query y metemos la información en la base de datos.

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

# Commit the changes to the database
connection.commit()

print("<h2>Registro exitoso</h2>")
```

Excepciones

Capturamos excepciones con el except, que va después del try, en este caso vamos a capturar errores de MYSQL.

Con el finally haya hecho el try o el except, vamos cerrar le cursor y la conexión a MYSQL.

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

Cierre página

En el else, decimos que nos faltan parámetros. Y cerramos el HTML.

```
else:  
    print("<h2>Faltan datos en el formulario.</h2>")  
  
print("</body></html>")
```