

Informe de Laboratorio 10

Tema: Bases de Datos MySQL

Estudiante	Escuela	Asignatura
Luis Gustavo Sequeiros Condori lsequeiros@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web I Semestre: II

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 3 Enero 2024	Al 6 Enero 2024

1. Objetivos

- Conocer las bases de datos
- Aprender a utilizar las bases de datos a través del SGBD MariaDB
- Valorar la potencialidad de los Sistemas Gestores de BD

2. Temas a Tratar:

- Expresiones regulares en Perl
- CGI en Perl
- HTML
- CSS
- Consultas a Bases de Datos con MariaDB
- Comandos básicos SQL

3. Ejercicios Propuestos

3.1. Archivo *consult.html*

Listing 1: Estructura de la página de consulta

```
1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <link rel="stylesheet" type="text/css" href="styles.css">
7     <title>Actores y películas</title>
8   </head>
9   <body>
10    <div class="site-wrapper">
```

```
11 <div class="mytitle">
12 <b>Practicando con bases de datos</b>
13 </div>
14 <div class="container">
15 <div class="content">
16 <a href="cgi-bin/first.pl">Primer Ejercicio</a>
17 </div>
18 <div class="content">
19 <a href="cgi-bin/second.pl">Segundo Ejercicio</a>
20 </div>
21 <div class="content">
22 <a href="cgi-bin/third.pl">Tercer Ejercicio</a>
23 </div>
24 </div>
25 <div class="container">
26 <form action="cgi-bin/fourth.pl" >
27 <label for="year">Ingrese un año para mostrar las películas</label>
28 <input type="text" name="year" placeholder="Example: 1987" required>
29 <input class="myinput" type="submit" value="Buscar películas">
30 </form>
31 </div>
32 </div>
33 </body>
34 </html>
```

Una estructura simple con una envoltura para toda la página. Posee un título, 3 opciones que representan los primeros tres ejercicios y un formulario para el cuarto ejercicio, el cual se relaciona con *fourth.pl*. Traté de hacer una estructura de tarjetas para que se vea ordenado y limpio.

3.2. Archivo *styles.css*

Listing 2: Hoja de estilos general

```
1 * {
2   margin: 0;
3   border: 0;
4   box-sizing: border-box;
5 }
6 body {
7   font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;
8 }
9 a {
10  text-decoration: none;
11  color: white;
12 }
13 div {
14  display: flex;
15 }
16 :root {
17   --1-color: #060606;
18   --2-color: #151821;
19   --3-color: #292f3e;
20   --4-color: #42475e;
21   --5-color: #5b5d85;
22 }
23
24 .site-wrapper {
25   align-items: center;
26   justify-content: center;
27   flex-direction: column;
28   row-gap: 5vh;
```

```
29 background-color: var(--5-color);
30 padding: 3vh 0;
31 height: 100vh;
32 }
33 .mytitle {
34 background-color: var(--2-color);
35 font-size: 200%;
36 border-radius: 12px;
37 color: white;
38 padding: 1.5%;
39 text-align: center;
40 }
41 .container {
42 justify-content: center;
43 align-items: center;
44 flex-direction: row;
45 column-gap: 5vh;
46 width: 70%;
47 }
48 .content {
49 align-items: center;
50 width: 70%;
51 background-color: var(--3-color);
52 border-radius: 15px;
53 padding: 2%;
54 justify-content: center;
55 max-height: 72vh;
56 }
57 .content a {
58 color: white;
59 font-size: 160%;
60 padding: 1.7em 0;
61 }
62 .myinput {
63 width: fit-content;
64 display: flex;
65 align-self: center;
66 color: white;
67 background-color: var(--1-color);
68 cursor: pointer;
69 }
70 form {
71 display: flex;
72 flex-direction: column;
73 width: 70%;
74 padding: 3% 4%;
75 background-color: var(--4-color);
76 border-radius: 15px;
77 row-gap: 2vh;
78 }
79 form label {
80 color: white;
81 }
82 input {
83 background-color: lightgray;
84 border-radius: 50em;
85 padding: 1% 2%;
86 color: var(--3-color);
87 font-size: 110%;
88 cursor: text;
89 }
90 .mytable {
91 width: 100%;
92 height: 100%;
93 overflow: auto;
```

```
94 }
95 table {
96     width: 100%;
97 }
98 table th{
99     background-color: var(--1-color);
100     color: white;
101     padding: 10px;
102 }
103 table td {
104     background-color: lightgray;
105     color: var(--1-color);
106     padding: 10px;
107 }
108 .back a {
109     background-color: var(--1-color);
110     border-radius: 50em;
111     font-size: 120%;
112     text-align: center;
113     border: none;
114     padding: 15% 20%;
115 }
```

Se empieza una normalización simple a los elementos y una un poco más especial para las referencias y los contenedores (utilizo flexbox). También se define una fuente para la página. Posteriormente se definen algunos colores que se utilizará. Empezando con el diseño en general, se configura una envoltura para toda la página, que tenga sus elementos centrados. Luego, se añade un título para hacer la página más llamativa. Aquí, nos encontramos con el diseño de los contenedores, *container* para un contenedor de contenidos, y *content* para el contenido en sí mismo. Se trata de dar un formato de tarjetas con bordes redondeados y al centro, esto para el caso de los primeros tres ejercicios. Para el caso del último, se incluye el diseño del *form*, de manera que la interfaz de ingreso de datos sea llamativa y sencilla para el usuario. Nos encontramos con el diseño para las respuestas del CGI. Dentro de un contenedor *content*, se encuentra otro contenedor llamado *mytable*, el cual fue creado para manejar el desbordamiento de la tabla, tanto horizontal como verticalmente. Por último tenemos el diseño de los botones volver.

3.3. Archivos CGI

Listing 3: CGI del primer ejercicio

```
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4  use CGI;
5  use DBI;
6  use Net::Address::IP::Local;
7
8  my $cgi = CGI->new;
9  $cgi->charset('UTF-8');
10 print $cgi->header('text/html; charset=UTF-8');
11 print <<HTML;
12 <!DOCTYPE html>
13 <html lang="es">
14   <head>
15     <meta charset="UTF-8">
16     <meta name="viewport" content="width=device-width, initial-scale=1.0">
17     <link rel="stylesheet" type="text/css" href="../styles.css">
18     <title>Actor 5</title>
19   </head>
```

```

20  <body>
21    <div class="site-wrapper">
22      <div class="mytitle">
23        <b>El actor de ID: 5</b>
24      </div>
25      <div class="content">
26        HTML
27        my $user = "alumno";
28        my $password = "pweb1";
29        my $ip = Net::Address::IP::Local->public_ipv4;
30        my $dsn = "DBI:MariaDB:database=pweb1;host=$ip";
31        my $dbh = DBI->connect($dsn, $user, $password) or die("No se pudo conectar!");
32        my $id = "5";
33        my $sth = $dbh->prepare("INSERT INTO Actor(ActorID, Name) Values(?,?)");
34        $sth->execute($id, "Wally Diego");
35
36        print "<div class='mytable'>\n<table>\n<tr>\n";
37        $sth = $dbh->prepare("DESC Actor");
38        $sth->execute();
39        while(my @head = $sth->fetchrow_array){
40          print "<th>$head[0]</th>\n";
41        }
42        print "</tr>\n";
43        $sth = $dbh->prepare("SELECT * FROM Actor WHERE ActorID=?");
44        $sth->execute($id);
45
46        while(my @row = $sth->fetchrow_array){
47          print "<tr>\n";
48          foreach my $data(@row){
49            print "<td>$data</td>\n";
50          }
51          print "</tr>\n";
52        }
53        $sth->finish;
54        $dbh->disconnect;
55        print <<HTML;
56          </table>
57          </div>
58          </div>
59          <div class="back">
60            <a href="../consult.html">Volver</a>
61          </div>
62        </div>
63      </body>
64    </html>
65  HTML

```

Al comienzo del archivo se puede observar la estructura de la página a mostrar, la cual es prácticamente la misma que *consult.html*, así como la creación del objeto *cgi*. Luego, se conecta a la base de datos MariaDB con el *y* se guarda la referencia en *dbh*. Aquí se hace uso de la biblioteca *DBI* y *Net::Address::IP::Local*, con esta última se obtiene el IP del servidor. Luego se prepara y ejecuta una inserción en la base de datos, en este caso del ActorID 5. A continuación, se obtienen los campos (columnas) para incluirlo en la página a mostrar. Nuevamente se prepara la referencia de la base de datos para obtener los datos del actor de ID 5, luego se ejecuta la petición, los registros son obtenidos uno a uno en *row*, para mostrarlos en el HTML. Luego simplemente se muestra un botón para volver a la página web de consulta.

Listing 4: CGI del segundo ejercicio

```
1 #!/usr/bin/perl
```

```

2  use strict;
3  use warnings;
4  use CGI;
5  use DBI;
6  use Net::Address::IP::Local;
7
8  my $cgi = CGI->new;
9  $cgi->charset('UTF-8');
10 print $cgi->header('text/html; charset=UTF-8');
11 print <<HTML;
12 <!DOCTYPE html>
13 <html lang="es">
14   <head>
15     <meta charset="UTF-8">
16     <meta name="viewport" content="width=device-width, initial-scale=1.0">
17     <link rel="stylesheet" type="text/css" href="../styles.css">
18     <title>Películas de 1985</title>
19   </head>
20   <body>
21     <div class="site-wrapper">
22       <div class="mytitle">
23         <b>Películas de 1985</b>
24       </div>
25       <div class="content">
26         HTML
27         my $user = "alumno";
28         my $password = "pweb1";
29         my $ip = Net::Address::IP::Local->public_ipv4;
30         my $dsn = "DBI:MariaDB:database=pweb1;host=$ip";
31         my $dbh = DBI->connect($dsn, $user, $password) or die("No se pudo conectar!");
32         my $year = "1985";
33
34         print "<div class='mytable'>\n<table>\n<tr>\n";
35         my $sth = $dbh->prepare("DESC Movie");
36         $sth->execute();
37         while(my @head = $sth->fetchrow_array){
38           print "<th>$head[0]</th>\n";
39         }
40         print "</tr>\n";
41         my $sth = $dbh->prepare("SELECT * FROM Movie WHERE Year=?");
42         $sth->execute($year);
43         while(my @row = $sth->fetchrow_array){
44           print "<tr>\n";
45           foreach my $data(@row){
46             print "<td>$data</td>\n";
47           }
48           print "</tr>\n";
49         }
50         $sth->finish;
51         $dbh->disconnect;
52         print <<HTML;
53           </table>
54           </div>
55           </div>
56           <div class="back">
57             <a href="../consult.html">Volver</a>
58           </div>
59         </div>
60       </body>
61     </html>
62     HTML

```

Este CGI se ejecuta de la misma manera que el anterior, pero esta vez muestra las películas del año 1985, por lo tanto los campos que muestra son de la tabla *Movie*.

Listing 5: CGI del tercer ejercicio

```
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4  use CGI;
5  use DBI;
6  use Net::Address::IP::Local;
7
8  my $cgi = CGI->new;
9  $cgi->charset('UTF-8');
10 print $cgi->header('text/html; charset=UTF-8');
11 print <<HTML;
12 <!DOCTYPE html>
13 <html lang="es">
14   <head>
15     <meta charset="UTF-8">
16     <meta name="viewport" content="width=device-width, initial-scale=1.0">
17     <link rel="stylesheet" type="text/css" href="../styles.css">
18     <title>Las más puntuadas</title>
19   </head>
20   <body>
21     <div class="site-wrapper">
22       <div class="mytitle">
23         <b>Las más puntuadas</b>
24       </div>
25       <div class="content">
26 HTML
27 my $user = "alumno";
28 my $password = "pweb1";
29 my $ip = Net::Address::IP::Local->public_ipv4;
30 my $dsn = "DBI:MariaDB:database=pweb1;host=$ip";
31 my $dbh = DBI->connect($dsn, $user, $password) or die("No se pudo conectar!");
32 my $score = "7";
33 my $votes = "5000";
34
35 print "<div class='mytable'>\n<table>\n<tr>\n";
36 my $sth = $dbh->prepare("DESC Movie");
37 $sth->execute();
38 while(my @head = $sth->fetchrow_array){
39   print "<th>$head[0]</th>\n";
40 }
41 print "</tr>\n";
42 $sth = $dbh->prepare("SELECT * FROM Movie WHERE Score > ? && Votes > ?");
43 $sth->execute($score, $votes);
44 while(my @row = $sth->fetchrow_array){
45   print "<tr>\n";
46   foreach my $data(@row){
47     print "<td>$data</td>\n";
48   }
49   print "</tr>\n";
50 }
51 $sth->finish;
52 $dbh->disconnect;
53 print <<HTML;
54   </table>
55   </div>
56   </div>
57   <div class="back">
58     <a href="../consult.html">Volver</a>
59   </div>
60 </div>
61 </body>
62 </html>
63 HTML
```

Se ejecuta como en los casos anteriores, pero esta vez muestra las películas con un Score mayor a 7 y Votos mayores a 5000. Lo único que cambia es la sentencia de la petición de datos.

Listing 6: CGI del cuarto ejercicio

```
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4  use CGI;
5  use DBI;
6  use Net::Address::IP::Local;
7
8  my $cgi = CGI->new;
9  $cgi->charset('UTF-8');
10 print $cgi->header('text/html; charset=UTF-8');
11 print <<HTML;
12 <!DOCTYPE html>
13 <html lang="es">
14   <head>
15     <meta charset="UTF-8">
16     <meta name="viewport" content="width=device-width, initial-scale=1.0">
17     <link rel="stylesheet" type="text/css" href="../styles.css">
18     <title>Películas por año</title>
19   </head>
20   <body>
21     <div class="site-wrapper">
22       <div class="mytitle">
23         <b>Pelis por año</b>
24       </div>
25       <div class="content">
26         HTML
27         my $user = "alumno";
28         my $password = "pweb1";
29         my $ip = Net::Address::IP::Local->public_ipv4;
30         my $dsn = "DBI:MariaDB:database=pweb1;host=$ip";
31         my $dbh = DBI->connect($dsn, $user, $password) or die("No se pudo conectar!");
32         my $year = $cgi->param('year');
33
34         print "<div class='mytable'>\n<table>\n<tr>\n";
35         my $sth = $dbh->prepare("DESC Movie");
36         $sth->execute();
37         while(my @head = $sth->fetchrow_array){
38           print "<th>$head[0]</th>\n";
39         }
40         print "</tr>\n";
41
42         $sth = $dbh->prepare("SELECT * FROM Movie WHERE Year=?");
43         $sth->execute($year);
44         while(my @row = $sth->fetchrow_array){
45           print "<tr>\n";
46           foreach my $data(@row){
47             print "<td>$data</td>\n";
48           }
49           print "</tr>\n";
50         }
51         $sth->finish;
52         $dbh->disconnect;
53         print <<HTML;
54         </table>
55         </div>
56         </div>
57         <div class="back">
58           <a href="../consult.html">Volver</a>
```



```
59     </div>
60   </div>
61 </body>
62 </html>
63 HTML
```

Para este último CGI, la interacción con el servidor es la misma, pero, esta vez se obtienen los datos de un formulario y con el método *get*. En este caso, se ingresa un año y el CGI realiza la petición a la base de datos con el año ingresado.

3.4. Recursos

- Video sobre el funcionamiento de la página <https://youtu.be/6ISN8tj86vQ>.
- Repositorio de GitHub para commits <https://github.com/gusCreator/pweb-course/tree/main/lab10>.