

Informe de Laboratorio 09

Tema: Ejemplo Unificado

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 30 Diciembre 2023	Al 3 Enero 2024

1. Temas a Tratar:

- Expresiones regulares en Perl.
- CGI en Perl.
- HTML
- CSS

2. Actividades

2.1. Archivo *consult.html*

Listing 1: Estructura de una 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>Consulta de Universidades licenciadas</title>
8   </head>
9   <body>
10    <div class="site-wrapper">
11      <div class="mytitle">
12        <b>Búsqueda de universidades licenciadas</b>
13      </div>
14      <form class="content" action="cgi-bin/consult.pl">
15        <label for="kind">Elija opción e ingrese una frase:</label>
16        <div class="myinput">
17          <select name="kind" required>
18            <option value="name">Nombre</option>
19            <option value="period">Periodo de Licenciamiento</option>
20            <option value="localRegion">Departamento Local</option>
21            <option value="studyProgram">Nombre del programa (Carrera)</option>
22          </select>
23          <input type="text" name="keyword" required>
24        </div>
```

```
25     <input type="submit" value="Buscar">
26   </form>
27 </div>
28 </body>
29 </html>
```

Una estructura simple con una envoltura para toda la página. Posee un título y un texto guía. Luego está el formulario para la entrada de datos, el cual se relaciona con *consult.pl*. En este formulario se escoge un tipo de información que está referenciado por *kind* y un texto obligatorio para iniciar la búsqueda, el cual está referenciado por *keyword*. A partir de estos parámetros se inicia una búsqueda en el CGI que dará como resultado otra página html.

2.2. 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   width: 100%;
13   height: 100%;
14 }
15 :root{
16   --dark: #160D18;
17   --darkblue: #23145B;
18   --blue: #09456C;
19   --greenblue: #026F6E;
20   --skyblue: #1CA39E;
21 }
22
23 .site-wrapper {
24   display: flex;
25   align-items: center;
26   justify-content: center;
27   height: 100vh;
28   flex-direction: column;
29   row-gap: 4%;
30   background-color: var(--darkblue);
31   padding: 3vh 0;
32 }
33 .mytitle {
34   background-color: var(--greenblue);
35   font-size: 200%;
36   border-radius: 12px;
37   color: white;
38   padding: 1.5%;
39 }
40 .content {
41   display: flex;
42   align-items: center;
43   flex-direction: column;
44   row-gap: 2vh;
45   width: 70%;
```

```
46 }
47 .content label {
48   color: white;
49   font-size: 120%;
50 }
51 .answer {
52   background-color: var(--skyblue);
53   border-radius: 15px;
54   padding: 1%;
55   overflow-y:auto ;
56 }
57 table {
58   display: flex;
59   width: 100%;
60   overflow-x: auto;
61   border: solid 10px none;
62 }
63 table th{
64   background-color: var(--dark);
65   color: white;
66   padding: 10px;
67 }
68 table td {
69   background-color: lightgray;
70   color: var(--dark);
71   padding: 10px;
72 }
73 .myinput {
74   display: flex;
75   flex-direction: row;
76   column-gap: 1%;
77   border-radius: 50%;
78   justify-content: space-between;
79   width: 80%;
80 }
81 .myinput select {
82   width: 30%;
83   border-radius: 50em 0 0 50em;
84   padding: 1%;
85   background-color: var(--skyblue);
86   color: var(--dark);
87   font-size: 100%;
88   cursor: pointer;
89 }
90 .myinput input {
91   background-color: lightgray;
92   border-radius: 0 50em 50em 0;
93   width: 70%;
94   padding: 1%;
95   color: var(--dark);
96 }
97 input {
98   background-color: var(--blue);
99   border-radius: 50em;
100  padding: 1% 2%;
101  color: white;
102  font-size: 120%;
103  cursor:pointer;
104 }
105 .back {
106   background-color: var(--blue);
107   border-radius: 50em;
108   font-size: 120%;
109   text-align: center;
110   border: none;
```

```
111 padding: 1% 2%;  
112 }
```

Se empieza una normalización simple a los elementos y una un poco más especial para las referencias. 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. Nos encontramos con el diseño para el CGI, un contenedor para las respuestas de la búsqueda, en el cual hay una tabla con un diseño llamativo. Algo interesante a recalcar aquí es que el desbordamiento horizontal de la tabla se maneja en el elemento con el mismo nombre, pero el desbordamiento vertical se maneja en el contenedor de la tabla, creando el resultado que se observa en la página. Luego están los diseños del formulario, los más importantes a mi parecer son el de la entrada de texto y el seleccionador. Los cuales pareciese que fuesen uno solo con sus bordes en los extremos. Por último tenemos el diseño de los botones submit y volver.

2.3. Archivo *consult.pl*

Listing 3: CGI para la búsqueda

```
1  #!/usr/bin/perl  
2  use strict;  
3  use warnings;  
4  use CGI;  
5  
6  my $cgi = CGI->new;  
7  $cgi->charset('UTF-8');  
8  print $cgi->header('text/html; charset=UTF-8');  
9  print <<HTML;  
10 <!DOCTYPE html>  
11 <html lang="es">  
12   <head>  
13     <meta charset="UTF-8">  
14     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
15     <link rel="stylesheet" type="text/css" href="../styles.css">  
16     <title>Página de Búsqueda - Universidades licenciadas</title>  
17   </head>  
18   <body>  
19     <div class="site-wrapper">  
20       <div class="mytitle">  
21         <b>Resultados de la búsqueda</b>  
22       </div>  
23       <div class="content answer">  
24 HTML  
25  
26 my $kind = $cgi->param('kind');  
27 my $keyword = $cgi->param('keyword');  
28 if(!($kind eq "period")){  
29   $keyword = uc($keyword);  
30 }  
31 my $flag;  
32 open(IN, "../data.csv" ) or die "<h2>Error al abrir el archivo</h2>";  
33 if(<IN>){  
34   print<<BLOCK;  
35   <table>  
36     <tr>  
37       <th>Código</th>  
38       <th>Nombre</th>  
39       <th>Tipo de Gestión</th>  
40       <th>Estado</th>
```

Página 5

```
104     $dict{"localRegion"} = $3;  
105     $dict{"studyProgram"} = $4;  
106 }  
107 return %dict;  
108 }
```

Al comienzo del archivo se puede observar la estructura de la página a mostrar, la cual es prácticamente la misma que *consult.html*. Luego, se extraen los datos obtenidos del formulario y se los referencia con variables. En la base de datos, todo estaba en mayúscula, por lo que si los datos están ingresado en minúscula con transformados. Luego se declara una bandera que indicará si se encontraron o no Resultados. Se abre el archivo con los datos y se crea una tabla en HTML para mostrar los resultados. A continuación se activa la codificación UTF-8 para las tildes presentes en la base de datos y se busca coincidencias de acuerdo a los parámetros establecidos, aquí se usa expresiones regulares. Si los datos coinciden, se imprime una fila de tabla mostrando datos relevantes acerca de la universidad encontrada. Luego simplemente se muestra un botón para volver y termina la página web. Al final del archivo perl se encuentra la subrutina *findInLine* que permite almacenar los datos necesarios de una universidad en un mapa, aquí también se utilizan expresiones regulares para la extracción de dichos datos.

2.4. Recursos

- Video sobre el funcionamiento de la página <https://youtu.be/GsnAyDq2Puo>.
- Repositorio de GitHub para commits <https://github.com/gusCreator/pweb-course/tree/main/lab09>.