

Proyecto Final Sistemas Web

**Samuel Olmedo Ortiz
Pablo Esteban Martínez García
Klever Damián García Reynoso**

Fecha:

09/12/2022

Universidad Veracruzana

Catedrático: Profe

Front-end:

Index.html

Para la pagina principal se ocupó el siempre confiable index.html con el cual es el, como ya sabemos se identifican las pagina principales. Haciendo las conexiones a los demás archivos de entorno y desarrollo respectivamente ocupamos archivos separados para el estilo de la pagina con css y js para la implementación limpia o lo mas limpia posible dentro de los archivos de la página.

Se tiene la clásica estructura con de construcción, también se tiene ya un header dentro del body para hacer lo que es un nav, que va en la parte superior de la página, es la que va a aparecer en todas las pestañas exceptuando las correspondientes al login

```
index.html > _
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>CafeSUS</title>
8
9   <link rel="shortcut icon" href="cafeteria/images/favicon.ico" type="image/x-icon">
10  <link rel="apple-touch-icon" href="cafeteria/images/apple-touch-icon.png">
11  <link rel="stylesheet" href="cafeteria/css/bootstrap.min.css">
12  <link rel="stylesheet" href="cafeteria/css/style.css">
13  <link rel="stylesheet" href="cafeteria/css/responsive.css">
14
15 </head>
16
17 <body>
18   <!-- Aquí esta el header -->
19   <header class="top-navbar">
20     <nav class="navbar navbar-expand-lg navbar-light bg-light">
21       <div class="container">
22         <a class="navbar-brand" href="index.html">
23           
24         </a>
25         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbars-rs-food" aria-controls="navbars-rs-food" aria-expanded="false" aria-label="Toggle navigation">
26           <span class="navbar-toggler-icon"></span>
27         </button>
28         <div class="collapse navbar-collapse" id="navbars-rs-food">
29           <ul class="navbar-nav ml-auto">
30             <li class="nav-item active"><a class="nav-link" href="index.html">Inicio</a></li>
31             <li class="nav-item"><a class="nav-link" href="cafeteria/menu.html">Menú</a></li>
32
33             <li class="nav-item"><a class="nav-link" href=""></a></li>
34             <li class="nav-item dropdown">
35               <a class="nav-link dropdown-toggle" href="#" id="dropdown-a" data-toggle="dropdown">Registro</a>
36               <div class="dropdown-menu" aria-labelledby="dropdown-a">
37                 <a class="dropdown-item" href="cafeteria/Registro/registro.html">Registro</a>
38                 <a class="dropdown-item" href="cafeteria/inicio de sesion/login.html">Inicio Sesion</a>
39               </div>
40             </li>
41           </ul>
42         </div>
43       </div>
44     </nav>
45   </header>
46
```

Implementamos una serie de divs para poder organizar las estructuras y de esa forma poder identificar en los documentos de css y js y poder asignar propiedades dentro de cada uno de los div, esta es la parte que del carrusel que va en la página de inicio.

```
index.html
index.html > html > body
46
47
48 <div id="slides" class="cover-slides">
49   <ul class="slides-container">
50     <li class="text-left">
51       
52       <div class="container">
53         <div class="row">
54           <div class="col-md-12">
55             <h1 class="m-b-20"><strong>Bienvenido a cafeSUS</strong></h1>
56             <p class="m-b-40">Vive la experiencia de tomar un buen café <br>
57             Todo tipo de cafe de todo el mundo en un solo lugar</p>
58             <p><a class="btn btn-lg btn-circle btn-outline-new-white" href="cafeteria/menu.html">Menu</a></p>
59           </div>
60         </div>
61       </div>
62     </li>
63     <li class="text-left">
64       
65       <div class="container">
66         <div class="row">
67           <div class="col-md-12">
68             <h1 class="m-b-20"><strong>Bienvenido a cafeSUS</strong></h1>
69             <p class="m-b-40">Vive la experiencia de tomar un buen café <br>
70             Todo tipo de cafe de todo el mundo en un solo lugar</p>
71             <p><a class="btn btn-lg btn-circle btn-outline-new-white" href="cafeteria/menu.html">Menu</a></p>
72           </div>
73         </div>
74       </div>
75     </li>
76     <li class="text-left">
77       
78       <div class="container">
79         <div class="row">
80           <div class="col-md-12">
81             <h1 class="m-b-20"><strong>Bienvenido a cafeSUS</strong></h1>
82             <p class="m-b-40">Vive la experiencia de tomar un buen café <br>
83             Todo tipo de cafe de todo el mundo en un solo lugar</p>
84             <p><a class="btn btn-lg btn-circle btn-outline-new-white" href="cafeteria/menu.html">Menu</a></p>
85           </div>
86         </div>
87       </div>
88     </li>
89   </ul>
90 </div>
```

Posteriormente se tiene un footer que contiene un mensaje de bienvenida o de inspiración.

Menu.html

De la misma forma se tiene el navbar en el menú para la interacción completa

```
index.html menu.html X
cafeteria > menu.html > html > body
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8
9 <title>CafeSUS</title>
10 <meta name="keywords" content="">
11 <meta name="description" content="">
12 <meta name="author" content="">
13
14 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">
15 <link rel="apple-touch-icon" href="images/apple-touch-icon.png">
16
17 <link rel="stylesheet" href="css/bootstrap.min.css">
18 <link rel="stylesheet" href="css/style.css">
19 <link rel="stylesheet" href="css/responsive.css">
20
21 </head>
22
23 <body>
24 <!-- Aquí esta el header -->
25 <header class="top-navbar">
26 <nav class="navbar navbar-expand-lg navbar-light bg-light">
27 <div class="container">
28 <a class="navbar-brand" href="index.html">
29 
30 </a>
31 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbars-rs-food" aria-controls="navbars-rs-food" aria-expanded="false" aria-label="To
32 <span class="navbar-toggler-icon"></span>
33 </button>
34 <div class="collapse navbar-collapse" id="navbars-rs-food">
35 <ul class="navbar-nav ml-auto">
36 <li class="nav-item"><a class="nav-link" href="index.html">Inicio</a></li>
37 <li class="nav-item active"><a class="nav-link" href="menu.html">Menu</a></li>
38
39 <li class="nav-item"><a class="nav-link" href=""></a></li>
40 <li class="nav-item dropdown">
41 <a class="nav-link dropdown-toggle" href="#" id="dropdown-a" data-toggle="dropdown">Registro</a>
42 <div class="dropdown-menu" aria-labelledby="dropdown-a">
43 <a class="dropdown-item" href="inicio de sesion/login.html">Registro</a>
44 <a class="dropdown-item" href="inicio de sesion/login.html">Inicio Sesión</a>
45 </div>
46 </li>
47 </ul>
48 </div>
49 </nav>
50 </header>
51
```

Aquí empieza la estructura de menú que donde ocurre ya la interacción con css

```
index.html menu.html X
cafeteria > menu.html > html > body
64
65 <!-- Aquí esta el Menú -->
66 <div class="menu-box">
67 <div class="container">
68 <div class="row">
69 <div class="col-lg-12">
70 <div class="heading-title text-center">
71 <h2>Bienvenido</h2>
72 <p>Observa lo que ofrecemos para ti te ofrecemos </p>
73 </div>
74 </div>
75 </div>
76
77 <div class="row inner-menu-box">
78 <div class="col-3">
79 <div class="nav flex-column nav-pills" id="v-pills-tab" role="tablist" aria-orientation="vertical">
80 <a class="nav-link active" id="v-pills-home-tab" data-toggle="pill" href="#v-pills-home" role="tab" aria-controls="v-pills-home" aria-selected="true">Todo</a>
81 <a class="nav-link" id="v-pills-profile-tab" data-toggle="pill" href="#v-pills-profile" role="tab" aria-controls="v-pills-profile" aria-selected="false">Bebidas</a>
82 <a class="nav-link" id="v-pills-messages-tab" data-toggle="pill" href="#v-pills-messages" role="tab" aria-controls="v-pills-messages" aria-selected="false">Almuerzos
83 <a class="nav-link" id="v-pills-settings-tab" data-toggle="pill" href="#v-pills-settings" role="tab" aria-controls="v-pills-settings" aria-selected="false">Combos y
84 </div>
85 </div>
86
87 <div class="col-9">
88 <div class="tab-content" id="v-pills-tabContent">
89 <div class="tab-pane fade show active" id="v-pills-home" aria-labelledby="v-pills-home-tab">
90 <div class="row">
91 <div class="col-lg-4 col-md-6 special-grid drinks">
92 <div class="gallery-single fix">
93 
94 <div class="formatotexto">
95 <h4>Cafés y también cafés tematicos</h4>
96 <p>Variedad y sabor, todo de calidad y para su deleite, escoge el que mas te guste <br></p>
97 <h5> $35-$90</h5>
98 </div>
99 </div>
100 </div>
101
102 <div class="col-lg-4 col-md-6 special-grid drinks">
103 <div class="gallery-single fix">
104 
105 <div class="formatotexto">
106 <h4>Cafés especiales</h4>
107 <p>Para el publico mas exigente y con un buen sabor y deleite</p>
108 <h5> $50-$100</h5>
109 </div>
110 </div>
111 </div>
112 </div>
113
```

En las siguientes líneas se tiene repetidamente las partes del menú, son llamadas a los estilos desde la carpeta css, para darles la forma que mas necesitan dentro del html. De igual manera de implementa un footer que es lo mismo que en el index.

Para la parte del css implementamos bootsrtrap para unas interacciones, animaciones, colores mas limpia.

```
cafeteria > css > # style.css > ...
1 |
2 @import url('https://fonts.googleapis.com/css?family=Rubik:300,400,500,700,900');
3
4
5 @import url(font-awesome.min.css);
6
7 body {
8   color: black;
9   font-size: 15px;
10  font-family: 'Rubik', sans-serif;
11  line-height: 1.80857;
12 }
13
14
15 a {
16   color: #1f1f1f;
17   text-decoration: none !important;
18   outline: none !important;
19   -webkit-transition: all .3s ease-in-out;
20   -moz-transition: all .3s ease-in-out;
21   -ms-transition: all .3s ease-in-out;
22   -o-transition: all .3s ease-in-out;
23   transition: all .3s ease-in-out;
24 }
25
26 h1,
27 h2, {
28   letter-spacing: 0;
29   font-weight: normal;
30   position: relative;
31   padding: 0 0 10px 0;
32   font-weight: normal;
33   line-height: 120% !important;
34   color: #1f1f1f;
35   margin: 0
36 }
37
38 h1 {
39   font-size: 24px
40 }
41
42 h2 {
43   font-size: 22px
44 }
45
46 h1 a,
47 h2 a {
48   color: #212121;
49   text-decoration: none !important;
50   opacity: 1
51 }
52
```

Se ocupo una fuente de Google para tener un estilo diferente invocamos para poder darle un poco mas de estilo así también de diferente formato. Indicamos como hace la función en el los parámetros que tenemos dentro de los html


```
index.html menu.html # style.css 9+ X
cafeteria > css > # style.css > ...
61 outline: none;
62 }
63
64 a,
65 .btn {
66     text-decoration: none !important;
67     outline: none !important;
68     -webkit-transition: all .3s ease-in-out;
69     -moz-transition: all .3s ease-in-out;
70     -ms-transition: all .3s ease-in-out;
71     -o-transition: all .3s ease-in-out;
72     transition: all .3s ease-in-out;
73 }
74
75 .btn-custom {
76     margin-top: 20px;
77     background-color: transparent !important;
78     border: 2px solid #ddd;
79     padding: 12px 40px;
80     font-size: 16px;
81 }
82
83 .lead {
84     font-size: 18px;
85     line-height: 30px;
86     color: #767676;
87     margin: 0;
88     padding: 0;
89 }
90
91 blockquote {
92     margin: 20px 0 20px;
93     padding: 30px;
94 }
95
96 ul, li, ol {
97     list-style: none;
98     margin: 0px;
99     padding: 0px;
100 }
101 button:focus {
102     outline: none;
103 }
104
105 .form-control::-moz-placeholder {
106     color: #2a2a2a;
107     opacity: 1;
108 }
109
110 .top-navbar {
111     position: relative;
112     z-index: 10;
```

Apartado donde podemos ver el formato de margen, tamaño y color que pueden tener las diferentes partes, algunas son parte del formato de bootstrap y para no perder el seguimiento de estos los dejamos en ingles en caso de que necesitésemos buscar alguna parte dentro de esto.

```
cafeteria > css > # style.css > ...
114
115 .top-navbar .navbar{
116     padding: 15px 0px;
117     position: fixed;
118     width: 100%;
119     border-radius: 0px 0px 100px 100px;
120     transition: height .3s ease-out, background .3s ease-out, box-shadow .3s ease-out;
121     -webkit-transform: translate3d(0, 0, 0);
122     transform: translate3d(0, 0, 0);
123     -webkit-box-shadow: 0 8px 6px -6px rgba(0, 0, 0, 0.4);
124     box-shadow: 0 8px 6px -6px rgba(0, 0, 0, 0.4);
125     z-index: 100;
126 }
127
128 .top-navbar .navbar-light .navbar-nav .nav-link{
129     color: orange;
130     font-size: 16px;
131     padding: 10px 20px;
132     border-radius: 4px;
133     text-transform: uppercase;
134 }
135
136 .top-navbar .navbar-light .navbar-nav .nav-item .nav-link:hover{
137     background: #eb8212;
138     color: #ffffff;
139 }
140 .top-navbar .navbar-light .navbar-nav .nav-item.active .nav-link{
141     background: #eb8212;
142     color: #ffffff;
143     border-radius: 4px;
144 }
145
146 .navbar-expand-lg .navbar-nav .dropdown-menu{
147     border: none;
148     border-radius: 0px;
149     padding: 10px;
150     box-shadow: 2px 5px 6px rgba(0,0,0,0.5);
151 }
152 .navbar-expand-lg .navbar-nav .dropdown-menu a{
153     padding: 10px 10px;
154     text-transform: uppercase;
155 }
156
157
158 .cover-slides{
159     height: 100vh;
160 }
161 .slides-navigation a {
162     background: #d65106;
163     position: absolute;
164     height: 50px;
```

Aquí podemos tener la forma que bootstrap tiene como intención en gran parte del nav ya que fue necesaria una guía para la interacción.

```
<> index.html <> menu.html # style.css 9+ X
cafeteria > css > # style.css > ...
156
157
158  .cover-slides{
159      height: 100vh;
160  }
161  .slides-navigation a {
162      background: #d65106;
163      position: absolute;
164      height: 50px;
165      width: 50px;
166      top: 50%;
167      font-size: 20px;
168      display: block;
169      color: #fff;
170      border-radius: 4px;
171      line-height: 63px;
172      text-align: center;
173      transition: all .3s ease-in-out;
174  }
175  .slides-navigation a i{
176      font-size: 40px;
177  }
178  .slides-navigation a:hover {
179      background: #010101;
180  }
181  .cover-slides .container{
182      height: 100%;
183      position: relative;
184      z-index: 2;
185  }
186  .cover-slides .container > .row {
187      -webkit-box-align: center;
188      -ms-flex-align: center;
189      align-items: center;
190  }
191  .cover-slides .container > .row {
192      height: 100%;
193  }
194
195  .btn{
196      text-transform: uppercase;
197      padding: 19px 36px;
198  }
199  .btn{
200      display: inline-block;
201      font-weight: 600;
202      text-align: center;
203      white-space: nowrap;
204      vertical-align: middle;
205      -webkit-user-select: none;
206      -moz-user-select: none;
207      -ms-user-select: none;
```

Tenemos la función de como se hace la parte de los div con lo que se les puede dar un estilo particular a casa uno para que no pierdan forma sin importan la resolución del receptor.

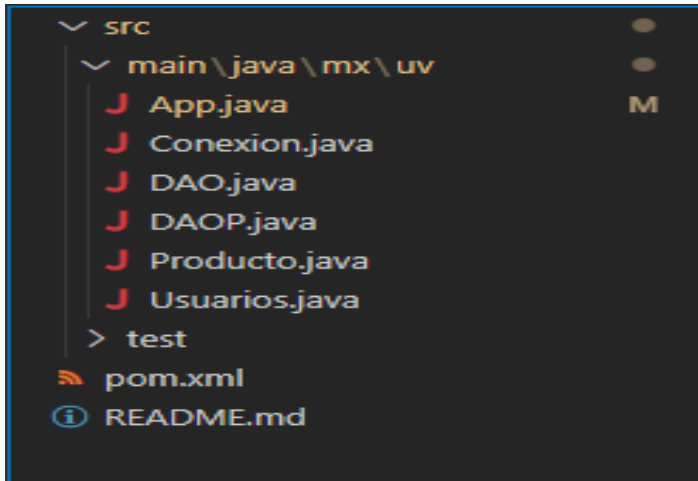

```

cafeteria > js > JS custom.js > <function> > Container.imagesLoaded() callback
4      $(window).on('load', function() {
5          $('.preloader').fadeOut();
6          $('#preloader').delay(550).fadeOut('slow');
7          $('body').delay(450).css({'overflow':'visible'});
8      });
9
10     $(window).on('scroll', function () {
11         if ($(window).scrollTop() > 50) {
12             $('.top-header').addClass('fixed-menu');
13         } else {
14             $('.top-header').removeClass('fixed-menu');
15         }
16     });
17
18     $('#slides').superslides({
19         inherit_width_from: '.cover-slides',
20         inherit_height_from: '.cover-slides',
21         play: 5000,
22         animation: 'fade',
23     });
24
25     $(".cover-slides ul li").append( "<div class='overlay-background'></div>" );
26
27     $(document).ready(function(){
28         $(window).on('scroll', function () {
29             if ($(this).scrollTop() > 100) {
30                 $('#back-to-top').fadeIn();
31             } else {
32                 $('#back-to-top').fadeOut();
33             }
34         });
35         $('#back-to-top').click(function(){
36             $("html, body").animate({ scrollTop: 0 }, 600);
37             return false;
38         });
39     });
40
41     var Container = $('.container');
42     Container.imagesLoaded(function () {
43         var portfolio = $('.special-menu');
44         portfolio.on('click', 'button', function () {
45             $(this).addClass('active').siblings().removeClass('active');
46             var filterValue = $(this).attr('data-filter');
47             $grid.isotope({
48                 filter: filterValue
49             });
50         });
51         var $grid = $('.special-list').isotope({
52             itemSelector: '.special-grid'
53         });

```

Dentro del js tenemos las funciones que se implementan dentro del body para poder hacer las funciones de interacciones dentro de los div el funcionamiento, lineamientos y limites que puede tener

Back-End



Tenemos que empezar con las configuraciones que debe tener el pom para que pueda tener las interacciones correctas, debemos resaltar lo que es el heroku y también el apartado con mysql

```
<dependency>
  <groupId>com.sparkjava</groupId>
  <artifactId>spark-core</artifactId>
  <version>2.9.4</version>
</dependency>

<!-- json -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.6</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.28</version>
</dependency>

<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-simple</artifactId>
  <version>1.7.21</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.json/json -->
<dependency>
  <groupId>org.json</groupId>
  <artifactId>json</artifactId>
  <version>20220924</version>
</dependency>

</dependencies>
```

```

101
102
103     <plugin>
104         <groupId>com.heroku.sdk</groupId>
105         <artifactId>heroku-maven-plugin</artifactId>
106         <version>0.4.4</version>
107         <configuration>
108             <jdkVersion>1.8</jdkVersion>
109             <!-- Use your own application name -->
110             <appName>CafeSuS</appName>
111             <processTypes>
112                 <!-- Tell Heroku how to launch your application -->
113                 <!-- You might have to remove the ./ in front -->
114                 <web>java -jar ./target/CafeSuS-jar-with-dependencies.jar</web>
115             </processTypes>
116         </configuration>
117     </plugin>
118 </plugins>
119 </build>
120
121 </project>

```

Tenemos las dependencias para hacer las conexiones con el servidor y el cómo va a tomar la interacción de java con la base de datos

Dentro de las configuraciones de los .java tenemos el app.java que tiene la mayor parte de la configuración conto con el DAO

Antes de todo esto tenemos que saber que hacen aparición los cors para que haga el puerto con heroku

```

1  package mx.uv;
2
3  import static spark.Spark.*;
4
5  import java.util.UUID;
6
7  import com.google.gson.*;
8
9  public class App
10 {
11
12     public static Gson gson = new Gson();
13
14     /**
15      * @param args
16      */
17
18     public static void main( String[] args )
19     {
20         port(getHerokuAssignedPort());
21         // CORS
22         options("/*", (request, response) -> {
23             String accessControlRequestHeaders = request.headers("Access-Control-Request-Headers");
24             if (accessControlRequestHeaders != null) {
25                 response.header("Access-Control-Allow-Headers", accessControlRequestHeaders);
26             }
27             String accessControlRequestMethod = request.headers("Access-Control-Request-Method");
28             if (accessControlRequestMethod != null) {
29                 response.header("Access-Control-Allow-Methods", accessControlRequestMethod);
30             }
31             return "OK";
32         });
33         before((req, res) -> res.header("Access-Control-Allow-Origin", "*"));
34         before((req, res) -> res.type("application/Json"));
35
36
37         get("/usuario", (req, res)-> gson.toJson(DAO.listaUsuarios()));
38         get("/reservaciones", (req, res)-> gson.toJson(DAOR.listaReservacion()));
39         // get("/productos", (req, res)-> gson.toJson(DAOR.listaReservacion()));
40
41

```

```

41
42     post("/registro", (req, res) -> {
43         String registro = req.body();
44         String id = UUID.randomUUID().toString();
45         Usuarios u = gson.fromJson(registro, Usuarios.class);
46         u.setId(id);
47         return DAO.registroUsuario(u);
48     });
49
50     // post("/producto-log", (req, res) -> {
51     //     String producto = req.body();
52     //     JSONObject obj = new JSONObject(producto);
53     //     System.out.print(obj);
54     //     System.out.print(obj.get("nombre").toString() + "\n");
55     //     // String nombre = obj.getJSONObject("nombre").toString();
56     //     // System.out.print(obj);
57     //     System.out.println(producto);
58     //     Producto p = gson.fromJson(producto, Producto.class);
59     //     return DAOP.registroProducto(p);
60     // });
61
62     post("/reservacion-log", (req, res) -> {
63         String reservacion = req.body();
64         String id = UUID.randomUUID().toString();
65         Reservacion r = gson.fromJson(reservacion, Reservacion.class);
66         r.setId(id);
67         return DAOR.registroReservacion(r);
68     });
69
70     post("/login", (req, res)->{
71         String login = req.body();
72         Usuarios u = gson.fromJson(login, Usuarios.class);
73         JsonObject objetoJson = new JsonObject();
74         for (Usuarios xUsuario : DAO.listaUsuarios()) {
75             if (xUsuario.getUsuario().equals(u.getUsuario())) {
76                 if (xUsuario.getPassword().equals(u.getPassword())) {
77                     objetoJson.addProperty("status", true);
78                     return objetoJson;
79                 }
80             }
81         }
82         objetoJson.addProperty("status", false);
83         return objetoJson;
84     });
85

```

```

87
88     static int getHerokuAssignedPort() {
89         ProcessBuilder processBuilder = new ProcessBuilder();
90         if (processBuilder.environment().get("PORT") != null) {
91             return Integer.parseInt(processBuilder.environment().get("PORT"));
92         }
93         return 80;
94     }
95

```

Tenemos las conexiones dentro de un html que contiene el login de la página principal las conexiones tienen su sección, se tienen precisamente los .log para lograr las conexiones y el login de los usuarios se hacen las conversiones al string y posteriormente llamamos al heroku. Y corra sobre el puerto 80:

Conexión.java

Se hace la conexión con la base de datos mysql

```
1  package mx.uv;
2
3  import java.sql.Connection;
4  import java.sql.DriverManager;
5  import java.sql.SQLException;
6
7  public class Conexion {
8
9      private static ProcessBuilder processBuilder = new ProcessBuilder();
10
11     private static String Driver = "com.mysql.cj.jdbc.Driver";
12     private static String url = "jdbc:mysql://db4free.net:3306/tablaparausuario";
13     private static String user= processBuilder.environment().get("USERDB");
14     private static String password= processBuilder.environment().get("PASSDB");
15
16     public Connection con;
17
18     public Connection getConnection() {
19         try {
20             Class.forName(Driver);
21             con = DriverManager.getConnection(url, user, password);
22         } catch (SQLException e) {
23             //System.out.println(e);
24         } catch (ClassNotFoundException e){
25             //System.out.println("no se encontro");
26         }
27         return con;
28     }
29 }
```


DAO.java

Tiene las acciones que va a llevar el login, para que sean interpretadas por el java con las conversiones y enviarlas a la base de datos

```
1  package mx.uv;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.sql.Statement;
8  import java.util.ArrayList;
9  import java.util.List;
10
11 public class DAO {
12     public static Conexion con = new Conexion();
13
14     public static String registroUsuario(Usuarios u) {
15         PreparedStatement stm = null;
16         Connection c = null;
17         String msj = null;
18
19         c = con.getConnection();
20         try {
21             String sql = "Insert into clientes (id, usuario, password, nombre, mail) values (?, ?, ?, ?, ?)";
22             stm = (PreparedStatement) c.prepareStatement(sql);
23             stm.setString(1, u.getId());
24             stm.setString(2, u.getUsuario());
25             stm.setString(3, u.getPassword());
26             stm.setString(4, u.getNombre());
27             stm.setString(5, u.getCorreo());
28
29         } catch (Exception e) {
30             System.out.print(e);
31         } finally{
32             if (stm != null) {
33                 try {
34                     stm.close();
35                 } catch (SQLException sqlEx) {
36                     sqlEx.printStackTrace();
37                 }
38                 stm = null;
39             }
40             try {
41                 c.close();
42                 System.out.println("Conexion Cerrada!");
43             } catch (SQLException sqlEx) {
44                 sqlEx.printStackTrace();
45             }
46         }
47         return msj;
48     }
49 }
```

```

49
50 public static List<Usuarios> listaUsuarios() {
51     Statement stm = null;
52     ResultSet rs = null;
53     Connection conn = null;
54     List<Usuarios> resultado = new ArrayList<>();
55
56     conn = con.getConnection();
57
58     try {
59         String sql = "SELECT * from clientes";
60         stm = (Statement) conn.createStatement();
61         rs = stm.executeQuery(sql);
62         while (rs.next()) {
63             Usuarios u = new Usuarios(rs.getString("id"), rs.getString("usuario"), rs.getString("password"),rs.getString("nombre"),rs.getString("mail"));
64             resultado.add(u);
65         }
66     } catch (Exception e) {
67         System.out.println(e);
68     } finally {
69         if (rs != null)
70             try {
71                 rs.close();
72             } catch (SQLException e) {
73                 System.out.println(e);
74             }
75         rs = null;
76         if (stm != null) {
77             try {
78                 stm.close();
79             } catch (Exception e) {
80                 System.out.println(e);
81             }
82             stm = null;
83         }
84         try {
85             conn.close();
86         } catch (Exception e) {
87             System.out.println(e);
88         }
89     }
90
91     return resultado;
92 }

```

DAOR.java

Lo mismo que dao pero para el registro del usuario y la base de datos

```
1 package mx.uv;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.ArrayList;
9 import java.util.List;
10
11 public class DAOR {
12     public static Conexion con = new Conexion();
13
14     public static String registroReservacion(Reservacion r) {
15         PreparedStatement stm = null;
16         Connection c = null;
17         String msj = null;
18
19         c = con.getConnection();
20         try {
21             String sql = "Insert into reservacion (id, fecha, hora, personas, nombre, correo, telefono) values (?, ?, ?, ?, ?, ?, ?)";
22             stm = (PreparedStatement) c.prepareStatement(sql);
23             stm.setString(1, r.getId());
24             stm.setString(2, r.getFecha());
25             stm.setString(3, r.getHora());
26             stm.setString(4, r.getPersona());
27             stm.setString(5, r.getNombre());
28             stm.setString(6, r.getCorreo());
29             stm.setString(7, r.getTelefono());
30
31             if (stm.executeUpdate() > 0)
32                 msj = "reservacion Agregado";
33             else
34                 msj = "no agregado";
35         } catch (Exception e) {
36             System.out.print(e);
37         } finally{
38             if (stm != null) {
39                 try {
40                     stm.close();
41                 } catch (SQLException sqlEx) {
42                     sqlEx.printStackTrace();
43                 }
44                 stm = null;
45             }
46             try {
47                 c.close();
48                 System.out.println("Conexion Cerrada!");
49             } catch (SQLException sqlEx) {
50                 sqlEx.printStackTrace();
51             }
52         }
53         return msj;
54     }
55 }
```