

SISTEMAS WEB

RAMIREZ GONZALEZ ALFREDO ISSAC

ZARATE RODRIGUEZ ANDRES



Principal (index.html)

```
1 <!DOCTYPE HTML>
2 <html Lang="es">
3   <!--Estilos propios y bootstrap, javascript, cabecita de la pagina-->
4   <head>
5     <title>Inicio - UVRSVR</title>
6     <!--Metadatos para la visualizacion-->
7     <meta charset="UTF-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9     <!--Estilos-->
10    <link rel="stylesheet" type="text/css" href="js/alertify/css/alertify.min.css"/>
11    <link rel="stylesheet" type="text/css" href="css/estylereser.css"/>
12    <link rel="stylesheet" type="text/css" href="js/alertify/css/themes/default.min.css"/>
13    <style>
14      html{font-size:16pt;}
15      .fondo {background: #0001E6;}
16      .white{color: #ffff;}
17      .negro{color: #000;}
18      .fondito {background: #020818;}
19      .borde{border: 1px #000 solid;}
20      .texto{ padding-top:5px;
21      margin-left:50px;}
22      .fondas {background: #142E45;}
23      .fondotitle{ background:url(img/Office.jpg);
24      height:100vh;
25      background-size:cover;
26      background-position:center;
27    }
28  </style>
29  <!--Alertify-->
30  <script src="js/jquery-3.3.1.min.js"></script>
31  <script src="js/alertify/alertify.min.js"></script>
32  <!--Jquery y el bootstrap en línea por si acaso-->
33  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEENzmM7KCKR/rE9/QpG6aAZGJwFDMVNA/GpGFF93hXp"
34  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQs
35  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFW
36
37  <script>
38    window.onload = function(){
39      alertify.success('Bienvenido');
40    }
41  </script>
```

Para la pantalla principal dentro de ‘head’ se cuentan con todos los links utilizados para dar forma a la página, sean desde estilos sean como JQuery y Bootstrap en línea, separados por diversos ‘div’

```
48 <link rel="stylesheet" type="text/css" href="css-index/bootstrap.min.css">
49 <link rel="shortcut icon" type="image/png" href="favicon.png" />
50 </head>
51
52 <body style="background-color: #00001D;">
53   <!--Parte principal de la pagina con la imagen-->
54   <section class="container-fluid d-flex fondotitle justify-content-center align-items-center">
55     <h1 class="display-1 black">UVRSVR</h1>
56   </section>
57   <!--Menu expandible responsivo-->
58   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
59     <a class="navbar-brand" href="#">UVRSVR</a>
60     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup"
61     <span class="navbar-toggler-icon"></span>
62   </button>
63   <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
64     <div class="navbar-nav">
65       <a class="nav-item nav-link active" href="index.html">Inicio<span class="sr-only">(current)</span></a>
66       <a class="nav-item nav-link" href="nota.html">Crear Nota</a>
67       <a class="nav-item nav-link" href="salones.html">Salones</a>
68     </div>
69   </div>
70 </nav>
71   <!-- Parte centrada principal de la pagina-->
72   <div class="container">
73     <div class="row">
74       <div class="col-sm-6 col-md-4 col-sg-4 fondo text-white-50 contenedor-logo-padre">
75         <div class="contenedor-logo">
76           
77         </div>
78       </div>
79       <div class="col-sm-6 col-md-8 col-sg-8 fondos">
80         <p class="text-justify white">En este espacio virtual puedes interactuar con otros miembros de la universidad.<br><br> Da a conocer q
81         <div class="embed-responsive embed-responsive-16by9">
82           <!--Insertar aquí un video explicando los motivos de este proyecto entre los 2 miembros del equipo-->
83           <video class="text-to embed-responsive-item" src="vid/video.mp4" controls width="80%" height="80%"></video></p>
84         </div>
85       </div>
86     </div>
87     <div class="text-white-50" href="presText.php">Ver presentación Textual</div>
88   </div>
```

Continuando con parte de la página principal donde se encuentra el menú expandible responsivo y uno de los container centrales donde se encuentra insertado un video explicativo acerca de la página.

```
index.html > html > head > style > .borde
90 </div>
91
92 <footer class="page-footer font-small pt-4 mt-4 fondito">
93 <!-- Footer Links -->
94 <div class="container-fluid text-center text-md-left">
95 <!-- Grid row -->
96 <div class="row">
97 <!-- Grid column -->
98 <div class="col-md-6 mt-md-0 mt-3">
99 <!-- Content -->
100 <h5 class="text-uppercase">Información</h5>
101 <ul class="list-unstyled text-white-50">
102 <li>Docente: Jose Rafael Rojano Caceres</li>
103 <li>Experiencia Educativa: Tecnologías WEB</li>
104 <li>Integrantes: Alfredo Issac Ramirez Gonzalez, Andrés Zárate Rodriguez</li>
105 <li>Carrera: Tecnologías Computacionales</li>
106 <li>Universidad Veracruzana. Facultad de Estadística e Informática. 2022</li>
107 </ul>
108 </div>
109 <!-- Grid column -->
110 <hr class="clearfix w-100 d-md-none pb-3">
111 <!-- Grid column -->
112 <div class="col-md-6 mb-md-0 mb-3">
113 <!-- Links -->
114 <h5 class="text-uppercase">Contáctanos</h5>
115 <ul class="list-unstyled text-white-50">
116 <li>Número: 228XXXXXXX</li>
117 <li>Correos: zs20018225@estudiantes.uv.mx </li>
118 </ul>
119 </div>
120 <!-- Grid column -->
121 </div>
122 <!-- Grid row -->
123 </div>
124 <!-- Footer Links -->
125 <!-- Copyright -->
126 <div class="footer-copyright text-center py-3">2022
127 <!-- <a class="cierra" href="cerrarSes.php">Cerrar Sesión</a> -->
128 </div>
129 <!-- Copyright -->
130 </footer>
```

Para la parte inferior de la página se implementó un ‘footer’, ‘div’, ‘h5’, ‘ul’, para la sección de información

Sección (Nota.html)

```

67 <div class="container">
68   <div class="row">
69     <div class="d-none contenedor-logo-padre d-sm-block col-md-12 col-xl-4 fondo text-white-50">
70       <div class="contenedor-logo d-none d-sm-block">
71         
72       </div>
73     </div>
74     <div class="col-sm-12 col-md-12 col-sg-8 col-xl-8 fondos">
75       <form class="form-register" action="" method="post" enctype="multipart/form-data">
76         <h2>Crear Nota</h2>
77         <div class="contenedor-inputs">
78           <p class="prro">Sus datos</p>
79
80           <input id="idNombre" type="text" name="idNombre" autocomplete="off" placeholder="Nombre" class="input-30" size="60" maxlength="6">
81           <input id="idMatricula" type="text" name="idMatricula" autocomplete="off" placeholder="Matricula" class="input-30" size="60" max
82           <select name="accion" id="accion" class="input-30b">
83             <option selected value="0">Accion</option>
84             <option value="Salicitar">Sollicitar</option>
85             <option value="Desocupar">Desocupar</option>
86           </select>
87
88           <p class="prro">Información aula</p>
89
90           <label>Salon</label>
91           <label>Hora Actual</label> <br/>
92           <select name="salon" id="salon" class="input-30b">
93             <option selected value="0">Salon</option>
94             <option value="105">105</option>
95             <option value="106">106</option>
96             <option value="205">205</option>
97             <option value="206">206</option>
98           </select>
99           <!-- <input type="text" id="salon" name="salon" autocomplete="off" class="input-30" size="60" placeholder="Salon o Módulo"/> -->
100           <label id="horaActual"></label>
101           <script type="text/javascript">
102             var today = new Date();
103             var day = today.getDay();
104             var daylist = ["Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado"];
105             var date = today.getFullYear()+'-' + (today.getMonth()+1)+'-' + today.getDate();
106             var time = today.getHours() + ":" + today.getMinutes() + ":" + today.getSeconds();
107             var datetime = date + ' ' + time;
108             document.getElementById("horaActual").innerHTML = datetime + ' <br/> Dia : ' + daylist[day];
109           </script>
110           <p class="prro">Nota</p>
111
112           <input id="nota" type="text" name="nota" autocomplete="off" placeholder="Agregue información adicional útil" class="input-50" si
113           <input type="submit" value="Guardar" name="BtnGuardar" class="btn-enviar"/>
114         </div>
115       </form>
116     </div>
117   </div>
118 </div>
```

Se hace uso de la misma plantilla de principal (index.html), implementando la sección ‘crear nota’

```

67 <div class="container">
68   <div class="row">
69     <div class="d-none contenedor-logo-padre d-sm-block col-md-12 col-xl-4 fondo text-white-50">
70       <div class="contenedor-logo d-none d-sm-block">
71         
72       </div>
73     </div>
74     <div class="col-sm-12 col-md-12 col-sg-8 col-xl-8 fondos">
75       <form class="form-register" action="" method="post" enctype="multipart/form-data">
76         <h2>Crear Nota</h2>
77         <div class="contenedor-inputs">
78           <p class="prro">Sus datos</p>
79
80           <input id="idNombre" type="text" name="idNombre" autocomplete="off" placeholder="Nombre" class="input-30" size="60" maxlength="6">
81           <input id="idMatricula" type="text" name="idMatricula" autocomplete="off" placeholder="Matricula" class="input-30" size="60" max
82           <select name="accion" id="accion" class="input-30b">
83             <option selected value="0">Accion</option>
84             <option value="Salicitar">Sollicitar</option>
85             <option value="Desocupar">Desocupar</option>
86           </select>
87
88           <p class="prro">Información aula</p>
89
90           <label>Salon</label>
91           <label>Hora Actual</label> <br/>
92           <select name="salon" id="salon" class="input-30b">
93             <option selected value="0">Salon</option>
94             <option value="105">105</option>
95             <option value="106">106</option>
96             <option value="205">205</option>
97             <option value="206">206</option>
98           </select>
99           <!-- <input type="text" id="salon" name="salon" autocomplete="off" class="input-30" size="60" placeholder="Salon o Módulo"/> -->
100           <label id="horaActual"></label>
101           <script type="text/javascript">
102             var today = new Date();
103             var day = today.getDay();
104             var daylist = ["Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes", "Sabado"];
105             var date = today.getFullYear()+'-' + (today.getMonth()+1)+'-' + today.getDate();
106             var time = today.getHours() + ":" + today.getMinutes() + ":" + today.getSeconds();
107             var datetime = date + ' ' + time;
108             document.getElementById("horaActual").innerHTML = datetime + ' <br/> Dia : ' + daylist[day];
109           </script>
110           <p class="prro">Nota</p>
111
112           <input id="nota" type="text" name="nota" autocomplete="off" placeholder="Agregue información adicional útil" class="input-50" si
113           <input type="submit" value="Guardar" name="BtnGuardar" class="btn-enviar"/>
114         </div>
115       </form>
116     </div>
117   </div>
118 </div>
```

Se encontrará la creación y asignación de identificadores para la creación de una nota

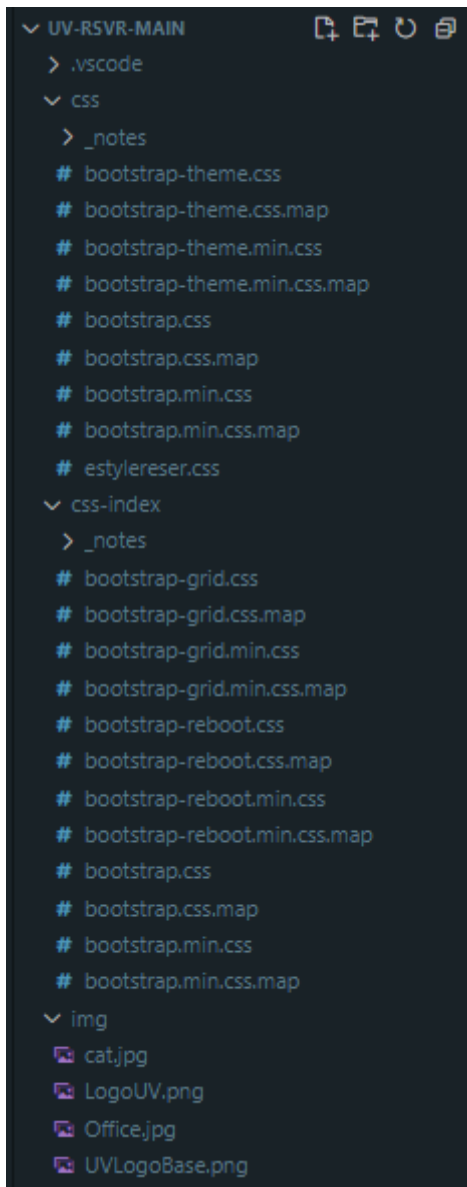
Sección (salones.html)

```

1 <!-- salones.html -->
2 <!-- 1. Sección de encabezado -->
3 <!-- 2. Sección de contenido -->
4 <!-- 3. Sección de pie de página -->
5
6 <div class="container">
7   <div class="row">
8     <div class="d-none contenedor-logo-padre d-sm-block col-md-12 col-xl-4 fondo text-white-50">
9       <div class="contenedor-logo d-none d-sm-block">
10        
11      </div>
12    </div>
13    <div class="col-sm-6 col-md-8 col-lg-8 fondos">
14      <table align="center" width="100%" bgcolor="#2082AA">
15        <tr>
16          <th colspan="2" style="text-align:center; color:white">Notas de los demas alumnos</th>
17        </tr>
18        <tr>
19          <th colspan="2" style="text-align:center; color:white">Nombre</th><th style="text-align:center"><font color="white">Fecha</font></th><th><font c
20        </tr>
21        <tr>
22          <th colspan="3" style="text-align:center; color:white">Ocupados</th>
23        </tr>
24        <tr>
25          <td><font color="white">X</font></td><td align="center"><font color="white">2022-06-05</font></td><td align="center"><fo
26        </tr>
27        <tr>
28          <td><font color="white">X</font></td><td align="center"><font color="white">2022-01-22</font></td><td align="center"><fo
29        </tr>
30        <tr>
31          <th colspan="3" style="text-align:center; color:white">Desocupados</th>
32        </tr>
33        <tr>
34          <td><font color="white">Z</font></td><td align="center"><font color="white">2022-04-03</font></td><td align="center"><f
35        </tr>
36        <tr>
37          <td><font color="white">Z</font></td><td align="center"><font color="white">2022-04-05</font></td><td align="center"><f
38        </tr>
39      </table>
40    </div>
41  </div>
42
43 <div class="page-footer font-small pt-4 mt-4 fondito">
44   <!-- Footer Links -->
45   <div class="container-fluid text-center text-md-left">
46     <!-- Grid row -->
47     <div class="row">
48

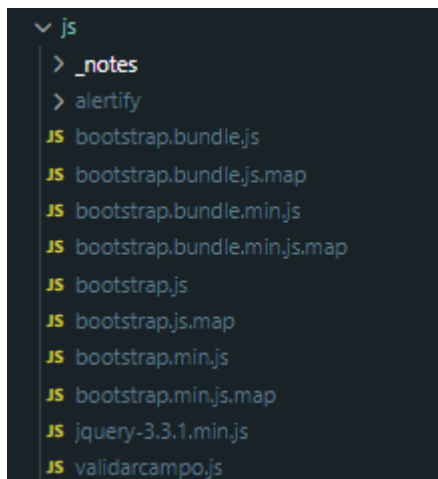
```

En la sección de salones se utiliza la misma platilla principal (index.html), implementando un ‘container’ para la representación de salones ocupados y desocupados.



Dentro del proyecto en la carpeta css se encuentran todos los estilos implementados, como también una carpeta con las imágenes usadas correspondientemente.

De ultimo una carpeta con todo los js implementados por Bootstrap y entre otros como de validación.



Conexión para la Base de Datos(Conexion.java)

```
AppDB > src > main > java > mx > equipo9uv > J Conexion.java > {} mx.equipo9uv
1 package mx.equipo9uv;
2
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Connection;
6 // import com.mysql.jdbc.Connection;
7
8 public class Conexion {
9     private static ProcessBuilder processBuilder = new ProcessBuilder();
10    //Si es local el string lleva en la primera seccion "localhost", pero en serv externo debe cambiarse
11    private static String url ="jdbc:mysql://127.0.0.1:3306/uvrsrvr";
12    private static String driverName = "com.mysql.jdbc.Driver";
13    //Estas variables del entorno donde se ejecute deben ser configuradas en la plataforma del despliegue
14    private static String username = processBuilder.environment().get(key: "USERDB");
15    private static String password = processBuilder.environment().get(key: "PASSDB");
16    //Variable de conexion
17    private static Connection connection = null;
18
19    public static Connection getConnection(){
20        try {
21            Class.forName(driverName);
22            connection = (Connection) DriverManager.getConnection(url, username, password);
23            System.out.println(x: "Conexion exitosa con la base de datos");
24        } catch (SQLException e){
25            System.out.println("Error con la consulta" + e);
26        } catch (ClassNotFoundException e) {
27            System.out.println(x: "Se ha generado un problema con el driver");
28        }
29        return connection;
30    }
31 }
32
```

Para la conexión a la base de datos en la especificamos la ULR y las credenciales de acceso

```
1 private static String url ="jdbc:mysql://127.0.0.1:3306/uvrsrvr";
2 private static String driverName = "com.mysql.jdbc.Driver";
3 //Estas variables del entorno donde se ejecute deben ser configuradas en la plataforma del despliegue
4 private static String username = processBuilder.environment().get(key: "USERDB");
5 private static String password = processBuilder.environment().get(key: "PASSDB");
6 //Variable de conexion
```

DAO.java

```
14 //
15 public class DAO {
16     //Conectar internamente con la base de datos
17     private static Conexion c = new Conexion();
18
19     public static List<Nota> traerNotas(){
20         Statement consulta = null;
21         ResultSet elementos = null;
22         Connection cc = null;
23         List<Nota> resultado = new ArrayList<>();
24
25         cc = c.getConnection();
26
27         try{
28             String sql = "SELECT * FROM nota";
29             consulta = (Statement) cc.createStatement();
30             elementos = consulta.executeQuery(sql);
31             while(elementos.next()){
32                 Nota n = new Nota(elementos.getString(columnLabel: "id"), elementos.getString(columnLabel: "nombre"), elementos.getString(columnLabel: "descripcion"));
33                 resultado.add(n);
34             }
35         } catch (Exception e) {
36             System.out.println(e);
37         } finally {
38             if (elementos != null) {
39                 try {
40                     elementos.close();
41                 } catch (SQLException sqlEx) {
42                     sqlEx.printStackTrace();
43                 }
44                 elementos = null;
45             }
46
47             if (consulta != null) {
48                 try {
49                     consulta.close();
50                 } catch (SQLException sqlEx) {
51                     sqlEx.printStackTrace();
52                 }
53                 consulta = null;
54             }
55         }
56     }
57 }
```

```
17     private static Conexion c = new Conexion();
18
19     public static List<Nota> traerNotas(){
20         Statement consulta = null;
21         ResultSet elementos = null;
22         Connection cc = null;
23         List<Nota> resultado = new ArrayList<>();
24
25         cc = c.getConnection();
```

Nos conectamos directamente a la base de datos y un 'ArrayList' de tipo nota

```
try{
    String sql = "SELECT * FROM nota";
    consulta = (Statement) cc.createStatement();
    elementos = consulta.executeQuery(sql);
    while(elementos.next()){
        Nota n = new Nota(elementos.getString(columnLabel: "id"), elementos.getString(columnLabel: "nombre"), elementos.getString(columnLabel: "descripcion"));
        resultado.add(n);
    }
} catch (Exception e) {
    System.out.println(e);
} finally {
    if (elementos != null) {
        try {
            elementos.close();
        } catch (SQLException sqlEx) {
            sqlEx.printStackTrace();
        }
        elementos = null;
    }

    if (consulta != null) {
        try {
            consulta.close();
        } catch (SQLException sqlEx) {
            sqlEx.printStackTrace();
        }
        consulta = null;
    }
}
try {
```

El siguiente apartado de código nos servirá para traer todas las notas que se encuentren, recuperando cada atributo de nota en la columna. Se implementa un 'try' 'catch' para el manejo de las excepciones al consultar los elementos de la tabla notas


```

AppDB > src > main > java > mx > equipo9uv > DAO.java > {} mx.equipo9uv
65
66 ic static String crearNota(Nota n) {
67     PreparedStatement consulta = null;
68     Connection cc = null;
69     String msj = "";
70
71     cc = c.getConnection();
72
73     try {
74         String sql = "insert into usuarios (id, nombre, matricula, accion, fechahora, salon, nota) values (?, ?, ?, ?, ?, ?, ?)";
75         consulta = (PreparedStatement) cc.prepareStatement(sql);
76         consulta.setString(parameterIndex: 1, n.getId());
77         consulta.setString(parameterIndex: 2, n.getNombre());
78         consulta.setString(parameterIndex: 3, n.getMatricula());
79         consulta.setString(parameterIndex: 4, n.getAccion());
80         consulta.setString(parameterIndex: 5, n.getFechahora());
81         consulta.setString(parameterIndex: 6, n.getSalon());
82         consulta.setString(parameterIndex: 7, n.getNota());
83
84         if (consulta.executeUpdate() > 0)
85             msj = "La nota se agrego";
86         else
87             msj = "La nota valio mauser";
88     } catch (Exception e) {
89         System.out.println(e);
90     } finally {
91         if (consulta != null) {
92             try {
93                 consulta.close();
94             } catch (SQLException sqlEx) {
95                 sqlEx.printStackTrace();
96             }
97             consulta = null;
98         }
99         try {
100             cc.close();
101             System.out.println("Closed connection!");
102         } catch (SQLException sqlEx) {
103             sqlEx.printStackTrace();
104         }
105     }
106     return msj;

```

En este apartado creamos una nota insertando 'insert into usuarios' en usuarios cada parámetro correspondiente

```

73     try {
74         String sql = "insert into usuarios (id, nombre, matricula, accion, fechahora, salon, nota) values (?, ?, ?, ?, ?, ?, ?)";
75         consulta = (PreparedStatement) cc.prepareStatement(sql);
76         consulta.setString(parameterIndex: 1, n.getId());
77         consulta.setString(parameterIndex: 2, n.getNombre());
78         consulta.setString(parameterIndex: 3, n.getMatricula());
79         consulta.setString(parameterIndex: 4, n.getAccion());
80         consulta.setString(parameterIndex: 5, n.getFechahora());
81         consulta.setString(parameterIndex: 6, n.getSalon());
82         consulta.setString(parameterIndex: 7, n.getNota());
83     }

```

```

107
108
109 ic static String modificarNota(Nota n){
110     PreparedStatement consulta = null;
111     Connection cc = null;
112     String msj = "";
113
114     cc = c.getConnection();
115
116     try{
117         String sql = "UPDATE nota set nombre = ?, matricula = ?, accion = ?, fechahora = ?, salon = ?, nota = ? where id= ?";
118         consulta = (PreparedStatement) cc.prepareStatement(sql);
119         consulta.setString(parameterIndex: 1, n.getNombre());
120         consulta.setString(parameterIndex: 2, n.getMatricula());
121         consulta.setString(parameterIndex: 3, n.getAccion());
122         consulta.setString(parameterIndex: 4, n.getFechahora());
123         consulta.setString(parameterIndex: 5, n.getSalon());
124         consulta.setString(parameterIndex: 6, n.getNota());
125         consulta.setString(parameterIndex: 7, n.getId());
126
127         if (consulta.executeUpdate() > 0)
128             msj = "La nota se modifico";
129         else
130             msj = "La nota no se modifico";
131     } catch (Exception e) {
132         System.out.println(e);
133     }finally {
134         if (consulta != null) {
135             try {
136                 consulta.close();
137             } catch (SQLException sqlEx) {
138                 sqlEx.printStackTrace();
139             }
140             consulta = null;
141         }
142         try {
143             cc.close();
144             System.out.println("Closed connection!");
145         } catch (SQLException sqlEx) {
146             sqlEx.printStackTrace();
147         }
148     }
149 }

```

Modificar nota funciona de la misma manera recibiendo todos los parámetros, pero en este caso ya no estará insertando un registro si no que lo actualizara ‘UPDATE’ hacia todos los parámetros en nota

```

AppDB > src > main > java > mx > equipo9uv > J DAOJava > {} mx.equipo9uv
151
152 ic static String eliminarNota(Nota n){
153     PreparedStatement consulta = null;
154     Connection cc = null;
155     String msj = "";
156
157     cc = c.getConnection();
158
159     try{
160         String sql = "DELETE FROM nota WHERE id= ?";
161         consulta = (PreparedStatement) cc.prepareStatement(sql);
162         consulta.setString(parameterIndex: 1, n.getId());
163
164         if (consulta.executeUpdate() > 0)
165             msj = "Se arrugo la nota y se tiro en la cesta";
166         else
167             msj = "Error: La nota no alcanzo el bote de basura";
168     } catch (Exception e) {
169         System.out.println(e);
170     }finally {
171         if (consulta != null) {
172             try {
173                 consulta.close();
174             } catch (SQLException sqlEx) {
175                 sqlEx.printStackTrace();
176             }
177             consulta = null;
178         }
179         try {
180             cc.close();
181             System.out.println("Closed connection!");
182         } catch (SQLException sqlEx) {
183             sqlEx.printStackTrace();
184         }
185     }
186     return msj;
187 }

```

Para eliminar nota ‘DELETE FROM nota’ solo estará haciendo uso del id, donde el id sea igual al consultado estará siendo eliminado, implementando su ‘try’ ‘catch’ para las excepciones y un ‘if’ que nos estará comunicando con un ‘msj’ si la nota se logro eliminar exitosamente o ocurrió un error y en dado caso si la conexión se ha cerrado.

Nota.java

```
AppDB > src > main > java > mx > equipo9uv > J Nota.java > 📁 Nota > 📄 Nota(String, String, String, String, String, String)
1 package mx.equipo9uv;
2
3 import java.sql.Date;
4 import java.time.LocalDateTime;
5 import java.time.format.DateTimeFormatter;
6
7 public class Nota {
8     String id;
9     String nombre;
10    String matricula;
11    String accion;
12    String fechahora;
13    String salon;
14    String nota;
15
16    public String getId() {
17        return id;
18    }
19
20    public void setId(String id) {
21        this.id = id;
22    }
23
24    public String getNombre() {
25        return nombre;
26    }
27
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31
32    public String getMatricula() {
33        return matricula;
34    }
35
36    public void setMatricula(String matricula) {
37        this.matricula = matricula;
38    }
39
40    public String getAccion() {
41        return accion;
42    }
43
44    public void setAccion(String accion) {
45        this.accion = accion;
46    }
47
48    public String getFechahora() {
49        return fechahora;
50    }
51
52    public void setFechahora(String fechahora) {
53        this.fechahora = fechahora;
54    }
55
56    public String getSalon() {
57        return salon;
58    }
59
60    public void setSalon(String salon) {
61        this.salon = salon;
62    }
63
64    public String getNota() {
65        return nota;
66    }
67
68    public void setNota(String nota) {
69        this.nota = nota;
70    }
71
72    public void guardar() {
73        try {
74            String sql = "INSERT INTO nota (id, nombre, matricula, accion, fechahora, salon, nota) VALUES (?, ?, ?, ?, ?, ?, ?)";
75            PreparedStatement ps = con.prepareStatement(sql);
76            ps.setString(1, getId());
77            ps.setString(2, getNombre());
78            ps.setString(3, getMatricula());
79            ps.setString(4, getAccion());
80            ps.setString(5, getFechahora());
81            ps.setString(6, getSalon());
82            ps.setString(7, getNota());
83            ps.executeUpdate();
84        } catch (SQLException e) {
85            e.printStackTrace();
86        }
87    }
88
89    public void actualizar() {
90        try {
91            String sql = "UPDATE nota SET nombre = ?, matricula = ?, accion = ?, fechahora = ?, salon = ?, nota = ? WHERE id = ?";
92            PreparedStatement ps = con.prepareStatement(sql);
93            ps.setString(1, getNombre());
94            ps.setString(2, getMatricula());
95            ps.setString(3, getAccion());
96            ps.setString(4, getFechahora());
97            ps.setString(5, getSalon());
98            ps.setString(6, getNota());
99            ps.setString(7, getId());
100           ps.executeUpdate();
101       } catch (SQLException e) {
102           e.printStackTrace();
103       }
104   }
105
106   public void eliminar() {
107       try {
108           String sql = "DELETE FROM nota WHERE id = ?";
109           PreparedStatement ps = con.prepareStatement(sql);
110           ps.setString(1, getId());
111           ps.executeUpdate();
112       } catch (SQLException e) {
113           e.printStackTrace();
114       }
115   }
116
117   public void listar() {
118       try {
119           String sql = "SELECT * FROM nota";
120           PreparedStatement ps = con.prepareStatement(sql);
121           ResultSet rs = ps.executeQuery();
122           while (rs.next()) {
123               System.out.println("ID: " + rs.getString(1) + " Nombre: " + rs.getString(2) + " Matricula: " + rs.getString(3) + " Accion: " + rs.getString(4) + " Fechahora: " + rs.getString(5) + " Salon: " + rs.getString(6) + " Nota: " + rs.getString(7));
124           }
125       } catch (SQLException e) {
126           e.printStackTrace();
127       }
128   }
129
130   public void mostrarMenu() {
131       System.out.println("Menu de opciones:");
132       System.out.println("1. Guardar");
133       System.out.println("2. Actualizar");
134       System.out.println("3. Eliminar");
135       System.out.println("4. Listar");
136       System.out.println("5. Salir");
137   }
138
139   public void ejecutarMenu() {
140       while (true) {
141           mostrarMenu();
142           int opcion = Integer.parseInt(System.out.println("Ingrese una opcion:"));
143           switch (opcion) {
144               case 1:
145                   guardar();
146                   break;
147               case 2:
148                   actualizar();
149                   break;
150               case 3:
151                   eliminar();
152                   break;
153               case 4:
154                   listar();
155                   break;
156               case 5:
157                   System.out.println("Salir");
158                   break;
159               default:
160                   System.out.println("Opcion no valida");
161           }
162       }
163   }
164
165   public static void main(String[] args) {
166       Nota n = new Nota();
167       n.ejecutarMenu();
168   }
169 }
```

Aquí se estarán declarando todas las variables que utilizamos para que posterior mente hagan uso los métodos correspondientes,

```
//Constructor que pone la hora solito
public Nota(String id, String nombre, String matricula, String accion, String salon,
            String nota) {
    this.id = id;
    this.nombre = nombre;
    this.matricula = matricula;
    this.accion = accion;
    this.fechahora = generarHora();
    this.salon = salon;
    this.nota = nota;
}

//Constructor que toma la hora para crear el objeto del DAO
public Nota(String id, String nombre, String matricula, String accion, String fechahora, String salon,
            String nota) {
    this.id = id;
    this.nombre = nombre;
    this.matricula = matricula;
    this.accion = accion;
    this.fechahora = fechahora;
    this.salon = salon;
    this.nota = nota;
}
```

Se encuentran dos constructores para la asignación de hora automática y otro que lo tomara la hora y genera un objeto de 'DAO'

```

public String generarHora(){ //Este metodo en realidad trae la fecha y la hora
    LocalDateTime horaActual = LocalDateTime.now();
    DateTimeFormatter horaFormateada = DateTimeFormatter.ofPattern(pattern: "yyyy-MM-dd HH:mm:ss");
    String horaTransformada = horaActual.format(horaFormateada);
    return horaTransformada;
}

```

Método para recuperar la fecha y hora en su formato correspondiente.

```

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getMatricula() {
    return matricula;
}

public void setMatricula(String matricula) {
    this.matricula = matricula;
}

public String getAccion() {
    return accion;
}

public void setAccion(String accion) {
    this.accion = accion;
}

public String getFechahora() {
    return fechahora;
}

public void setFechahora(String fechahora) {
    this.fechahora = fechahora;
}

public String getSalon() {
    return salon;
}

public void setSalon(String salon) {
    this.salon = salon;
}

public String getNota() {
    return nota;
}

```

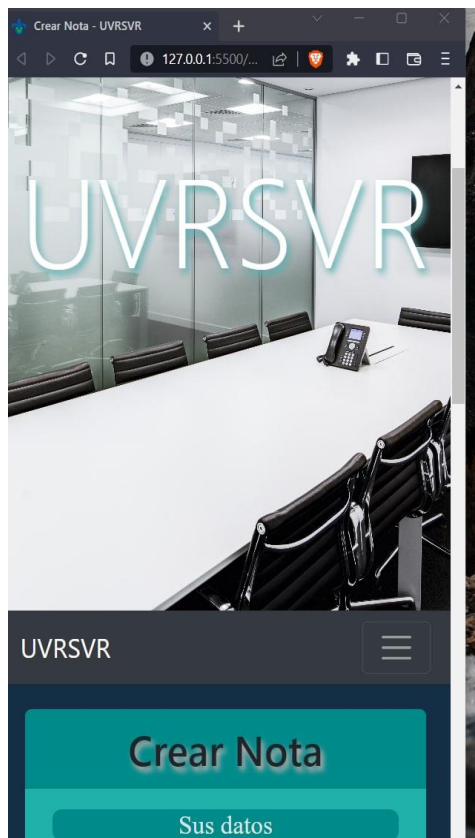
Correspondientes getter y setter

Accesibilidad

Escritorio

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5500/nota.html'. The page has a dark blue header with the text 'UVRSVR' and navigation links 'Inicio', 'Crear Nota', and 'Salones'. On the left, there is a vertical light blue bar containing a logo of a fleur-de-lis with a stylized 'V' inside. The main content area is teal and titled 'Crear Nota'. It contains several sections: 'Sus datos' with input fields for 'Nombre' and 'Matricula'; 'Accion' with a dropdown menu; 'Información aula' with a 'Salon' dropdown menu; 'Hora Actual' showing '2022-12-12 14:27:6'; 'Dia :Lunes'; and a 'Nota' input field at the bottom.

Movil



This screenshot provides a detailed view of the 'Crear Nota' form on a mobile device. The form is displayed on a teal background. It includes the following elements from top to bottom: a 'Sus datos' button; 'Nombre' and 'Matricula' input fields; an 'Accion' dropdown menu; an 'Información aula' button; a 'Salon' dropdown menu; 'Hora Actual' (2022-12-12 14:27:6) and 'Dia :Lunes'; a 'Nota' input field; an 'Ingresar solicitud' button; and a text input field labeled 'Agregue información adicional útil'.