

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 - UVRVR</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-KJ3o2DKtIkvYIK3UENzmM7KCKR/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-JZR6Spejh4U02d8jOt6vLEHfe/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">UVRVR</h1>
56   </section>
57   <!--Menu expandible responsivo-->
58   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
59     <a class="navbar-brand" href="#">UVRVR</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
100 </div>
101
102 <footer class="page-footer font-small pt-4 mt-4 fondito">
103 <!-- Footer Links -->
104 <div class="container-fluid text-center text-md-left">
105 <!-- Grid row -->
106 <div class="row">
107 <!-- Grid column -->
108 <div class="col-md-6 mt-md-0 mt-3">
109 <!-- Content -->
110 <h5 class="text-uppercase">Información</h5>
111 <ul class="list-unstyled text-white-50">
112 <li>Docente: Jose Rafael Rojano Caceres</li>
113 <li>Experiencia Educativa: Tecnologías WEB</li>
114 <li>Integrantes: Alfredo Issac Ramirez Gonzalez, Andrés Zárate Rodriguez</li>
115 <li>Carrera: Tecnologías Computacionales</li>
116 <li>Universidad Veracruzana. Facultad de Estadística e Informática. 2022</li>
117 </ul>
118 </div>
119 <!-- Grid column -->
120 <hr class="clearfix w-100 d-md-none pb-3">
121 <!-- Grid column -->
122 <div class="col-md-6 mb-md-0 mb-3">
123 <!-- Links -->
124 <h5 class="text-uppercase">Contáctanos</h5>
125 <ul class="list-unstyled text-white-50">
126 <li>Número: 228XXXXXXX</li>
127 <li>Correos: zs20018225@estudiantes.uv.mx </li>
128 </ul>
129 </div>
130 <!-- Grid column -->
131 </div>
132 <!-- Grid row -->
133 </div>
134 <!-- Footer Links -->
135 <!-- Copyright -->
136 <div class="footer-copyright text-center py-3">2022
137 <!-- <a class="cierra" href="cerrarSes.php">Cerrar Sesión</a> -->
138 </div>
139 <!-- Copyright -->
140 </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;

```

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

```

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>

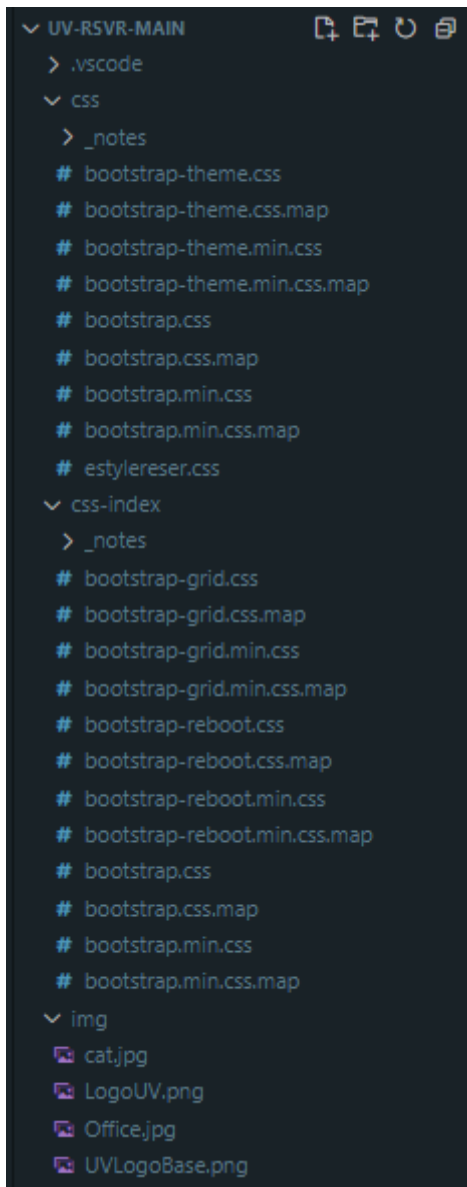
```

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

Sección (salones.html)

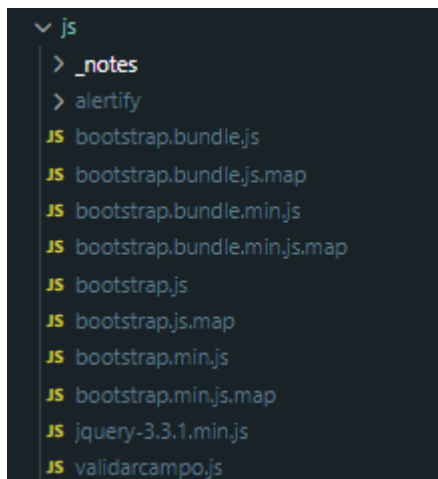
[illegible]

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     }

```

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

```
App06 > src > main > java > mx > equipo9uv > J Nota.java > ts Nota > @ Nota(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        guardar();
74    }
75
76    public void actualizar() {
77        actualizar();
78    }
79
80    public void eliminar() {
81        eliminar();
82    }
83
84    public void consultar() {
85        consultar();
86    }
87
88    public void listar() {
89        listar();
90    }
91
92    public void imprimir() {
93        imprimir();
94    }
95
96    public void mostrar() {
97        mostrar();
98    }
99
100    public void ocultar() {
101        ocultar();
102    }
103
104    public void resetear() {
105        resetear();
106    }
107
108    public void validar() {
109        validar();
110    }
111
112    public void verificar() {
113        verificar();
114    }
115
116    public void comprobar() {
117        comprobar();
118    }
119
120    public void confirmar() {
121        confirmar();
122    }
123
124    public void cancelar() {
125        cancelar();
126    }
127
128    public void abortar() {
129        abortar();
130    }
131
132    public void reiniciar() {
133        reiniciar();
134    }
135
136    public void restaurar() {
137        restaurar();
138    }
139
140    public void revertir() {
141        revertir();
142    }
143
144    public void deshacer() {
145        deshacer();
146    }
147
148    public void deshacer() {
149        deshacer();
150    }
151
152    public void deshacer() {
153        deshacer();
154    }
155
156    public void deshacer() {
157        deshacer();
158    }
159
160    public void deshacer() {
161        deshacer();
162    }
163
164    public void deshacer() {
165        deshacer();
166    }
167
168    public void deshacer() {
169        deshacer();
170    }
171
172    public void deshacer() {
173        deshacer();
174    }
175
176    public void deshacer() {
177        deshacer();
178    }
179
180    public void deshacer() {
181        deshacer();
182    }
183
184    public void deshacer() {
185        deshacer();
186    }
187
188    public void deshacer() {
189        deshacer();
190    }
191
192    public void deshacer() {
193        deshacer();
194    }
195
196    public void deshacer() {
197        deshacer();
198    }
199
200    public void deshacer() {
201        deshacer();
202    }
203
204    public void deshacer() {
205        deshacer();
206    }
207
208    public void deshacer() {
209        deshacer();
210    }
211
212    public void deshacer() {
213        deshacer();
214    }
215
216    public void deshacer() {
217        deshacer();
218    }
219
220    public void deshacer() {
221        deshacer();
222    }
223
224    public void deshacer() {
225        deshacer();
226    }
227
228    public void deshacer() {
229        deshacer();
230    }
231
232    public void deshacer() {
233        deshacer();
234    }
235
236    public void deshacer() {
237        deshacer();
238    }
239
240    public void deshacer() {
241        deshacer();
242    }
243
244    public void deshacer() {
245        deshacer();
246    }
247
248    public void deshacer() {
249        deshacer();
250    }
251
252    public void deshacer() {
253        deshacer();
254    }
255
256    public void deshacer() {
257        deshacer();
258    }
259
260    public void deshacer() {
261        deshacer();
262    }
263
264    public void deshacer() {
265        deshacer();
266    }
267
268    public void deshacer() {
269        deshacer();
270    }
271
272    public void deshacer() {
273        deshacer();
274    }
275
276    public void deshacer() {
277        deshacer();
278    }
279
280    public void deshacer() {
281        deshacer();
282    }
283
284    public void deshacer() {
285        deshacer();
286    }
287
288    public void deshacer() {
289        deshacer();
290    }
291
292    public void deshacer() {
293        deshacer();
294    }
295
296    public void deshacer() {
297        deshacer();
298    }
299
300    public void deshacer() {
301        deshacer();
302    }
303
304    public void deshacer() {
305        deshacer();
306    }
307
308    public void deshacer() {
309        deshacer();
310    }
311
312    public void deshacer() {
313        deshacer();
314    }
315
316    public void deshacer() {
317        deshacer();
318    }
319
320    public void deshacer() {
321        deshacer();
322    }
323
324    public void deshacer() {
325        deshacer();
326    }
327
328    public void deshacer() {
329        deshacer();
330    }
331
332    public void deshacer() {
333        deshacer();
334    }
335
336    public void deshacer() {
337        deshacer();
338    }
339
340    public void deshacer() {
341        deshacer();
342    }
343
344    public void deshacer() {
345        deshacer();
346    }
347
348    public void deshacer() {
349        deshacer();
350    }
351
352    public void deshacer() {
353        deshacer();
354    }
355
356    public void deshacer() {
357        deshacer();
358    }
359
360    public void deshacer() {
361        deshacer();
362    }
363
364    public void deshacer() {
365        deshacer();
366    }
367
368    public void deshacer() {
369        deshacer();
370    }
371
372    public void deshacer() {
373        deshacer();
374    }
375
376    public void deshacer() {
377        deshacer();
378    }
379
380    public void deshacer() {
381        deshacer();
382    }
383
384    public void deshacer() {
385        deshacer();
386    }
387
388    public void deshacer() {
389        deshacer();
390    }
391
392    public void deshacer() {
393        deshacer();
394    }
395
396    public void deshacer() {
397        deshacer();
398    }
399
400    public void deshacer() {
401        deshacer();
402    }
403
404    public void deshacer() {
405        deshacer();
406    }
407
408    public void deshacer() {
409        deshacer();
410    }
411
412    public void deshacer() {
413        deshacer();
414    }
415
416    public void deshacer() {
417        deshacer();
418    }
419
420    public void deshacer() {
421        deshacer();
422    }
423
424    public void deshacer() {
425        deshacer();
426    }
427
428    public void deshacer() {
429        deshacer();
430    }
431
432    public void deshacer() {
433        deshacer();
434    }
435
436    public void deshacer() {
437        deshacer();
438    }
439
440    public void deshacer() {
441        deshacer();
442    }
443
444    public void deshacer() {
445        deshacer();
446    }
447
448    public void deshacer() {
449        deshacer();
450    }
451
452    public void deshacer() {
453        deshacer();
454    }
455
456    public void deshacer() {
457        deshacer();
458    }
459
460    public void deshacer() {
461        deshacer();
462    }
463
464    public void deshacer() {
465        deshacer();
466    }
467
468    public void deshacer() {
469        deshacer();
470    }
471
472    public void deshacer() {
473        deshacer();
474    }
475
476    public void deshacer() {
477        deshacer();
478    }
479
480    public void deshacer() {
481        deshacer();
482    }
483
484    public void deshacer() {
485        deshacer();
486    }
487
488    public void deshacer() {
489        deshacer();
490    }
491
492    public void deshacer() {
493        deshacer();
494    }
495
496    public void deshacer() {
497        deshacer();
498    }
499
500    public void deshacer() {
501        deshacer();
502    }
503
504    public void deshacer() {
505        deshacer();
506    }
507
508    public void deshacer() {
509        deshacer();
510    }
511
512    public void deshacer() {
513        deshacer();
514    }
515
516    public void deshacer() {
517        deshacer();
518    }
519
520    public void deshacer() {
521        deshacer();
522    }
523
524    public void deshacer() {
525        deshacer();
526    }
527
528    public void deshacer() {
529        deshacer();
530    }
531
532    public void deshacer() {
533        deshacer();
534    }
535
536    public void deshacer() {
537        deshacer();
538    }
539
540    public void deshacer() {
541        deshacer();
542    }
543
544    public void deshacer() {
545        deshacer();
546    }
547
548    public void deshacer() {
549        deshacer();
550    }
551
552    public void deshacer() {
553        deshacer();
554    }
555
556    public void deshacer() {
557        deshacer();
558    }
559
560    public void deshacer() {
561        deshacer();
562    }
563
564    public void deshacer() {
565        deshacer();
566    }
```

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