



# **INSTITUTO POLITECNICO NACIONAL**

## **UNIDAD PROFESIONAL**

**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA Y  
CIENCIAS SOCIALES Y ADMINISTRATIVAS**

**INGENIERIA EN INFORMATICA**

**Programación Orientada a Objetos**

**SECUENCIA: 2NM42**

**Equipo 07**

**Porcentaje de Participación**

De Santiago Landeros Edwin Uriel	20 %
Reveles Ramirez Jesus Emilio	20 %
Villegas Nolasco Elide Yoltzin	20 %
Herrera Cano Edwin Rubén	20 %
Palmieri Mondragón Gerardo	20 %

**PROFESOR. Oviedo Galdeano Mario**

**Tercer Departamental: Calificaciones de un grupo**

**Fecha de Entrega**

México D. F. junio de 2022



## Tabla de Contenidos

<b>Introducción.....</b>	<b>3</b>
<b>Diagramas de Composición de Clases.....</b>	<b>4</b>
<b>Diagrama de Clases.....</b>	<b>5</b>
<b>Diagrama de Casos de Uso.....</b>	<b>7</b>
<b>Pseudocódigo de todos los métodos.....</b>	<b>8</b>
Clase Alumnos.java	
Clase Conectar.java	
Clase Interfaz.java	
Clase Registros.java	
<b>Implementación de todos los métodos.....</b>	<b>12</b>
<b>Tablas de la BD.....</b>	<b>17</b>
<b>Capturas de pantalla de pruebas.....</b>	<b>18</b>
<b>Comentarios Finales.....</b>	<b>22</b>

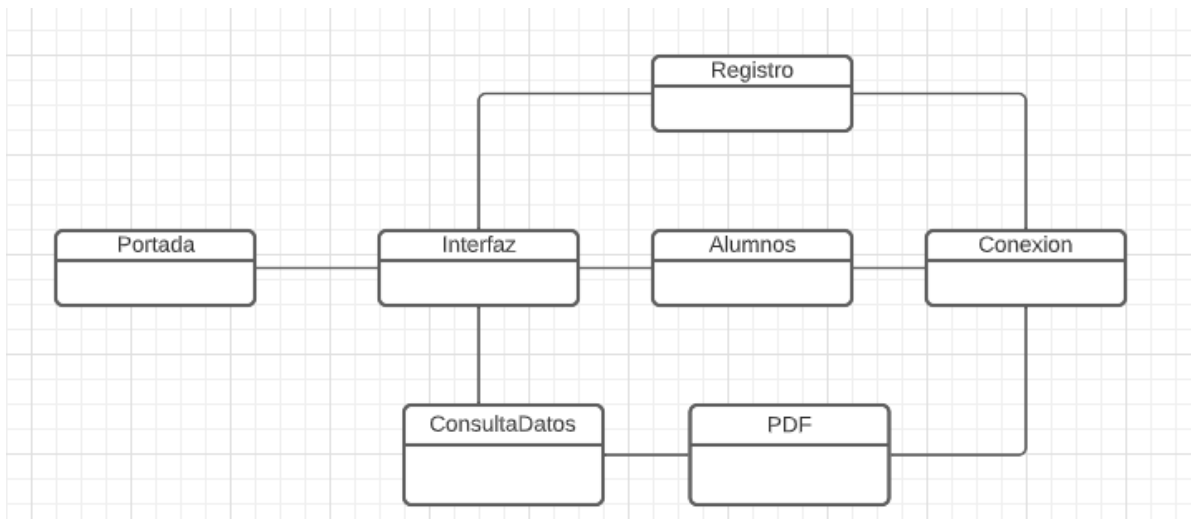
## **Introducción**

Para la ejecución de este proyecto, vamos a utilizar la herramienta de Netbeans, además de Php MyAdmin, y MySQL. En este se requiere una aplicación en Java con una GUI, para una base de datos, que permita consultar, eliminar, crear y actualizar una serie de registros.

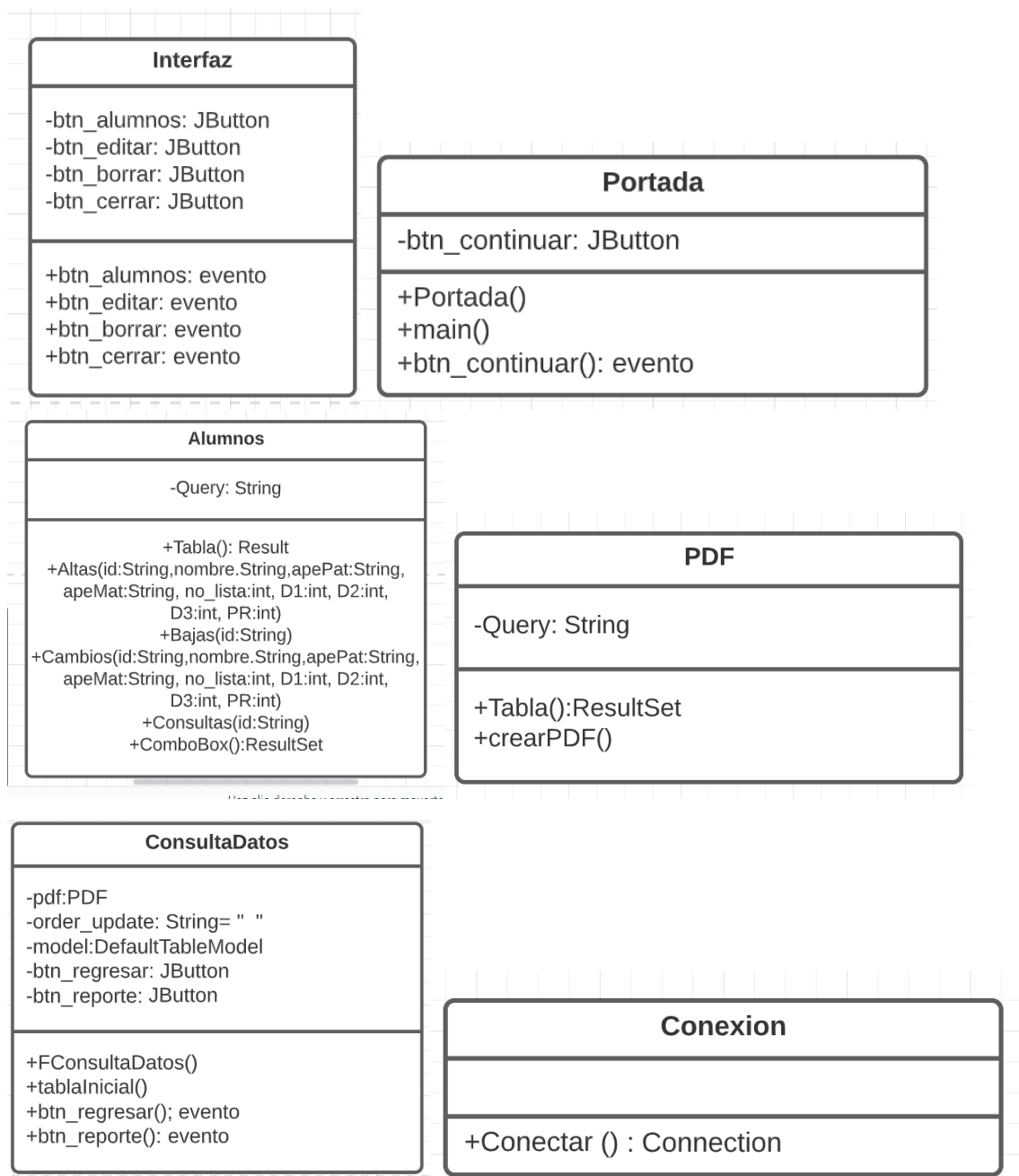
Se tomará como base el proceso en el cual, mostrará los registros de los estudiantes con el mayor promedio, los estudiantes con menor promedio, estudiantes con promedio superior a la media del grupo y los estudiantes con promedio igual o menor que la media del grupo.

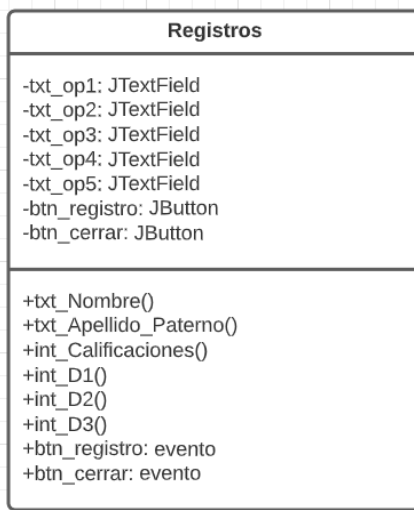
Todo esto con la finalidad de que cada uno de los integrantes del equipo se familiarice con el uso y manejo de base de datos, utilizando la programación orientada a objetos, así como diversas herramientas adicionales, las cuales permitirán que exista una relación entre las ya mencionadas. A continuación, se explicará el proceso que se llevó a cabo para este proyecto, donde se demuestra la funcionalidad de la aplicación creada.

## Diagramas de Composición de Clases

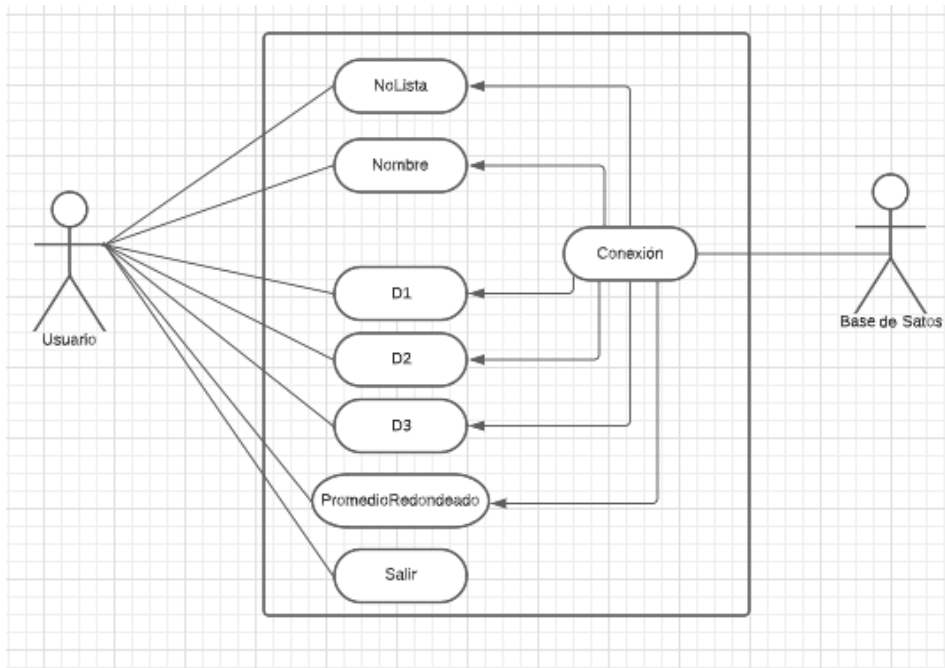


## Diagrama de Clases





## Diagrama de Casos de Uso



## Pseudocódigo de todos los métodos

```
//Alumnos métodos de la clase Alumnos-07
sub main(String[] args)
!Enunciados
{
    Interfaz app <- Nuevo Interfaz();
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    app.setResizable(falso);
    app.setLocationRelativeTo(nulo);
}
Fin Sub
```

La implementación de este método está en la página (11) a la línea (13)

```
//Conectar
sub Conectar()
{
    !Enunciados
    public Connection conexion()
    {
        try{
            class.forName("com.mysql.jdbc.Driver")
            cn <-
            DriverManager.getConnection("jdbc:mysql://localhost/calificaciones",
            "root", "1234");
        }
        retornar cn;
    }
}
Fin sub
```

La implementación de este método está en la página (13) a la línea (13)

```
//Interfaz métodos de la clase Alumnos-07
sub Interfaz()
!Enunciados
{
    title <- new Label("Bienvenido");
    title.setBounds(250,30,150,30);
    title.setFont(new Font("Consolas", 1, 13));
    agregar(title);

//Table
candena[][] data = {
    { "Kundan Kumar Jha", "4031", "CSE", "Kundan Kumar Jha", "4031", "CSE"
},

```



```

{ "Anand Jha", "6014", "IT", "Kundan Kumar Jha", "4031", "CSE" } };
cadena[] names = { "Nombre", "Apellido", "D1", "D2", "D3", "Promedio" };
tab <- Nuevo JTable(data, names);
tab.setBounds(55, 120, 490, 300);
agregar(tab);
// Buttons
alumnos <- Nuevo Button("Registrar alumno");
alumnos.setBounds(55, 70, 150, 30);
agregar(alumnos);
    alumnos.addActionListener(this);

    editar <- Nuevo Button("Editar alumno");
    editar.setBounds(225, 70, 150, 30);
    agregar(editar);
    editar.addActionListener(this);

    borrar <- Nuevo Button("Borrar alumno");
    borrar.setBounds(395, 70, 150, 30);
    agregar(borrar);
    borrar.addActionListener(this);

    cerrar <- Nuevo Button("Cerrar");
    cerrar.setBounds(250, 510, 100, 30);
    agregar(cerrar);
    cerrar.addActionListener(this);

// General settings
setTitle("Alumnos");
setLayout(null);
setSize(600, 600);
setVisible(true);

sub actionPerformed(ActionEvent e)
si(e.getSource() == cerrar)
    System.exit(0);
fin si
si(e.getSource() == alumnos){
    escribir("Studenten");
    String title <- "Registrar alumno";
    Registros app <- nuevo Registros(title);
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    app.setResizable(falso);
    app.setLocationRelativeTo(nulo);
}
fin si
si(e.getSource() == editar){
    escribir("Edit");
    String title <- "Editar alumno";

```

```

        Registros app <- nuevo Registros(title);
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.setResizable(falso);
        app.setLocationRelativeTo(nulo);
    fin si
    si(e.getSource() == borrar){
        escribir("Delete");
    fin si
}
Fin sub

```

La implementación de este método está en la página (19) a la línea (14)

```

//Registros métodos de la clase Alumnos-07
sub Registros()
{
    !Enunciados
    title <- new Label(titulo);
    title.setBounds(250,30,150,30);
    title.setFont(new Font("Consolas", 1, 13));
    add(title);
    //Labels
    title <- new Label(titulo);
    title.setBounds(250,30,150,30);
    title.setFont(new Font("Consolas", 1, 13));
    add(title);

    nombre <- new Label("Nombre");
    nombre.setBounds(50,70,150,30);
    nombre.setFont(new Font("Consolas", 1, 13));
    add(nombre);

    apellidoPaterno <- new Label("Apellido paterno");
    apellidoPaterno.setBounds(50,110,150,30);
    apellidoPaterno.setFont(new Font("Consolas", 1, 13));
    add(apellidoPaterno);

    calificaciones <- new Label("Calificaciones");
    calificaciones.setBounds(250,190,150,30);
    calificaciones.setFont(new Font("Consolas", 1, 13));
    add(calificaciones);

    d1 <- new Label("D1:");
    d1.setBounds(75,230,25,30);
    add(d1);

    d2 <- new Label("D2:");
    d2.setBounds(230,230,25,30);

```

```

        add(d2);

        d3 <- new Label("D3:");
        d3.setBounds(385,230,25,30);
        add(d3);

// Text fields
        op1 = new TextField();
        op1.setBounds(200,70,200,30);
        add(op1);

        op2 = new TextField();
        op2.setBounds(200,110,200,30);
        add(op2);

        op3 = new TextField();
        op3.setBounds(120,230,50,30);
        add(op3);

        op4 = new TextField();
        op4.setBounds(275,230,50,30);
        add(op4);

        op5 = new TextField();
        op5.setBounds(430,230,50,30);
        add(op5);

// Buttons
        cerrar = new Button("Cerrar");
        cerrar.setBounds(350, 310, 100, 30);
        add(cerrar);
        cerrar.addActionListener(this);

        registro = new Button("Enviar");
        registro.setBounds(150, 310, 100, 30);
        add(registro);
        registro.addActionListener(this);

// General settings
        setTitle("Registro");
        setLayout(null);
        setSize(600, 380);
        setVisible(true);

sub actionPerformed(ActionEvent e){
    si(e.getSource() == cerrar){
        System.exit(0);
    }
}

```

```
        finis  
    }  
Fin sub
```

La implementación de este método está en la página (19) a la línea (16)

## Implementación de todos los métodos

### Clase Alumnos

#### Método Interfaz

```
9      public class Alumnos {  
10  
11      public static void main(String[] args) {  
12  
13          Interfaz app = new Interfaz();  
14      }  
15  }
```

### Clase Conectar

#### Método conexión

```
6      public class Conectar {  
7  
8          Connection cn;  
9  
10     public Connection conexion(){  
11         try{  
12             Class.forName("com.mysql.jdbc.Driver");  
13  
14             cn = DriverManager.getConnection("jdbc:mysql://localhost/calificaciones", "root", "1234");  
15  
16             System.out.println("Conectado");  
17         }catch(Exception e){  
18  
19             System.out.println(e);  
20         }  
21  
22         return cn;  
23     }  
24 }
```

### Clase Interfaz

```

19 public class Interfaz extends JFrame implements ActionListener{
20     private Label title;
21     private Button alumnos, editar, borrar, cerrar, actualizar, reporte;
22     private JTable tab;
23     String[][] data = new String[30][7];
24     String[] columnNames = {"Lista", "Nombre", "Apellido", "D1", "D2", "D3", "Promedio" };
25     Conectar con = new Conectar();
26     Connection cn = con.conexion();
27
28
29     public Interfaz() {
30
31         title = new Label("Bienvenido");
32         title.setBounds(250,30,150,30);
33         title.setFont(new Font("Consolas", 1, 13));
34         add(title);
35
36         tabla();
37         tab = new JTable(data, columnNames);
38         tab.setBounds(55, 120, 490, 300);

```

Evento Registrar

Evento btn\_alumnos

```

44 // Buttons
45 alumnos = new Button("Registrar alumno");
46 alumnos.setBounds(55, 70, 150, 30);
47 add(alumnos);
48 alumnos.addActionListener(this);

```

Evento btn\_editar

```

50 editar = new Button("Editar alumno");
51 editar.setBounds(225, 70, 150, 30);
52 add(editar);
53 editar.addActionListener(this);

```

Evento btn\_borrar

```

54 borrar = new Button("Borrar alumno");
55 borrar.setBounds(395, 70, 150, 30);
56 add(borrar);
57 borrar.addActionListener(this);

```

Evento btn\_actualizar

```

59 actualizar = new Button("Actualizar");
60 actualizar.setBounds(395, 70, 150, 30);
61 add(actualizar);
62 actualizar.addActionListener(this);

```

Evento btn\_reporte

```

65         reporte = new Button("Reporte PDF");
66         reporte.setBounds(395, 70, 150, 30);
        add(reporte);
        reporte.addActionListener(this);
69

```

### Evento btn\_cerrar

```

70         cerrar = new Button("Cerrar");
71         cerrar.setBounds(250, 510, 100, 30);
        add(cerrar);
        cerrar.addActionListener(this);
74
75 // General settings

```

### Metodo Crear tabla

```

85 public void tabla() {
86     // Table
87     String sql = "SELECT * FROM alumno";
88     Statement st;
89
90     try{
91
92         st = cn.createStatement();
93         ResultSet rs = st.executeQuery(sql);
94
95         while(rs.next()){
96             int row = rs.getRow();
97
98             for(int i=0; i<7; i++){
99                 data[row-1][i] = rs.getString(i+1);
100             }
101
102         }
103
104     }catch(SQLException e){
105         System.err.println("Error");

```

### Método actualizar

```

109 public void actionPerformed(ActionEvent e) {
110     if(e.getSource() == cerrar){
111         System.exit(0);
112     }
113     if(e.getSource() == actualizar){
114         System.out.println("Actualizando");
115         tabla();
116         tab.repaint();
117     }
118     if(e.getSource() == alumnos){
119         String title = "Registro";
120         Registros app = new Registros(title);
121     }
122     if(e.getSource() == editar){
123         String title = "Editar alumno";
124         int row = tab.getSelectedRow();
125         Object id = tab.getValueAt(row, 0);
126         Registros app = new Registros(title);
127         app.setId(id);
128         app.fillOut();
129     }

```

## Clase Registros

### Metodo Registro

```
9      public class Registros extends JFrame implements ActionListener{
10          private Label title, nombre, apellidoPaterno, calificaciones, d1, d2, d3;
11          private TextField op1, op2, op3, op4, op5;
12          private Button registro, cerrar;
13          String titulo, name, surname;
14          int c1, c2, c3, promedio;
15          Conectar con = new Conectar();
16          Connection cn = con.conexion();
17          Object id;
18
19      public Registros(String titulo) {
20
21          // Labels
22          title = new Label(titulo);
23          title.setBounds(250,30,150,30);
24          title.setFont(new Font("Consolas", 1, 13));
25          add(title);
26          this.titulo = titulo;
27
28          nombre = new Label("Nombre");
```

### Metodo Registros

```
19      public Registros(String titulo) {
20
21          // Labels
22          title = new Label(titulo);
23          title.setBounds(250,30,150,30);
24          title.setFont(new Font("Consolas", 1, 13));
25          add(title);
26          this.titulo = titulo;
27
28          nombre = new Label("Nombre");
29          nombre.setBounds(50,70,150,30);
30          nombre.setFont(new Font("Consolas", 1, 13));
31          add(nombre);
32
33          apellidoPaterno = new Label("Apellido paterno");
34          apellidoPaterno.setBounds(50,110,150,30);
35          apellidoPaterno.setFont(new Font("Consolas", 1, 13));
36          add(apellidoPaterno);
37
38          calificaciones = new Label("Calificaciones");
39          calificaciones.setBounds(250,190,150,30);
40          calificaciones.setFont(new Font("Consolas", 1, 13));
41          add(calificaciones);
42
43          d1 = new Label("D1:");
44          d1.setBounds(75,230,25,30);
```

### Metodo resultado

```

97
98 public void fillOut(){
99     String consulta = "SELECT * FROM alumno WHERE alumno.NumeroLista = " + id;
100     Statement st;
101
102     try{
103         st = cn.createStatement();
104         ResultSet rs = st.executeQuery(consulta);
105         while(rs.next()){
106             for(int i=1; i<=7; i++){
107                 System.out.println(rs.getString(i));
108             }
109             op1.setText(rs.getString(2));
110             op2.setText(rs.getString(3));
111             op3.setText(rs.getString(4));
112             op4.setText(rs.getString(5));
113             op5.setText(rs.getString(6));
114         }
115     }catch(SQLException err){
116         System.err.println("Error al cargar el alumno");
117     }
118 }

```

## Clases Registro

### Método Promedio

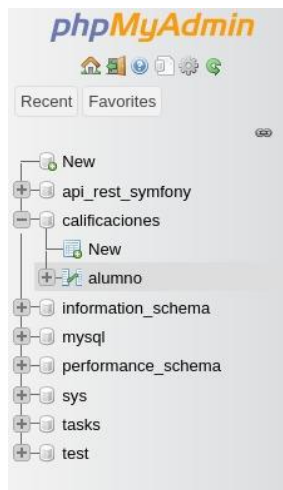
```

133
134 public void promedio(String d1, String d2, String d3){
135     c1 = Integer.parseInt(d1);
136     c2 = Integer.parseInt(d2);
137     c3 = Integer.parseInt(d3);
138     float prm = (c1+c2+c3)/3f;
139     promedio = Math.round(prm);
140 }
141
142 public void clean(){
143     op1.setText("");
144     op2.setText("");
145     op3.setText("");
146     op4.setText("");
147     op5.setText("");
148 }
149
150 public void setId(Object id){
151     this.id = id;
152 }
153
154 public void actionPerformed(ActionEvent e){
155     if(e.getSource() == cerrar){
156         setVisible(false);
157     }
158
159     if(e.getSource() == registro){

```



## Tablas de la BD



Extra options

					NumeroLista	Nombre	ApellidoPaterno	D1	D2	D3	PromedioRedondeado
<input type="checkbox"/>	Edit	Copy	Delete		1	Francisco	Lopez	9	7	8	8
<input type="checkbox"/>	Edit	Copy	Delete		2	Roman	Chavez	7	7	7	7
<input type="checkbox"/>	Edit	Copy	Delete		3	Carlos	Gomez	8	7	8	8
<input type="checkbox"/>	Edit	Copy	Delete		4	Guillermo	Flores	6	8	6	7
<input type="checkbox"/>	Edit	Copy	Delete		5	Roberto	Lopez	7	9	7	8
<input type="checkbox"/>	Edit	Copy	Delete		6	Maria	Garcia	8	4	8	6
<input type="checkbox"/>	Edit	Copy	Delete		7	Lesley	Martines	10	10	10	10
<input type="checkbox"/>	Edit	Copy	Delete		8	Jocelyn	Contreras	10	10	10	10
<input type="checkbox"/>	Edit	Copy	Delete		20	Andrea	Gomez	7	8	5	7
<input type="checkbox"/>	Edit	Copy	Delete		21	Carl	Marx	10	9	10	10
<input type="checkbox"/>	Edit	Copy	Delete		22	Mario	Oviedo	10	9	9	9

☐ Check all    With selected:    Edit    Copy    Delete    Export

☐ Show all    Number of rows: 25    Filter rows: Search this table    Sort by key: None

Server: localhost Database: calificaciones Table: alumno

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 <b>NumeroLista</b>	int(10)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 <b>Nombre</b>	char(50)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 <b>ApellidoPaterno</b>	char(50)	utf8mb4_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	4 <b>D1</b>	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	5 <b>D2</b>	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	6 <b>D3</b>	int(10)			No	None			Change  Drop  More
<input type="checkbox"/>	7 <b>PromedioRedondeado</b>	int(10)			No	None			Change  Drop  More

☐ Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after PromedioRedondeado Go

## Capturas de pantalla de prueba exhaustivas

### Ventana de Registro

Bienvenido

Lista	Nombre	Apellido	D1	D2	D3	Promedio
1	Franisco	Lopez	9	7	8	8
2	Roman	Chavez	7	7	7	7
3	Carlos	Gomez	8	9	8	8
4	Guillermo	Flores	6	8	6	7
5	Roberto	Lopez	7	9	7	8
6	Maria	Garcia	8	4	8	6
7	Lesley	Martinez	10	10	10	10
8	Jocelyn	Contreras	10	10	10	10
20	Andrea	Gomez	7	8	5	7
21	Carl	Marx	10	9	10	10

**Registro**

Nombre

Apellido paterno

**Calificaciones**

D1:  D2:  D3:

dependency information available  
 dependency information available



447/2/538:0MB

Lista	Nombre	Apellido	D1	D2	D3	Promedio
1	Francisco	Lopez	9	7	8	8
2	Roman	Chavez	7	7	7	7
3	Carlos	Gomez	8	9	8	8
4	Guillermo	Flores	8	8	6	7
5	Roberto	Lopez	7	9	7	8
6	Maria	Garcia	8	4	8	6
7	Lesley	Martinez	10	10	10	10
8	Jocelyn	Contreras	10	10	10	10
20	Andrea	Gomez	7	8	5	7
21	Carl	Marx	10	9	10	10
22	Mario	Oviedo	10	9	9	9

Bienvenido

Registrar alumno Editar alumno Borrar alumno Actualizar Reporte PDF Cerrar

**Editar alumno**

Nombre

Apellido paterno

**Calificaciones**

D1:  D2:  D3:

Enviar Cerrar

## Visualización del registro exitoso

Lista	Nombre	Apellido	D1	D2	D3	Promedio
1	Francisco	Lopez	9	7	8	8
2	Roman	Chavez	7	7	7	7
3	Carlos	Gomez	8	9	8	8
4	Guillermo	Flores	8	8	6	7
5	Roberto	Lopez	7	9	7	8
6	Maria	Garcia	8	4	8	6
7	Lesley	Martinez	10	10	10	10
8	Jocelyn	Contreras	10	10	10	10
20	Andrea	Gomez	7	8	5	7
21	Carl	Marx	10	9	10	10
22	Mario	Oviedo	10	9	9	9

Bienvenido

Registrar alumno Editar alumno Borrar alumno Actualizar Reporte PDF Cerrar

**Editar alumno**

Nombre

Apellido paterno

**Calificaciones**

D1:  D2:  D3:

Enviar Cerrar

**Datos actualizados**

OK

Conectado

## Visualización del reporte creado

Bienvenido

Lista	Nombre	Apellido	D1	D2	D3	Promedio
1	Francisco	Lopez	9	7	8	8
2	Roman	Chavez	7	7	7	7
3	Carlos	Gomez	8	7	8	8
4	Guillermo	Flores	6	8	6	7
5	Roberto	Lopez	7	9	7	8
6	Maria	Garcia	8	4	8	6
7	Lesley	Martines	10	10	10	10
8	Jocelyn	Contreras	10	10	10	10
20	Andrea	Gomez	7	8	5	7
21	Carl	Marx	10	9	10	10
22	Mario	Oviedo	10	9	9	9

Reporte creado

OK

Registrar alumno Editar alumno Borrar alumno Actualizar Reporte PDF Cerrar

```

/srv/http/html/Java/POO/Alum... -/Documents
-A 4 K
~@ Main. 98 K
~@ add. 39 K
~@ add2. 67 K
~@ del. 63 K
~@ del2. 73 K
~@ del3. 75 K
~@ edit. 64 K
~@ edit2. 69 K
~@ edit3. 63 K
~@ report. 79 K
~@ report2. 159 K
-A Boo 4 K
-A Dispositivos 4 K
-A E 4 K
-A Material Redes julio 4 K
-A O 4 K
-A Standard Notes Backu 4 K
-A g 4 K
-A mya 4 K
-G-A #20_3MNGO SI EQ7 319 K
-G-A 20222-2013040842-C 92 K
-G-A 20222-2013040842-C 90 K
-G-A 11604088-C1 975 K
-R-G CE 837 K
-G-B CV-Gerardo Palmier 59 K
-G-A CV-Gerardo Palmier 37 K
-G-A Constancia-SICAD 848 K
-G-A Dise o de interf 271 K
-G-B Goals. 5.4 K
-G-A Lista 1.8 K
-G-A PAMG970915HMCLNR06 95 K
-G-B PalmieriMondragon 5.1 K
-G-A Portal de tr mite 75 K
-D api-laravel.postma 606 B
-D api-rest-node.post 5.2 K
-G-A carga_menor_2_la 257 K
-G-A carga_menor_2_la 328 K
-g credencialUpilcsa. 61 K
-D desktop_entr 1.4 K
-G-A tarjetaN5596119413 82 K
toad:toad 1.8 K 06/07 12
21/33

```

# lista	Nombre	Apellido	1er Parcial	2do Parcial	3er Parcial	Promedio
1	Francisco	Lopez	9	7	8	8
2	Roman	Chavez	7	7	7	7
3	Carlos	Gomez	8	7	8	8
4	Guillermo	Flores	6	8	6	7
5	Roberto	Lopez	7	9	7	8
6	Maria	Garcia	8	4	8	6
7	Lesley	Martines	10	10	10	10
8	Jocelyn	Contreras	10	10	10	10
20	Andrea	Gomez	7	8	5	7
21	Carl	Marx	10	9	10	10
22	Mario	Oviedo	10	9	9	9

## **Comentarios Finales**

De Santiago Landeros Edwin Uriel

Finalmente, por un lado, me gustaría enfatizar que, al realizar el proyecto con una base de datos, que permita consultar, eliminar, crear y actualizar una serie de registros. la materia en lo personal me ayudo a entender no por completo pero si entender sobre el uso de XAMPP, paquete que consiste principalmente en el sistema XAMPP que es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. ma de gestión de bases de datos, donde los principales problemas que experimentábamos era en la clase de pdf debido al uso de algunas extensiones que en equipo no teníamos del todo claro

Tomamos como base algunas actitudes que teníamos que corregir para trabajar un poco más conectados para que nadie se quedara atrás sin entender lo que estaba pasando en el desarrollo del proyecto y desarrollo del informe.

Herrera Cano Edwin Rubén

Una serie de acciones que se combinan en forma repetida para transformar las entradas en salidas en el pseudocódigo que el profesor nos corrigió nos hizo el trabajo de notar errores que teníamos en la composición de las clases, En las Pruebas de validación el producto satisface las necesidades de cómo se anotaron en el documento. En las pruebas de usuario la aportación de todos cuando teníamos errores para correrlo nos dio un resultado bueno ya que el compañero Gerardo nos ayudaba cuando no entendíamos esperado a los usuarios.

Palmieri Mondragón Gerardo

Me doy cuenta de que he reflexionado sobre la figura de proyecto, la importancia de la realización de proyecto en la Programación, me he parado a distinguir entre los proyectos y las operaciones que hemos realizado a lo largo de todo el semestre en curso que son tareas constantes y repetitivas que se realizan como parte del funcionamiento habitual de un sistema. Vender, atender a los clientes, entregar los bienes, comprar, organizar los almacenes, facturar, cobrar, pagar las después de una operación. Todas estas tareas no son proyectos para dejarlos de lado ya que en ellas están contenidas los conocimientos prácticos y teóricos que funcionaron clase con clase en cada revisión para tener una retroalimentación para todo el equipo.

Reveles, Ramírez Jesús Emilio

En primer lugar, la fase de codificación, y la fase de programación, consiste en traducir a un lenguaje de programación lo que queríamos expresar, las funcionalidades y otros requisitos técnicos definidos durante la realización del informe.

## Villegas Nolasco Elide Yolotzin

La comunicación dentro la sesión donde el profesor nos corrigió, usando ejemplos y dentro del documento compartido que mostrábamos en las revisiones no ayudo a poder puntualizar los errores exactos y poder dejar una “nota” en donde el equipo presente también sabia de que se trataba para que después en la transformación del documento todos estuviéramos al tanto de las correcciones del documento, la transferencia de archivos, en las videoconferencias dentro de las salas de equipo fue un lugar y momento en donde nos poníamos de acuerdo para que podíamos investigar sobre el uso de la base de datos.