

# CRUD

## Actualización y eliminación

**Programación Orientada a Objetos**



# Preparando formulario

- Agrega un botón para actualizar y otro para eliminar y coloca un nombre adecuado para cada uno.

Profesores

Estudiantes

ID Estudiante

Limpiar campos

Nombre del alumno:

Dirección:

Contacto:

Usuario:

Guardar

Actualizar

Eliminar

Nombre o carnét

Buscar

Title 1	Title 2	Title 3	Title 4

# Actualización de datos



# Modelo - ModelEstudiantes

```
public boolean ActualizarAlumnoModel(int ID, String nombre_completo, String direccion, String contacto, String ip, String port, String db, String user, String pwd) {
    Connection con;
    try {
        con = ModelConexion.getConnection(ip, port, db, user, pwd);
        String query = "UPDATE Alumnos SET nombre_alumno = ?, direccion = ?, telefono = ? WHERE idalumno = ?";
        ps = con.prepareStatement(query);
        ps.setString(1, nombre_completo);
        ps.setString(2, direccion);
        ps.setString(3, contacto);
        ps.setInt(4, ID);
        ps.execute();
        return true;
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Ocurrió un error al actualizar el registros, vuelva a intentarlo y si el problema persiste consulte con el administrador del sistema. " + e.toString());
        return false;
    }
}
```

# Controlador - ControllerEstudiantes

```
public boolean ActualizarAlumno() {  
    return modelEst.ActualizarAlumnoModel(ID, nombre_completo, direccion, contacto, ControllerConfiguracion.IP, ControllerConfiguracion.Port, ControllerConfiguracion.DB, ControllerConfiguracion.User, ControllerConfiguracion.Password);  
}
```



# Vista – PanelUsuarios

```
29 public PanelUsuarios() {
30     initComponents();
31     String[] TitulosEstudiantes = {"ID", "Nombre del alumno", "Dirección", "Teléfono", "↩
Usuario", "Clave"};
32     modelo = new DefaultTableModel(null, TitulosEstudiantes);
33     TablaEstudiantes.setModel(modelo);
34     TablaEstudiantes.getColumnModel().getColumn(0).setMaxWidth(0);
35     TablaEstudiantes.getColumnModel().getColumn(2).setMaxWidth(0);
36     TablaEstudiantes.getColumnModel().getColumn(5).setMaxWidth(0);
37     CargarTabla();
38     BtnActualizar.setEnabled(false);
39     BtnEliminar.setEnabled(false);
40 }
```

# Vista - PanelUsuarios

```
42 void LimpiarCampos() {  
43     TxtID.setText("");  
44     TxtNombreAlumno.setText("");  
45     TxtDireccion.setText("");  
46     TxtContacto.setText("");  
47     TxtUsuario.setText("");  
48     claveEncrypt = "";  
49     BtnEliminar.setEnabled(false);  
50     BtnActualizar.setEnabled(false);  
51     BtnRegistrar.setEnabled(true);  
52 }
```

# Vista – PanelUsuarios

```
91 final void ActualizarAlumnos() {
92     if (TxtNombreAlumno.getText().trim().isEmpty() || TxtDireccion.getText().trim().isEmpty() ||
93         TxtContacto.getText().trim().isEmpty() || TxtUsuario.getText().trim().isEmpty()) {
94         JOptionPane.showMessageDialog(PnlAlumnos, "Existen campos vacíos.", "Información incompleta",
95             JOptionPane.WARNING_MESSAGE);
96     } else {
97         ControllerAlumnos objAlumnosEdit = new ControllerAlumnos();
98         objAlumnosEdit.ID = Integer.parseInt(TxtID.getText());
99         objAlumnosEdit.nombre_completo = TxtNombreAlumno.getText();
100         objAlumnosEdit.direccion = TxtDireccion.getText();
101         objAlumnosEdit.contacto = TxtContacto.getText();
102         boolean valor = objAlumnosEdit.ActualizarAlumno();
103         if (valor == true) {
104             JOptionPane.showMessageDialog(PnlAlumnos, "Alumno actualizado exitosamente.", "Proceso completado",
105                 JOptionPane.INFORMATION_MESSAGE);
106             CargarTabla();
107             LimpiarCampos();
108         }
109     }
110 }
```



# Vista – PanelUsuariso

- Crea el evento MouseClicked para la TablaEstudiantes y agrega el siguiente código

```
268 private void TablaEstudiantesMouseClicked(java.awt.event.MouseEvent evt) {  
269     // TODO add your handling code here:  
270     BtnActualizar.setEnabled(true);  
271     BtnEliminar.setEnabled(true);  
272     BtnRegistrar.setEnabled(false);  
273     if (evt.getClickCount() == 1) {  
274         //Captura de la fila a la que se hace click  
275         JTable rcp = (JTable) evt.getSource();  
276         TxtID.setText(rcp.getModel().getValueAt(rcp.getSelectedRow(), 0).toString());  
277         TxtNombreAlumno.setText(rcp.getModel().getValueAt(rcp.getSelectedRow(), 1).toString());  
278         TxtDireccion.setText(rcp.getModel().getValueAt(rcp.getSelectedRow(), 2).toString());  
279         TxtContacto.setText(rcp.getModel().getValueAt(rcp.getSelectedRow(), 3).toString());  
280         TxtUsuario.setText(rcp.getModel().getValueAt(rcp.getSelectedRow(), 4).toString());  
281     }  
282 }
```

# Vista - PanelUsuarios

- Crea el evento ActionPerformed para el botón actualizar e invoca al método ActualizarAlumnos()

```
289 private void BtnActualizarActionPerformed(java.awt.event.ActionEvent evt) {  
290     // TODO add your handling code here:  
291     ActualizarAlumnos();  
292 }
```

# Eliminación de datos



# Modelo - ModelEstudiantes

- Crea el siguiente método

```
74 public boolean EliminarAlumnoModel(int ID, String ip, String port, String db, String user, String pwd) {
75     Connection con;
76     try {
77         con = ModelConexion.getConnection(ip, port, db, user, pwd);
78         String query = "DELETE Alumnos WHERE idalumno = ?";
79         ps = con.prepareStatement(query);
80         ps.setInt(1, ID);
81         ps.execute();
82         return true;
83     } catch (SQLException e) {
84         JOptionPane.showMessageDialog(null, "Ocurrió un error al eliminar el registros, verifique que no e←
85         xistan datos dependientes, si el problema persiste consulte con el administrador del sistema. ");
86         return false;
87     }
88 }
```

# Controlador - ControllerAlumnos

- Crea el siguiente método

```
128 public boolean EliminarAlumno() {  
129     return modelEst.EliminarAlumnoModel(ID, ControllerConfiguracion.IP, ControllerConfigurac  
uracion.Port, ControllerConfiguracion.DB, ControllerConfiguracion.User, ControllerConfigurac  
ion.Password);  
130 }  
131 }
```



# Vista - PanelUsuarios

```
109 final void EliminarAlumnos() {
110     if (TxtID.getText().trim().isEmpty()) {
111         JOptionPane.showMessageDialog(PnlAlumnos, "Seleccione un registro.", "Información incompleta", JOp
tionPane.WARNING_MESSAGE);
112     } else {
113         int Confirmacion = JOptionPane.YES_NO_OPTION;
114         JOptionPane.showConfirmDialog(this, "¿Está seguro que quiere eliminar el registro?", "Confirmar ac
ción", Confirmacion);
115         if (Confirmacion == JOptionPane.YES_OPTION) {
116             ControllerAlumnos objAlumnosDel = new ControllerAlumnos();
117             objAlumnosDel.ID = Integer.parseInt(TxtID.getText());
118             boolean valor = objAlumnosDel.EliminarAlumno();
119             if (valor == true) {
120                 JOptionPane.showMessageDialog(PnlAlumnos, "Alumno eliminado exitosamente.", "Proceso compl
etado", JOptionPane.INFORMATION_MESSAGE);
121                 CargarTabla();
122                 LimpiarCampos();
123             }
124         }
125     }
126 }
```