



UNIVERSIDAD MAYOR DE SAN SIMON FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA INGENIERIA DE SISTEMAS



Trabajos

Materia: Sistemas de Información II.

Docente: Flores Soliz Juan Marcelo.

Estudiantes: Eliot Rodrigo Heredia Villanueva.

Trabajo 3

USER STORY – ACCESO A LA PLATAFORMA

Descripción:

Como docente postulante, quiero acceder a una plataforma digital para rendir el examen, para demostrar mis competencias de forma ordenada y segura.

Criterios de Aceptación:

- Debo poder iniciar sesión con un usuario y contraseña únicos.
- La plataforma debe estar disponible durante el periodo de evaluación.
- El acceso debe funcionar desde distintos dispositivos (PC, laptop).
- Si las credenciales son incorrectas, se debe mostrar un mensaje de error.

Tareas Técnicas:

- Implementar el módulo de autenticación (login).
- Validar credenciales contra la base de datos de docentes.
-
- Incluir mensajes de error en caso de acceso fallido.
- Probar la compatibilidad en distintos dispositivos.

3	
---	--

USER STORY – INSTRUCCIONES CLARA

Descripción:

Como docente evaluado, quiero recibir instrucciones claras y detalladas sobre el examen, para comprender el proceso y evitar errores durante la evaluación.

Criterios de Aceptación:

- Debe mostrarse un instructivo digital antes de iniciar el examen.
- El instructivo debe contener ejemplos o preguntas de práctica.
- El lenguaje debe ser claro y comprensible.
- Debe existir un botón de “Confirmar que he leído las instrucciones”.

6	
---	--

Tareas Técnicas:

- Diseñar la sección de instrucciones previas al examen.
- Incluir ejemplos de preguntas interactivas.
- Implementar el botón de confirmación antes de iniciar.
- Validar que no se pueda comenzar sin aceptar las instrucciones.

USER STORY – EVALUACIÓN INTEGRAL

Descripción:

Como postulante a la universidad, quiero que el examen incluya preguntas teóricas y prácticas, para evidenciar tanto mis conocimientos como mis habilidades pedagógicas.

Criterios de Aceptación:

- El examen debe incluir al menos 60% de preguntas teóricas y 40% prácticas.
- Las preguntas prácticas deben corresponder al área de docencia del postulante.
- La evaluación debe medir precisión de respuestas y capacidad pedagógica.
- El examen debe tener límite de tiempo configurado.

Tareas Técnicas:

5	
---	--

- Crear banco de preguntas teóricas y prácticas.
- Implementar lógica de distribución (60/40).
- Configurar temporizador en la plataforma.
- Desarrollar módulo de calificación automática.

USER STORY – RETROALIMENTACIÓN DEL DESEMPEÑO

Descripción:

Como docente candidato, quiero obtener retroalimentación de mi desempeño en el examen, para identificar mis fortalezas y áreas de mejora.

Criterios de Aceptación:

- Debe generarse un reporte con calificación y observaciones.
- La retroalimentación debe incluir sugerencias de mejora.
- El informe debe estar disponible en un máximo de 5 días hábiles.

- El reporte debe ser descargable en formato PDF.

7	
---	--

- **Tareas Técnicas:**

- Implementar módulo de generación de reportes.
- Conectar la calificación con el sistema de retroalimentación.
- Crear plantilla PDF para la retroalimentación.
- Configurar envío automático de resultados.

USER STORY – TRANSPARENCIA DEL PROCESO

Descripción:

Como aspirante a ser parte del plantel docente universitario, quiero que el proceso de examen sea transparente y justo, para tener la seguridad de que la selección se basa en méritos y competencias.

Criterios de Aceptación:

- El mecanismo de evaluación debe garantizar anonimato en la corrección.
- Los criterios de evaluación deben publicarse antes del examen.
- Los resultados deben ser verificables y auditables.
- Debe generarse un acta final con los resultados.

- **Tareas Técnicas:**

4	
---	--

- Implementar un sistema de anonimización de respuestas.
- Publicar criterios de evaluación en la plataforma.
- Configurar bitácoras de auditoría para resultados.
- Crear módulo de generación de actas finales.

Trabajo 2

Corregido

Código del programa

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConexionBD {
    private static final String URL = "jdbc:mysql://localhost:3306/edificios?serverTimezone=UTC";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    private static Connection conexion;

    public static Connection getConexion() {
        if (conexion == null) {
            try {
                conexion = DriverManager.getConnection(URL, USER, PASSWORD);
                System.out.println(" Conexion exitosa a la base de datos");
            } catch (SQLException e) {
                System.err.println(" Error de conexion: " + e.getMessage());
            }
        }
        return conexion;
    }

    public static void cerrarConexion() {
        if (conexion != null) {
            try {
                conexion.close();
            }
        }
    }
}
```

```

        System.out.println(" Conexión cerrada");
    } catch (SQLException e) {
        System.err.println(" Error al cerrar conexión: " + e.getMessage());
    }
}
}
}
}

```

```

public class RegistrarEdificio extends javax.swing.JFrame {
    public RegistrarEdificio() {
        initComponents();
        this.setSize(470, 350);
        this.setLocationRelativeTo(null);
    }
    @SuppressWarnings("unchecked")
    private void initComponents() {
        JTextFieldNombre = new javax.swing.JTextField();
        JTextFieldAnioC = new javax.swing.JTextField();
        JTextFieldDireccion = new javax.swing.JTextField();
        JTextFieldNumeroPisos = new javax.swing.JTextField();
        botonRegistrar = new javax.swing.JButton();
        botonVerEdificios = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        JTextFieldNombre.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JTextFieldNombreActionPerformed(evt);
            }
        });

        JTextFieldAnioC.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JTextFieldAnioCActionPerformed(evt);
            }
        });

        JTextFieldDireccion.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JTextFieldDireccionActionPerformed(evt);
            }
        });

        JTextFieldNumeroPisos.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                JTextFieldNumeroPisosActionPerformed(evt);
            }
        });

        botonRegistrar.setText("registrar");
        botonRegistrar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        botonRegistrarActionPerformed(evt);
    }
});

botonVerEdificios.setText("ver edificios");
botonVerEdificios.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        botonVerEdificiosActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Segoe UI Black", 1, 12));
jLabel1.setText("nombre:");

jLabel2.setFont(new java.awt.Font("Segoe UI Black", 1, 12));
jLabel2.setText("año: ");

jLabel3.setFont(new java.awt.Font("Segoe UI Black", 1, 12));
jLabel3.setText("direccion:");

jLabel4.setFont(new java.awt.Font("Segoe UI Black", 1, 12));
jLabel4.setText("nº pisos:");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(botonRegistrar, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(botonVerEdificios, javax.swing.GroupLayout.PREFERRED_SIZE, 114,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(0, 234, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(JTextFieldNumeroPisos, javax.swing.GroupLayout.PREFERRED_SIZE,
130, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(43, 43, 43)
            .addComponent(JTextFieldNombre, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
            .addComponent(jLabel3)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addComponent(jLabel1)
                .addComponent(JTextFieldNumeroPisos)
                .addComponent(JTextFieldNombre)
                .addComponent(JTextFieldDireccion)
                .addComponent(botonRegistrar)
                .addComponent(botonVerEdificios)
            ))
    )
    .addComponent(jLabel4)
    .addComponent(jLabel3)
    .addComponent(jLabel2)
    .addComponent(jLabel1)
    .addComponent(JTextFieldNumeroPisos)
    .addComponent(JTextFieldNombre)
    .addComponent(JTextFieldDireccion)
    .addComponent(botonRegistrar)
    .addComponent(botonVerEdificios)
);

```

```

        .addGap(9, 9, 9)
        .addComponent(jLabel2)))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
        .addComponent(jTextFieldDireccion,
javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextFieldAnioC, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 130, javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(46, 46, 46)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextFieldNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel11))
            .addGap(24, 24, 24)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel13)
                .addComponent(jTextFieldDireccion, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jTextFieldAnioC, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel12))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel14)
                .addComponent(jTextFieldNumeroPisos, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 112,
Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(botonRegistrar, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(botonVerEdificios, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        );
    pack();
}

private void jTextFieldAnioCActionPerformed(java.awt.event.ActionEvent evt)
{
}

private void jTextFieldDireccionActionPerformed(java.awt.event.ActionEvent evt)
{
}

private void jTextFieldNumeroPisosActionPerformed(java.awt.event.ActionEvent evt)
{
}

```

```

    }

    private void JTextFieldNombreActionPerformed(java.awt.event.ActionEvent evt)
    {

    }

    private void botonRegistrarActionPerformed(java.awt.event.ActionEvent evt)
    {

        String nombre = JTextFieldNombre.getText().trim();
        String direccion = JTextFieldDireccion.getText().trim();
        String anioStr = JTextFieldAnioC.getText().trim();
        String pisosStr = JTextFieldNumeroPisos.getText().trim();

        if (nombre.isEmpty() || direccion.isEmpty() || anioStr.isEmpty() || pisosStr.isEmpty()) {
            javax.swing.JOptionPane.showMessageDialog(this, "Todos los campos son obligatorios.");
            return;
        }

        int anio;
        try {
            anio = Integer.parseInt(anioStr);
            if (anio < 1 || anio > 2025) {
                javax.swing.JOptionPane.showMessageDialog(this, "El año debe estar entre 1 y 2025.");
                return;
            }
        } catch (NumberFormatException e) {
            javax.swing.JOptionPane.showMessageDialog(this, "El año debe ser un número válido.");
            return;
        }

        int numPisos;
        try {
            numPisos = Integer.parseInt(pisosStr);
            if (numPisos <= 0) {
                javax.swing.JOptionPane.showMessageDialog(this, "El número de pisos debe ser mayor que 0.");
                return;
            }
        } catch (NumberFormatException e) {
            javax.swing.JOptionPane.showMessageDialog(this, "El número de pisos debe ser un número válido.");
            return;
        }

        //verificar si el edificio existe
        try {
            java.sql.Connection conn = ConexionBD.getConexion();

            String sqlExiste = "SELECT COUNT(*) FROM edificio WHERE LOWER(nombre) = LOWER(?)";
            try (java.sql.PreparedStatement pst = conn.prepareStatement(sqlExiste)) {
                pst.setString(1, nombre);
                java.sql.ResultSet rs = pst.executeQuery();
                if (rs.next() && rs.getInt(1) > 0) {
                    javax.swing.JOptionPane.showMessageDialog(this, "Ya existe un edificio con ese nombre.");
                    return;
                }
            }
        }
    }
}

```



```

        String sqlInsert = "INSERT INTO edificio (nombre, anio_construccion, direccion, num_pisos)
VALUES (?, ?, ?, ?)";
        try (java.sql.PreparedStatement pst = conn.prepareStatement(sqlInsert)) {
            pst.setString(1, nombre);
            pst.setInt(2, anio);
            pst.setString(3, direccion);
            pst.setInt(4, numPisos);

            int filas = pst.executeUpdate();
            if (filas > 0) {
                javax.swing.JOptionPane.showMessageDialog(this, "Edificio registrado
correctamente.");
                JTextFieldNombre.setText("");
                JTextFieldDireccion.setText("");
                JTextFieldAnioC.setText("");
                JTextFieldNumeroPisos.setText("");
            }
        }

    } catch (java.sql.SQLException e) {
        javax.swing.JOptionPane.showMessageDialog(this, "Error al registrar edificio: " +
e.getMessage());
    }
}

private void botonVerEdificiosActionPerformed(java.awt.event.ActionEvent evt)
{
    VerEdificios ver = new VerEdificios();
    ver.setVisible(true);
    this.dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(() -> {
        new RegistrarEdificio().setVisible(true);
    });
}

private javax.swing.JTextField jTextFieldAnioC;
private javax.swing.JTextField jTextFieldDireccion;
private javax.swing.JTextField jTextFieldNombre;
private javax.swing.JTextField jTextFieldNumeroPisos;
private javax.swing.JButton botonRegistrar;
private javax.swing.JButton botonVerEdificios;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
}

```

```

public class VerEdificios extends javax.swing.JFrame {
    public VerEdificios() {

```

```
initComponents();
cargarEdificiosEnTabla();
this.setLocationRelativeTo(null);
}
@SuppressWarnings("unchecked")
private void initComponents() {

    botonVolver = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tablaEdificios = new javax.swing.JTable();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    botonVolver.setText("volver");
    botonVolver.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            botonVolverActionPerformed(evt);
        }
    });

    tablaEdificios.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "nombre ", "anio", "direccion", "num pisos"
        }
    ));
    jScrollPane1.setViewportView(tablaEdificios);
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(19, 19, 19)
                .addComponent(botonVolver, javax.swing.GroupLayout.Alignment.TRAILING)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 435, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(53, 53, 53)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 196, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(botonVolver)
                .addContainerGap())
    );
    pack();
}
```

```

private void botonVolverActionPerformed(java.awt.event.ActionEvent evt)
{
    RegistrarEdificio registrar = new RegistrarEdificio();
    registrar.setVisible(true);
    this.dispose();
}

private void cargarEdificiosEnTabla() {
    try {
        java.sql.Connection conn = ConexionBD.getConexion();
        String sql = "SELECT nombre, anio_construccion, direccion, num_pisos FROM edificio";
        try (java.sql.Statement st = conn.createStatement();
            java.sql.ResultSet rs = st.executeQuery(sql)) {

            javax.swing.table.DefaultTableModel modelo = new javax.swing.table.DefaultTableModel(
                new Object[][]{},
                new String[]{"nombre", "anio", "direccion", "num pisos"}
            );

            while (rs.next()) {
                String nombre = rs.getString("nombre");
                int anio = rs.getInt("anio_construccion");
                String direccion = rs.getString("direccion");
                int numPisos = rs.getInt("num_pisos");

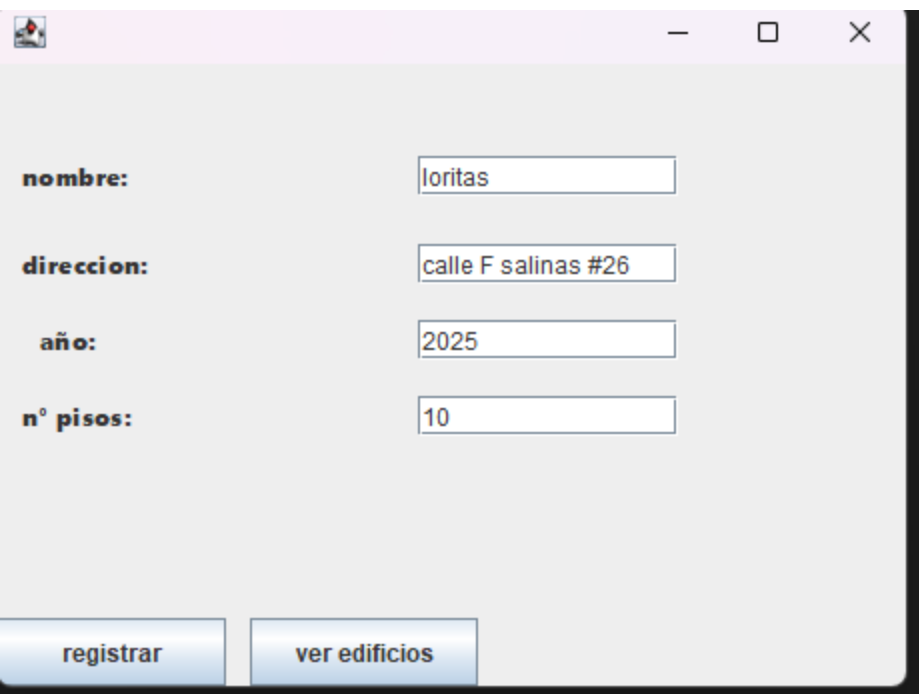
                modelo.addRow(new Object[]{nombre, anio, direccion, numPisos});
            }
            tablaEdificios.setModel(modelo);
        }
    } catch (java.sql.SQLException e) {
        javax.swing.JOptionPane.showMessageDialog(this, "Error al cargar edificios: " +
e.getMessage());
    }
}

private javax.swing.JButton botonVolver;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tablaEdificios;
}

```

Ejecución

registro



A screenshot of a web application window for registering a building. The window has a title bar with a small icon and standard minimize, maximize, and close buttons. The form contains four labeled input fields: 'nombre:' with the value 'loritas', 'direccion:' with 'calle F salinas #26', 'año:' with '2025', and 'nº pisos:' with '10'. At the bottom of the form are two buttons: 'registrar' and 'ver edificios'.

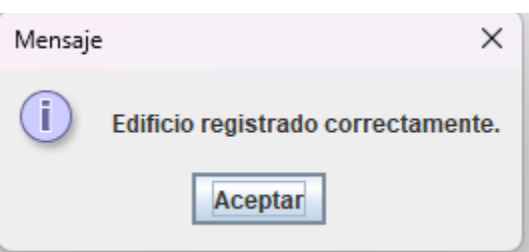
nombre:

direccion:

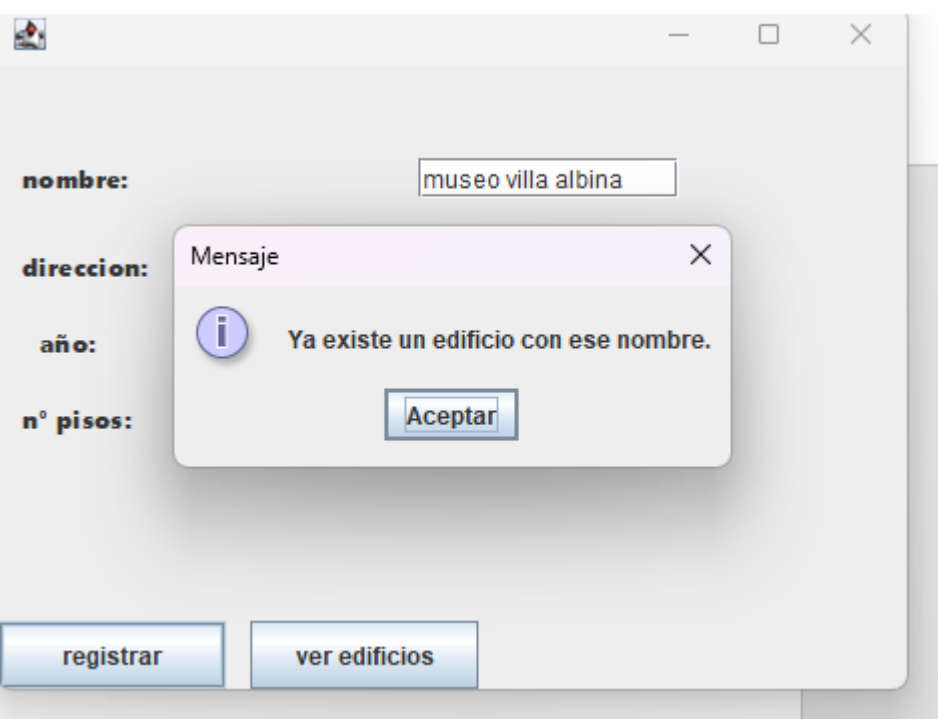
año:

nº pisos:

Mensaje de aprobación



Caso de que el edificio este registrado



Poder ver los edificios registrados

nombre	anio	direccion	num pisos
museo villa albi...	2019	av pairumani	2
convento santa t...	1760	calle baptista	2
Palacio Portales	1915	potosi buenos ai...	3
edificio del Juzg...	1929	calle Sucre y Sa...	14
condominio Ubu...	2022	av julio rodrigue...	25
edificio municip...	2025	calle lanza	19
edificio los tiem...	1989	av ramon rivero	22
edificio Colon	1994	mexico y Jose B...	22
Torre Florence	2024	AV Gabriel Rene ...	23
Kings Tower	2025	AV America	22
Torre Attenza	2022	calle Teofilo Var...	21

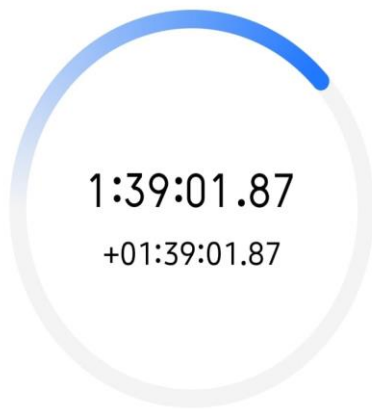
Base de datos

edificios.edificio: 23 filas en total (exacto)						
#	id_edificio	nombre	anio_construccion	direccion	num_pisos	
1	1	museo villa albina	2.019	av pairumani	2	
2	2	convento santa teresa	1.760	calle baptista	2	
3	3	Palacio Portales	1.915	potosi buenos aires	3	
4	4	edificio del Juzgado	1.929	calle Sucre y San Martin	14	
5	6	condominio Ubuntu torre sur	2.022	av julio rodrigues morales	25	
6	7	edificio municipal del GAMC	2.025	calle lanza	19	
7	8	edificio los tiempos	1.989	av ramon rivero	22	
8	9	edificio Colon	1.994	mexico y Jose Ballivian	22	
9	10	Torre Florence	2.024	AV Gabriel Rene Moreno	23	
10	11	Kings Tower	2.025	AV America	22	
11	12	Torre Attenza	2.022	calle Teofilo Vargas	21	
12	13	Torre Santa Rosa 1	2.020	calle Miguel Achata	18	
13	14	Torre santa Rosa 2	2.020	calle Miguel Achata	18	
14	15	Torre Ipetra	2.024	AV Melchor Perez de Olguin	20	
15	16	Torre Valhalla	2.025	B.M Pulacayo	20	
16	17	torre Kratos	2.018	AV Beiging	18	
17	18	condominio Itabua 1	2.023	AV Santa Cruz	19	
18	19	torre sofer	1.990	Av Oquendo 0654	17	
19	20	Rectorado UMSS	1.832	AV Ballivian 591	5	
20	21	edificio de la universidad catolica	1.950	M. Marquez	4	
21	22	sky Box	2.019	calle Antezana y AV Heroinas	10	
22	31	Garden Tower	2.013	AV Zenon Salinas 1565	9	
23	37	loritas	2.025	calle F salinas #26	10	

Tiempo tardado hasta la ejecución exitosa

Cronómetro

⋮



Trabajo 2

Código del programa

```
import java.sql.*; import
java.util.Scanner;

public class GestionEdificios {

    private static final String URL =
"jdbc:mysql://localhost:3306/edificios?serverTimezone=UTC";
    private static final String USER = "root"; private
    static final String PASSWORD = "";

    private Connection conexion;
    private Scanner scanner;

    public GestionEdificios() {
        scanner = new Scanner(System.in);
    }

    public void conectar() throws SQLException {
        conexion = DriverManager.getConnection(URL, USER, PASSWORD);
        System.out.println(" Conexión exitosa a la base de datos");
    }

    public void cerrar() { try {
        if (conexion != null && !conexion.isClosed()) conexion.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    scanner.close();
    System.out.println(" Saliendo del programa...");
```

```
}

private String preguntar(String pregunta) {
    System.out.print(pregunta);
    return scanner.nextLine();
}

private Integer validarNumero(String valor, String nombreCampo) { try {
    return Integer.parseInt(valor);
} catch (NumberFormatException e) {
    System.out.println(" Error: " + nombreCampo + " debe ser un número
válido.");
}
```

```

        return null;
    }
}

private Integer validarAño(String valor) { Integer
    n = validarNumero(valor, "Año"); if (n == null
    || n < 1 || n > 2025) {
        System.out.println(" Error: El año debe ser un número entre 1 y
2025.");
        return null;
    }
    return n;
}

public void registrarEdificio() throws SQLException { String
    nombre;
    while (true) {
        nombre = preguntar("Nombre del edificio: ");
        String sqlExiste = "SELECT COUNT(*) FROM Edificio WHERE nombre
= ?";

        try (PreparedStatement pst =
conexion.prepareStatement(sqlExiste)) {
            pst.setString(1, nombre); ResultSet rs =
            pst.executeQuery(); rs.next();
            if (rs.getInt(1) > 0) {
                System.out.println(" Ese nombre ya existe, ingresa
otro.");

                } else {
                    break;
                }
            }
        }

        Integer anio; do {
            anio = validarAño(preguntar("Año de construcción: "));
        } while (anio == null);

        String direccion = preguntar("Dirección: ");

        Integer numPisos; do {
            numPisos = validarNumero(preguntar("Número de pisos: "), "Número
de pisos");

```



```

    } while (numPisos == null);

    String sqlInsert = "INSERT INTO Edificio (nombre, anio_construccion,
direccion, num_pisos) VALUES (?, ?, ?, ?)";
    try (PreparedStatement pst = conexion.prepareStatement(sqlInsert,
Statement.RETURN_GENERATED_KEYS)) {
        pst.setString(1, nombre); pst.setInt(2,
anio); pst.setString(3, direccion);
        pst.setInt(4, numPisos);
        int affectedRows = pst.executeUpdate();

        if (affectedRows > 0) {
            try (ResultSet keys = pst.getGeneratedKeys()) { if
(keys.next()) {
                System.out.println(" Edificio insertado con ID: " +
keys.getInt(1));
            }
        }
    }
}

public void quitarEdificio() throws SQLException { Integer id;
do {
    id = validarNumero(preguntar("Ingresa el ID del edificio a
eliminar: "), "ID");
} while (id == null);

    String sqlDelete = "DELETE FROM Edificio WHERE id_edificio = ?"; try
(PreparedStatement pst = conexion.prepareStatement(sqlDelete)) {
        pst.setInt(1, id);
        int affectedRows = pst.executeUpdate();

        if (affectedRows > 0) {
            System.out.println(" Edificio con ID " + id + "
eliminado.");
        } else {
            System.out.println(" No se encontró ningún edificio con ID "
+ id + ".");
        }
    }
}

```

```

public void verEdificios() throws SQLException {
    String sqlSelect = "SELECT * FROM Edificio ORDER BY id_edificio
ASC";

    try (Statement st = conexion.createStatement()) {
        ResultSet rs = st.executeQuery(sqlSelect);

        boolean hayDatos = false;
        System.out.println("\nID | Nombre | Año Construcción | Dirección
| N° Pisos");
        System.out.println(" ");
        -----
        while (rs.next()) {
            hayDatos = true;
            int id = rs.getInt("id_edificio"); String
            nombre = rs.getString("nombre");
            int anio = rs.getInt("anio_construccion"); String
            direccion = rs.getString("direccion"); int numPisos =
            rs.getInt("num_pisos");

            System.out.printf("%d | %s | %d | %s | %d\n", id, nombre, anio,
direccion, numPisos);
        }

        if (!hayDatos) {
            System.out.println("❏ No hay edificios registrados.");
        }

    }
}

```

```

public void menu() throws SQLException { while
(true) {
    System.out.println("\n=== MENÚ DE EDIFICIOS ===");
    System.out.println("1 - Registrar edificio");
    System.out.println("2 - Quitar edificio"); System.out.println("3
- Ver edificios"); System.out.println("4 - Finalizar programa");

    String opcion = preguntar("Selecciona una opción: ");

    switch (opcion) { case "1":
        registrarEdificio(); break;
        case "2":
            quitarEdificio(); break;
    }
}
}

```

```

        case "3":
            verEdificios();
            break;
        case "4":
            cerrar();
            return;
        default:
            System.out.println(" Opción inválida, intenta
nuevamente.");
    }
}

}

}

public static void main(String[] args) {
    GestionEdificios app = new GestionEdificios();
    try {
        app.conectar();
        app.menu();
    } catch (SQLException e) {
        System.err.println(" Error: " + e.getMessage());
        app.cerrar();
    }
}
}

```

Ejecución

```

Conexión exitosa a la base de datos

=== MENÚ DE EDIFICIOS ===
1 - Registrar edificio
2 - Quitar edificio
3 - Ver edificios
4 - Finalizar programa
Selecciona una opción:

```

1.-

```
Selecciona una opción: 1
Nombre del edificio: Garden Tower
Año de construcción: 2013
Dirección: AV Zenon Salinas 1565
Número de pisos: 9
Edificio insertado con ID: 31
```

2.-

```
Selecciona una opción: 2
Ingresa el ID del edificio a eliminar: 5
Edificio con ID 5 eliminado.
```

3.-

```
Selecciona una opción: 3

ID | Nombre | Año Construcción | Dirección | Nº Pisos
-----
1 | museo villa albina | 2019 | AV pairumani | 2
2 | convento santa teresa | 1760 | calle baptista | 2
3 | Palacio Portales | 1915 | potosi buenos aires | 3
4 | edificio del Juzgado | 1929 | calle Sucre y San Martin | 14
6 | condominio Ubuntu torre sur | 2022 | av julio rodrigues morales | 25
7 | edificio municipal del GAMC | 2025 | calle lanza | 19
8 | edificio los tiempos | 1989 | av ramon rivero | 22
9 | edificio Colon | 1994 | mexico y Jose Ballivian | 22
10 | Torre Florence | 2024 | AV Gabriel Rene Moreno | 23
11 | Kings Tower | 2025 | AV America | 22
12 | Torre Attenza | 2022 | calle Teofilo Vargas | 21
13 | Torre Santa Rosa 1 | 2020 | calle Miguel Achata | 18
14 | Torre santa Rosa 2 | 2020 | calle Miguel Achata | 18
15 | Torre Ipetra | 2024 | AV Melchor Perez de Olguin | 20
16 | Torre Valhalla | 2025 | B.M Pulacayo | 20
17 | torre Kratos | 2018 | AV Beiging | 18
18 | condominio Itabua 1 | 2023 | AV Santa Cruz | 19
19 | torre sofer | 1990 | Av Oquendo 0654 | 17
20 | Rectorado UMSS | 1832 | AV Ballivian 591 | 5
21 | edificio de la universidad catolica | 1950 | M. Marquez | 4
22 | sky Box | 2019 | calle Antezana y AV Heroínas | 10
30 | casio | 2012 | av melchor | 14
31 | Garden Tower | 2013 | AV Zenon Salinas 1565 | 9
```

4.-

```
Selecciona una opción: 4
Saliendo del programa...
PS C:\Users\Usuario\Tarea 3\practica>
```

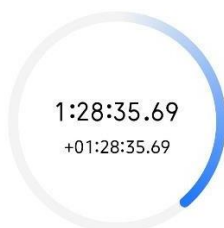
Base de Datos

#	id_edificio	nombre	anio_construccion	direccion	num_pisos
1	1	museo villa albina	2.019	AV pairumani	2
2	2	convento santa teresa	1.760	calle baptista	2
3	3	Palacio Portales	1.915	potosi buenos aires	3
4	4	edificio del Juzgado	1.929	calle Sucre y San Martin	14
5	6	condominio Ubuntu torre sur	2.022	av julio rodriguez morales	25
6	7	edificio municipal del GAMC	2.025	calle lanza	19
7	8	edificio los tiempos	1.989	av ramon rivero	22
8	9	edificio Colon	1.994	mexico y Jose Ballivian	22
9	10	Torre Florence	2.024	AV Gabriel Rene Moreno	23
10	11	Kings Tower	2.025	AV America	22
11	12	Torre Attenza	2.022	calle Teofilo Vargas	21
12	13	Torre Santa Rosa 1	2.020	calle Miguel Achata	18
13	14	Torre santa Rosa 2	2.020	calle Miguel Achata	18
14	15	Torre Ipetra	2.024	AV Melchor Perez de Olguin	20
15	16	Torre Valhalla	2.025	B.M Pulacayo	20
16	17	torre Kratos	2.018	AV Beiging	18
17	18	condominio Itabua 1	2.023	AV Santa Cruz	19
18	19	torre sofer	1.990	Av Oquendo 0654	17
19	20	Rectorado UMSS	1.832	AV Ballivian 591	5
20	21	edificio de la universidad catolica	1.950	M. Marquez	4
21	22	sky Box	2.019	calle Antezana y AV Heroinas	10
22	30	casio	2.012	av meldhor	14
23	31	Garden Tower	2.013	AV Zenon Salinas 1565	9

Tiempo Tardado hasta la ejecución exitosa

Cronómetro

:



Trabajo 1

¿Cuál es el significado de UML, que aplicaciones tiene en el trabajo de un programador y es necesario aplicarlo en todos los casos de programación?

Este fragmento ha sido extraído del libro Martin Fowler – UML Distilled: A Brief Guide to the Standard Object Modeling Language (tercera edición, 2003/2004), originalmente en inglés y en formato PDF (página 14). La traducción ha sido realizada por mí mismo.

El Lenguaje Unificado de Modelado (UML) es una familia de notaciones gráficas, respaldadas por un metamodelo único, que ayudan a describir y diseñar sistemas de software, particularmente sistemas de software contruidos utilizando el estilo orientado a objetos (OO). Esa es una definición algo simplificada. De hecho, el UML es distintas cosas para diferentes personas. Esto proviene tanto de su propia historia como de las diferentes perspectivas que tienen las personas sobre qué hace que un proceso de ingeniería de software sea efectivo. Como resultado, mi tarea en gran parte de este capítulo es contextualizar este libro explicando las diferentes formas en que la gente ve y utiliza el UML.

Según lo expone Martin Fowler en su libro UML Distilled, no es necesario aplicar UML en todos los contextos de programación. De hecho, sugiere un enfoque pragmático: utilizar únicamente aquella parte del lenguaje que realmente aporta valor. En sus palabras, basta con emplear el 20% del UML que permite resolver el 80% del trabajo.

Más que una obligación rígida, el UML debería ser una herramienta creativa y útil para apoyar el diseño y la arquitectura del software. Su uso cobra especial sentido en proyectos complejos, donde la visualización y la comunicación clara del diseño son fundamentales.

Este fragmento ha sido extraído del libro Grady Booch, James Rumbaugh, Ivar Jacobson – *The Unified Modeling Language User Guide* (2.ª edición, 2005), originalmente en inglés y en formato pdf (página 20 según pdf y capítulo 2) la traducción ha sido realizada por mí mismo.

El Lenguaje de Modelado Unificado (UML) es un lenguaje estándar para escribir planos de software. El UML puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema intensivo en software. El UML es adecuado para modelar sistemas que van desde sistemas de información empresariales hasta aplicaciones distribuidas basadas en la web e incluso sistemas embebidos en tiempo real. Es un lenguaje muy expresivo, que aborda todas las vistas necesarias para desarrollar y luego desplegar dichos sistemas. A pesar de su expresividad, el UML no es difícil de entender ni de usar. Aprender a aplicar el UML de manera efectiva comienza con la formación de un modelo conceptual del

lenguaje, lo cual requiere aprender tres elementos principales: los bloques básicos del UML, las reglas que dictan cómo pueden combinarse estos bloques, y algunos mecanismos comunes que se aplican en todo el lenguaje.

(pagina 15 según pdf y capítulo 1)

Para desarrollar dicho software de forma oportuna y predecible, usando eficientemente recursos humanos y materiales, se requiere un proceso de desarrollo sólido y adaptable. Es fundamental involucrar a los usuarios para exponer los requerimientos reales del sistema y construir una base arquitectónica sólida resistente al cambio. Además, se necesita contar con las personas adecuadas, herramientas apropiadas y enfoque correcto para minimizar retrabajos y desperdicios.

Este fragment ha sido extraído del libro Craig Larman – Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development originalmente en ingles y en format pdf (capítulo 1 pagina 21 según pdf) la traducción ha sido realizada por mi mismo

El UML (*Unified Modeling Language*) no es un método de análisis y diseño orientado a objetos, sino un **lenguaje de notación gráfica** que sirve para representar, visualizar y documentar modelos de un sistema. Permite mostrar diferentes perspectivas, como los casos de uso, la estructura de clases o el comportamiento dinámico.

Según Larman, aprender UML no garantiza por sí mismo un buen diseño, ya que lo importante es la capacidad de analizar, evaluar y mejorar los modelos, siendo esta la verdadera habilidad valiosa para un programador.

¿Que es lo que hace diferente al proceso unificado de los demás procesos y bajo que circunstancias no podría ser usado?

Este fragmento ha sido extraído del libro Grady Booch, James Rumbaugh, Ivar Jacobson – The Unified Modeling Language User Guide (2.ª edición, 2005), originalmente el ingles y en formato pdf (página 366 según pdf) la traducción ha sido realizada por mí mismo.

Un proceso es un conjunto de pasos parcialmente ordenados destinados a alcanzar un objetivo. En ingeniería de software, ese objetivo es entregar un producto de software de manera eficiente y predecible que satisfaga las necesidades del negocio.

El UML es en gran parte independiente del proceso, lo que significa que puede utilizarse con diversos procesos de ingeniería de software. El Proceso Unificado de Rational (RUP) es uno de esos enfoques de ciclo de vida que se adapta especialmente bien al UML.

El objetivo de RUP es permitir la producción de software de la más alta calidad que satisfaga las necesidades del usuario final dentro de calendarios y presupuestos predecibles. RUP captura algunas de las mejores prácticas actuales del desarrollo de software en una forma que puede adaptarse a una amplia variedad de proyectos y organizaciones.

En términos de gestión, RUP proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de software, facilitando el control y la coordinación durante el desarrollo.

(escrito en mis propias palabras)

El Proceso Unificado se distingue de otros procesos en:

1. Es un proceso iterativo e incremental que presenta resultados parciales en cada uno de sus ciclos.
2. Se enfoca en casos de uso para dirigir el diseño y los requisitos.
3. Establece como prioridad una arquitectura robusta antes de implementar todo.
4. Administra los riesgos desde el comienzo.
5. Se ajusta a distintos proyectos y equipos, es flexible y escalable.

No puede ser usado cuando los proyectos son muy pequeños o muy volátiles, también en proyectos de mantenimiento y de entrega pronta.

¿Como y por que se crearon los cimientos de los sistemas de información?

Francia: Documentación organizada y profesionalización de la IS, informatización y lenguajes de indización (Revista *Documentaliste*, 1993).

Alemania: Evolución de la documentación entre 1900–1945, estrechamente vinculada a bibliotecas especializadas (Behrends).

La Revolución científica de los siglos XVII-XVIII originó: · La aparición del periodismo científico y de las publicaciones periódicas. · La necesidad de disponer de información especializada. · La utilización de las revistas de resúmenes para el acceso y control de las publicaciones periódicas.

Reino Unido: Desarrollo temático de la IS desde 1945 hasta 1994, basado en el *Journal of Documentation* (Vickery, 1994).

España: Flujo constante de publicaciones sobre historia de la IS (López Yepes & Martínez Montalvo; Sagredo Fernández & García Moreno).

Estados Unidos: American Documentation Institute (ahora ASIS) y estudios sobre científicos de la información estadounidenses (Farkas-Conn; Lilley & Trice).

Los sistemas de información surgieron para **organizar, gestionar y recuperar información de manera eficiente**, con desarrollo temprano en Francia alrededor de 1880, expandiéndose luego a Alemania, Reino Unido, España y Estados Unidos, adaptándose a contextos académicos, profesionales y tecnológicos específicos.

¿Por qué es necesario esto teniendo un gran avance en la tecnología y la aplicación de la inteligencia artificial que puede hacer todo esto con una sola orden?

Organización y contexto:

La información no es solo datos; necesita **clasificación, contexto y relación con otros datos**.

La IA puede procesar datos rápidamente, pero **no siempre entiende el contexto completo ni los objetivos específicos del usuario**.

Control y confiabilidad:

Los sistemas de información bien diseñados aseguran **calidad, consistencia y seguridad de los datos**, algo crítico para decisiones empresariales o científicas.

La IA puede cometer errores o producir resultados inconsistentes si no hay un sistema que valide o supervise los datos.

Toma de decisiones humanas:

La IA ayuda, pero **la interpretación y el juicio final siguen siendo humanos**.

Los sistemas de información proporcionan estructuras y herramientas para que los humanos puedan analizar y decidir de manera efectiva.

Evolución y adaptación:

La IA aprende de los datos, pero necesita **bases de información históricas, completas y bien organizadas** para funcionar.

Los sistemas de información permiten mantener **registros y metodologías** que facilitan el aprendizaje de la IA.