


JCALENDAR

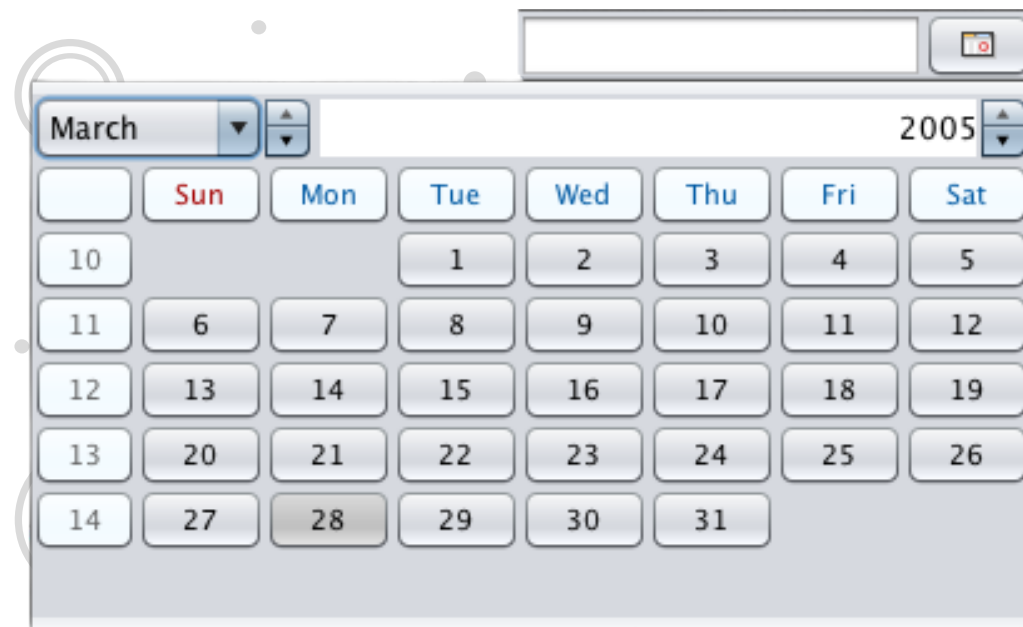
| SUN | MON | TUE | WED | THU | FRI | SAT | Events |
|-----|-----|-----|-----|-----|-----|-----|---|
| | | | | 01 | 02 | 03 | |
| 04 | 05 | 06 | 07 | 08 | 09 | 10 | |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 | |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 |  Java |

integrantes:

- Juan Dabid Rodriguez Fonseca
- Jesús David Peraza Royero

DESCRIPCION CONTROL

JCalendar es un componente gráfico de la librería JCalendar que proporciona una vista de calendario mensual interactiva. Permite al usuario seleccionar una fecha de forma visual, facilitando la interacción con datos temporales dentro de una aplicación. Este componente incluye navegación por meses y años, y puede integrarse fácilmente con otros controles como JDateChooser. Además, JCalendar está compuesto por varios subcomponentes como el selector de día, el selector de mes y el selector de año, los cuales pueden ser manipulados de forma individual. Esta modularidad permite un mayor control sobre el comportamiento y apariencia del calendario en la interfaz gráfica.





DESCRIPCION DEL CODIGO





Teniendo en cuenta el uso y funcionamiento de la librería JCalendar, se propuso desarrollar un programa que gestionara un repositorio de tareas utilizando la fecha seleccionada mediante sus componentes. Esta funcionalidad se implementó a través de una interfaz gráfica siguiendo el patrón de diseño MVC (Modelo-Vista-Controlador).

DESCRIPCION DE PROPIEDADES

Background Colors :

-  rgb: [199,3,3]
-  rgb: [255,255,255]
-  rgb: [210,228,238]
-  rgb: [242,242,242]

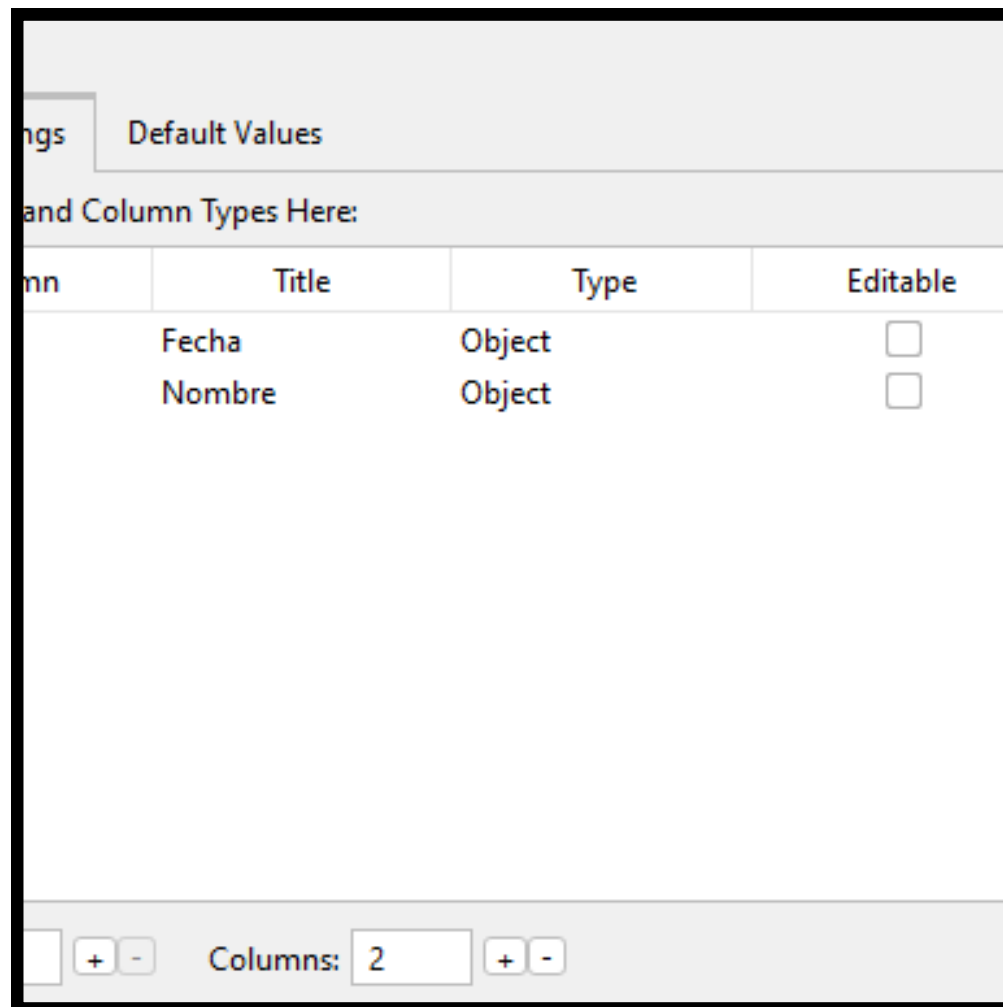
Foreground Colors :

-  rgb: [0,0,0]
-  rgb: [255,255,255]
-  rgb: [164,0,0]
-  rgb: [0,90,164]

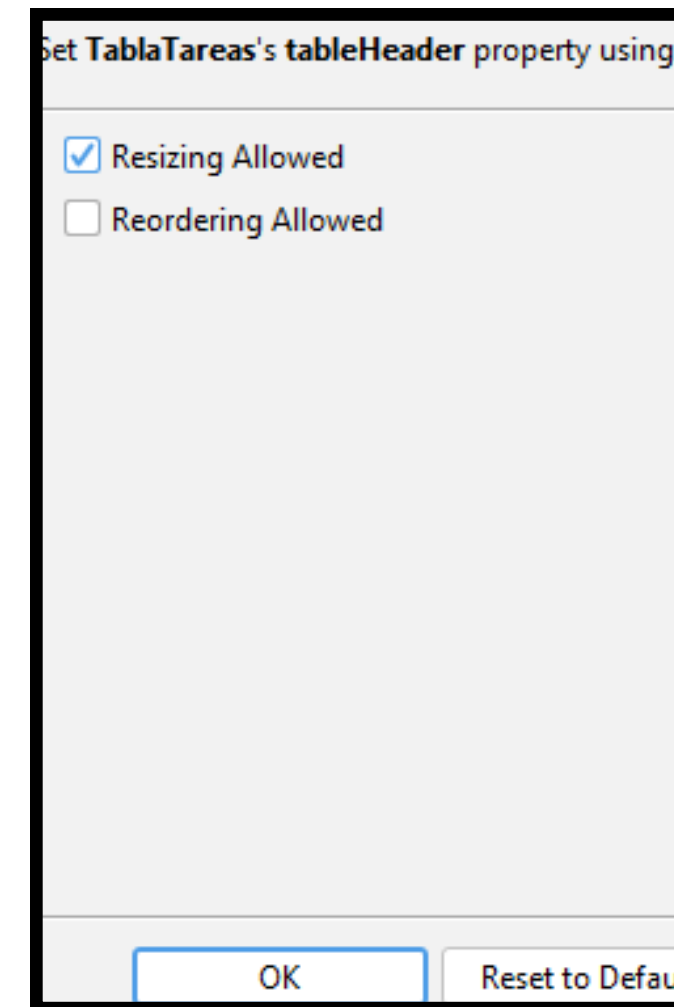
Font (Tipo de Fuente y Estilo):

Tipo: Arial
Estilo: Bold - Plain
Tamaño: 12 - 36

PROPIEDADES DE JTABLE



Se insertaron los nombres de las columnas que va tener la tabla y se cambio la propiedad para que no se pueden editar las celdas.



Se cambio la propiedad que permitía que las celdas fueran reorganizadas

PROPIEDADES ADICIONALES (USO DE LIBRERIA FLATLAF)

Se utilizó la librería FlatLaf para aplicar un look and feel personalizado a la interfaz. Este tipo de apariencia solo puede establecerse mediante código y debe aplicarse antes de crear cualquier componente Swing, por lo que se incluyó en el método main. Gracias al uso de algunas propiedades específicas y al administrador de diseño de Swing, algunos componentes como JTextField y JSpinner presentan bordes redondeados. Además, se personalizó la tabla (JTable) con un color en su encabezado y una alternancia de colores entre filas para mejorar la visualización.

Implementación en código:

```
try {  
    FlatLightLaf.setup();  
    UIManager.put( "Component.arc", 999 );  
    UIManager.put( "TextComponent.arc", 999 );  
    UIManager.put("TableHeader.background", new Color(223,223,223));  
    UIManager.put("Table.alternateRowColor", new Color(255,153,153));  
} catch (Exception ex) {  
    System.err.println("No se pudo activar el Look and Feel");  
}
```

PROPIEDADES ADICIONALES (JCALENDAR)

Los componentes de la librería JCalendar funcionan de manera modular, similar a piezas de lego, lo que permite acceder y modificar sus partes internas de forma individual. Teniendo esto en cuenta, en componentes como JDateChooser y JCalendar, que incluyen subcomponentes como JYearChooser o JDateEditor (los cuales contienen campos de texto JTextField), se deshabilitaron dichos campos para evitar que el usuario ingrese caracteres inválidos o especiales, eliminando así la necesidad de validaciones extensas. Esta configuración solo puede hacerse mediante código, por lo que se implementó directamente en la clase controlador en el método iniciar.

Implementación en código:

```
((JTextField)frameMain.getEscogerDiaDeBusqueda().getDateEditor().getUiComponent()).setEnabled(false);
((JTextField) ((JSpinner)frameMain.getCalendario().getYearChooser().getSpinner()).getEditor()).setEnabled(false);
((JTextField) ((JSpinner)frameMain.getEscogerDiaDeBusqueda().getJCalendar().getYearChooser().getSpinner()).
getEditor()).setEnabled(false);
```


CODIGO MODELO

Se crearon tres clases como parte del modelo. La primera es la clase Tarea, que representa las actividades o tareas que el usuario desea registrar. La segunda es la clase Fecha, que define el día en que se realizará cada tarea. Por último, se implementó la clase RepositorioDeTareas, encargada de almacenar, organizar y gestionar todas las tareas creadas.



`jcalendar.modelo`



`Fecha.java`



`Tarea.java`



`TareaRepositorio.java`

CLASE FECHA

Permite crear una instancia de tipo Fecha con los siguientes datos:
dd,mm,aa.



```
public class Fecha {
    private int dd;
    private int mm;
    private int aa;
    public Fecha() {
        Calendar Fecha=Calendar.getInstance();
        this.dd =Fecha.get(Calendar.DAY_OF_MONTH);
        this.mm =(Fecha.get(Calendar.MONTH)+1) ;
        this.aa =Fecha.get(Calendar.YEAR);
    }
    public Fecha(int dd, int mm, int aa) {
        this.dd = dd;
        this.mm = mm;
        this.aa = aa;
    }
    public int getDd() {
        return dd;
    }
    public int getMm() {
        return mm;
    }
    public int getAa() {
        return aa;
    }
}
```

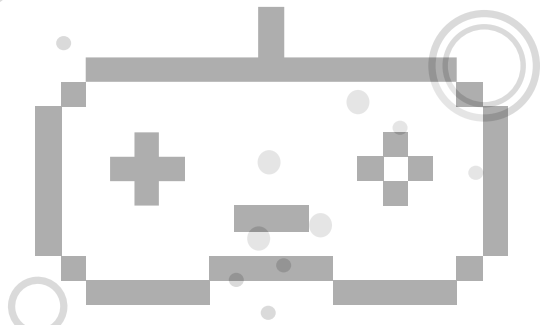
```
    public void setMm(int mm) {
        this.mm = mm;
    }
    public void setAa(int aa) {
        this.aa = aa;
    }
    @Override
    public String toString() {
        return dd + "/" + mm + "/" + aa;
    }
}
```


CLASE TAREA

Permite crear una instancia de tipo Tarea con los siguientes datos:
fecha,nombre,contenido

```
public class Tarea {
    private Fecha fecha;
    private String nombre;
    private String contenido;
    public Tarea() {
        this.fecha=new Fecha();
        this.nombre="";
        this.contenido="";
    }
    public Tarea(String nombre, String contenido) {
        this.nombre = nombre;
        this.contenido = contenido;
    }
    public Tarea(Fecha fecha, String nombre, String contenido) {
        this.fecha = fecha;
        this.nombre = nombre;
        this.contenido = contenido;
    }
    public String getNombre() {
        return nombre;
    }
}
```

```
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public String getContenido() {
    return contenido;
}
public void setContenido(String contenido) {
    this.contenido = contenido;
}
public Fecha getFecha() {
    return fecha;
}
public void setFecha(Fecha fecha) {
    this.fecha = fecha;
}
@Override
public String toString() {
    return "Tarea{" + "fecha=" + fecha.toString() + ", nombre=" + nombre + ", contenido=" + contenido + '}';
}
```





CLASE TAREAREPOSITORIO

Repositorio de tareas con almacenamiento en memoria usando un HashMap.

El repositorio de tareas implementa un sistema de almacenamiento en memoria utilizando un HashMap, donde cada tarea se asocia a una fecha como clave. Este repositorio permite realizar operaciones básicas tipo CRUD:

- Listar: obtener todas las tareas almacenadas.
- Cargar: insertar una nueva tarea o actualizar una existente si ya hay una registrada para la misma fecha.
- Buscar: consultar una tarea específica utilizando su fecha.
- Eliminar: borrar una tarea, ya sea mediante la fecha o directamente por el objeto de la tarea.

¿QUÉ ES Y CÓMO FUNCIONA UN HASHMAP?

Un HashMap es una estructura de datos utilizada para almacenar pares clave-valor. Funciona de manera similar a un diccionario, donde a cada clave se le asocia un valor específico. Por ejemplo, si la clave es un nombre y el valor un número de teléfono, se puede acceder rápidamente al número con solo conocer el nombre.

Un HashMap permite gestionar datos mediante las siguientes operaciones:

- put(clave, valor): Guarda o actualiza un valor asociado a una clave.
- get(clave): Recupera el valor vinculado a una clave.
- remove(clave): Elimina una clave y su valor.
- containsKey(clave): Verifica si una clave existe.
- isEmpty(): Comprueba si el mapa está vacío.

También se puede obtener el conjunto de claves (keySet()) o de valores (values()).



```
public class TareaRepositorio {  
    private final HashMap<String, Tarea> tareas;  
    public TareaRepositorio() {  
        tareas = new HashMap<>();  
    }  
    public List<Tarea> listar() {  
        return new ArrayList<>(tareas.values());  
    }  
    public void cargar(Tarea tarea) {  
        tareas.put(tarea.getFecha().toString(), tarea);  
    }  
    public Tarea buscar(String fecha) {  
        return tareas.get(fecha);  
    }  
    public Tarea buscar(Tarea tarea) {  
        return tareas.get(tarea.getFecha().toString());  
    }  
    public boolean eliminar(String fecha) {  
        return tareas.remove(fecha) != null;  
    }  
    public boolean eliminar(Tarea tarea) {  
        return tareas.remove(tarea.getFecha().toString()) != null;  
    }  
}
```

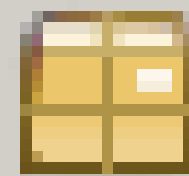
En este caso la clave es la fecha
y el valor un objeto de tipo
tarea

DISEÑO DE LA VISTA

Se optó por una vista que incorpora diferentes componentes con el objetivo de cumplir con la propuesta de código y demostrar el uso de los controles disponibles en la librería JCalendar. A continuación, se listan los componentes utilizados en la interfaz junto con la cantidad correspondiente de cada uno:

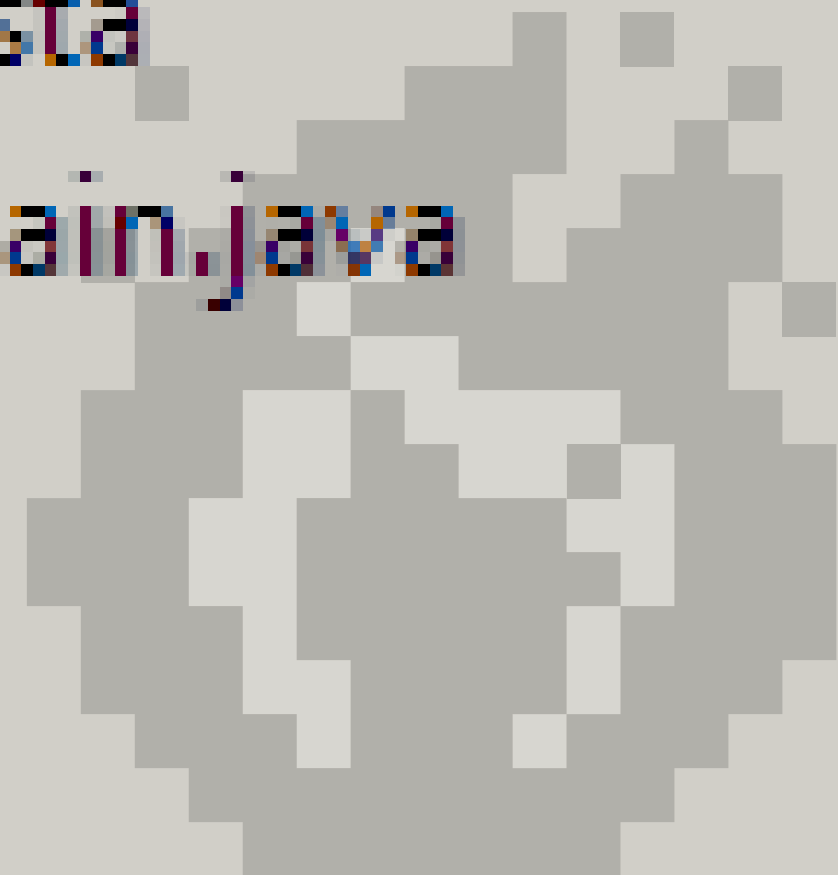
Componentes utilizados:

- **JPanel**
Cantidad: 5
- **JDateChooser**
Cantidad: 1
- **JCalendar**
Cantidad: 1
- **JButton**
Cantidad: 5
- **JTable**
Cantidad: 1
- **JLabel**
Cantidad: 9
- **JTextArea**
Cantidad: 1
- **JTabbedPane**
Número de pestañas: 3
- **TextField**
Cantidad: 1



jcalendar.vista

FrameMain.java



PESTAÑA CALENDARIO

JCalendar

Calendario

Herramientas

Tareas

julio

2025

Seleccionar día

| | dom | lun | mar | mié | jue | vie | sáb |
|----|-----|-----|-----|-----|-----|-----|-----|
| 27 | | | 1 | 2 | 3 | 4 | 5 |
| 28 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 29 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 30 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 31 | 27 | 28 | 29 | 30 | 31 | | |

1/7/2025

Nombre:

Cargar

Contenido:

Eliminar

Esta pestaña permite al usuario seleccionar la fecha en la que desea programar una tarea. Luego de seleccionar la fecha, el usuario puede ingresar el nombre de la tarea y una descripción. La información registrada se guarda en el repositorio de tareas.

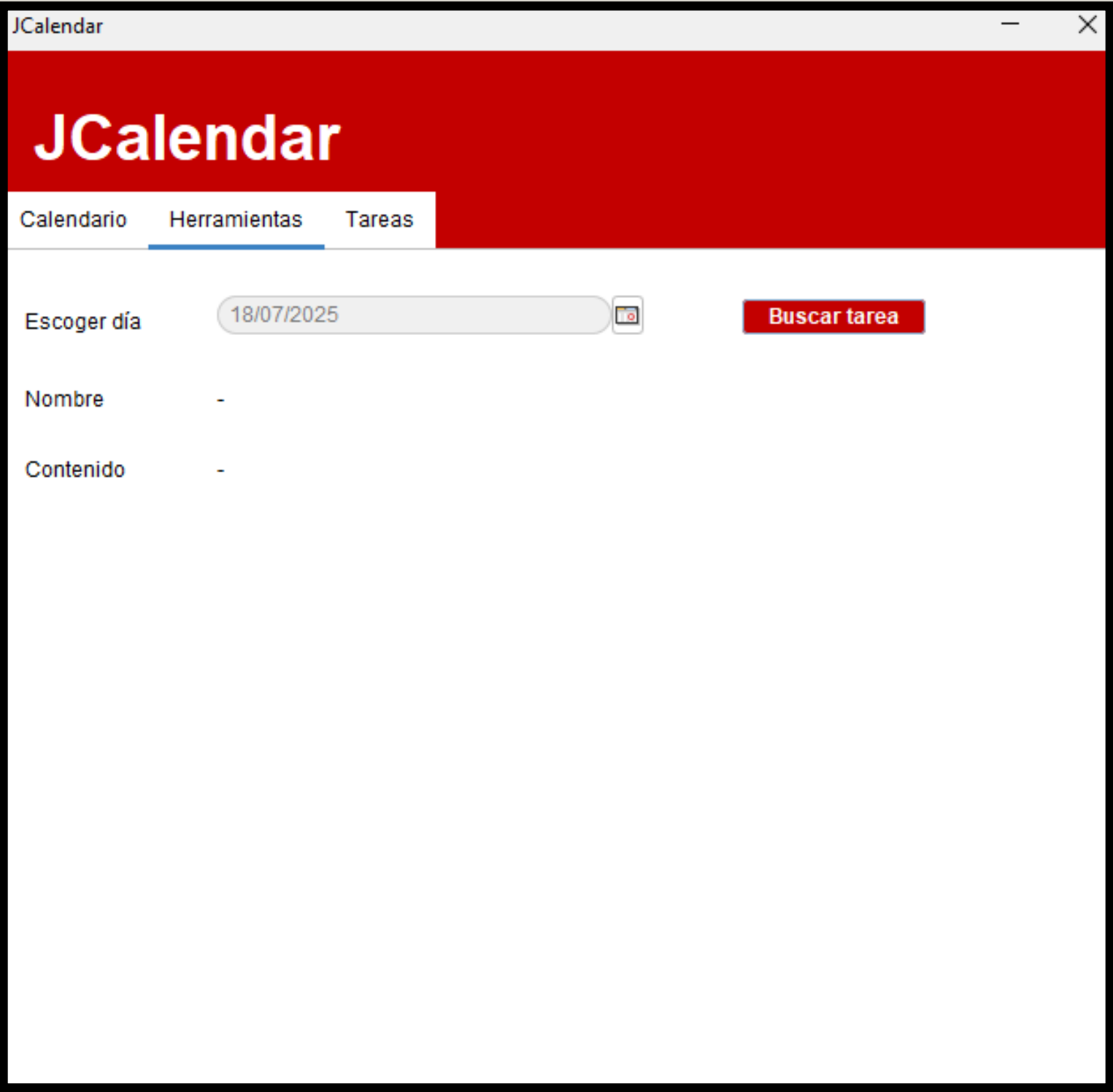
Adicionalmente, si ya existe una tarea asociada a la fecha seleccionada, el usuario tiene la opción de actualizar su contenido y nombre, o eliminarla si lo considera necesario.

Componentes utilizados en la pestaña:

- **JPanel**
Cantidad: 2
- **JCalendar**
Cantidad: 1
- **JButton**
Cantidad: 3
- **JLabel**
Cantidad: 3
- **JTextArea**
Cantidad: 1
- **JTextField**
Cantidad: 1



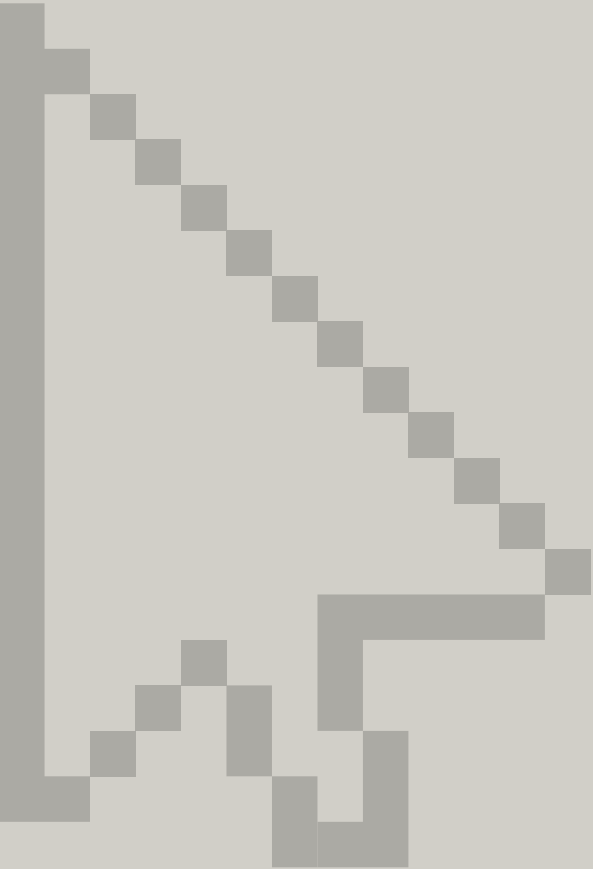
PESTAÑA HERRAMIENTAS



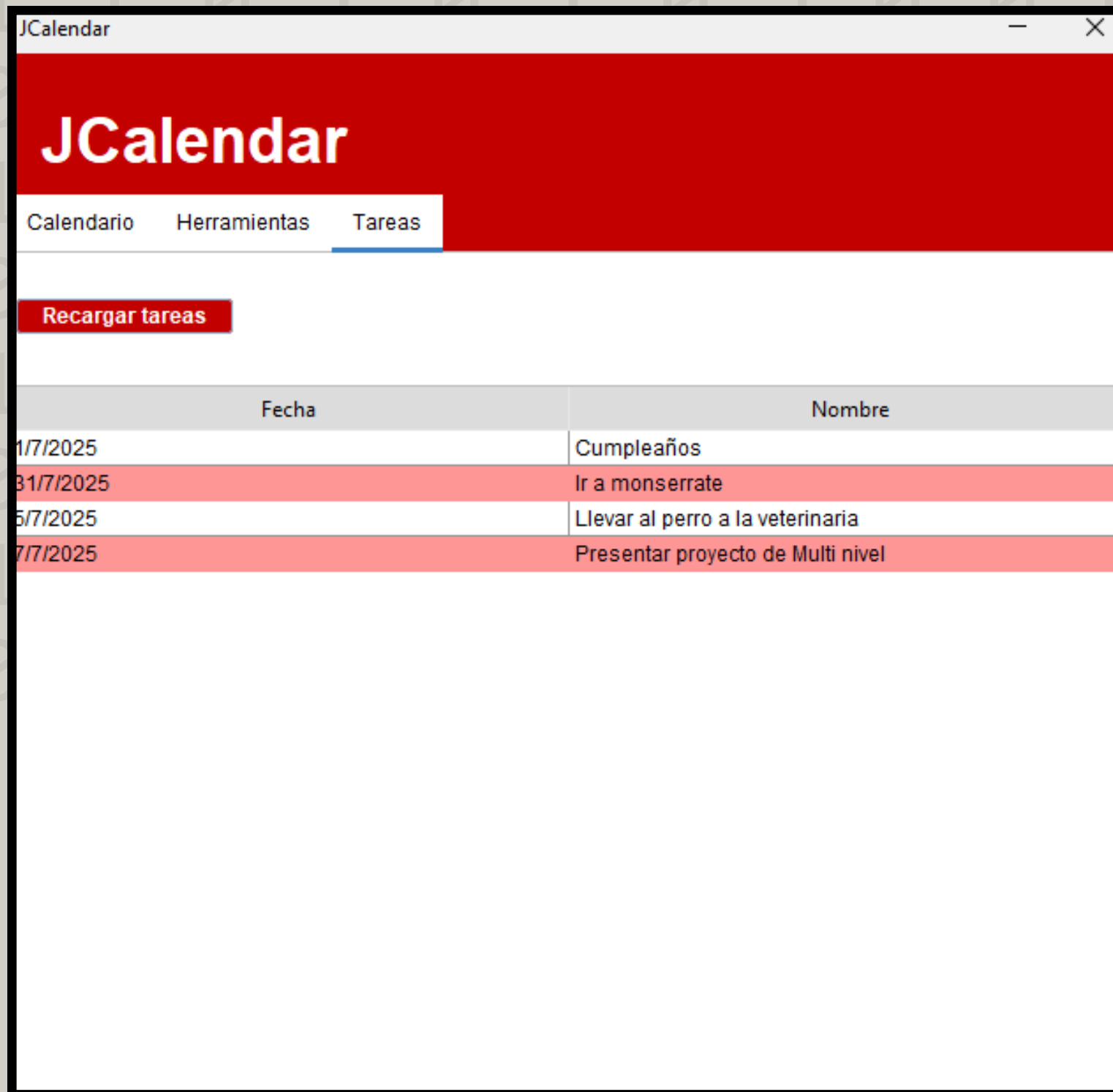
Esta pestaña permite al usuario buscar una tarea registrada en el repositorio mediante el uso de los componentes `JDateChooser` y `JButton`. Si se encuentra una tarea correspondiente a la fecha seleccionada, sus datos se muestran en los respectivos `JLabel` de la interfaz.

Componentes utilizados en la pestaña:

- **JPanel**
Cantidad: 1
- **JDateChooser**
Cantidad: 1
- **JButton**
Cantidad: 1
- **JLabel**
Cantidad: 5



PESTAÑA TAREAS



| Fecha | Nombre |
|-----------|-----------------------------------|
| 1/7/2025 | Cumpleaños |
| 31/7/2025 | Ir a monserate |
| 5/7/2025 | Llevar al perro a la veterinaria |
| 7/7/2025 | Presentar proyecto de Multi nivel |

Esta pestaña permite visualizar las tareas registradas en el repositorio mediante el componente JTable. Además, cuenta con un botón de recarga que actualiza la tabla en caso de que se eliminen o agreguen nuevas tareas.

Componentes utilizados:

- **JPanel**
Cantidad: 1
- **JButton**
Cantidad: 1
- **JTable**
Cantidad: 1



CODIGO CONTROLADOR

Clase principal del controlador de eventos de la aplicación. Se encarga de organizar la interacción entre la vista y el modelo, escuchando los eventos de los botones y ejecutando las acciones correspondientes. El controlador cuenta con los siguientes datos: frameMain, repositorio.

```
/**
 *Constructor basico de controlador que instacia los atributos de la clase
 *con sus respectivos constructores basicos
 */
public Controlador() {
    frameMain = new FrameMain();
    repositorio = new TareaRepositorio();
}

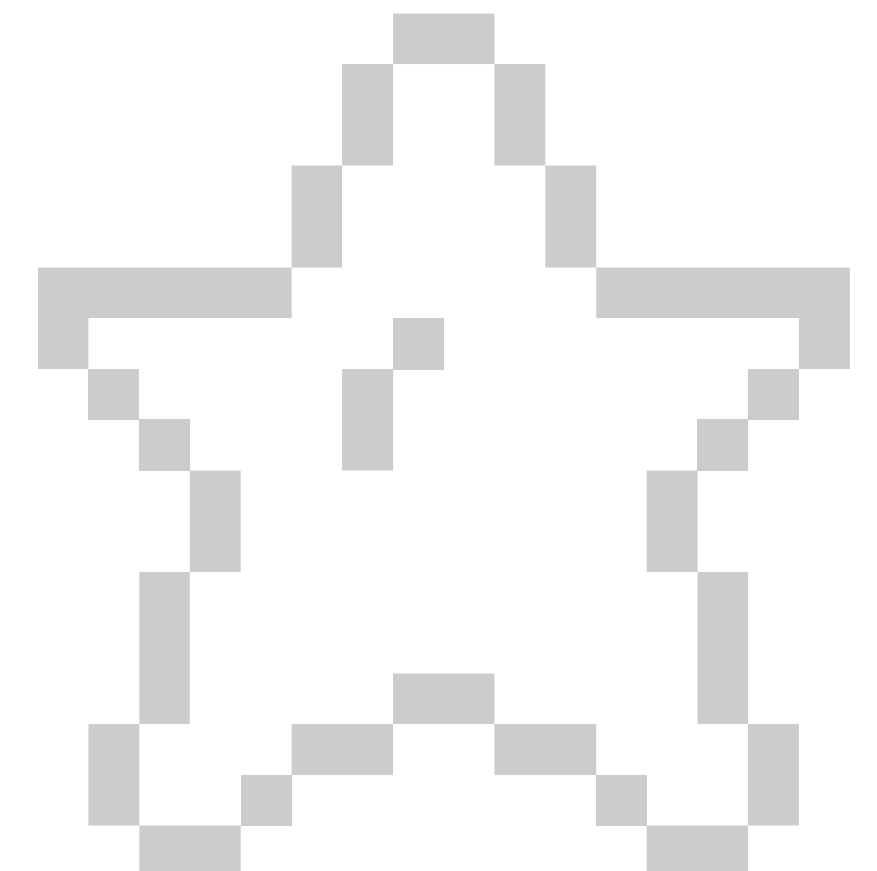
/**
 *Metodo que permite iniciar la conexión entre el modelo y la vista,
 *añadiendo también actionListener a los componentes de la interfaz
 *que desencadenen eventos, tambien se desabilita los editores en los
 *componentes de Jcalendar, calendar al ser un componente con varios componentes
 *este se puede desarmar y configurar cada uno por aparte.
 */
public void iniciar() {
    frameMain.setTitle("JCalendar");
    frameMain.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frameMain.setLocationRelativeTo(null);
    iniciarListener(frameMain, this);
    frameMain.getPanelTareas().setVisible(false);
    ((JTextField) frameMain.getEscogerDiaDeBusqueda().getDateEditor().getUiComponent()).setEnabled(false);
    ((JTextField) ((JSpinner) frameMain.getCalendario().getYearChooser().getSpinner()).getEditor()).setEnabled(false);
    ((JTextField) ((JSpinner) frameMain.getEscogerDiaDeBusqueda().getJCalendar().getYearChooser().getSpinner()).
    getEditor()).setEnabled(false);
    frameMain.setVisible(true);
}
```



jcalendar.controlador



Controlador.java



```
/**
 *Permite gestionar los diferentes eventos que se presenten en la interaccion
 *del usuario con la interfaz.
 * @param e Evento
 */
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource().equals(frameMain.getBtnSeleccionarDia())) {
        accionSeleccionarDia();
    }
    if (e.getSource().equals(frameMain.getBtnCargar())) {
        accionCargar();
    }
    if (e.getSource().equals(frameMain.getBtnEliminar())) {
        accionEliminar();
    }
    if (e.getSource().equals(frameMain.getBtnBuscarTarea())) {
        accionBuscarTarea();
    }
    if (e.getSource().equals(frameMain.getBtnRecargarTabla())) {
        accionRecargarTabla();
    }
}
```

```

/**
 * Convierte datos de tipo calendar extraidos de un JDateChooser o JCalendar,
 * en un dato de tipo Fecha
 * @param Obj componente de la libreria Jcalendar
 * @return Fecha Retorna un objeto de tipo Fecha
 */
public Fecha Tomar_Fecha(Object Obj) {
    Calendar fecha=null;
    if(Obj instanceof JDateChooser){
        fecha = ((JDateChooser)Obj).getCalendar();
    }else if(Obj instanceof JCalendar ){
        fecha = ((JCalendar)Obj).getCalendar();
    }
    return new Fecha(
        fecha.get(Calendar.DAY_OF_MONTH),
        (1 + fecha.get(Calendar.MONTH)),
        fecha.get(Calendar.YEAR)
    );
}

/**
 * Metodo que permite la seleccion de una fecha para cargar una tarea
 * si la tarea ya existe dentro del panel de tareas muestra los datos de
 * la tarea existente si no los muestra vacios
 */
private void accionSeleccionarDia() {
    limpiarFormulario(this.frameMain.getPanelTareas());
    frameMain.getPanelTareas().setVisible(true);
    frameMain.getTextoFecha().setText(Tomar_Fecha(frameMain.getCalendario()).toString());
    Tarea tarea = repositorio.buscar(Tomar_Fecha(frameMain.getCalendario()).toString());
    if (tarea == null) {
        return;
    }
    frameMain.getNombreTarea().setText(tarea.getNombre());
    frameMain.getContenidoTarea().setText(tarea.getContenido());
}

```

```

/**
 * Metodo que permite elimianr una tarea, donde primero
 * verifica que exista una tarea en esa fecha, si existe la elimina
 * si no muestra error y sale del metodo
 */
private void accionEliminar() {
    if (!repositorio.eliminar(Tomar_Fecha(frameMain.getCalendario()).toString())) {
        JOptionPane.showMessageDialog(
            frameMain, "Error: no había taréa asignada",
            "Error", JOptionPane.ERROR_MESSAGE);
        return;
    }

    JOptionPane.showMessageDialog(frameMain, "Se eliminó la tarea",
        "Good", JOptionPane.PLAIN_MESSAGE);

    frameMain.getPanelTareas().setVisible(false);
}

```

```

/**
 * Metodo que permite insertar una tarea en el repositorio,
 * si inserta la tarea con datos incompletos muestra error y se sale del metodo
 * sino crea un dato de tipo tarea y la carga en el repositorio
 */
private void accionCargar() {

    if (frameMain.getNombreTarea().getText().isBlank()
        || frameMain.getContenidoTarea().getText().isBlank()) {
        JOptionPane.showMessageDialog(
            frameMain,
            "Error: no agregaste parte de la informacion",
            "Error",
            JOptionPane.ERROR_MESSAGE);
        return;
    }

    Fecha fecha=Tomar_Fecha(frameMain.getCalendario());
    Tarea tarea = new Tarea(
        fecha,
        frameMain.getNombreTarea().getText(),
        frameMain.getContenidoTarea().getText()
    );

    repositorio.cargar(tarea);
    JOptionPane.showMessageDialog(
        frameMain, "Se cargó la tarea perfectamente", "Good",
        JOptionPane.PLAIN_MESSAGE);
    frameMain.getPanelTareas().setVisible(false);
}

```



```
/**
 * Metodo que permite buscar la tarea a traves de la fecha
 * donde primero se toma la fecha seleccionada del componente de JCalendar
 * para luego enviar los datos de esa fecha como una cadena de caracteres al
 * metodo buscar del repositorio sino existe ninguna tarea se muestra en pantalla
 * que no existe ninguna tarea programada para esa respectiva fecha, pero si existe
 * se muestran sus datos.
 */
private void accionBuscarTarea() {
    limpiarFormulario(this.frameMain.getPanelTareas());
    Tarea tarea = repositorio.buscar(Tomar_Fecha(frameMain.getEscogerDiaDeBusqueda()).toString());
    if(tarea!=null){
        System.out.println(tarea.toString());
        frameMain.getNombreBuscado().setText(tarea.getNombre());
        frameMain.getContenidoBuscado().setText(tarea.getContenido());
    }else{
        JOptionPane.showMessageDialog(
            frameMain,
            "No existe ninguna tarea programada para "+Tomar_Fecha(frameMain.getEscogerDiaDeBusqueda()).toString()
        );
    }
}
```



```
/**
 * Metodo que permite actualizar la tabla de tareas,
 * si no existe ninguna tarea registrada muestra en pantalla
 * no hay ninguna tarea registrada
 */
private void accionRecargarTabla() {
    if(repositorio.listar().isEmpty()){
        JOptionPane.showMessageDialog(
            frameMain,
            "No hay ninguna tarea registrada."
        );
    }else{
        DefaultTableModel plantilla = (DefaultTableModel) frameMain.getTablaTareas().getModel();

        plantilla.setRowCount(0); // Limpia la tabla

        for (Tarea tarea : repositorio.listar()) {

            Object[] datos = {
                tarea.getFecha().toString(),
                tarea.getNombre(),};

            plantilla.addRow(datos);
        }
    }
}
```

```

/**
 * Limpia todos los campos de texto en el contenedor dado.
 * @param container contenedor que puede tener JTextField y otros
 * componentes
 */
public static void limpiarFormulario(Container container) {
    for (Component c : container.getComponents()) {
        if (c instanceof JTextField) {
            ((JTextField) c).setText("");
        } else if (c instanceof JTextArea) {
            ((JTextArea) c).setText("");
        } else if (c instanceof Container hijoContenedor) {
            limpiarFormulario(hijoContenedor);
        }
    }
}

```

```

/**
 * Asocia este controlador como listener de todos los botones y menús del
 * contenedor.
 *
 * @param container contenedor base (ej. JFrame o JPanel) donde buscar
 * componentes
 * @param listener el ActionListener asociado, sera el que escucha los
 * eventos de los componentes.
 */
public static void iniciarListener(Container container, ActionListener listener) {
    for (Component c : container.getComponents()) {
        if (c instanceof JButton) {
            ((JButton) c).addActionListener(listener);
        } else if (c instanceof JMenu) {
            for (int i = 0; i < ((JMenu) c).getItemCount(); i++) {
                if (((JMenu) c).getItem(i) != null) {
                    ((JMenu) c).getItem(i).addActionListener(listener);
                }
            }
        } else if (c instanceof Container) {
            iniciarListener((Container) c, listener);
        }
    }
}

```

RESULTADOS

CARGAR TAREA

JCalendar

Calendario Herramientas Tareas

julio 2025 Seleccionar día

| | dom | lun | mar | mié | jue | vie | sáb |
|----|-----|-----|-----|-----|-----|-----|-----|
| 27 | | | 1 | 2 | 3 | 4 | 5 |
| 28 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 29 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 30 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 31 | 27 | 28 | 29 | 30 | 31 | | |

3/7/2025

Nombre: Cumpleaño santiago Cargar

Contenido: cumpleaños en la casa de santiago
Nota: Llevar regalo Eliminar



Calendar

JCalendar

Calendario Herramientas Tareas

julio 2025 Seleccionar día

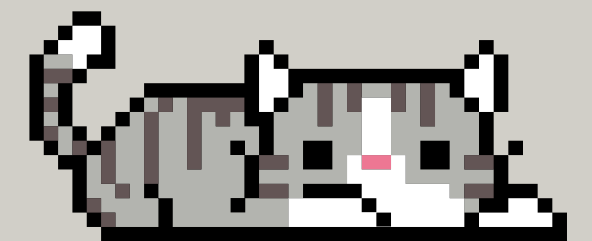
| | dom | lun | mar | mié | jue | vie | sáb |
|----|-----|-----|-----|-----|-----|-----|-----|
| 27 | | | 1 | 2 | 3 | 4 | 5 |
| 28 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 29 | 13 | 14 | 15 | | | | |
| 30 | 20 | 21 | 22 | | | | |
| 31 | 27 | 28 | 29 | | | | |

3/7/2025

Nombre: Cumpleaño santiago Cargar

Contenido: cumpleaños en la casa de santiago
Nota: Llevar regalo Eliminar

Good
Se cargó la tarea perfectamente
OK



ELIMINAR TAREA

JCalendar

CalendarioHerramientasTareas

julio2025Seleccionar día

| | dom | lun | mar | mié | jue | vie | sáb |
|----|-----|-----|-----|-----|-----|-----|-----|
| 27 | | | 1 | 2 | 3 | 4 | 5 |
| 28 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 29 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 30 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 31 | 27 | 28 | 29 | 30 | 31 | | |

3/7/2025

Nombre: Cargar

Contenido:

cumpleaños en la casa de santiago
Nota: Llevar regalo

 Eliminar



JCalendar

CalendarioHerramientasTareas

julio2025Seleccionar día

| | dom | lun | mar | mié | jue | vie | sáb |
|----|-----|-----|-----|-----|-----|-----|-----|
| 27 | | | 1 | 2 | 3 | 4 | 5 |
| 28 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 29 | 13 | 14 | 15 | | | | |
| 30 | 20 | 21 | 22 | | | | |
| 31 | 27 | 28 | 29 | | | | |

3/7/2025

Nombre: Cargar

Contenido:

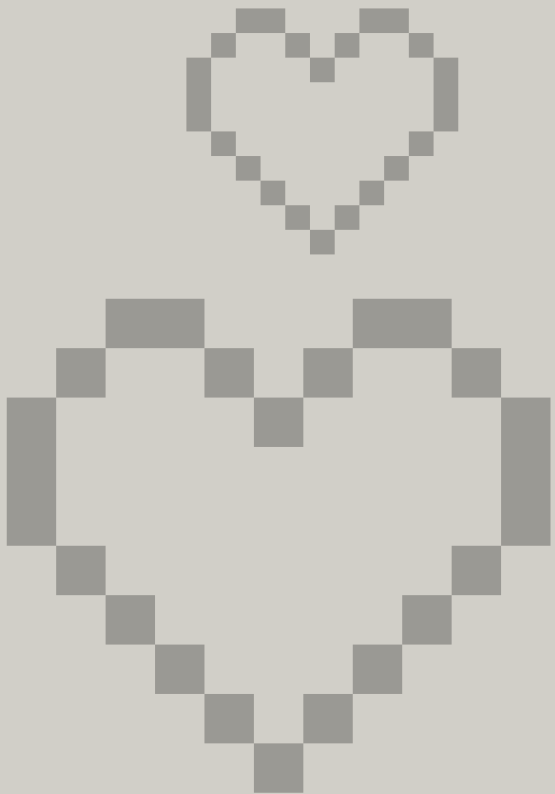
cumpleaños en la casa de santiago
Nota: Llevar regalo

 Eliminar

Good

Se eliminó la tarea

OK



BUSCAR TAREA

JCalendar

Calendario

Herramientas

Tareas

Escoger día

3/07/2025

Buscar tarea

Nombre

-

Contenido

-

julio

2025

dom

lun

mar

mié

jue

vie

sáb

27

28

29

30

31

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31



JCalendar

Calendario

Herramientas

Tareas

Escoger día

3/07/2025

Buscar tarea

Nombre

Cumpleañ santiago

Contenido

cumpleaños en la casa de santiago Nota: Llevar regalo



VER TAREAS

JCalendar

JCalendar

CalendarioHerramientasTareas

Recargar tareas

| Fecha | Nombre |
|------------|---------------------|
| 17/12/2025 | Cumpleaños santiago |
| 22/7/2025 | Salida a cine |

REFERENCIAS LIBRERIAS UTILIZADAS

FormDev. (s.f.). FlatLaf – Flat Look and Feel for Java Swing. <https://www.formdev.com/flatlaf/>
Toedter, K. (s.f.). JCalendar – Java Date Choosers and Calendars. <https://toedter.com/jcalendar/>

REPOSITORIO DONDE SE PUEDE ENCONTRAR EL CODIGO REALIZADO

Rodriguez, Peraza. (2025). JCalendarPrueba [Código fuente]. GitHub. <https://github.com/XhuaSpy/jcalendar>