

1. ¿Cuál es el componente central de la ventana principal? Escribe la línea completa en la que se define.

```
frame = new JFrame();
frame.getContentPane().setBackground(new Color(192, 192, 192));
frame.setBounds(100, 100, 450, 300);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().setLayout(null);
```

2. ¿Dónde se define el tamaño de la ventana principal? Escribe el bloque completo que lo configura.

```
frame.setBounds(100, 100, 450, 300);
```

El método `setBounds` es el encargado de darle la forma que deseemos.

3. ¿Qué clases utilizas para definir los diferentes componentes que has añadido en la aplicación? ¿Cómo indicas el nombre que tendrá cada componente, el cual usarás para referirte al componente en el código?

```
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
```

El nombre por el cual identificaremos al componente, se lo diremos en la declaración y definición de dicho componente, aquí le proporciono el código donde se cambiaría el nombre identificativo:

```
JButton btnAñadir = new JButton("Añadir tarea");
```

4. ¿Dónde y cómo se indica el texto que se va a mostrar en un label o en un botón? Escribe una línea de ejemplo que hayas encontrado en tu archivo.

```
JButton btnActualiza = new JButton("Actualizar Tarea");
```

El texto entre comilla sería el texto que mostraría el botón al iniciar la aplicación

5. Indica cómo se crea y se nombra el componente. Pon ejemplo del código.
6. Indica cómo se crea un elemento nuevo en la lista. Pon ejemplo del código.

```
DefaultListModel<String> tareasAñadir = new DefaultListModel<>();
```

```
String titulo = JOptionPane.showInputDialog(frame, "TÍTULO DE LA ACTIVIDAD QUE DESEA AÑADIR");
tareasAñadir.addElement(titulo);
```

7. Indica cómo se elimina un elemento de la lista. Pon ejemplo del código.

```
String tareaEliminar = JOptionPane.showInputDialog(frame,"TITULO DE LA ACTIVIDAD A ELIMINAR");

boolean bandera = false;

for (int i = 0; ((i < tareasAniadir.getSize()) && bandera == false); i++) {

    if(tareaEliminar.equals( tareasAniadir.get(i)))
    {
        bandera = true;
        tareasAniadir.remove(i);
    }
}
```

8. Indica cómo se modifica un elemento de la lista. Pon ejemplo del código.

```
String tareaActualizar = JOptionPane.showInputDialog(frame,"TITULO DE LA ACTIVIDAD A
ACTUALIZAR");

boolean bandera = false;

for (int i = 0; ((i < tareasAniadir.getSize()) && bandera == false); i++) {

    if(tareaActualizar.equals( tareasAniadir.get(i)))
    {
        bandera = true;
        String resp = JOptionPane.showInputDialog(frame,"¿DESEA ACTUALIZAR EL TITULO ?");
        if(resp.equals("si")|| resp.equals("Si")|| resp.equals("SI")) {
            String titNuevo = JOptionPane.showInputDialog(frame,"DIME EL NUEVO TITULO QUE
DESEAS");
            tareasAniadir.set(i, titNuevo);

        } else
            JOptionPane.showMessageDialog(frame, "OPERACIÓN CANCELADA");
    }
}
```

9. ¿Qué opción de escritura y lectura del archivo que contiene la información has elegido?

*Para la escritura:*

Cuando el usuario añade una tarea, se utiliza `FileWriter` con el segundo argumento como `true`, lo que permite agregar nuevas tareas sin sobrescribir las anteriores. Luego, `PrintWriter` escribe la tarea en el archivo, añadiendo una nueva línea después de cada tarea.

*Para la lectura:*

el sistema de almacenamiento guarda las tareas añadidas en el archivo, lo que permite actualizarlas o eliminarlas y luego volver a escribir el archivo completo cuando se realizan modificaciones.

10. ¿Cómo escribes y lees datos en el archivo? Indica que métodos utilizas en la interfaz para llamar a los métodos de manipulación de ficheros. ¿Qué tipo de métodos son?

En la manera que yo lo he hecho no necesito llamar a ningún método he aprovechado el método que se nos crea al hacer una acción sobre el botón y ahí he implementado pues que la tarea que añada la añada también a un fichero vacío que he creado y a la igual manera he implementado el actualizar y el eliminar en el fichero

11. Adjunta un ejemplo del código en el que realices una escritura de datos en el fichero.

```
btnAniadir.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        String titulo = JOptionPane.showInputDialog(frame,"TITULO DE LA ACTIVIDAD QUE DESEA AÑADIR");

        tareasAniadir.addElement(titulo);

        //String desc = JOptionPane.showInputDialog(frame,"DAME UNA BREVE DESCRIPCIÓN SOBRE DICHA ACTIVIDAD");

        try(FileWriter fw = new FileWriter("C:\\Practica1.4\\Prueba.txt", true)){
            PrintWriter pw = new PrintWriter(fw);

            pw.write(titulo);
            pw.write("\n");

            pw.close();
            fw.close();

        } catch(Exception ex) {
        }

    }

});
```

12. Adjunta un ejemplo del código en el que realices una lectura de datos en el fichero.

```
btnEliminar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String tareaEliminar = JOptionPane.showInputDialog(frame,"TITULO DE LA ACTIVIDAD A ELIMINAR");

        boolean bandera = false;

        for (int i = 0; ((i < tareasAniadir.getSize()) && bandera == false); i++) {

            if(tareaEliminar.equals( tareasAniadir.get(i)))
            {
                bandera = true;
                tareasAniadir.remove(i);
                try(FileWriter fw = new FileWriter("C:\\Practica1.4\\Prueba.txt")){
                    PrintWriter pw = new PrintWriter(fw);

                    for (int a = 0; a < tareasAniadir.getSize(); a++) {
                        String tarea = tareasAniadir.getElementAt(a);
                        pw.write(tarea);
                        pw.write("\n");
                    }

                }

            }

        }

    }

});
```

```

        pw.close();
        fw.close();

    } catch (Exception ex) {

    }

}

}

});

```

13. ¿Has utilizado un fichero en formato CSV? ¿Qué estructura has utilizado para almacenar la información? Adjunta un ejemplo del fichero.

No, he usado un archivo .txt en el que almaceno todas las tareas que el usuario añade de manera que en cada línea aparecerá el título de la tarea, cada vez que le usuario actualiza el título o elimina la tarea automáticamente el fichero se actualizará a los datos recientes que aparecen en el JList

