

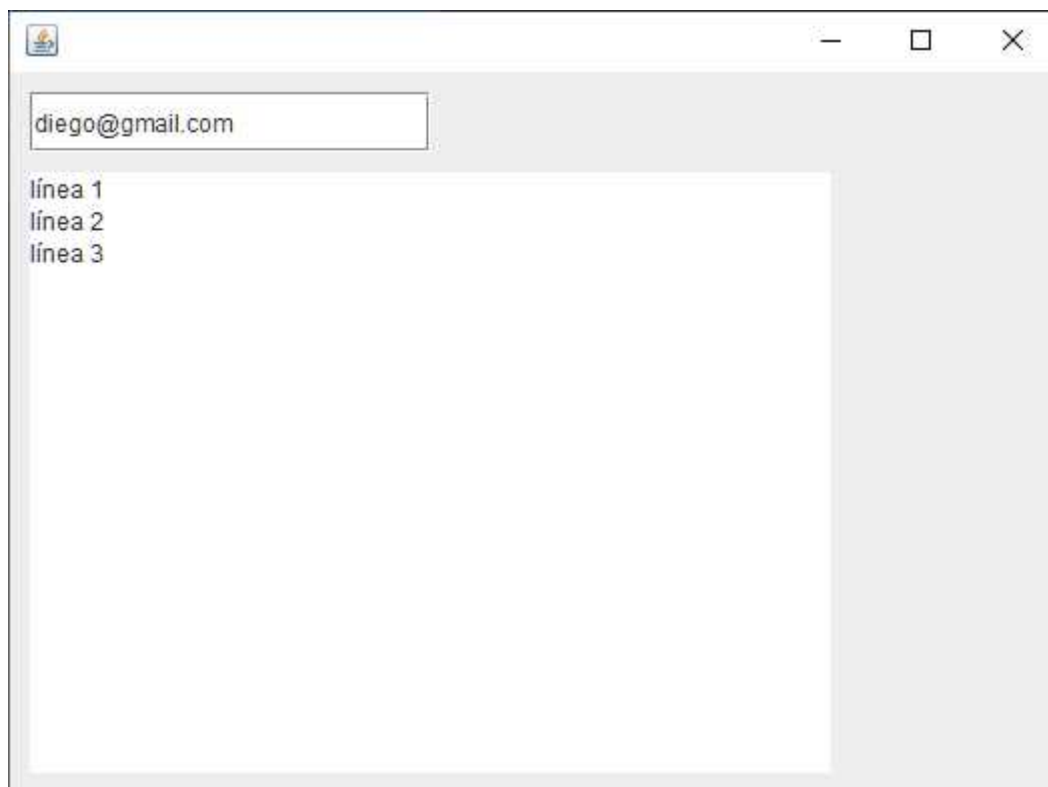
35 - Swing - JTextArea

[Listado completo de tutoriales](#)

El control de tipo JTextArea permite ingresar múltiples líneas, a diferencia del control de tipo JTextField.

Problema 1:

Confeccionar un programa que permita ingresar un mail en un control de tipo JTextField y el cuerpo del mail en un control de tipo JTextArea.



Programa:

[Ver video](#)

```
import javax.swing.*;
public class Formulario extends JFrame{
    private JTextField textfield1;
    private JTextArea textareal;
    public Formulario() {
        setLayout(null);
        textfield1=new JTextField();
        textfield1.setBounds(10,10,200,30);
        add(textfield1);
        textareal=new JTextArea();
```

```

        textarea1.setBounds(10, 50, 400, 300);
        add(textarea1);
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,540,400);
        formulario1.setVisible(true);
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

Como vemos crear un control de tipo JTextArea es similar a la creación de controles de tipo JTextField:

```

        textarea1=new JTextArea();
        textarea1.setBounds(10,50,400,300);
        add(textarea1);

```

El inconveniente que tiene este control es que si ingresamos más texto que el que puede visualizar no aparecen las barras de scroll y no podemos ver los caracteres tipeados.

Para salvar el problema anterior debemos crear un objeto de la clase JScrollPane y añadir en su interior el objeto de la clase JTextArea, luego el programa definitivo es el siguiente:

```

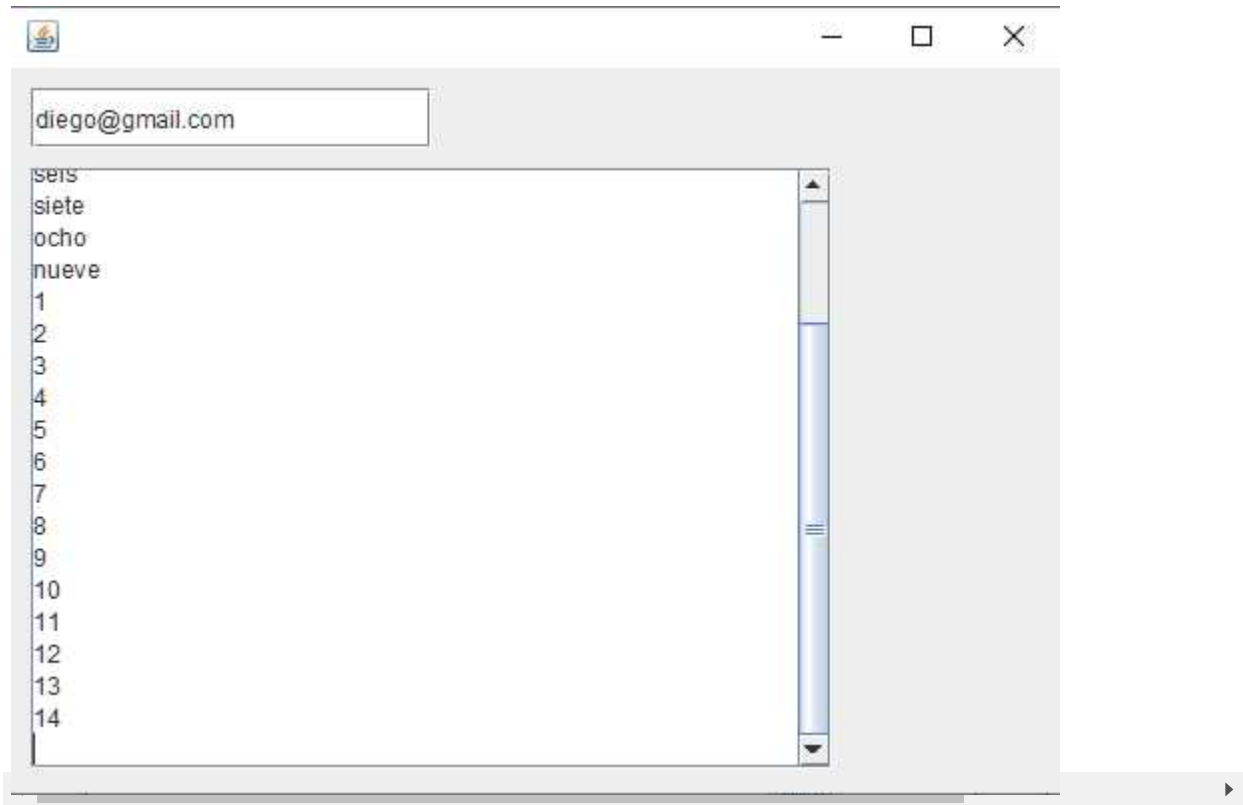
import javax.swing.*;

public class Formulario extends JFrame{
    private JTextField textfield1;
    private JScrollPane scrollpanel1;
    private JTextArea textarea1;
    public Formulario() {
        setLayout(null);
        textfield1=new JTextField();
        textfield1.setBounds(10,10,200,30);
        add(textfield1);
        textarea1=new JTextArea();
        scrollpanel1=new JScrollPane(textarea1);
        scrollpanel1.setBounds(10,50,400,300);
        add(scrollpanel1);
    }

    public static void main(String[] ar) {
        Formulario formulario1=new Formulario();
        formulario1.setBounds(0,0,540,400);
    }
}

```

```
        formulario1.setVisible(true);  
        formulario1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```



Declaramos los dos objetos:

```
private JScrollPane scrollpane1;  
private JTextArea textarea1;
```

Primero creamos el objeto de la clase JTextArea:

```
textarea1=new JTextArea();
```

Seguidamente creamos el objeto de la clase JScrollPane y le pasamos como parámetro el objeto de la clase JTextArea:

```
scrollpane1=new JScrollPane(textarea1);
```

Definimos la posición y tamaño del control de tipo JScrollPane (y no del control JTextArea):

```
scrollpane1.setBounds(10,50,400,300);
```

Finalmente añadimos el control de tipo JScrollPane al JFrame:

```
add(scrollpane1);
```

Problema 2:

Confeccionar un programa que permita ingresar en un control de tipo JTextArea una carta. Luego al presionar un botón mostrar un mensaje si la carta contiene el String "argentina".

Programa:

[Ver video](#)

```
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener {
    private JScrollPane scrollpane1;
    private JTextArea textareal;
    private JButton boton1;
    public Formulario() {
        setLayout(null);
        textareal=new JTextArea();
        scrollpane1=new JScrollPane(textareal);
        scrollpane1.setBounds(10,10,300,200);
        add(scrollpane1);
        boton1=new JButton("Verificar");
        boton1.setBounds(10,260,100,30);
        add(boton1);
        boton1.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource()==boton1) {
            String texto=textareal.getText();
            if (texto.indexOf("argentina")!= -1) {
                setTitle("Si contiene el texto \"argentina\"");
            } else {
                setTitle("No contiene el texto \"argentina\"");
            }
        }
    }

    public static void main(String[] ar) {
        Formulario formuariol=new Formulario();
        formuariol.setBounds(0,0,400,380);
        formuariol.setVisible(true);
        formuariol.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

```
}  
}
```

Cuando se presiona el botón se ejecuta el método `actionPerformed` y procedemos a extraer el contenido del control `TextArea` a través del método `getText`:

```
String texto=textarea1.getText();
```

Luego mediante el método `indexOf` de la clase `String` verificamos si el `String` "argentina" está contenido en la variable `texto`:

```
if (texto.indexOf("argentina")!= -1) {  
    setTitle("Si contiene el texto \"argentina\"");  
} else {  
    setTitle("No contiene el texto \"argentina\"");  
}
```

Si queremos introducir una comilla doble dentro de un `String` de Java debemos antecederle la barra invertida (luego dicho caracter no se lo considera parte del `String`):

```
setTitle("Si contiene el texto \"argentina\"");
```

Problema propuesto

1. Disponer dos controles de tipo `JTextArea`, luego al presionar un botón verificar si tienen exactamente el mismo contenido.

[Ver video](#)

[Solución](#)

Retornar