

Eclipse. Entorno gráfico

Instalando plugins

Un plugin es un programa que añade funcionalidades específicas a otro programa que hace de programa principal. Su presencia es muy habitual en los navegadores web, en sistemas de reproducción de música y audio, en CMS y en herramientas de desarrollo como Eclipse.

Los plugins no son actualizaciones, sino propiedades añadidas a programas originales.

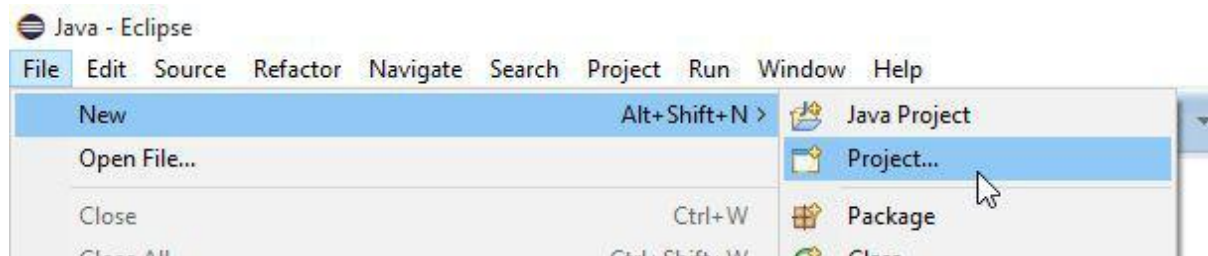
Plugin WindowBuilder

Eclipse no dispone de una interfaz gráfica de usuario para crear aplicaciones con ventanas, para ello será necesario instalar plugins. Uno de los plugins más populares es el WindowBuilder, desarrollado por Google y que nos permite desarrollar de forma rápida y cómoda la GUI de las aplicaciones Java.

Vamos a crear un proyecto con entorno gráfico paso a paso.

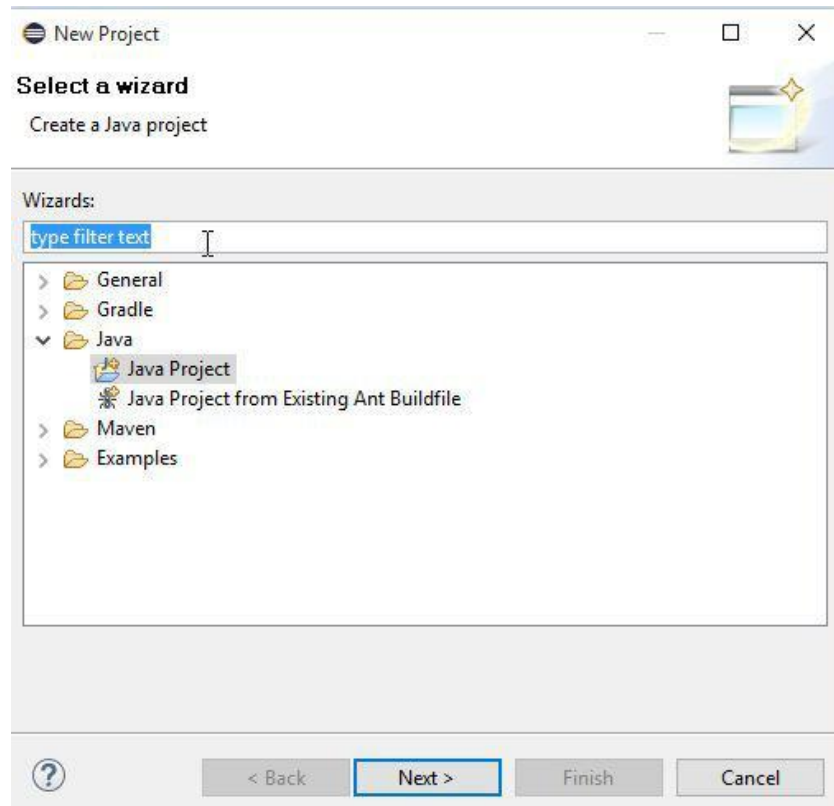
El primer paso será crear un proyecto. Abrimos Eclipse.

1. File > New Project

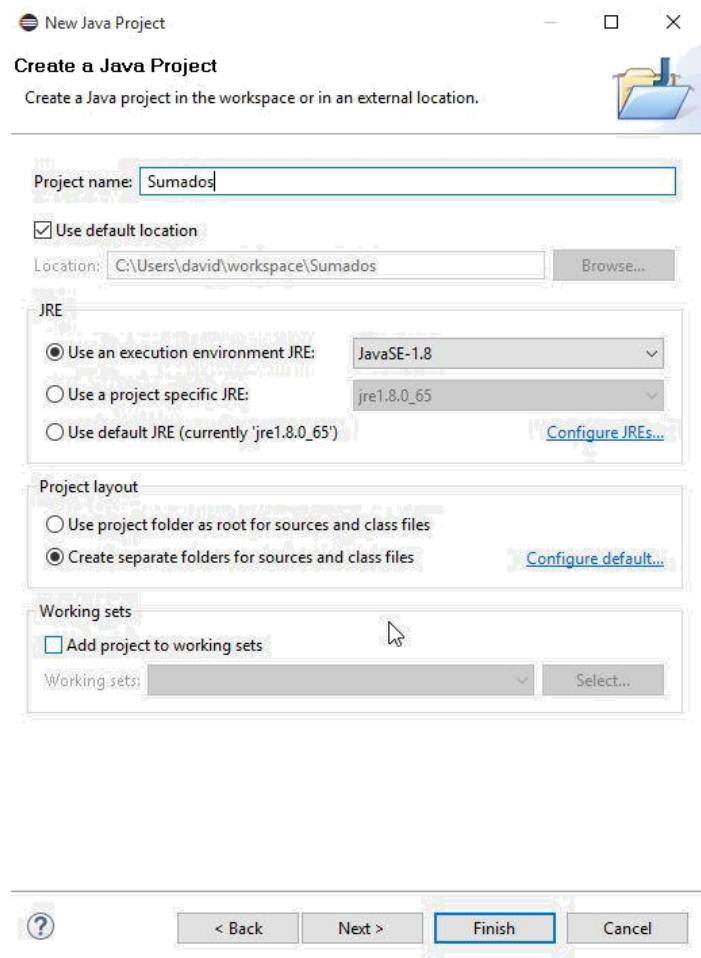


Entornos de desarrollo: Eclipse

2. Seleccionamos Java Project



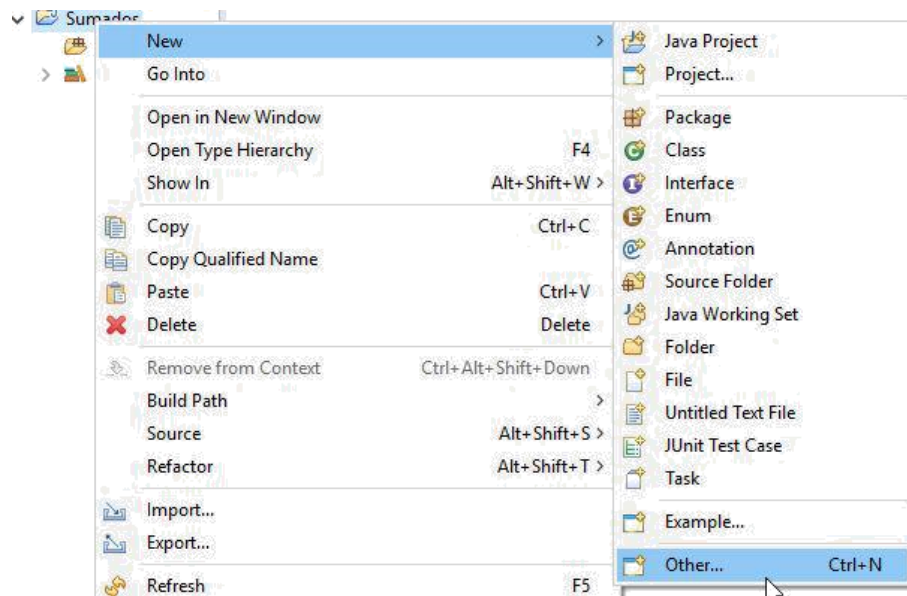
3. Introducimos el nombre del proyecto. Al primer proyecto le voy a llamar Sumados. Dejamos el resto de opciones por defecto.



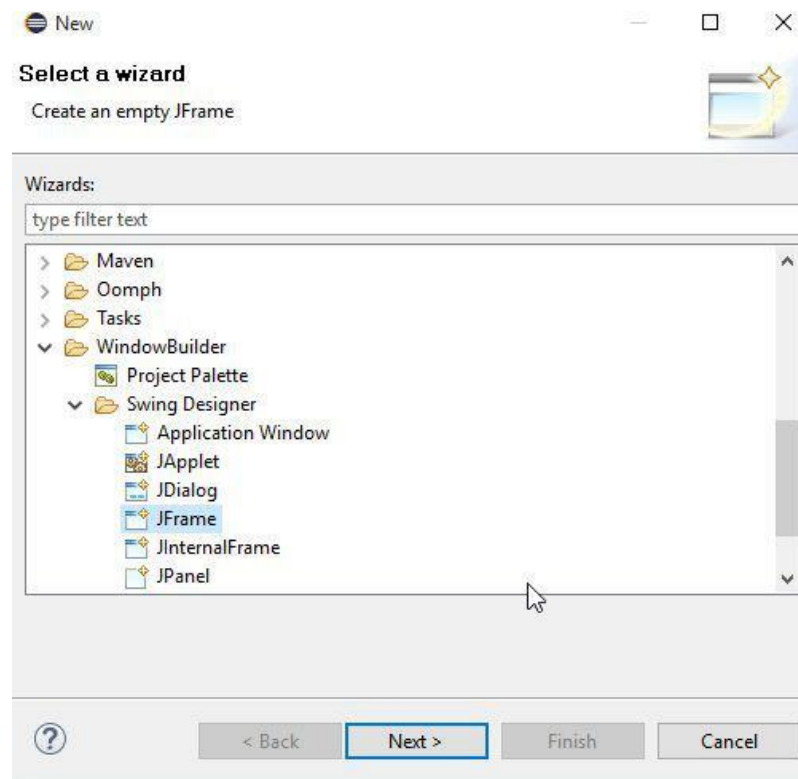
4. Pulsamos Finish. Nos crea estructura proyecto Java



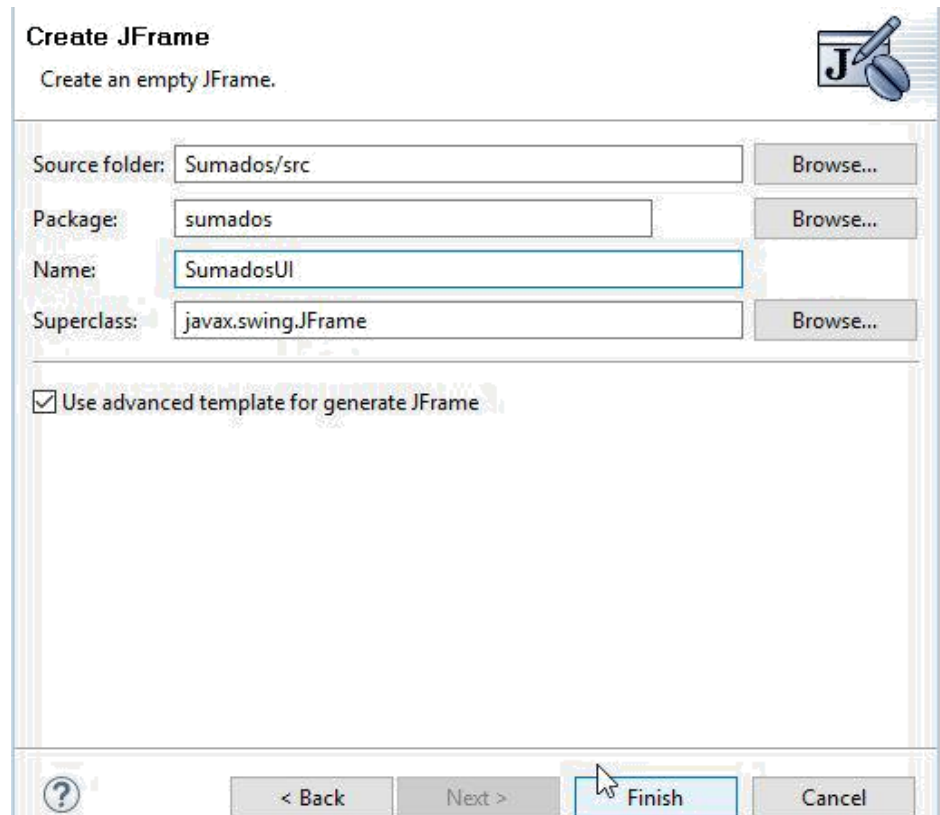
5. Ahora seleccionamos File>New>Other



6. En la ventana New. Abrimos la carpeta WindowsBuilder > Swing Designer > JFrame y pulsamos Next.



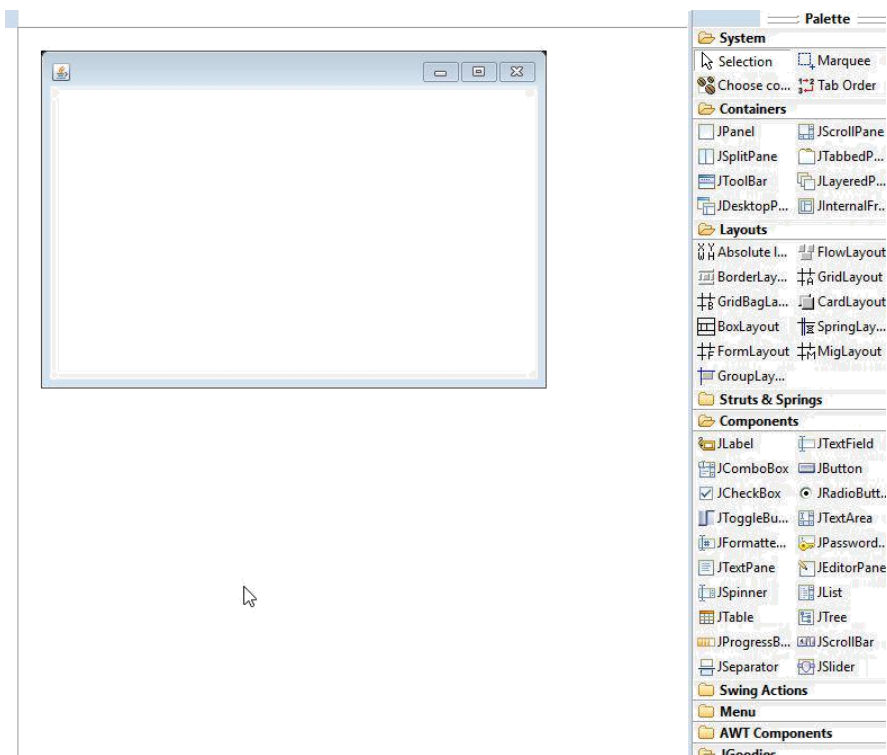
7. Rellenamos los datos de la clase JFrame. Se llamará SumadosUI y guardaremos en el paquete sumados. Pulsamos Finish



8. Nos crea la clase. Si te fijas en la parte inferior de la ventana de editor aparecen dos pestañas Source y Design. Pulsamos en Design

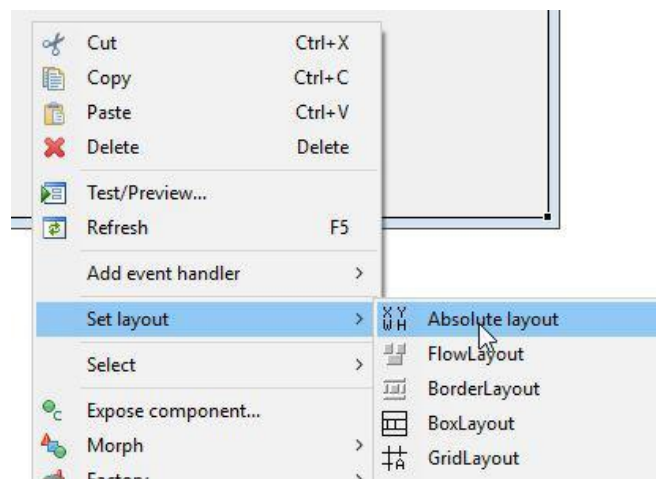


9. Nos abre un panel gráfico donde podremos añadir componentes y una paleta de componentes y controles. (muy similar a NetBeans).

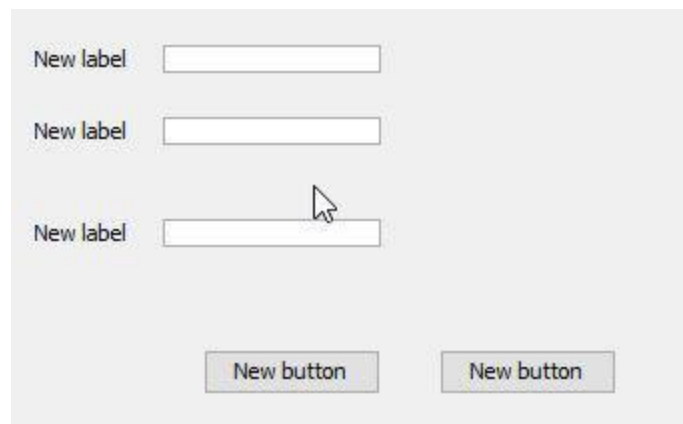


Yo tengo la paleta a la derecha por comodidad, por defecto aparece a la izquierda pero la podeis mover haciendo click sobre ella y situandola.

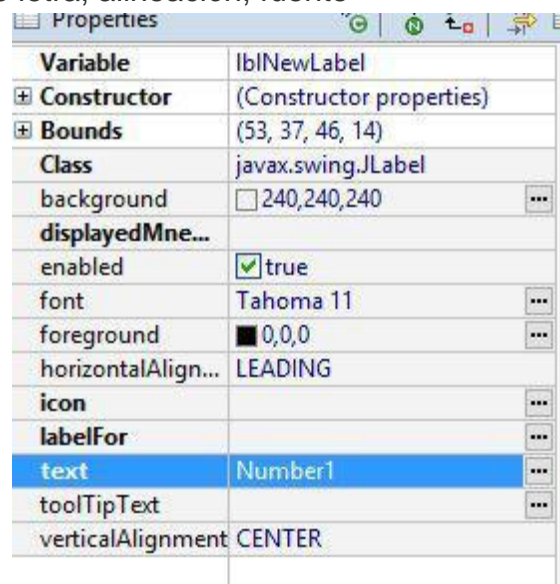
10. El primer paso será configurar el Layout de JFrame presionando el botón derecho del mouse sobre el formulario generado y seleccionamos la opción SetLayout > Absolute layout (esto nos permite luego disponer controles visuales como JButton, JLabel etc. en posiciones fijas dentro del JFrame)



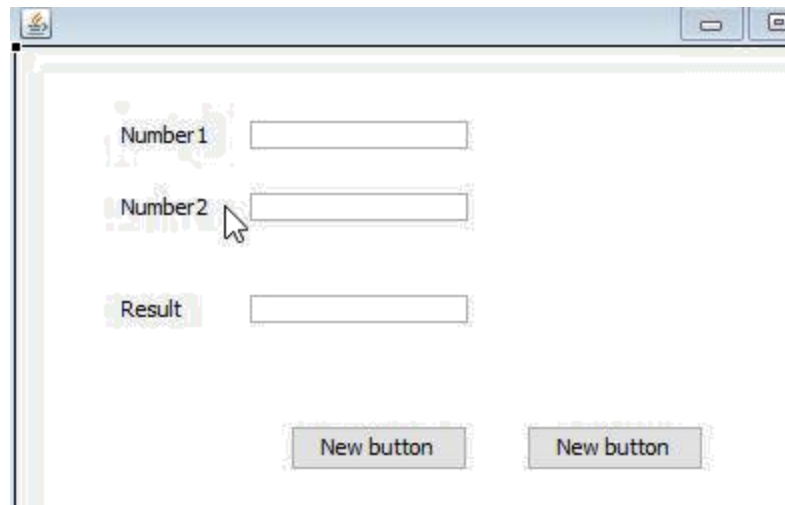
11. De la ventana Palette seleccionamos tres JLabel, tres JTextField y dos JButton y los situamos en el panel.



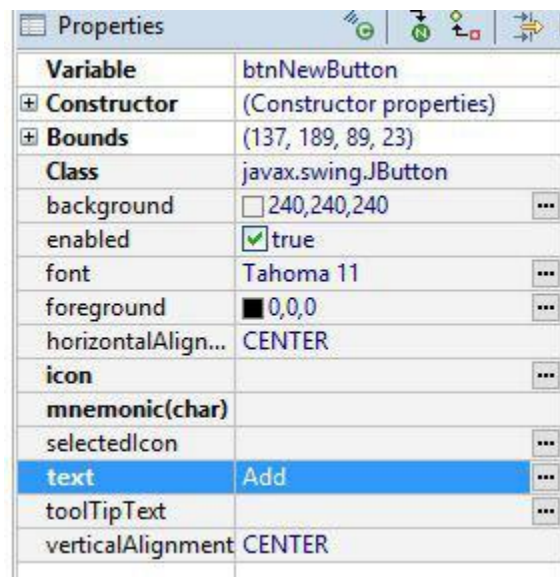
12. Configuramos Label. Selecciona el primer NewLabel y en la ventana propiedades en text introducimos Number1. Observa que podemos cambiar tipo de letra, alineación, fuente



13. Modifica las propiedades de los otros Labels con el texto Number2 y result para obtener lo siguiente

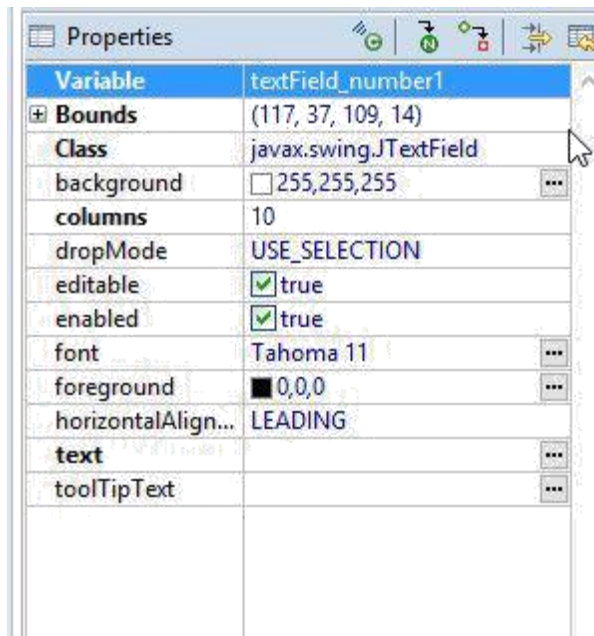


14. Seleccionamos el primer JButton y en la propiedad text introducimos Add.



15. Modificamos el text del otro JButton e introducimos Reset

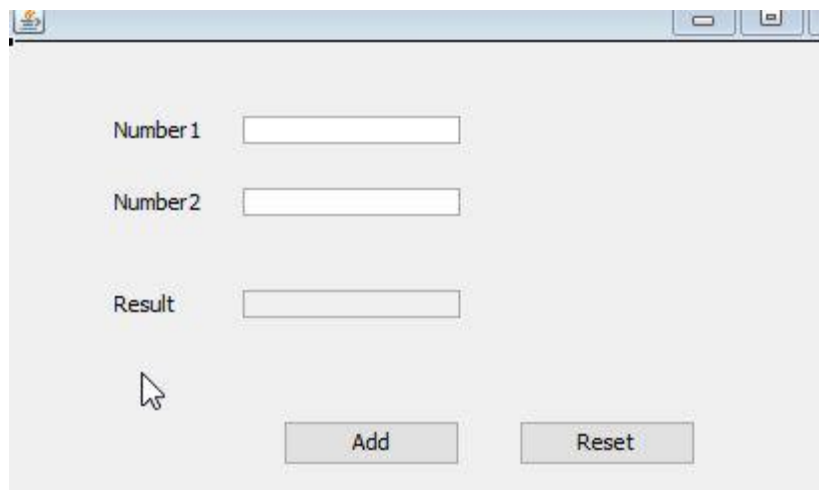
16. Seleccionamos el textField del Number1 y en propiedades cambiamos el valor de variable por textField_number1. Esto facilitará después identificarlo para hacer uso de sus métodos.



17. Al textField de number 2 le asignamos el identificador textField_number2

18. Seleccionamos el textField del result. Le asignamos el identificador textField_result y en propiedades pondremos la propiedad editable a false, para que no se pueda modificar.

19. Tenemos la interfaz ya preparada



20. Vamos a empezar por programa Reset. Hacemos doble click sobre el botón.

21. Nos abre el procedimiento que se ejecutará al pulsar el botón

```
btnReset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
    }
});
```

22. Introducimos el código

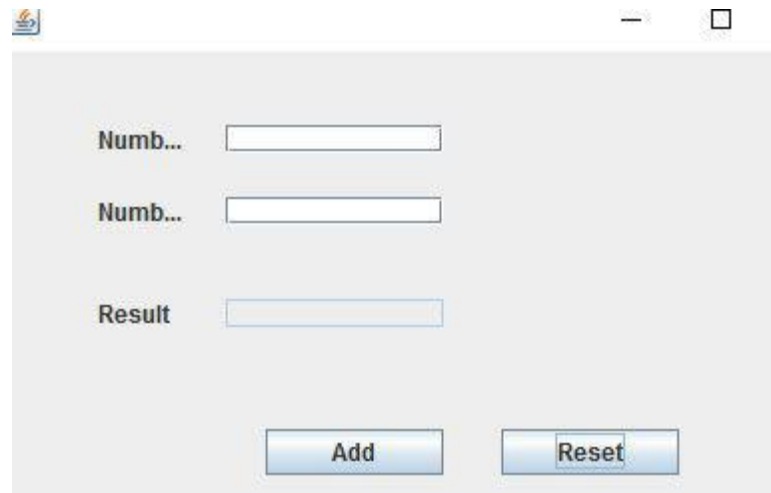
Entornos de desarrollo: Eclipse

```

public void actionPerformed(ActionEvent arg0) {
    textField_number1.setText(""); //introduzco cadena "" en textfield number1
    textField_number2.setText(""); //introduzco cadena "" en textfield number2
    textField_result.setText(""); //introduzco cadena "" en textfield result
}

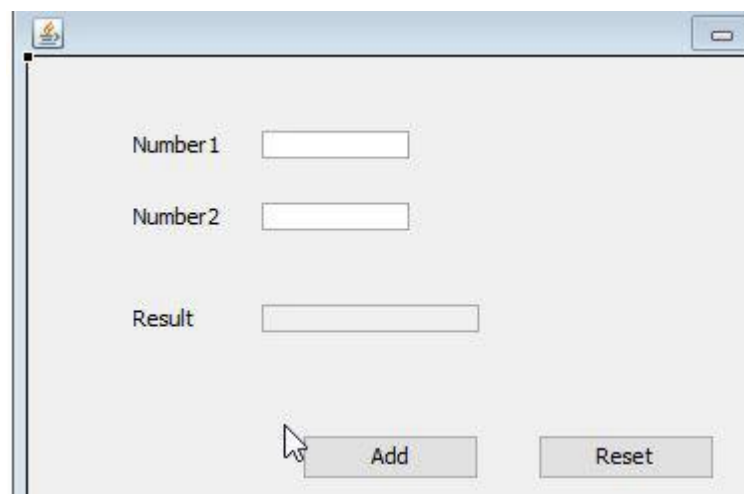
```

23. Guardo y compruebo funcionamiento pulsando Run> Run o bien el botón



24. Funcionalidad correcta. Vemos que no se visualizan bien los labels. Cerramos ejecución. Pulsamos en la pestaña design para modificar el diseño.

25. Modifico diseño interfaz



26. Nos resta programar botón Add, para ello pulso click sobre Add

```

public void actionPerformed(ActionEvent e) {
}

```

27. Añadimos código

```
int num1,num2,result;
```

```
//recojo contenido textField y hago casting a enteros  
num1=Integer.parseInt(textField_number1.getText()); //re  
num2=Integer.parseInt(textField_number2.getText());
```

```
//calculo suma  
result=num1+num2;
```

```
//cargo result en textField_result. Para ello hago ca  
textField_result.setText(Integer.toString(result));
```

28. Comprobamos funcionamiento



Number1 2

Number2 3

Result 5

Add Reset

Practicando

Realiza un programa en Java que solicite un campo usuario y contraseña y "simule" conexión. Si al pulsar conectar el usuario es "yo" y la contraseña "123" mostrará el mensaje "***Bienvenido al sistema señor***" y de ser incorrecto mostrará el mensaje: "***Clave incorrecta señor***"

Método mostrar mensaje en panel

```
JOptionPane.showMessageDialog(null, "Bienvenido al sistema señor");
```

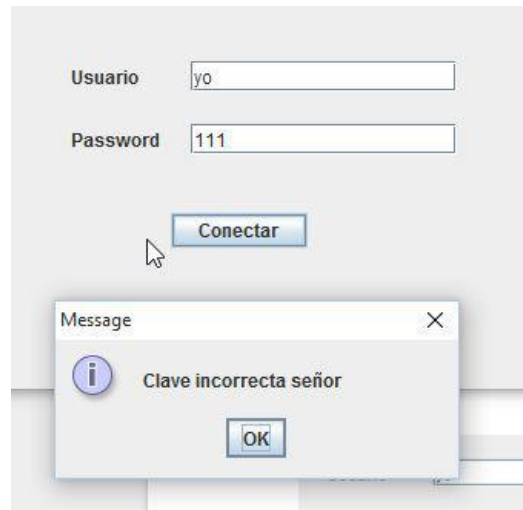
Interfaz programa



Acierto



Error



Tras mostrarle la aplicación a nuestro jefe. Nos solicita que el campo de password aparezca encriptado al introducir los datos. Averigua por tu cuenta como modificar el programa para cumplir con sus requisitos.

