

# Interfaces de Usuario.

Emanuel Izaguirre Cortes

La interfaz de usuario o UI (User Interface) es un concepto que abarca arquitectura de información, patrones y diferentes elementos visuales que nos permiten interactuar de forma eficaz con sistemas operativos y softwares de diversos dispositivos.

Las características de la interfaz de usuario.

- Claridad.

Una buena interfaz transmite la información de manera precisa para evitar que el usuario cometa errores durante la interacción.

- Consistencia

Se trata de dar al usuario la información que necesita y pide.

- Efectividad

Esta característica es la que hace efectiva una interfaz permitiendo a la persona crear patrones de uso de manera sencilla y práctica.

## Flexibilidad.

Una buena interfaz también se caracteriza por permitir que el usuario realice acciones y des haga acciones.

- Atractivo visual.

Cuando se trata de experiencia del usuario, nunca puede subestimarse la importancia del atractivo visual. Además de tener grandes funcionalidades, una buena interfaz es agradable a la vista.



# Interfaces de usuario.

Enunciado Izaguirre cortes.

El diseño de la interfaz del usuario debe estar desarrollando bases y altos estándares de calidad.

El estándar más conocido y seguido es la Norma ISO 9241, que se centra en la ergonomía de la interacción entre la persona y el sistema, específicamente en aspectos como la facilidad de la comunicación y el Diálogo.

Dentro de una página web, aplicación o cualquier Herramienta digital, el diseño de la interfaz del usuario debe garantizar facilidad a los errores, Personalización, control, adecuación, en aprendizaje y, sobre todo, conformidad con las expectativas del usuario.

Además el diseño de una interfaz busca la adecuación al usuario y la autodescripción. Así mismo, existen Principios de coherencia y calidad gráfica que deben seguirse para desarrollarse una interfaz, como emplear códigos de colores para apoyar las tareas de los usuarios, minimalismo y estandarización.

Los elementos que generalmente se presentan en una UI para una Plataforma digital son.

Controles de input: botones campos de texto, checkboxes, radio buttons, listas drops down, switches, campos de Fecha  
Componentes de navegación.

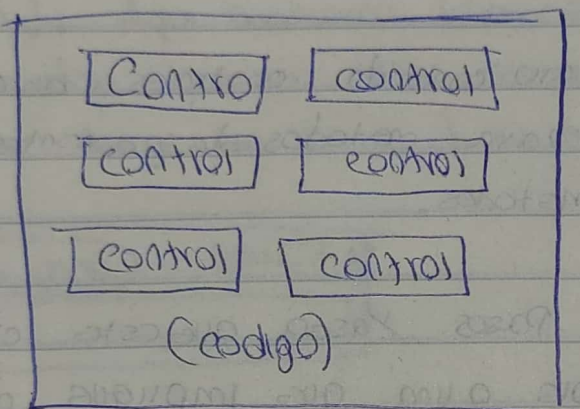


Emanuel Izaguirre Cortes.

## Estructura de una aplicación de GUI de Java.

Veamos la estructura de una aplicación GUI de Java.

En estas notas vamos a utilizar los términos aplicación, Programa, Proyecto como sinónimos. Una aplicación GUI consiste de un marco con controles y códigos asociados. Gráficamente esto es:



La estructura del proyecto se compone de:

▶ Ventana de marco que crea para la interfaz de usuario (También conocido como Formulario).

Controles: Funciones gráficas ubicadas en el marco para permitir la interacción del usuario.

▶ Propiedades: Cada característica de un marco o control está especificada por una propiedad. Las propiedades de ejemplo: incluyen nombres, títulos, tamaños, color, posición y contenido.



En este curso, crearemos aplicaciones en múltiples Formatos.

El código de cada Formulario normalmente aparecerá en su propio archivo con extensión de java.

Seguiremos 3 pasos para crear una aplicación Java GUI:

1. Crear el marco
2. Crear la interfaz de usuario colocando controles en el marco
3. Escribir el código para métodos de eventos de control  
 (Quizás escriba otros métodos.)

Se siguen otros mismos pasos y sea que este creando una aplicación muy simple o una que implique muchos controles y muchas líneas de código.

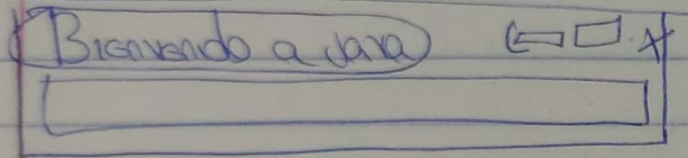
Cada uno de estos pasos requiere que este creando código Java, y a veces, mucho código. La naturaleza basada en eventos de las aplicaciones Java le permite crear su aplicación en etapas y probarla en cada etapa. Esto minimiza los errores y le brinda a usted o Programador, confianza a medida que su aplicación toma forma.

A medida que Avance en sus habilidades de programación recuerde siempre adoptar este enfoque secuencial para crear una aplicación Java. Construya un poco, pruebe un poco, modifique un poco y pruebe nuevamente.

Rápidamente tendrá una solución completa.

## Control de Giro.

Los controles que utilizamos en las aplicaciones de gui serán componentes swing. Estos componentes están definidos con el paquete javax.swing y todos tienen nombres que comienzan con J. Aquí analizamos brevemente varios controles para darles una idea que son como, se ven y que hacen, vea más componentes swings en varios de los proyectos.



## Control de marco J:

El control de marco es el "contenedor" básico para otros controles. Es el marco para un Proyecto Java. La propiedad del título establece la información del título, cada aplicación que construyamos comienza.



# Interfaz Gráfica Gui 15/enero/2024

Raguna Cortes Emanuel

GUI (Graphical User Interface) es un conjunto de componentes gráficos o compuestos GUI (widgets) que permiten la interacción entre el usuario y la aplicación. Es un de componentes añadidos.

## • Contenedores y componentes

La interfaz de usuario es la parte del programa que permite al usuario interactuar con él. La API de Java proporciona una biblioteca de clases para el desarrollo de interfaces gráficas de usuario (en realidad son dos). La biblioteca proporciona un conjunto de herramientas para la construcción de interfaces gráficas que tienen una apariencia y se comportan de forma semejante en todas las plataformas en las que se ejecuten.

La estructura básica de la biblioteca gira entorno a componentes y contenedores. Los contenedores contienen componentes y los componentes a su vez, de forma que los eventos pueden tratarse tanto en contenedores como en componentes. La API está constituida por clases, interfaces y destructores. AWT, swing, FX.

Las interfaces gráficas de usuario (GUI) ofrecen al usuario ventanas, cuadros de diálogo, barras de herramientas, botones, listas, desplegables y muchos otros elementos con los que ya estamos muy acostumbrados a tratar.



# Interfaz Gráfica Gui.

Emmanuel Izaguirre Corcos.

## Abstracts windowing Toolkit (AWT)

"Look & Feel" dependiente de la Plataforma la apariencia de Ventanas, menús, etc. es distinta en windows, mac, motif, y otros sistemas. Funcionalidad independiente de la Plataforma Basico y experimental. Estándar hasta la versión JDK 1.1.5

## Swing / Java Foundation Class C desde JDK 1.1.5

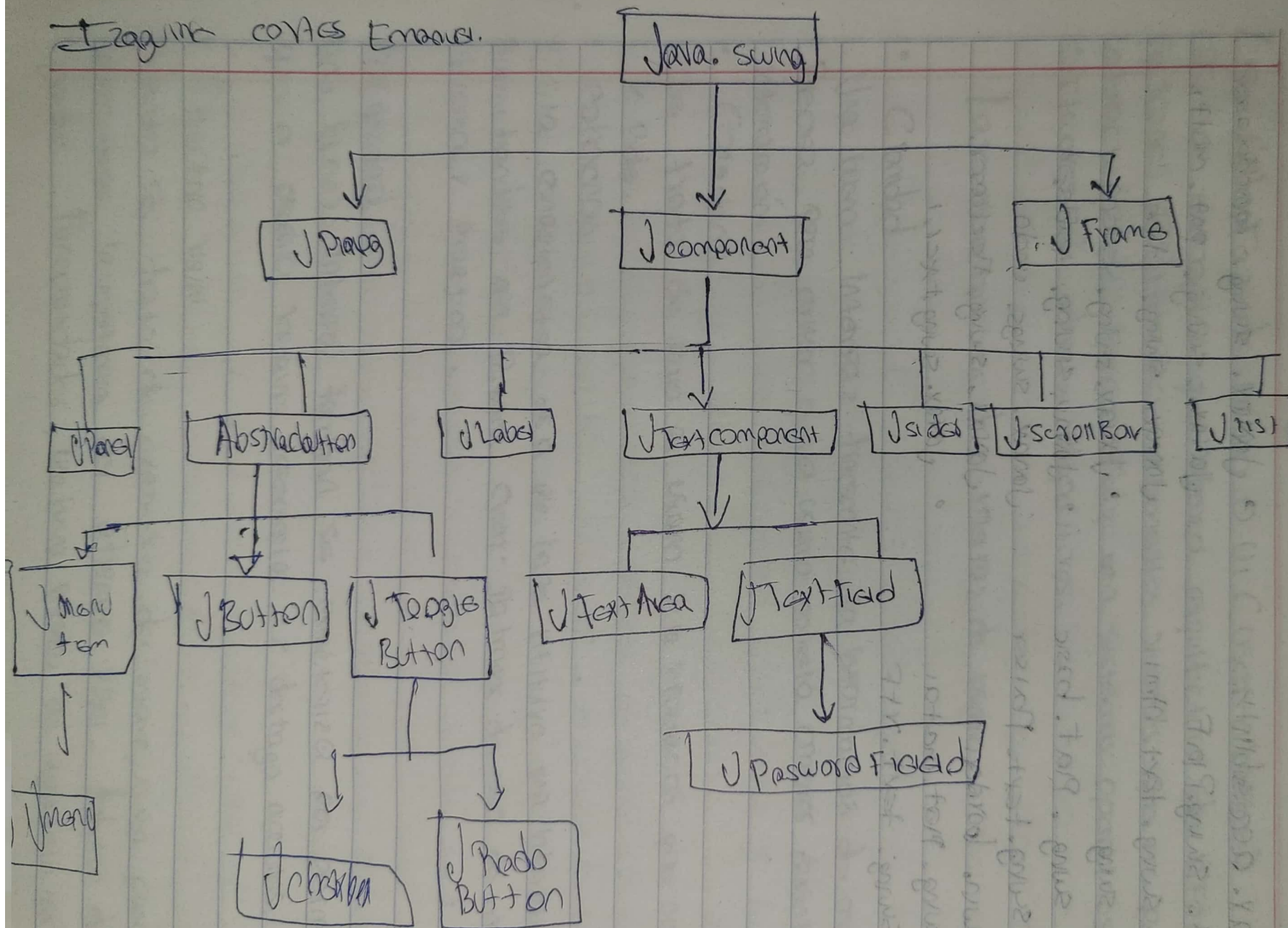
Look & Feel. Funcionalidad independiente de la Plataforma Desarrollado 100% en Java

Portable: Si se ejecuta en "Look & Feel" soportado por swing o se programa uno puede asegurarse que la gui se vera igual en cualquier Plataforma mucho mas completo que AWT.

Java FX es una familia de productos y tecnologías de Oracle Corporation (inicialmente son Microsystems), para la creación de Rich Internet Applications (RIAs). Esto es aplicaciones web que tienen características y capacidades de aplicaciones de escritorio, incluyendo aplicaciones multimedia interactiva. Las tecnologías incluidas bajo la denominación Java FX son Java FX Script y Java FX Mobile aunque hay mas productos Java FX planeados.



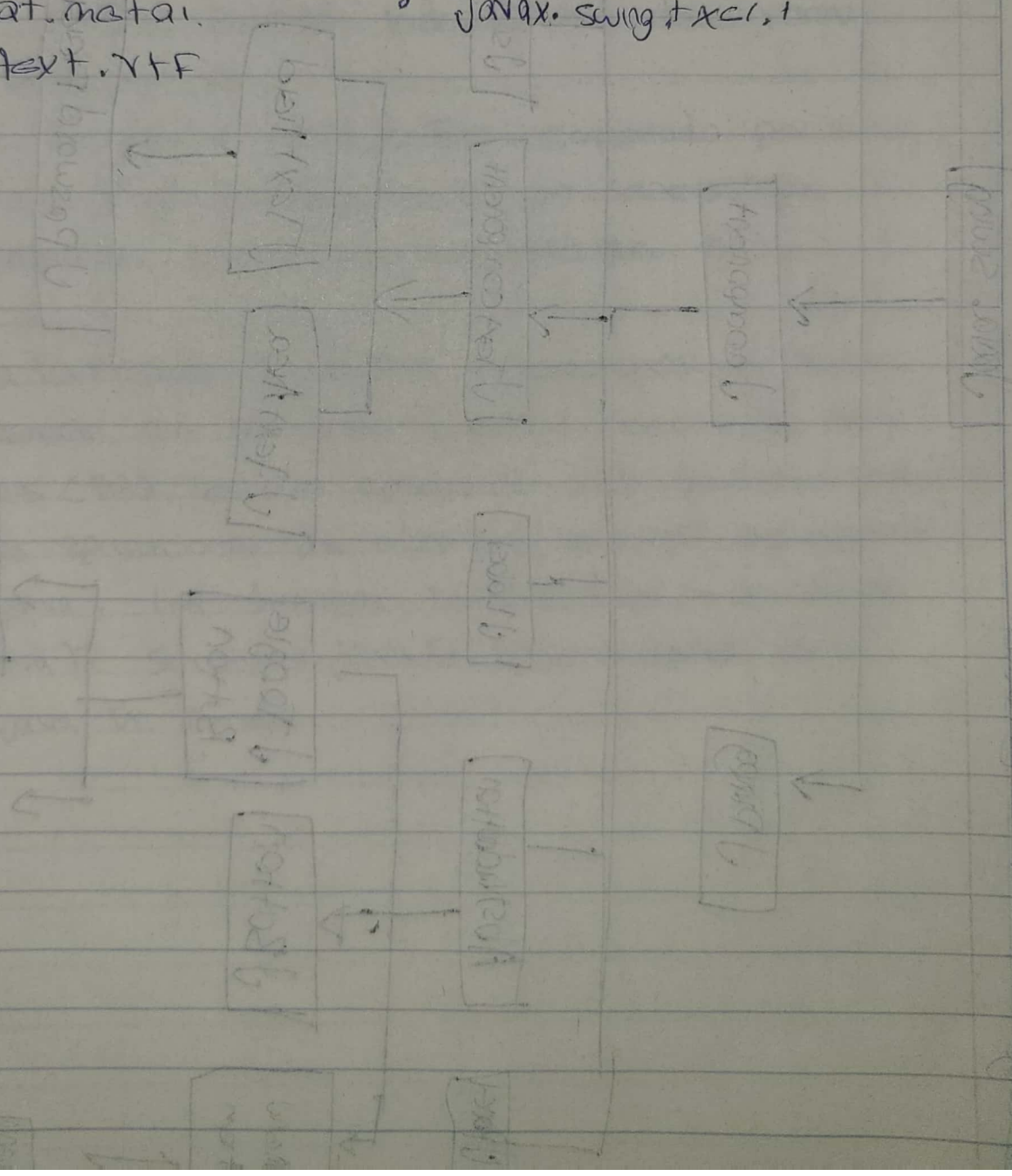
Diagrama de clases Empleado.





# Interfaces Graficos.

- javax.accessibility
- javax.swing.plaf
- javax.swing.text.html
- java.swing
- javax.swing.plaf.basic
- javax.swing.text.Parser
- javax.swing.border
- java.swing.plaf.metal
- javax.swing.text.YTF
- javax.swing.event
- javax.swing.table
- javax.swing.undo
- javax.swing.FFileChooser
- javax.swing.text

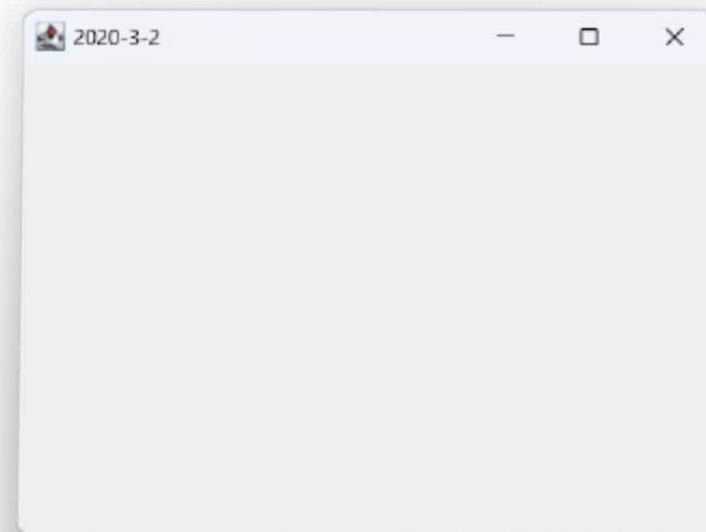




```

1 package azzzz3330168_PA_ejemplos;
2
3
4
5 import javax.swing.JFrame;
6
7
8 public class ejemplojframe01 extends JFrame {
9
10
11     public static void main(String[] args) {
12         JFrame ventana = new JFrame ();
13         ventana.setTitle("2020-3-2");
14         ventana.setBounds(10,20,400,300);
15         ventana.setVisible(true);
16
17     }
18 }

```

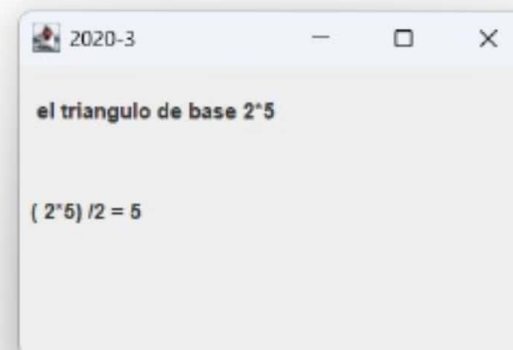




```

3
4
5 import javax.swing.JFrame;
6 import javax.swing.JLabel;
7
8 public class ejemploframe02_b extends JFrame {
9     static JLabel a , b ;
10
11     public ejemploframe02_b() {
12         this.setTitle("2020-3");
13         this.setBounds(10,20,300,200);
14         a = new JLabel ("el triangulo de base 2*5 ");
15         a.setBounds(10,10,200,30);
16         b = new JLabel();
17         b.setBounds(10,50,300,30);
18         add(a);
19         add(b);
20     }
21
22
23     public static void main(String[] args) {
24         ejemploframe02_b ventana = new ejemploframe02_b();
25         b.setText(" ( 2*5) /2 = 5 ");

```





type filter text

a2223330168\_PA\_Unidad1

JRE System Library [jre]

src

a222333\_PA\_ejercicios

a2223330168\_PA\_ejemplos

ejemploframe01\_a.java

ejemploframe01\_b.java

ejemploframe01\_c.java

ejemploframe02\_a.java

a2223330168\_PA\_Tareas

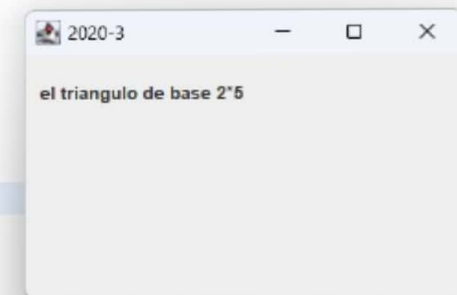
```
6 import javax.swing.JLabel;
7
8
9 public class ejemploframe02_a extends JFrame {
10     JLabel a , b ;
11     public ejemploframe02_a() {
12         this.setTitle("2020-3");
13         this.setBounds(10,20,300,200);
14         a = new JLabel ("el triangulo de base 2*5 ");
15         a.setBounds(10,10,200,30);
16         b = new JLabel();
17         b.setBounds(10,50,300,30);
18         add(a);
19         add(b);
20     }
21
22
23     public static void main(String[] args) {
24         ejemploframe02_a ventana = new ejemploframe02_a();
25         ventana.setVisible(true);
26     }
27
28
29 }
30
```

Outline

a2223330168\_PA\_ejemplos

ejemploframe02\_a

- a : JLabel
- b : JLabel
- ejemploframe02\_a()
- main(String[]): void





```

import javax.swing.JLabel;

public class ejemploframe02_a extends JFrame {
    JLabel a ,b ;
    public ejemploframe02_a() {
        this.setTitle("2020-3");
        this.setBounds(10,20,400,300);
        a = new JLabel ("el triangulo de base 2*5 ");
        a.setBounds(10,10,200,30);
        b = new JLabel();
        b.setBounds(10,50,300,30);
        add(a);
        add(b);
    }

    public static void main(String[] args) {
        ejemploframe02_a ventana = new ejemploframe02_a();
    }
}

```

