

La Cantuta

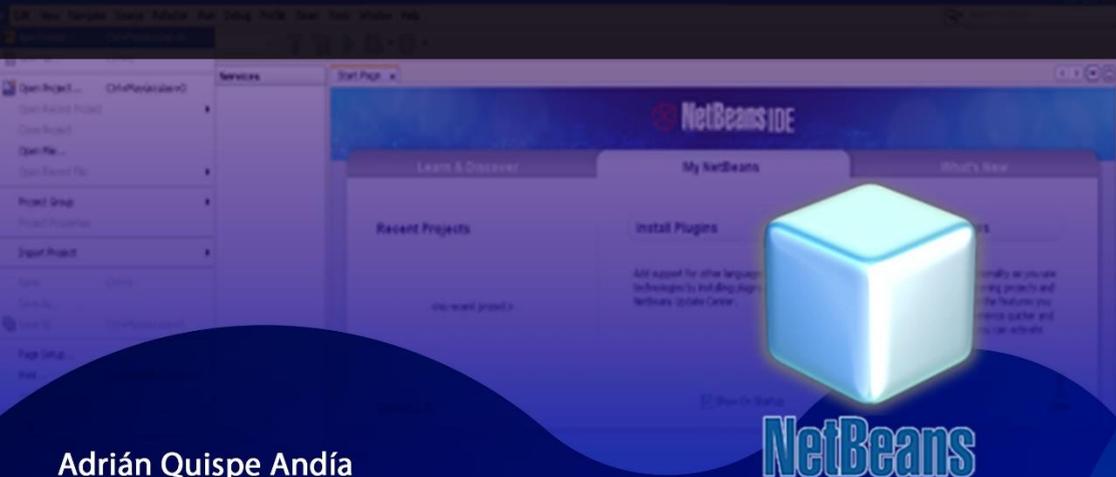
Fondo Editorial

Universidad Nacional de Educación Enrique Guzmán y Valle



```
12 //
13 // @param args the command line arguments
14 //
15 public static void main(String[] args) {
16     int dato1 ,dato2,potencia;
17     dato1=2;
18     dato2=3;
19     potencia=(int) Math.pow(dato1, dato2);
20     System.out.println(potencia);
21
22 }
23
24 }
```

Lenguaje de programación Java Netbeans y aplicaciones visuales educativas



Adrián Quispe Andía
Guillermo Pastor Morales Romero
Teresa Guía Altamirano
Shirley María Teresa Quispe Guía

fondoeditorial.une.edu.pe



Universidad Nacional de Educación Enrique Guzmán y Valle (UNE)

Lenguaje de programación Java Netbeans y aplicaciones visuales educativas



Adrián Quispe Andía

Guillermo Pastor Morales Romero

Teresa Guía Altamirano

Shirley Maria Teresa Quispe Guía

Lima - Perú

2023

Lenguaje de programación Java Netbeans y aplicaciones visuales educativas

© **Adrián Quispe Andía**

aquispe@une.edu.pe

Guillermo Pastor Morales Romero

gmorales@une.edu.pe

Teresa Guía Altamirano

tguia@une.edu.pe

Shirley Maria Teresa Quispe Guía

shirley.quispe@upn.edu.pe

Editada por:

© Universidad Nacional de Educación Enrique Guzmán y Valle (UNE) - **Fondo Editorial “La Cantuta”**

Dirección: Enrique Guzman y Valle N° 951, Lurigancho-Chosica 15472, Perú

ISNI: 0000 0000 8534 4267

fondoeditorial@une.edu.pe

Teléf. móvil: +51 999 140 920

Portal Web: <https://www.une.edu.pe/>

Primera edición digital: Octubre 2023

Libro digital disponible en: <https://fondoeditorial.une.edu.pe/>

Hecho el Depósito Legal en la Biblioteca Nacional del Perú N° 2023-10227

ISBN: 978-612-4148-49-1

DOI: <https://doi.org/10.54942/lacantuta.29>

Corrección de estilo: Luis Pablo Diaz Tito

luisp.diaz@upsjb.edu.pe / Tel. de contacto: +51 955 129 801

Diseño y Diagramación: Gráfica “imagen”

Manuel Enrique Sampen Antonio

sampen25@gmail.com / Tel. de contacto: +51 990 064 589

Revisión por pares ciegos aprobado por el **Consejo Editorial del Fondo Editorial “La Cantuta”**.

Libro resultado de Investigación y con revisión por pares doble ciego.

Sello editorial: Fondo Editorial (978-612-4148)



No está permitida la reproducción total o parcial de este libro, su tratamiento información, la transmisión de ninguna otra forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares del copyright.

AGRADECIMIENTOS

Mi especial agradecimiento a la Universidad Enrique Guzmán y Valle, y a las autoridades, por su apoyo en la publicación de este libro producto de investigación.

Los autores

Tabla de contenido

AGRADECIMIENTOS	5
INTRODUCCIÓN	8
CAPITULO I.....	9
Conceptos básicos de Java	9
1.1 Concepto de Java	12
1.2 Variables	12
1.3 Operadores	13
1.4 Lectura de datos:	15
1.5 Estructuras de control.....	15
1.5.1 Estructura condicional simple: Condicional if.....	16
1.5.2 Estructura condicional doble: if..else.....	16
1.5.3 Estructura de selección múltiple: switch.....	17
1.6 Estructuras repetitivas	17
1.6.1 Estructura de repetición: while	18
1.6.2 Estructura de repetición: do while	19
1.6.3 Estructura de repetición: for.....	20
1.7 Arreglos.....	21
1.7.1 Arreglos unidimensionales.....	21
1.7.2 Arreglos bidimensionales: (matriz).....	22
1.7.3 Arreglo tridimensional:	23
CAPITULO II	24
Aplicaciones visuales educativas	24
2.1 Programas de Java en consola y panel:	25
2.2 Aplicaciones visuales educativas:	34
2.2.1 Aplicaciones visuales secuenciales	34

2.2.2 Aplicaciones visuales con estructura de control if ..else ..	40
2.2.3 Aplicaciones visuales con estructura de control switch case ..	51
2.2.4 Aplicaciones visuales con estructura de control repetitiva while ..	56
2.2.5 Aplicaciones visuales con estructura de control repetitiva do while ..	63
2.2.6 Aplicaciones visuales la estructura de control repetitiva for ..	66
2.2.7 Aplicaciones con menús: ..	70
2.2.8 Aplicaciones con arreglos ..	75
2.2.8 Gráficos en Java: ..	82
2.2.9 Ejercicios Propuestos: ..	83
REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS ..	85
LINKOGRAFIA: ..	86

INTRODUCCIÓN

El presente libro titulado: “Lenguaje de programación Java Netbeans y aplicaciones visuales educativas” producto de una investigación, plantea de manera simple los conceptos y la programación Java con aplicaciones educativas con una gran variedad de ejercicios, desde los simples y complejos.

Los destinatarios del libro son, en general, los estudiantes de pregrado de la Facultad de Ciencias con especialidad de informática, matemática en informática y matemática y otros interesados en la programación Java.

La estructura del libro está compuesta por capítulos en forma didáctica, sencilla y práctica. El Capítulo I titulado: Conceptos generales se refiere a la conceptualización del lenguaje de programación Java, variables, operadores, estructura de control, arreglos el Capítulo II titulado: “Aplicaciones visuales educativas”, se refiere a diversas aplicaciones por cada estructura de control, menús y arreglos además de programación de gráficos.

Los autores



CAPITULO I

Conceptos básicos de Java

Según Abarca R. (2006). "El aprendizaje constructivista es como un viaje fascinante hacia el conocimiento, donde el estudiante se convierte en el arquitecto de su propia comprensión". Que es uno de los pilares del lenguaje de programación.

"La programación es el lenguaje que nos permite dar vida a nuestras ideas. Alonso, J. (1991) nos muestra en su libro 'Motivación y aprendizaje en el aula. Cómo enseñar a pensar' cómo enseñar a pensar de manera lógica y creativa a través de la programación."

"La programación es el lenguaje que nos permite dar vida a nuestras ideas y transformar el mundo digital. Coloma, O (2005) nos guía en su libro 'Informática y Software Educativo', hacia el fascinante mundo del lenguaje de programación y su aplicación en el ámbito educativo en el Perú."

"La programación es el arte de dar vida a las ideas a través del código. En su libro 'Fundamentos de programación', Joyanes, L (2008) nos sumerge en el fascinante mundo de la programación y nos muestra cómo construir algoritmos eficientes y estructuras de datos sólidos, este libro es una guía invaluable para aquellos que desean dominar el lenguaje de programación de manera práctica y sencilla."

"La colaboración entre Oracle y Sun Microsystems en 2010 marcó un hito en la industria de la tecnología. En su página web oficial, Oracle (2010) nos brinda acceso a información valiosa sobre el lenguaje de programación y su relación con Sun Microsystems "

"En su obra 'Desafío a la Mente: Computadoras y Educación' publicada en 1987, Papert, S. nos invita a explorar el lenguaje de programación como una herramienta poderosa para la educación. Su libro, nos sumerge en el fascinante mundo de las computadoras y su potencial transformador en el ámbito educativo.

"Lenguaje de Programación: una herramienta fundamental en el mundo de la informática, que permite a los nuevos programadores dar vida a sus ideas y solucionar problemas de manera eficiente. En el artículo 'Definición de Lenguaje de programación' de Perez, J. y Merino, M. (2009), se explora en detalle la naturaleza y el propósito de los lenguajes de programación."

"Lenguajes de programación: ¿qué son y para qué sirven?" es un artículo escrito por Morales, R. en 2014. En este recurso, el autor explora de manera fascinante la naturaleza y la utilidad de los lenguajes de programación en el mundo actual.

Esta obra inédita ofrece una perspectiva informativa y esencial para comprender el papel fundamental que desempeñan los lenguajes de programación en la tecnología y la sociedad.

1.1 Concepto de Java

Orientado a objetos. El IDE es de libre distribución por lo que se considera un software libre. Tiene cierta similitud con el lenguaje de programación C++, C# y PHP. Se pueden hacer aplicaciones en modo consola y visuales bajo formularios.

1.2 Variables

Una variable es un espacio de memoria que tiene un nombre y un tipo de dato asociados, además de un valor que puede cambiarse durante el tiempo de ejecución.

El formato para la declaración de una variable o grupo de variables es el siguiente:

<tipo de dato> <lista de variables>;

Opcionalmente puede inicializarse una variable al momento de su declaración.

Ejemplos:

```
int x = 23, y, z ;
```

```
float a, b = 2.90, c;
```

```
char prim = 'A' ,seg ;
```

1.3 Operadores

Los objetos que se manejan en Java tienen un tipo de dato asociado, el cual determina la cantidad de espacio de almacenamiento, así como el conjunto de operaciones que podrán realizarse con los valores almacenados. Las operaciones serán representadas a través de identificadores específicos llamados operadores.

a) Operadores aritméticos

Se aplican sobre objetos con valores numéricos.

Sean: $x = 20$, $y = 30$

Operador	Operación	Ejemplo	Resultado
+	Adición	$z = x+y$	$z = 50$
-	Sustracción	$z = y-x$	$z = 10$
*	Multiplicación	$z = x*y$	$z = 600$
/	División	$z = y/x$	$z = 1.5$
%	Módulo	$z = y\%x$	$z = 10$
++	Incremento	$x++$	$x = 21$
--	Decremento	$x--$	$x = 19$

b) Operadores relacionales

Los operadores relacionales se usan para realizar comparaciones dando como resultado dos posibles valores de verdad: FALSO y VERDADERO.

Operador	Significado	Ejemplo
==	Igual que	$x == 7$
!=	Diferente de	$a != 10$
>	Mayor que	$p > q$
>=	Mayor igual que	$k >= m$
<	Menor que	$7 < y$
<=	Menor igual que	$c <= d$

c) Operadores lógicos

Los operadores lógicos se aplican sobre los enunciados que resultan de las operaciones relacionales, y el resultado siempre será un valor de verdad. Los operadores lógicos son:

Operador	Significado	Ejemplo
!	NO (Negación)	$!(a > b)$
&&	Y (Conjunción)	$(x > 3) \&\& (x < 10)$
	O (Disyunción)	$(m <= 2) \ \ (m >= 12)$

d) Operadores de asignación

Los operadores de asignación sirven para asignar un valor a una variable.

Operador	Significado	Ejemplo	Equivalente
=	Asignación	$x = y$	
+=	Suma asigna	$x += y$	$x = x + y$
-=	Resta asigna	$x -= y$	$x = x - y$

*=	Multiplicación asigna	x *= y	x = x * y
/=	Divide asigna	x /= y	x = x / y
%=	Residuo asigna	x %= y	x = x % y

1.4 Lectura de datos:

Formato: números enteros:

```
int n1;
```

```
n1=Integer.parseInt(jTextField1.getText())
```

Formato: números enteros y decimales:

```
float n1;
```

```
n1=Float.parseFloat(jTextField1.getText())
```

Formato: números enteros y decimales(double)

```
double n1;
```

```
n1=Double.parseDouble(jTextField1.getText())
```

Formato: números enteros: control combobox

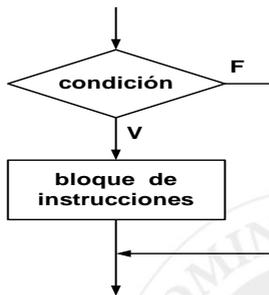
```
Int marca;
```

```
marca=jComboBox1.getSelectedIndex();
```

1.5 Estructuras de control

Son estructuras de programación que se utilizan para controlar los programas bajo un algoritmo estructurado.

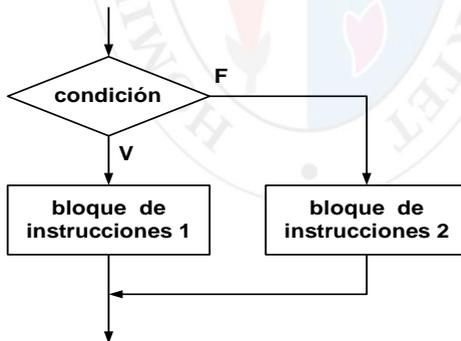
1.5.1 Estructura condicional simple: Condicional if



```
if ( < condición > )  
    < bloque de instrucciones >
```

Si el bloque consta de 2 o más instrucciones debe encerrarse entre llaves { }.

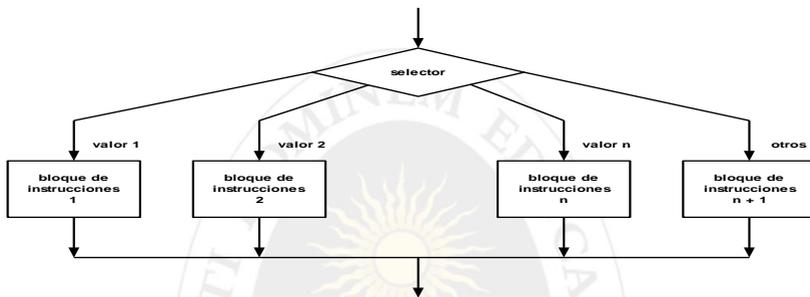
1.5.2 Estructura condicional doble: if..else



```
if ( < condición > )  
    < bloque de instrucciones 1 >  
else  
    < bloque de instrucciones 2 >
```

Si los bloques constan de 2 o más instrucciones deben encerrarse entre llaves { }.

1.5.3 Estructura de selección múltiple: switch



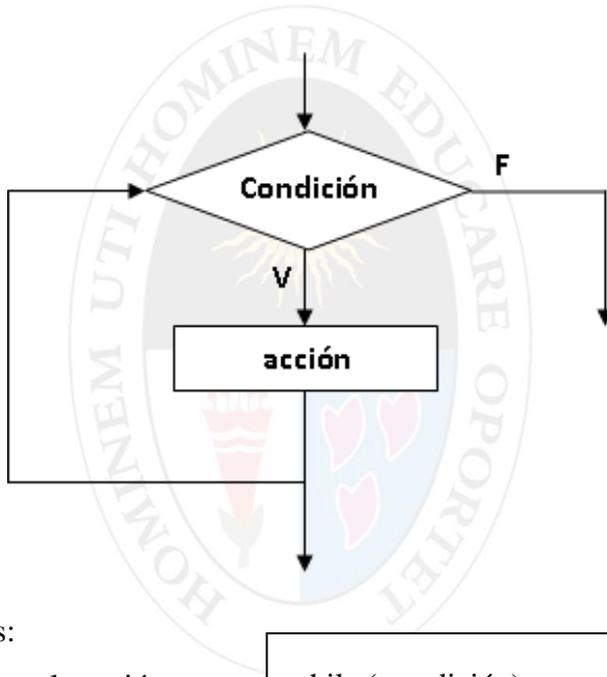
```
switch( < selector > )
{
    case < valor 1 > : < bloque de instrucciones 1 >
                    break;
    case < valor 2 > : < bloque de instrucciones 1 >
                    break;
    case < valor n > : < bloque de instrucciones n >
                    break;
    default : < bloque de instrucciones n+1 >
}
}
```

1.6 Estructuras repetitivas

También conocidas como estructuras iterativas o bucles, permiten ejecutar un conjunto de instrucciones (cuerpo del bucle) mientras se cumpla una determinada condición; cada repetición de secuencias de instrucciones se conoce como iteración.

1.6.1 Estructura de repetición: while

Esta estructura primero verifica la condición y luego ejecuta la acción. La acción puede ser una acción simple o una acción compuesta (bloque de instrucciones encerradas entre llaves).



Sintaxis:

Para una sola acción:

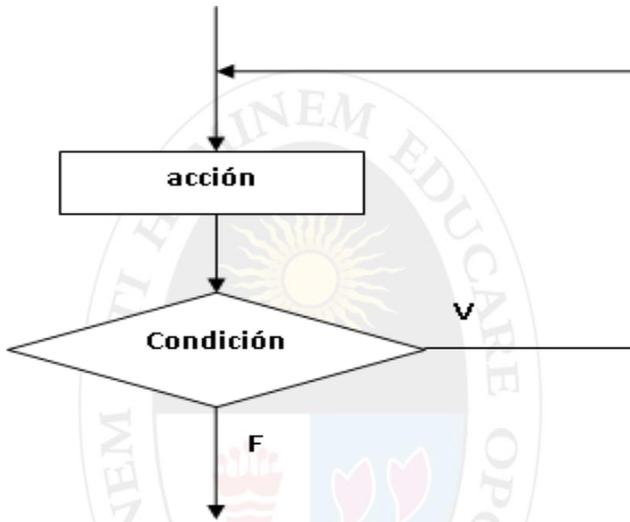
```
while ( condición)
```

Para varias acciones:

```
while (condición){  
    Acción 1;  
    Acción2;  
    .  
    Acción n:  
}
```

1.6.2 Estructura de repetición: do while

Este tipo de estructura primero ejecuta la acción y luego verifica la condición. La acción puede ser simple o compuesta.



Sintaxis:

Para una sola acción:

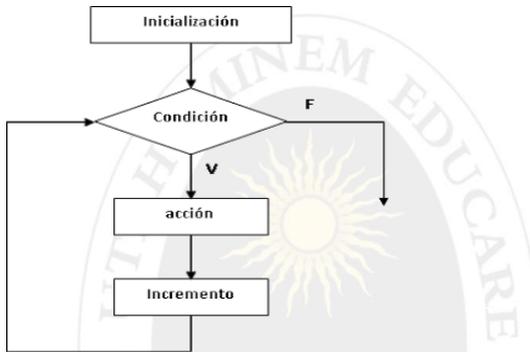
```
do
Acción;
while ( condición)
```

Para varias acciones:

```
do {
    Acción 1;
    Acción2;
    .
    Acción n;
} while (condición);
```

1.6.3 Estructura de repetición: for

Este tipo de estructuras incluye un contador como parte de su estructura, lo cual, quiere decir que se conoce el número de veces que se repetirá el cuerpo del bucle.



Sintaxis:

```
for (inicio;condición;incremento)  
Acción:
```

Para una sola acción:

```
for (inicio;condición;incremento) {  
Acción 1;  
Acción2;  
Acción n;  
}
```

Para varias acciones:

Nota: El contador del for puede estar declarado dentro de él mismo, en este caso el contador se considera como variable local

al for, siendo inaccesible fuera de él. Por lo tanto, dos o más for no anidados, pueden declarar contadores con el mismo nombre.

1.7 Arreglos

Son estructura de datos que utilizan la memoria para almacenar datos y operarlos en entrada y salida, son unidimensionales (vector), bidimensionales (matriz) o multidimensionales.

1.7.1 Arreglos unidimensionales

Un arreglo unidimensional es un conjunto de elementos que tienen las siguientes características: Se almacenan en posiciones consecutivas de memoria.

Todos los elementos son del mismo tipo.

Todos los elementos tienen el mismo nombre y se diferencian entre sí por un número que indica su posición.

Ejemplos:

edad

23	45	34	19	62
0	1	2	3	4

letra

'k'	'p'	'c'	'x'	'a'	'z'	'm'
0	1	2	3	4	5	6

talla

1.77	1.70	1.65	1.80	1.91	1.78	1.67	1.82
0	1	2	3	4	5	6	7

Declaración de arreglos unidimensionales

En lenguaje Java un arreglo unidimensional se declara según la siguiente sintaxis:

Tipo de dato Nombre del arreglo [# de elementos];
--

Ejemplos:

```
int edad[5];
```

```
char letra[7];
```

```
float talla[8];
```

1.7.2 Arreglos bidimensionales: (matriz)

Declaración de Arreglos Bidimensionales

En lenguaje Java un arreglo bidimensional se declara según la siguiente sintaxis:

Tipo de dato Nombre del arreglo [# de elementos] [# de elementos]; Filas columnas
--

Ejemplos:

```
int edad[5] [5];
```

```
char letra[7] [7];
```

```
float talla[8] [8];
```

1.7.3 Arreglo tridimensional:

Declaración de arreglos tridimensionales

En lenguaje Java un arreglo tridimensional se declara según la siguiente sintaxis:

Tipo de dato	Nombre del arreglo	[# de elementos]	[# de elementos]	[# de elementos];
		<i>l</i>	<i>j</i>	<i>k</i>

Ejemplos:

```
int edad[5] [5] [5];
```

```
char letra[7] [7] [7];
```

```
float talla[8] [8] [8];
```

The logo of the Universidad de Cuenca is a circular emblem. It features a sun with rays at the top, a shield in the center divided into four quadrants (top-left: red and white, top-right: blue and white, bottom-left: red and white, bottom-right: blue and white), and a book at the bottom. The Latin motto "HOMINUM UBI HOMINEM EDUCARE DEBENT" is inscribed around the perimeter of the circle.

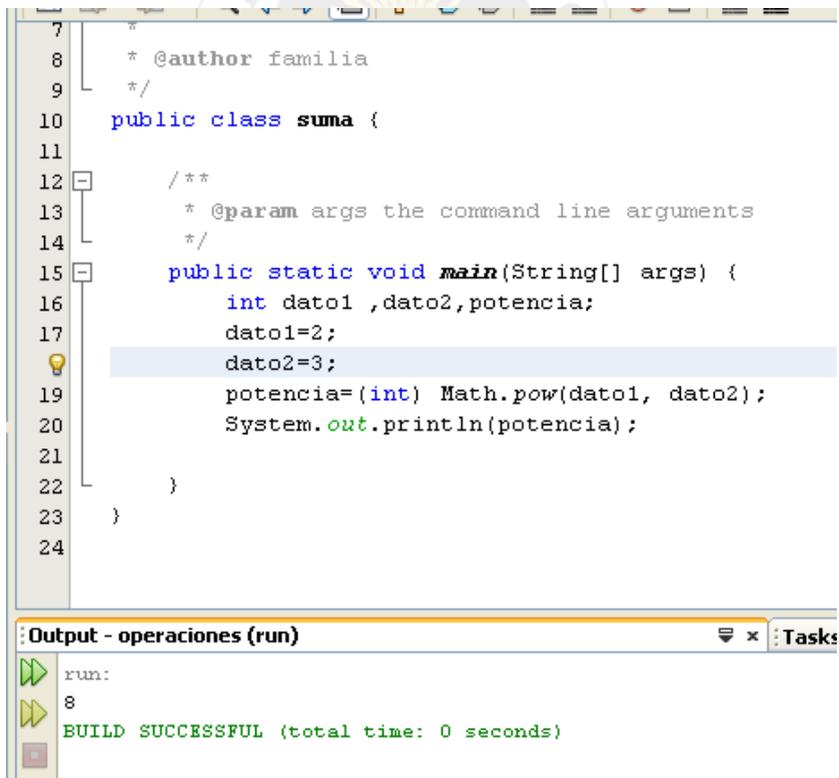
CAPITULO II

Aplicaciones visuales educativas

2.1 Programas de Java en consola y panel:

El lenguaje de programación Java permite bajo Net Beans programar aplicaciones en consola y paneles, es decir que el resultado saldrá en paneles debajo del código fuente ejecutado.

Aplicación: Programa que eleva un número 2 al cubo.
(consola)



```
7  *
8  * @author familia
9  */
10 public class suma {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String[] args) {
16         int dato1 ,dato2,potencia;
17         dato1=2;
18         dato2=3;
19         potencia=(int) Math.pow(dato1, dato2);
20         System.out.println(potencia);
21     }
22 }
23 }
24 }
```

Output - operaciones (run)

```
run:
8
BUILD SUCCESSFUL (total time: 0 seconds)
```

Aplicación: Programa que calcula área del triángulo.

(consola)

```

7      *
8      * @author familia
9      */
10     public class triangulo {
11
12         /**
13          * @param args the command line arguments
14          */
15         public static void main(String[] args) {
16             double area,base,altura;
17             base=5;
18             altura=7;
19             area=(base*altura)/2;
20             System.out.println("el area es: "+area);
21         }
22     }
23

```

Output - operaciones (run)

```

run:
el area es: 17.5
BUILD SUCCESSFUL (total time: 0 seconds)

```

Aplicación: Programa que calcula el área del triángulo

pidiendo datos. (consola)

```

10
11     * @author familia
12     */
13     public class triangulo1 {
14
15         /**
16          * @param args the command line arguments
17          */
18         public static void main(String[] args) {
19             double area,base,altura;
20             Scanner mide= new Scanner(System.in);//lee los datos
21             System.out.println("ingresa la base:");
22             base=mide.nextDouble();
23             System.out.println("ingresa la altura:");
24             altura=mide.nextDouble();
25             area=(base*altura)/2;
26             System.out.println("la altura es: "+area);
27         }
28     }
29

```

Output - operaciones (run)

```

run:
ingresa la base: 3
ingresa la altura: 4
la altura es: 6.0
BUILD SUCCESSFUL (total time: 7 seconds)

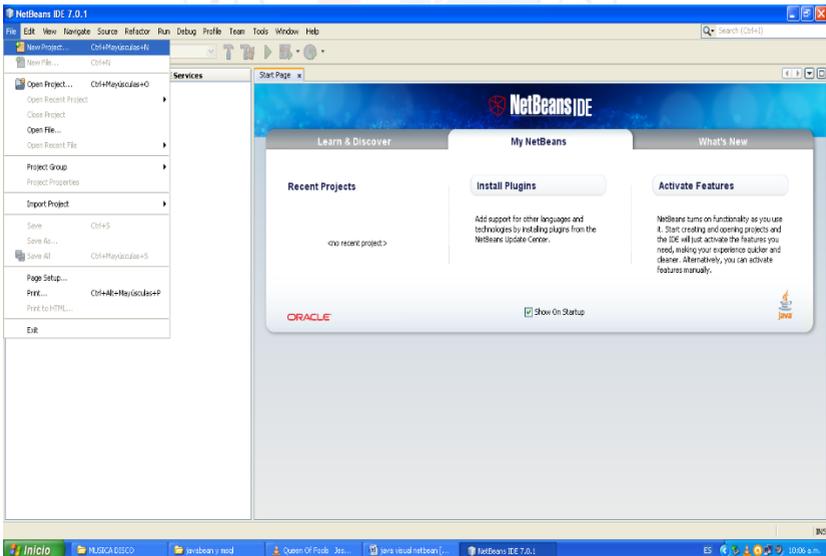
```

Aplicación: Programa que calcula el área del triángulo pidiendo datos. (panel)

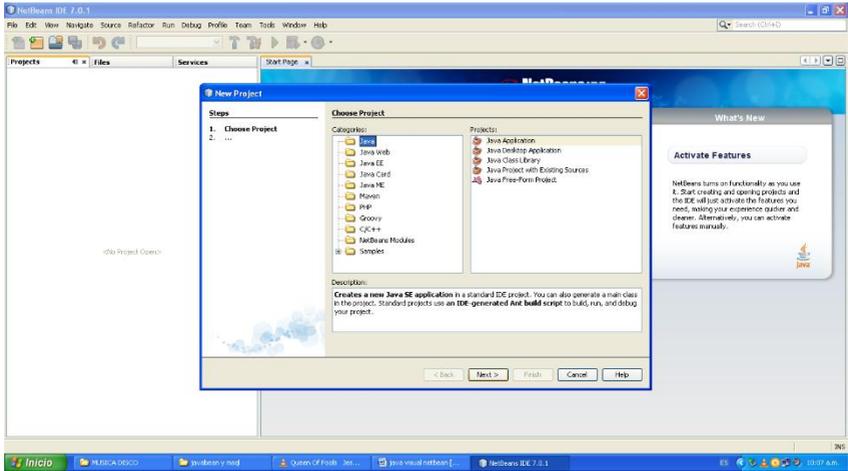


Java visual netbeans

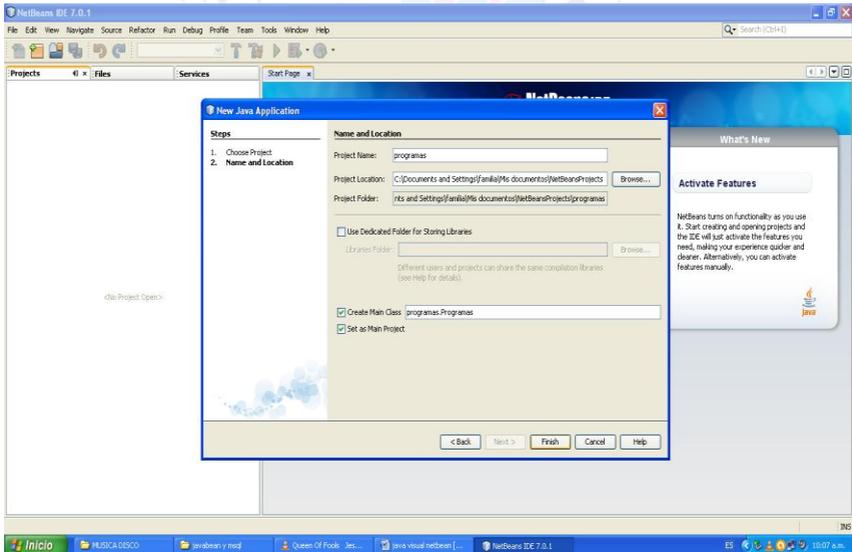
1.-Entorno: New project (nuevo proyecto)



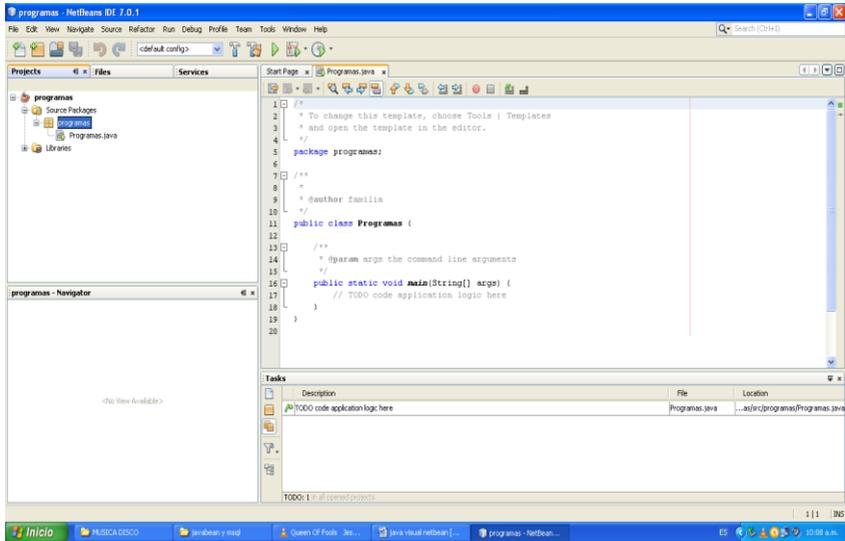
2.- Seleccione categories: carpeta Java y projects: Java application and clic en botón next



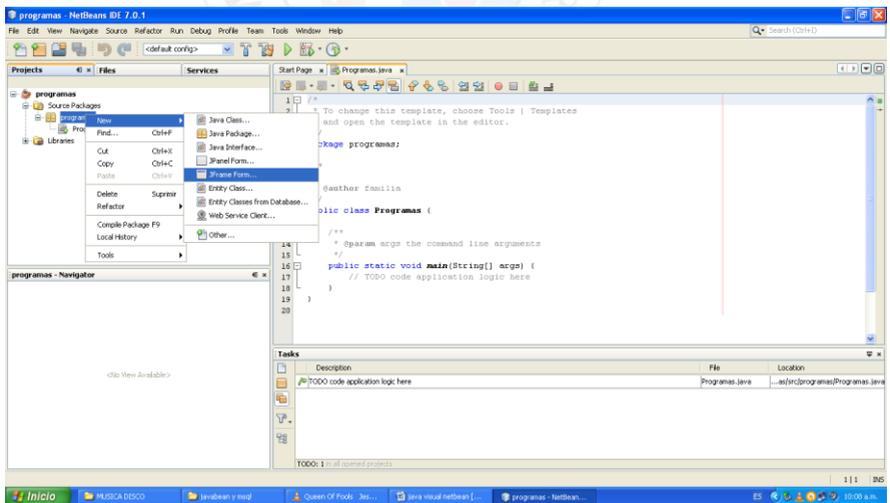
3.- Asignar nombre al proyecto y click en botón finalizar



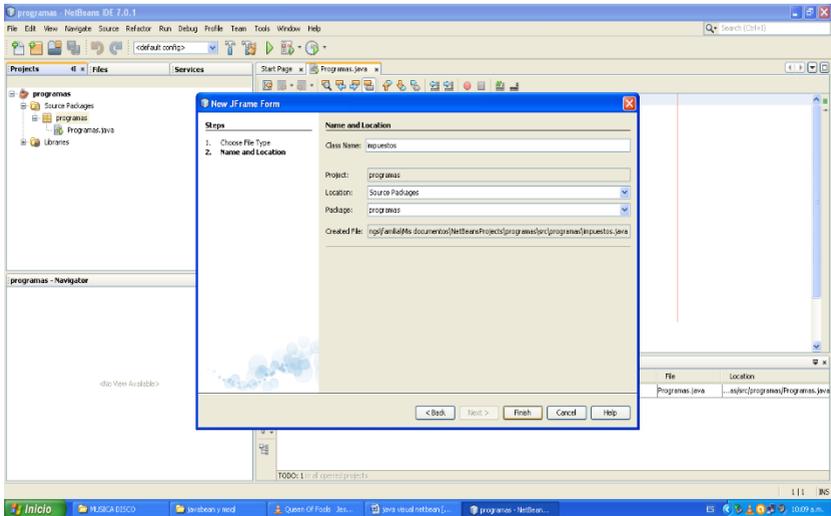
4.- Luego aparece el entorno con el nombre asignado:



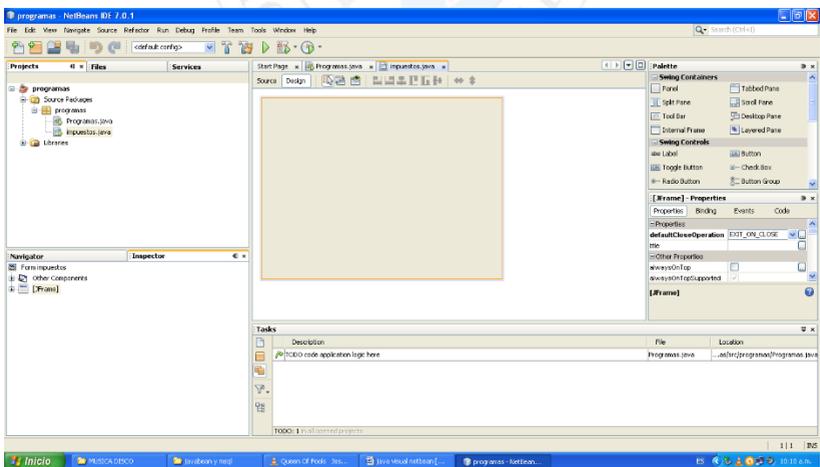
5.- Luego seleccione el paquete (programas) y clic derecho luego la opción new y elegir JFrame form para crear la aplicación



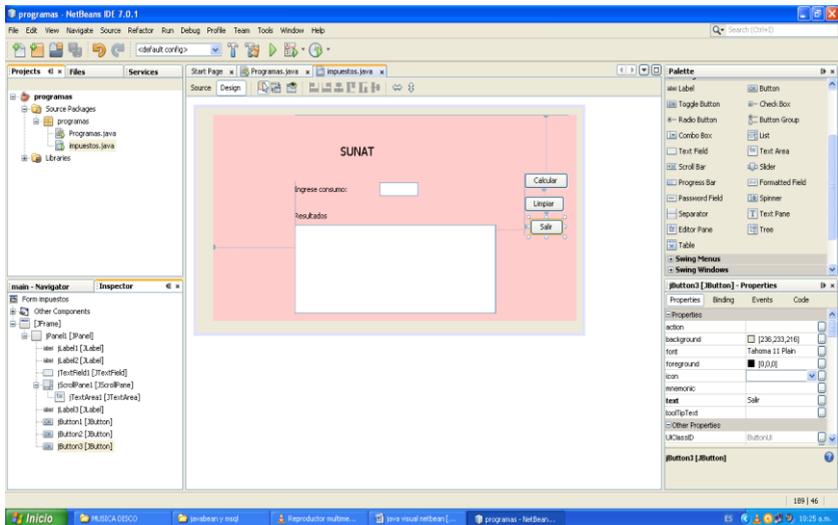
6.- Asignarle nombre a la aplicación y clic en botón finish para crearlo. Así puede crear más aplicaciones en el mismo paquete del proyecto.



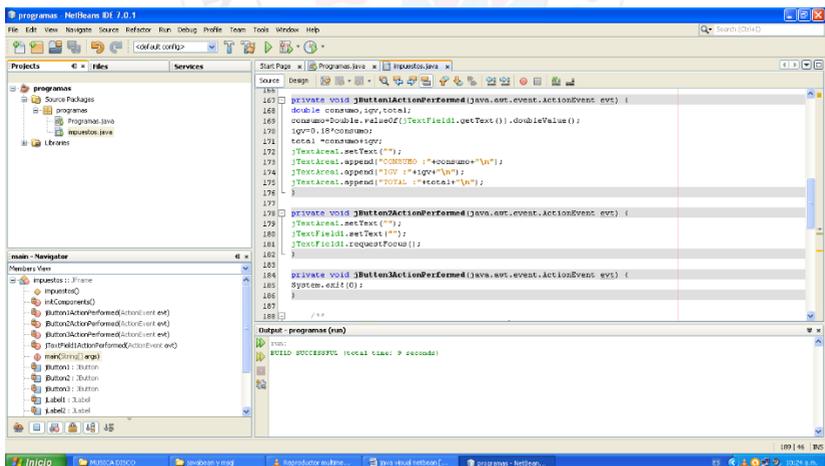
7.- Se apertura el formulario y las herramientas visuales



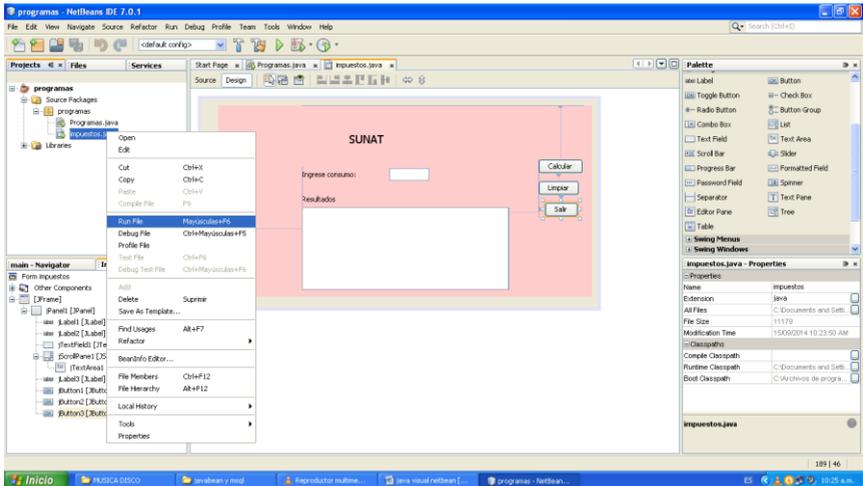
8.- Se diseña el formulario con los controles necesarios:



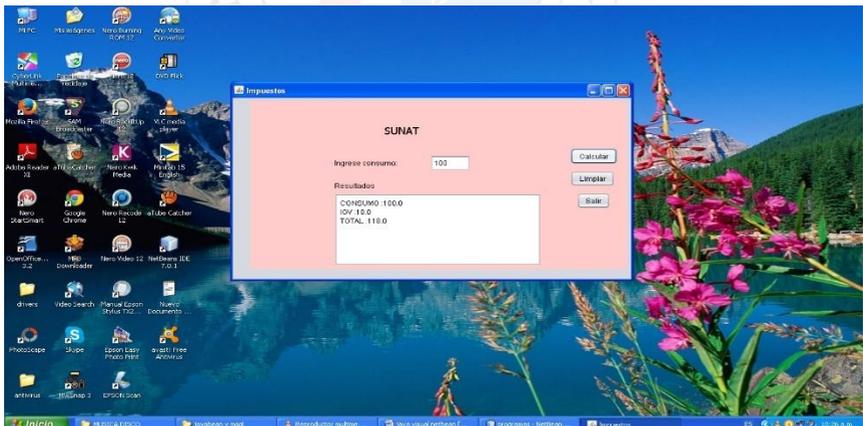
9.- Luego se codifica en cada botón las instrucciones



10.- Para ejecutar la aplicación y ver los resultados: clic derecho en la aplicación y seleccionar la opción Run File.



11. Se muestra el resultado final:



CÓDIGO:

Botón Calcular:

```
double consumo,igv,total;
```

declaración de variables

```
consumo=Double.valueOf(jTextField1.getText()).doubleValue  
();
```

Lectura de datos

```
igv=0.18*consumo;
```

Proceso

```
total =consumo+igv;
```

Proceso

```
jTextArea1.setText("");
```

Limpia

```
jTextArea1.append("CONSUMO :"+consumo+"\n");
```

salida

```
jTextArea1.append("IGV :"+igv+"\n");
```

```
jTextArea1.append("TOTAL :"+total+"\n");
```

Botón Limpiar:

```
jTextArea1.setText("");
```

```
jTextField1.setText("");
```

```
jTextField1.requestFocus();
```

Botón Salir :

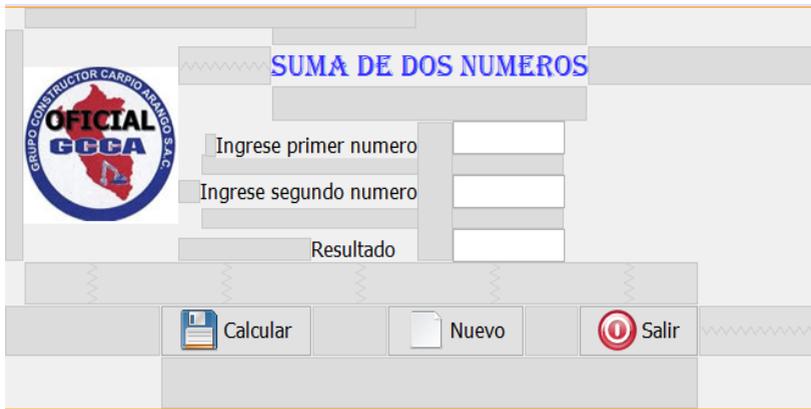
```
System.exit(0);
```

12.- Si desea llevarse a su memoria USB el proyecto y aplicaciones: explore el disco C:/mis documentos /NetBeans Projects/ de ahí copie la carpeta del proyecto a su memoria USB y listo se habrá grabado el proyecto, paquetes y aplicaciones.

2.2 Aplicaciones visuales educativas:

2.2.1 Aplicaciones visuales secuenciales

Aplicación 1: Programa que suma dos números



Codificación:

Botón Calcular :

```
float n1,n2,s;  
n1=Float.parseFloat(jTextField1.getText());  
n2=Float.parseFloat(jTextField2.getText());  
s=n1+n2;  
jTextField3.setText("");  
jTextField3.setText(""+s);
```

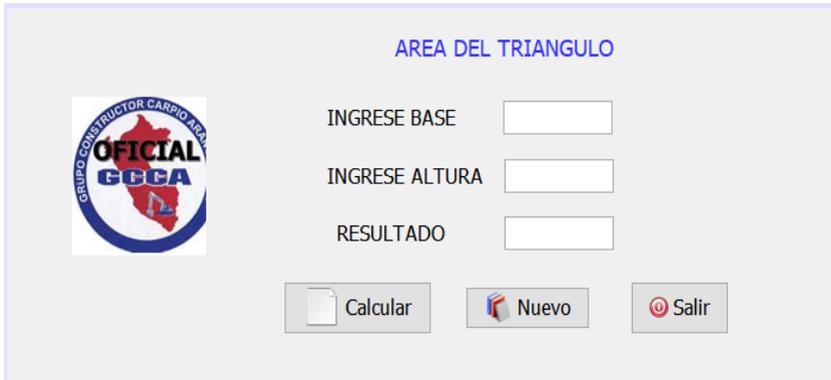
Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField1.requestFocus();
```

Botón Salir :

```
System.exit(0);
```

Aplicación 2: Programa que calcula el Área del triángulo $a = (b \cdot h) / 2$



Codificación:

Botón Suma :

```
float b,h,a;  
b=Float.parseFloat(jTextField1.getText());  
h=Float.parseFloat(jTextField2.getText());  
a=(b*h)/2;  
jTextField3.setText("");  
jTextField3.setText(""+a);
```

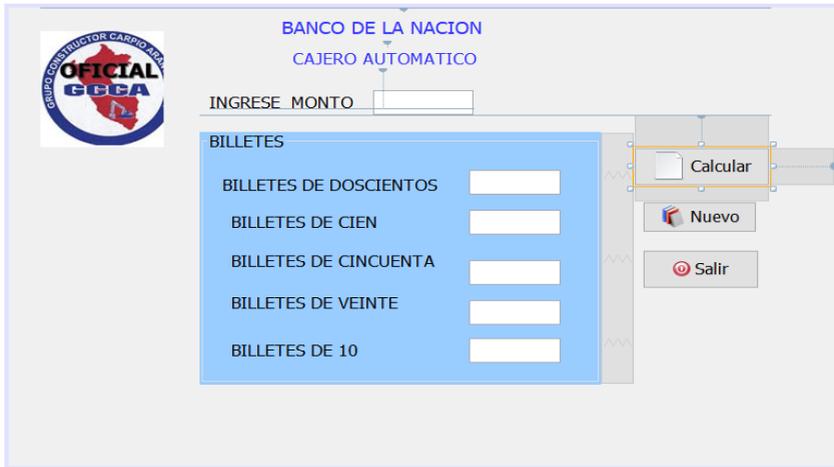
Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField1.requestFocus();
```

Botón Salir :

```
System.exit(0);
```

Aplicación 3: Programa que simula un cajero automático



Codificación:

Botón Calcular

```
int monto,dosc,cien,cinc,vein,diez;
monto=Integer.parseInt(jTextField1.getText());
dosc=monto/200;
monto=monto%200;
cien=monto/100;
monto=monto%100;
cinc=monto/50;
monto=monto%50;
vein=monto/20;
monto=monto%20;
diez=monto/10;
monto=monto%10;
jTextField2.setText(""+dosc);
jTextField3.setText(""+cien);
jTextField4.setText(""+cinc);
jTextField5.setText(""+vein);
jTextField6.setText(""+diez);
```

Botón Nuevo

```
jTextField1.setText("");
```

```
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField6.setText("");  
jTextField1.requestFocus();
```

Botón Salir

```
System.exit(0);
```

Aplicación 4: Programa que calcula las ventas de una tienda



Codificación:

Botón Calcular:

```
float precio,cantidad,importe,igv,totalpago;  
precio=Float.parseFloat(jTextField1.getText());  
cantidad=Float.parseFloat(jTextField2.getText());  
importe=precio*cantidad;  
igv=(float)0.18*importe;  
totalpago=importe+igv;  
jTextField3.setText("");
```

```
jTextField3.setText(""+importe);  
jTextField4.setText("");  
jTextField4.setText(""+igv);  
jTextField5.setText("");  
jTextField5.setText(""+totalpago);
```

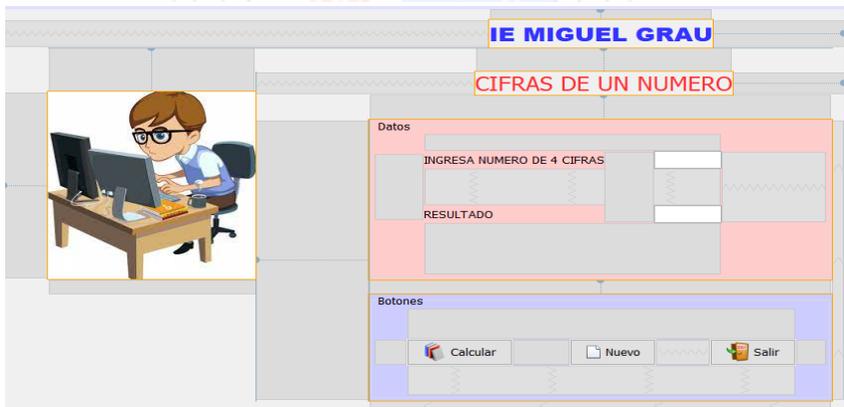
Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField1.requestFocus();
```

Botón Salir :

```
System.exit(0);
```

Aplicación 5: Programa que suma cifras de 4 dígitos



Codificación:

Botón Calcular:

```
int numero,c1,c2,c3,c4,suma;  
numero=Integer.parseInt(jTextField1.getText());
```

```
c1=numero/1000;  
numero=numero% 1000;  
c2=numero/100;  
numero=numero% 100;  
c3=numero/10;  
numero=numero% 10;  
c4=numero/1;  
numero=numero% 1;  
suma=c1+c2+c3+c4;  
jTextField2.setText("");  
jTextField2.setText(""+suma);
```

Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 6: Programa que calcula el área del círculo $a=3.141516*r^2$



Codificación:

Botón Calcular:

```
float r,area;  
r=Float.parseFloat(jTextField1.getText());  
area= (float)3.1416*r*r;  
jTextField2.setText("");  
jTextField2.setText(""+area);
```

Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

2.2.2 Aplicaciones visuales con estructura de control if ..else

Aplicación 7: Programa de seguridad y enlace usando if



SISTEMA DE SEGURIDAD



USUARIO

CLAVE

Codificación:

Botón validar:

```
int clave;
```

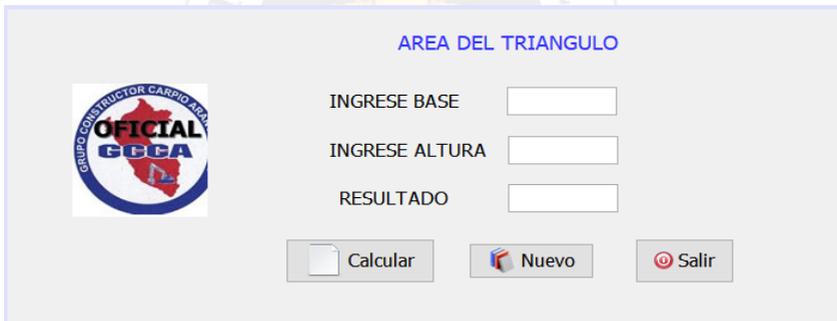
```
clave=Integer.parseInt(jPasswordField1.getText());
if( clave==123456)
{
    this.setVisible(false);
    venta c= new venta();
    c.setVisible(true);
}
```

venta: es el nombre del programa que es invocado o llamado

Botón Salir :

```
System.exit(0);
```

Aplicación 8: Programa que calcula el Área del triángulo
consistenciando los datos $a = (b * h) / 2$



Codificación:

Botón Calcular:

```
float b,h,a;
b=Float.parseFloat(jTextField1.getText());
h=Float.parseFloat(jTextField2.getText());
if (b>0 && h>0)
{
    a=(b*h)/2;
    jTextField3.setText("");
    jTextField3.setText(""+a);
}
else
```

```
jTextField3.setText("Error de datos “);
```

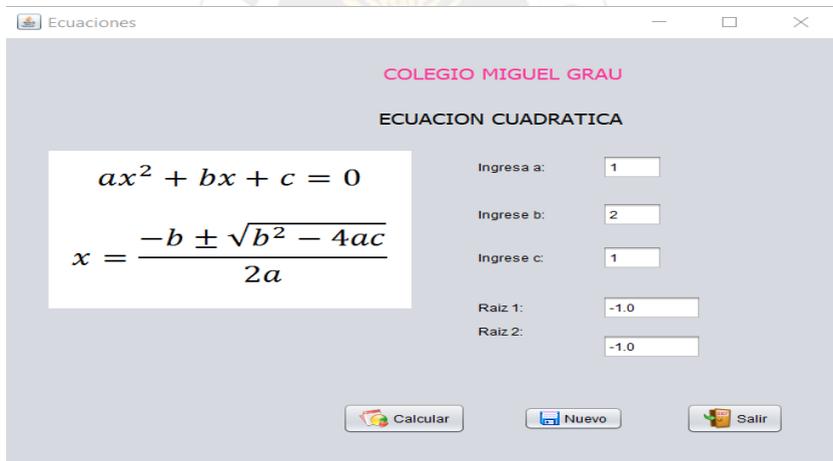
Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 9: Programa que calcula las raíces de una Ecuación cuadrática



Codificación:

Botón Calcular:

```
double a,b,c,r1,r2,d;  
a=Float.parseFloat(jTextField1.getText());  
b=Float.parseFloat(jTextField2.getText());  
c=Float.parseFloat(jTextField3.getText());  
d=b*b-4*a*c;  
if (a!=0 && d>=0)  
{
```

```
r1=(-b+(Math.sqrt(b*b-4*a*c)))/(2*a);  
r2=(-b-(Math.sqrt(b*b-4*a*c)))/(2*a);  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField4.setText(""+r1);  
jTextField5.setText(""+r2);  
}  
else
```

```
jTextField4.setText("Error de datos");
```

Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField1.requestFocus();
```

Botón Salir :

```
System.exit(0);
```

Aplicación 10: Programa que encuentra el mayor de 3 números



Codificación:

Botón calcular:

```
int n1,n2,n3,mayor;
n1=Integer.parseInt(jTextField1.getText());
n2=Integer.parseInt(jTextField2.getText());
n3=Integer.parseInt(jTextField3.getText());
if (n1>n2 && n1>n3)
{
    mayor=n1;
    jTextField4.setText("");
    jTextField4.setText(""+mayor);
}
if (n2>n1 && n2>n3)
{
    mayor=n2;
    jTextField4.setText("");
    jTextField4.setText(""+mayor);
}
if (n3>n1 && n3>n2)
{
    mayor=n3;
    jTextField4.setText("");
    jTextField4.setText(""+mayor);
}
if (n1==n2 && n1==n3)
{
    jTextField4.setText("los números son iguales");
}
}
```

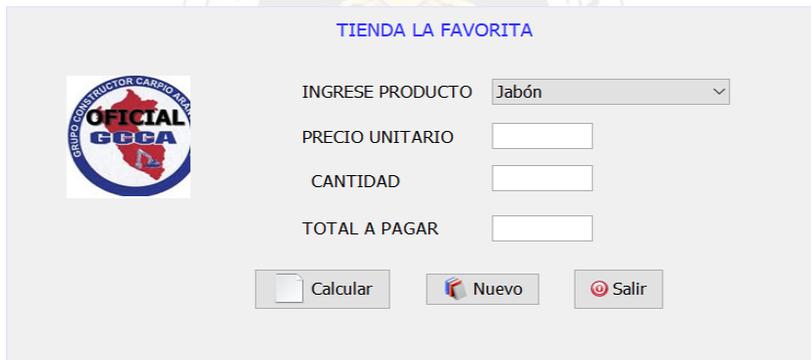
Boton Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField1.requestFocus();
```

Botón Salir:

System.exit(0);

Aplicación 11: Si se compra desde 100 o más unidades de un artículo se obtiene un descuento del 40%, si se compra desde 26 hasta 99 el descuento es del 20%, si se compra desde 10 hasta 25 unidades el descuento es del 12%. Para cantidades menores no hay descuento. Escribir un programa en Java que permita ingresar el costo unitario del artículo y la cantidad de unidades compradas y muestre el monto a pagar.



Codificación:

Botón Calcular:

```
float precio,cantidad,subtotal=0,total=0;
precio=Float.parseFloat(jTextField1.getText());
cantidad=Float.parseFloat(jTextField2.getText());
subtotal=precio*cantidad;
if (cantidad>100)
{
    total=(float)(subtotal-0.4*subtotal);
    jTextField3.setText("");
}
```

```
        jTextField3.setText(""+total);
    }
    if (cantidad>=26 && cantidad<=99)
    {
        total=(float)(subtotal-0.2*subtotal);
        jTextField3.setText("");
        jTextField3.setText(""+total);
    }
    if (cantidad>=10 && cantidad<=25)
    {
        total=(float)(subtotal-0.12*subtotal);
        jTextField3.setText("");
        jTextField3.setText(""+total);
    }
    if (cantidad<=10)
    {
        total= total=(float)(subtotal-0*subtotal);
        jTextField3.setText("");
        jTextField3.setText(""+total);
    }
}
```

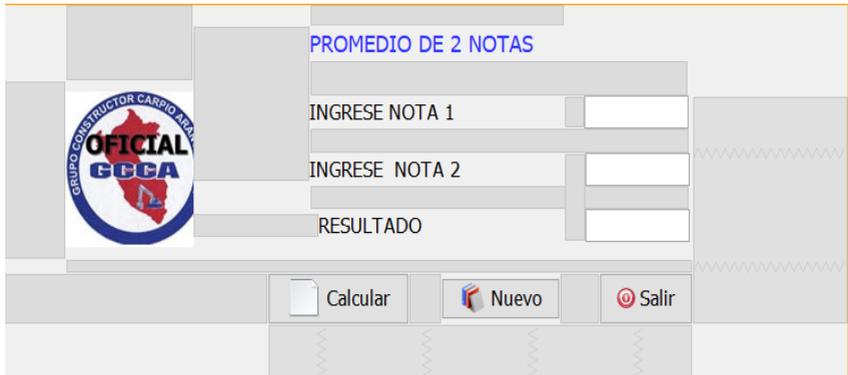
Botón Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 12: Programa que calcula el promedio de 2 notas válidas $prom = (n1+n2)/2$



Codificación:

Botón Promedio:

```
float n1,n2,prom;  
n1=Float.parseFloat(jTextField1.getText());  
n2=Float.parseFloat(jTextField2.getText());  
if (( n1>=0 && n1<=20) && ( n2>=0 && n2<=20))  
{  
    prom=(n1+n2)/2;  
    jTextField3.setText("");  
    jTextField3.setText(""+prom);  
}  
else  
    jTextField3.setText("Error de notas");
```

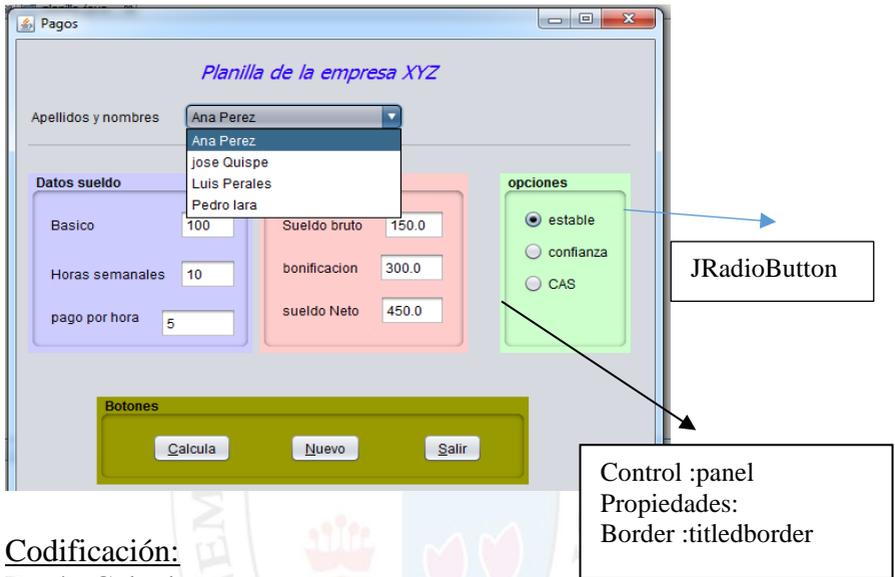
Boton Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 13: Programa que simula una planilla usando JRadioButton y JComboBox



Codificación:

Botón Calcular

```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
float basico, horas, pagohoras, boni=0, bruto, neto;
basico=Float.parseFloat(jTextField1.getText());
horas=Float.parseFloat(jTextField2.getText());
pagohoras=Float.parseFloat(jTextField3.getText());
bruto=basico+horas*pagohoras;
if (jRadioButton1.isSelected() == true)
{
boni=300;
}
if (jRadioButton2.isSelected() == true)
{
boni=200;
}
}
```

```
if (jRadioButton3.isSelected() == true)
{
boni=100;
}
neto =bruto+boni;
jTextField4.setText(""+bruto);
jTextField5.setText(""+boni);
jTextField6.setText(""+neto);
}
```

Botón Nuevo

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jRadioButton1.setSelected(false);
jRadioButton2.setSelected(false);
jRadioButton3.setSelected(false);
jTextField1.requestFocus();
}
```

Botón Salir

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}
```

Aplicación 14: Programa que simula un subsidio

The screenshot shows a Java Swing window titled "Subsidio" with a standard Mac OS-style title bar. The main content area has a light gray background and is titled "GOBIERNO DEL PERU" in blue. Below the title, the word "SUBSIDIO" is centered. There are three main sections: 1. "Apellidos y Nombres" with a text box containing "Ana" and a dropdown arrow. 2. "Datos" (highlighted in light purple) containing three text boxes: "Total de hijos" with "4", "Hijos en edad escolar" with "2", and "pago" with "960.0". 3. "Opciones" (highlighted in light green) containing two unchecked checkboxes: "Viuda" and "Otro". At the bottom, a "Botones" section (highlighted in olive green) contains three buttons: "Calcular" (with a calculator icon), "Nuevo" (with a document icon), and "Salir" (with a power icon).

Codificación:

Botón Calcular:

```
int nhijos,nhijosescolar;
double subsidio=0,total;
nhijos=Integer.parseInt(jTextField1.getText());
nhijosescolar=Integer.parseInt(jTextField2.getText());
if (nhijos<=2)
{
    subsidio=380;
}
if (nhijos>2 &&nhijos<=5)
{
    subsidio=760;
}
if (nhijos>5)
{
```

```
        subsidio=950;
    }
    total=subsidio+nhijosescolar*100;
    if (jCheckBox1.isSelected()==true)
    {
        total=subsidio+nhijosescolar*100+200;
    }
    if (jCheckBox2.isSelected()==true)
    {
        total=subsidio+nhijosescolar*100+250;
    }
    jTextField3.setText("");
    jTextField3.setText(""+total);
```

Botón Nuevo:

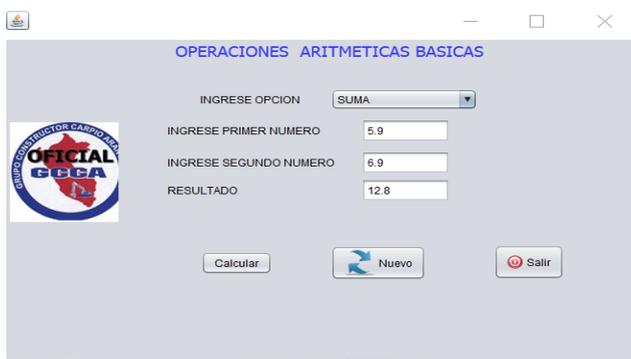
```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jCheckBox1.setSelected(false);
jCheckBox2.setSelected(false);
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

2.2.3 Aplicaciones visuales con estructura de control switch case

Aplicación 15: Programa con las 4 operaciones aritméticas



Codificación :

Botón procesar :

```
int op;
float n1,n2,r;
n1=Float.parseFloat(jTextField1.getText());
n2=Float.parseFloat(jTextField2.getText());
op=jComboBox1.getSelectedIndex();
switch (op)
{
case 0:
{
r=n1+n2;
jTextField3.setText(""+r);
break;
}
case 1:
{
r=n1-n2;
jTextField3.setText(""+r);
break;
}
case 2:
{
r=n1*n2;
```

```
        jTextField3.setText(""+r);  
        break;  
    }  
    case 3:  
    {  
        r=n1/n2;  
        jTextField3.setText(""+r);  
        break;  
    }  
}
```

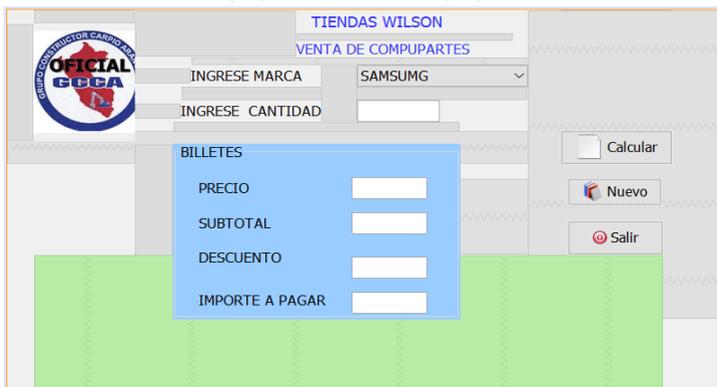
Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 16: Programa que simula una tienda de accesorios de computadoras vende monitores de diferentes marcas y cada marca tiene un porcentaje de descuento como sigue a continuación:



Codificación:

Botón Calcular:

```
int marca,cantidad;
float subtotal=0,descuento=0,importe=0,precio=0;
marca=jComboBox1.getSelectedIndex();
cantidad=Integer.parseInt(jTextField1.getText());
switch (marca)
{
case 0:
{
precio= 200;
subtotal=precio*cantidad;
descuento= (float)(subtotal*0.10);
importe=subtotal-descuento;
}
break;
case 1:
{
precio= 300;
subtotal=precio*cantidad;
descuento= (float)(subtotal*0.15);
importe=subtotal-descuento;
break;
}
case 2:
{
precio= 400;
subtotal=precio*cantidad;
descuento= (float)(subtotal*0.13);
importe=subtotal-descuento;
break;
}
case 3:
{
```

```
precio= 250;
subtotal=precio*cantidad;
descuento= (float)(subtotal*0.12);
importe=subtotal-descuento;
break;
}
case 4:
{
precio= 220;
subtotal=precio*cantidad;
descuento= (float)(subtotal*0.17);
importe=subtotal-descuento;
break;
}
}
jTextField2.setText(""+precio);
jTextField3.setText(""+subtotal);
jTextField4.setText(""+descuento);
jTextField5.setText(""+importe);
```

Botón Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0)
```

2.2.4 Aplicaciones visuales con estructura de control repetitiva while

Aplicación 17: Programa que suma la serie numérica
 $S=1+2+3+4..+n$



Codificación:

Botón procesar:

```
int s=0,i=1,n;  
n= Integer.parseInt(jTextField1.getText());  
while(i<=n)  
{  
    s=s+i;  
    i=i+1;  
}  
jTextField2.setText("");  
jTextField2.setText(""+s);
```

Botón Nuevo:

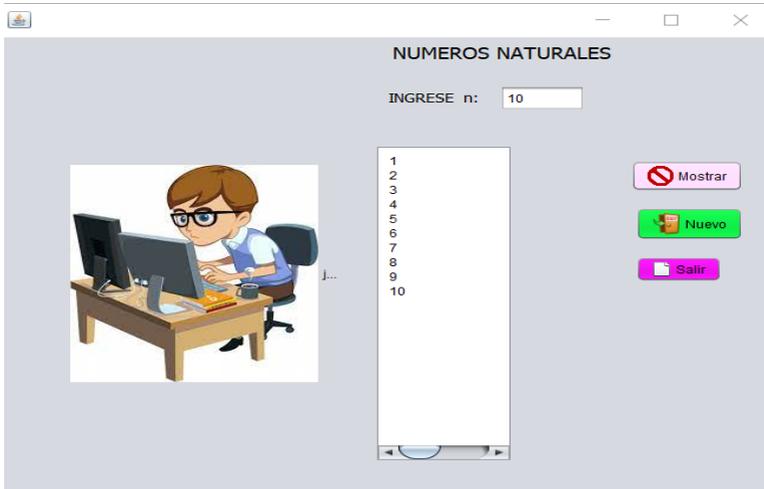
```
jTextField1.setText("");
```

```
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 18: Programa que muestra números naturales



Codificación:

Botón procesar:

```
int i=1,n;  
n= Integer.parseInt(jTextField1.getText());  
while (i<=n)  
{  
jTextArea1.append(" "+i+"\n");  
i=i+1;  
}
```

Botón Nuevo :

```
jTextField1.setText("");
```

```
jTextArea1.setText("");  
jTextField1.requestFocus();
```

Botón Salir :

```
System.exit(0);
```

Aplicación 19: Programa que muestra el número de dígitos de un número



Codificación:

Botón conteo:

```
int c=0,n;  
n= Integer.parseInt(jTextField1.getText());  
while (n>0)  
{  
    n=(n/10);  
    c=c+1;  
}  
jTextField2.setText(""+c);
```

Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");
```

```
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 20: Programa que muestra el número de dígitos pares de un número



Codificación:

Botón conteo:

```
int d,c=0,n;  
n= Integer.parseInt(jTextField1.getText());  
while (n>0)  
{  
    d=(n%10);  
    if (d%2==0)  
    {  
        c=c+1;  
    }  
    n=n/10;  
}  
jTextField2.setText(""+c);
```

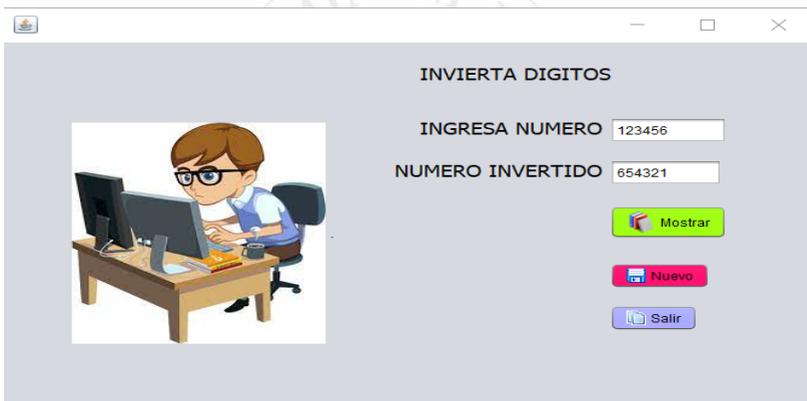
Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 21: Programa que muestra número de dígitos invertidos



Codificación:

Botón conteo:

```
int d,n,i=0;  
n= Integer.parseInt(jTextField1.getText());  
while (n>0)  
{  
    d=(n%10);  
    n=n/10;  
    i=i*10+d;  
}  
jTextField2.setText(""+i);
```

Botón Nuevo:

```
jTextField1.setText("");
```

```
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 22: Programa que detecta si un número es Cubo perfecto ($153 = 1^3 + 5^3 + 3^3$)



Codificación:

Botón conteo:

```
int t,d,s=0,n;  
n= Integer.parseInt(jTextField1.getText());  
t=n;  
while (t>0)  
{  
    d=(t%10);  
    t=t/10;  
    s=s+d*d*d;  
}  
if (n==s)  
    jTextField2.setText("CUBO PERFECTO");  
else
```

```
jTextField2.setText("NO ES CUBO PERFECTO");
```

Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 23: Programa que detecta el número mayor de varios dígitos



Codificación:

Botón conteo:

```
int d,m=0,n;  
n= Integer.parseInt(jTextField1.getText());  
while (n>0)  
{  
    d=n%10;  
    if (d>m)  
    {  
        m=d;  
    }  
}
```

```
n=n/10;  
}  
jTextField2.setText(""+m);
```

Botón Nuevo:

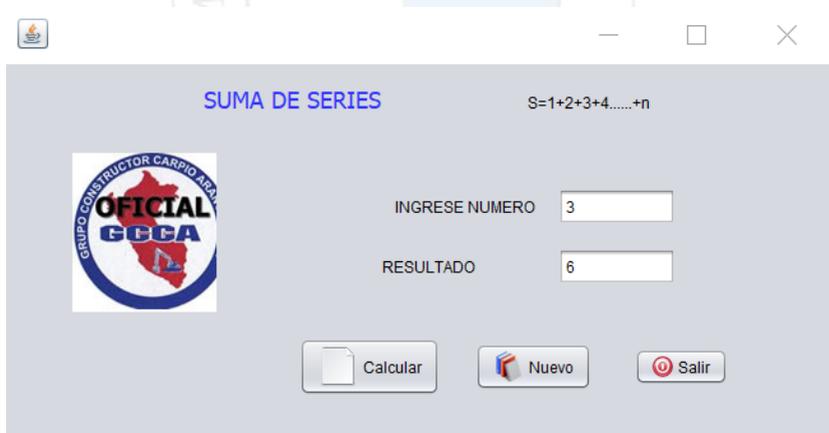
```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

2.2.5 Aplicaciones visuales con estructura de control repetitiva do while

Aplicación 24: Programa que suma la serie numérica
 $S=1+2+3+4+...+n$



Codificación:

Botón Calcular:

```
int s=0,i=1,n;  
n= Integer.parseInt(jTextField1.getText());  
do
```

```
{  
    s=s+i;  
    i=i+1;  
}while(i<=n);  
jTextField2.setText("");  
jTextField2.setText(""+s);
```

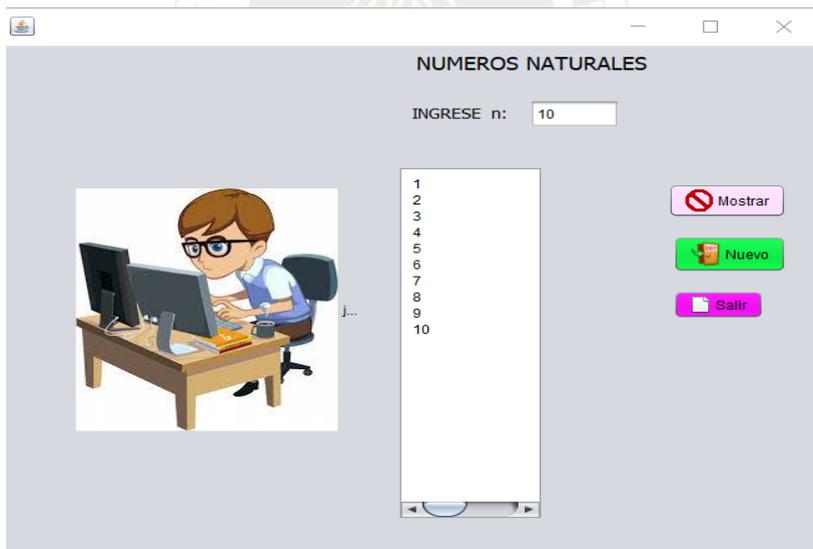
Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 25: Programa que muestra números naturales



Codificación:

Botón procesar:

```
int i=1,n;  
n= Integer.parseInt(jTextField1.getText());
```

```
do
{
jTextArea1.append(" "+i+"\n");
i=i+1;
while (i<=n);
}
```

Botón Nuevo:

```
jTextField1.setText("");
jTextArea1.setText("");
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 26: Programa que suma números fraccionarios



Codificación:

Botón procesar:

```
float s=0,i=1,n;
n= Integer.parseInt(jTextField1.getText());
```

```
do
{
    s=s+(1/i);
    i=i+1;
}
while (i<=n);
jTextField2.setText(""+s);
```

Botón Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField1.requestFocus();
```

Botón Salir:

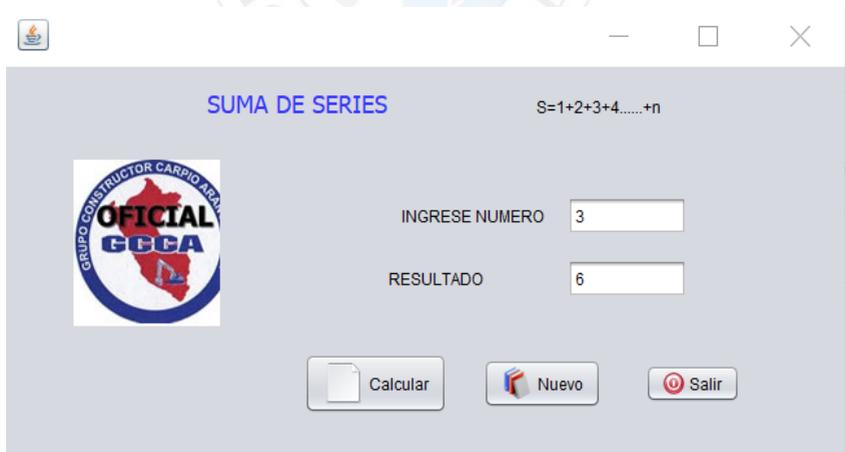
```
System.exit(0);
```

2.2.6 Aplicaciones visuales la estructura de control repetitiva

for

Aplicación 27: Programa que suma la serie numérica

$S=1+2+3+4+.....+n$



Codificación:

Botón procesar:

```
int s=0,i,n;  
n= Integer.parseInt(jTextField1.getText());  
for(i=1;i<=n; i++)  
{  
    s=s+i;  
}
```

```
jTextField2.setText("");  
jTextField2.setText(""+s);
```

Botón Nuevo:

```
    jTextField1.setText("");  
    jTextField2.setText("");  
    jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 28: Programa que simula una tabla de multiplicar



Codificación:

Botón procesar:

```
int i,n;  
n= Integer.parseInt(jTextField1.getText());  
for(i=1;i<=12;i++)  
{  
jTextArea1.append(" "+i+"*" +n+"="+i*n+"\n");  
}
```

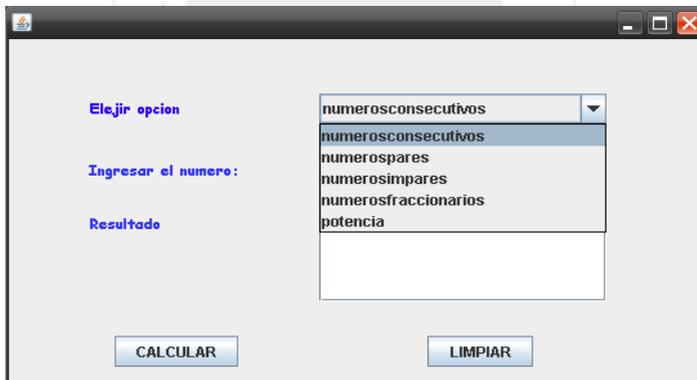
Botón Nuevo:

```
jTextField1.setText("");  
jTextArea1.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 29: Programa que Suma de series numéricas



Codificación:

Botón Calcular:

```
private void  
calcularActionPerformed(Java.awt.event.ActionEvent evt) {  
int i,n,s,var;  
s=0; n=Integer.valueOf(jTextField1.getText()).intValue();
```

```
var=jComboBox1.getSelectedIndex();
switch(var)
{
    case 0:
        { for (i=1;i<=n;i++){
            s=s+i;
        }
        break;
    }
    case 1:
        { for (i=1;i<=n;i++){
            s=s+(2*i);
        }
        break;
    }
    case 2:
        { for (i=1;i<=n;i++){
            s=s+(2*i-1);
        }
        break;
    }
    case 3:
        { for (i=1;i<=n;i++){
            s=s+(1/i);
        }
        break;
    }
    case 4:
        { for (i=1;i<=n;i++){
            s=s+(i*i);
        }
        break;
    }
}
```

```
TextArea1.setText("");  
TextArea1.append("La suma de la serie es: "+s+"\n");  
}
```

Botón Limpiar:

```
private void  
limpiarActionPerformed(Java.awt.event.ActionEvent evt) {  
    jTextField1.setText("");  
    jTextArea1.setText("");  
    jTextField1.requestFocus();  
}
```

2.2.7 Aplicaciones con menús:

Aplicación 30: Creación de menús en Java



Pasos:

- 1.- Crear un proyecto
- 2.- Crear un formulario principal donde ira el menú (principal)
- 3.- Crear otro formularios que serán enlazados (cuadrado, triángulo y círculo)

4.- En el formulario principal codificar de acuerdo al sub menú lo siguiente:

Sub menu cuadrado:

```
private void jMenuItem1ActionPerformed(Java.awt.event.ActionEvent evt)
{
    cuadrado vercuadrado=new cuadrado();
    vercuadrado.setVisible(true);
}
```

Sub menú triangulo :

```
triangulo vertriangulo=new triangulo();
vertriangulo.setVisible(true);
```

Sub menú circulo :

```
private void jMenuItem3ActionPerformed(Java.awt.event.ActionEvent evt)
{
    circulo vercirculo=new circulo();
    vercirculo.setVisible(true);
}
```

Código del formulario: cuadrado



Botón Calcular:

```
private void
jButton1ActionPerformed(Java.awt.event.ActionEvent evt) {
    float lado,area;
    lado=Float.parseFloat(jTextField1.getText());
    if ( lado>0)
    {
```

```
        area=lado*lado;
        jTextField2.setText(""+area);
    }
    else
        jTextField2.setText("Error de datos");
}
```

Botón Nuevo:

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField1.requestFocus();
}
```

Botón Salir:

```
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
```

código del formulario: cuadrado



Botón calcular:

```
private void
jButton1ActionPerformed(Java.awt.event.ActionEvent evt) {
    float base,altura,area;
    base =Float.parseFloat(jTextField1.getText());
    altura =Float.parseFloat(jTextField2.getText());
    if (base>0 && altura>0)
    {
        area=(base*altura)/2;
        jTextField3.setText(""+area);
    }
    else
        jTextField3.setText("eroor de datos");
}
```

Botón Nuevo:

```
private void
jButton2ActionPerformed(Java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField1.requestFocus(); }
}
```

Botón Salir:

```
private void
jButton3ActionPerformed(Java.awt.event.ActionEvent evt) {
    dispose();
}
```

código del formulario: circulo



Botón Calcular:

```
private void
jButton1ActionPerformed(Java.awt.event.ActionEvent evt) {
    float r,area;
    r=Float.parseFloat(jTextField1.getText());
    if ( r>0)
    {
        area= (float)3.1416*r*r;
        jTextField2.setText(""+area);
    }
    else
        jTextField2.setText("error de datos");
}
```

Botón Nuevo:

```
private void
jButton2ActionPerformed(Java.awt.event.ActionEvent evt) {
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField1.requestFocus();
}
```

Botón Salir:

```
private void
jButton3ActionPerformed(Java.awt.event.ActionEvent evt) {
```

```
dispose();  
}
```

2.2.8 Aplicaciones con arreglos

Aplicación 31: Programa que suma 5 números en un arreglo unidimensional



Codificación:

Botón procesar:

```
int i,suma=0;  
int n[]=new int[5];  
n[0]=Integer.parseInt(jTextField1.getText());  
n[1]=Integer.parseInt(jTextField2.getText());  
n[2]=Integer.parseInt(jTextField3.getText());  
n[3]=Integer.parseInt(jTextField4.getText());  
n[4]=Integer.parseInt(jTextField5.getText());
```

```
for(i=0;i<5;i++)  
{  
    suma=suma+n[i];  
}  
jTextField6.setText("");  
jTextField6.setText(""+suma);
```

Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 32: Programa que promedia 5 números en un arreglo unidimensional



Codificación:

Botón procesar:

```
int i,suma=0;
float prom=0;
int n[]=new int[5];
n[0]=Integer.parseInt(jTextField1.getText());
n[1]=Integer.parseInt(jTextField2.getText());
n[2]=Integer.parseInt(jTextField3.getText());
n[3]=Integer.parseInt(jTextField4.getText());
n[4]=Integer.parseInt(jTextField5.getText());
for(i=0;i<5;i++)
{
    suma=suma+n[i];
}
prom=suma/5;
jTextField6.setText("");
jTextField6.setText(""+prom);
```

Botón Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 33: Programa que suma 6 números en un arreglo bidimensional.



Botón procesar:

```
int i,j,suma=0;
int n[][]=new int[3][2];
n[0][0]=Integer.parseInt(jTextField1.getText());
n[0][1]=Integer.parseInt(jTextField2.getText());
n[1][0]=Integer.parseInt(jTextField3.getText());
n[1][1]=Integer.parseInt(jTextField4.getText());
n[2][0]=Integer.parseInt(jTextField5.getText());
n[2][1]=Integer.parseInt(jTextField5.getText());
for(i=0;i<=2;i++)
    for(j=0;j<=1;j++)
        suma=suma+n[i][j];
jTextField7.setText("");
jTextField7.setText(""+suma);
```

Botón Nuevo:

```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField6.setText("");  
jTextField7.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 34: Programa que promedia 9 números en un arreglo bidimensional.



Botón procesar:

```
int i,j,suma=0;
float prom;
int n[][]=new int[3][3];
n[0][0]=Integer.parseInt(jTextField1.getText());
n[0][1]=Integer.parseInt(jTextField2.getText());
n[0][2]=Integer.parseInt(jTextField3.getText());
n[1][0]=Integer.parseInt(jTextField4.getText());
n[1][1]=Integer.parseInt(jTextField5.getText());
n[1][2]=Integer.parseInt(jTextField6.getText());
n[2][0]=Integer.parseInt(jTextField7.getText());
n[2][1]=Integer.parseInt(jTextField8.getText());
n[2][2]=Integer.parseInt(jTextField9.getText());
for(i=0;i<3;i++)
for(j=0;j<3;j++)
suma=suma+n[i][j];
prom=suma/9;
jTextField10.setText("");
jTextField10.setText(""+prom);
```

Botón Nuevo:

```
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");
jTextField6.setText("");
jTextField7.setText("");
jTextField8.setText("");
jTextField9.setText("");
jTextField10.setText("");
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

Aplicación 35: Programa que encuentra el mayor de 5 números en un arreglo unidimensional



Botón procesar:

```
int i,m=0;
int n[]=new int[5];
n[0]=Integer.parseInt(jTextField1.getText());
n[1]=Integer.parseInt(jTextField2.getText());
n[2]=Integer.parseInt(jTextField3.getText());
n[3]=Integer.parseInt(jTextField4.getText());
n[4]=Integer.parseInt(jTextField5.getText());
for(i=0;i<5;i++)
{
    if (n[i]>m)
    {
        m=n[i];
    }
}
```

```
}  
jTextField6.setText("");  
jTextField6.setText(""+m);
```

Botón Nuevo:

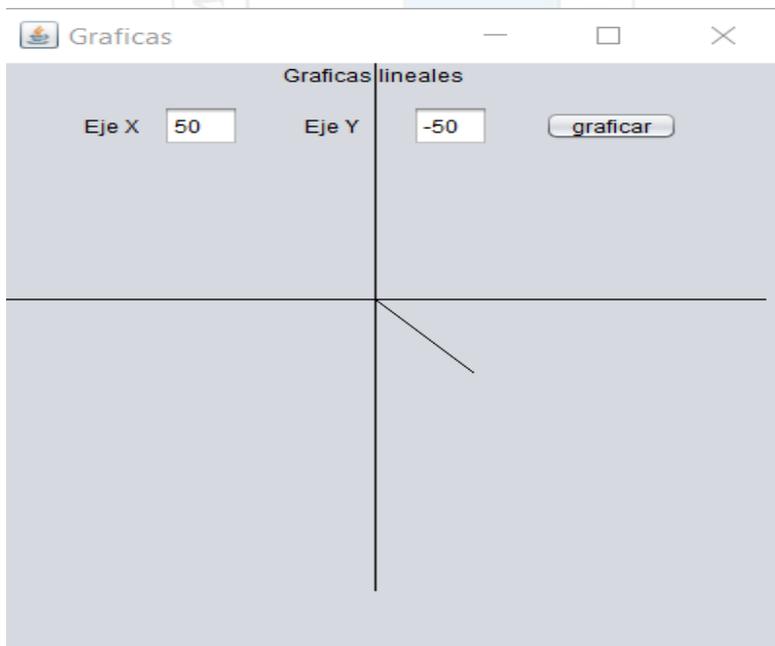
```
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextField5.setText("");  
jTextField1.requestFocus();
```

Botón Salir:

```
System.exit(0);
```

2.2.8 Gráficos en Java:

Aplicación 36: Creación de gráficas en Java.



Codificación:

```
package graficar1;
import Java.awt.Graphics;
public class inicio extends Javax.swing.JFrame {
    public inicio() {
        initComponents();
    }
private void
btngraficarActionPerformed(Java.awt.event.ActionEvent evt) {
    Graphics g =this.getGraphics();
    g.drawLine(200,0, 200,400);
    g.drawLine(0,200, 400,200 );
    int x,y;
    x=Integer.parseInt(txt1.getText());
    y=Integer.parseInt(txt2.getText());
    if (x>=0)
    { x=x+200;}
    if (y>=0)
    { y=y;}
    else if (y<=0)
    {
        y=y*-1;
        y=y+200;}
    g.drawLine(x, y, 200,200);
}
```

2.2.9 Ejercicios propuestos:

1. Hacer un programa que calcule el volumen de una esfera.
2. Hacer un programa que dado 3 notas calcule el promedio.
3. Hacer un programa que simule una boleta de venta.

4. Hacer un programa que sume las siguientes serie:
 $s=2+4+6+\dots+n$ (pares)
5. Hacer un programa que sume las siguientes serie:
 $s=1+3+5+\dots+n$ (impares)
6. Hacer un programa que sume las siguientes serie:
 $s=1^3+2^3+3^3+\dots+n^3$ (cubos)
7. Hacer un programa que sume las siguientes serie:
 $s=1+1/2+1/3+\dots+1/n$ (fraccionario)
8. Hacer un programa que sume las siguientes series:
 $s=2^2+4^2+6^2+\dots+n^2$ (pares al cuadrado)
9. Hacer un programa que ingresado un número de 3 cifras sume sus cifras.
10. Hacer un programa que ingrese 6 números en un arreglo de dos dimensiones (matriz) (3x2) y obtener la suma de los números ingresados.
11. Hacer un programa que ingrese 5 números y los ordene ascendentemente.
12. Hacer un programa que ingrese 5 notas y calcule su promedio.
13. Hacer un programa que ingrese 5 números y ubicar el mayor y menor.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- Abarca R. (2006). *Teoría del aprendizaje constructivista*. tercera edición.
- Alonso, J. (1991). *Motivación y aprendizaje en el aula. Cómo enseñar a pensar*. Madrid: Santillana.
- Coloma, O. (2005). *Informática y Software Educativo*. Lumbreas Editores. 1ra. Edición. Perú
- Joyanes, L. (2008). *Fundamentos de programación*. Editorial: s.a. Mcgraw-Hill / interamericana de España.
- Oracle (2010). *Oracle and Sun Microsystems*. Disponible en: <https://www.oracle.com/sun/>
- Papert, S. (1987). *Desafío a La Mente: Computadoras y Educación*. Buenos Aires: Impreso en ediciones Galápagos.
- Perez, J., Merino, M. (2009). *Definición de Lenguaje de programación*. Disponible en: <https://definicion.de/lenguaje-de-programacion>
- Morales, R. (2014). *Lenguajes de programación: ¿qué son y para qué sirven?*. Disponible en: <https://colombiadigital.net/actualidad/articulos->

informativos/item/7669-lenguajes-de-programacion-que-son-y-para-que-sirven.html

LINKOGRAFIA:

<https://es.wikipedia.org/wiki/NetBeans>

<https://programacionparaJava.blogspot.pe/2014/09/agregar-eliminar-modificar-y-listar-en.html>

<https://netbeans.apache.org/download/index.html>

