

INSTITUTO TECNOLÓGICO DE OAXACA

INGENIERÍA EN SISTEMAS
COMPUTACIONALES

TÓPICOS AVANZADOS DE
PROGRAMACIÓN

SU

LIC. ADELINA CRUZ NIETO

UNIDAD III

INVESTIGACIÓN I

ANAYA BAUTISTA GUADALUPE
BERENICE

14/07/18

A) Los componentes visuales y no visuales de NetBeans.

¿Que es un componente?

Un componente es un elemento genérico con una funcionalidad muy concreta, cuya única finalidad es la reutilización. Cada uno de ellos está destinado a realizar una tarea típica en una aplicación.

¿Que es un componente visual de NetBeans?

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (ventanas, botones, cuadros de texto, barras de scroll etc.)

¿Que es un componente no visual de NetBeans?

Un componente no visual es aquel (que aunque no se pueda visualizar directamente) se encuentra creado e implementado dentro de nuestro JFrame y es necesario para tener una buena distribución de nuestros componentes visuales.

¿como creamos e implementamos un componente en NetBeans?

Para crear un componente en NetBeans nos podemos auxiliar de la paleta de componentes que ya viene preestablecida en nuestro IDE, esta paleta es de gran ayuda ya que sólo con arrastrar el componente a nuestro JFrame o a nuestro JPanel ya queda fija en nuestra interfaz, para personalizarlo simplemente se debe de seleccionar nuestro componentes y darle click izquierdo y propiedades y ahí es donde podemos hacerle los ajustes que nosotros deseemos(cambiarle forma,cambiarle color,cambiarle texto,cambiarle tamaño etc.) .

Otra manera de crearlo es a través de código y al crearlo así es necesario conocer todos los métodos que tiene nuestro componente(se ejemplifican varios):

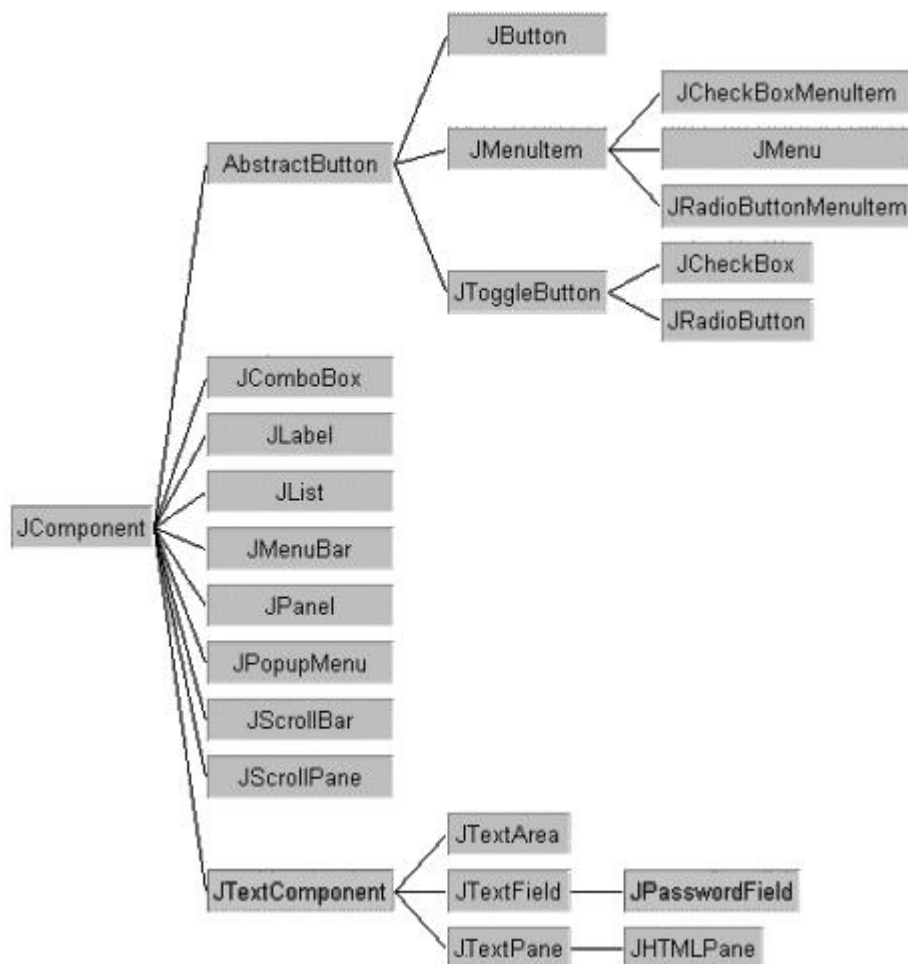
```
<tipo de componente> nombre de la variable;
```

```
Nombre de la variable = new <tipo de componente>();
```

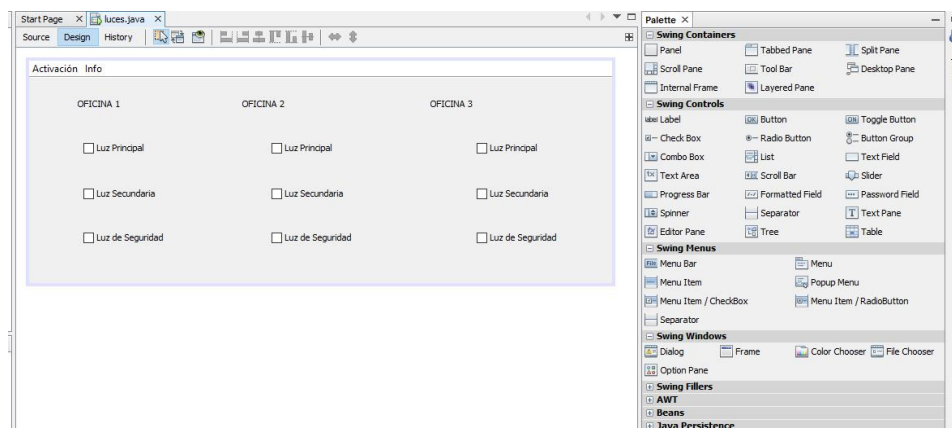
```
Add();
```

setLayout(<tipodeLayout>, en su defecto null);
setBounds(x,y,ancho,largo);
setVisible(boolean);
setResizable(boolean);.

Ejemplos de componentes:



Vista de Netbeans: (Diseño creado y la paleta default.)



Vista de código creado automáticamente por NetBeans al usar la paleta:

```
oficina3 = new javax.swing.JMenuItem();
Todas = new javax.swing.JMenuItem();
MenuNoche = new javax.swing.JMenu();
ofi1 = new javax.swing.JMenuItem();
ofi2 = new javax.swing.JMenuItem();
ofi3 = new javax.swing.JMenuItem();
todo = new javax.swing.JMenuItem();
menuSalir = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
Principales = new javax.swing.JMenuItem();
Secundarias = new javax.swing.JMenuItem();
Seguridad = new javax.swing.JMenuItem();
acerca = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowClosing(java.awt.event.WindowEvent evt) {
        formWindowClosing(evt);
    }
});

jPanel1.setToolTipText("");
jPanel1.setName(""); // NOI18N

checkLuzPrincipal.setText("Luz Principal");
checkLuzPrincipal.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        checkLuzPrincipalActionPerformed(evt);
    }
});
```

Vista de código creado por el usuario para el funcionamiento del diseño previo:

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

private void acercaActionPerformed(java.awt.event.ActionEvent evt) {
    JOptionPane.showMessageDialog(null, "CONTROL DE LUCES \n ");
}

private void oficina2ActionPerformed(java.awt.event.ActionEvent evt) {
    checkLuzPrincipal.setSelected(true);
    checkLuzSecundaria.setSelected(true);
    checkLuzdeSeguridad.setSelected(false);
}

private void checkLuzPrincipalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void menuApagartodoActionPerformed(java.awt.event.ActionEvent evt) {
    checkLuzPrincipal.setSelected(false);
    checkLuzSecundaria.setSelected(false);
    checkLuzdeSeguridad.setSelected(false);
    checkLuzPrincipal1.setSelected(false);
    checkLuzSecundaria1.setSelected(false);
    checkLuzdeSeguridad1.setSelected(false);
    checkLuzPrincipal2.setSelected(false);
    checkLuzSecundaria2.setSelected(false);
}
```

B) Definición conceptual de componentes paquetes y librerías.

Componente

Un componente es el elemento básico de una interfaz gráfica. Los componentes permiten al usuario interactuar con la aplicación y proporcionar información desde el programa al usuario sobre el estado de la aplicación.

Cabe mencionar que los componentes nunca se encuentran de forma aislada, sino agrupados dentro de contenedores. Los contenedores contienen y organizan la situación de los componentes; además, son en sí mismos componentes y como tales pueden ser situados dentro de otros contenedores.

Librerías

Paquetes

Los paquetes Java agrupan las clases y/o interfaces en librerías (bibliotecas).

En los paquetes las clases son únicas, comparadas con las de otros paquetes, y permiten controlar el acceso. Esto es, los paquetes proporcionan una forma de ocultar clases, evitando que otros programas o paquetes accedan a clases que son de uso exclusivo de una aplicación determinada.

La sentencia `import` se utiliza para incluir una lista de paquetes en los que buscar una clase determinada, y su sintaxis es:

```
import nombre_paquete.Nombre_Clase;
```

Esta sentencia, o grupo de ellas, deben aparecer antes de cualquier declaración de estas.

C) Uso de librerías proporcionadas por el lenguaje (Java).

Java es un lenguaje de programación desarrollado para una multitud de plataformas y procesadores.

Librerías/paquetes más utilizados en java:

java.lang

Contiene clases esenciales para el lenguaje java y es el único paquete se importa automáticamente.

Colección de tipo básico siempre importados a cualquier unidad de compilación. Aquí están las declaraciones de objetos, clases, wrappers.

java.io

Archivos de stream y acceso aleatorio. Librería estándar de entrada y salida.

Interfaces Clases

DataInputBufferedInputStream

DataOutputBufferedOutputStream

ExternalizableBufferedReader

FilefilterBufferedwrite

Soporta flujos de entrada y salida java.

FilenameFilterByteArrayInputStream ObjectInputByteArrayOutputStream

SerializableDataOutputStream File InputStream reader Writer,etc.

java.net

Soporta facilidades de red (URL, sockets TCP, sockets UDP, direcciones IP, conversiones binarias a texto).

java.util

Clase de diccionarios, tabla de hash, stack, técnica de codificación hora, fecha, etc.

Interfaces Clases

Collection AdstractCollection

Comparator AdstracList

Enumeration AdstrectMap

EventListenerAdstrectSecquentialList

IteratorAbstractSet
List ArrayList
Observer Collection
SortedSetEventObject
Random Stack
Timer
Vector
Date,etc.

java.awt

La librería java.awt proporciona un Abstract Window Toolkit para programación GUI (Interfaz gráfica de usuario), dibujo de gráficos e imágenes, así como también eventos colores, tipo de letras, botones, campos de texto, etc.

Es la librería visual más antigua de java usando esta librería, se podrán construir los tres tipos de programas más comunes como son FRAME, WINDOW y APPLET.

javax.swing

Es la librería de componentes visuales más nueva que proporciona java, un conjunto de componentes que se ejecutan uniformemente en todas las plataformas.

Interfaces Clases
Action AbstractAction
ComboBoxEditorActionMap
Icon Box.Filler
ListModelCellRendererPane
MenuElementDebugGraphics
WindowsConstantsDefaultListSelectionModel
JApplet
Jbutton
JCheckBox
JFrameJMenu
JLabel
JPanel

TextField

Tree

Windows

Timer

java.sql

Junto con el paquete javax.sql, permite la conexión a bases de datos, el envío de sentencias SQL y la interpretación de los resultados de las consultas.

InterfacesClases

Array Date

Blob DriverManager

CallableStatementDriverPropertyInfo

ClobSQLPermission

Connection Timer

DatabaseMetaData Timestamp

Driver Type

Ref

SQLData

SQLInput

SQLOutput

Struct

java.sound.midi

Paquete con clases e interfaces que permiten la captura, procesamiento y reproducción de música.

java.math

Proporciona cálculos matemáticos.

java.applet

El paquete java.applet permite la creación de applets a través de la clase Applet, proporciona interfaces para conectar un applet a un documento Web y para audición de audio.

BIBLIOGRAFIA:

<https://docs.oracle.com/javase/7/docs/api/java/awt/Container.html>

<https://docs.oracle.com/javase/7/docs/api/javax/swing/JComponent.html>

<http://nereida.deioc.ull.es/~cleon/pa/0304/prct/p4/node1.html>

<http://itpn.mx/recursosisc/4semestre/topicosavanzados/Unidad%20III.pdf>

<http://www.tesoem.edu.mx/alumnos/cuadernillos/2013.001.pdf>