

**Programación Avanzada I**

**Guía de Estudio**

**Tarea**

**Nombre del Alumno**

Julio Cesar Amador Oviedo No. De Cuenta 201120060090

**Hora:** 10:00 a.m. – 1:00 a.m.

**Ingeniero: Adelman Eliud Varela**

**Fecha de Entrega**

Viernes, 14 de febrero del 2020.

1. R/ Es un Entorno de desarrollo integrado o en inglés Integrated development Environment (IDE)
2. R/ Es un entorno de desarrollo integrado libre hecho principalmente para el lenguaje de programación Java.
3. R/ Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en java
4. R/ Java Enterprise Edition (JEE) anteriormente conocido como java 2 es una plataforma de programación java para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación java
5. R/ **Netbeans: es un entorno de desarrollo Java**. Es libre y su uso principal es el desarrollo de aplicaciones Java, aunque también permite el desarrollo de aplicaciones en otros lenguajes de programación tales php, html. Además de ser un producto gratuito y libre sin restricciones de uso

**Eclipse:** Se trata de una aplicación de programación multiplataforma que permite el desarrollo de aplicaciones para Android que también utilizan el lenguaje Java para su implementación.

Existen otros entornos de desarrollo Java (JBuilder, JCreator) pero dado que se trata de software comercial tienen una menor difusión que los aquí tratados.

**Scratch:** Es una aplicación gráfica en 2 dimensiones en la cual los niños de entre 6 y 11 años pueden desarrollar aplicaciones mediante el uso del ratón sin necesidad de uso de líneas de código permitiendo un acercamiento a las estructuras y conceptos más básicos de programación.

**Alice:** Es el paso natural desde Scratch a las 3 dimensiones para los adolescentes de entre 11 y 14 años que pueden desarrollar aplicaciones mediante el uso del ratón y mediante la introducción de las primeras líneas de código permitiendo un acercamiento al lenguaje Java.

**Greenfoot:** entornos de desarrollo JavaEs una aplicación gráfica en 2 dimensiones en la cual la programación se realiza mediante la introducción de código Java a un nivel básico y permite el aprendizaje de los conceptos fundamentales de Java tales como herencia, abstracción, polimorfismo a un mayor nivel que lo introducido en otros entornos de desarrollo Java tales Scratch o Alice. Está enfocado a adolescentes de entre 12 y 16 años.

**BlueJ:** se trata de un entorno de desarrollo previo a los entornos de desarrollo Java de carácter profesional, donde el adolescente se sumerge en el manejo del lenguaje Java y adquiere habilidades para la implementación de aplicaciones Java. Está recomendado para edades a partir de 16 años.

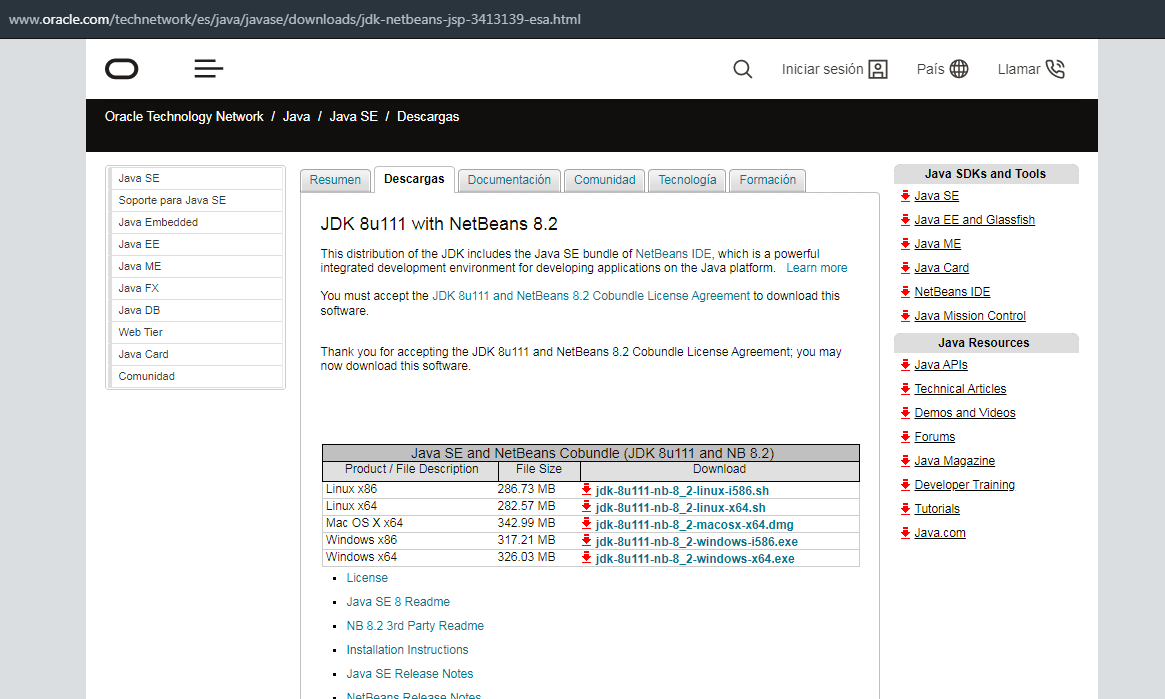
Los entornos de desarrollo Java Greenfoot y BlueJ han sido desarrollados por la Universidad de Kent en colaboración con Oracle.

**AndroidStudio:** Se trata de un software para el desarrollo profesional de aplicaciones para Android, en el cual es necesario un conocimiento previo del lenguaje Java y xml.

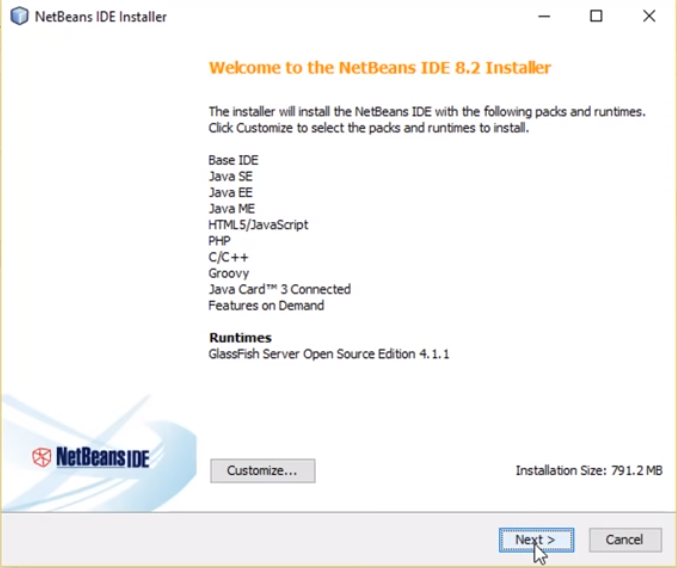
Está enfocado a jóvenes a partir de los 18 años y profesionales del desarrollo de aplicaciones informáticas.

1. R/ Primero hay que descargar el programa de instalación de Netbeans al igual que el kit de desarrollo JDK ambos son encontrados en la siguiente dirección

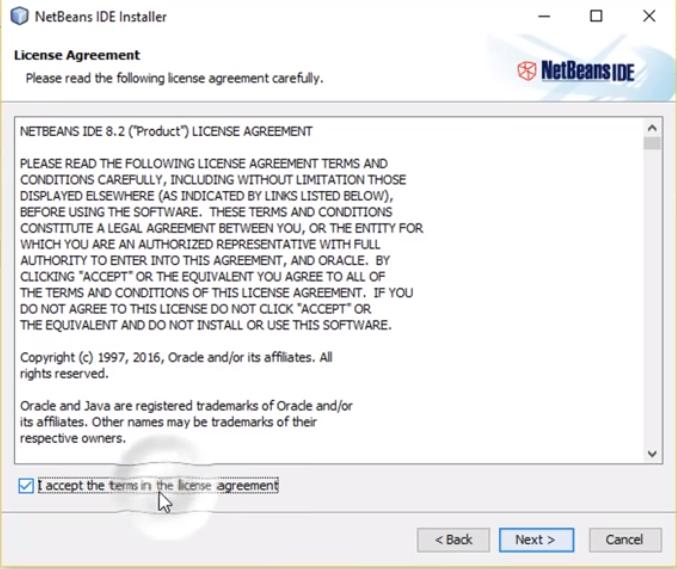
<https://www.oracle.com/technetwork/es/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html> antes de descargar la aplicación se debe revisar para que sistema operativo debe ir soportado, esta información la veremos en la opción de Mi PC o Mi Equipo propiedades a este. Y aquí podemos ver el tipo de sistema que tenemos instalado una vez revisado esto procedemos a descargar la aplicación acorde al sistema operativo como se ve en pantalla ya sea 32 o 64 bits.



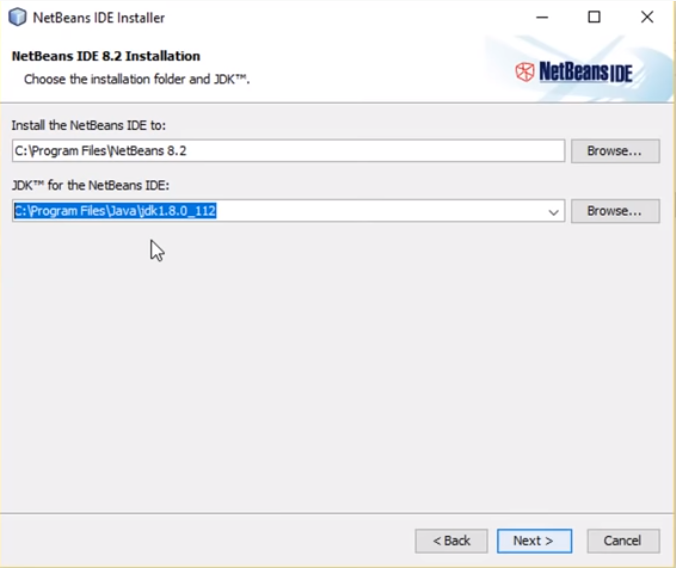
Una vez descargado ejecutamos el paquete de instalación y procedemos a la instalación de este



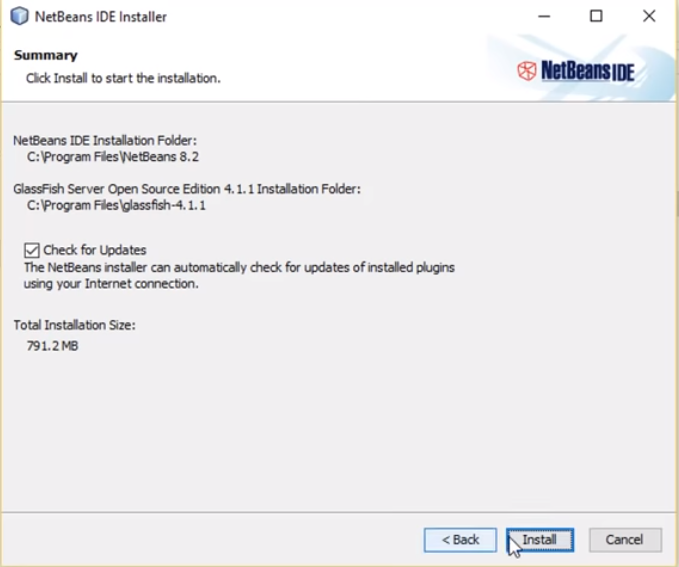
Aparecerá otra ventana a la cual le damos aceptar los términos.



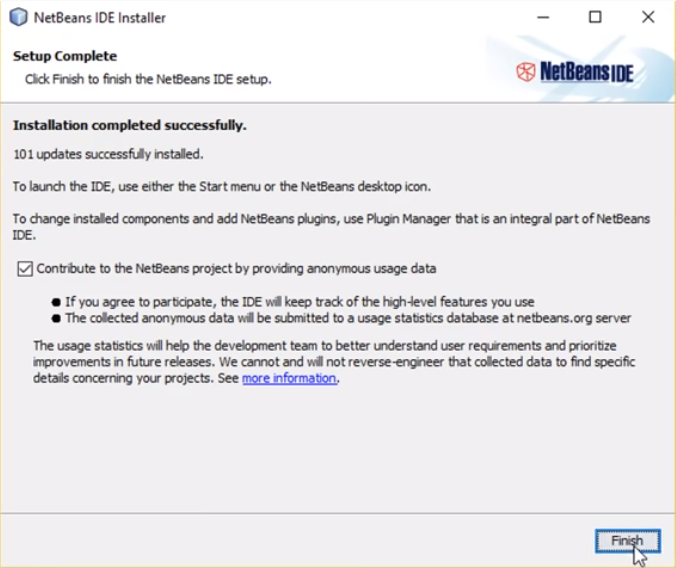
Luego pedirá las rutas donde será instalado el programa por default lo dejamos como aparece en pantalla y damos siguiente

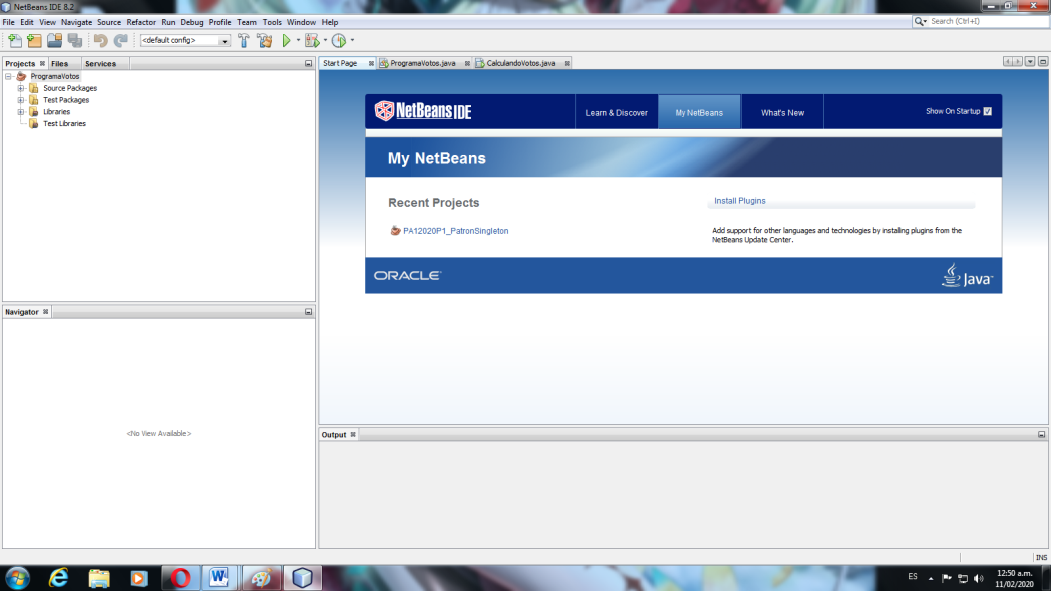


Luego de todo esto procedemos a dar instalación al programa de Netbeans JDK



Luego de haber terminado la instalación procedemos a dar finalizar a la ventana y ya podremos ejecutar el programa Netbeans

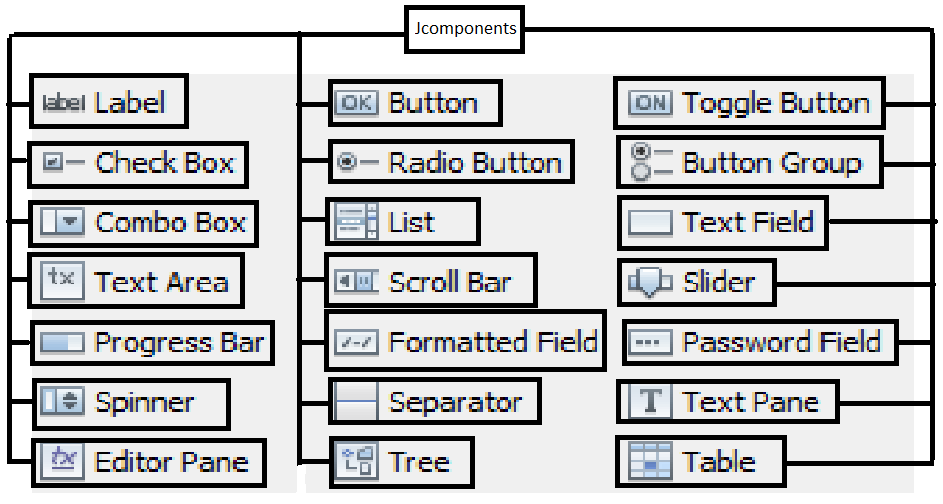


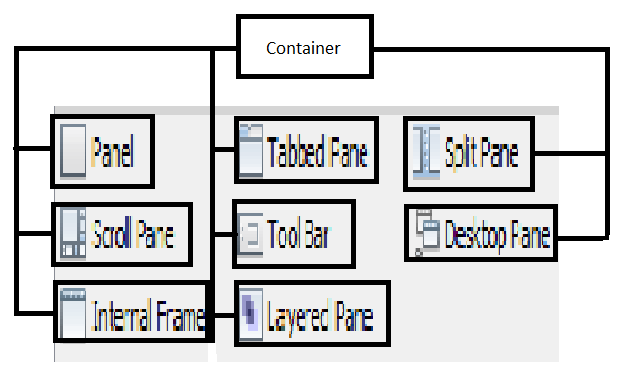


1. R/ Es una Máquina virtual o en inglés Java Virtual Machine,( JVM) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java
2. R/ Los paquetes en Java (packages) son la forma en la que Java nos permite agrupar de alguna manera lógica los componentes de nuestra aplicación que estén relacionados entre sí.

Los paquetes permiten poner en su interior casi cualquier cosa como: clases, interfaces, archivos de texto, entre otros. De este modo, los paquetes en Java ayudan a darle una buena organización a la aplicación ya que permiten modularizar o categorizar las diferentes estructuras que componen nuestro software.

1. R/ Es un archivo que contiene código byte, código binario que puede ser ejecutado utilizando la Máquina Virtual Java.
2. R/ Los archivos con la extensión .jar son archivos Zip pero con otra extensión es decir son archivos comprimidos los cuales pueden ser abiertos con winzip o winrar o cualquier otra herramienta que soporte archivos ZIP.
3. R/ Los archivos .class contienen código byte, código binario mientras que el archivo .jar son archivos de tipo ZIP pero con otra extensión es decir archivos comprimidos por lo general ambos siempre van de la mano
4. R/ Un lenguaje de programación es fuertemente tipificado cuando no se permiten violaciones de los tipos de datos, es decir, dado el valor de una variable de un tipo concreto, no se puede usar como si fuera de otro tipo distinto a menos que se haga una conversión
5. R/ Es llamada Interfaz Gráfica o en inglés Graphical User Interface (GUI) al conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación. Es decir ventanas, botones, combos, listas, cajas de diálogo, campos de texto, etc.
6. R/ Swing es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, listas desplegables y tablas.
7. R/ Es un Kit de herramientas de ventana abstracta o en inglés Abstract Window Toolkit (AWT) es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.
8. R/ Es llamada Interfaz Gráfica o en inglés Graphical User Interface (GUI) al conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación. Es decir ventanas, botones, combos, listas, cajas de diálogo, campos de texto, etc.
9. Component es una clase abstracta que representa todo lo que tiene una posición, un tamaño, puede ser pintado en pantalla y puede recibir eventos
10. R/ La clase Conteiner es una clase abstracta derivada de Component, que representa a cualquier componente que pueda contener otros componentes.
11. R/ JComponent es la clase abstracta de la cual derivan casi todos los componentes de swing de peso ligero, es decir es la clase de la cual la mayoría de los componentes swing derivan o heredan.
12. R/ **Un Panel** en java son objetos contenedores y la finalidad de este es la agrupación de de otros objetos como botones, campos de texto etc. **Un Dialogo** no es más que una ventana que nos permite mostrar en él mensajes por ejemplo de error de advertencia o información entre el usuario y el sistema informático. **Un marco** es un software utilizado por los desarrolladores de software para implementar la estructura estándar de una aplicación.
13. R/ Los hijos de la clase jcomponent son los botones cajas de texto etiquetas o casillas de verificación.
14. R/ La clase conteiner consta de tres tipos de contenedores de alto nivel **Jframe:** Se visualiza como una ventana principal con marco y barra de título. **Jdialog:** Se visualiza como una ventana independiente de la ventana principal para mostrar información, como por ejemplo el contenido de un directorio. **Japplet:** Permite crear aplicaciones con interface gráfica que se ejecutan en el contexto de un navegador web.
15. R/ La clase **Windows** Representa el nivel más básico de una ventana, sin bordes ni menús. Una instancia de una ventana no puede estar contenida dentro de otro contenedor. Y de esta descienden **Dialog** (para implementar diálogos) y **Frame** que es una ventana más compleja ya que tiene borde y menú
16. R/





1. R/ El método **"parse"**, entre sus múltiples funciones, nos permite convertir caracteres númericos a datos numéricos, es decir, convertir un número almacenado como String a un dato del tipo int, double u otro según se requiera.
2. R/ Declaramos primero una variable (String ejem = Null;) esto debe ir inicializado ya que al hacer la conversión pedirá inicializar la variable, (int ejemplo=Integer.parseInt(ejem);) la variable en este punto ha sido convertida a entero ahora con el ejemplo en decimales utilizando la misma variable (double ejemplo=Integer.parseInt(ejem);)
3. R/ un ejemplo simple sería crear un objeto asignando un dato y luego este valor pasarlo (convertirlo a String) y luego mostar el valor del objeto

Object o = "hola";

String str = objectToString(o);

System.out.println("objeto a string=" + str);

1. R/ Para hacer el cambio podemos realizarlo con una asignación para ello podemos hacer lo siguiente:

Object o = new tipobjeto();

1. R/ Cuando el usuario de un programa o applet mueve el raton, o hace un clic o usa el teclado, genera un Evento. En Java los eventos, como cualquier otra cosa, se representan como instancias u objetos de alguna clase. Para programar es necesario aprender a utilizar los eventos en una interfaz grafica.
2. R/ ActionEvent, ItemEvent, KeyEvent, MouseEvent, AdjustmentEvent
3. R/ **ActionEvent**: El usuario hace un clic sobre el boton.

**ItemEvent**: El usuario selecciona o deselecciona el interruptor (Checkbox)

**KeyEvent**: El usuario pulsa o suelta una tecla

**MouseEvent**: El usuario pulsa o suelta un botón del ratón, el cursor del ratón entra o sale o el usuario mueve o arrastra el ratón

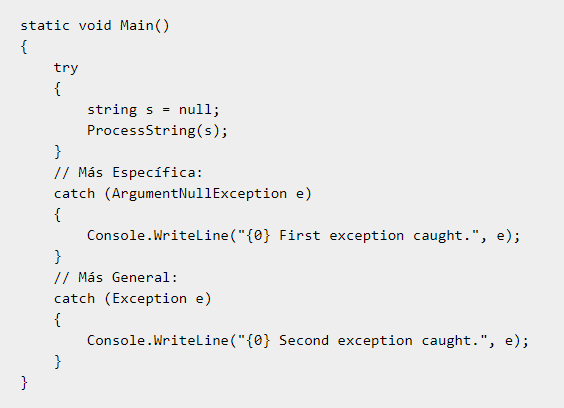
**AdjustmentEvent**: El usuario mueve la barra de desplazamiento

**ComponentEvent:** El componente se mueve, cambia de tamaño, se esconde o se exhibe

**FocusEvent:** El componente gana o pierde el foco

1. R/ **Try** se utiliza cuando se desean prever excepciones en el transcurso de la ejecución de un programa. Es necesario ya que ahí se colocan las instrucciones que se desean realizar pese a la excepción, posteriormente se usa un **catch** donde se especifica la excepción que se sabe puede ocurrir.
2. R/ El try se compone del evento a realizar este puede ser de diferentes índoles ya sea capturando datos, trasformando variables o cualquier otra aplicación donde la idea es que trate de efectuar la sentencia en él detallada y si esta está correcta proceda de lo contrario

Nos indica a través de un catch el problema o error ejemplo:



1. R/ **Errores tipo int**: donde los datos ingresados no son números

**Tipo Double:** donde los datos ingresados no coinciden con los aceptados en el campo

**Tipo Char:** donde los datos ingresados no son caracteres o estos están mesclados

**Por condición:** Donde se evalua atraves de un un evento if, for o while y este está fuera de la condición estipulada

**Por captura de Datos:** donde es mandatario que sea llenado un dato y este es ignorado o saltado por el usuario

1. R/ Consiste en eventos que ejecutan una tarea cuando una determinada acción se lleva a cabo por el usuario. Esta acción puede ser cualquier cosa que el usuario pueda hacer, como mover el ratón o pulsar una tecla en el teclado. Inicia tu programa de desarrollo de Java
2. R/ El objeto excepción contiene información sobre un determinado error. Lo que permite al programador controlar los errores ocasionados durante la ejecución de un programa informático
3. R/ El elemento Joptiongroup representa un grupo de opciones (option) temáticamente definido, que forma parte de un control. Las opciones agrupadas pueden ser etiquetadas mediante el atributo label, que usarán los navegadores para mostrar como nombre del grupo.
4. R/ Model es una colección de objetos o clases por las cuales un programa puede examinar y manipular algunas partes específicas de su mundo. En otras palabras, la interfaz orientada a objetos a algún servicio o sistema. Para un **combobox** se llena creando y llamando un método a este de la siguiente forma

Public void datosconbox () {

(Nombre de la caja de texto).addItem(“El dato que se quiere”);

}

==🡺Luego desde la aplicación de Java se llama el método

datosconbox ();

Y al ejecutar el programa aparecerán los datos en el combo box ingresados anteriormente en el método para la **lista** se procede a crear un constructor para que este adiera información de una en una

DefaultListModel modelolista;

Luego desde la aplicación de java inicializo la lista

modelolista= new DefaultListMode();

“nombre de la lista”.sermodel(modelolista);

Una vez realizado esto la información ingresada ya sea por un campo de texto o cualquier medio a través de un botón se capturara esa información y será almacenada en la lista de la siguiente forma

String valor = “Nombre del campo de texto”.getText();

Modelolista.addElement(valor);

Una ves realizado esto desde el formulario al ingresar un valor x y presionar el botón creado aparecerá en nuestra lista la información ingresada

1. R/ **StackTrace** permite saber la clase, el método e incluso la línea de código donde se produce la excepción. Y no sólo eso, sino que además muestra los métodos anteriores por los que ha pasado el flujo del programa antes de producirse la excepción.
2. R/ Los cuadros de dialogo (JOptionPane) son “herramientas” muy útiles al momento de ingresar datos y mostrar información; ya que con estas aparecen mensajes de dialogo o cuadros de dialogo para poder interactuar con el usuario
3. R/ **NULL** significa que una variable no tiene valor y apunta a ninguna parte en la memoria. **empty() (Vacio)** es más un significado literal de vacío , por ejemplo, la cadena "" está vacía, pero no es NULL.
4. R/ **Los patrones de diseño** son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.
5. R/ Es un patrón de diseño utilizado dentro de la programación orientada a objetos que, como su nombre lo sugiere, del inglés “single” que significa “único”, trata de diseñar clases que solo puedan ser instadas una vez, es decir, que de esa clase, durante toda la ejecución del programa, únicamente podrá ser creado un objeto.
6. R/ **Que es Static en Java** Una clase, método o campo declarado como estático puede ser accedido o invocado sin la necesidad de tener que instanciar un objeto de la clase. Uno de los ejemplos típicos de uso de métodos y variables estáticas es la clase java.lang.Math: