INSITUTO UNIVERSITARIO DE TECNOLOGIA

PARA LA INFORMATICA

IUTEPI

MATERIA: PROGRAMACIÓN II

eclipse

ESCUELA: ANÁLISIS DE SISTEMAS

ÚLTIMA ACTUALIZACIÓN: SEPTIEMBRE 2019

ELABORADO POR: TSU. MINERVA TABORDA

Prefacio

El presente documento forma parte del programa de estudios de la materia Programación VI ofrecido a sus estudiantes en el Instituto Universitario de Tecnología para la Informática – IUTEPI. El objetivo fundamental de este tutorial es ayudar al estudiante,através de 20 ejercicios, a resolver complicaciones de distinta índole (matemáticos,administrativos, gráficos, contables etc.) empleando como herramienta un lenguaje de programación. Así mismo, sirve de apoyo complementario bajo la modalidad de auto-aprendizaje publicado en su campus virtual, a todos los alumnos que cursan la materia.

UNIDAD 1 JAVA

Breve Historia: ***Java*** surgió en 1991 cuando un grupo de ingenieros de *Sun Microsystems* trataron de diseñar unnuevo lenguaje de programación destinado a electrodomésticos. Desarrollaron un código "neutro" que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una "*máquina hipotética o virtual*" denominada ***Java Virtual Machine*** (***JVM***).
Como lenguaje de programación para computadores, ***Java*** se introdujo a finales de 1995. Laclave fue la incorporación de un intérprete ***Java*** en la versión 2.0 del programa Netscape Navegador,produciendo una verdadera revolución en Internet. ***Java 1.1*** apareció a principios de 1997,mejorando sustancialmente la primera versión del lenguaje. ***Java 1.2***, más tarde rebautizado como***Java 2***, nació a finales de 1998.
La compañía ***Sun*** describe el lenguaje ***Java*** como "*simple, orientado a objetos, distribuido,interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea ydinámico*".
Que es **JAVA?**: Es en un entorno informático para trabajar y entretenerse. Con Java podrá jugar a juegos en línea, charlar con personas de todo el mundo, calcular los intereses de su hipoteca y ver imágenes en 3D, entre muchas otras cosas.
La compañía ***Sun***, creadora de ***Java***, distribuye gratuitamente el *Java(tm) Development Kit* (***JDK***). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en ***Java.***
Existe también una versión reducida del***JDK***, denominada ***JRE*** (*Java Runtime Environment*) destinada únicamente a ejecutar código ***Java***(no permite compilar).

El Entorno de Desarrollo de JAVA:

Los ***IDE***s (*Integrated Development Environment*), tal y como su nombre indica, son entornosde desarrollo integrados. En un mismo programa es posible escribir el código ***Java***, compilarlo yejecutarlo sin tener que cambiar de aplicación. Algunos incluyen una herramienta para realizar***Debug*** gráficamente, frente a la versión que incorpora el ***JDK*** basada en la utilización de unaconsola (denominada habitualmente ventana de comandos de MS-DOS de Windows y el Block de Notas.
https://www.youtube.com/watch?v=L1oMLsiMusQ; https://www.youtube.com/watch?v=vJTeIJx_Kn0.

Eclipse: Es un entorno de desarrollo integrado de código abierto multi-plataforma que nos permite entre varias cosas más desarrollar aplicaciones en JAVA. Está también abierto a la posibilidad de instalarle plugins para aumentar sus características y llevar a cabo tareas más especificas y complejas. Eclipse es también altamente usado por programadores y es de hecho una herramienta que deberías saber usar en cualquier momento de tu vida profesional.

JCreator: es un entorno de desarrollo que posee una versión profesional y otra versión libre, sin embargo la versión libre es bastante sencilla, desde el punto de vista del aprendizaje entre menos ayudas tengas mas aprenderás por lo cual la versión libre sería la ideal, sin embargo creo que esta versión es demasiado simple y en realidad se trata de aprender con rapidez y facilidad de manera autónoma, sin embargo no estaría mal tenerla como parte de nuestro repertorio.

El JDK 8(última versión JDK 9)(JAVA Development Kit) es el conjunto de librerías, paquetes, clases, métodos, etc. que posee JAVA, de este modo, es importante tenerlo instalado para poder desarrollar aplicaciones en Java, no pretendo profundizar mucho en este aspecto, pues consiste en instalar la aplicación y debería ser todo, sin embargo a veces se hace necesario configurar las variables de entorno (según el IDE seleccionado).

Introducción a la OOP (programación orientada a objetos)

Junto con el paradigma de la **orientación a procedimientos**, son las dos filosofías generales de diseño más importantes. A diferencia de la orientación a procedimientos (OP), la **orientación a objetos** (OO) no concibe

los procesos como una secuencia de procedimientos con su entrada y salida sino que se basa en un conjunto de objetos interactuando.

http://www.javaya.com.ar/detalleconcepto.php?codigo=74&inicio=0

UNIDAD 2 Conceptos Básicos:

Veamos a continuación los aspectos más destacados de esta filosofía general de diseño.

¿Qué son las clases en Java?Una *clase* es una agrupación de *datos* (variables o campos) y de *funciones* (métodos) que operan sobre esos datos. A estos datos y funciones pertenecientes a una clase se les denomina *variables* y *métodos* o *funciones de miembro*. La programación orientada a objeto se basa en la programación de clase. Un programa se construye a partir de un conjunto de clases.

Tipos de clases:

abstract : Una clase abstract tiene al menos un método abstracto. Una clase abstracta no se instancia, sino que se utiliza como clase base para la herencia.

final : Una clase final se declara como la clase que termina una cadena de herencia. No se puede heredar de una clase *final*. Por ejemplo, la clase *Math* es una clase *final.*

Public: Las clases public son accesibles desde otras clases, bien sea directamente o por herencia. Son accesibles dentro del mismo paquete en el que se han declarado. Para acceder desde otros paquetes, primero tienen que ser importadas.

Synchronizable: Este modificador especifica que todos los métodos definidos en la clase son sincronizados, es decir, que no se puede acceder al mismo tiempo.

Control de acceso
Cuando se crea una nueva clase en Java, se puede especificar el nivel de acceso que se quiere para las variables de instancia y los métodos definidos en la clase:

Public:public void CualquieraPuedeAcceder(){}
Cualquier clase desde cualquier lugar puede acceder a las variables y métodos de instacia públicos.
protected : protected void SoloSubClases(){}
Sólo las subclases de la clase y nadie más puede acceder a las variables y métodos de instancia protegidos.
Prívate: private String NumeroDelCarnetDeIdentidad;
Las variables y métodos de instancia privados sólo pueden ser accedidos desde dentro de la clase. No son accesibles desde las subclases.
friendly (sin declaración específica)

¿Qué son los objetos en JAVA?
Objetos: Un ***objeto*** (en inglés, ***instance***) es un ejemplar concreto de una clase. Las ***clases*** son como tipos de variables, mientras que los ***objetos*** son como variables concretas de un tipo determinado.

Classname unObjeto;
Classname otroObjeto;

¿Qué son los mensajes en JAVA?

Para poder crear una aplicación Java que sea un tanto robusta, es muy probable que necesitemos más de un objeto (más de una instancia de una clase), y probablemente algunos de estos objetos no deben estar aislados unos de otros, pues bien, para comunicarse, esos objetos se envían mensajes entre sí.

Los mensajes son simples llamadas a las funciones o métodos del objeto en particular con el cual se quiere comunicar para solicitarle que ejecute alguna "accion" según sus métodos y/o atributos.

¿Qué es la herencia en Java?

¿Qué significa esto de la herencia?, ¿quién hereda qué?, esto sólo significa que en Java puedes crear una clase partiendo de otra que ya exista.Es decir, puedes crear una clase a través de una clase existente, y ésta clase tendrá todas las variables y los métodos de su "superclase", y además se le podrán añadir otras variables y métodos propios.Se llama "Superclase" a aquella clase de la cual desciende una clase, más adelante veremos un poco más de esto al detalle.

Uso de Comentarios:

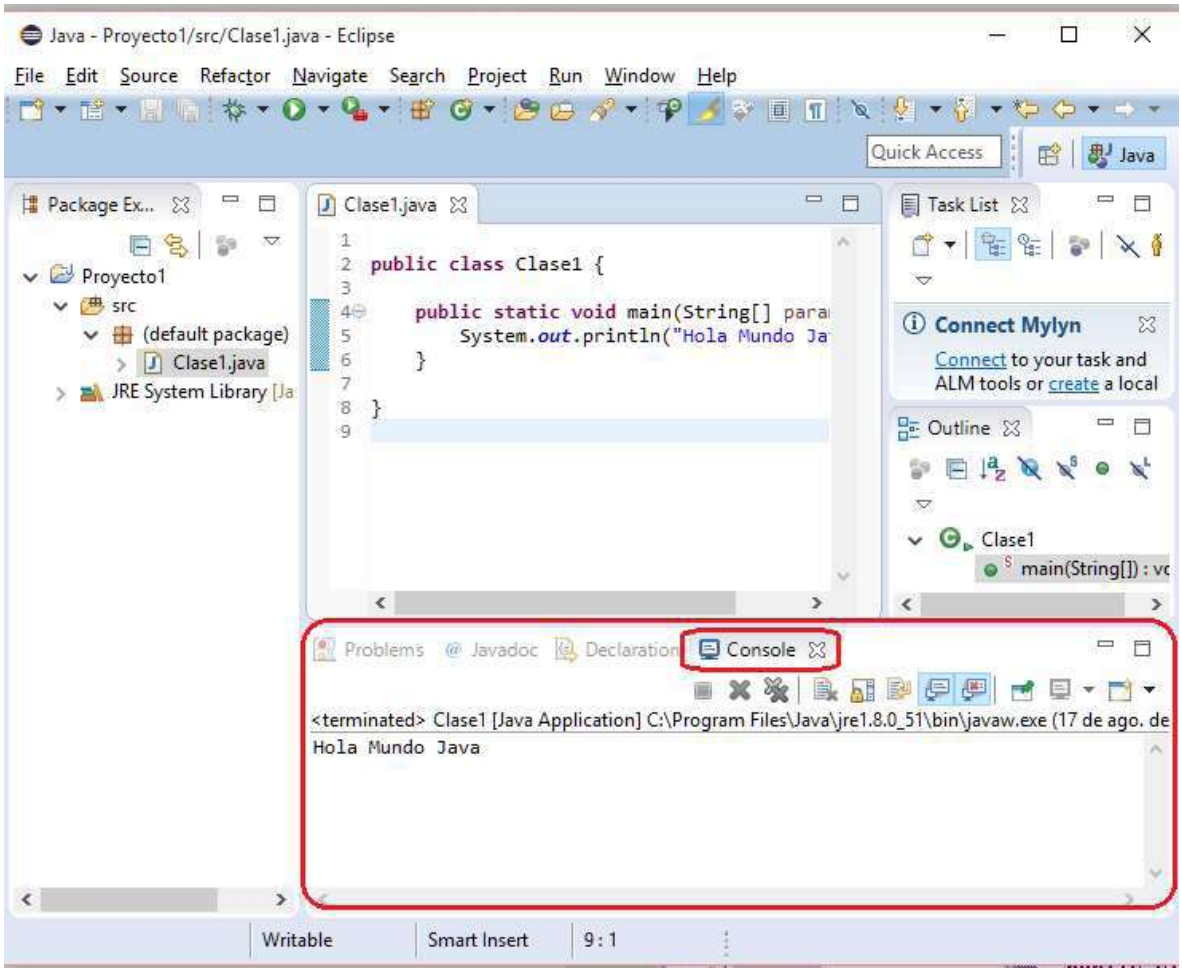
En Java hay tres tipos de comentarios:

```
// comentarios para una sola línea

/* comentarios de una o
más líneas
*/

/** comentario de documentación, de una o más líneas
*/
```

https://www.programarya.com/Cursos/Java/Java-Ejemplos-y-Conceptos

UNIDAD 3Pasos para crear un programa en Eclipse

- Todo programa en Eclipse requiere la creación de un "Proyecto", para esto debemos seleccionar desde elMenú de opciones
- Como segundo paso veremos que todo programa en Java requiere como mínimo una clase. Para crear una clase debemos seleccionar desde el menú de opciones: En el diálogo que aparece debemos definir el nombre de la clase (en nuestro primer ejemplo la llamaremos Clase1 (con mayúscula la letra C).
- Todo programa en Java debe definir la función main. Esta función la debemos codificar dentro de la clase: "Clase1"
- Procedemos a copiar lo siguiente: System.out.println(“Hola mundo Java”);
- Como último paso debemos compilar y ejecutar el programa, esto lo podemos hacer la barra de herramientas: Run, y en la parte inferior podemos ver el mensaje en la consola.



UNIDAD 4Tipos de Variable en JAVA:

Una Variables es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo.

Tipos de Variables: Una variable puede almacenar:
- Valores Enteros (100, 260, etc.)
- Valores Reales (1.24, 2.90, 5.00, etc.)
- Cadenas de caracteres ("Minerva", "Compras", "Listado", etc.).

Operadores en JAVA:
Java es un lenguaje rico en operadores, que son casi idénticos a los de C/C++. Se explicaran brevemente:

Operadores Aritmetios: Son operadores binarios (requieren siempre dos operandos) que realizan las operaciones aritméticas habituales: ***suma*** (+), ***resta*** (-), ***multiplicación*** (*), ***división*** (/) y ***resto de la división.***

Operadores de asignación:

Operador	Utilización	Expresión equivalente
+=	op1 += op2	op1 = op1 + op2
-=	op1 -= op2	op1 = op1 - op2
*=	op1 *= op2	op1 = op1 * op2
/=	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2

%=	op1 %= op2	op1 = op1 % op2

Operadores incrementales
Java dispone del operador ***incremento*** (++) y ***decremento*** (--). El operador (++).

Operador condicional ?:
Este operador, tomado de C/C++, permite realizar bifurcaciones condicionales sencillas. Su formageneral es la siguiente: booleanExpression ? res1 : res2

Operadores relacionales
Sirven para realizar comparaciones de igualdad, desigualdad y relación de menor o mayor.

Operador	**Utilizació**	**El resultado es tru2**
>	op1 > op2	si op1 es mayor que op2
>=	op1 >= op2	si op1 es mayor o igual que op2
<	op1 < op2	si op1 es menor que op2
<=	op1 <= op2	si op1 es menor o igual que op2
==	op1 == op2	si op1 y op2 son iguales
!=	op1 != op2	si op1 y op2 son diferente

Operadores lógicos
Los operadores lógicos se utilizan para construir ***expresiones lógicas***, combinando valores lógicos (***true*** y/o ***false***) o los resultados de los operadores ***relacionales***.

Operador	**Nombre**	**Utilización**	**Resultado**
&&	AND	op1 && op2	true si op1 y op2 son true. Si op1 es false ya no se evalúa op2
||	OR	op1 || op2	true si op1 u op2 son true. Si op1 es true ya no se evalúa op2
!	negación	! op	true si op es false y false si op es true
&	AND	op1 & op2	true si op1 y op2 son true. Siempre se evalúa op2
|	OR	op1 | op2	true si op1 u op2 son true. Siempre se evalúa op2

A medida que vamos avanzando veremos por medio de ejercicios (programas)la utilización de estos operadores en JAVA.

Ejercicio 2Carga un programa en JAVA que permita la suma de 2 números enteros y muestre por pantalla el resultado

```
Public Class Suma}
    Public static void main (String[] args)}
int num1=7;
        int num2=5;
        int resultado=0;
resultado= (num1 + num2),
System.out.print("La suma de los dos números  es:"+ (suma);
            }
}
```

El resultado por consola es (La suma de los números es: 12).

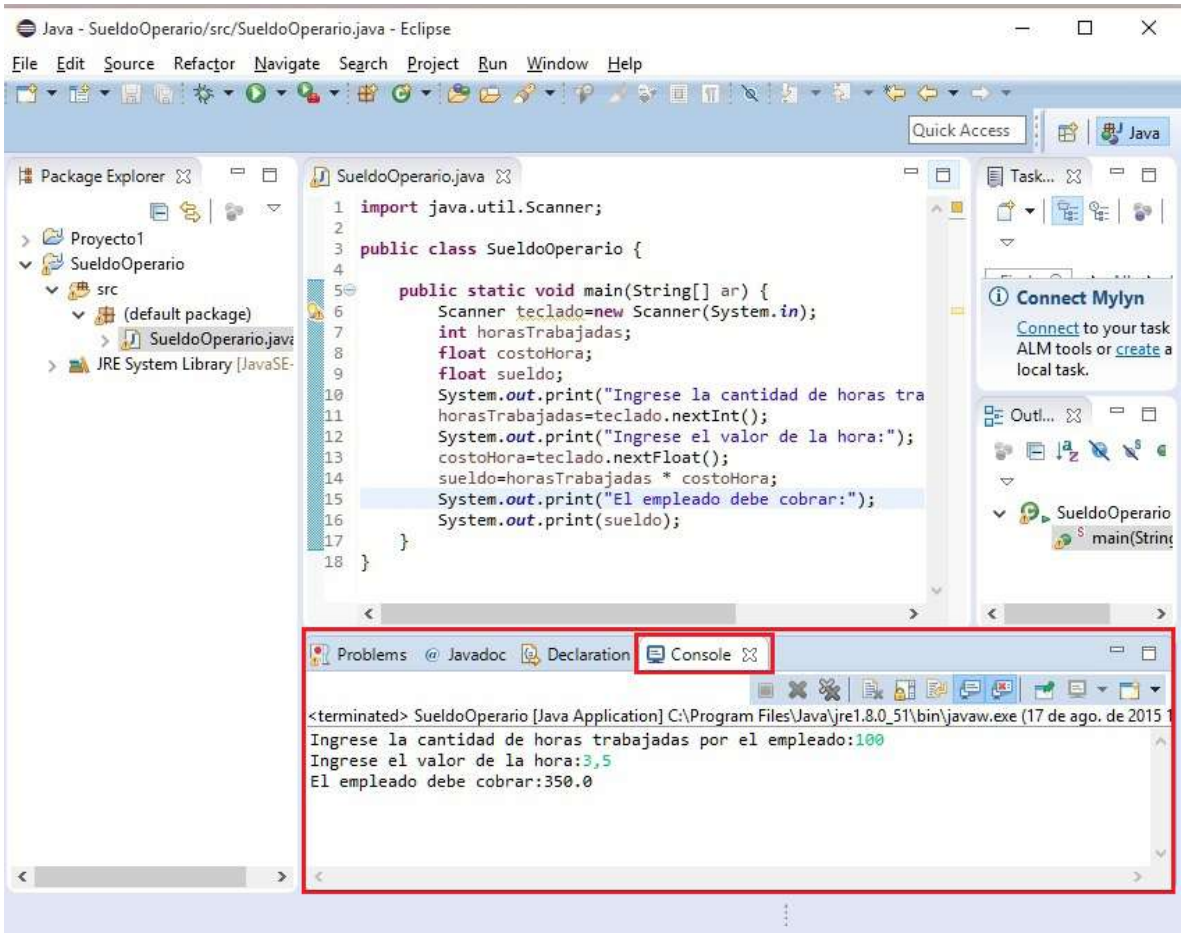
Comandos de estrada y salida de Datos: permite al usuario la entrada de datos por teclado.

Sintaxis de la instrucción Scanner:

- import java.util.ScannerImportamos la librería:
- Creamos la clase (nombre de la clase en Mayúscula al comienzo
- Se define la function main
- Scanner teclado=new Scanner(System.in);
- Se inicializan las variables (;)
- System.out.printl("ingrese valor por favor:")
- `horasTrabajadas=teclado.nextInt();` lee el valor por teclado

BufferedReader : ejemplo

Ejercicio: 3 Carga un programa en JAVA que **permita**Calcular el sueldo mensual de un Empleado: la cantidad de horas trabajadas y el pago por hora las introduce el usuario.**Datos conocidos:**Horas trabajadas en el mes.Pago por hora.**Proceso:**Cálculo del sueldo multiplicando la cantidad de horas por el pago por hora.
Información resultante: sueldo mensual.



http://www.javaya.com.ar/detalleconcepto.php?codigo=77&inicio=0

UNIDAD5Estructuras de Programación.

Estructura de programación **secuencial**;

Cuando en un problema sólo participan operaciones, entradas y salidas se ladenomina una estructura secuencial. Ejemplo el ejercicio de la suma de dos números.

Estructuras Condicionales: Cuando hay que tomaruna decisión aparecen las estructurascondicionales.La palabra clave "if" indica que estamos en presencia de una estructura condicional;seguidamente disponemos la condición entre paréntesis.

Estructura condicional simple: Cuando se presenta la elección tenemos la opción de realizar una actividad o no realizar ninguna.Ejemplo **Ejercicio 4**

```
import java.util.Scanner;
public class EstructuraCondicionalSimple1 {
public static void main(String[] ar) {
Scanner teclado=newScanner(System.in);
float sueldo;
System.out.print("Ingrese el sueldo:");
sueldo=teclado.nextFloat();
                if (sueldo>3000) {
System.out.println("Esta persona debe abonarimpuestos");
}
}
```

Estructura condicional Compuesta: **Ejercicio 5**

```
import java.util.Scanner;
public class EstructuraCondicionalCompuesta1 {
public static void main(String[] ar) {
Scanner teclado=new Scanner(System.in);
int num1,num2;
System.out.print("Ingrese primer valor:");
Num1=teclado.nextInt();
System.out.print("Ingrese segundo valor:");
num2=teclado.nextInt();
if (num1>num2) {
System.out.print(num1);
} else {
System.out.print(num2);
}
}
}
```

Problemas Propuestos para el alumno: Realizar un programa que lea por teclado dos números, si el primero es mayor al segundo: mostrar su suma y diferencia, en caso contrario mostrar el producto y ladivisión del primero respecto al segundo.

Estructuras condicionales anidadas Decimos que una estructuracondicional es anidada cuando por larama del verdadero o el falso de unaestructura condicional hay otraestructura condicional. Ejemplo:

Ejercicio 6: Carga un programa que pida por teclado tres notas de un alumno, calcule elpromedio e imprima alguno de estos mensajes:Si el promedio es >=7 mostrar "Promocionado".Si el promedio es >=4 y <7 mostrar "Regular".Si el promedio es <4 mostrar "Reprobado".

Ejercicio: 6

```
public class EstructuraCondicionalAnidada1 {
  public static void main(String[] ar) {
     Scanner teclado=new Scanner(System.in);
int nota1,nota2,nota3;
     System.out.print("Ingrese primer nota:");
     nota1=teclado.nextInt();
     System.out.print("Ingrese segunda nota:");
     nota2=teclado.nextInt();
     System.out.print("Ingrese tercer nota:");
     nota3=teclado.nextInt();
     int promedio=(nota1 + nota2 + nota3) / 3;
if (promedio>=7) {
        System.out.print("Promocionado");
     } else {
        if (promedio>=4) {
          System.out.print("Regular");
        } else {
          System.out.print("Reprobado");
}
}
  }
```

UNIDAD 6Bucles o Ciclos

Hasta ahora hemos empleado estructuras SECUENCIALES y CONDICIONALES. Existe otro tipo de estructuras tan importantes como las anteriores que son las estructuras REPETITIVAS. Lo cual, permite ejecutar una instrucción o un conjunto de instrucciones varias veces. Se denomina también **lazo** o **loop**. El código incluido entre las **laves** {}.

Ciclo FOR en Java

Los ciclos FOR (o ciclos para) son una estructuras de control cíclica, nos permiten ejecutar una o varias líneas de código de forma iterativa (o repetitiva), pero teniendo cierto control y conocimiento sobre las iteraciones. En el ciclo FOR, es necesario tener un valor inicial y un valor final, y opcionalmente podemos hacer uso del tamaño del "paso" entre cada "giro" o iteración del ciclo.Ejemplo ejercicio 7Desarrollar un

programa que permita la carga de 10 valores por teclado y nos muestreposteriormente la suma de los valores ingresados y su promedio. Ver Ejercicio 7

Ejercicio: 7 For

```
import java.util.Scanner;
public class EstructuraRepetitivaFor2 {
   public static void main(String[] ar) {
      Scanner teclado=new Scanner(System.in);
int suma,f,valor,promedio;
suma=0;
      for(f=1;f<=10;f++) {
System.out.print("Ingrese valor:");
         valor=teclado.nextInt();
         suma=suma+valor;
      }
      System.out.print("La suma es:");
      System.out.println(suma);
      promedio=suma/10;
      System.out.print("El promedio es:");
      System.out.print(promedio);
   }
```

Ciclo While en Java: Podemos observar que se ingresa por teclado la variable n. El operador puede cargar cualquier valor.Si el operador carga 10 el bloque repetitivo se ejecutará 10 veces, ya que la condiciónes ?Mientras x<=n ?, es decir ?mientras x sea menor o igual a 10?; pues x comienza enuno y se incrementa en uno cada vez que se ejecuta el bloque repetitivo.A la prueba del diagrama la podemos realizar dándole valores a las variables; porejemplo ejercicio 8, si ingresamos 5 el seguimiento es el siguiente:

n x
1 (Se imprime el contenido de x)
2 " "
3 " "
4 " "
5 " "
6 (Sale del while porque 6 no es menor o igual a 5)

Ejercicio 8 While

```
import java.util.Scanner;
public class EstructuraRepetitivaWhile2 {
   public static void main(String[] ar) {
      Scanner teclado=new Scanner(System.in);
int n,x;
      System.out.print("Ingrese el valor final:");
n=teclado.nextInt();
      x=1;
      while (x<=n) {
         System.out.print(x);
         System.out.print(" - ");
x = x + 1;
}
   }
}
```

Ciclo Do While:es otra estructurarepetitiva, la cual ejecuta al menos unavez su bloque repetitivo, a diferenciadel while o del for que podían noejecutar el bloque.Esta estructura repetitiva se utilizacuando conocemos de antemano quepor lo menos una vez se ejecutará elbloque repetitivo. Ejemplo ejercicio 9.Realizar un programa que permita ingresar el peso (en kilogramos) de piezas. Elproceso termina cuando ingresamos el valor 0. Se debe informara) Cuántas piezas tienen un peso entre 9.8 Kg. y 10.2 Kg. cuántas con más de 10.2Kg. y cuántas con menos de 9.8 Kg.b) La cantidad total de piezas procesadas.

Ejercicio 9 DO

```
import java.util.Scanner;
public class EstructuraRepetitivaDoWhile3 {
   public static void main(String[] ar) {
      Scanner teclado=new Scanner(System.in);
      int cant1,cant2,cant3,suma;
      float peso;
      cant1=0;
      cant2=0;
      cant3=0;
do {
         System.out.print("Ingrese el peso de la pieza
(0 pera finalizar):");
peso=teclado.nextFloat();
         if (peso>10.2) {
            cant1++;

} else {
            if (peso>=9.8) {
               cant2++;
```

Continuación : Ejercicio 9 DO

```
 } else {
               if (peso>0) {
cant3++;
               }
            }
         }
      } while(peso!=0);
      suma=cant1+cant2+cant3;
      System.out.print("Piezas aptas:");
      System.out.println(cant2);
      System.out.print("Piezas con un peso superior a
10.2:");
      System.out.println(cant1);
      System.out.print("Piezas con un peso inferior a 9.8:");
System.out.println(cant3);
   }
}
```

UNIDAD 7 Vectores Y Matrices

Un vector es una estructura de datosque permite almacenar un conjunto de datos del MISMO tipo.Con un único nombre se define unvector y por medio de un subíndicehacemos referencia a cada elementodel mismo (componente)ver Ejercicio 10.Se desea guardar los sueldos de 5 operarios.Según lo conocido deberíamos definir 5 variables si queremos tener en un ciertomomento los 5 sueldos almacenados en memoria.Empleando un vector solo se requiere definir un único nombre y accedemos a cadaelemento por medio del subíndice.

Ejercicio 10 Vectores

```
import java.util.Scanner;
public class PruebaVector1 {
   private Scanner teclado;
   private int[] sueldos;
   public void cargar()
   {
      teclado=new Scanner(System.in);
      sueldos=new int[5];
      for(int f=0;f<5;f++) {
System.out.print("Ingrese valor de la componente:");
         sueldos[f]=teclado.nextInt();
      }
   }
public void imprimir() {
for(int f=0;f<5;f++) {
        System.out.println(sueldos[f]);
     }
   }
   public static void main(String[] ar) {
PruebaVector1 pv=new PruebaVector1();
      pv.cargar();
      pv.imprimir();
   }
```

Matrices en java. Una matriz es una estructura de datosque permite almacenar un conjunto de datos del MISMO tipo.Con un único nombre se define lamatriz y por medio de DOS subíndiceshacemos referencia a cada elementode la misma (componente). Una matriz se la puede representar por un conjunto de vectores. Ejercicio 11. Crear una matriz de 3 filas por 5 columnas con elementos de tipo Integer, cargar suscomponentes y luego imprimirlas.

Problemas propuestos 1-Crear una matriz de 2 filas y 5 columnas. Realizar la carga de componentes porcolumna (es decir primero ingresar toda la primer columna, luego la segundacolumna y así sucesivamente.

2-Crear y cargar una matriz de 3 filas por 4 columnas. Imprimir la primera fila. Imprimir laúltima fila e imprimir la primera columna.

Ejercicio 11 Matrices

```
import java.util.Scanner;
public class Matriz1 {
   private Scanner teclado;
   private int[][] mat;
      public void cargar() {
      teclado=new Scanner(System.in);
      mat=new int[3][5];
      for(int f=0;f<3;f++) {
         for(int c=0;c<5;c++) {
            System.out.print("Ingrese
componente:");
            mat[f][c]=teclado.nextInt();
         }
      }
   }
```

ContinuaciónEjercicio 11 Matrices

```
 public void imprimir() {
for(int f=0;f<3;f++) {
         for(int c=0;c<5;c++) {
            System.out.print(mat[f][c]+" ");
         }
         System.out.println();
      }
   }
      public static void main(String[] ar) {
Matriz1 ma=new Matriz1();
      ma.cargar();
      ma.imprimir();
   }
}
```

UNIDAD 8 QUÉ ES EL AWT

El AWT (*Abstract Windows Toolkit*) es la parte de ***Java*** que se ocupa de construir interfacesgráficas de usuario o**Interfaces Visuales**. También implementaremos el **COMANDO SWING** y las cómo se trabaja en JAVA con etiquetas.Confeccionar el programa "Hola Mundo" utilizando una interfaz gráfica de usuario Ejercicio 12.

Cuando ejecutamos nuestro proyecto tenemos como resultado una ventana similar a esta:

Ejercicio 12

La clase JFrame encapsula el concepto de una ventana,e implementa unaaplicación que muestre una ventana debemos plantear una clase que herede de la**clase JFrame:public class Formulario extends JFrame{**
Definimos luego como atributo de la clase un objeto de la clase JLabel:**private JLabel label1**
En el constructor de la clase llamamos al método heredado de la clase JFrame llamado**setLayout** y le pasamos como parámetro un valor null, con esto estamos informándole ala clase JFrame que utilizaremos posicionamiento absoluto para los controles visuales dentro del JFrame.

```
Ejercicio 12
import javax.swing.*;
public class Formulario extends JFrame{
private JLabel label1;
public Formulario() {
setLayout(null);
label1=new JLabel("Hola Mundo.");
label1.setBounds(10,20,200,30);
add(label1);
}
public static void main(String[] ar) {
Formulario formulario1=new Formulario();
formulario1.setBounds(10,10,400,300);
formulario1.setVisible(true);
}
}
```

public Formulario() {
setLayout(null);
Luego tenemos que crear el objeto de la clase JLabel y pasarle como parámetro alconstructor el texto a mostrar
label1=new JLabel("Hola Mundo.");
Ubicamos al objeto de la clase JLabel llamando al método setBounds, este requie como parámetros la columna, fila, ancho y alto del JLabel. Finalmente llamamos almétodo add (metodo heredado de la clase JFrame) que tiene como objetivo añadir elcontrol JLabel al control JFrame.
label1=new JLabel("Hola Mundo.");
label1.setBounds(10,20,200,30);
add(label1);
}
Finalmente debemos codificar la main donde creamos un objeto de la clase Formulario,llamamos al método setBounds para ubicar y dar tamaño al control y mediante elmétodo **setVisible** hacemos visible el JFrame:**public static void main(String[] ar) {**
Formulario formulario1=new Formulario();
formulario1.setBounds(10,10,400,300);
formulario1.setVisible(true);
Con este ejercicio se trabajo en JAVA con ventanas, clase,atributos, método, Herencia, y otros procedimientos para finalmente ver una interfaz visible que permite mostrar una etiqueta HOLA Mundo.(Ejercicio13).

https://javaparajavatos.wordpress.com/category/ejercicios-de-herencia
Ejercicio 13 JButton
Botones en JAVA

El tercer control visual de uso muycomún es el que provee la claseJButton.Confeccionar una ventana que muestre un botón. Cuando se presione finalizar laejecución del programa Java. Ejercicio 13. La mecánica para atrapar el clic del objeto de la clase JButton se hace mediante laimplementación de una interface. Por ejemplo la interface:**ActionListener** {
public void **actionPerformed(ActionEvent e)** {
}
Definimos un objeto de la clase JButton:
JButton boton1

```
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame
implements ActionListener JButton boton1;
public Formulario() {
setLayout(null);
boton1=new JButton("Finalizar");
boton1.setBounds(300,250,100,30);
add(boton1);
boton1.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
if (e.getSource()==boton1) {
System.exit(0);
}
}
```

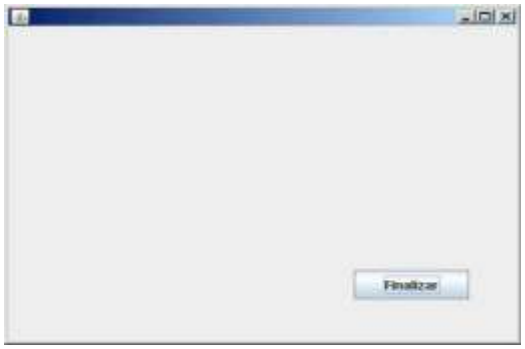
Luego mediante la llamada delmétodo addActionListener le pasamos la referencia del objeto de la clase JButtonutilizando la palabra clave (this).El this almacena la dirección de memoria donde sealmacena el objeto de la clase JFrame.

```
boton1.setBounds(300,250,100,30);
        add(boton1);
        boton1.addActionListener(this);
```

Finalmente en el método actionPerformed(se encarga de ejecutará cada vez que hagamos clic sobre el objeto) mediante el acceso al método getSource() del objetoque llega comoparámetro podemos analizar que botón se presionó.

```
public void actionPerformed(ActionEvent e) {
if (e.getSource()==boton1) {
                System.exit(0);
```

A continuación se muestra en eclipse la ventana (**Ejercicio 14**)



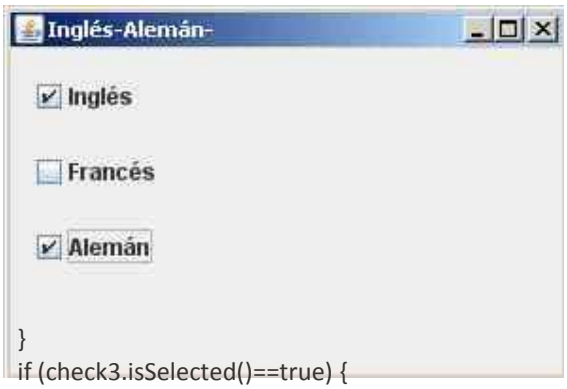
Problema propuesto: Confeccionar una ventana que contenga tres objetos de la clase JButton con lasetiquetas "1", "2" y "3". Al presionarse cambiar el título del JFrame indicando cuál botónse presionó

http://files.uladech.edu.pe/docente/18010218/PROGRAMACION_VISUAL_I/SESION_02/Tema_2_Objetos_JLabel_JTextField_JButton.pdf

JCheckBox:

El control JCheckBox permiteimplementar un cuadro de selección(básicamente un botón de dos estados) . **Ejercicio 15**Confeccionar un programa que muestre3 objetos de la clase JCheckBox conetiquetas de tres idiomas. Cuando se loselecciona mostrar en el título delJFrame todos los JCheckBoxseleccionados hasta el momento.

```
import javax.swing.*;
import javax.swing.event.*;
public class Formulario extends JFrame implements ChangeListener{
private JCheckBox check1,check2,check3;
public Formulario() {
setLayout(null);
check1=new JCheckBox("Inglés");
check1.setBounds(10,10,150,30);
check1.addChangeListener(this);
add(check1);
check2=new JCheckBox("Francés");
check2.setBounds(10,50,150,30);
check2.addChangeListener(this);
add(check2);
```




```
}
if (check3.isSelected()==true) {
cad=cad+"Alemán-";
}
setTitle(cad);
}
public static void main(String[] ar) {
```

```
check3=new JCheckBox("Alemán");
check3.setBounds(10,90,150,30);
check3.addChangeListener(this);
add(check3);
}
public void stateChanged(ChangeEvent e){
String cad="";
if (check1.isSelected()==true) {
cad=cad+"Inglés-";
}
if (check2.isSelected()==true) {
cad=cad+"Francés-";
```

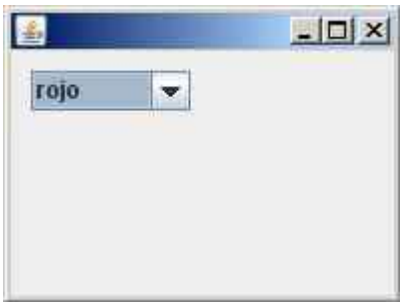
El control JComboBox permiteseleccionar un String de una lista.Para inicializar los String que contendráel JComboBox debemos llamar almétodo addItem tantas veces comoelementos queremos cargar.Un evento muy útil con este control escuando el operador selecciona un Itemde la lista. Para capturar la selecciónde un item debemos implementar lainterface ItemListener que contiene unmétodo llamada itemStateChanged.

JComboBox (lista)

El control JComboBox permiteseleccionar un String de una lista.Para inicializar los String que contendráel JComboBox debemos llamar almétodo addItem tantas veces comoelementos queremos cargar.Un evento muy útil con este control escuando el operador selecciona un Itemde la lista. Para capturar la selecciónde un item debemos implementar lainterface ItemListener que contiene unmétodo llamada temStateChanged.
Ejercicio 17

Cargar en un JComboBox los nombres de varios colores. Al seleccionar alguno mostraren la barra de título del JFrame el String seleccionado.

```
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ItemListener{
private JComboBox combo1;
public Formulario() {Ejercicio 16
setLayout(null);
combo1=new JComboBox();
combo1.setBounds(10,10,80,20);
add(combo1);
combo1.addItem("rojo");
combo1.addItem("vede");
combo1.addItem("azul");
combo1.addItem("amarillo");
combo1.addItem("negro");
combo1.addItemListener(this);
}
public void itemStateChanged(ItemEvent e) {
if (e.getSource()==combo1) {
String seleccionado=(String)combo1.getSelectedItem();
setTitle(seleccionado);
}
}
public static void main(String[] ar) {
Formulario formulario1=new Formulario();
formulario1.setBounds(0,0,200,150);
formulario1.setVisible(true);
```



```
}
}
```

JTextField:

El control permite al operador del programa ingresar una cadena de caracteres por teclado.**Ejercicio 17**Confeccionar un programa que permita ingresar el nombre de usuario y cuando se presione un botón mostrar el valor ingresado en la barra de títulos del JFrame.

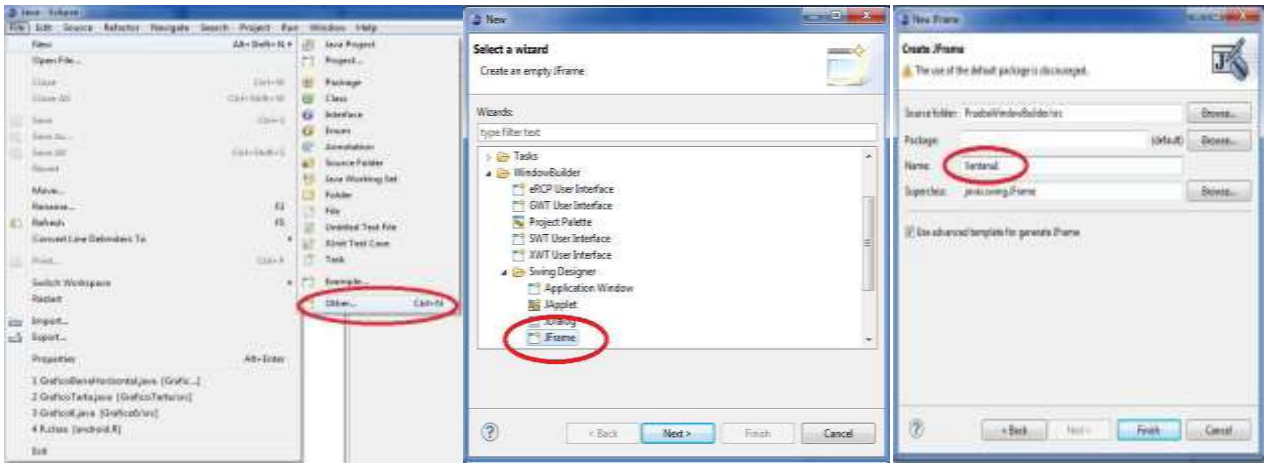
```
import javax.swing.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener{
private JTextField textfield1;
private JLabel label1;
private JButton boton1;
public Formulario() {
setLayout(null);
label1=new JLabel("Usuario:");
label1.setBounds(10,10,100,30);
add(label1);
textfield1=new JTextField();
textfield1.setBounds(120,10,150,20);
add(textfield1);
boton1=new JButton("Aceptar");
boton1.setBounds(10,80,100,30);
add(boton1);
boton1.addActionListener(this);
}
public void actionPerformed(ActionEvent e) {
if (e.getSource()==boton1) {
String cad=textfield1.getText();
setTitle(cad);
}
}
public static void main(String[] ar) {
Formulario formulario1=new Formulario();
formulario1.setBounds(0,0,300,150);
formulario1.setVisible(true);
}
}
```

Ventana ejercicio 17

Problema Propuesto: Confeccionar un programa que permita ingresar dos números en controles de tipo JTextField, luego sumar los dos valores ingresados y mostrar la suma en la barra del título del control JFrame.

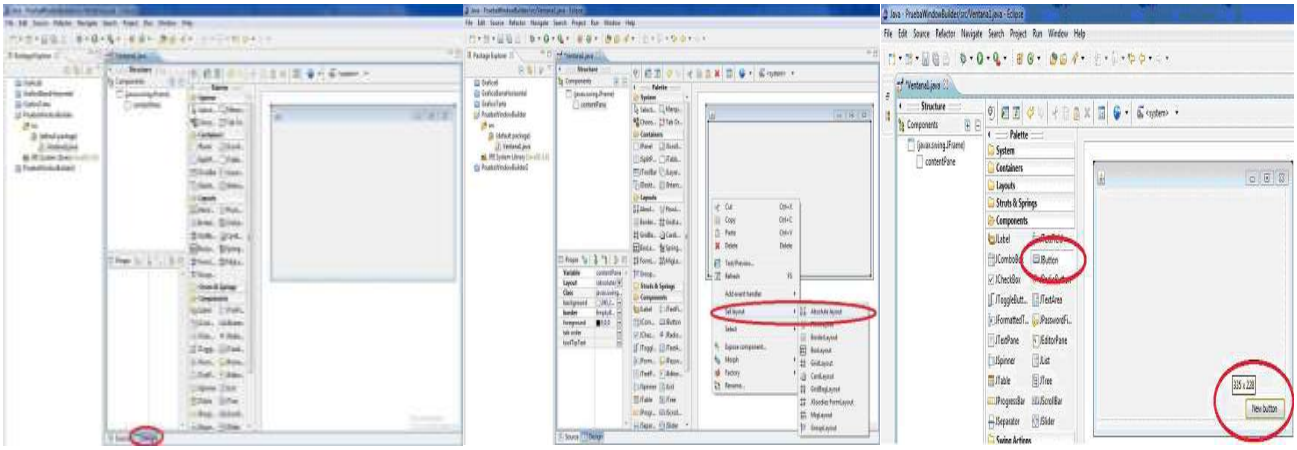
UNIDAD. 9Plug-in WindowBuilder para crear interfaces visuales.

El objetivo de este concepto es conocer el empleo del plug-in WindowBuilder para el desarrollo de interfaces visuales arrastrando componentes. A medida que se arrastra componentes visuales sobre un formulario se genera en forma automática el código Java, esto nos permite ser más productivos en el desarrollo de la interfaz de nuestra aplicación y nos ayuda a concentrarnos en la lógica de nuestro problema:
Paso 1 -Crea un nuevo proyecto.
Paso 2 - Seleccionamos el nombre de nuestro proyecto (lo llamaremos PruebaWindowBuilder):
Paso 3- Ahora seleccionamos la opción del menú File -> New -> Other
Paso 4 - Seleccionamos la opción la opción JFrame:
Paso 5 - Seguidamente presionamos el botón Next > y definimos el nombre de la clase a crear (Ventana1):



6 - Configuramos el Layout de JFrame presionando el botón derecho del mouse sobre el formulario generado yseleccionamos la opción SetLayout > Absolute layout (esto nos permite luego disponer controles visuales comoJButton, JLabel etc. en posiciones fijas dentro del JFrame):

7 - De la ventana Palette seleccionamos con el mouse un objeto de la clase JButton (presionamos con el mousedicha componente, deberá aparecer seleccionada) y luego nos desplazamos con el mouse sobre el JFrame ypresionamos el botón del mouse nuevamente (en este momento aparece el botón dentro del JFrame):



UNIDAD. 10 Menú en JAVA

Swing - JMenuBar, JMenu, JMenuItem: Cuando necesitamos implementar unmenú horizontal en la parte superior deun JFrame requerimos de un objeto dela clase JMenuBar, uno o más objetosde la clase JMenu y por último objetosde la clase JMenuItem.Par la captura de eventos debemosimplementar la interface ActionListenery asociarlo a los controles de tipoJMenuItem, el mismo se dispara al presionar con el mouse el JMenuItem.

Ejercicio 18:Confeccionaremos un menú de opciones que contenga tres opciones que permitacambiar el color de fondo del JFrame a los colores: rojo, verde y azul.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Formulario extends JFrame implements ActionListener{
private JMenuBar mb;
private JMenu menu1;
private JMenuItem mi1,mi2,mi3;
public Formulario() {
setLayout(null);
mb=new JMenuBar();
setJMenuBar(mb);
menu1=new JMenu("Opciones");
mb.add(menu1);
mi1=new JMenuItem("Rojo");
mi1.addActionListener(this);
menu1.add(mi1);
mi2=new JMenuItem("Verde");
mi2.addActionListener(this);
menu1.add(mi2);
mi3=new JMenuItem("Azul");
mi3.addActionListener(this);
menu1.add(mi3);
}
public void actionPerformed(ActionEvent e) {
Container f=this.getContentPane();
if (e.getSource()==mi1) {
f.setBackground(new Color(255,0,0));
}
if (e.getSource()==mi2) {
f.setBackground(new Color(0,255,0));
}
if (e.getSource()==mi3) {
f.setBackground(new Color(0,0,255));
}
}
public static void main(String[] ar) {
Formulario formulario1=new Formulario();
formulario1.setBounds(10,20,300,200);
formulario1.setVisible(true);
}
}
```

Ventana **Ejercicio 18**

http://www.javaya.com.ar/detalleconcepto.php?codigo=133&inicio=40

UNIDAD12 IDE Eclipse. Trabajar con el WINDOW BUILDERAPPLETS:

Definición:es una mini-aplicación, escrita en *Java*, que se ejecuta en un browser (*Netscape Navigator*, m*icrosoft Internet Explorer*, …) al cargar una página HTML que incluye información sobre el *applet* a ejecutar por medio de las *tags* <APPLET>... </APPLET>.

Las *applets* no tienen ventana propia: se ejecutan en la ventana del browser (en un "*panel*"). Por la propia naturaleza "abierta" de Internet, las *applets* tienen importantes restricciones de seguridad

Algunas características de las applets: Las características de las *applets* se pueden considerar desde el puntode vista del programador y desde el del usuario.

1. No tienen un método *main()* con el que comience la ejecución.
2. Todas las *applets* derivan de la clase *java.applet.Applet*.
3. Se heredan otros muchos métodos de las super-clases de *Applet* que tienen que ver con la generación de interfaces gráficas de usuario (AWT).
4. Las *applets* disponen de métodos relacionados con la obtención de información, como por ejemplo: *getAppletInfo()*, *getAppletContext()*, *getParameterInfo()*, *getParameter()*, *getCodeBase()*, *getDocumentBase()*, e *isActive()*.

CÓMO INCLUIR UN APPLET EN UNA PÁGINA HTML

- Escribir el programa fuente en cualquier editor y guardarlo con extensión .java

- Compilar el fichero fuente mediante: javac miProgramaApplet.java
- Escribir la pagina web que contendrá al applet y guardar el código con extensión .html
- El códigomínimo será:

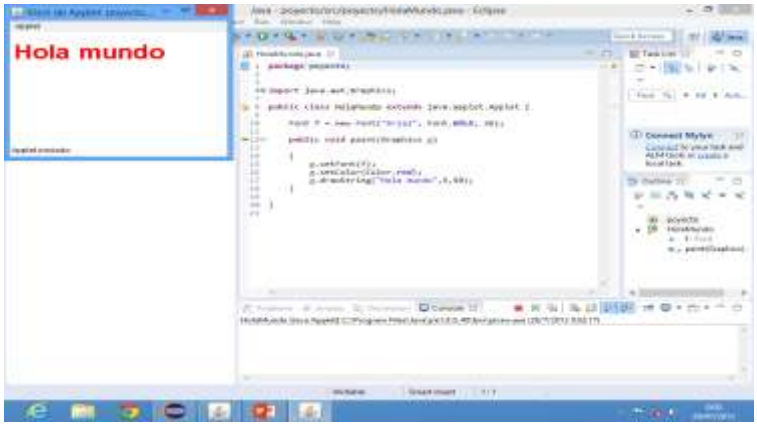
```
<HTML>
<BODY>
<APPLET code="miProgramaApplet.class" width=400 height=400>
</APPLET>
</BODY>
</HTML>
```

- 2 posibilidades para ejecutar el applet:

  - Lanzar un navegador y cargar la pagina html, o bien
  - Usar el programa provisto por Sun para ver applets: appletviewer miProgramaApplet.html

Ejemplo sencillo: ejercicio 19

```
// HolaMundoCruelApplet.java
                    import java.awt.Graphics;
                    import java.awt.Font;
                    import java.awt.Color;
                     public class HolaMundoCruelApplet extends java.applet.Applet
                    {
                    Font f = new Font("Arial", Font.BOLD, 36);
                     public void paint(Graphics g)
                    {
                    g.setFont(f);
                    g.setColor(Color.red);
                    g.drawString("Hola mundo cruel",5,50);
                    }
                    }
                    Crear HolaMundo.HTML:
                    <HTML>
                    <BODY>
                    <applet code="HolaMundoCruelApplet.class" width=400 height=400>
                    </applet>
                    </BODY>
                    </HTML>
```



https://www.programarya.com/Cursos/Java/

UNIDAD 12 Thread o Hilo

Un ***thread*** o ***hilo*** es un ***flujo secuencial simple*** dentro de un ***proceso***. Un único***proceso*** puede tener varios ***hilos*** ejecutándose. Por ejemplo el programa ***Netscape*** sería un proceso,mientras que cada una de las ventanas que se pueden tener abiertas simultáneamente trayendopáginas HTML estaría formada por al menos un ***hilo.***

De tal forma que, los Sistemas Operativos actuales son capaces de ejecutar varios programas "simultáneamente" aunque sólo sedisponga de una CPU: reparten el tiempo entre dos (o más) actividades, o bien utilizan los tiemposmuertos de una actividad (por ejemplo, operaciones de lectura de datos desde el teclado) paratrabajar en la otra. En ordenadores con dos o más procesadores la multitarea es real, ya que cadaprocesador puede ejecutar un ***hilo*** o ***thread*** diferente.

En ***Java*** hay dos formas de crear nuevos ***threads***. La primera de ellas consiste en crear una nueva clase que herede de la clase ***java.lang.Thread*** y sobrecargar el método ***run()*** de dicha clase. El segundo método consiste en declarar una clase que implemente la interface ***java.lang.Runnable***, la cual declarará el método ***run()***; posteriormente se crea un objeto de tipo ***Thread*** pasándole argumento al constructor el objeto creado de la nueva clase (la que implementa la interface ***Runnable***). Como ya se ha apuntado, tanto la clase ***Thread*** como la interface ***Runnable*** pertenecen al package ***java.lang***, por lo que no es necesario importarlas.

Ejemplo ejercicio 20: imprime números impares del 1 al 10. Cree dos instancias (hilos) de cada uno y muestre la salida. Realice el programa utilizando herencia en otra clase.

Clase pares :

```
public class pares extends Thread{
public void run(){
int n;
 int contador=0;
 for (n=1;n<=10;n++){
 if (n % 2 == 0) {
System.out.print(" "+n+"\n");
contador=contador+n;
 }
 }
System.out.print("La suma de los pares es : "+contador + "\n");
System.out.print(" ********** \n");
}
}
```

Clase impares :

```
public class impares extends Thread{
public void run(){
 int n, c=0;
intcontador=0;
 for ( n=1;n<=10;n++){
c= n%2;
 if (c != 0) {
 contador=contador+n;
 System.out.println(" "+n);
 }
  }
  System.out.print("La suma de los impares es : "+contador + "\n");
}
```

```
Ej:20 consola

}
```

Clase principal **:**

```
public class principal {
public static void main(String []args){
pares eje1 = new pares();
 eje1.start();
impares eje2 = new impares();
 eje2.start();
}
}
```

```
run:
 2
 4
 6
 8
 10
La suma de los pares es : 30
 ********************
 1
 3
 5
 7
 9
La suma de los impares es : 25
```

http://www.youtube.com/watch?v=fx
http://www.victomanolo.wordspress.com/ejercicios-2

UNIDAD .13 Plug-in WINDOW BUILDER

El objetivo de este concepto es conocer el empleo del Plug-in WindowBuilder para el desarrollo de interfaces visuales arrastrando componentes. A medida que uno arrastra componentes visuales sobre un formulario se genera en forma automática el código Java. Pasos para crear un JFrame con el WindowBuilder

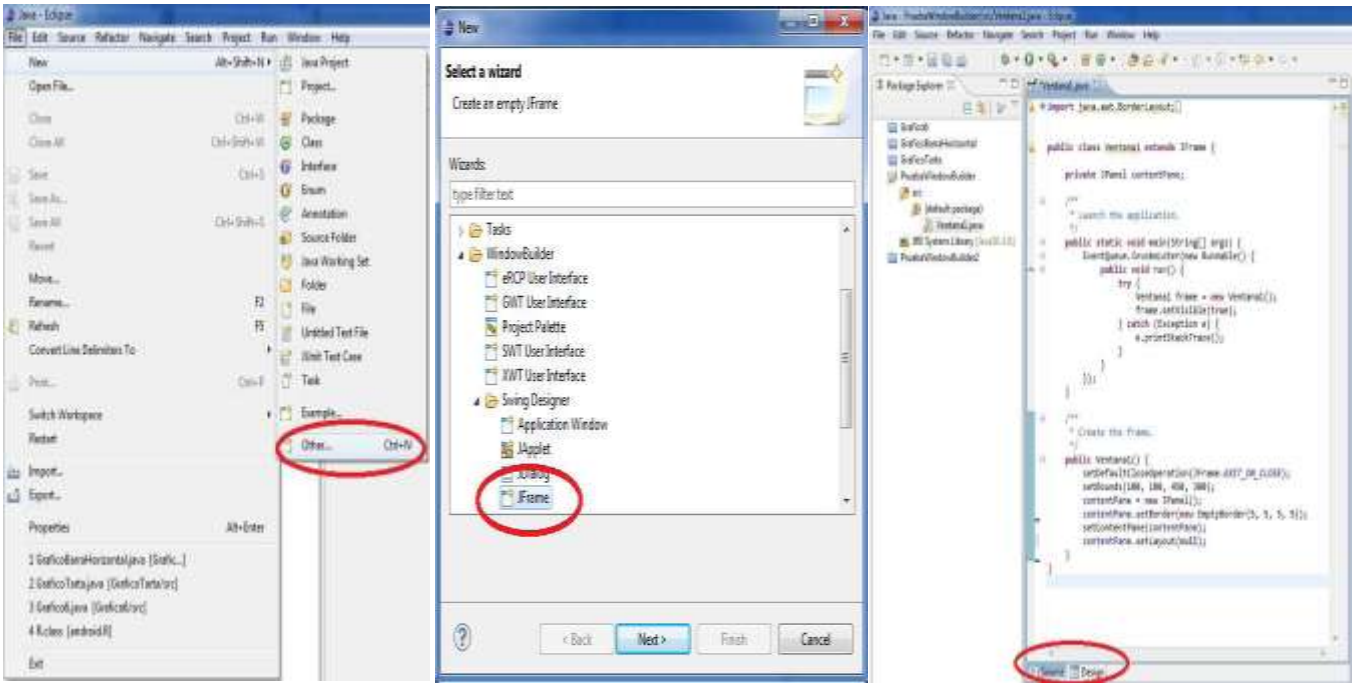
1 - Creación de un proyecto. New Java Proyect

2. Seleccionamos el nombre de nuestro proyecto (lo llamaremos como ejemplo PruebaWindowBuilder).

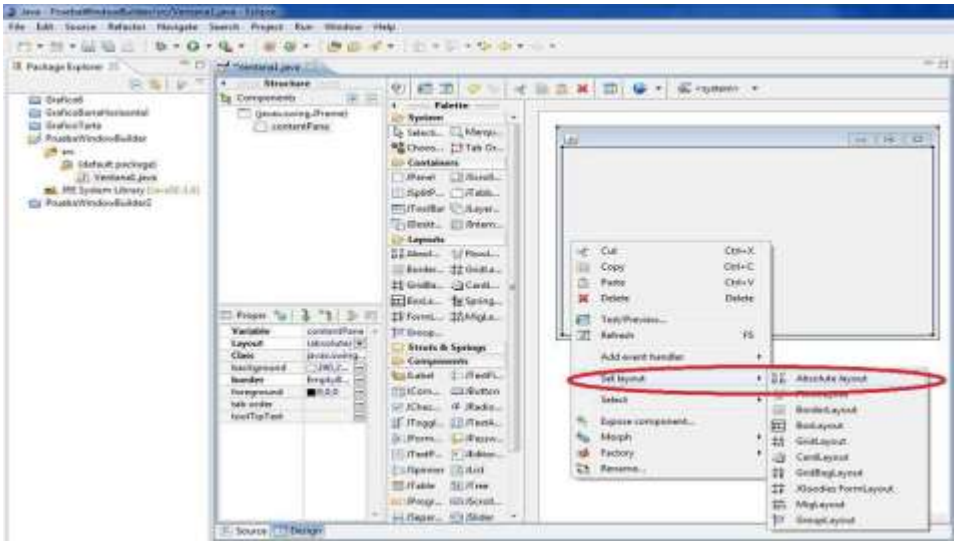
3. Ahora seleccionamos la opción del menú File -> New -> Other ...(fig. 1)

4. Seleccionamos la opción la opción JFrame.(fig. 2).

5-Seguidamente presionamos el botón Next > y definimos el nombre de la clase a crear (Ventana1)

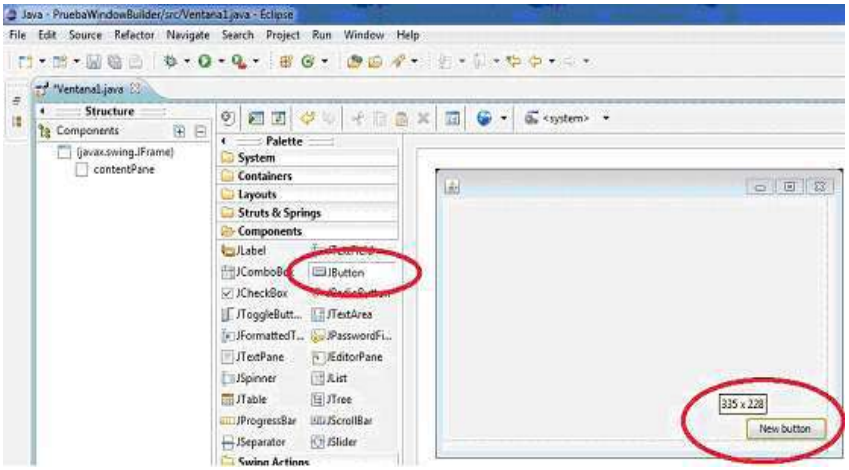


Luego en vista de "Design":



6-Configuramos el Layout de JFrame presionando el botón derecho del mouse sobre el formulario generado yseleccionamos la opción SetLayout > Absolute layout

7. Por último se comienza a arrastrar los objetos para crear una interface

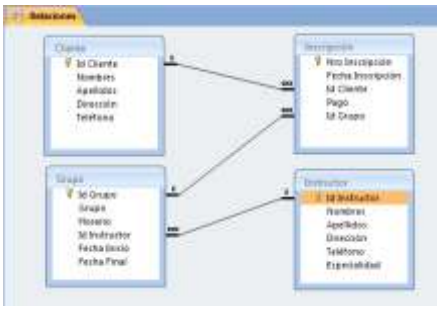


Ejemplo:

Crear un proyecto y luego un JFrame con las siguientes componentes visuales :Dos controles de tipo JLabel, dos JTextField y un JButton para realizar una suma de dos números enteros.

UNIDAD 14. Base de Datos en JAVA

JDBC son las siglas en ingles de Java Database Connectivity. Es un conjunto de clases que nos permite acceder a diversos gestores de bases de datos en forma transparente.Veremos como conectarnos con el motor de base de datos MySQL. Instalación de MySQL integrado en el WampServer Utilizaremos esta herramienta en lugar de descargar solo el MySQL con la finalizar de facilitar la instalación y configuración del motor de base de datos (la instalación de esta herramienta es sumamente sencilla), además utilizaremos otra software que provee el WampServer que es el PhpMyAdmin (es un programa web que nos permite administrar las bases de datos del MySQL) que nos facilitará la creación de la base de datos.Ver pag. web
http://www.javaya.com.ar/detalleconcepto.php?codigo=132&inicio=40
Descargar Driver:
http://www.MYSQL.com/downloads/connector/J/



Problema Propuesto
Partiendo de las siguientes tablas crea formularios para completar un programa que permita la inscripción de un alumno a una academia de especialización en Páginas web, Programación Java, Mantenimiento de redes.

http://javadhc.blogspot.com

BIBLIOGRAFÍA RECOMENDADA

- **Aprendiendo JAVA 1.1 en 21 Días.** *Laura Lemay y Carles L. Perkins* Ed. Prencice Hall
- **Java Unleashed.***Varios autores* Ed. Samsnet
- **The Java FAQ.** *Kanerva* Ed. Addison Wesley
- **JDBC DB access with JAVA.***Hamilton, Cattell y Fisher* Ed. Addison Wesley
- **Manual de JAVA.***Patrick Naughton* Ed. McGraw-Hill

. Aprenda Java. Javier García de Jalón. San Sebastián febrero del 2000. Editada por Javier García de Jalón (jgjalon@ceit.es y jgjalon@etsii.upm.es).
JAVA Diseño e Implementación de Actividades.Walter Mora Flores

Direcciones en INTERNET

http://www.java.sun.com : Dirección donde Sun mantiene información sobre Java.
http://www.javasoft.com : Dirección de la división comercial de Sun dedicada al desarrollo de JAVA.
http://www.developer.com/directories/pages/dir.java.html : Directorio con utilidades JAVA.
http://www.jars.com/: Directorio con utilidades JAVA.
http://ftp.sunet.se/no_sugar/index.htm: Utilidades diversas.
http://ftp.sunet.se/pub/lang/java/ : Directorio FTP con información JAVA.
http://www.xcf.berkeley.edu/lists.html : Hoja con información para suscribirse a listas de correo de Java.
http://www.productlist.com/i_java_faq.htm : Uno de los muchos FAQ's sobre Java.
http://www.unav.es/cti/manuales/Java/curso_java_links.html : Material didáctico complementario al curso.
https://www.programarya.com/Cursos/Java/
https://www.youtube.com/watch?v=L1oMLsiMusQ
https://www.youtube.com/watch?v=vJTeIJx_Kn0
https://www.itcr.ac.cr
http://www.unav.es/cti/manuales/Java/indice
http://www.youtube.com/watch?v=fx
http://www.youtube.com/watch?v=nIYip

GRUPOS DE NOTICIAS

- comp.lang.java.programmer
- comp.lang.java
- comp.lang.java.tech

Definición de términos:

ActionListener:permite hacer cambios a los controles (jcomboBox, Jlabel,)
ActionPerformed: Registra los objetos que gestionarán los dos tipos de eventos soportados
DataInputStream, DataOutputStream: Se utilizan para escribir y leer datos directamente en los formatospropios de Java.
IDE: es un entorno de desarrollo visual. En general incluyen explorador de proyectos, editor, diseñador visual, una paleta de componentes, inspector de propiedades, depurador y compilador
InputStream ,outputStream: lectura y escritura de bytes.
ItemListener: Para capturar la selecciónde un items debemos implementar la interface ItemListener que contiene un método llamada itemStateChanged.
Extends: para heredar de una clase a otra
Finally: cierra el try o fichero.
setBackgroun: método para dar o cambiar color
setBounds, get BoundsObtienen: establecen la posición y el tamaño de un componente
setLayou: Determina el layout manager para este container
setVisible: Permiten chequear o establecer la visibilidad de un componente
getSource:
Reader, Writer: lectura y escritura de caracteres.
Scrollpane: para mostrar la barra de direccionamiento.
Try: habre un fichero para lectura escritura de datos
Thread: hilo
this: cactura de eventos.
Wrappers: (*envoltorios*) son *clases diseñadas para ser un complemento de los tipos primitivos.*
.