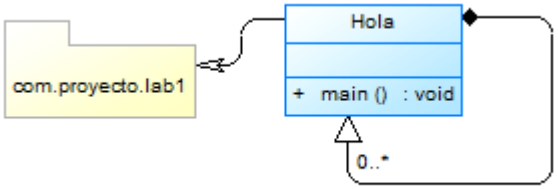
| | MEDIA TÉCNICA DESARROLLO DE SOFTWARE<br>GUIA DE APRENDIZAJE # No.2<br>Módulo Elementos de software 1<br>TEMA: Mi primer Programa en Java | |
|---|---|---|

**Docente:** Juan Carlos Pérez P.
**Alumno :** _____**Fecha :** _____**Nota:**____

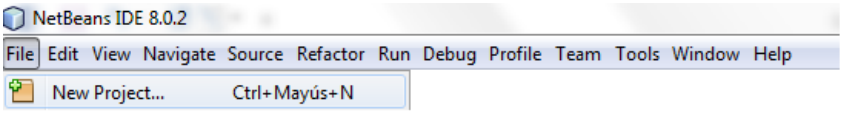
**Justificación***:* Se pretende con éste contribuir a que el alumno se inicie y conozca el entorno del IDE en la creación de pequeños programas en java

**Objetivos:** Crear un programa en Java
Manejar algunas funciones básicas de java
Reconocer el IDE de programación a utilizar
Conocer algunas variables y tipos de datos básicos de java.
Reconocer los símbolos relacionales, asignación en java.
Reconocer los símbolos lógicos en java

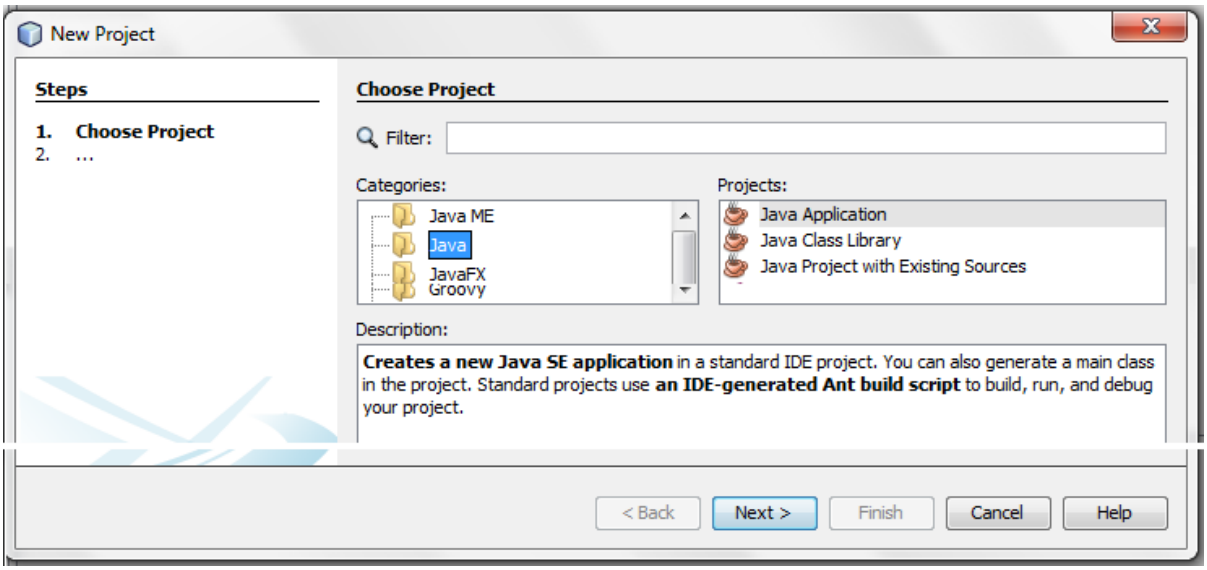
**Contenido.**
**Crear la primera aplicación como lo muestra el siguiente diagrama de clases en UML**



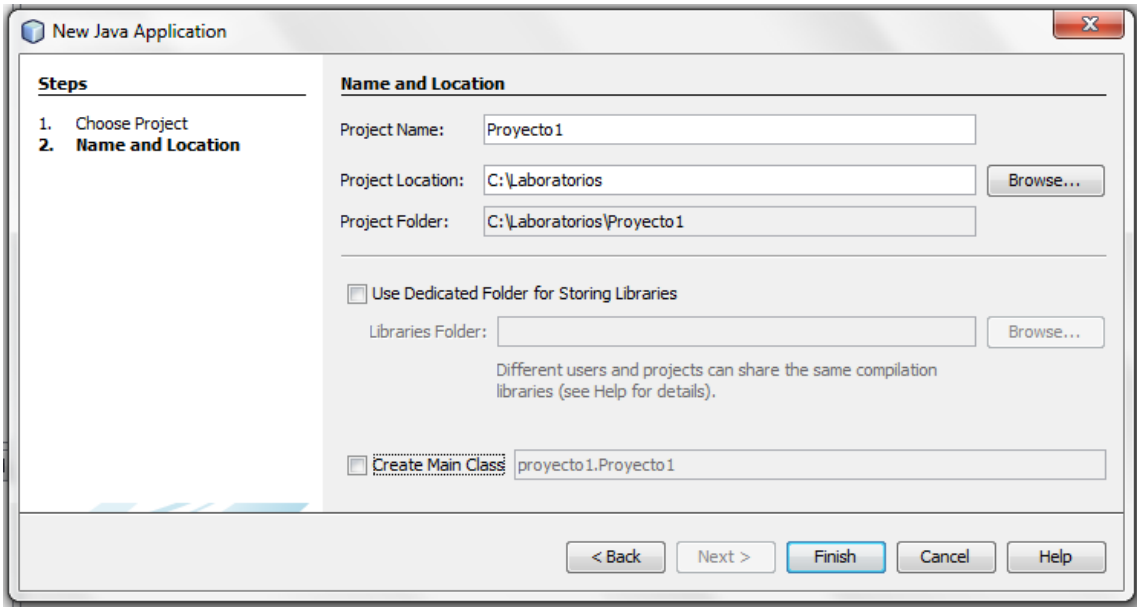

**Pasos:**
**1.**



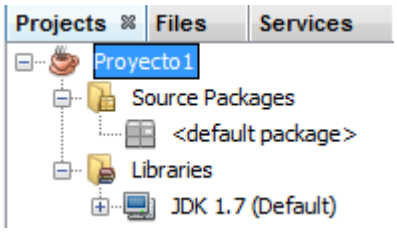

**2.**



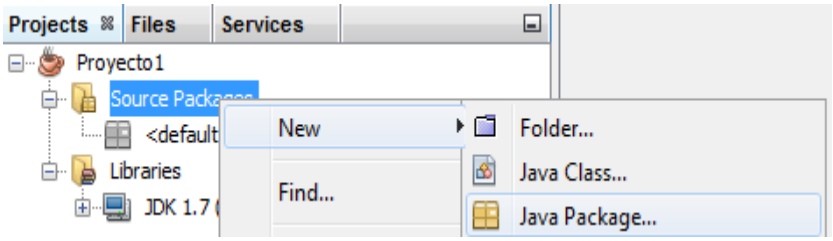
**3.**



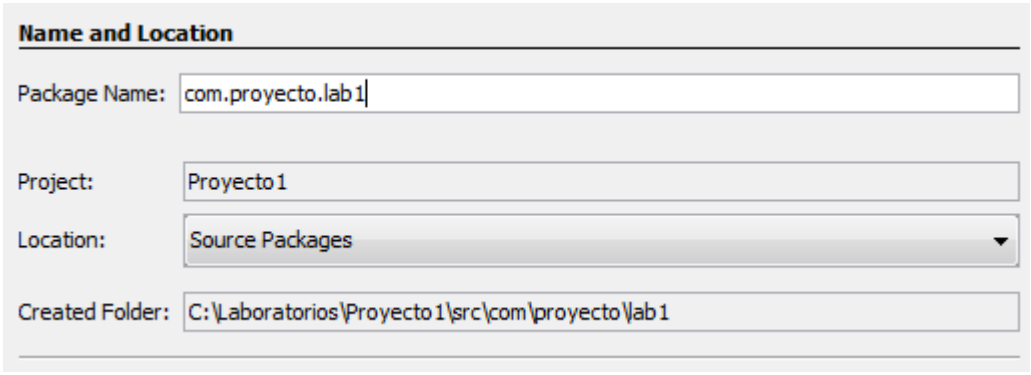

**Una vez creado el Proyecto**



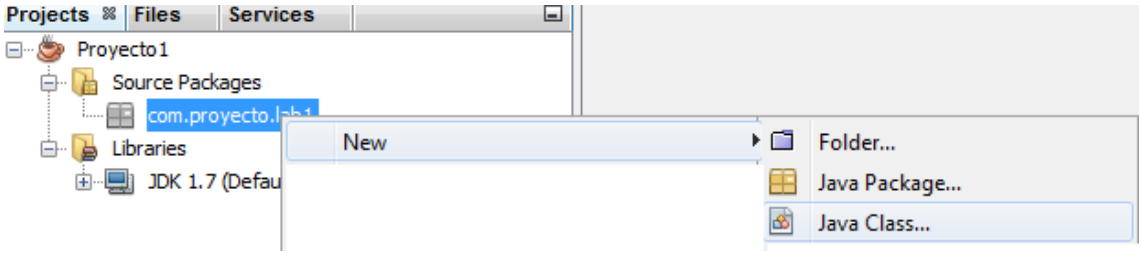

**4. Creamos los paquetes.**

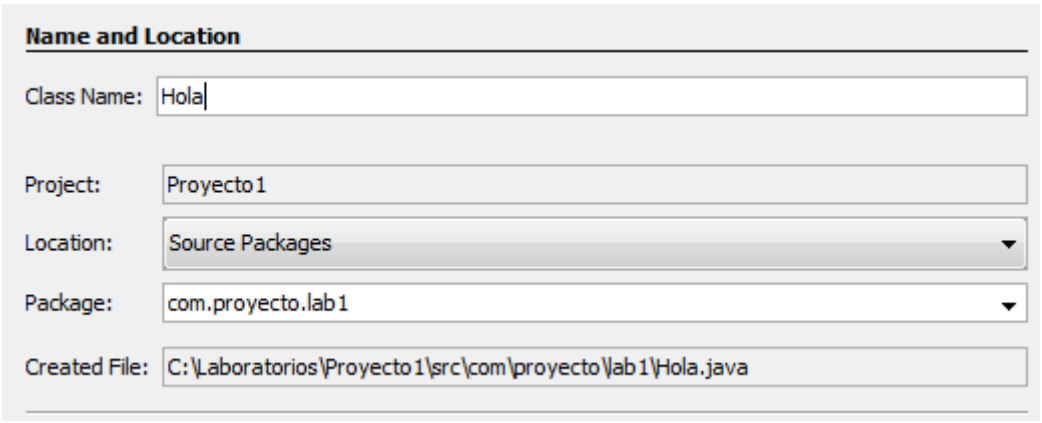



**5.**

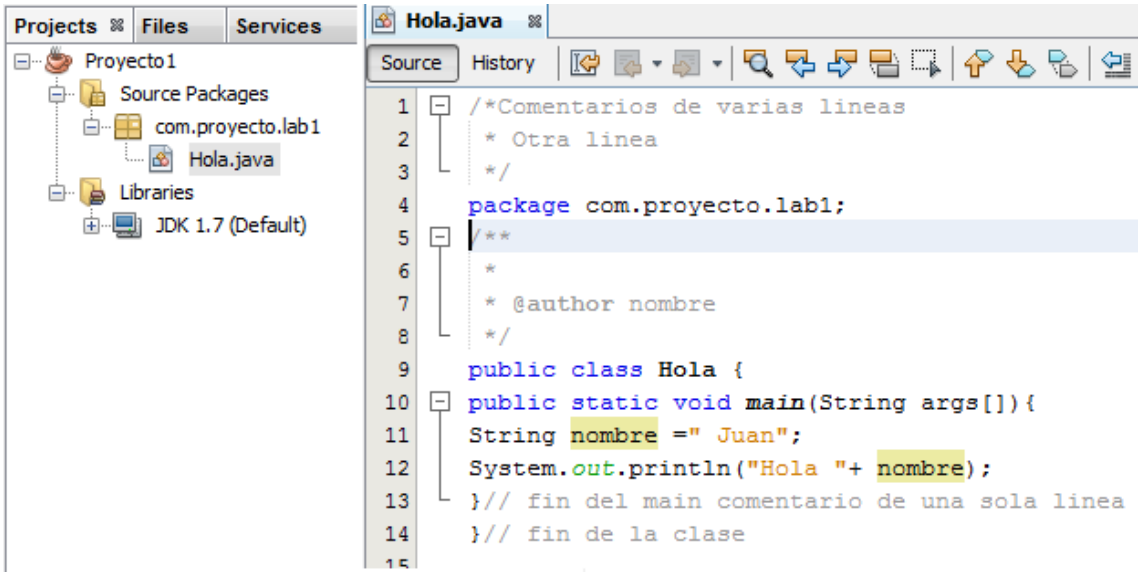


## Agregar la primera Clase llamada Hola.java

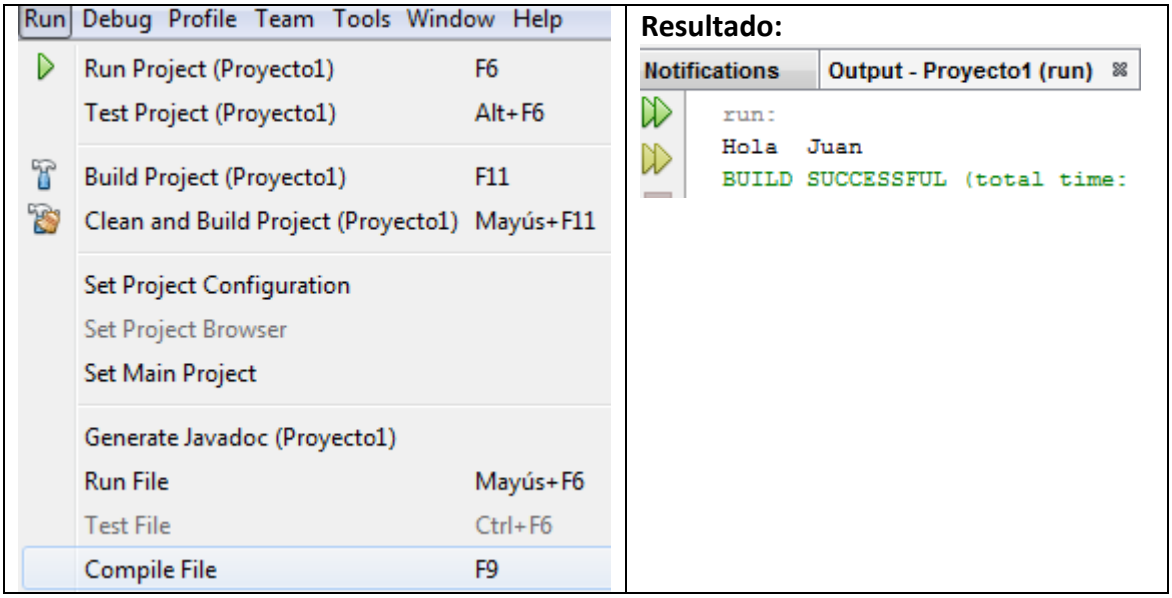



## Digitamos :



**Nuestro programa Hola.java debe compilarse y crear un archivo Hola.class para poderlo ejecutar**




**public static void main (String args[]) { ... }**
En una clase se definen uno o más métodos.
Las palabras **public y static** son atributos del método que discutiremos más tarde.
La palabra void indica que el método main no retorna ningún valor.
La forma (**String args[]**) es la definición de los argumentos que recibe el método main. En este caso se recibe un argumento. Los paréntesis [] indican que el argumentos es un arreglo y la palabra String es el tipo de los elementos del arreglo.
Por lo tanto main recibe como argumento un arreglo de String que corresponden a los argumentos con que se invoca el programa.
**La instrucción System.out.println (...);** despliega un String u ó otro tipo de dato según sea el parámetro de entrada en una **nueva línea**, es similar al método **System.out.print (...);// imprime en una línea el dato** en la consola.

Para realizar un programa en java existen diversos editores. En el curso de java trabajaremos con Netbeans 8.0.2.

**Variables y tipos de datos**

Una variable es un nombre que contiene un valor que puede cambiar a lo largo del programa.

En Java hay dos tipos principales de variables:

**Variables de tipos primitivos.**

Están definidas mediante un valor único que puede ser entero, de punto flotante, carácter o booleano.

| Tipo | Descripción | Clase Envolvente |
|---|---|---|
| boolean | 1 byte, valores true y false | **Boolean** |
| char | 2 bytes, Unicode, comprende el código ascii | **Char** |
| byte | 1 byte, valor entero entre -128 y 127 | **Byte** |
| short | 2 bytes, valor entero entre -32768 y 32767 | **Short** |
| int | 4 bytes, valor entero entre -2,147,483,648 y 2,147,483,647 | **Integer** |
| long | 8 bytes, valor entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807 | **Long** |
| float | 4 bytes, de -3.402823E38 a -1.401298E-45 y de 1.401298E-45 a 3.402823E38 | **Float** |
| double | 8 bytes, de -1.79769312486232E308 a -4.94065645841247E-324 y de 4.94065645841247E-324 a 1.79769313486232E308 | **Double** |

**Variables de referencia**. Estas variables son referencias o nombres de arrays, colecciones u objetos de una determinada clase.

Hay variables miembro de una clase y variables locales. Las variables miembro son declaradas fuera de cualquier método preferentemente abajo de la cabecera de la clase, y las locales se crean en el interior de un método y se destruyen al finalizar dicho método. Ambas pueden ser de tipos primitivos o de referencia.

Los nombres de las variables pueden ser creados con mucha libertad. Pueden ser un conjunto de letras y números, sin algunos caracteres especiales para Java como son: -,*, /, etc. Ejemplos:

```
int x;                          // x es inicializada a 0
int x=5;                        //x es inicializada a 5
int vector [ ];                 //vector inicializado a null
string saludo="hola";           //saludo es inicializada con la cadena "hola"
```

**Operadores y Expresiones**

Los operadores en Java son binarios (requieren siempre de dos operandos) que realizan operaciones aritméticas habituales: suma (+), resta (-), multiplicación (*), división (/) y modulo (%).

**Operadores de Asignación**

El operador por excelencia es el =. Java dispone de otros operadores de asignación que realizan operaciones acumulativas sobre una variable. La siguiente tabla muestra estos operadores:

| 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 |

**Operadores Unarios**

Los operadores más (+) y menos (-) sirven para mantener o cambiar el signo de una variable.

**Operador instanceof**

Permite saber si un objeto pertenece a una determinada clase o no. Su formato es: *nombreObjeto instanceof className* y devuelve true o false.

**Operador condicional (**ternario**) ? :**
Permite realizar bifurcaciones condicionales sencillas.Su formato es:
*Expresion Booleana ? res1: res2*
Donde se evalúa la expresión y se devuelve res1 si el resultado es true y res2 si el resultado es false.

**Operadores incrementales:** Java dispone del operador incremento (++) el cual incrementa en una unidad la variable a la que se aplica, y decremento (--) la reduce en una unidad. Pueden ser utilizados de dos formas:

1. Precediendo a la variable ( ++i ). En este caso primero se incrementa la variable y después se utiliza.
2. Siguiendo a la variable ( i++ ). Es este caso primero se usa la variable y luego se incrementa su valor.

**Operadores relacionales**

El resultado de estos operadores es siempre un valor boolean (true o false).

| Operador | Utilización | El resultado es true |
|---|---|---|
| > | 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 es igual que op2 |
| != | op1 != op2 | si op1 y op2 son diferentes |

**Operadores lógicos**

La siguiente tabla muestra los operadores lógicos de Java.

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

**Operadores de concatenación de cadenas de caracteres. (+)**

El operador más (+) se utiliza para llevar a cabo esta acción. Ejemplo: "El total asciende a:" + result + "unidades".

**Promoción**

Según Bruce Eckel, en Piensa en Java 2da edición: **”**

Al hacer operaciones matemáticas o de bit sobre tipos de datos primitivos, se descubrirá que si son más pequeños que un **int** (es decir, **char, byte,** o **short),** estos valores se promocionarán a **int** antes de hacer las operaciones, y el valor resultante será de tipo **int.** Por tanto, si se desea asignar el valor devuelto, de nuevo al tipo de menor tamaño, será necesario utilizar una conversión. (Y dado que se está haciendo una asignación, de nuevo hacia un tipo más pequeño, se podría estar perdiendo información.)
En general, el tipo de datos de mayor tamaño en una expresión será el que determine tamaño del resultado de esa expresión; si se multiplica un float y un double, el resultado será double; si se suman un int y un long, el resultado será long.**”**

**Actividades:** Implementar algunos programas en java

**Evaluación:**

1. Defina package, class, main, compilar.
2. Elabore un programa en java que despliegue
   Su nombre, edad, estatura, peso, nombre

**Recursos:** Software Netbeans 8.02, jdk.1.7 , Guía de aprendizaje
http://www.juanperezu.com

**Bibliografía:**
www.lawebdelprogramador.com
www.java.sum.com,
www.lawebdelprogramador.com
Piensa en Java ,Mac Graw Hill.