



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y
COMPUTACIÓN

Investigación de Lenguajes - JAVA

Autores:

Raquel VILLÓN
Adriana RODRÍGUEZ
Marcelo SÁNCHEZ

Profesor:

Ing. Javier TIBAU

23 de octubre de 2013

Índice general

1. Introducción	5
2. Características	7
2.1. Orientado a Objetos	7
2.2. Robusto	7
2.3. Portable	7
2.4. MultiThread	7
3. Historia	9
4. Tutorial de Instalación	11
5. Hola Mundo y otros Programas Introdutorios	17
Bibliografía	21

Capítulo 1

Introducción

En este primer capítulo se podrá analizar de acerca de qué es Java, conocer acerca de las principales funcionalidades así como también las características que destaca a este lenguaje orientado a objetos.

Java basicamente es un lenguaje compilador e interpretador [1] esto quiere decir que el código que genera es revisado por una máquina virtual (JVM) que lo que hace es mantener alejado el código en bytecodes sin la necesidad de revisar que sistema operativo es la que contiene la máquina en la que se está trabajando. Además, la máquina virtual ayuda con la gestión de la memoria ya que este consta con la ayuda del recolector de basura (que mantiene un control de liberación de memoria) llevan un control dinámico entre si.

El recolector de basura (*Garbage Collector*) es el encargado principal del control de la localidad de la memoria para los objetos; dicho por [2] es el responsable que los objetos estén correctamente referenciados, siendo aquellos denominados como 'vivos' mientras que los objetos que no están referenciados son considerados como objetos 'muertos' en lo que allí el recolector de basura se encarga de liberar el espacio de memoria que se había establecido para ese objeto. Un breve ejemplo sobre como ocupa espacio en la memoria tanto en llamadas como espacio se puede observar en la *Figura 1.1*.

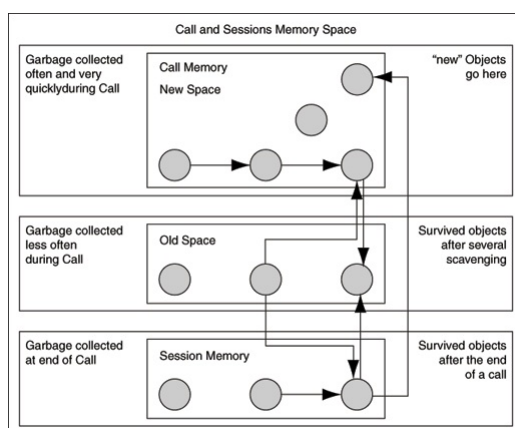


Figura 1.1: Uso del Garbage Collector en la memoria.

Se dice que Java es un OOP (Object-Oriented-Programming) [5], contribuyendo en su sintaxis al manejo de objetos, que es un software formado a partir de un comportamiento y estado. La figura *Figura 1.2* muestra como está compuesto un objeto. *Estado* es la característica que tiene el objeto (nombre, dirección, teléfono, etc.), mientras que el *comportamiento* se refiere a la actividad que este realiza(respirar, caminar, etc.)

Otro aspecto principal con lo que está proporcionado java es por el API (Application Programmer Interface) en el posee una colección de componentes que trabajan entre si para el uso en la gestión de interfaces, redes, y en especial algoritmos que maneja encriptación y certificados. Esta información es útil principalmente para el manejo de seguridad, cabe mencionar que el JVM también lleva crédito en el control de seguridad.

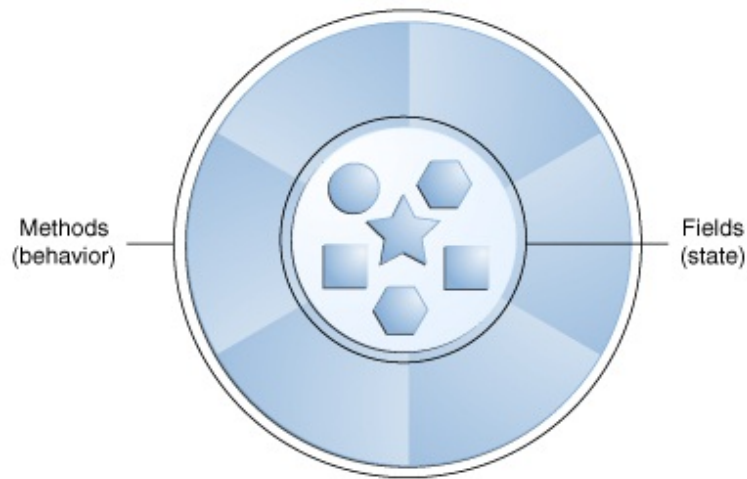


Figura 1.2: A software Object.

Capítulo 2

Características

Entre las principales características que posee Java tenemos:

2.1. Orientado a Objetos

Mencionado anteriormente, es una de las características principales ya que por su manejo de objetos provee una mayor facilidad en la sintáxis y en el control de métodos.

2.2. Robusto

Que el lenguaje sea robusto indica que posee una mayor rigidez al momento de codificar el código, porque se va compilando para verificar que lo que se esté codificando este sin errores.

2.3. Portable

La portabilidad, ayuda en el hecho que no es necesario especificar un sistema operativo, porque consta con el JVM que hace que esta condición sea indiferente.

2.4. MultiThread

Considerado como los pocos lenguajes que maneja hilos, que son aquellos que poseen una acción específica en un periodo de tiempo. [6] indica que los hilos permiten a una aplicación ejecutar múltiples hilos parcialmente.

Capítulo 3

Historia

Según los hermanos Deitel [3], el lenguaje java fue creado en 1991 por la compañía Sun Microsystems, la cual es una empresa dedicada a la fabricación y venta de servidores y componentes de componentes informáticos. Empezó como un proyecto de investigación de la compañía denominado Green y fue pensado para programar los dispositivos electrónicos inteligentes.

Deitel también indica que Java está basado en C++. Al principio su nombre fue Oak, pero luego se descubrió que ese nombre ya lo tenía otro lenguaje de programación. Luego, en una reunión de la gente de Sun en una cafetería, se propuso el nombre de Java, el cual es una variedad de café.

Al principio el proyecto Green tuvo problemas debido a que el avance de los dispositivos electrónicos inteligentes no se desarrollaba tan rápido como Sun había previsto. Sin embargo, cuando la World Wide Web explotó, Sun vio rápidamente el potencial que Java tenía para darle dinamismo a las páginas web. De esa manera el proyecto Green pudo avanzar y finalizar con una conferencia en mayo de 1995, en la que Sun Microsystems anunció formalmente la existencia del lenguaje de programación Java.

A medida que la World Wide Web avanzaba Java tenía mayor acogida y se desarrolló de tal modo en el que luego se lo utilizó para desarrollar grandes aplicaciones empresariales, aplicaciones para dispositivos móviles, radiolocalizadores, asistentes personales digitales, entre otros.

En el 2009, Sun Microsystems fue comprada por Oracle por un valor de 7.400 millones de dólares según el portal 20minutos.es[4]. Y actualmente se distribuye la versión número 7 de Java, así como también sus diferentes líneas. Estas líneas se especializan en un ambiente, por ejemplo: Java SE está creada especialmente para aplicaciones de escritorio; Java EE es especializado en crear aplicaciones web; Java ME es utilizado para desarrollar aplicaciones que se ejecutan en dispositivos móviles, etc.

Capítulo 4

Tutorial de Instalación

Para poder correr aplicaciones creadas en Java se necesita mínimo instalar Java Runtime Environment (JRE). Pero si lo que se quiere es desarrollar aplicaciones en Java, entonces necesita Java Development Kit (JDK) que contiene al JRE y herramientas para desarrollar, depurar y monitorear aplicaciones en Java (Oracle.com).

En este documento mostraremos la instalación de Java SE 7, distribución para el desarrollo de aplicaciones de escritorio, en plataforma Windows 8 de 64bits. Para esto necesitaremos lo siguiente:

1. Descargar el JDK directamente de la página oficial de Oracle¹.

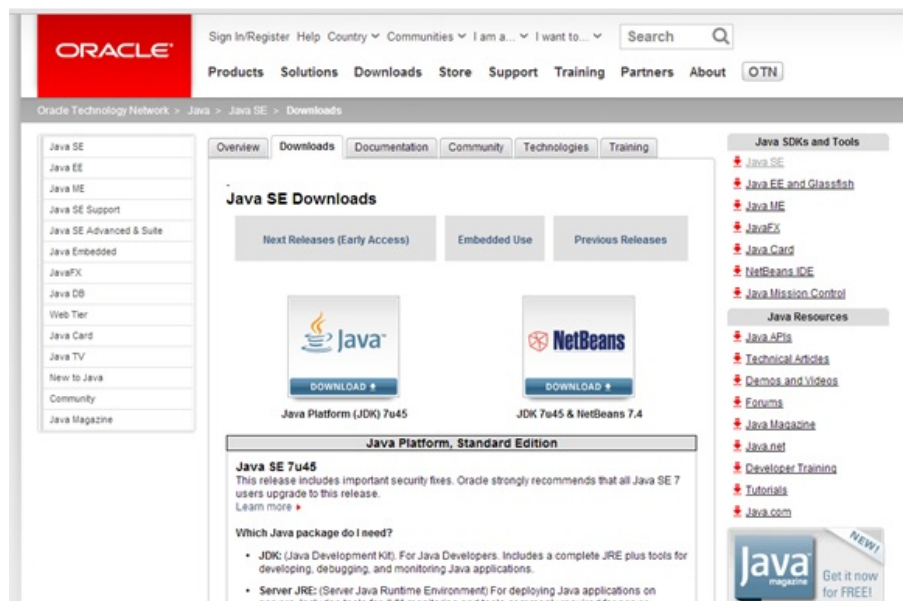


Figura 4.1: Página de descarga del JDK.

En la Figura 4.1 se muestra la opción para descargar sólo el JDK o el JDK con el IDE de desarrollo creado por Oracle, NetBeans. Escogeremos descargar sólo el JDK.

2. Para poder descargar el JDK debemos aceptar la licencia Y escoger el instalador adecuado para el Sistema operativo que se esté utilizando como se muestra en la Figura 4.2.
3. Cuando finalice la descarga, se podrá comenzar la instalación ayudado por una wizzard de windows de la Figura 4.3 a la Figura 4.8.

¹<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

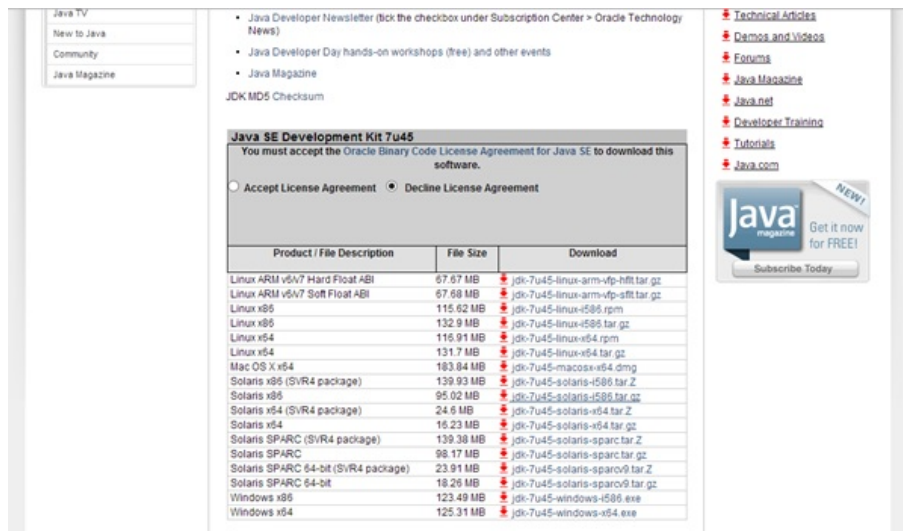


Figura 4.2: Página de descarga del JDK.



Figura 4.3: Página de descarga del JDK.



Figura 4.4: Página de descarga del JDK.

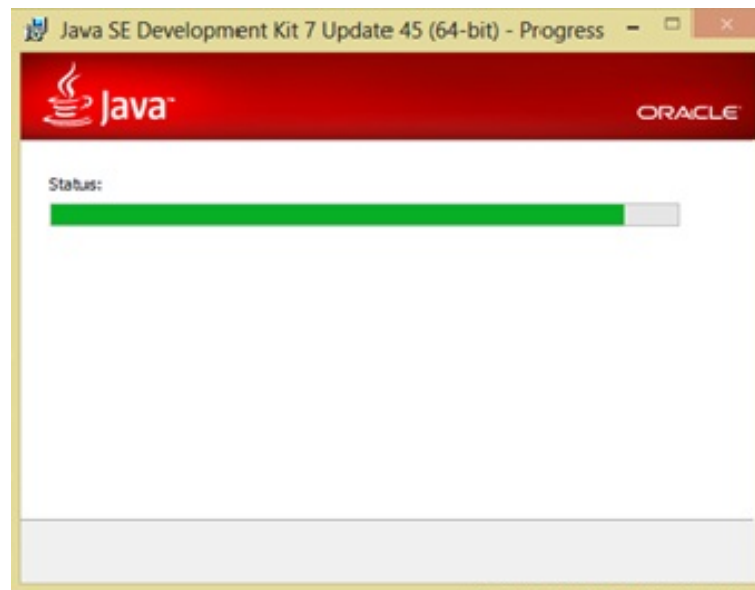


Figura 4.5: Página de descarga del JDK.

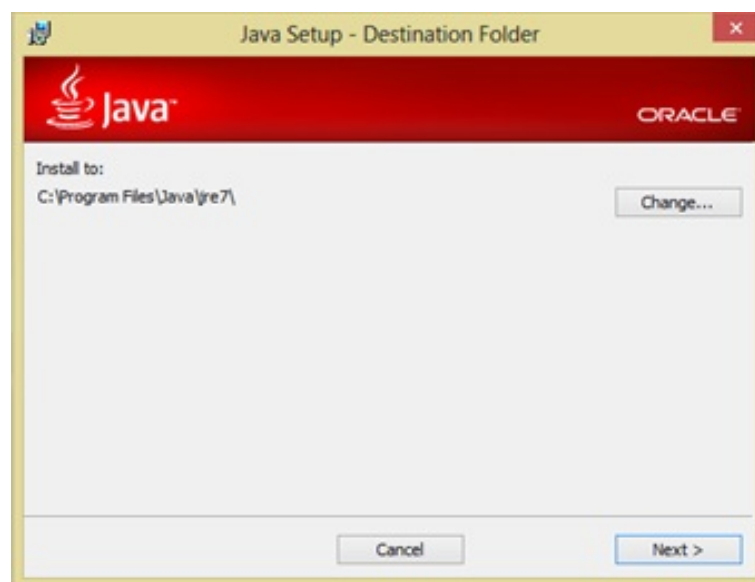


Figura 4.6: Página de descarga del JDK.



Figura 4.7: Página de descarga del JDK.



Figura 4.8: Página de descarga del JDK.

Y listo!

Anteriormente hubiera sido necesario agregar una nueva variable de entorno para que el sistema sepa dónde buscar el compilador y ejecutor de java. Pero desde el sistema operativo Windows 7, hacer esta especificación ya no es necesaria.

Capítulo 5

Hola Mundo y otros Programas Introdutorios

Cada vez que se usa una computadora, esta ejecuta múltiples aplicaciones que realizan tareas para el usuario. Los programadores, son los encargados de elaborar estas aplicaciones, escribiendo programas de cómputo que permiten a los usuarios cumplir sus tareas diarias. Ahora vamos a considerar la tarea de programar una aplicación sencilla, la cual muestra únicamente una línea de texto con el mensaje “Hola mundo”, en consola, esta aplicación sera la base para demostrar que programar en Java es fácil, todo programador ha escrito estas líneas de código, probablemente en sus inicios en este lenguaje de programación. Para esto necesitaremos hacer uso de un IDE¹, en este ejemplo usaremos el IDE de Java, Netbeans en su mas reciente version, el cual ya habremos instalado previamente. Para crear una aplicación en Java debemos ir al menú de Archivo y, a continuación hacer clic en Nuevo Proyecto.

En la ventana que se despliega a continuación (fig. 5.1) en el área de Categoría hacemos clic en Java y en el área de Proyecto, hacemos clic en Java Application, luego damos clic en siguiente.

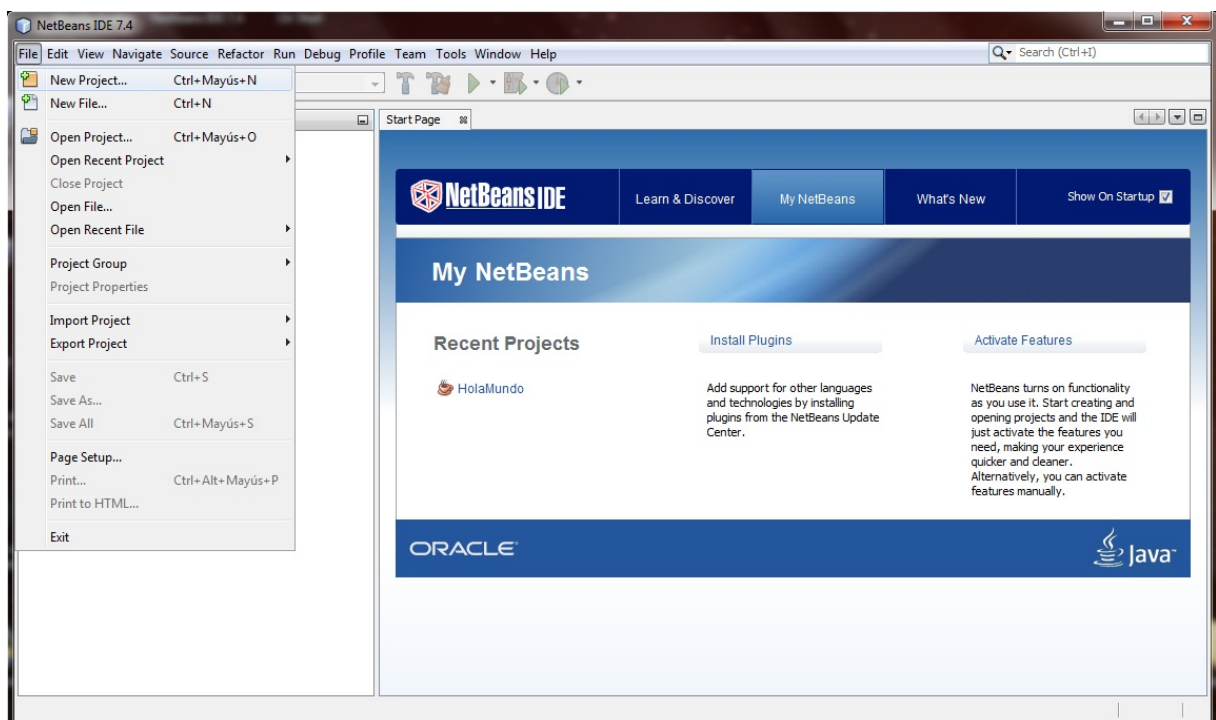


Figura 5.1: Ventana Principal, Netbeans.

¹Existen otros IDEs para programar en Java tales como; Netbeans, JCreator, Processing, BlueJ, Kawa, JBuilder, DrJava, entre otros disponibles en la web

En la nueva ventana(fig. 5.2), en el recuadro de Nombre del Proyecto, escribimos un nombre², luego hacemos clic en siguiente.

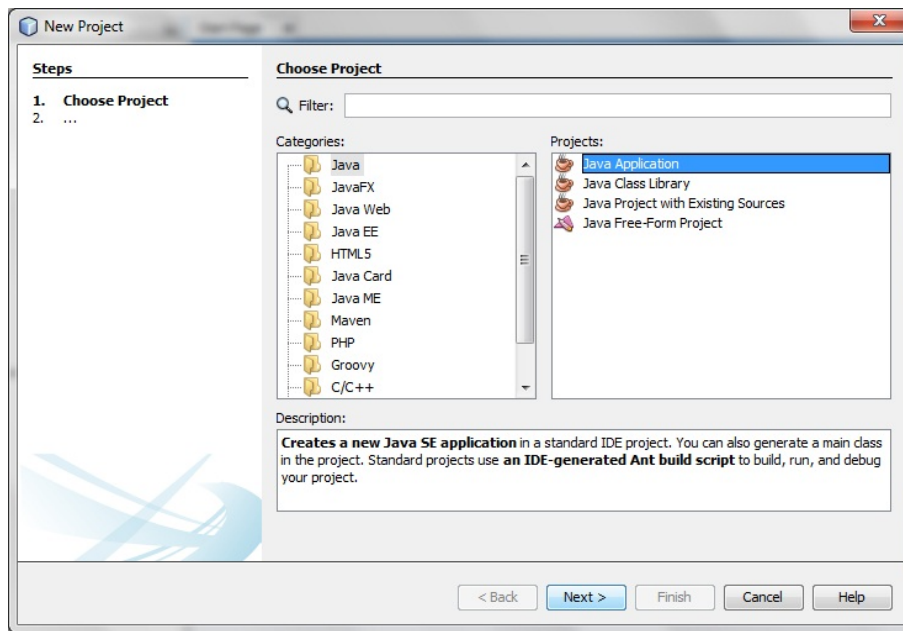


Figura 5.2: Ventana Nuevo Proyecto, Netbeans.

Ya en la ventana de nuestro nuevo proyecto(fig. 5.3), podemos notar que luego de la línea:

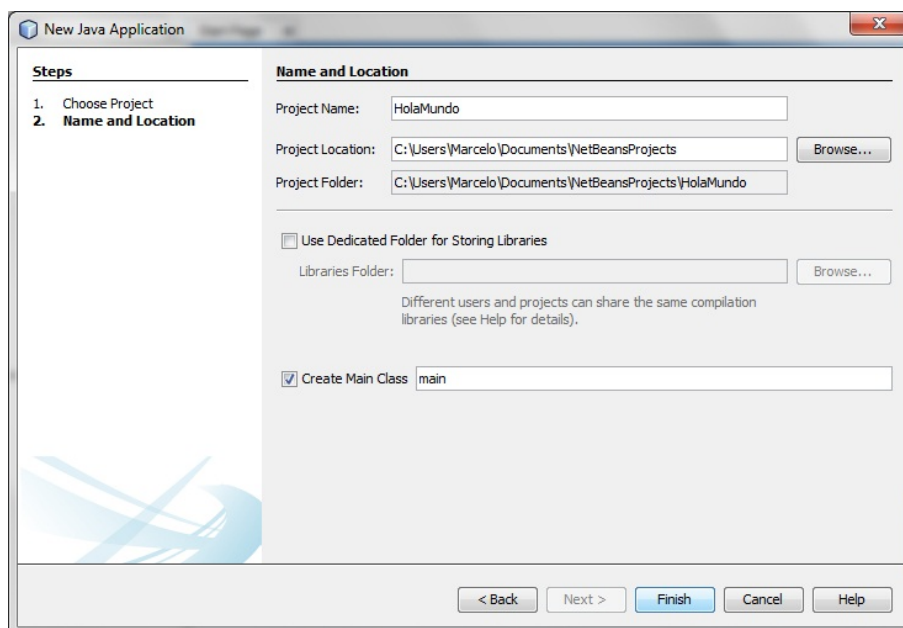


Figura 5.3: Ventana Nueva Aplicación Java, Netbeans.

```
1 public static void main (String argv[])
2 {
3     // Todo el CODIGO va AQUI
4
5 }
```

²El nombre no debe contener espacios en blanco ni caracteres especiales, para evitar posibles conflictos, también se recomienda que sea corto y altamente descriptivo.

Existe un comentario que dice “Todo el CÓDIGO va AQUÍ”, y es porque es allí donde debemos escribir la función para imprimir el mensaje en pantalla, el código quedaría de la siguiente manera.

```
1  /**
2   * Este es un comentario
3   */
4  public class HolaMundo {
5      public static void main (String argv[])
6      {
7          // Este tambien es un Comentario
8          System.out.println("Hola mundo!");
9      }
10 }
```

Una vez escritas las líneas de código, pulsamos la tecla F5 o en su defecto vamos al menú Ejecutar, Ejecutar Proyecto.

Y si esta todo correctamente escrito como en la fig. 5.4, en consola se debería mostrar el mensaje: Hola mundo!.

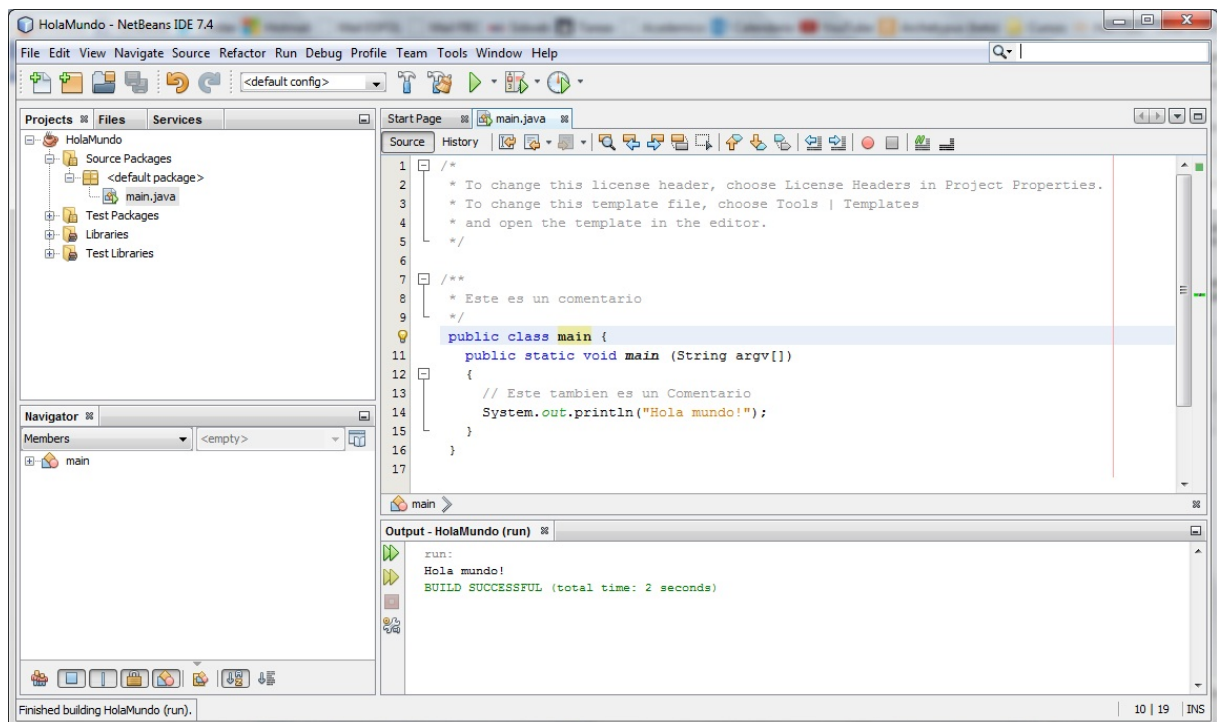


Figura 5.4: Ejecución correcta, Aplicación Java.

Uno de las aplicaciones básicas que se realizan en Java, al momento que se está aprendiendo a programar, es la aplicación que pide información y devuelve otra a cambio, formularios sencillos que nos ayudan a comprender un poco, como la maquina es educada para resolver nuestros requerimientos, con los comandos de Java. Procederemos a crear un nuevo proyecto con los pasos descritos en las fig 5.1, 5.2, 5.3.

En la línea de comando ingresamos lo siguiente;

```
1 public class solicitardatos {
2     public static void main(String[] args) {
3         String anioS;
4         String nombre;
5         int anio, edad;
6         nombre = JOptionPane.showInputDialog("Como te llamas");
7         anioS = JOptionPane.showInputDialog("En que anio naciste"
8         );
9         anio = Integer.parseInt(anioS);
10        edad = 2013 - anio;
11        System.out.println("Hola " + nombre + " tienes " + edad +
12        " años");
13    }
14 }
```

Primeramente lo que hacemos es definir las variables que utilizaremos, dos variables tipo String y dos enteros tipo int:

```
1 String anioS;
2 String nombre;
3 int anio, edad;
```

Luego con la función *JOptionPane.showInputDialog()*; mostramos una ventana en la cual pedimos el nombre y lo almacenamos en la variable tipo String nombre.

```
1 nombre = JOptionPane.showInputDialog("Como te llamas");
```

Lo mismo para el año de nacimiento:

```
1 anioS = JOptionPane.showInputDialog("En que anio naciste");
```

Puesto que queremos calcular la edad para mostrarla en consola, necesitamos convertir el String en entero tipo int, esto lo hacemos con la función.

```
1 anio = Integer.parseInt(anioS);
```

Luego hacemos el cálculo de la edad, con una resta simple y la almacenamos en la variable edad.

```
1 edad = 2013 - anio;
```

Finalmente con toda la información que tenemos, se imprime en consola, con la instrucción:

```
1 System.out.println("Hola " + nombre + " tienes " + edad + "
    años");
```

Bibliografía

- [1] Belmonte, O. (2005). *Introducción al lenguaje de programación Java*. [en línea]. Disponible en: <http://www3.uji.es/belfern/pdidoc/IX26/Documentos/introJava.pdf>.
- [2] Sun Microsystems. (2006). *Memory Management in the Java Virtual Machine* [en línea]. Disponible en: <http://www.oracle.com/technetwork/java/javase/memorymanagement-whitepaper-150215.pdf>
- [3] DEITEL, Paul J. y Harvey M. DEITEL (2007). *Cómo Programar en Java*. México..Pearson Education.
- [4] Portal 20Minutos.(2013). *Oracle compra Sun Microsystems* [en línea]. Disponible en: <http://www.20minutos.es/noticia/464024/0/oracle/compra/sun/>
- [5] Sun Microsystems. *Object-Oriented Programming Concepts* [en línea]. Disponible en: <http://docs.oracle.com/javase/tutorial/java/concepts/>
- [6] Oracle. *API Java-Thread* [en línea]. Disponible en: <http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/Thread.html>