



CAP.

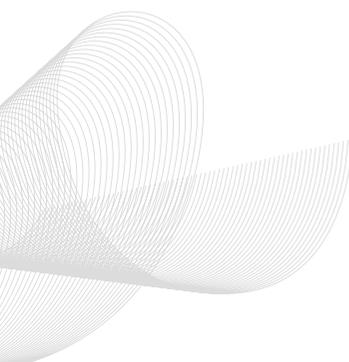
1

Introducción al lenguaje Java

CAPACIDAD:

- Reconocer los conceptos básicos del lenguaje Java.
- Configurar e instalar las aplicaciones Java en una computadora.
- Reconocer, diferenciar e instalar los diferentes IDE para las aplicaciones Java.

CONTENIDO:

- 1.1. Introducción
 - 1.2. Orígenes del lenguaje Java
 - 1.3. Los programas en Java
 - 1.4. Evolución de la plataforma Java
 - 1.5. Tipos de aplicaciones Java
 - 1.6. Requisitos para la instalación de Java
 - 1.7. El JDK 7 y su instalación
 - 1.8. El IDE JCreator y su instalación
 - 1.9. El IDE NetBeans y su instalación
 - 1.10. El IDE JDeveloper 11g y su instalación
- 

1.1. INTRODUCCIÓN

Muchos de nosotros hemos escuchado acerca de Java sin saber que probablemente nuestras vidas están girando alrededor de este término, veamos un ejemplo simple; si usted tiene un celular en sus manos las aplicaciones con las que cuenta pueden ser Java, pero a qué se debe que Java esté en aparatos de uso diario existiendo otros lenguajes como Visual Basic, lo que diferencia a Java de otras aplicaciones es la portabilidad con la que cuentan sus aplicaciones. Otra característica favorable de Java es que cuenta con un texto plano lo que hace menos pesadas sus aplicaciones y que estas se puedan ejecutar en aparatos de poca memoria.

Java, en la actualidad, ha revolucionado la programación y ha promovido un código derivado de C y C++ que en corto tiempo ha tenido mucho éxito en las tecnologías de información debido a su filosofía y la forma de operar de sus **aplicaciones cliente-servidor**, tanto en plataforma como en la web. Por tanto, son muchas las empresas que están apostando por Java como un potente lenguaje de programación duradero en el tiempo y 100% confiable.

En este primer capítulo se verá a Java desde una perspectiva global, conociendo su historia, sus principales características y cómo se trabaja la programación orientada a objetos en Java.



1.2. ORÍGENES DEL LENGUAJE JAVA

Java fue diseñado por James Gosling licenciado en Ciencias de la Computación de la Universidad de Calgary, cuando era empleado de *Sun Microsystem* este es reconocido como el diseñador e implementador del compilador y la máquina virtual de Java (JVM), por lo que fue elegido miembro de la Academia Nacional de Ingeniería de Estados Unidos (NAE).



Inicialmente Java fue diseñado para dispositivos electrónicos relativamente pequeños como la calculadora, el microondas, el refrigerador y la televisión interactiva. Debido a la existencia de múltiples tipos de electrodomésticos y a los constantes cambios en los mismos era necesario una herramienta que no dependiera del tipo aparato, es decir, se necesitaba un código neutro; la idea principal era ejecutar los códigos en una máquina virtual que lo hiciera portable convirtiendo el código en uso particular por el electrodoméstico.

Inicialmente, Java no fue acogido por las empresas de electrodomésticos, siendo así que en el año 1995 Java se introduce como lenguaje de programación para computadoras, la incorporación de un intérprete Java en la versión 2.0 del navegador Netscape hizo de Java un lenguaje potente revolucionando la internet.

Java se creó como una herramienta de programación para ser usada en un proyecto de *set-top-box* en una pequeña operación denominada "The Green Project" en Sun Microsystems en el año 1991. El

equipo (*Green Team*), compuesto por trece personas y dirigido por James Gosling, trabajó durante 18 meses en Sand Hill Road en Menlo Park en su desarrollo.

Java como lenguaje se le denominó inicialmente “Oak” del castellano roble, luego pasó a denominarse Green tras descubrirse que “Oak” era ya una marca comercial registrada en Estados Unidos para adaptadores de tarjetas gráficas y finalmente, se renombró a Java.

Para el nombre que quedó registrado al final se han escrito muchas hipótesis, sobre el término algunos señalan que podría tratarse de las iniciales de sus creadores: James Gosling, Arthur Van Hoff, y Andy Bechtolsheim. Aunque la hipótesis que más fuerza tiene es la que Java debe su nombre a un tipo de café disponible en la cafetería cercana a la empresa Sun, de ahí que el ícono de Java sea una taza de café caliente. Un pequeño indicio que da fuerza a esta teoría es que los 4 primeros bytes (el número mágico) de los archivos .class que genera el compilador son en hexadecimal, 0xCAFEBABE. Otros afirman que el nombre fue sacado al parecer de una lista aleatoria de palabras. A pesar de todas estas teorías, ninguna ha podido ser comprobada.

La promesa inicial de Gosling era: “*Write Once, Run Anywhere*” (“escríbelo una vez, ejecútalo en cualquier lugar”), se cumplió al poder Java proporcionar un lenguaje independiente de la plataforma y un entorno de ejecución (la JVM) ligero y gratuito para las plataformas más populares. Es decir, que los binarios (bytecode) de las aplicaciones Java pudiesen ejecutarse en cualquier plataforma.

1.2.1. Comparando Java con C++

Partiendo del paradigma orientado a objetos, el lenguaje C++ es capaz de interpretar dos estilos de programación como son la programación estructurada y la orientada a objetos, a esta característica se le llama programación hacia atrás; este término no es un estereotipo de Java, puesto que este programa trabaja totalmente orientado a objetos sin perder la esencia del paradigma. Hay que reconocer que Java hereda de C++ muchas características como el punto y coma al finalizar una instrucción o su forma de declarar una variable; si usted programó en C++ se sentirá familiarizado con Java.

Ahora, comparar dos lenguajes de programación resulta un tanto difícil si comparamos que ambos lenguajes tienen un objeto parecido, el cual es el desarrollo de aplicaciones.

Veamos algunas similitudes entre los dos lenguajes:

- Presentan un lenguaje enriquecido de instrucciones que hacen a los algoritmos más complejos sencillos de expresarlos.
- Presentan consistencia y falta de ambigüedad en sus instrucciones.
- Presentan un correcto control sobre los valores multidimensional; así tenemos los arreglos, estructuras y las clases.
- Presentan un desarrollo de aplicaciones basado en la programación modular, permitiendo que la programación se desarrolle en componentes independientes que trabajan bajo un mismo fin.
- Presentan un soporte de interacción con el entorno, es decir, proporcionan entrada y salida a sus aplicaciones.
- Presentan portabilidad de sus aplicaciones haciendo que estas puedan ejecutarse en cualquier plataforma de trabajo.

Ahora veamos algunas características que diferencian a Java de C++:

- En C++ la potencia de sus aplicaciones es el uso adecuado de los punteros que permiten acceder a la memoria volátil de la computadora (RAM) que en muchas ocasiones presentaba errores de colisión en la memoria. Java no presenta esta característica, por tanto evita todo acceso a la memoria volátil. El JDK de Java proporciona métodos de acceso a la memoria de manera efectiva y controlable.
- Para mantener un valor entre procesos C++ usa variables globales, mientras que en Java lo único global es el nombre de sus clases y las variables que necesite hacerlas globales se las declara dentro de la clase, pero al exterior de sus métodos.
- Java no usa la sentencia GoTo, en vez de ello usa *break* o *continue* para cortar o continuar alguna instrucción respectivamente. Goto hace que el código sea como un “código spaghetti” lleno de saltos, vueltas y revueltas. En Java se espera un orden sin saltos, en cambio; GoTo es como un canguro pegando saltos.



Fig. 1.1

- Las conversiones inseguras de C++ se realizan por medio de los moldeados de tipo (*type Casting*), el cual permite cambiar el tipo de un puntero. Java hace una comparación en tiempo de ejecución de la compatibilidad dejando al programador controlar dichas excepciones.

Finalmente, veamos un cuadro comparativo entre Java y C++.

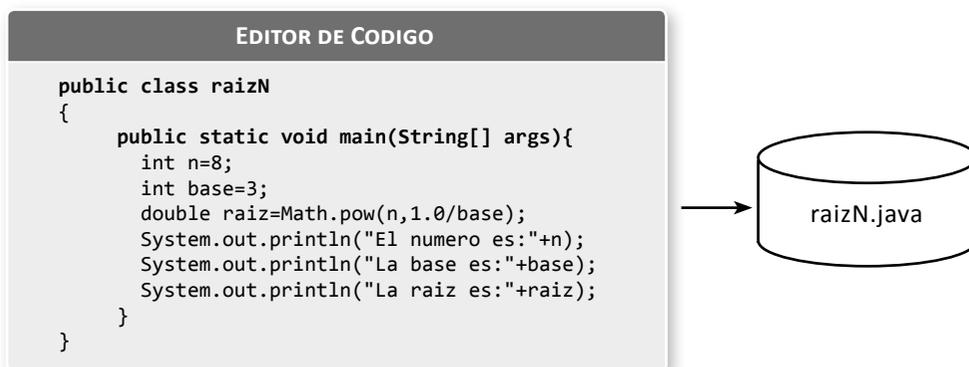
JAVA	C++
Es totalmente portable.	Parcialmente portable
Controla la memoria dinámica, el <i>garbage collection</i> y no tiene punteros.	Memoria automática gestionada por le RTS, memoria dinámica gestionada por el programador.
Administración de métodos virtuales.	Administración parcial de los métodos virtuales y no virtuales.
No cuenta con herencia múltiple.	Aplica la herencia múltiple.
Control de clases en tiempo de ejecución.	Es muy limitada.
Tratamiento genérico de tipos (Wrappers).	Manejo de punteros.

1.3. LOS PROGRAMAS EN JAVA

El proceso de programación en Java consta de un conjunto de actividades necesarias para la solución de un problema particular. Programar no es sencillo, hay que tener un control especial de lo que entendió del problema con lo que resultará en un programa Java.

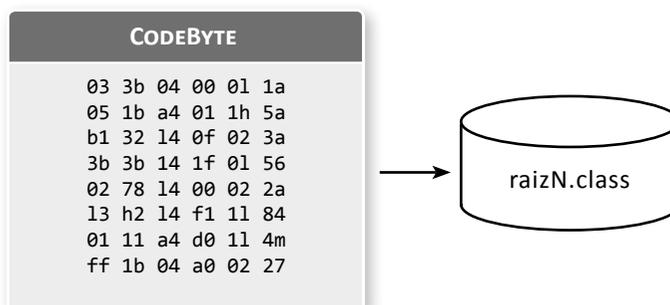
Normalmente son 5 las etapas por la que un programa en Java debe pasar, estas son: Edición, Compilación, Carga, Verificación y Ejecución; de estos 5 mínimamente podríamos nombrar a Edición, Compilación y Ejecución.

1. **Edición:** en esta etapa el programador digita las instrucciones Java en un editor en el cual podrá corregir alguna parte del código si fuera necesario o grabar el archivo cuando determine que el código es el correcto, cuando esto suceda se creará el archivo con extensión .java.



Para el siguiente caso hemos editado un código Java de consola que permite calcular la raíz N de un número entero positivo. La clase principal se llama raizN y tiene un método *void main* que permitirá operar las instrucciones. Al final se guardó el archivo en una unidad de la computadora con el mismo nombre de la clase y con la extensión .java.

2. **Compilación:** en esta etapa de transición se crean códigos de bytes y al ser guardados se crea el archivo con el mismo nombre del archivo Java pero con extensión .class.



A partir de la compilación del archivo Java y creado el archivo .class puede ser incrustado en una página web como un *applet*.

3. **Carga:** en esta etapa los códigos de bytes generados son enviados a la memoria de la computadora por medio del cargador de clases obtenidos desde la unidad donde se guardaron los archivos java y class.



4. **Verificación:** en esta etapa se verifica que el código de bytes sea válido y no contenga violaciones de seguridad Java, enviando el código a un intérprete que tendrá por misión hacer que este código sea entendible por la computadora.
5. **Ejecución:** en esta etapa el código de bytes depurado es enviado a la Máquina Virtual de Java (JVM) para su ejecución y visualización en el entorno que el usuario crea conveniente de aquí se desprende el término de portabilidad ya que dependiendo de la JVM podrá ejecutar la aplicación en una u otra plataforma.

1.4. EVOLUCIÓN DE LA PLATAFORMA JAVA

La evolución de Java es controlada por la Comunidad de Procesos Java (JCP) que usa los JSR's, la cual propone y especifica los cambios dentro de la plataforma Java.

- JDK 1.0 (Enero 1996)
Contiene 8 paquetes y 212 clases
- JDK 1.1 (Febrero 1997): es la versión más estable y robusta.
Contiene 2 paquetes y 504 clases
Mejor rendimiento en la JVM
Paquete de evento AWT
Clases anidadas
Serialización de objetos
API JavaBeans
Archivo JAR
JDBC Java Data Base Connectivity
RMI Remote Method Invocation
- J2SE 1.2 (Diciembre 1998): también llamada Java 2
Contiene 59 paquetes y 1520 clases
JFC Clases Swing
Java 2D
API Collections
- J2SE 1.3 (Mayo 2000)
Contiene 77 paquetes y 1595 clases
JNDI Java Naming and Directory Interface

API Java Sound
JIT Just in Time
Orientada a la resolución de errores

- J2SE 1.4 (Febrero 2002)

Contiene 103 paquetes y 2175 clases
API de entrada y salida de bajo nivel
Clases collection
Procesador XML
Soporte a criptografía con Java Cryptography Extension (JCE)
Inclusión de Java Secure Socket Extensio JSSE
Inclusión de Java Authentication and Authorization Service (JAAS)

- J2SE 5.0 (Java 5)

Contiene 131 paquetes y 2656 clases
Tipos genéricos
Autoboxing – Unboxing
Enumeraciones
Metadatos en clases y métodos

- Java SE 6 (Diciembre 2006)

Collection Framework
Drag and Drop
Internationalization Support
I/O Support
JAR

- Java Web Start

Java DB 10.2 JDBC 4 Early Access
Java Management Extensions (JMX)
Java Platform Debugger Architecture (JPDA)
Java Virtual Machine Tool Interface (JVM TI)
Reflection
Remote Method Invocation (RMI)
Swing

- Java SE7 (Julio 2011)

Soporte XML dentro del mismo lenguaje
Soporte para closures
Superpaquete

1.5. TIPOS DE APLICACIONES JAVA

El código Java no tiene distinción cuando se trata de las aplicaciones que desarrolla, veamos algunos tipos:

- Aplicaciones Java de Consola
- Aplicaciones GUI
- Aplicaciones Java Applets
- Aplicaciones Servlets
- Aplicaciones JavaBeans

Aplicaciones Java de Consola

Son aplicaciones que permiten concentrarse netamente en la implementación del problema y no en su entorno de desarrollo. Se caracterizan por:

- No permitir el uso del mouse
- Trabajar la ejecución en modo D.O.S.
- No contar con elementos visuales

En la Fig. 1.2. se muestra la ejecución de una aplicación de consola.



Fig. 1.2

Aplicaciones GUI

Son aplicaciones que no necesitan de un navegador ya que se ejecutan como una aplicación de escritorio. La mayoría de las aplicaciones Java están implementadas por una Interfaz Gráfica de Usuario (GUI), por su no exposición de los datos en la web y ser una aplicación interna dentro de una organización.



Fig. 1.3

Aplicaciones Java Applets

Estas aplicaciones se caracterizan por ser pequeñas aplicaciones que se incorporan en una página web y que necesitan de un navegador compatible con Java, en realidad todos los navegadores son compatibles, pero se tiene que descargar un Plugin según el tipo de navegador. Los applets se ejecutan siempre en el lado cliente ya que son descargados desde un servidor web conjuntamente con la página HTML.

Los applets hacen referencia al archivo compilado .class, si tenemos una aplicación compilada llamada raizN.class entonces el código HTML sería de la siguiente manera:

```
<html>
<head>
  <title>Aplicacion de Prueba</title>
</head>
<body>
  Raiz N
  <APPLET>
    CODE="raizN.class"
    WIDTH=200
    HEIGHT=70
  </APPLET>
</body>
</html>
```

Aplicaciones Servlets

Son objetos de Java que se ejecutan dentro de un servidor web, como por ejemplo TomCat que los reutiliza, recibe peticiones mediante una URL y genera respuestas al Cliente. Genera páginas web dinámicas a partir de los valores que sean enviados al servidor como se muestra en la Fig.1.4.

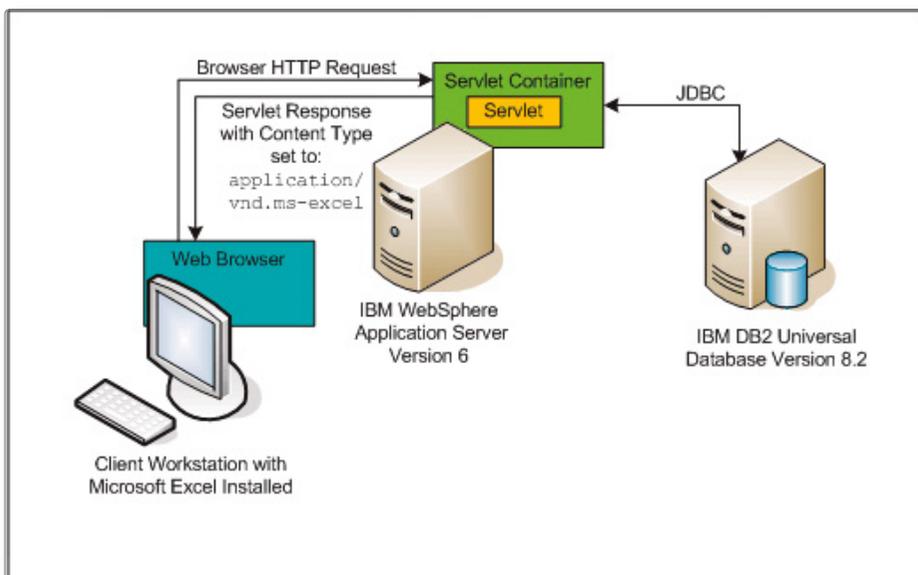


Fig. 1.4

Aplicaciones JavaBeans

Es un modelo de componentes que se usa para encapsular varios objetos en otro único. Se define como un componente de software reutilizable que se puede administrar desde una herramienta de construcción, un ejemplo de código JavaBean es el que sigue:

```
public class VendedorBean
implements java.io.Serializable {

    private String vendedor;
    private double monto;

    public VendedorBean() {}

    public void setVendedor(String ven) {
        vendedor = ven;
    }

    public void setMonto(double mon) {
        monto = mon;
    }

    public String getVendedor() {
        return vendedor;
    }

    public int getEdad() {
        return monto;
    }
}
```

1.6. REQUISITOS PARA LA INSTALACIÓN DE JAVA

Los requisitos mínimos para la instalación de Java son los siguientes:

REQUERIMIENTO	ESPECIFICACIÓN
Sistema Operativo	Windows 8, 7, Vista SP2, XP S3 Windows Server 2008, 2012
Memoria RAM	128MB 64MB
Espacio de Disco	124MB
Navegador Web	IE 7.0 o superior Firefox 12 o superior Chrome

1.6.1. Instalando Java

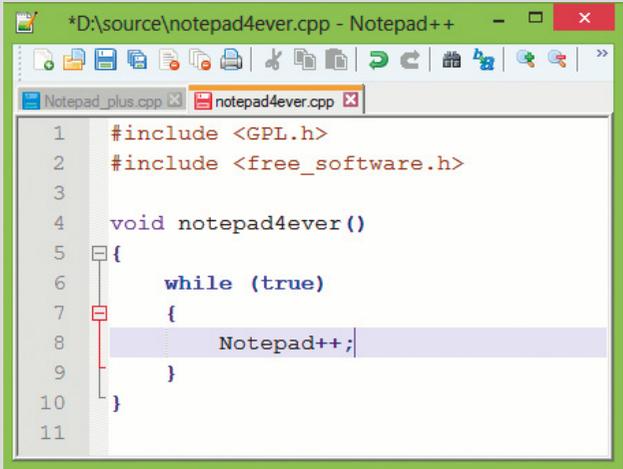
Java SE7 es la versión más actualizada de Java, la cual contiene muchas características nuevas dentro del lenguaje. Veamos algunas:

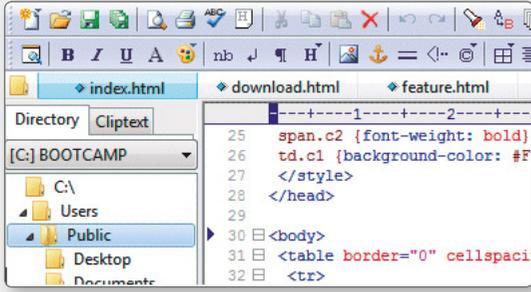
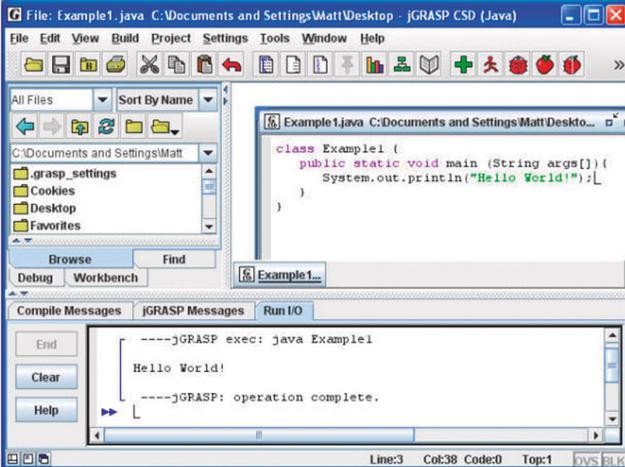
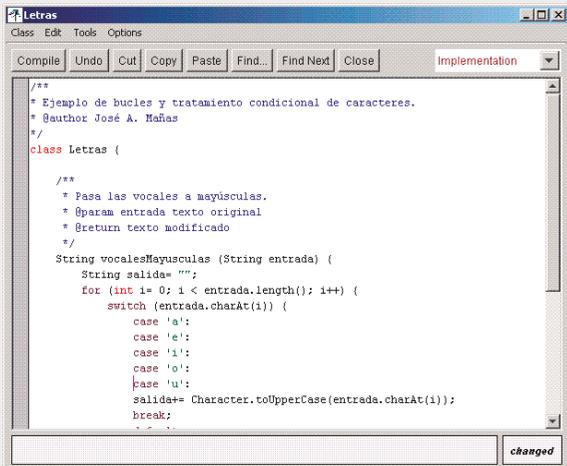
- Mejoras en el rendimiento, estabilidad y especialmente la seguridad.
- Mejoras en los Plugins Java para el desarrollo de aplicaciones para la web.
- Mejoras en la edición y optimización del código Java, que permiten un trabajo efectivo de los desarrolladores.
- Mejora en la JVM de Java ofreciendo soporte para diferentes lenguajes no necesariamente Java.

Debemos tener en cuenta que para ejecutar aplicaciones Java en una computadora personal se necesitan de las siguientes aplicaciones:

- Editor del Lenguaje
- Compilador Java
- Máquina Virtual de Java
- Librerías

Editor del Lenguaje: son aplicaciones que permiten editar un código Java, es decir, solo podrá digitar el código pero no podrá compilarlo ni ejecutarlo. Podríamos nombrar las siguientes aplicaciones:

APLICACIÓN	UBICACIÓN O DESCARGA
Bloc de Notas NotePad	En todas las versiones de Windows
NotePad++	<p data-bbox="431 1201 682 1226">http://notepad-plus-plus.org/</p>  <pre data-bbox="508 1372 935 1696"> 1 #include <GPL.h> 2 #include <free_software.h> 3 4 void notepad4ever() 5 { 6 while (true) 7 { 8 Notepad++; 9 } 10 } 11 </pre>

APLICACIÓN	UBICACIÓN O DESCARGA
EditPlus	<p data-bbox="458 236 677 258">http://www.editplus.com/</p> 
JGrasp	<p data-bbox="458 626 758 649">http://www.eng.auburn.edu/grasp/</p> 
BlueJ	<p data-bbox="458 1197 857 1220">http://www.bluej.org/download/download.html</p> 

Compilador Java

Como ya habíamos comentado líneas arriba, los compiladores generan un archivo .Class que es enviado a la memoria de la computadora en código de Bytes llamado ByteCode.

Máquina Virtual de Java (JVM)

La ejecución de las aplicaciones implementadas para Java se llevan a cabo gracias a la Máquina Virtual de Java garantizando así la portabilidad de las mismas, esta define una computadora lógica que ejecutará las instrucciones Bytecode y su intérprete ejecutará las instrucciones generadas dentro del archivo .class. Veamos algunas características de la JVM:

- Administra de manera correcta la memoria no usada gracias a su *Garbage Collector* o recolector de desechos.
- Reserva espacios en la memoria principal de la computadora para aquellos objetos que se crean en la aplicación, estos usan el operador new.
- Es la encargada de asignar variables a los registros y pilas.
- Administración de funciones del sistema huésped para el acceso a los dispositivos de la computadora.
- Administra la seguridad en las aplicaciones Java.
- No gestiona punteros, en su lugar crea objetos de clases.
- Administra de manera correcta los Casting entre los diferentes tipos de datos.

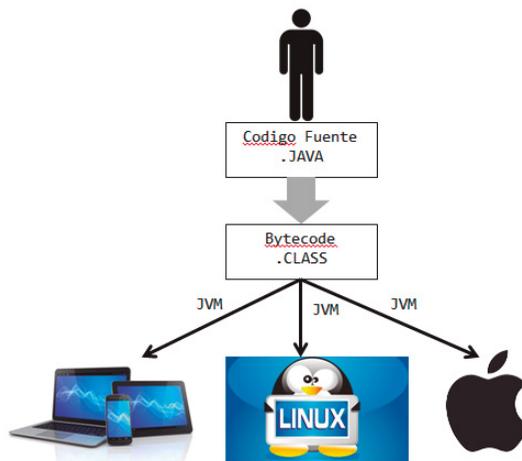


Fig. 1.5

Como se muestra en la Fig. 1.5 el código es compilado y enviado a los diferentes entornos por medio de la JVM.

Librerías

Es un apoyo al desarrollo de las aplicaciones en Java ofreciendo métodos agrupados que permiten agilizar el desarrollo y relacionarse con sistemas externos como las API's de acceso a datos o las bibliotecas de Interfaz de Usuario como la AWT y la Swing y API's para la captura, procesamiento y reproducción de audio en Java.

Ahora se estará preguntando, ¿cuánto software descargaré para poder tener el Java en mi computadora? Pues la respuesta se puede resumir a solo dos aplicaciones:

APLICACIÓN	CONTENIDO
JDK 7 (Java Development Kit)	<ul style="list-style-type: none"> • Compilador Java • Máquina Virtual de Java • Librerías
IDE (Entorno de Desarrollo Integrado)	<ul style="list-style-type: none"> • Netbeans • JCreator • JDeveloper • Eclipse

1.7. EL JDK 7 Y SU INSTALACIÓN

El JDK es un Kit que contiene el compilador de Java, la JVM y las librerías necesarias para generar una aplicación Java. Para descargar el JDK 7 ingrese a la siguiente URL:

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

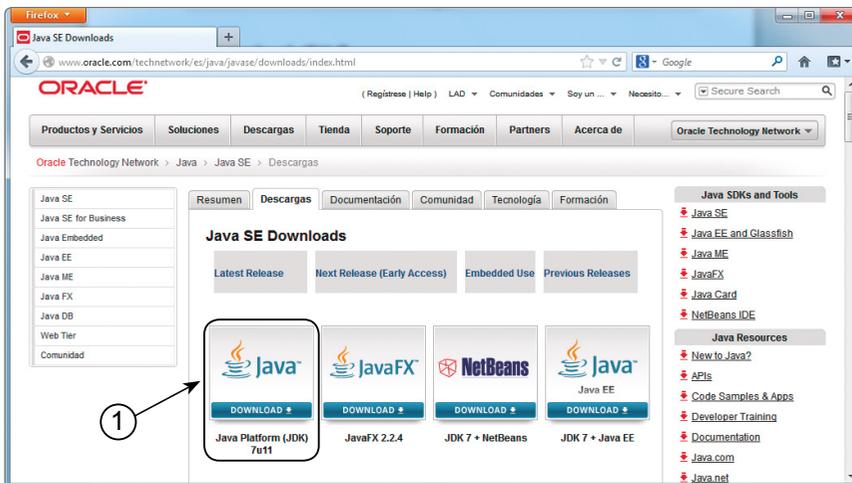


Fig. 1.6

En la Fig. 1.6 se muestra la sección de descargas de Java desde la página oficial de Oracle; como notará Java ya no pertenece a la compañía Sun. Para descargar el JDK seleccionaremos Download del Java Platform (JDK) como lo muestra la Fig.1.7



Fig. 1.7

Seguidamente debe seleccionar:

- Accept License Agreement
- Windows x86 – 88.77 MB – jdk-7u11-windows-i586.exe esto se debe a que la versión de Windows que se utilizó para este material es Windows 7 de 32 Bits. Como lo muestra la Fig. 1.8.

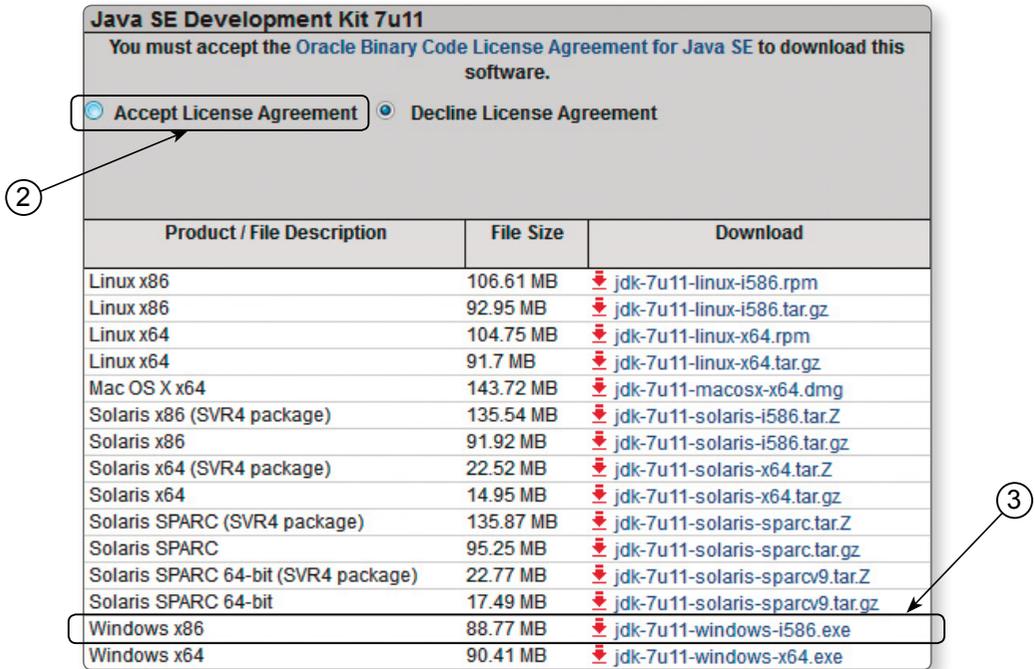


Fig. 1.8

Luego debe guardar el archivo en su Unidad de Disco estable para poder instalarlo desde allí como lo muestra la Fig. 1.9:

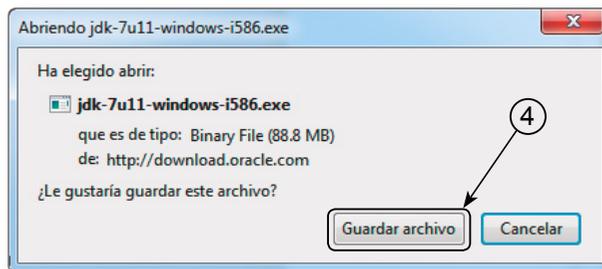


Fig. 1.9

Una vez finalizada la descarga del archivo JDK se procederá con la instalación, en la Fig. 1.10 presione Next>.

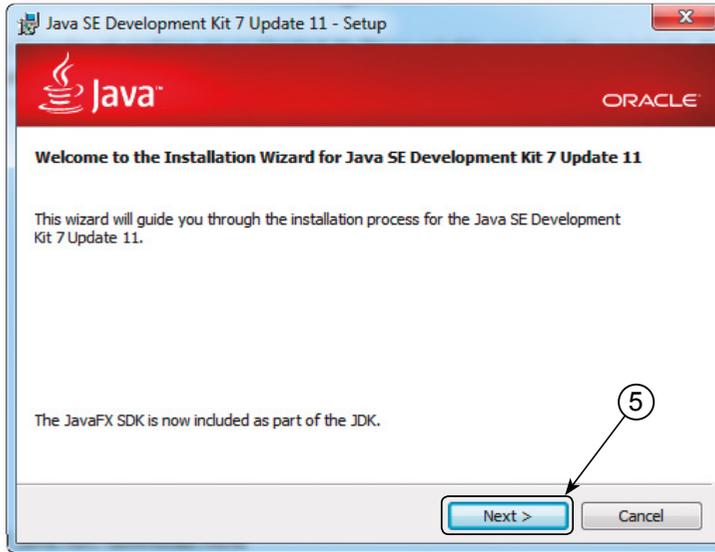


Fig. 1.10

En la Fig. 1.11 se muestra los objetos que se instalarán, si lo desea aquí puede cambiar la ubicación de la instalación presionando el botón Change..., luego presione Next>.

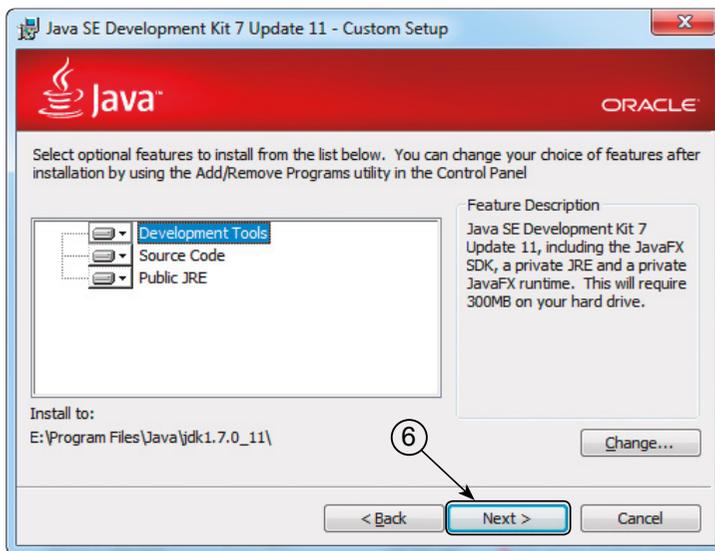


Fig. 1.11

En la Fig. 1.12 se muestra el avance de la instalación, en esta ventana no se selecciona ningún elemento.

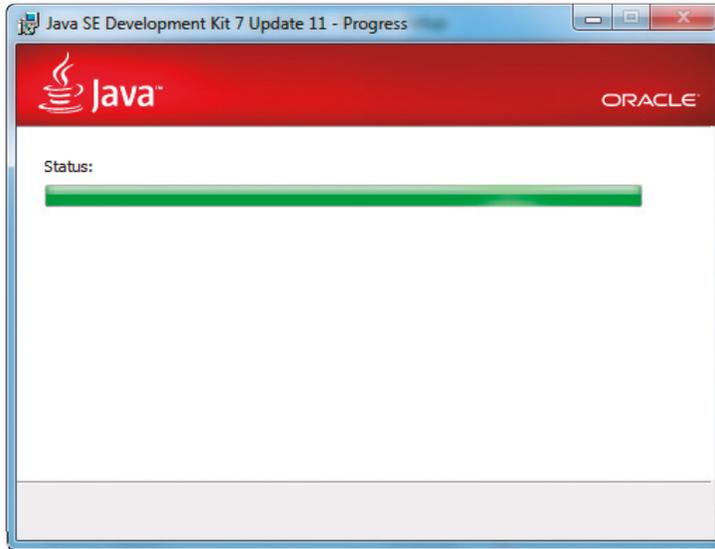


Fig. 1.12

En la Fig. 1.13 se muestra la ventana de selección de destino de los archivos JRE7 para lo cual no es necesario cambiarlos, pero si lo desea puede seleccionar Change, pero recuerde que no es necesario cambiarlos.

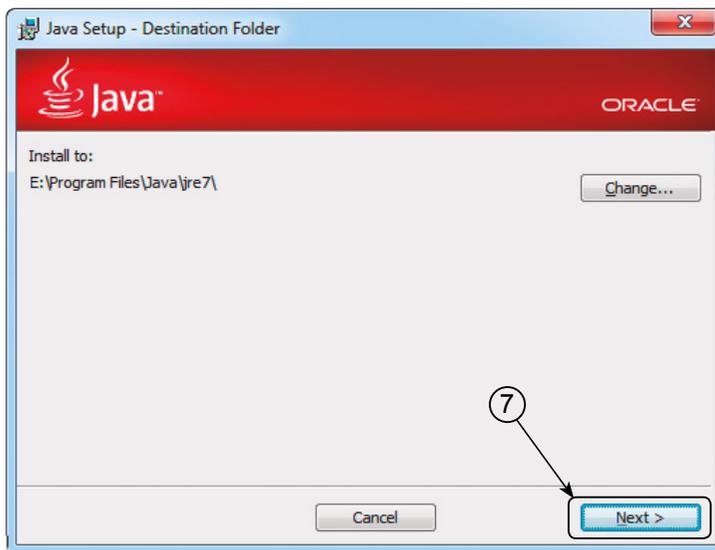


Fig. 1.13

En la Fig. 1.14 se muestra el proceso de instalación:



Fig. 1.14

Finalmente, si todo es correcto se debe mostrar la ventana que muestra la Fig. 1.15. Para finalizar la instalación presione Close.



Fig. 1.15

También se puede descargar el JDK de la siguiente URL:

<http://jdk7.java.net/download.html>

Tal como lo muestra la Fig. 1.16 primero debe seleccionar Accept License Agreement, desde la plataforma Windows seleccione exe 88.94MB y proceda con la instalación.

The screenshot shows the Java Platform, Standard Edition 7 Update 12 Developer Preview Releases page. The page title is "Java™ Platform, Standard Edition 7 Update 12 Developer Preview Releases" dated January 03, 2013. The main heading is "7u12 Build b08". Below this, it states "Java SE 7 Update 8 (7u8) is being renumbered to Java SE 7 Update 12 (7u12)." and provides a link for details. There is a "Please note" section and a "License Agreement" section where the "Accept License Agreement" radio button is selected. A table lists download links for various platforms and architectures.

Platforms	JRE	JDK
Windows	32-bit exe (md5) 30.13 MB	exe (md5) 88.94 MB
	64-bit exe (md5) 31.62 MB	exe (md5) 90.64 MB
Mac OS X	64-bit dmg(md5) 50.51 MB	dmg(md5) 144.09 MB
	32-bit tar.gz (md5) 46.10 MB	tar.gz (md5) 93.23 MB
Linux	64-bit tar.gz (md5) 44.85 MB	tar.gz (md5) 91.98 MB
	Linux ARM) tar.gz(md5) 67.47 MB

Fig. 1.16

1.8. EL IDE JCREATOR Y SU INSTALACIÓN

El JCreator es una aplicación que permite crear aplicaciones con Java bajo el entorno Windows. La compañía que lo produce es Xinox Software, la cual ofrece dos versiones del producto: la gratuita llamada JCreator LE y la licenciada llamada JCreator Pro; esta última incorpora comandos, plantillas y otros complementos que la versión libre no los tiene, esto no implica que no pueda desarrollar aplicaciones Java.

Para descargar las versiones de JCreator visite la siguiente URL:

<http://www.jcreator.org/download.htm>

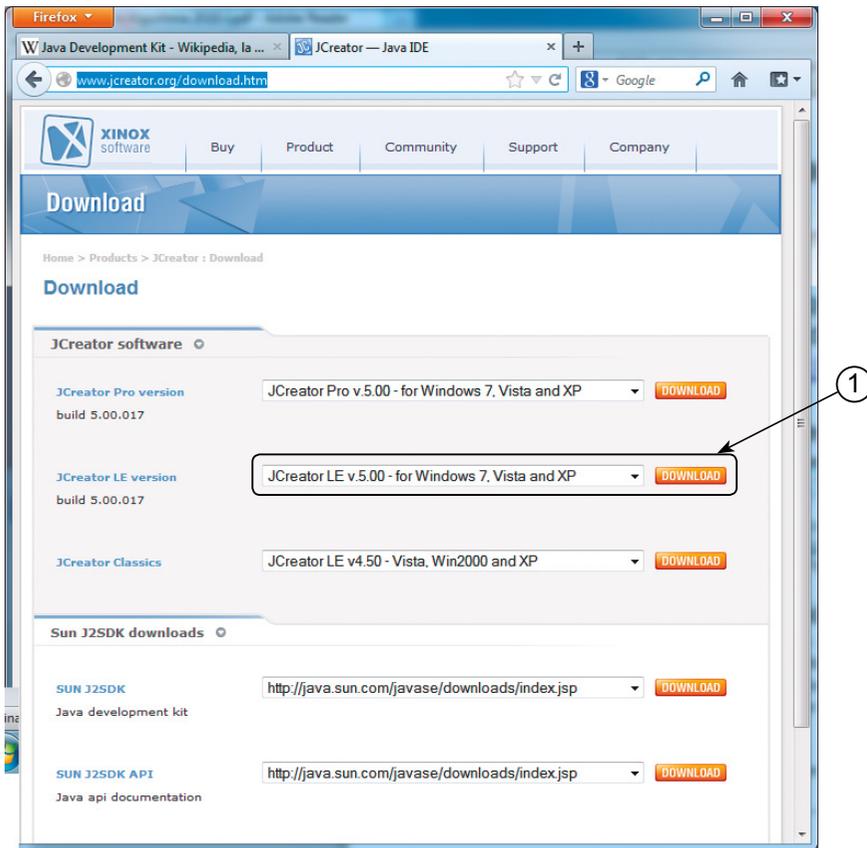


Fig. 1.17

Una vez descargado procederemos a instalarlo, presionando Next> desde la Fig. 1.18.



Fig. 1.18

Luego aceptamos en contrato como lo muestra la Fig. 1.19

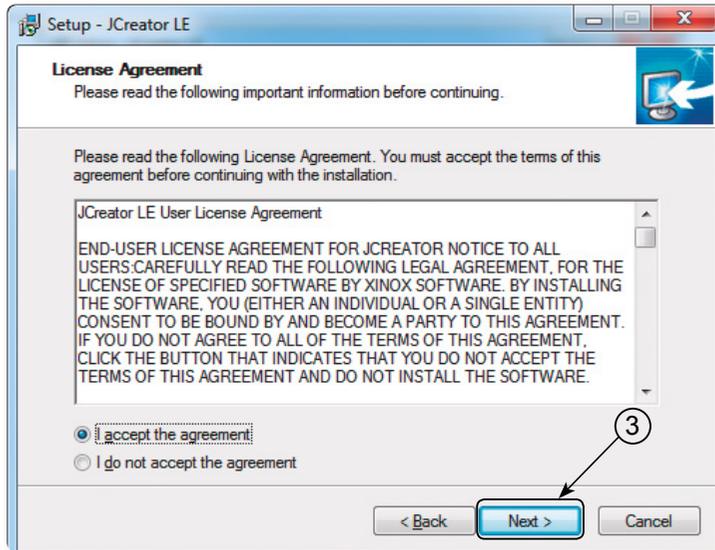


Fig. 1.19

Seguidamente presionamos Next, a menos que desee cambiar la ubicación de los archivos de instalación para esto presione Browse..., finalmente presione Next>.

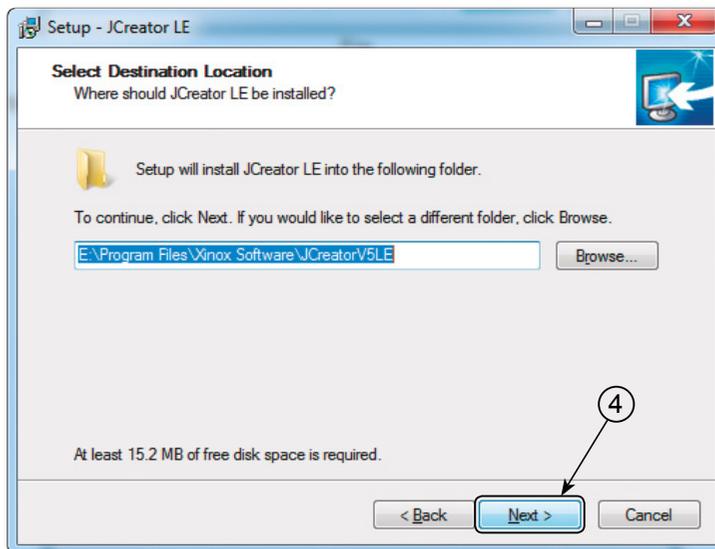


Fig. 1.20

Siga los pasos del asistente de instalación hasta llegar a la ventana mostrada en la Fig. 1.21, presione Finish para empezar a usar JCreator.



Fig. 1.21

Por ser la primera vez que se instala este software se muestra la siguiente ventana:

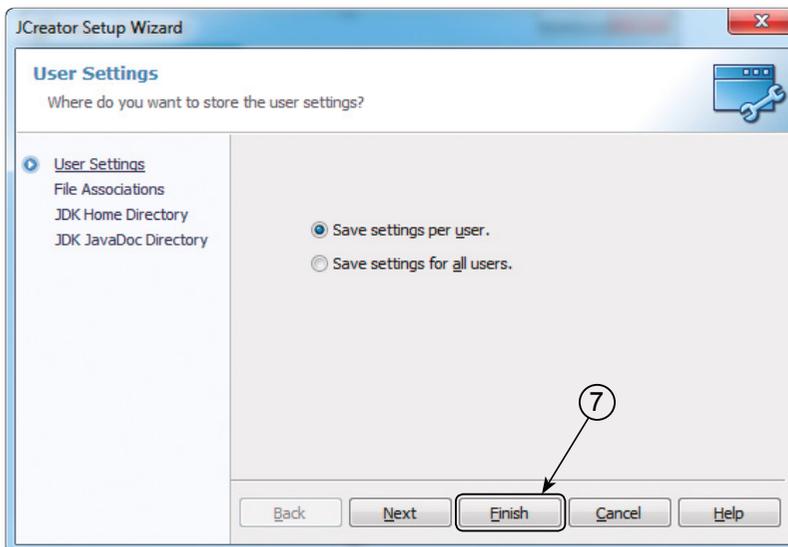


Fig. 1.22

Para lo cual solo presione Finish y se mostrará la pantalla inicial del JCreator como lo muestra la Fig. 1.23.

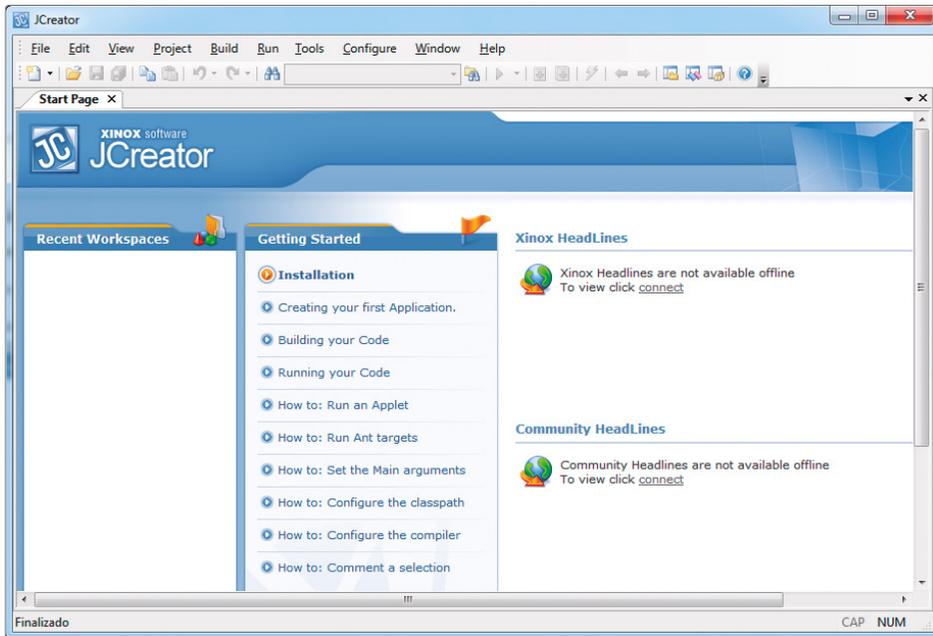


Fig. 1.23

1.9. EL IDE NETBEANS Y SU INSTALACIÓN

NetBeans IDE es una aplicación de código abierto diseñada para el desarrollo de aplicaciones Java entre las distintas plataformas.

Con NetBeans se puede:

- Crear interfaces gráficas de forma visual.
- Desarrollar aplicaciones web.
- Crear aplicaciones compatibles con teléfonos móviles.

Para descargar la version 7.2 de NetBeans visite la siguiente URL:

<http://netbeans.org/>



Fig. 1.23

Se descargará la versión más actualizada hasta el momento de la edición de este material, seleccione el idioma del IDE (en este caso es English), la plataforma de trabajo Windows y finalmente presione Download Free 240MB para tener toda la tecnología de NetBeans completa (ver fig. 1.24).

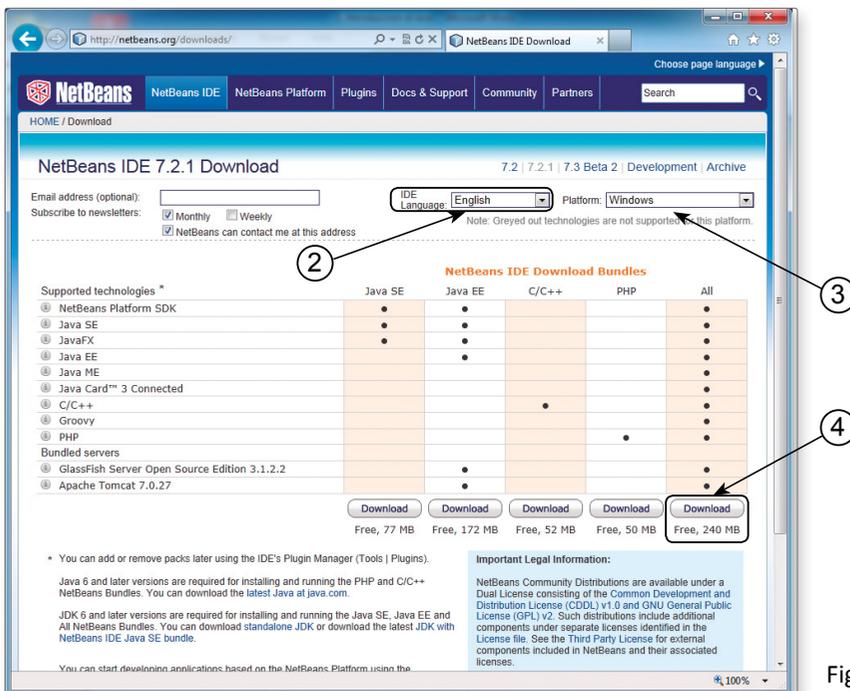


Fig. 1.24

Una vez descargado el IDE NetBeans se procederá a instalarlo de la siguiente manera:

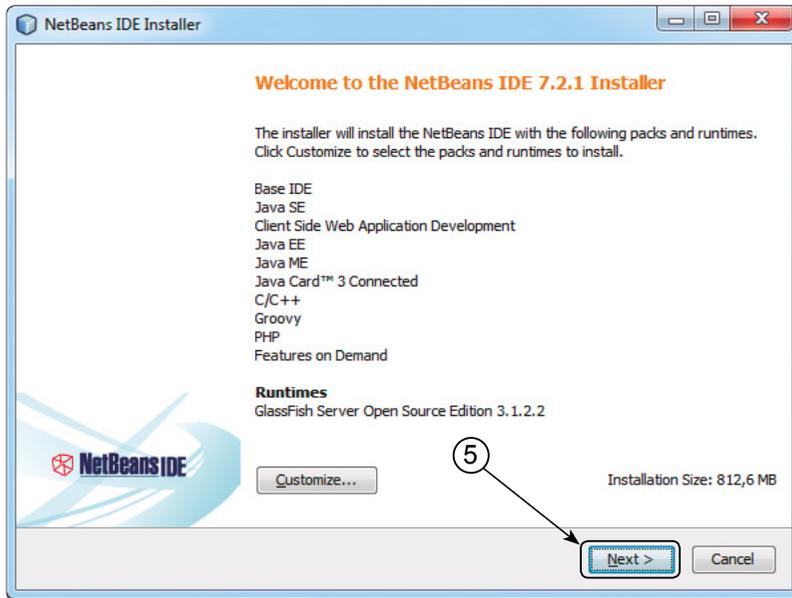


Fig. 1.25

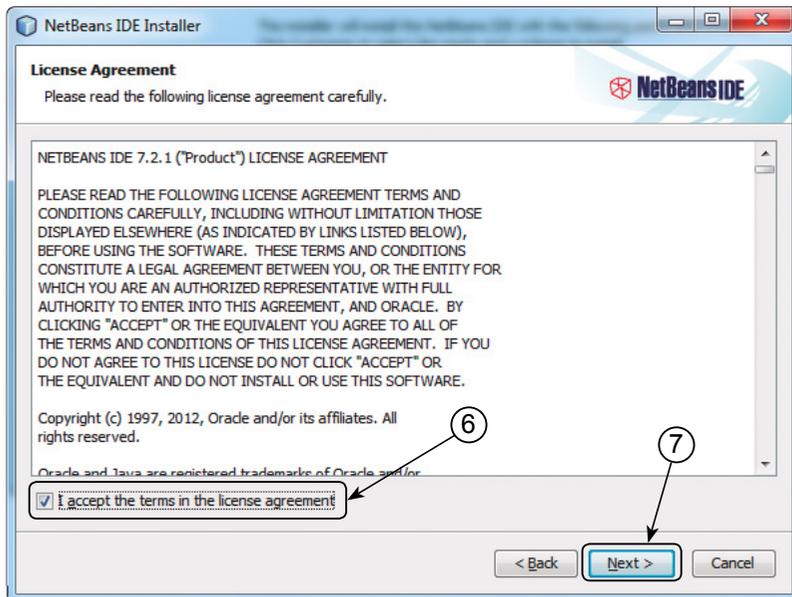


Fig. 1.26

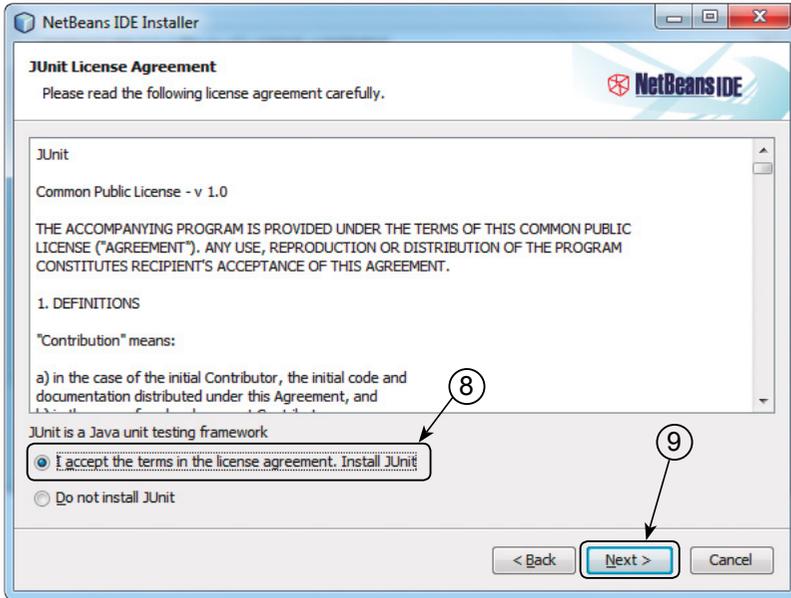


Fig. 1.27

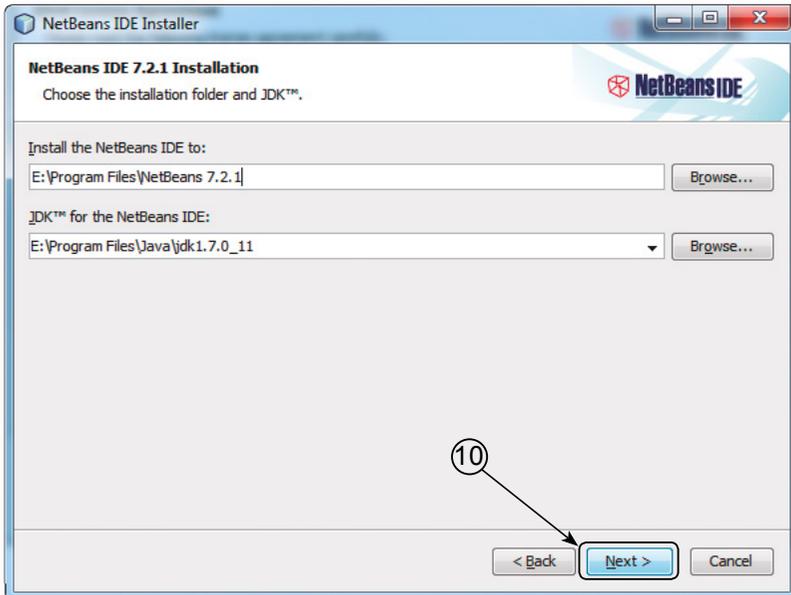


Fig. 1.28

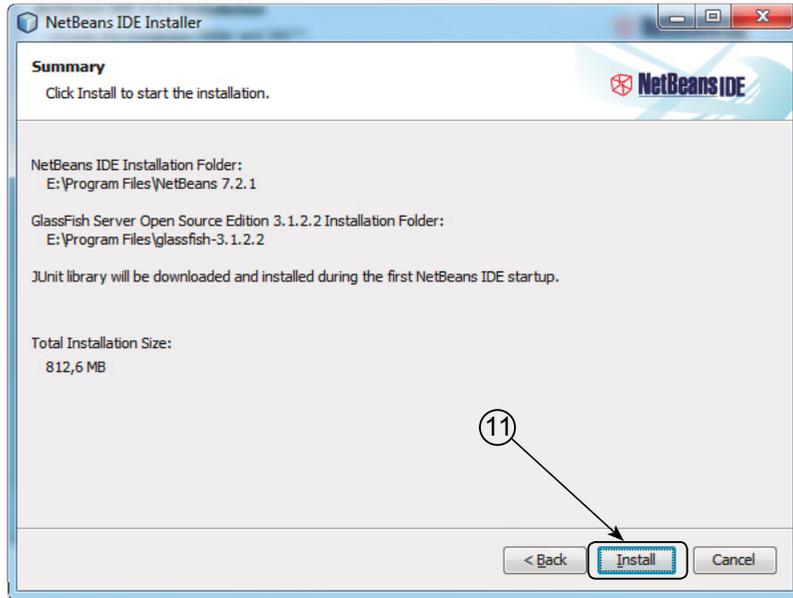


Fig. 1.29

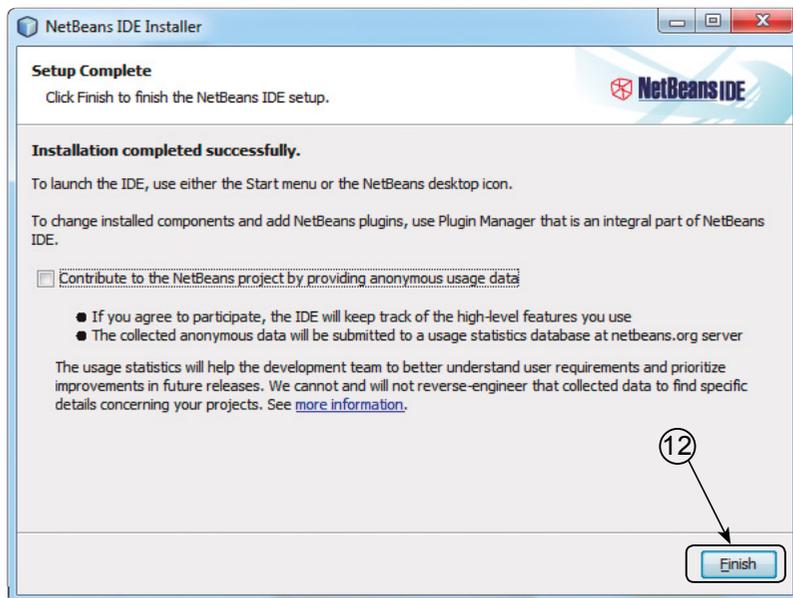


Fig. 1.30

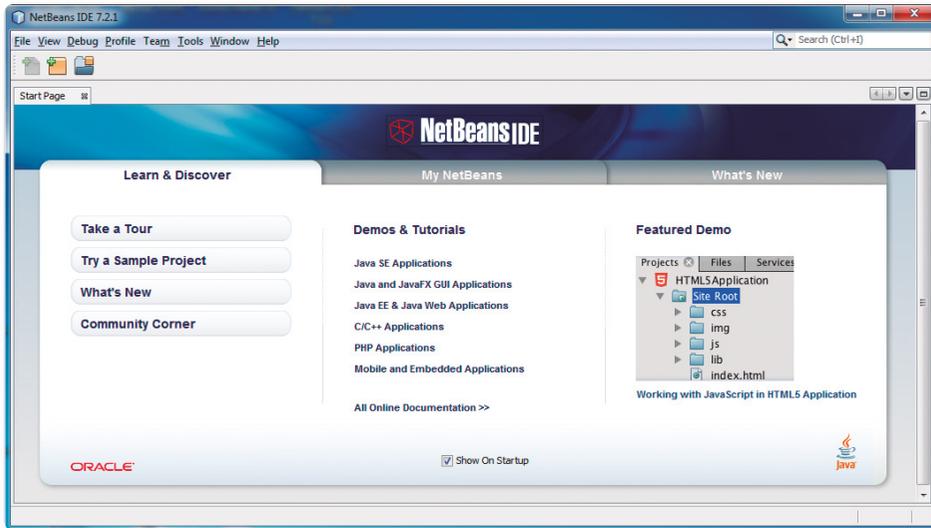


Fig. 1.31

1.10. EL IDE JDEVELOPER 11G Y SU INSTALACIÓN

JDeveloper es un entorno de desarrollo integrado, desarrollado por la compañía Oracle Corporation para los lenguajes Java, HTML, XML, SQL, PL/SQL, Javascript, PHP, Oracle ADF, UML y otros. Es un software propietario pero gratuito desde el 2005.

Con JDeveloper se puede realizar lo siguiente:

- Crear interfaces gráficas de forma visual.
- Desarrollar aplicaciones web.
- Crear aplicaciones compatibles con teléfonos móviles.

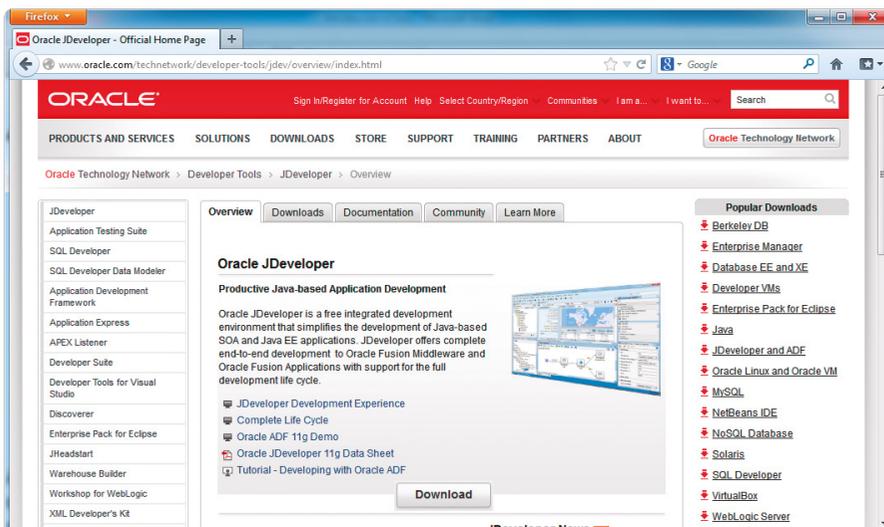


Fig. 1.32

Release Downloads for Oracle JDeveloper 11g (11.1.2.3.0)

This page consolidates all download links for Oracle JDeveloper. Visit the [Installation Guide for Oracle JDeveloper](#) for an overview of the installation process and the [Oracle JDeveloper Certification Information](#) for platform specific information.

Important Note - This version of JDeveloper doesn't include the SOA and WebCenter pieces - to use these components you'll need to [download Oracle JDeveloper 11.1.1.6.0](#).

The downloads below are provided for customers under the [OTN JDeveloper License Agreement](#). Current customers should download their software via our [Oracle Software Delivery Cloud](#), which offers different license terms.

Accept License Agreement Decline License Agreement

Oracle JDeveloper 11g (11.1.2.3.0) (Build 6276.1) Installations

This is the release of Oracle JDeveloper 11g (11.1.2.3.0) (Build 6276.1). See the [Documentation](#) tab for Release Notes, Installation Guides and other release specific information. You can also view the [List of New Features and Samples](#) provided for this release.

Studio Edition: 11.1.2.3.0

Windows Install (Size: 1.2 GB)

This download is the complete version of JDeveloper with all the features. This is the recommended Download.

+ Prerequisites & Recommended Install Process

Java Edition: 11.1.2.3.0

Generic Install (Size 112 MB)

This download contains only the core Java and XML features, it doesn't contain J2EE, ADF, UML and Database features. Fewer features means smaller download and improved performance.

+ Prerequisites & Recommended Install Process

Oracle Application Development Runtime Installer

This installer enables you to extend your Java EE server with the libraries needed to run Oracle ADF. [Download](#)

Fig. 1.33



Fig. 1.34

Fig. 1.36

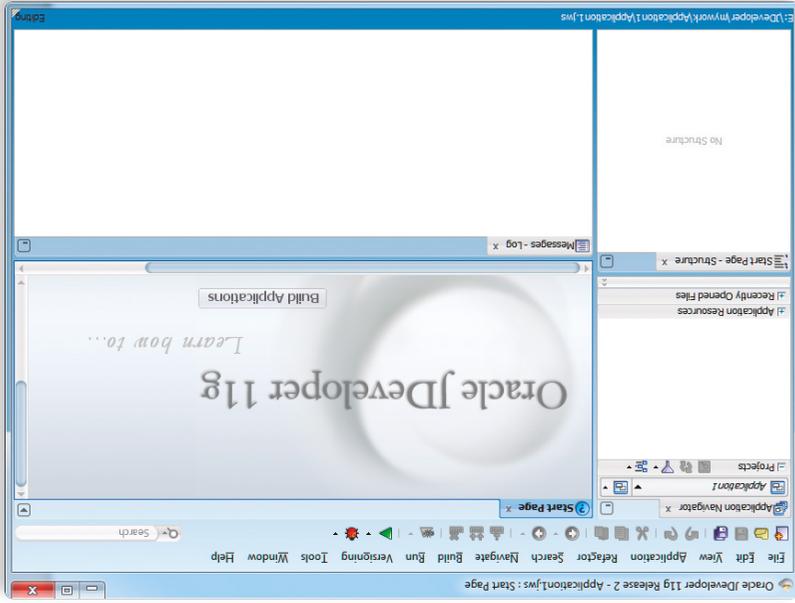
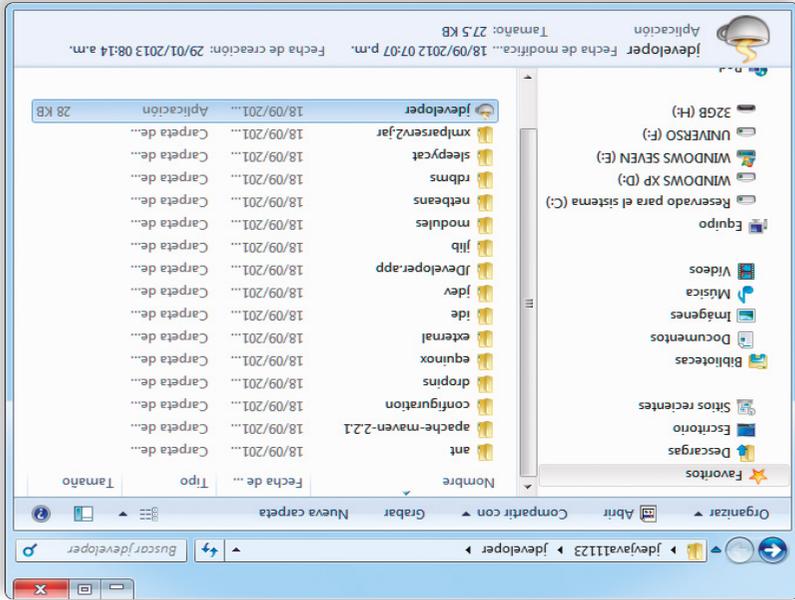


Fig. 1.35



Impreso en los Talleres Gráficos de



Surquillo

☎ 719-9700