



JAVA Básico

Luis Felipe Wanumen Silva

Edwin Rivas Trujillo

Darin Jairo Mosquera Palacios

ECOE
EDICIONES

Square[0]);

+ moves[next].toString()

+ move);

Wanumen Silva, Luis Felipe

Java básico / Luis Felipe Wanumen Silva, Edwin Rivas Trujillo, Darin Jairo Mosquera Palacios. -- 1a. ed. - Bogotá : Ecoe Ediciones, 2017.

208 p. - (Ingeniería y salud en el trabajo. Informática)

“Incluye: Ejemplos para explicar fácilmente los tipos primitivos de datos y los tipos de objetos de datos. Desarrollo de interfaces reales, para el posterior desarrollo de aplicaciones. Ejemplos reales de manipulación de cadenas de caracteres. Ejemplos ejecutables y estructurados que involucran la programación orientada a objetos.” -- Contiene datos biográficos de los autores.

ISBN 978-958-771-541-5 -- 978-958-771-542-2 (e-book)

1. Java (Lenguaje de programación de computadores)

2. Programación (Computadores electrónicos) I. Rivas Trujillo, Edwin II. Mosquera Palacios, Darín Jairo III. Título IV. Serie

CDD: 005.275 ed. 23

CO-BoBN- a1007553



Colección: Ingeniería y salud en el trabajo

Área: Informática

ECOE
EDICIONES

© Luis Felipe Wanumen Silva

© Edwin Rivas Trujillo

© Darin Jairo Mosquera Palacios

© Ecoe Ediciones Ltda.

e-mail: info@ecoediciones.com

www.ecoediciones.com

Carrera 19 # 63C 32, Tel.: 248 14 49

Bogotá, Colombia

Primera edición: Bogotá, septiembre de 2017

ISBN: 978-958-771-541-5

e-ISBN: 978-958-771-542-2

Dirección editorial: Angélica García Reyes

Corrección de estilo: Laura Lobatón Sanabria

Diagramación: Olga L. Pedraza Rodríguez

Carátula: Wilson Marulanda Muñoz

Impresión: La Imprenta Editores

Calle 77 # 27 A - 39

*Prohibida la reproducción total o parcial por cualquier medio
sin la autorización escrita del titular de los derechos patrimoniales.*

Impreso y hecho en Colombia - Todos los derechos reservados

AGRADECIMIENTOS

De manera muy especial a Dios Padre Todopoderoso y a nuestros familiares, por su paciencia en las noches que trabajamos fuertemente en el desarrollo de este material. A nuestras esposas, a la Virgen María, a nuestros amigos que nos alentaron durante ese proceso. Aunque los agradecimientos a la Universidad Distrital Francisco José de Caldas se han dejado de últimas, no quiere decir que sea menos importante esta institución, si hay en esta tierra una Institución que patrocina la elaboración de documentos a sus profesores es la Universidad Distrital. Gracias a ella por habernos entregado los espacios en grupos de investigación y por haber dado el tiempo al interior de nuestros planes de trabajo para que este esfuerzo se hiciera realidad en un manual de bases de datos.

Luis Felipe Wanumen
Darin Jairo Mosquera
Edwin Rivas

CONTENIDO

PREFACIO	XV
INTRODUCCIÓN	XVII
CAPÍTULO 1. ALGUNOS ENTORNOS DE DESARROLLO EN JAVA.....	1
1.1 El verdadero compilador es el JDK.....	1
1.2 <i>J creator</i>	2
1.3 <i>RealJ</i>	6
CAPÍTULO 2. MANEJANDO EL COMPILADOR DE JAVA.....	7
2.1 Cómo crear aplicaciones <i>standalone</i> utilizando comandos de Java ...	7
2.2 Cómo crear aplicaciones <i>applet</i> utilizando comandos de Java.....	9
2.3 Tópicos fundamentales.....	10
CAPÍTULO 3. CONCEPTOS BÁSICOS	15
3.1 Librerías, métodos, clases y funciones que más se van a utilizar en el curso	15
3.2 Diferencias entre un applet y una aplicación <i>standalone</i>	20
3.3 Estructura básica de un programa en Java.....	21
3.4 Por qué utilizar Java	21
3.5 <i>Applets</i> en una página HTML.....	22
3.6 Estructura de los métodos o funciones más usados en una clase de Java.....	22
3.7 Funciones para manejar el ratón y el teclado en Java.....	24
3.8 Interfaz de la clase <i>applet</i> en Java	25
3.9 Palabras clave definidas en Java.....	26
3.10 Palabras reservadas	26
3.11 Literales	26
3.12 Imprimiendo un texto en un <i>applet</i> desde cualquier función.....	27

CAPÍTULO 4. MI PRIMERA EXPERIENCIA	29
4.1 Comentarios	29
4.2 Variables estáticas.....	30
4.3 Variables tipo CHAR	32
CAPÍTULO 5. VARIABLES Y ATRIBUTOS	37
5.1 Declaración de atributos	37
5.2 Acceso a los atributos: <i>private</i>, <i>protected</i> o <i>public</i>	37
5.3 Variables de tipo <i>static</i> y <i>final</i>	38
5.4 Atributos de tipo <i>transient</i> y <i>volatile</i>	38
CAPÍTULO 6. TIPOS DE DATOS.....	39
6.1 Tamaño de los tipos de datos.....	39
6.2 Variables y objetos de datos en Java.....	40
6.3 Tipos de datos en variables de Java.....	40
6.4 Diagrama explicativo sobre tipos y objetos de datos.....	42
6.5 Objeto de tipo <i>double</i>.....	42
6.6 Objeto de tipo <i>float</i>	43
6.7 Objeto de tipo <i>integer</i>	43
6.8 Utilizando la función <i>parseinteger</i> () de la clase <i>integer</i>	44
6.9 Objeto de tipo <i>long</i>.....	45
6.10 Objetos de tipo <i>string</i>.....	45
CAPÍTULO 7. ESTRUCTURAS DE CONTROL.....	47
7.1 La estructura de control <i>for</i>	47
7.1.1 El <i>for</i> con varias expresiones conectadas con <i>and</i>.....	49
7.1.2 El <i>for</i> con varias expresiones conectadas con <i>or</i>.....	51
7.1.3 El <i>for</i> inicializando únicamente la variable.....	52
7.1.4 El <i>for</i> estableciendo el incremento de la variable	53
7.1.5 El <i>for</i> estableciendo la condición de la variable.....	54
7.1.6 El <i>for</i> sin argumentos para los parámetros	55
7.1.7 Control general de flujo.....	55
7.1.8 Cláusulas <i>continue</i> y <i>break</i>	56
7.2 La estructura de control <i>if</i>.....	58
7.3 Manejo del ciclo <i>while</i> en Java.....	60
7.3.1 Equivalencia entre el <i>while</i> y el <i>for</i>	61
7.3.2 El <i>while</i> anidado	62

7.3.3 El <i>while</i> que incrementa y/o decrementa variable en el argumento.....	63
7.4 Excepciones.....	66
 CAPÍTULO 8. OPERACIONES Y ASIGNACIONES	67
8.1 Simplificación de operaciones en Java	68
8.2 Separadores	69
8.3 Haciendo <i>casting</i> o conversiones en las operaciones.....	69
8.4 Operaciones matemáticas	69
8.5 Operadores de comparación.....	70
 CAPÍTULO 9. PROFUNDIZANDO EN LAS CONVERSIONES	71
9.1 Conversión de tipos primitivos.....	71
9.2 Conversión de cadenas en tipos primitivos.....	72
9.3 Conversión de tipos primitivos a cadenas	72
 CAPÍTULO 10. NUESTROS PRIMEROS PROGRAMAS SENCILLOS	75
10.1 Capturando un dato y desplegándolo en un <i>applet</i>	75
10.2 Capturando un dato y desplegándolo en un autónomo	77
10.3 Muestra una caja de diálogo	78
10.4 Programa que saluda al usuario	78
10.5 Calculando la raíz cuadrada de un número en un <i>applet</i>	79
 CAPÍTULO 11. PROGRAMAS AUTÓNOMOS	83
11.1 Estructura de los programas autónomos	83
11.2 Centrando una ventana y colocándole título	84
11.3 El emperador y los indios	85
11.4 Ejemplo de múltiplos en un intervalo	98
11.5 Programa de las Torres de Hanói	103
 CAPÍTULO 12. PROGRAMAS ESPECIALES	109
12.1 Programa que maneja números aleatorios	109
12.2 Programa que muestra cómo hacer un menú	111
 CAPÍTULO 13. PROGRAMAS DE CONVERSIÓNDE NÚMEROS	115
13.1 Programa que convierte un número binario en un número decimal	115
13.2 Programa que convierte un número decimal en un número binario.....	119

CAPÍTULO 14. MANEJO DE EVENTOS.....	121
14.1 Ases 1: capturando eventos con <i>ActionListener</i>	123
14.2 El <i>actionPerformed</i>	125
14.3 El <i>actionPerformed</i> deshabilitando un botón	127
14.4 Ases 2: capturando eventos con función <i>Action</i>	129
14.5 Ases 3: eventos con <i>actionListener</i> en <i>frame</i>	130
14.6 Evento en <i>JButton</i> con <i>actionPerformed</i> en un <i>frame</i>	132
14.7 Ases 4: eventos con <i>actionListener</i> en <i>JFrame</i>	133
14.8 Ases 5: problemas corriendo aplicaciones	134
14.9 Ases 6: eventos con <i>actionListener</i> en <i>applet</i>	135
14.10 Ases 7: cómo imprimir una matriz cuando se presiona un botón.....	136
14.11 El <i>actionPerformed</i> con varios botones	142
14.12 Personalizando el <i>actionPerformed</i> en un menú	146
14.13 Capturar las teclas presionadas y desactivadas	148
14.14 Programa que muestra las teclas que se van presionando	149
14.15 Convertir un evento en un objeto.....	157
14.16 Eventos en clases especializadas y representación gráfica en un <i>applet</i>	159
CAPÍTULO 15. MANEJO DETALLADO DE LA CLASE STRING	165
15.1 Funciones principales de cadenas.....	165
15.2 Manejo de funciones <i>length ()</i> y <i>substring ()</i>	166
15.3 Manejo de función <i>regionmatches ()</i>	167
15.4 Manejo de función <i>replace ()</i>	168
15.5 Concatenación de cadenas.....	169
15.6 Invertir una palabra	171
15.7 Contando el número de palabras digitadas	172
CAPÍTULO 16. PROGRAMAS PARA APLICAR LOS CONCEPTOS VISTOS	175
16.1 Consiguiendo la fecha actual e imprimiéndola.....	175
16.2 Programa que calcula días, meses y años de vida	177
16.3 Búsqueda binaria	180
16.4 Permutaciones sin repetición	186
16.5 Ejemplo de una calculadora como <i>applet</i>	191
16.6 Programa que calcula un polinomio en un valor determinado....	198

ÍNDICE DE CUADROS

CUADRO	2.1 “Hola.java”	9
CUADRO	2.2 Respuesta de Java	10
CUADRO	2.3 Aplicación <i>applet</i>	11
CUADRO	2.4 Código HTML para <i>applet</i>	11
CUADRO	3.1 Estructura general de un <i>applet</i>	23
CUADRO	3.2 Estructura general de una aplicación <i>standalone</i>	23
CUADRO	3.3 Métodos Java.....	24
CUADRO	3.4 Interfaz de la clase <i>applet</i>	27
CUADRO	3.5 Palabras clave de Java.....	28
CUADRO	3.6 Palabras reservadas de Java.....	28
CUADRO	3.7 Imprimir un texto en un <i>applet</i> con función distinta a <i>paint ()</i>	30
CUADRO	4.1 Variable <i>static</i>	32
CUADRO	4.2 Generación caracteres UNICODE.....	37
CUADRO	6.1 Variables en Java.....	42
CUADRO	6.2 Tipos de datos en variables	42
CUADRO	6.3 Objetos de tipo <i>double</i>	44
CUADRO	6.4 Objetos de tipo <i>float</i>	45
CUADRO	6.5 Objetos de tipo <i>Integer</i>	45
CUADRO	6.6 Función <i>parseinteger</i>	46
CUADRO	6.7 Objetos de tipo <i>long</i>	47
CUADRO	6.8 Objetos tipo <i>string</i>	47
CUADRO	7.1 Estructura de la instrucción <i>for</i>	49
CUADRO	7.2A Ejemplo usando <i>for</i>	50
CUADRO	7.2B Ejemplo usando <i>for</i> . Resultado.....	50
CUADRO	7.3A <i>For</i> con varias expresiones	51
CUADRO	7.3B <i>For</i> con varias expresiones. Resultado.....	51
CUADRO	7.4A <i>For</i> con varias expresiones, cambiando variable.....	52
CUADRO	7.4B <i>For</i> con varias expresiones, cambiando variable. Resultado.....	52
CUADRO	7.5A <i>For</i> con varias expresiones	53
CUADRO	7.5B <i>For</i> con varias expresiones. Resultado.....	54
CUADRO	7.6 <i>For</i> inicializando la expresión.....	54
CUADRO	7.7 <i>For</i> estableciendo el incremento	55

CUADRO 7.8	<i>For</i> estableciendo el incremento de la variable.....	56
CUADRO 7.9	<i>For</i> sin argumentos.....	57
CUADRO 7.10	Uso de etiquetas en <i>bucles</i>	58
CUADRO 7.11	Uso de <i>Return</i>	58
CUADRO 7.12A	<i>For</i> con secuencias <i>continue</i> y <i>break</i>	59
CUADRO 7.12B	<i>For</i> con secuencias <i>continue</i> y <i>break</i> . Resultado.....	60
CUADRO 7.13A	Estructura de la instrucción <i>if</i>	60
CUADRO 7.13B	Estructura de la instrucción <i>if</i> . Resultado, si cumple	61
CUADRO 7.13C	Estructura de la instrucción <i>if</i> . Resultado, si no cumple.....	61
CUADRO 7.14A	<i>Applet</i> con instrucción <i>if</i>	61
CUADRO 7.14B	Resultado.....	62
CUADRO 7.15	Ciclo <i>while</i>	62
CUADRO 7.16	El bucle <i>while</i>	63
CUADRO 7.17	Equivalencias entre <i>while</i> y <i>for</i>	64
CUADRO 7.18	Uso de <i>while</i> anidado.....	65
CUADRO 7.19A	<i>While</i> que incrementa y/o decrementa la variable.....	66
CUADRO 7.19B	<i>While</i> que incrementa y/o decrementa la variable. Resultado.....	67
CUADRO 7.20	<i>Try-catch-throw</i>	68
CUADRO 9.1	Conversión de datos <i>string</i> en datos primitivos	74
CUADRO 9.2	Conversión de datos primitivos en cadenas	74
CUADRO 10.1	Instrucción <i>repaint</i>	78
CUADRO 10.2	Lectura de un dato autónomo	79
CUADRO 10.3	Caja de diálogo	80
CUADRO 10.4	Programa que saluda al usuario	80
CUADRO 10.5	Cálculo de la raíz cuadrada de un número	82
CUADRO 11.1	Estructura de un programa autónomo.....	85
CUADRO 11.2	Ubicación de la ventana	87
CUADRO 11.3	Variables del problema	91
CUADRO 11.4	Cajas de texto.....	91
CUADRO 11.5	Clase “procesa”	92
CUADRO 11.6	Vector <i>array</i>	92
CUADRO 11.7	Contenido del vector	93
CUADRO 11.8	Función “fusila ()”	94
CUADRO 11.9	Instrucciones si no se usa <i>if</i>	96

CUADRO 11.10 Nuevo análisis del vector.....	96
CUADRO 11.11 Cuenta de los saltos.....	96
CUADRO 11.12APrograma de iteración.....	97
CUADRO 11.12B Programa de iteración. Resultado	100
CUADRO 11.13 Programa de intervalo	102
CUADRO 11.14 Programa de las Torres de Hanói.....	108
CUADRO 12.1 Programa que maneja números aleatorios	111
CUADRO 12.2 Cómo crear un menú.....	113
CUADRO 13.1 Programa para convertir un binario en decimal.....	117
CUADRO 13.2 Programa para convertir un decimal en binario.....	121
CUADRO 14.1 Modificación función <i>actionPerformed</i>	124
CUADRO 14.2 Interfaz <i>ActionListener</i>	125
CUADRO 14.3 Compilación.....	126
CUADRO 14.4 Resultado.....	127
CUADRO 14.5 Eventos sin la función <i>public boolean action</i>	127
CUADRO 14.6 <i>ActionPerformed</i> deshabilitando un botón	129
CUADRO 14.7 Capturar eventos con <i>action</i>	131
CUADRO 14.8 Compilación.....	132
CUADRO 14.9 Eventos con <i>actionListener</i>	132
CUADRO 14.10 Evento en <i>JButton</i>	134
CUADRO 14.11 Eventos con <i>actionListener</i> en <i>JFrame</i>	135
CUADRO 14.12 Error.....	136
CUADRO 14.13 Captura de eventos con objeto <i>actionListener</i>	137
CUADRO 14.14 Instrucciones para ubicar los objetos	139
CUADRO 14.15 Creación de objetos.....	140
CUADRO 14.16 Instrucciones para inicializar la matriz	140
CUADRO 14.17 Matriz creada	140
CUADRO 14.18 Implementación de la nueva clase	141
CUADRO 14.19 Programa para imprimir el contenido de una matriz	142
CUADRO 14.20 <i>ActionPerformed</i> con varios botones	144
CUADRO 14.21 Creación de un menú	148
CUADRO 14.22 Capturar teclas ASCII.....	150
CUADRO 14.23 Captura los eventos de presionar una tecla	152
CUADRO 14.24 Convertir un evento en un objeto.....	160
CUADRO 14.25 Instrucciones para la creación de objetos de interfaz.....	162

CUADRO 14.26 Instrucciones en contexto gráfico	163
CUADRO 14.27 Código para la creación de “Acción”	164
CUADRO 14.28 Programa completo para graficar.....	165
CUADRO 15.1 Funciones <i>length</i> y <i>substring</i>	168
CUADRO 15.2 Función <i>matches</i>	169
CUADRO 15.3 Función <i>replace</i>	170
CUADRO 15.4 Concatenación de cadenas.....	172
CUADRO 15.5 Inversión de una palabra.....	173
CUADRO 15.6 Contar palabras digitadas	174
CUADRO 16.1 Creación clase <i>date</i>	177
CUADRO 16.2 Cálculo de días, meses y años	179
CUADRO 16.3 Implementación de la búsqueda binaria	185
CUADRO 16.4 Programa de permutaciones.....	191
CUADRO 16.5 Calculadora como <i>applet</i>	193
CUADRO 16.6 Código para validar el polinomio	201
CUADRO 16.7 Coeficientes del polinomio	202
CUADRO 16.8 Evaluación del polinomio	203
CUADRO 16.9 Captura polinomio de grado <i>n</i> y evaluación	203



PREFACIO

Este libro nace de la necesidad de fundamentar el lenguaje Java, no sólo haciendo uso de este, sino, también, conceptualizando cómo relacionar los diversos conocimientos adquiridos.

El libro se considera un material importante para la enseñanza de temas básicos de Java. Se recomienda su lectura y estudio a todas aquellas personas que trabajen en el desarrollo de aplicaciones informáticas y con implementación de sistemas de información; así como a los estudiantes de carreras afines o relacionadas con las tecnologías de la información.



INTRODUCCIÓN

En esta ocasión, el libro entregado intenta dar una visión inicial en los primeros dos capítulos sobre los entornos y compiladores, mostrando los conceptos iniciales de Java, los cuales permitirán a las personas que están empezando con este lenguaje comprender los fundamentos del mismo y cómo se debe trabajar en un entorno de estos. Si una persona quiere sacarle provecho a este material y ya ha tenido formación con otro lenguaje de programación puede saltarse los primeros capítulos.

Propiamente se entra en el lenguaje Java en el capítulo tres, en donde se hace énfasis en las diferencias entre aplicaciones tipo *applets* y tipo *standalone*. De esta manera el lector tiene una idea más clara del tipo de aplicaciones que se muestran en el desarrollo del texto.

Los capítulos cuatro, cinco y seis profundizan en las variables, los atributos y los tipos de datos. De modo que con esta información se da paso a los capítulos siete y ocho, en los cuales se profundiza en las estructuras de control y los operadores y asignaciones.

Las anteriores cuestiones son muy similares en otros lenguajes de programación, sin embargo Java incorporó, de forma pionera, el concepto de “tipos de objetos de datos” y los diferencia de los tipos primitivos de datos. Este concepto puede parecer confuso para quienes se inician en el lenguaje Java, por esta razón los capítulos nueve y diez abordan este tema y lo aplican en programas sencillos.

Los capítulos once, doce, trece y catorce muestran programas sencillos que utilizan tipos primitivos de datos y de objetos de datos. Allí mismo, se manejan eventos y se profundiza en los objetos de interfaz gráfica. Adicionalmente, los capítulos quince y dieciséis trabajan con mayor profundidad la clase *String*, que ofrece múltiples opciones para la manipulación de cadenas, mostrando, además, programas que aplican estos conceptos.

CAPÍTULO 1

ALGUNOS ENTORNOS DE DESARROLLO EN JAVA

1.1 El verdadero compilador es el JDK

A al comenzar a programar en Java es importante diferenciar dos cosas: el ambiente de desarrollo y el compilador. El ambiente de desarrollo es la herramienta que visualmente sirve para que el desarrollador introduzca el código fuente. El compilador es la herramienta que finalmente vuelve ejecutable el código y crea un archivo binario que es posible ejecutarlo en distintas plataformas. Existen múltiples ambientes de desarrollo en Java, pero el compilador para el caso de este lenguaje es el llamado *JDK* (*Java Development Kit*), que no es más que el encargado de transformar un código fuente en código ejecutable.

Ya comprendido este concepto diremos que la lista de ambientes de desarrollo es grande, pero cualquiera que se use, internamente, envía el código fuente digitado por el usuario al *JDK* para que este lo compile. Si el *JDK* se daña, obviamente no será posible realizar el proceso de compilación de un programa fuente.

Inicialmente se trabaja directamente en un bloc de notas en conjunto con el *JDK*, pero esto termina siendo un estilo de programación tortuoso, debido a la falta de herramientas que ayuden a ingresar el código fuente de una forma más amigable. De esta forma surgen los entornos de desarrollo, como por ejemplo *JCreator*, que no es mas que un ambiente en que podemos visualmente apreciar el código fuente que estamos ingresando y manipularlo.

Por lo anterior, es necesario comprender que el compilador de Java es el JDK y no los ambientes de desarrollo, de manera tal que si este se llegase a dañar, por más ambientes de desarrollo en Java que se tengan, ninguno podrá compilar y ejecutar aplicaciones Java.

1.2 *Jcreator*

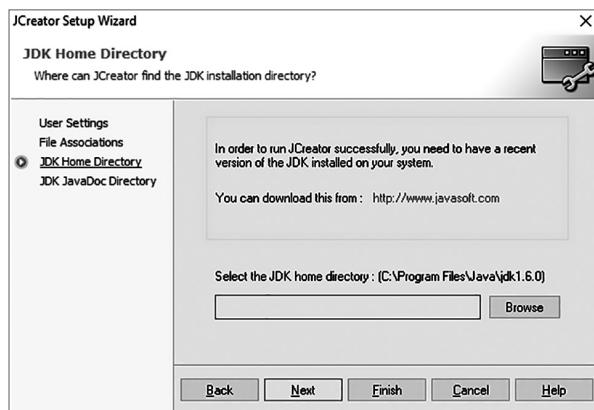
Jcreator es un entorno de desarrollo que tiene opciones para: 1) guardar documentos o exportarlos, 2) visualizarlos de varias maneras y añadir recursos a los diversos proyectos, 3) construir los binarios ejecutables, 4) configurar la ruta del JDK en donde se especifica el tipo de versión concreta que realizará el trabajo de compilar. Las mismas opciones están presentes en el software llamado *realJ*, sin embargo la numeración de líneas que tiene *JCreator*, lo hacen un software útil para leer con mayor fluidez el código fuente que se escribe.

Lo interesante del entorno de desarrollo *JCreator* es que no necesita grandes cantidades de memoria para su ejecución, que es libre y que incluso existen versiones portables que permiten potencialmente tener un compilador a la mano, aún cuando en la máquina de desarrollo no se haya instalado entorno de desarrollo alguno.

Otra de las diferencias está en que muestra el número de la línea de código, lo cual en muchas ocasiones suele ser bastante útil. Sin embargo *realJ*, cuando se compila un programa y genera errores, permite, al hacer clic sobre la ventana de inspección, llegar a la línea en cuestión, mientras que con *Jcreator* nos tendremos que desplazar hasta esta manualmente.

Al momento de iniciar la instalación de *JCreator* si no se cuenta con una versión del compilador de Java instalada se presenta el mensaje de la siguiente figura:

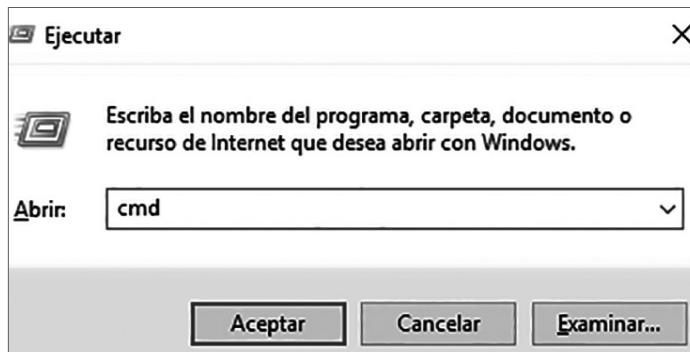
FIGURA 1.1



-Fuente: elaboración propia-

Si no se tiene la seguridad de cuál es la versión de Java instalada, se puede abrir una consola de Windows presionando la tecla de Windows y al mismo tiempo la tecla “e” y se digita la palabra “cmd”, tal como muestra la figura 1.2.

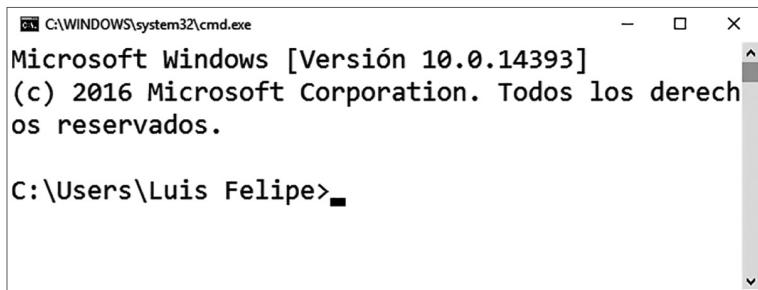
FIGURA 1.2



-Fuente: elaboración propia-

En la figura 1.2 se hace clic sobre el botón aceptar y se obtiene una consola similar a la de la figura 1.3.

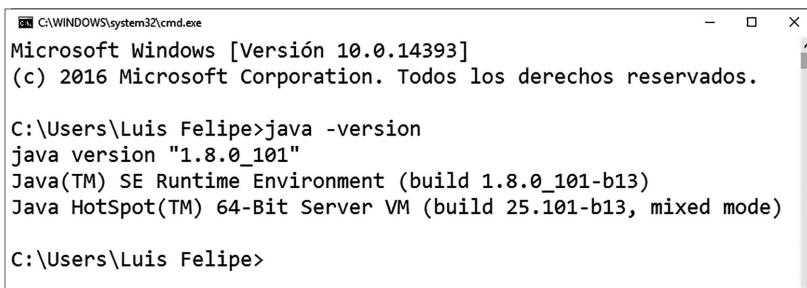
FIGURA 1.3



-Fuente: elaboración propia-

En la consola de la figura 1.3 se escribe el comando “java -version” con el fin de obtener la versión de java que tenemos instalada. En nuestro caso se ha digitado este comando y se ha obtenido la respuesta de sistema mostrada en la figura 1.4.

FIGURA 1.4



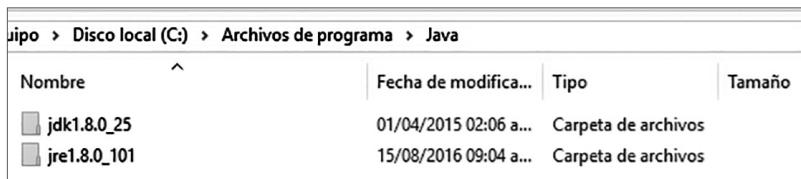
```
C:\Users\Luis Felipe>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:\Users\Luis Felipe>
```

-Fuente: elaboración propia-

En la figura 1.4 se aprecia que la versión de Java instalada en la máquina es la versión 1.8.0_101. No la versión llamada *Runtime*, de Java. Esto quiere decir que es la versión para ejecución de aplicaciones, no para creación de aplicaciones. Lo usual y normal es que se tenga instalada la misma versión de ejecución que de creación y compilación. Sin embargo esta situación no siempre es cierta en muchas máquinas, posiblemente porque intencionalmente se ha instalado una versión de compilación distinta a la versión que se tiene de ejecución. Una forma práctica de salir de dudas es buscar en la carpeta de archivos de programa los nombres de las carpetas existentes. Este ejercicio se ha realizado y se ha obtenido el listado de carpetas mostrado en la figura 1.5.

FIGURA 1.5



Usted > Disco local (C:) > Archivos de programa > Java

Nombre	Fecha de modifica...	Tipo	Tamaño
jdk1.8.0_25	01/04/2015 02:06 a...	Carpeta de archivos	
jre1.8.0_101	15/08/2016 09:04 a...	Carpeta de archivos	

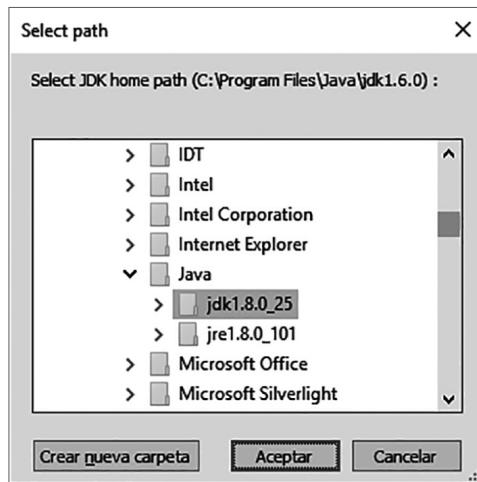
-Fuente: elaboración propia-

En la figura 1.5 se aprecia que la versión de ejecución de Java comienza con las letras “jre” que proviene de las palabras en inglés *Java Runtime Environment* que significan ambiente de ejecución de Java. De esta forma podemos afirmar que la versión de Java para ejecución es la 1.8.0_101, en tanto que la versión de compilación es la 1.8.0_25. Esto no quiere decir que las aplicaciones que se desarrollen con esta versión de compilación no se puedan ejecutar si se tiene aquella versión de ejecución. De hecho, las diferencias entre una versión y otra está en los últimos números, no en los primeros tres números. Lo más importante es que los primeros tres números sean idénticos, para asegurar que no se presenten problemas. En el caso mostrado en la figura 1.5, tanto la versión de compilación como de ejecución corresponden a la versión 1.8.0. Nuevamente repetimos que si se presentan problemas es mejor tener la versión idéntica de Java.

Es muy importante tener claras las rutas del JDK (*Java Development Kit* = Kit de desarrollo de Java, que sirve para compilar aplicaciones) y la ruta del JRE (*Java Runtime Environment* = Ambiente de ejecución de Java, que sirve para ejecutar aplicaciones).

Retomando el problema planteado en la figura 1.1, en donde se solicitaba la ruta del JDK de Java, para el caso de la máquina en la que se han tomado los anteriores pantallazos, se ha seleccionado la ruta mostrada en la figura 1.6.

FIGURA 1.6 Selección de ruta del jdk



-Fuente: elaboración propia-

En donde se puede apreciar que, a pesar de que en el explorador de archivos de Windows aparezca la ruta inicial “c:\Archivos de Programa\Java”, la ruta real es “C:\Program Files\Java”. Esto parece confundir a las personas que no conocen como funciona Windows. Sin embargo, al volver al listado de carpetas de la figura 1.5 y hacer clic en la barra de direcciones, se ve que Windows muestra la ruta real del disco duro donde están estas carpetas.

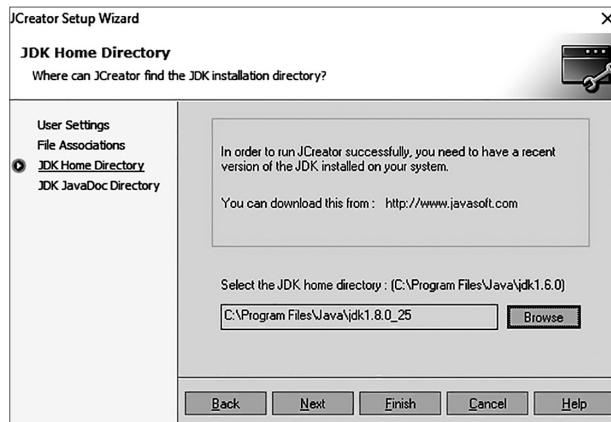
FIGURA 1.7

Portapapeles	Organizar	Nuevo
<input type="checkbox"/> C:\Program Files\Java		
	Nombre	Fecha de modifica... Tipo
rápido	jdk1.8.0_25	01/04/2015 02:06 a... Carpeta de a
	jre1.8.0_101	15/08/2016 09:04 a... Carpeta de a

-Fuente: elaboración propia-

Finalmente, una vez se le ha indicado a *JCreator* la ruta del JDK aparece una ventana similar a la de la figura 1.8.

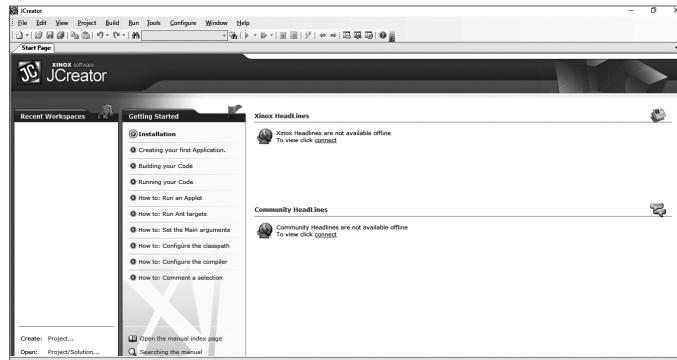
FIGURA 1.8



-Fuente: elaboración propia-

Realizadas las configuraciones necesarias se tiene el ambiente de desarrollo de Java listo para ser usado mediante el programa *JCreator* tal como se muestra en la figura 1.9.

FIGURA 1.9



-Fuente: elaboración propia-

1.3 RealJ

RealJ es un *software* que permite ejecutar aplicaciones Java, aunque no es tan popular como el *JCreator*. Sin embargo hoy en día se pueden usar otros entornos de programación como *Netbeans*, el cual requiere mayor poder de computo. Sin embargo para aplicaciones de escritorio donde los recursos son limitados *JCreator* es una opción ligera que no requiere mayores capacidades computacionales y por tanto es el recomendado en ambientes de aprendizaje inicial en donde muchos estudiantes tienen computadores que no tienen grandes capacidades computacionales.

CAPÍTULO 2

MANEJANDO EL COMPILADOR DE JAVA

Es muy importante no solamente estar familiarizados con las interfaces amigables de Java, como el *realJ*, sino también con los comandos en línea, que se pueden ejecutar si se tiene instalado Java, con el fin de crear aplicaciones, compilar y mostrar errores en el proceso, tanto en la creación de aplicaciones *standalone* como las de tipo *applet*. A continuación se explicara cada una.

2.1 Cómo crear aplicaciones *standalone* utilizando comandos de Java

1. Cree un archivo denominado “*hola.java*” en una carpeta llamada “*tecnó*” y escriba lo siguiente en dicho archivo:

CUADRO 2.1 “*Hola.java*”

```
class hola
{
    public static void main(String[] args)
    {
        System.out.println("Hola a todos");
    }
}
```

1. En una consola digite el comando:

```
javac -g -verbose hola.java
```

2. Con lo cual la respuesta del motor de Java será:

CUADRO 2.2 Respuesta de Java

```
[parsing started hola.java]
[parsing completed 170ms]
[loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/lang/Object.
class)][loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/lang/
String.class)]
[checking hola]
[loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/lang/
System.class)]
[loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/io/PrintStream.
class)]
[loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/io/
FilterOutputStream.class)]
[loading C:\JDK1.3.1_01\JRE\lib\rt.jar(java/io/OutputStream.
class)]
[wrote hola.class]
[total 610ms]
```

Con el anterior comando, se le pide a Java que muestre en detalle lo que el compilador está haciendo.

Podemos observar que se hizo un *parsing*, que significa una pasada por el archivo, en 170 milisegundos, cargó unas librerías y al final generó un archivo “hola.class”.

¡OJO! Como esto es un programa en Java, no un *Applet*, es posible correrlo desde la línea de comandos digitando:

```
java hola
```

De modo que aparece en la consola de línea de comandos la respuesta:

```
Hola a todos
```

2.2 Cómo crear aplicaciones *applet* utilizando comandos de Java

1. Cree el siguiente archivo en el block de notas o en el editor de DOS o en su defecto en cualquier editor de archivos planos:

CUADRO 2.3 Aplicación *applet*

```
import java.awt.Graphics;
import java.applet.Applet;
public class HolaApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hola a todos!!!", 5, 25);
    }
}
```

2. Guárdelo con el nombre de “MiApplet.java”.
3. Se compila mediante el comando:

```
javac MiApplet.java
```

4. De modo que se genera un archivo denominado “MiApplet.class”.
5. Antes de poder ver el *Applet* es necesario que se cree una página. En este caso, cree una página HTML denominada “MiApplet.html”, la cual contiene el siguiente código fuente:

CUADRO 2.4 Código HTML para *applet*

```
<HTML>
<HEAD>
    <TITLE>HOLA</TITLE>
</HEAD>
<BODY>
    <applet code="MiApplet.class" height="100"
width="100">
    </applet>
</BODY>
</HTML>
```

JAVA Básico

El lenguaje de programación Java es uno de los lenguajes más usados en las aplicaciones ya que es un lenguaje multiplataforma, se ejecuta en diversos dispositivos y tiene altos cánones de seguridad. Los autores ofrecen un manual de programación que, además de mostrar las funciones y prestaciones de las diferentes APIS de programación, permite usarlas desde su primera lección.

El primer capítulo muestra el manejo del entorno de programación (los conceptos usados son igualmente útiles para cualquier otro entorno). Los capítulos 3 al 9 se encargan de mostrar los elementos básicos del lenguaje (destacando las diferencias con otros). Finalmente, los últimos capítulos dan los elementos para que el lector pueda construir aplicaciones utilizando el lenguaje Java.

El libro va dirigido a docentes y estudiantes de programas relacionados con las tecnologías de la información y las comunicaciones.

Colección: Ingeniería y salud en el trabajo
Área: Informática

ECOE
EDICIONES

www.ecoediciones.com

Incluye

- Ejemplos para explicar fácilmente los tipos primitivos de datos y los tipos de objetos de datos.
- Desarrollo de interfaces reales, para el posterior desarrollo de aplicaciones.
- Ejemplos reales de manipulación de cadenas de caracteres.
- Ejemplos ejecutables y estructurados que involucran la programación orientada a objetos.

Edwin Rivas Trujillo

Ingeniero Electricista y Mtr. en Sistemas de Generación de Energía Eléctrica (U. del Valle, Colombia). M y Ph.D. I. Eléctrica, Electrónica y Automática (U. Carlos III de Madrid- España). Docente y Director del grupo de investigación de Compatibilidad e Interferencia Electromagnética de la U. Distrital Francisco José de Caldas (Colombia). Autor de artículos de investigación científica, posee derechos de autor en software de aplicaciones web y móviles.

Luis Felipe Wanumen Silva

Ingeniero de Sistemas y Especialista en Ingeniería de Software (U. Distrital Francisco José de Caldas); M. I. Sistemas y Computación (U. Javeriana). Docente de la U. Distrital e integrante del grupo de investigación METIS de la misma. Autor de artículos de investigación y libros de programación de sistemas y bases de datos. Desarrollador de Spring y actualmente trabaja en el área de juegos móviles multiplataforma.

Darín Jairo Mosquera Palacios

Ingeniero de Sistemas, Especialista y Magister en Teleinformática (U. Distrital Francisco José de Caldas). Docente de Planta y Director del grupo de investigación Orion de la misma Universidad. Autor de artículos de investigación científica, posee derechos de autor en software de aplicaciones web y móviles.

ISBN 978-958-771-541-5



9 789587 715415

e-ISBN 978-958-771-542-2