

UNIVERSIDAD ESTATAL A DISTANCIA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES

Roberto Morales Hernández

Guía de estudio para el curso
Lógica para computación
Código: 3071

San José, Costa Rica
2008



Derechos reservados
Universidad Estatal a Distancia de Costa Rica

Producción académica y
asesoría pedagógica:
Ana María Sandoval Poveda

Encargado de cátedra:
Roberto Morales Hernández

Especialista en contenidos:
Nuria Rodríguez Sama

Diseño de esquemas:
Roberto Morales Hernández

Trazo de esquemas:
Ana María Sandoval Poveda

Corrección filológica:
Fiorella Monge Lezcano



Tabla de contenidos

<i>Presentación</i>	<i>IV</i>
<i>Objetivo general</i>	<i>V</i>
<i>Requisitos del curso</i>	<i>V</i>
<i>Material de apoyo</i>	<i>VI</i>
<i>Desglose de capítulos por tema</i>	<i>VI</i>
<i>Consejos de estudio</i>	<i>VIII</i>
<i>Tema 5 Implementación de algoritmos en diferentes lenguajes de programación</i>	<i>1</i>
<i>Propósitos del tema</i>	<i>1</i>
<i>Guía para el capítulo 12</i>	<i>3</i>
<i>Sumario</i>	<i>3</i>
<i>Comentarios generales</i>	<i>4</i>
<i>Ejercicios sugeridos</i>	<i>11</i>
<i>Guía para el capítulo 13</i>	<i>17</i>
<i>Sumario</i>	<i>17</i>
<i>Comentarios generales</i>	<i>18</i>
<i>Ejercicios sugeridos</i>	<i>24</i>
<i>Referencias bibliográficas</i>	<i>33</i>
<i>Referencias electrónicas de consulta para el estudiante</i>	<i>35</i>

Presentación

El curso **Lógica para computación** pretende que los estudiantes, que inician la carrera de Informática administrativa desarrollen el proceso lógico mental. Esta habilidad les permitirá conocer las herramientas necesarias para incursionar en el mundo de la programación de computadoras.

Este curso es introductorio y pretende proporcionarle las bases necesarias para enfrentar las materias del programa de estudios.

A lo largo de este curso, usted se enfrentará a varios temas complementarios entre sí. A continuación se le ofrece una breve descripción de ellos.

1. El primer tema se denomina **Definiciones básicas de computación**. Con este tema, se busca que el estudiante adquiera los conocimientos básicos para identificar la estructura básica de una computadora y los diferentes componentes que la forman. También se estudiarán los lenguajes de programación y los diferentes códigos utilizados para trabajar en computación.
2. El segundo tema recibe el nombre de **Introducción a la lógica de programación**. Los conocimientos trabajados en los capítulos de este tema le permitirán explicar los elementos de la lógica, su empleo en las labores de programación de computadoras y en la resolución de los ejercicios planteados.
3. El tercer tema, **Técnicas de diseño de algoritmos**, le dará la oportunidad de utilizar diferentes técnicas de construcción de algoritmos que resuelvan casos reales de programación y orientados a datos y a miniespecificaciones (documentación).

4. El cuarto tema corresponde a **Desarrollo de diagramas de flujo**. Lo estudiado en este tema le dará la posibilidad de desarrollar diagramas de flujo con estándares establecidos a nivel internacional y complementos de programación visual.
5. El quinto y último tema corresponde a la **Implementación de algoritmos en diferentes lenguajes de programación**. En esta parte del curso, se trabajará con el uso de la lógica y los algoritmos para resolver problemas de diversa complejidad empleando los lenguajes de programación: C++ y Java.

Objetivo general

El propósito general de este curso es introducirlo en el uso de algoritmos que le permitan resolver problemas. Los problemas pueden ser tanto específicos como de propósito general y para su resolución se requiere una comprensión e implementación de la lógica, que se consigue con la aplicación de técnicas actuales de desarrollo de *software* orientado a objetos y la consideración criterios de calidad apropiados para su solución.

Requisitos del curso

Este curso es introductorio y corresponde al primer acercamiento a la Informática. Como tal, no tiene requisitos y es parte del plan de diplomado de la carrera de Informática administrativa.

Material de apoyo

Para trabajar los cinco temas de este curso requerirá de los siguientes materiales:

- Unidad didáctica: RAMÍREZ, FELIPE (2007). *Introducción a la programación. Algoritmos y su implementación en Visual Basic.NET, C#, C++ y Java*. Segunda edición. México D.F.: Alfaomega Grupo Editor.
- Folleto de orientaciones generales para este curso específico: MORALES HERNÁNDEZ, ROBERTO (2008). *Orientaciones para el curso Lógica para computación*. San José: EUNED.
- Esta guía de estudio: MORALES HERNÁNDEZ, ROBERTO (2008). *Guía de Estudio para el curso Lógica para computación*. EUNED.

Desglose de capítulos por tema

El curso **Lógica para computación** está compuesto por los cinco temas que se detallaron anteriormente. Estos temas se estudian a través de 11 capítulos de la unidad didáctica.

El siguiente cuadro le indica qué capítulos corresponden a cada tema por estudiar:

Tema	Capítulo	Números de página
Tema 1: Definiciones básicas de computación		
➤ La computadora y los niveles de datos	1	1 – 16
➤ Lenguajes de programación y su clasificación	2	17 – 34
Tema 2: Introducción a la lógica de programación		
➤ Introducción a la lógica de programación	3	35 – 54
➤ La naturaleza de los datos	4	55 – 74

➤ Operadores y reglas de precedencia	5	75 – 92
Tema 3: Técnicas de diseño de algoritmos		
➤ Algoritmos para el análisis de casos reales	6	93 – 134
➤ Algoritmos orientados a datos y miniespecificaciones	7	135 – 174
Tema 4: Desarrollo de diagramas de flujo		
➤ Diagramas de flujo	8	175 – 208
➤ Programación visual usando Raptor	9	209 – 266
Tema 5: Implementación de algoritmos en diferentes lenguajes de programación		
➤ Implementación de algoritmos en C++	12	335 – 366
➤ Implementación de algoritmos en Java	13	367 – 441

Cada uno de los temas de la unidad didáctica cuenta con sus propios objetivos de estudio, contenidos, actividades y exámenes para la autoevaluación. En esta guía encontrará orientaciones para su estudio y ayudas adicionales para potenciar la comprensión de los temas. Los cinco temas tienen una estructura similar:

- Objetivos de estudio para el tema. Son diferentes de los que propone el autor de la unidad didáctica, pues se adaptan a las necesidades de la carrera Informática administrativa de la UNED.
- Una guía de lectura para cada capítulo que forma el tema en estudio. Todos los temas se componen por dos o más capítulos de la unidad didáctica. Las guías de lectura incluyen:
 - Un sumario que detalla los aspectos que se estudiarán.
 - Un índice que señala las páginas de la unidad didáctica que debe abarcar en su estudio.
 - Cometarios generales sobre los contenidos. En esta sección encontrará actividades, esquemas resumen y conceptos relacionados con los temas.

- Ejercicios sugeridos de la unidad didáctica **con sus correspondientes respuestas**. Aquí, se incluye una guía de análisis de los mapas mentales de la unidad didáctica y las respuestas de los exámenes cortos.
- Un cierre correspondiente a los capítulos que abarca el tema.

Es necesario recordarle que todos estos recursos se complementan con la ayuda en línea que encontrará en el sitio del curso. Todas las indicaciones para utilizarlo las encontrará en *Las orientaciones para el curso*.

Consejos de estudio

*Colaboración
Lic. Ana María Sandoval Poveda*

Probablemente usted tiene una manera de estudiar que ha probado a lo largo de los años y sabe la calidad de resultados que obtiene con ella. De todas maneras, nunca está de más contar con algunas recomendaciones al respecto.

- Prepárese antes de comenzar a estudiar. Procure buscar un lugar bien iluminado. De ser posible ubíquese en un lugar solitario y acondicionado para trabajar. Tenga a mano papel, lápices, lapiceros, los materiales del curso y, llegado el momento, una computadora.
- Por diversas razones, es posible que requiera dedicar al estudio horas en las que se encuentra un poco cansada o cansado. Intente descansar un poco antes de estudiar, pues así le será más sencillo concentrarse.
- Determine de antemano el tiempo que dedicará al estudio. Procure repartir el estudio de cada tema en varios días de la semana, esto

le dará oportunidad de asimilar poco a poco los contenidos que le permitirán alcanzar los objetivos.

- Lea detalladamente *Las orientaciones del curso*. En este documento encontrará un cronograma de actividades (página 9) en el que se detalla el ritmo de lectura que debe seguir para realizar todas las actividades en el tiempo adecuado.
- Recuerde que al estudiar con una unidad didáctica impresa la lectura es el primer instrumento que debe utilizar. Siempre es recomendable un ejercicio de pre-lectura o lectura exploratoria. Esto consiste en revisar rápidamente el tema por trabajar, los subtemas que lo forman, las imágenes que complementan el texto y las actividades que deberá realizar al concluir el estudio. Desarrollar este ejercicio, le proporcionará un “panorama” de cada tema.
- Algunas personas tienen un estilo de aprendizaje visual. Para ellas el subrayado suele ser de mucha ayuda. La técnica de subrayar las ideas principales le puede ayudar a localizarlas rápidamente cuando deba utilizarlas nuevamente. Es necesario evitar la tentación de subrayar todo el texto, pues esto no le ayudará en situaciones posteriores.
- Otra técnica que le proporciona buenos resultados a las personas con estilo de aprendizaje visual es la elaboración de representaciones gráficas, tales como: esquemas, mapas mentales, mapas conceptuales o ilustraciones comentadas. Las personas con un estilo de aprendizaje kinestésico (relacionado con el movimiento y los gestos) también se benefician del uso de estas técnicas.
- Para las personas con un estilo de aprendizaje auditivo, puede ser beneficioso leer el texto en voz alta y explicar oralmente y con sus propias palabras los conceptos que estudia.

- Cuando se estudia a distancia es normal que se estudie mucho material en poco tiempo. Ante esta situación, es recomendable que se tenga a mano una hoja o un documento electrónico para anotar todas las consultas que surjan. Posiblemente, muchas de estas preguntas las responderá usted conforme avance en el estudio de los temas, por lo que se recomienda tachar las que pueda contestar sin ayuda. Ante la eventualidad de dejar alguna duda sin respuesta, es preferible tener un registro y revisarlo constantemente para formular las preguntas en el momento adecuado.
- Cada vez que lea una sección de la unidad didáctica es recomendable que determine cuál o cuáles son las ideas más importantes, cómo las usaría usted para su labor en el área de la Informática y cómo se relacionan con lo que estudió en los temas anteriores. Esto le permitirá detectar las conexiones entre los conceptos y la dependencia de unos con otros.
- Recuerde que usted tiene la posibilidad de aclarar sus dudas en las tutorías presenciales, la plataforma virtual del curso, por medio de un correo electrónico y demás servicios del PADD, o al llamar a la cátedra durante las horas de atención a estudiantes.

TEMA 5

Implementación de algoritmos en diferentes lenguajes de programación

Propósitos del tema

La implementación de los algoritmos es el último paso para convertir o traducir los algoritmos creados para solucionar una situación planteada a un lenguaje de programación. Como se nombró anteriormente, existen muchos lenguajes de programación. Para este curso se implementarán los algoritmos en dos lenguajes conocidos como C++ y Java.

Con este tema se pretende que tenga un conocimiento básico de cómo traducir los algoritmos creados en los ejercicios y temas anteriores a un lenguaje real de computadora. De esta forma sabrá cómo ejecutarlo para ver su presentación en pantalla y conocerá todo el proceso de sintaxis y semántica de un lenguaje de programación.

Para esta parte del proceso no es necesario conocer en profundidad el manejo del lenguaje, porque si se ha estudiado con ahínco los temas de algoritmos y diagramas de flujo, aprender un lenguaje de programación será un proceso natural.

Con el estudio de este tema, se pretende facilitar la implementación de algoritmos en dos lenguajes de programación. Para su estudio guíese por los siguientes objetivos:

- Determinar cuáles son los requerimientos de ambiente para desarrollar un programa de computación en los lenguajes C++ y Java.

- Valorar las generalidades del lenguaje C++ y Java, así como la estructura básica de sus programas.
- Realizar cambios de estado, peticiones de datos, muestra de datos, condicionales y bucles usando C++ y Java.
- Compilar y ejecutar programas desarrollados en C++ y Java.

Para alcanzar estos objetivos es necesario que estudie los capítulos 12 y 13 del libro *Introducción a la programación*, de Felipe Ramírez.

Guía para el capítulo 12

Este capítulo corresponde al tema **Implementación de algoritmos C++** y los contenidos por trabajar son los siguientes:

Sumario

- Implementación de algoritmos en C++.

Este tema lo encontrará detallado de la siguiente manera:

Subtema	PÁGINAS
Implementación de algoritmos en C++	336
Ambiente de desarrollo	336 – 338
Estructura básica de un programa	339
Características generales del lenguaje	339
Tipos de datos	340
Declaración de variables y arreglos	340 – 341
Operadores principales	341 – 342
Cambios de estado	342
Mostrado de datos	342 – 343
Petición de datos	343 – 344
Estructuras de decisión	344 – 345
Estructuras de control	346 – 348
Compilación y ejecución de programas	348
Mapa mental del capítulo 12	363

Comentarios generales

Es necesario que conozca algunos detalles del lenguaje de programación C++. En el siguiente cuadro se resumen algunos datos sobre él:

Paradigma	<i>multiparadigma</i> : orientado a objetos, imperativo, programación genérica
Apareció en:	1985
Diseñado por:	Bjarne Stroustrup
Tipo de dato:	fuerte, estático
Implementaciones:	GNU Compiler Collection, Microsoft Visual C++, Borland C++ Builder, Dev-C++, C-Free
Dialectos:	ISO C++, ANSI C++ 1998, ANSI C++ 2003
Influido por:	C, Simula
Ha influido a:	Ada, C#, Java, PHP, D

Cuadro 1. Datos sobre el programa C++.

C++ (se pronuncia "ce más más") es un lenguaje de programación. Fue diseñado, a mediados de los años 80, por Bjarne Stroustrup como extensión del lenguaje de programación C. Como puede ver en el cuadro 1, este lenguaje abarca tres paradigmas de la programación: la programación estructurada, la genérica y la orientada a objetos. A continuación, se detalla cada uno de ellos:

- 1. Programación estructurada:** es una forma de programar bastante clara. Utiliza únicamente estructuras secuenciales, selectivas e iterativas.
- 2. Programación genérica:** este tipo de programación está más centrada en los algoritmos que en los datos.

3. Programación orientada a objetos (POO u OOP según sus siglas en inglés): es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

Los objetos son entidades que combinan *estado*, *comportamiento* e *identidad* definidos de la siguiente manera:

- El ***estado*** está compuesto de datos. Corresponde a uno o varios atributos asignados a valores concretos (datos).

- El ***comportamiento*** se define por los procedimientos o métodos con que opera el objeto, es decir, las operaciones que se pueden realizar con él.

- La ***identidad*** es la propiedad que diferencia un objeto del resto, dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).

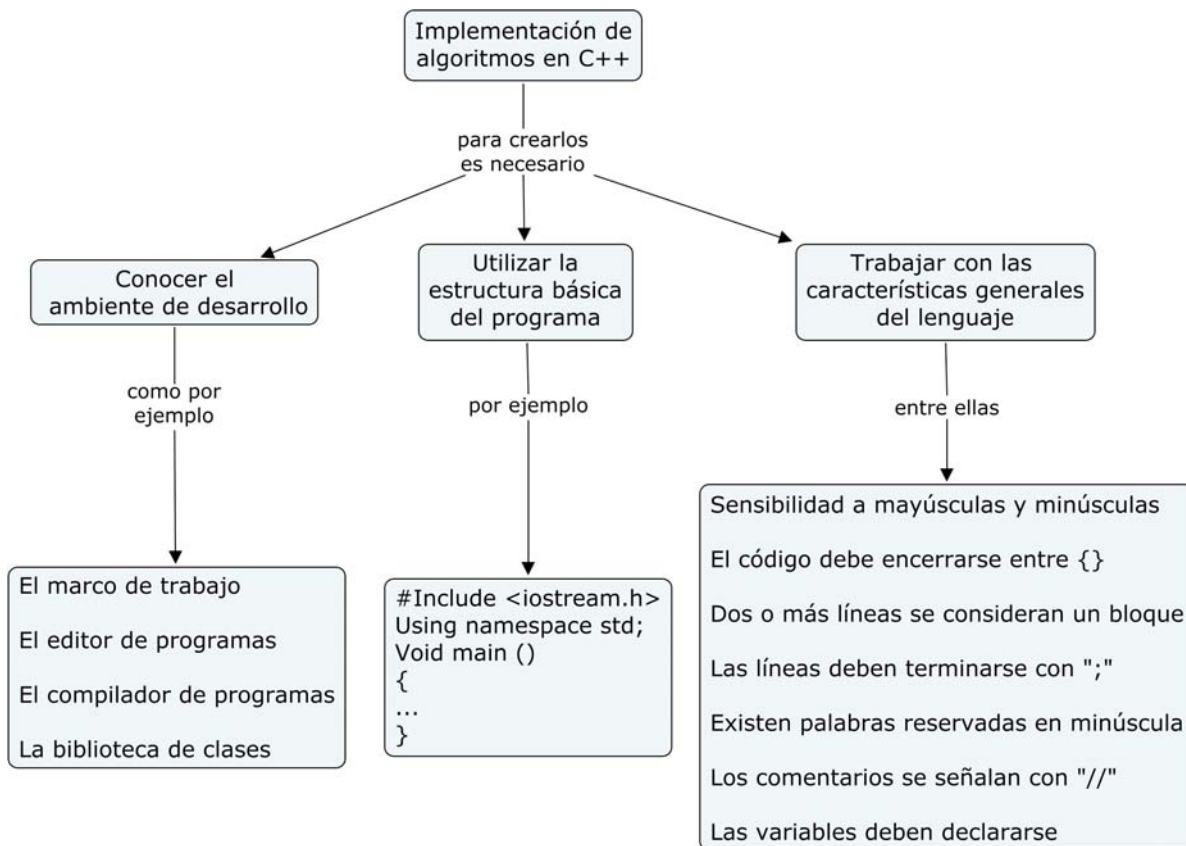
El C++ está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento. Su uso se popularizó a principios de la década de los noventa.

El lenguaje C++ está considerado como el más potente, porque permite trabajar tanto a alto como a bajo nivel de programación de computadoras. Aun así, es uno de los programas que menos automatismos trae: obliga al usuario a hacerlo casi todo manualmente (de la misma forma que el lenguaje C). Esta particularidad produce que su aprendizaje pueda resultar difícil.

A partir de la página 336 y hasta la 339 encontrará información sobre el lenguaje de programación C++. Se explica acerca del ambiente de

desarrollo, la estructura básica de un programa y las características generales del lenguaje.

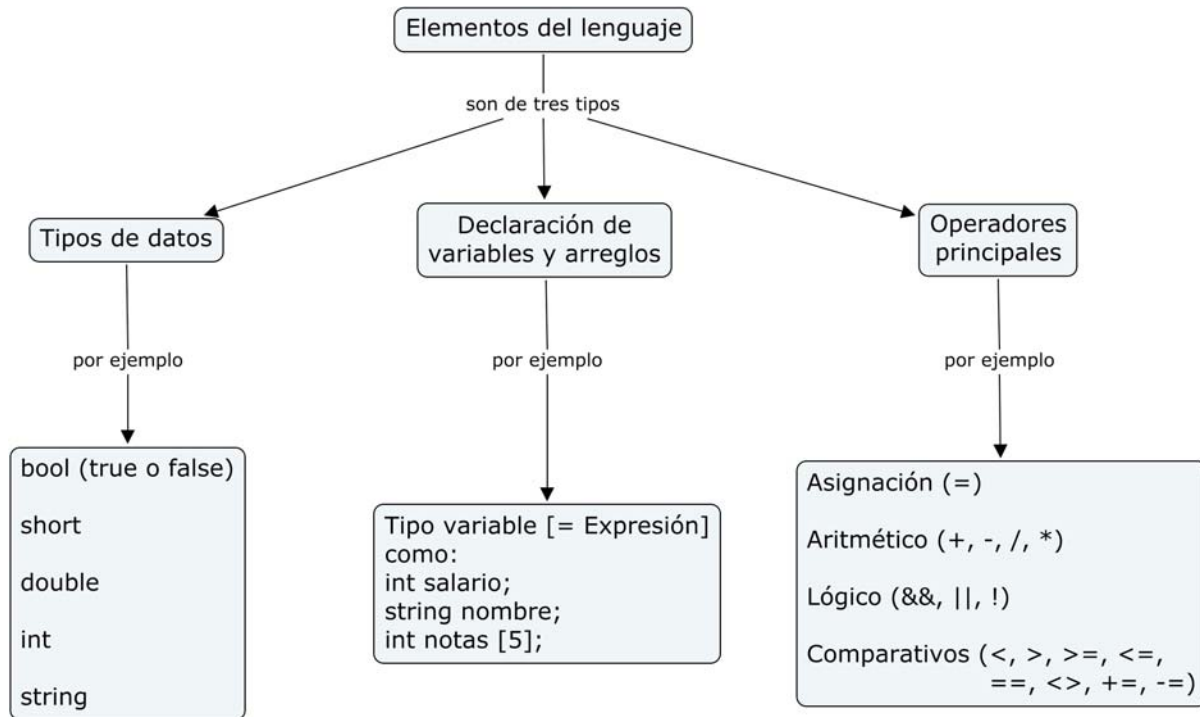
A continuación se presenta un esquema de esta sección de la unidad didáctica:



Esquema 1. Implementación de algoritmos en C++.

Revise el esquema anterior. Determine si conoce lo necesario para implementar algoritmos en este lenguaje. Repase el material de la unidad didáctica para aclarar conceptos.

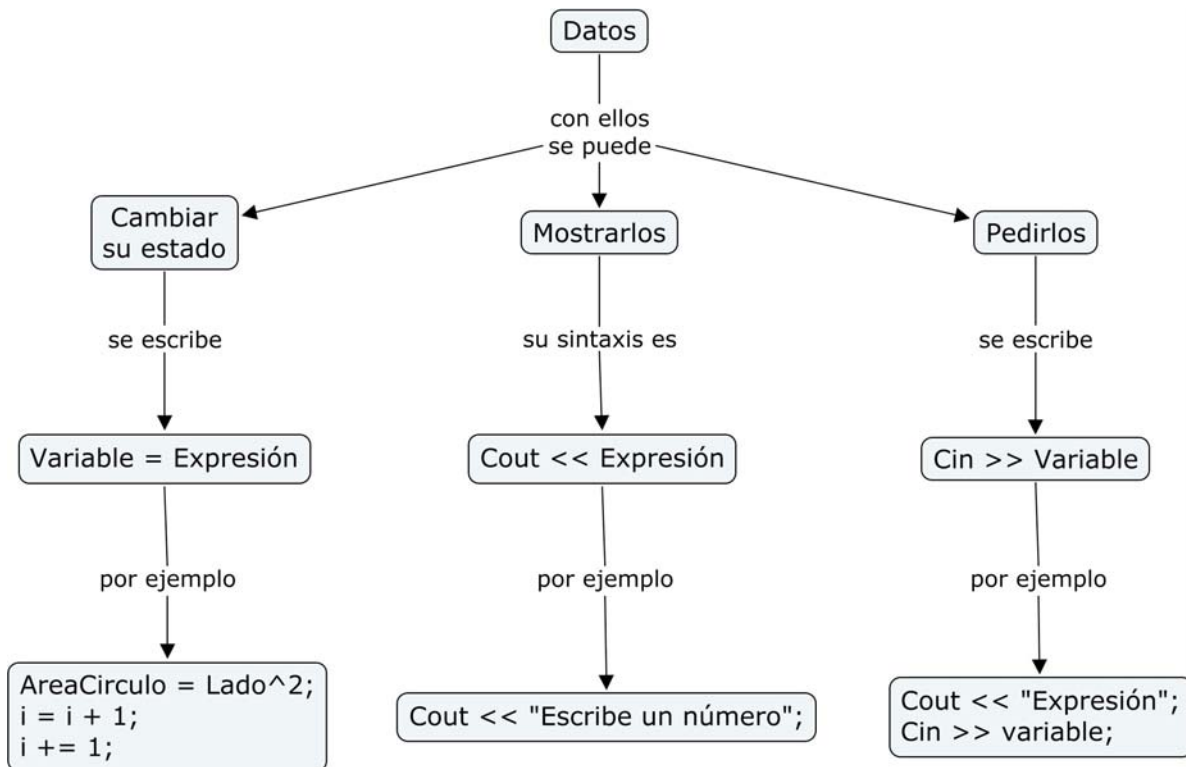
Como todo lenguaje de programación, el C++ tiene sus elementos. Ellos forman parte de la sintaxis establecida. Estas reglas deben respetarse para que las expresiones y estructuras utilizadas en la resolución de un problema sean correctas. Revise con detalle las páginas de la 340 hasta la 342 y analice el siguiente esquema:



Esquema 2. Elementos del lenguaje en C++.

¿Conoce los tres tipos de elementos? Descríbalos con sus propias palabras y proporcione ejemplos aplicados a actividades específicas propuestas por usted.

A partir de la página 342 y hasta la 344 de la unidad didáctica, se detalla la forma de utilizar los datos en el lenguaje de programación C++. A continuación, un resumen de esta temática en el siguiente esquema:

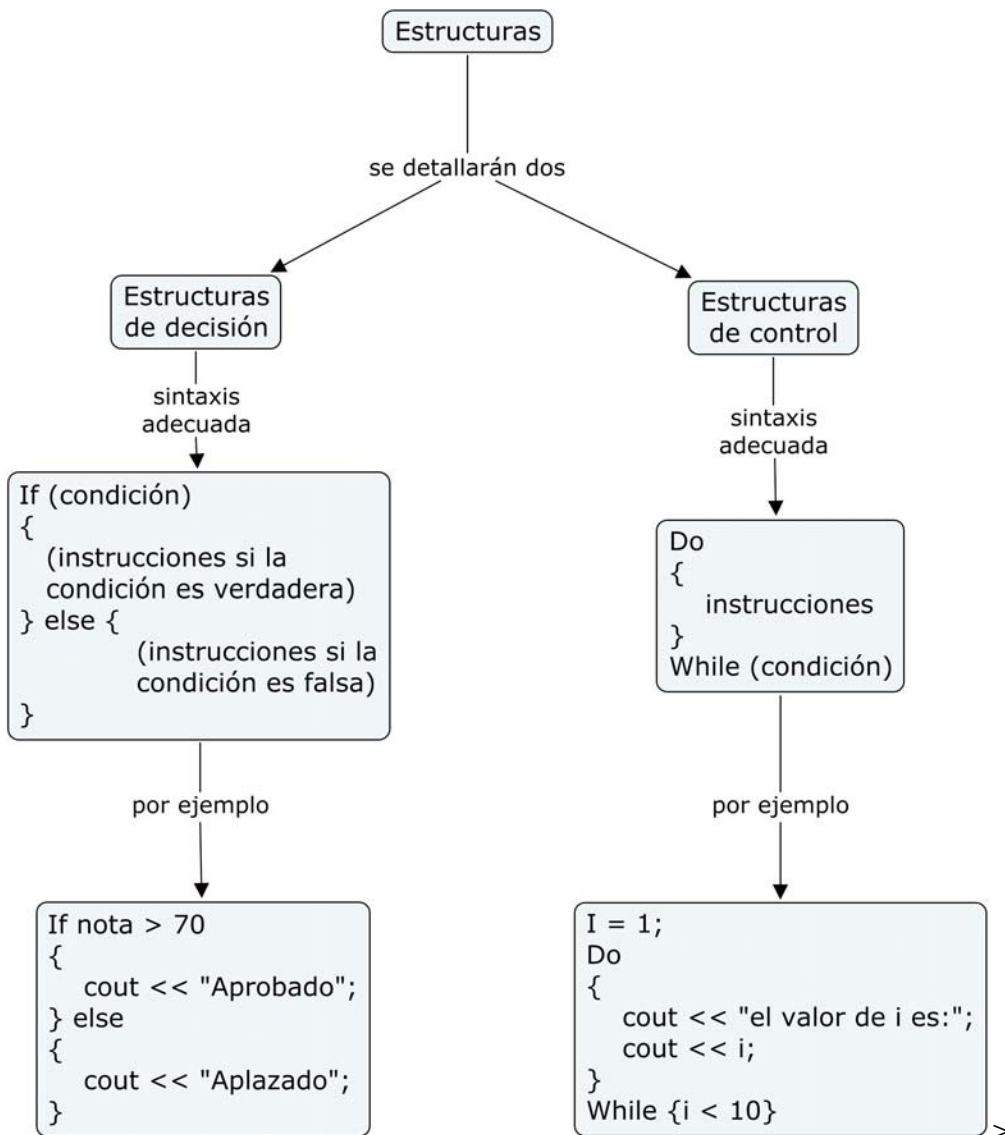


Esquema 3. Manejo de los datos en C++.

Intente determinar la necesidad de usar símbolos diferentes a los utilizados en la escritura "a mano". ¿Es importante conocer la sintaxis específica de este programa?, ¿por qué?

A partir de las páginas 344 y hasta la 348, se explica con detalle dos estructuras de programación usadas en este programa: la de decisión y la de control, aunque existen otras formas de organización usadas para trabajar en este lenguaje, en este caso, sólo interesarán estas dos.

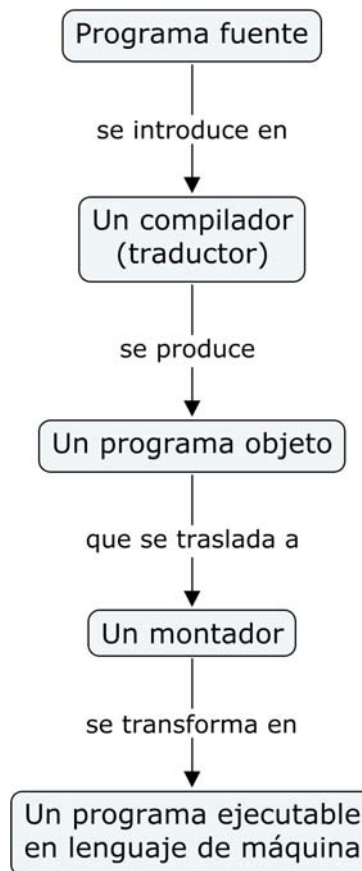
Las estructuras de decisión y las de control deben utilizarse según la necesidad que se tenga a la hora de resolver un problema. El siguiente esquema presenta un resumen de estas dos estructuras:



Esquema 4. Estructuras en C++.

Todo programa generado en C++ debe compilarse antes de su ejecución. Estos aspectos se detallan en la figura 12.01 de la página 338.

La compilación inicia al pasar el programa fuente por el compilador, se genera un programa objeto y se trasmite a un montador o enlazador; por último, se transforma en un programa ejecutable en lenguaje de máquina. En el esquema se ilustra el proceso de compilación en C++:



Esquema 5. Fases de la compilación.

Intente describir los pasos del proceso con sus propias palabras o elabore un esquema diferente que le permita representar su visión de las fases de compilación.

EJERCICIOS SUGERIDOS

Ejercicios	Página
12.06	360
12.07	361
12.08	362
Mapa mental	363
Terminología	364
Preguntas	364
Examen rápido	365

Ejercicio 12.06

Caso 6: Regla de tres simple

```
1.  /* Programa Regla de tres */
2.  #include <iostream.h>    /* Bibliotecas */
3.  #include <conio.h>
4.  void main()
5.  { /* Declaración de variables */
6.      int PrimerElemento;
7.      int SegundoElemento;
8.      int TercerElemento;
9.      int x;
10.     clrscr(); /* Limpiar pantalla */
11.     /* Ingresar los datos */
12.     cout <<"Ingrese el valor del primer elemento:";
13.     cin >>PrimerElemento;
14.     cout <<"Ingrese el valor del segundo elemento:";
15.     cin >>SegundoElemento;
16.     cout <<"Ingrese el valor del tercer elemento:";
17.     cin >>TercerElemento;
18.     /* Cálculo de los datos */
```

```
19.         x = (SegundoElemento * TercerElemento) / PrimerElemento;
20.         /* Mostrar los datos*/
21.         cout <<"El valor de X es: " << x;
22.         getch();
23.     }
```

Ejercicio 12.07

Caso 7: Cálculo del área de un polígono irregular

```
1.     /* Programa para calcular el área de un polígono */
2.     #include <iostream.h>     /* Bibliotecas */
3.     #include <conio.h>
4.     void main()
5.     {
6.     /* Declaración de variables */
7.     int lado_cuadrado;
8.     int altura_triángulo;
9.     int altura_rectángulo;
10.    int A, B, D, área_total;
11.    clrscr(); /* Limpiar pantalla */
12.    /* Ingresar los datos */
13.    cout <<"Digite el lado del cuadrado:";
14.    cin >>lado_cuadrado;
15.    cout <<"Digite la altura del triángulo:";
16.    cin >>altura_triángulo;
17.    cout <<"Digite la altura del rectángulo:";
18.    cin >>altura_rectángulo;
19.    /* Cálculo de los datos */
20.    A = lado_cuadrado^2;
21.    B = (lado_cuadrado * altura_triángulo) / 2;
22.    B = B * 3;
23.    D = lado_cuadrado * altura_rectángulo;
24.    área_total = A + B + D;
```

```
25.  /* Mostrar los datos */
26.  cout <<"El área total del polígono compuesto es:" << área_total;
27.  getch();
28.  }
```

Ejercicio 12.08

Caso 8: Determinando un descuento

```
1.  /* Programa para calcular un descuento */
2.  #include <iostream.h>    /* Bibliotecas */
3.  #include <conio.h>
4.  void main()
5.  {
6.      /*Declaración de variables */
7.      int i = 0;
8.      int TotalVentas = 0;
9.      int TipoCliente = 0;
10.     int Entrada[5];      /* Arreglo de 5 posiciones */
11.     double Des = 0;
12.     double VentaNeta = 0;
13.     clrscr(); /* Limpiar pantalla */
14.     i = 1;
15.     do /* Inicio del ciclo o loop */
16.     {
17.         /* Ingreso de los datos */
18.         cout <<"Digite el precio del artículo" << i << ":";
19.         cin >>Entrada[i - 1];
20.         TotalVentas = Entrada[i - 1] + TotalVentas;
21.         i = i + 1;
22.     }
23.     while (i <= 5); /* Fin de la estructura de control */
24.     cout <<"Digite el tipo de cliente 1 NORMAL 2 PREMIUM:";
25.     cin >>TipoCliente;
```

```
26.     if (TipoCliente == 1) /* Tipo de Cliente 1 NORMAL */
27.     if (TotalVentas <= 1 000)
28.     Des = TotalVentas * 0,15;
29.     else
30.     if (TotalVentas <= 2 000)
31.     Des = TotalVentas * 0,20;
32.     else
33.     Des = TotalVentas * 0,25;
34.     else
35.     if (TotalVentas <= 1 000)
36.     Des = TotalVentas * 0,20;
37.     else
38.     if (TotalVentas <= 2 000)
39.     Des = TotalVentas * 0,25;
40.     else
41.     Des = TotalVentas * 0,30;
42.     /* Cálculo de los datos */
43.     VentaNeta = TotalVentas – Des;
44.     /* Mostrar los datos */
45.     cout <<"El monto total de la venta es:" << VentaNeta;
46.     getch();
47. }
```

Analizar el mapa mental

El mapa mental de la página 363 de la unidad didáctica corresponde a la visión particular del autor del libro. Con los conocimientos que usted ha adquirido, hasta el momento, podría elaborar un mapa diferente a ese. Recuerde que un **mapa mental** depende de la persona que lo elaboró y las relaciones mentales que ha establecido con el resto de sus conocimientos.

Una forma de analizar esta estructura es seguir los pasos que se detallan a continuación:

- Elabore una lista de los temas, estudiados en este capítulo, que considere más importantes para el tema **Implementación de algoritmos en C++**.
- ¿Todos los temas están completamente cubiertos? Anote las preguntas que tenga al respecto.
- Verifique en el mapa mental la ubicación que el autor elaboró de cada tema y observe las relaciones que hay entre ellos y el resto del mapa.
- Compruebe que se abarcaran los objetivos de la página 335 de la unidad didáctica. ¿Qué hizo falta ahondar para lograr lo propuesto por el autor?, ¿podría usted responder a esas dudas?
- Proponga su propio mapa para este capítulo.

Terminología

En la página 364 se encuentra el índice analítico del capítulo. Lea los términos y conceptos que se incluyen e intente definirlos con sus propias palabras.

Remítase a la cotidianidad en algunas de estas palabras para ayudarse en la tarea de acrecentar el significado de cada uno.

Si tiene problemas para definir alguno de los términos, búsquelo en la página que se indica al lado de él y revise la teoría que se explica al respecto.

Preguntas

Posibles respuestas:

12.1.1. La respuesta a esta pregunta depende el tiempo que cada persona tarde en escribir el programa, depurarlo y ejecutarlo hasta obtener los resultados esperados.

12.1.2. Depende, cuando una persona comienza a programar en un lenguaje, es normal que cometa errores de sintaxis y semántica; por lo tanto, una persona novel en el lenguaje cometerá más errores y corregirá más veces.

12.1.3. En opinión del autor de esta guía: si no se conoce con detalle los pormenores del lenguaje de programación es un poco difícil.

12.1.4. Las miniespecificaciones ayudan a la programación a la hora de pasar a código C++, pero se requiere hacer ajustes para que tenga correspondencia con el lenguaje.

Examen rápido

Parte 1

- 1) (c)
- 2) (a)
- 3) (c)
- 4) (c)
- 5) (c)

Parte 2

- 6) V
- 7) F
- 8) F
- 9) V
- 10) F

Guía para el capítulo 13

Este capítulo corresponde al tema **Implementación de algoritmos Java** y los contenidos por trabajar son los siguientes:

Sumario

- Implementación de algoritmos en Java.

Estos temas los encontrará distribuidos de la siguiente manera:

Subtema	Páginas
Implementación de algoritmos en Java	368
Ambiente de desarrollo	368 – 370
Estructura básica de un programa	371
Características generales del lenguaje	371
Tipos de datos	372
Declaración de variables y arreglos	372 – 373
Operadores principales	373 – 374
Cambios de estado	374
Mostrado de datos	375
Peticiónes de datos	375 – 377
Estructuras de decisión	377 – 378
Estructuras de control	379 – 381
Compilación y ejecución de programas	381
Mapa mental del capítulo 13	397

Comentarios generales

Al igual que con el C++ para el Java se comenzará al señalar algunas generalidades:

Paradigma	Orientado a objetos
Apareció en:	1990
Diseñado por:	Sun Microsystems
Tipo de dato:	Fuerte, estático
Implementaciones:	Numerosas
Influido por:	Objective-C, C++, Smalltalk, Eiffel
Ha influido:	C#, J#, JavaScript
Sistema operativo:	Multiplataforma
Licencia de <i>software</i>:	GPL / Java Community Process

Java es un lenguaje de programación orientado a objetos. Fue desarrollado por la empresa Sun Microsystems a principios de los años noventa. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como **punteros** (o apuntadores). Un puntero es una variable manipulable que se refiere a una región de la memoria. Java es más utilizado hoy, en comparación con C++, sobre todo para el desarrollo de aplicaciones para Internet.

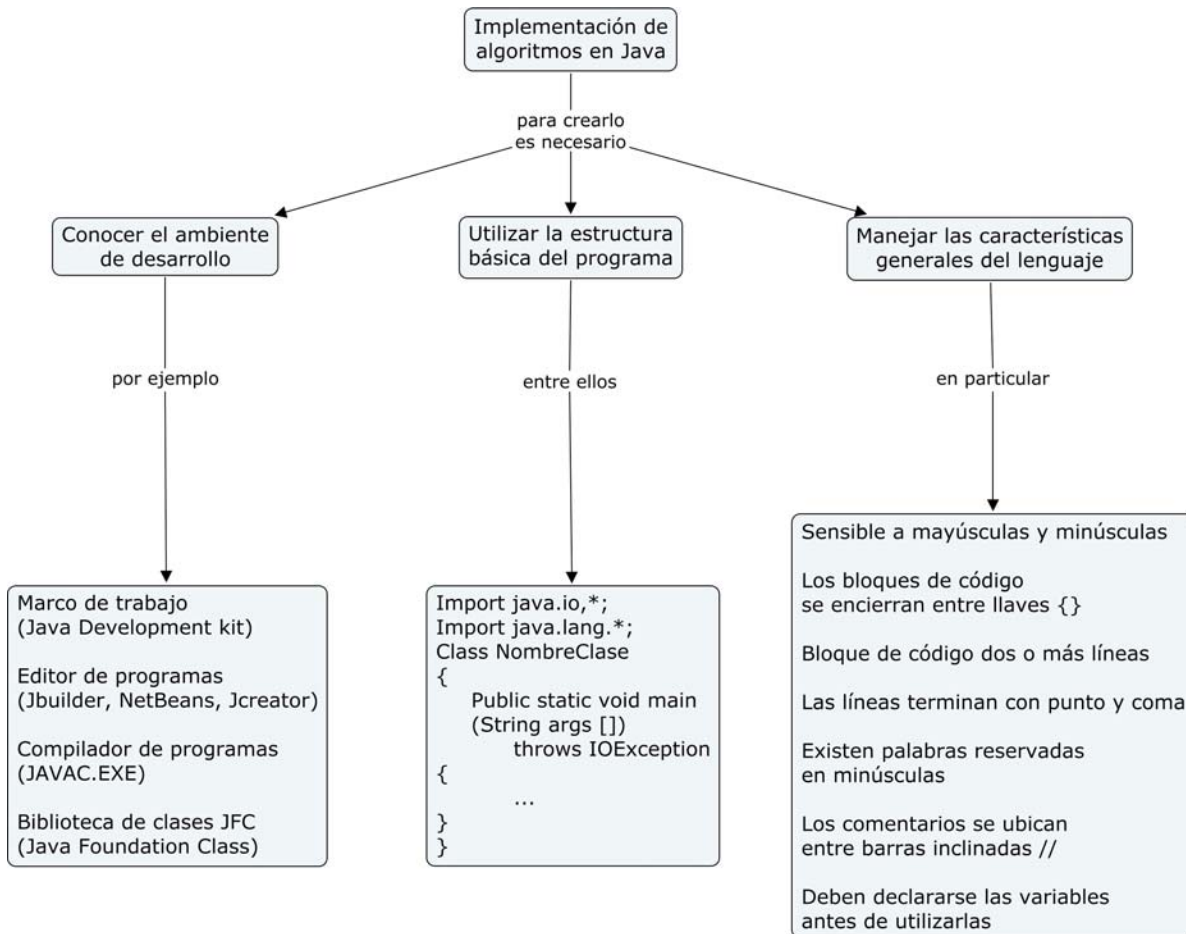
Las aplicaciones Java están compiladas en un *bytecode* (es un código intermedio, más abstracto que el código máquina), aunque la compilación en código máquina nativo también es posible.

El Java posee una **máquina virtual**, que es un *software* destinado a crear un entorno entre la plataforma de la computadora y el usuario final. Permite que el lenguaje de programación se ejecute sin problemas.

La implementación original y de referencia del compilador (la **máquina virtual** y las librerías de clases de Java) fueron desarrollados por la misma empresa, en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process (El Proceso de la Comunidad Java, o *Java Community Process*, establecido en 1998. Este es un procedimiento formalizado que permite a las empresas y programadores involucrarse en la definición de futuras versiones y características de la plataforma).

Entre noviembre de 2006 y mayo de 2007, la empresa Sun Microsystems liberó la mayor parte de las tecnologías Java. Esta liberación se desarrolló de acuerdo con las especificaciones del Java Community Process, de tal manera que casi todo el lenguaje Java de Sun es ahora **software libre**. Valga aclarar que la **biblioteca de clases** de Sun, se requiere para ejecutar los programas Java y no es libre. El *software* libre puede bajarse de Internet.

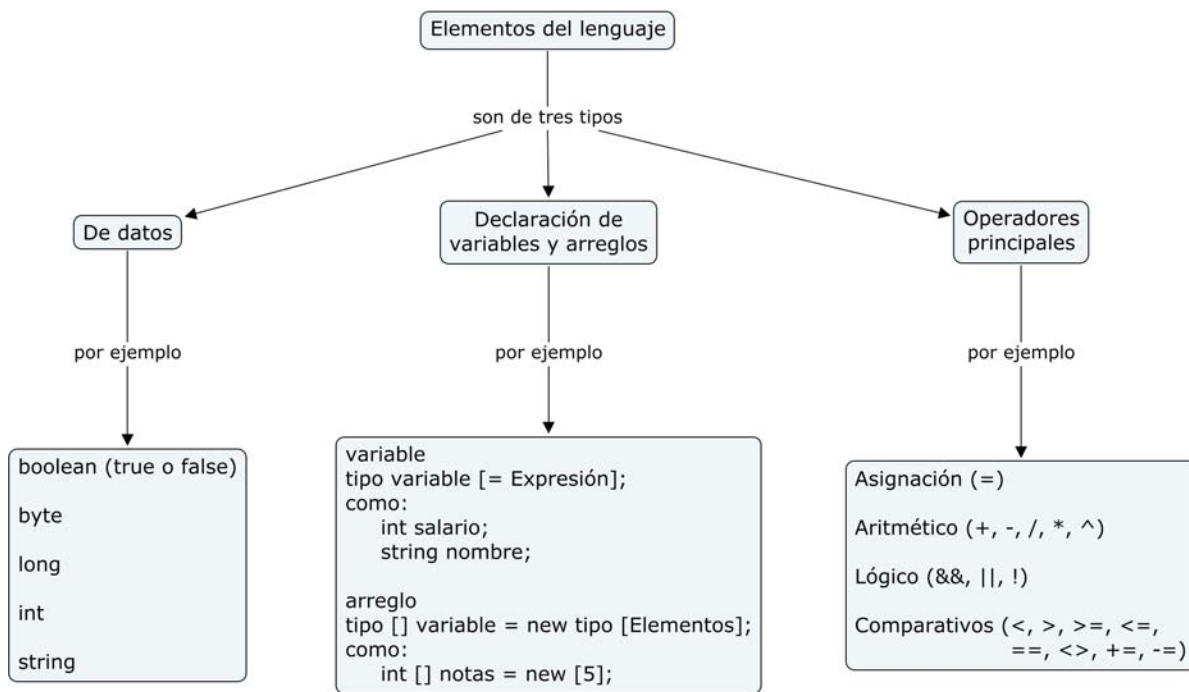
En su unidad didáctica, de la página 368 hasta la 370, encontrará una explicación acerca del ambiente de desarrollo, la estructura básica de un programa y las características generales del lenguaje. A continuación, se presenta un esquema de este compendio.



Esquema 6. Implementación de algoritmos en Java.

¿Comprende estos términos? Verifique en la unidad didáctica para aclarar los aspectos que no tenga claros.

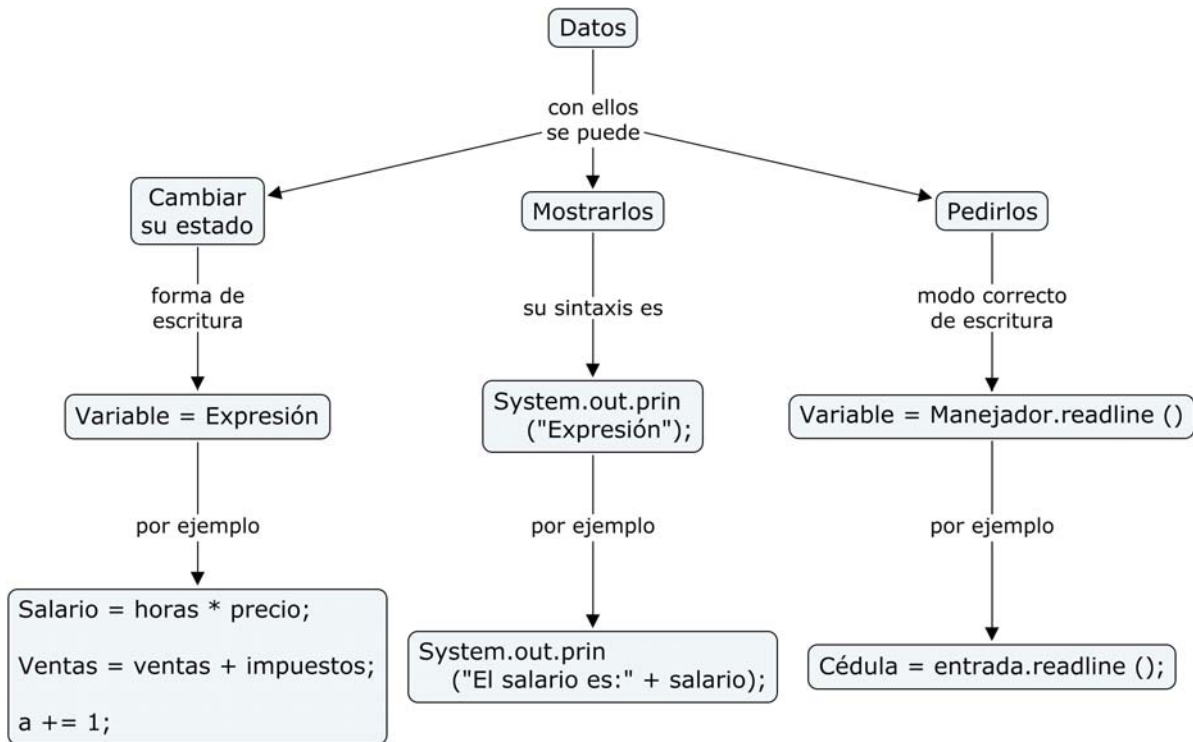
Como todo lenguaje de programación, Java tiene sus elementos particulares. Estos elementos forman parte de la sintaxis preestablecida. Esta forma adecuada de escritura debe respetarse para que las expresiones y las estructuras que se utilicen en la resolución de un problema se lean correctamente. El detalle de este tema lo encontrará de la página 372 hasta la 374.



Esquema 7. Elementos del lenguaje en Java.

Compare los elementos del lenguaje de Java con los elementos del lenguaje presentados para C++. ¿Encuentra diferencias llamativas? Enumérelas y determine en qué elementos son dispares.

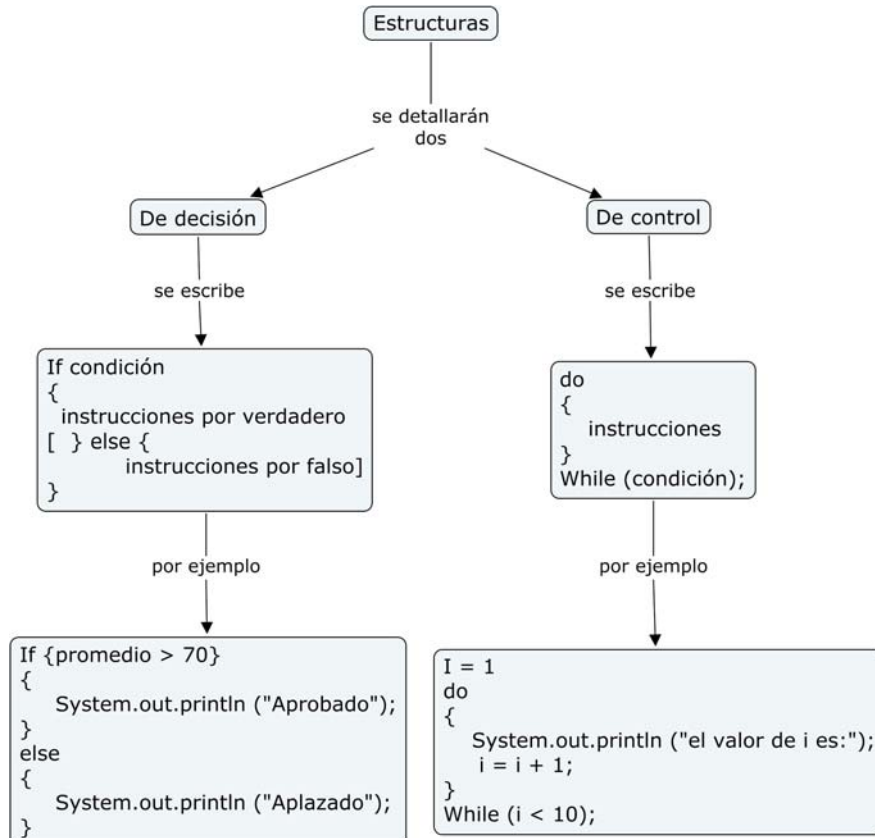
A partir de la página 374 y hasta la 377, se detalla la forma de utilizar los datos en el lenguaje de programación Java. Seguidamente se presenta un esquema de esta temática:



Esquema 8. Manejo de los datos en Java.

¿Es clara la sintaxis que se usa para trabajar en Java? Defina, con sus propias palabras, las reglas adecuadas para trabajar en este lenguaje.

Al igual que el C++, el Java cuenta con dos estructuras de programación: la de decisión y la de control. Esta información la encontrará en las páginas 376 hasta la 381. En este lenguaje existen otras estructuras pero, por el momento, sólo se estudiarán estas dos. A continuación, se presenta un esquema de estas dos estructuras:

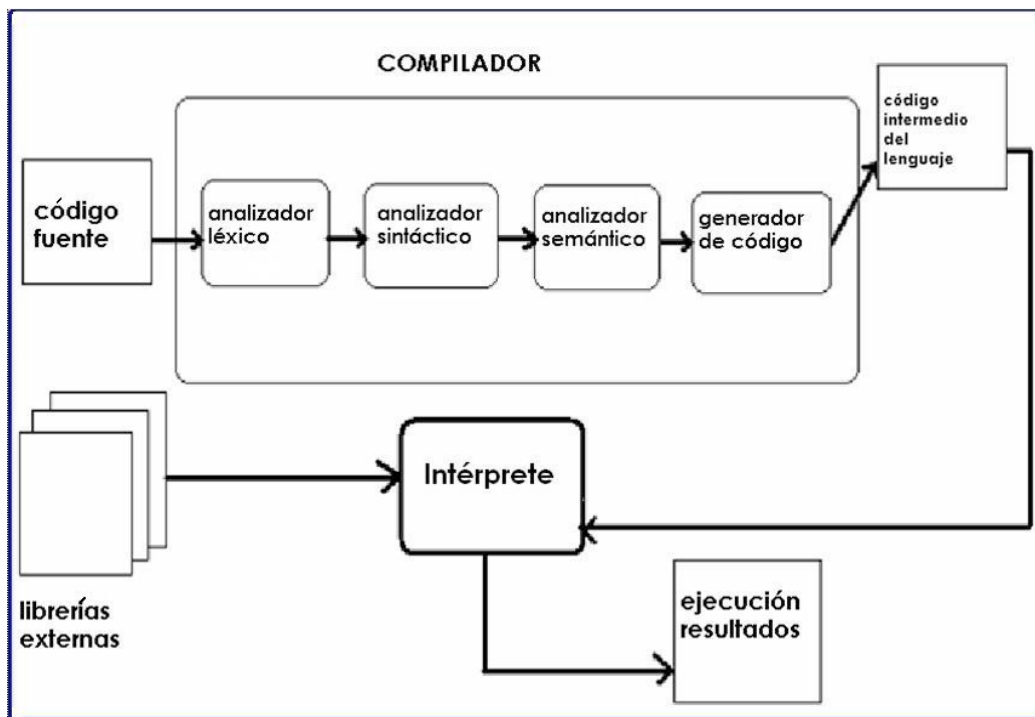


Esquema 9. Estructuras en Java.

De la misma manera que en el C++, todos los programas realizados en Java deben compilarse antes de probar su efectividad. En este lenguaje, el código primero se compila con el compilador de Java **javac**. Éste genera un código virtual llamado **ByteCode**. Este código es luego interpretado por una **máquina virtual** de Java (o JVM de Java Virtual Machine), la JVM debe estar instalada para que se ejecute en la máquina

en la que se desee el código compilado con javac. En la página 370 puede observar la figura 13.01, en la cual se detalla este proceso.

Este lenguaje utiliza un **intérprete híbrido** que es un sistema que realiza una compilación "parcial" a un código intermedio, seguida de un proceso de interpretación. ¿Cómo se realiza este proceso? Esto se ilustra en la siguiente figura:



Esquema 10. Proceso de compilación.

Después de estudiar este capítulo, resuelva los ejercicios sugeridos a continuación.

EJERCICIOS SUGERIDOS

Ejercicios	Página
13.06	394

13.07	394
13.08	396
Mapa mental	397
Terminología	398
Preguntas	398
Examen rápido	399

Ejercicio 13.06

Caso 6: Regla de tres simple

```

1.  import java.io.*;
2.  import java.lang.*;
3.  class TresSimple
4.  { /* Inicia la clase */
5.      public static void main(String args[])
6.          throws java.io.IOException
7.      {
8.          BufferedReader entrada = new
          BufferedReader (new InputStreamReader(System.in));
9.          /* Declaración de variables */
10.         double primerElemento;
11.         double segundoElemento;
12.         double tercerElemento;
13.         double x;
14.         /* Ingreso de los datos */
15.         System.out.print("Ingrese el primer elemento:");
16.         primerElemento =
          Double.parseDouble(entrada.readLine());
17.         System.out.print("Ingrese el segundo elemento:");
18.         segundoElemento =
          Double.parseDouble(entrada.readLine());
19.         System.out.print("Ingrese el tercer elemento:");

```

```
20.         tercerElemento =
           Double.parseDouble(entrada.readLine());
21.         /*Cálculo de datos*/
22.         x = (segundoElemento * tercerElemento) /
           primerElemento;
23.         /*Salida de resultados*/
24.         System.out.println ("El valor de la x es:" + x);
25.     }
26. } /*fin de la clase*/
```

Ejercicio 13.07

Caso 7: Cálculo del área de un polígono irregular

```
1.     import java.io.*;
2.     import java.lang.*;
3.     class Poligono
4.     { /* Inicia la clase */
5.         public static void main(String args[])
6.             throws java.io.IOException
7.         {
8.             BufferedReader entrada = new BufferedReader (new
           InputStreamReader(System.in));
9.             /* Declaración de variables */
10.            int ladoCuadrado;
11.            int alturaTriángulo;
12.            int alturaRectángulo;
13.            int A, B, D, áreaTotal;
14.            /* Ingreso de los datos */
15.            System.out.print("Digite el lado del cuadrado:");
16.            ladoCuadrado = Integer.parseInt(entrada.readLine());
17.            System.out.print("Digite la altura del triángulo:");
18.            alturaTriángulo = Integer.parseInt(entrada.readLine());
19.            System.out.print("Digite la altura del rectángulo:");
```

```
20.     alturaRectángulo = Integer.parseInt(entrada.readLine());
21.     /* Cálculo de datos */
22.     A = ladoCuadrado ^ 2;
23.     B = ladoCuadrado * alturaTriángulo / 2;
24.     B = B * 3;
25.     D = ladoCuadrado * alturaRectángulo;
26.     areaTotal = A + B + D;
27.     /* Salida de resultados */
28.     System.out.println ("El área del polígono compuesto es:" +
        áreaTotal);
29.     }
30. }     /* fin de la clase */
```

Ejercicio 13.08

Caso 8: Determinando un descuento

```
1.     import java.io.*;
2.     import java.lang.*;
3.     class Descuento
4.     { /* Inicia la clase */
5.     public static void main(String args[])
6.     throws java.io.IOException
7.     {
8.     /* Declaración de variables y vector */
9.     int i, TotalVentas, TipoCliente;
10.    int[] Ventas = new int[5]; /* Arreglo de 5 posiciones */
11.    double Des, VentaNeta;
12.    BufferedReader entrada = new BufferedReader (new
        InputStreamReader(System.in));
13.    i = 1;
14.    TotalVentas = 0;
15.    Des = 0;
16.    do /* Inicio de la estructura de control o ciclo loop */
```

```
17.  {
18.  /* Ingreso de los datos */
19.  System.out.print("Digite el precio del artículo: " + i);
20.  Ventas[i - 1] = Integer.parseInt(entrada.readLine());
21.  TotalVentas = TotalVentas + Ventas[i - 1];
22.  i = i + 1;
23.  }
24.  while (i <= 5); /* fin de la estructura de control o ciclo loop */
25.  System.out.print("Digite el tipo de cliente: un 1 si es normal o un 2 si es
    premium:");
26.  TipoCliente = Integer.parseInt(entrada.readLine());
27.  /* Estructura de decisión */
28.  if ( TipoCliente == 1 ) { // Tipo de cliente 1 NORMAL
29.      if ( TotalVentas < 1 001 )
30.          Des = TotalVentas * 0,15;
31.      else if ( TotalVentas <= 2 000 )
32.          Des = TotalVentas * 0,20;
33.      else
34.          Des = TotalVentas * 0,25;
35.  }
36.  else { // Tipo de Cliente 2 PREMIUN
37.      if ( TotalVentas < 1 001 )
38.          Des = TotalVentas * 0,20;
39.      else if ( TotalVentas <= 2 000 )
40.          Des = TotalVentas * 0,25;
41.      else
42.          Des = TotalVentas * 0,30;
43.  }
44.  /* Cálculo de datos */
45.  VentaNeta = TotalVentas - Des;
46.  /* Salida de resultados */
47.  System.out.println ("El monto total de la venta es: " + VentaNeta);
```

```
48.    }  
49.    } /* fin de la clase */
```

Analizar el mapa mental

En la página 397 de la unidad didáctica encontrará el mapa mental del capítulo. Como es de su conocimiento, esta estructura corresponde a la visión particular del autor del libro. Con lo estudiado en este tema, los anteriores y su experiencia en el campo, podría elaborar un mapa diferente a ese.

Para analizar esta estructura siga los pasos que se detallan a continuación:

- Elabore una lista de los temas estudiados en este capítulo que considere más importantes para el tema Implementación de algoritmos en Java. Compárelos con la lista elaborada para C++.
- ¿Todos los temas están completamente cubiertos? ¿Aún persiste alguna duda sobre ellos? Anote las preguntas que tenga al respecto.
- Verifique, en el mapa mental, la ubicación que el autor elaboró de cada tema y observe las relaciones que hay entre ellos y el resto del mapa. ¿Incorporaría usted algún cambio?
- Compruebe que se abarcaran los objetivos de la página 367 de la unidad didáctica. ¿Qué falta por ahondar para lograr lo propuesto por el autor? ¿Podría responder a esas dudas?
- Proponga su propio mapa para este capítulo.

Terminología

En la página 398 de la unidad didáctica, se encuentra el índice analítico del capítulo. Lea los términos y conceptos que se incluyen e intente definirlos con sus propias palabras. Maneje los usos cotidianos de algunos de estos términos para ayudarse en la tarea de acrecentar el significado de cada uno.

Si tiene problemas para definir alguno de los términos, búsquelo en la página que se indica al lado de él y revise la teoría que se explica al respecto.

Preguntas

A continuación se presentan posibles respuestas a las preguntas planteadas:

13.1.1. El tiempo utilizado para resolver los problemas depende del dominio de las estructuras del lenguaje de programación que tenga el usuario y de su correcta aplicación.

13.1.2. Este tiempo varía según el programador y la experiencia que tenga en el uso del programa. Probablemente, un usuario novel deba corregir más errores de sintaxis que un programador más habituado a trabajar con este lenguaje.

13.1.3. Se vuelve un poco más fácil con la experiencia del uso del lenguaje C++, ya que tiene mucha similitud.

13.1.4. Las miniespecificaciones sirven para entender como se representan los problemas en un lenguaje de programación; sí es necesario modificarlas, puesto que se depende de la sintaxis del lenguaje.

13.2. A continuación se anotan dos aspectos, uno de cada lenguaje:

C++: cuesta un poco más por ser el primer lenguaje que se utilizó.

Java: se debe tener más cuidado con las especificaciones propias de la sintaxis.

13.3. El lenguaje es irrelevante. El buen programador esta capacitado para convertir los algoritmos que diseñe a los lenguajes que se le permita aprender.

Examen rápido

2) (c)	6) V
3) (a)	7) F
4) (b)	8) V
5) (b)	9) F
Parte 2	10) V

Los dos lenguajes de programación estudiados pertenecen a los más actuales que encontrará en el mercado. El uso de estos lenguajes y del razonamiento necesario para programar en ellos, le abre muchas puertas en el área de la Informática. Para afianzar estos temas es preciso que pueda trabajarlos con soltura. Se le propone la experiencia de proponerse un problema nuevo y buscar la forma de programar una solución para él; por ejemplo: determinar si tres números positivos corresponden a las medidas de los lados de un triángulo, o no. Utilice la siguiente guía para revisar su trabajo:

- Programe en C++ y Java el procedimiento que permita responder al problema.
- ¿Cuál de las dos programaciones le resultó más sencilla?
- ¿Qué ventajas puede determinar del uso de C++, ¿y desventajas?
- ¿Qué ventajas puede determinar del uso de Java?, ¿y desventajas?

Como habrá notado, con este quinto tema se culmina el curso **Lógica para Computación**. Es necesario que usted comprenda que el razonamiento lógico es básico para el trabajo en las ciencias de la Computación y la Informática. Este tipo de razonamiento le permite plantear y diseñar algoritmos que, a la postre, le permiten resolver los problemas propuestos.

¿Cómo relaciona usted los cinco temas estudiados en este curso?

La respuesta a esta pregunta le permitirá integrar los temas de estudio a su proceso de pensamiento. Esta asimilación será la clave para que usted emplee estos conocimientos tanto en cursos posteriores como en su ámbito laboral.

Referencias utilizadas por el autor

- ALEGSA, *Diccionario de Informática, Internet y tecnologías*. Extraído durante los meses de octubre y noviembre de 2007 desde:
<<http://www.alegsa.com.ar/Diccionario/diccionario.php>>.
- ATRIA, JOSÉ TULLIO, *Lógica matemática*. Facultad de Ciencias de la Educación de la Universidad de Carabobo. Extraído en octubre de 2007 desde:
<<http://profatria.tripod.com/logicamatematica/>>.
- BAASE, SARA y GELDER, ALLEN (2002). *Algoritmos computacionales. Introducción al análisis y diseño*. Tercera edición, México D.F: Pearson Educación.
- BECK, KENT (2002). *Una explicación de la programación extrema. Aceptar el cambio*. Madrid: Pearson Educación. S.A.
- BLOW, LISA (2000). *Fundamentos de Computación*. México: Pearson Educación.
- Calzado Fernández, Francisco José, *Aprende lógica*. Extraído en octubre de 2007 desde:
<<http://w3.cnice.mec.es/eos/MaterialesEducativos/mem2003/logica/>>.
- COSTA RICA (1997), *Ley No. 6683, Ley de derecho de autor y derechos conexos*. Extraído en octubre de 2007 desde: <www.asamblealegislativa.go.cr>.
- DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN, *2.6.1 Operadores aritméticos*. Instituto Tecnológico de La Paz, Baja California Sur. Extraído en noviembre de 2007 desde:
<http://sistemas.itlp.edu.mx/tutoriales/pascal/u2_2_6_1.html>.
- Diseño de páginas web (textos sobre) en
Diseño de páginas web, alojamiento, posicionamiento y dominios. Extraído en noviembre de 2007 desde:
<<http://www.masadelante.com/faq-servidor.htm>>.
- GUIARTE MULTIMEDIA S.L., *Operadores y operandos*. Extraído en octubre de 2007 desde: <<http://www.desarrolloweb.com/articulos/2165.php>>.
- HIGUERA TOLEDANO, MARÍA TERESA, *Ejecución y compilación de un programa*. Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid. Extraído en octubre de 2007 desde:
<<http://asds.dacya.ucm.es/teresa/FC/T6.pdf>>.

- INSTITUTO SUPERIOR DE MONTE CRISTO, *Análisis de caja negra*. Extraído en noviembre de 2007 desde:
<http://www.cursosaulavirtual.com.ar/moodle/file.php/3/Analisis_de_Caja_Negra.doc>.
- KUHN, THOMAS (1971). *La Estructura de las Revoluciones Científicas 1*. Madrid: Fondo de Cultura Económica S.L.
- MATPEC, *Desde 0 – conexión a Internet*. Extraído en noviembre de 2007 desde:
<<http://www.matpec.com.ar/desde0/desde0-2-dial-up.htm>>.
- MARTIN, JOHN (2003). *Lenguajes formales y teoría de la computación*. Tercera edición, México: McGraw-Hill Interamericana.
- PINTO ALARCÓN, MÓNICA, *Intérpretes* [PDF]. Lenguajes y Ciencias de la Computación, Universidad de Málaga. Extraído en noviembre de 2007 desde:
<<http://www.lcc.uma.es/~pinto/apuntes/software%20de%20sistemas/Intérpretes.pdf>>.
- PRATT, TERRENCE Y ZELKOWITZ, MARVIN (2000). *Lenguajes de programación. Diseño e Implementación*. Tercera edición, México D.F.: Prentice Hall Hispanoamericana S.A.
- TUCKER, ALLEN y NOONAN, ROBERT (2003). *Lenguajes de programación. Principios y paradigmas*. España: McGraw-Hill Interamericana.
- UNIVERSIDAD NACIONAL DE COLOMBIA, *3.5. Pruebas de escritorio*. Extraído en noviembre de 2007 desde:
<http://www.virtual.unal.edu.co/cursos/ingenieria/2001839/modulo1/cap_07/leccion_5.htm>.
- VALVERDE, AGUSTÍN, *Aula de lógica computacional*. Universidad de Málaga. Extraído durante los meses de octubre y noviembre de 2007 desde:
<<http://www.matap.uma.es/~valverde/Logica/index.html>>.

Referencias electrónicas de consulta para el estudiante

Actividades de juegos de lógica hoja 8:

<http://actividadesinfor.webcindario.com/jle9.htm>

Borland Software Corporation: <http://www.borland.com/downloads/index.html>

Buscadores: www.google.com, www.altavista.com

Dfd ®: <http://es.geocities.com/edisonm172/descarga.html>

Diccionario de Informática, Internet y tecnologías:

<http://www.alegsa.com.ar/Diccionario/diccionario.php>.

EDGE Diagrammer: <http://es.brothersoft.com/EDGE-Diagrammer-download-102760.html>

Institute for human and machine cognition: <http://cmap.ihmc.us/>

Java Technology, Sun Microsystems, Inc: <http://www.sun.com/java/>

Lógica 3D, juego: <http://biboz.net/juegos/logica3d/>

Recorrido en video de Visio 2003:

<http://www.microsoft.com/latam/office/visio/prodinfo/videos.msp>

SmartDraw: <http://www.smartdraw.com/specials/diagramasdeflujo.asp?id=45679>

Wikipedia, la enciclopedia libre: <http://es.wikipedia.org>



**Derechos reservados
Universidad Estatal a Distancia de Costa Rica**