

## INTRODUCCIÓN A LA PROGRAMACIÓN

### Conceptos Básicos

#### Objetivo

Aquí se presentan algunas definiciones básicas que servirán de conceptos introductorios. Muchas de ellas son conocidas total o parcialmente de forma instintiva y se discutirán en el contexto más amplio de software y sistemas de software aplicados a la resolución de problemas en el mundo real.

El eje temático gira alrededor del concepto de modelo como representación del ambiente, los datos y los procesos en el mundo real y de la noción de programa como expresión ordenada, completa y correcta de la especificación de una solución (algoritmo) computable de dicho problema.

Simultáneamente con las definiciones rigurosas, se presentan algunos ejemplos sencillos de abstracciones de problema del mundo real y su modelización.

#### Definiciones :

La informática es la ciencia que estudia el análisis y resolución de problemas utilizando computadores.

La palabra ciencia se relaciona con una metodología fundamentada y racional para el estudio y resolución de los problemas. En este sentido la Informática se vincula especialmente con la Matemática y la Ingeniería.

La resolución de problemas utilizando las herramientas informáticas puede tener aplicaciones en áreas muy diferentes tales como biología, comercio, control industrial, administración, robótica, educación, arquitectura, diseño, etc.

Los temas propios de la ciencia Informática abarcan aspectos tales como la arquitectura física y lógica de las computadoras, las metodologías de análisis y diseño de sistemas de software, los lenguajes de programación, los sistemas operativos, la inteligencia artificial, los sistemas de comunicaciones, los sistemas de tiempo real, el diseño y aplicación de bases de datos, etc.

Un computador es una máquina digital y sincrónica, con cierta capacidad de cálculo numérico y lógico, controlada por un programa almacenado y con posibilidad de comunicación con el mundo exterior.

En esta definición -que es incompleta y perfectible- se establece que una computadora moderna es esencialmente, digital; esto significa que las señales eléctricas que maneja, y la información que procesa se representa en forma discreta, normalmente con valores binarios (0 o 1).

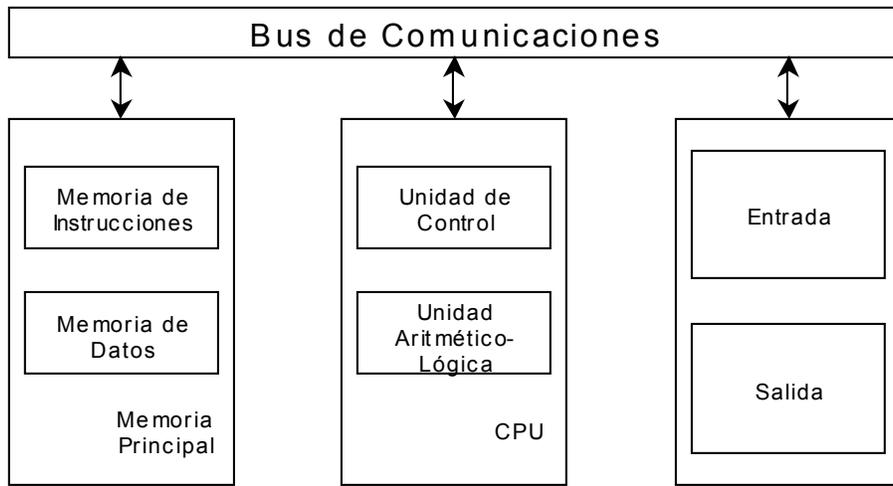
Además, se afirma que es sincrónica, es decir, que realiza las operaciones coordinadas por un reloj central que envía pulsos de sincronismo a todos los elementos que componen el computador. Esto significa que todas las operaciones internas se realizan en instantes de tiempo predefinidos y coordinados con el reloj.

Internamente los computadores tienen cierta capacidad de cálculo numérico y lógico, en un subsistema conocido como Unidad Aritmético-Lógica. Normalmente, las operaciones que pueden realizarse en ella son muy simples (suma, or, and, comparaciones).

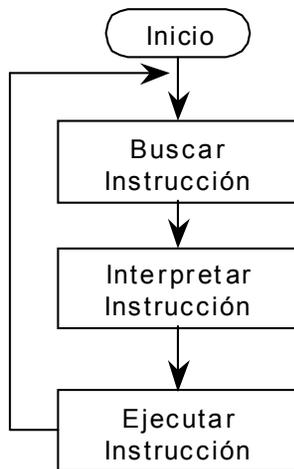
La frase "controlada por un programa" es quizás la más importante de esta definición. Significa que internamente se tienen órdenes o instrucciones almacenadas, que el computador podrá leer, interpretar y ejecutar ordenadamente.

Por último, se requiere que el computador pueda vincularse con el mundo real (por ejemplo, a través de un teclado como entrada de información y una pantalla como salida de la misma).

Esquemáticamente, un computador puede representarse a través de sus bloques funcionales más importantes:



El funcionamiento de un computador se puede sintetizar usando un esquema en el que se representa una secuencia infinita de pasos:



- Buscar la próxima instrucción a ejecutar de la memoria de instrucciones.
- Interpretar qué hacer con la instrucción en la Unidad de Control.
- Ejecutar las operaciones interpretadas por la unidad de control, utilizando la ALU, de ser necesario. Estas operaciones pueden comprender lectura/escritura de la memoria de datos o entrada/salida.

La representación esquemática del computador incluye la división conceptual de la memoria central en memoria de instrucciones (Mi) donde residen las órdenes que la computadora debe interpretar y ejecutar y memoria de datos (Md) donde se almacena la información con la cual la computadora realizará los procesos (cálculos, decisiones, actualizaciones) que sean necesarios para la resolución del problema.

El bloque marcado como entrada/salida (e-s) representa los dispositivos que permiten la comunicación con el mundo real. Por ejemplo, el controlador de video que vincula el procesador central de la computadora con la pantalla o el circuito controlador de multimedia que permite tener salida por un parlante o entrada por un micrófono.

Las líneas de comunicación indicadas como bus de comunicación normalmente permiten el paso de tres grandes categorías de información: direcciones, datos y control. En el esquema simplificado se acepta que estas líneas permiten la comunicación interna y externa de datos, direcciones y señales de control.

Por último, tradicionalmente la combinación de la unidad de control y la unidad aritmético-lógica, se llama unidad central de procesamiento, que en las computadoras personales está representada por el microprocesador (por ejemplo, 486, 586, Pentium, etc.).

Un programa es un conjunto de instrucciones, ejecutables sobre una computadora, que permite cumplir una función específica.

En esta definición se asocia al programa con una función determinada o requerimiento a satisfacer por la ejecución del conjunto de instrucciones que lo forman.

Ejemplos de programa son aquellos que permiten emitir los cheques de pago de sueldos de una empresa, los que realizan el control de procesos químicos en una fábrica, etc.

Normalmente, los programas alcanzan su función objetivo en un tiempo finito. Sin embargo, se pueden tener aplicaciones (por ejemplo, el programa de control de un sistema de alarmas) que se ejecutan indefinidamente, porque poseen un requerimiento de tiempo infinito.

Un programa sin errores que se ejecuta sobre un computador, que comienza y termina, puede ser incorrecto si no cumple con lo solicitado en los requerimientos.

Un dato es una representación de un objeto del mundo real mediante la cual se pueden modelizar aspectos de un problema que se desea resolver con un programa sobre un computador.

Es muy difícil representar datos reales en un computador (definido como una máquina digital binaria, es decir, solo con capacidad de almacenar ceros y unos). Los programas de simulación o juegos que incluyen ciudades, personajes y distintas acciones ¿qué son dentro del computador? Seguramente algo menos real que una los verdaderos objetos.

Representar datos, aun los más simples como los dígitos decimales, una letra, un nombre o un color requiere una transformación desde el mundo real a alguna forma de representación binaria que pueda ser interpretada por el computador.

Datos más complejos como una imagen, una canción o la trayectoria de un avión también son representados en forma binaria. Sin embargo, para establecer la forma de modelizarlos e interpretarlos será necesario un análisis cuidadoso por parte de quien escriba un programa que los utilice.

A *priori* las dos tareas más importantes con las que se enfrenta quien debe escribir un programa para resolver un problema sobre una computadora son:

- definir el conjunto de instrucciones cuya ejecución ordenada conduce a la solución
- elegir la representación adecuada de los datos del problema.

En síntesis, los computadores son una poderosa herramienta para la resolución de problemas, cuya potencial utilidad depende de la capacidad de programación de soluciones adecuadas a cada problema.

La función esencial de un ingeniero de sistemas o un especialista informático es explotar la potencialidad de los computadores (velocidad, exactitud, confiabilidad) para resolver situaciones del mundo real.

Para esto debe analizar los problemas del mundo real, ser capaz de sintetizar sus aspectos principales y poder especificar la función objetivo que se desea. Posteriormente, debe expresar la solución en forma de programa, manejando los datos del mundo real mediante una representación válida para un computador.

### **Modelaje de problemas del mundo real**

El mundo real es naturalmente complejo y en muchas ocasiones los problemas a resolver resultan difíciles de sintetizar. Imaginemos, por ejemplo, el caso de incorporar un computadora en una empresa cualquiera. Posiblemente el dueño entienda que su objetivo es aumentar las ganancias del negocio a través de la inversión que significa incorporar una máquina y un sistema de software.

Sin embargo, no se puede asegurar que esta inversión se traduzca directamente en mayor rentabilidad. Tendremos que analizar cuáles aspectos de la operación normal del negocio (emisión de remisiones y facturas, control de cuentas corrientes, seguimiento de cobros a clientes, liquidación de impuestos, control de bancos, etc.) pueden requerir una automatización y en qué casos y de qué forma esta automatización puede traducirse en un beneficio (en tiempo, en empleados, en calidad del servicio al cliente, en confiabilidad, etc.).

El proceso de análisis del mundo real para interpretar los aspectos esenciales de un problema y expresarlo en términos precisos se denomina abstracción..

Abstraer un problema del mundo real y simplificar su expresión, tratando de encontrar los aspectos principales que se pueden resolver (requerimientos) los datos que se han de procesar y el contexto del problema se denomina modelaje.

Hay modelos con diferentes grados de exactitud que dependerán de los objetivos a cumplir. Generalmente, es suficiente la utilización de modelos simples para modelar muchas de las situaciones del mundo real, no se modela exactamente los objetos y situaciones involucrados en cada uno de los problemas a resolver.

A partir de un modelo esencial del problema a resolver podemos analizar sus diferencias con respecto a los requerimientos definidos, ajustando el modelo, si fuera necesario.

Hasta ahora se han definido instrucción, programa y dato. Es interesante discutir dos términos que aparecerán habitualmente en la escritura de programas: precondition y postcondition.

Una precondition es una información que se conoce como verdadera ante de iniciar el programa.

En cualquier modelo, siempre se suponen unas condiciones iniciales, que deben cumplirse para que el modelo se ajuste a la representación que se hace, esas precondiciones, limitan el modelo como tal y lo que puede esperarse de él.

Una poscondición es una información que deberá ser verdadera al concluir un programa, si se cumple adecuadamente el requerimiento pedido.

#### Modelos conocidos del mundo real

Nuestro conocimiento del mundo, lo adquirimos a través de modelos, estudiados a lo largo de la vida escolar, todos los modelos tienen algunas inexactitudes aceptables:

- En el *lanzamiento parabólico* se supone que el objeto es lanzado al vacío (es decir, sin tener en cuenta la resistencia del aire); no se considera la posibilidad de que haya viento en cualquier dirección; se considera la fuerza de la gravedad constante y no se discuten posibles cambios de altitud en el terreno. Las fórmulas que caracterizan el alcance y la altura máxima son esencialmente correctas, aunque si se realizan mediciones reales utilizando, por ejemplo, un cañón y proyectiles, posiblemente haya errores menores en el alcance de los mismos y en la altura máxima alcanzada.
- En el *movimiento rectilíneo uniformemente acelerado*, se supone que el móvil no sufre la resistencia del aire; que el terreno es perfectamente horizontal; que no hay una parte de la fuerza que se "consume" en vencer el rozamiento del piso y que la forma del móvil no tiene importancia. Las curvas de recorrido y velocidad son conceptualmente correctas, pero deberían corregirse si, por ejemplo, el móvil se desliza sobre arena o en pendiente.
- Por último, cuando se estudian problemas de colisiones, generalmente se ignora la deformación que provoca el choque en esos objetos y el gasto de energía que ello significa. En el modelo ideal estudiado en Física, las bolas de billar no se detendrían nunca; sin embargo, sabemos que sí lo hacen debido al consumo de energía de los choques y al rozamiento sobre el paño de la mesa de billar. Nuevamente, el modelo esencial sirve para comprender lo básico del problema y si se pretende una mayor exactitud se debe refinar y ajustar, acercándolo al mundo real, a costo de incrementar la complejidad.

#### Especificación de los problemas del mundo real

Hasta aquí se ha expresado que los problemas del mundo real son complejos y que, por ese motivo, es necesario buscar sus aspectos esenciales para representarlos mediante un modelo. Además, los objetivos buscados al enfrentar el problema deben ser expresados rigurosamente.

El proceso de analizar los problemas del mundo real y determinar en forma clara y concreta el objetivo que se desea se denomina especificación.

Especificar un problema significa establecer en forma unívoca el contexto, las precondiciones y el resultado esperado, del cual se derivan las poscondiciones.

Naturalmente, para tener especificaciones unívocas es deseable poder expresar los problemas en un lenguaje correcto, formalmente riguroso y que no permita múltiples interpretaciones. Siempre se debe pensar que el modelo debe ser interpretado sin errores por un autómata, como un computador. Lo ideal sería poder reducir los problemas del mundo real a fórmulas matemáticas, como en alguno de los modelos de la Física que se han mencionado. Sin embargo las áreas de aplicación de la Informática no siempre lo permiten. Las aproximaciones serán notaciones variadas (gráficas, algebraicas, funcionales, textuales) que tratarán de representar el análisis del problema en forma unívoca.

La calidad de la especificación condiciona todo el proceso de desarrollo de soluciones por computador.

Normalmente, el mundo y el lenguaje del usuario, quien utilizará el programa desarrollado, son muy diferentes del mundo del especialista en informática que desarrolla el programa. El punto de contacto básico es el análisis del problema y su traducción desde el mundo real (usuario) al modelo abstracto que podrá resolverse sobre una computadora. La exactitud con la cual esta traducción aproxima el modelo a los objetivos del usuario condiciona fuertemente el resultado final del sistema.

### Expresión de soluciones ejecutables en un computador

Así como la especificación es fundamental para convertir el problema real en un modelo que permita la utilización de computadoras en su solución, la escritura de un programa que represente una solución ejecutable constituye la piedra basal de la Ciencia de la Computación.

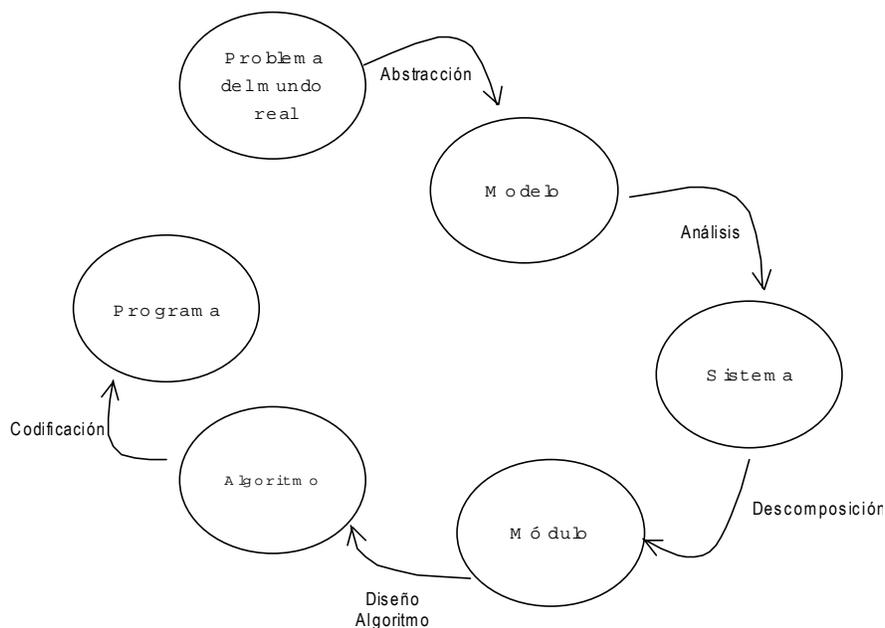
El curso de Introducción a la Programación, estará dedicado primordialmente a la expresión de soluciones como algoritmos computacionales y posteriormente como programas. Cada acción primitiva a ejecutar en un computador corresponde a una instrucción, y cada instrucción debe tener un modo único de escribirse (sintaxis) y un modo único de interpretarse y ejecutarse (semántica) para poder ser utilizadas por un computador.

El conjunto de instrucciones permitidas y definidas por sus reglas sintácticas y su valor semántico, para la expresión de soluciones a problemas, constituye un lenguaje de programación.

Los distintos lenguajes de programación tienen diferente sintaxis y semántica y las primitivas propias de cada uno generalmente se escriben de forma diferente, hay lenguajes de programación destinados a tareas específicas, por lo que sus primitivas estarán dirigidas básicamente a la realización de esas tareas específicas. La riqueza expresiva y la legibilidad de las primitivas de un lenguaje de programación son muy importantes para escribir programas que respondan a lo especificado.

### Del problema real a su solución por computador

Para expresar la solución a un problema en un computador, deben seguirse varios pasos, antes de tener un programa corriendo :



- Para llegar al modelo, a partir del problema se debe realizar un proceso de abstracción.
- A partir del modelo se debe elaborar el análisis de la solución como sistema. Esto significa una descomposición en módulos y un estudio de los datos del problema.
- Cada uno de los módulos debe tener un proceso de refinamiento para expresar su resolución en forma ordenada, lo que llevará a construir los algoritmos correspondientes. Es importante destacar que no siempre un problema puede tener una solución algorítmica.
- A partir de los algoritmos se pueden escribir y probar programas en un lenguaje determinado y con datos representativos del problema.
- Llegar a un modelo no significa que el problema sea computable, es decir, resoluble por computadora.
- Por ejemplo, se puede tener un algoritmo perfecto para calcular números primos. De hecho puede escribirse un programa para computar un número finito de números primos en un rango determinado. Sin embargo, si se quisieran calcular todos los números primos, no habría una solución computable en un tiempo finito.

### **Importancia del contexto**

El contexto del problema real, (por ejemplo, el modo de operación en la atención de clientes de un negocio comercial) es muy importante para analizar y diseñar la solución utilizando computadores.

Además, en muchas ocasiones el contexto impone restricciones que son importantes para la elaboración de la solución. Por ejemplo, si se está estudiando el software de un cajero automático, tener la respuesta a una consulta de saldo en menos de 1 minuto es un dato determinante del contexto, sobre el tipo de solución aceptable.

La caracterización del contexto es muy amplia. Por ejemplo, el número de empleados que pueden ser atendidos simultáneamente, los clientes con diferente prioridad, la posibilidad de pagos con tarjetas de crédito, el tipo y la cantidad de líneas telefónicas disponibles, etc., son datos que acompañan al modelo esencial del sistema y deben tenerse en cuenta antes de diseñar una solución.

### **Descomposición de problemas. Concepto de modularización**

Generalmente, del modelo general del problema se pasa a una descomposición en subproblemas menores (módulos), que en lo posible deben tener una función bien definida en el sistema.

La modularización o descomposición del modelo es muy importante y no solo se refiere a los procesos o funciones a cumplir, sino también a la distribución y utilización de los datos de entrada y salida, así como de los datos intermedios necesarios para alcanzar una solución.

Un módulo de programa encapsula una función y datos. Normalmente los datos que son propios del módulo serán llamados locales y aquellos datos que deba recibir o enviar a otros módulos serán datos de comunicación. Los datos que pueden ser conocidos y utilizados por todos los módulos de un sistema se llaman globales.

Se ha mencionado varias veces la palabra algoritmo, de la que seguramente, se tienen algunas ideas, aquí se aceptará la siguiente definición:

Algoritmo es la especificación rigurosa de la secuencia de pasos (instrucciones) a realizar para alcanzar un resultado deseado en un tiempo finito.

La expresión especificación rigurosa se refiere a que se debe expresar un algoritmo en forma unívoca, es decir, que la sintaxis y la semántica de cada instrucción deben estar perfectamente definidas.

Alcanzar el resultado en tiempo finito significa que se supone que un algoritmo comienza y termina. Una generalización de esta definición (por ejemplo, para sistemas de tiempo real que no terminan nunca, tales como un sistema de alarma) cambia la terminología "tiempo finito" por "número finito" de instrucciones.

### **Las partes de un programa**

Los componentes básicos de un programa son instrucciones y datos.

Las instrucciones (acciones) representan las operaciones que ejecutará el computador al interpretar el programa. Todos los lenguajes de programación tienen un conjunto mínimo de instrucciones que deben contener asignación, decisión e iteración.

Puede demostrarse que un lenguaje con solo tres instrucciones: asignación, decisión e iteración, permite escribir cualquier algoritmo.

Los datos son los valores de información de los que se necesita disponer, y en ocasiones transformar, para ejecutar la función del programa.

Conceptualmente, por una parte se tienen datos constantes que no cambian durante la ejecución del programa. Tal es el caso del valor de la fuerza de gravedad en los ejemplos de Física.

Por otra parte, se pueden tener datos variables, es decir, que durante la ejecución del programa pueden cambiar. Por ejemplo, la cantidad de dinero que solicita un cliente en un cajero automático, el número de materias inscritas por un estudiante en la universidad ó la velocidad de un móvil, en los ejemplos de Física.

Tanto los datos constantes como los variables deben guardarse en la memoria de datos de una computadora. En ambos casos estarán representados simbólicamente por un nombre que se asocia con una dirección única de memoria. Por esto, el contenido de la dirección de memoria correspondiente a un dato constante se asigna una sola vez en los programas. En cambio, el contenido (valor) de la dirección de memoria correspondiente a un dato variable puede asignarse y cambiar muchas veces durante la ejecución del programa.

Además de datos e instrucciones, al leer un programa encontraremos comentarios, es decir, texto aclaratorio para el programador o el usuario, que no es entendido ni ejecutado por la computadora. Este texto sirve para clarificar qué hace el programa, y será de gran importancia cuando se intente modificarlo o corregirlo.

Una visión interesante de un programa es que corresponde a una transformación de datos. A partir de un contexto determinado por las precondiciones, el programa transforma la información y debería llegar al resultado esperado produciendo un nuevo contexto, caracterizado por las poscondiciones.

### **Software**

La definición de Informática dada al comienzo, abarca aspectos relacionados tanto con la estructura propia de las computadoras como con su programación y control lógico.

Normalmente, los aspectos vinculados con la arquitectura física de las computadoras, las comunicaciones entre computadoras y los dispositivos periféricos que utilizan se denominan hardware.

A su vez, todo lo que se refiere a la programación básica y de aplicaciones, de modo de dar una utilidad definida al hardware, se denomina software.

Es interesante notar que a veces resulta muy difícil separar hardware y software. Por ejemplo, el controlador de una lavadora programable es un pequeño computador, que contiene hardware y software (los programas predefinidos que el usuario selecciona).

Esto explica la fuerte vinculación que hoy tienen la electrónica y la informática. Esencialmente la electrónica provee el cambio tecnológico que impulsa el desarrollo de la informática pero, a su vez, los cambios de

software (nuevos lenguajes, nuevos sistemas operativos, nuevos algoritmos) impactan en la concepción de circuitos y sistemas electrónicos.

A lo largo del curso de Introducción a la Programación, nos centraremos en el software y particularmente en el análisis y diseño de algoritmos dedicados a resolver problemas simples. Para ello, se estudiarán las clases de estructuras de datos adecuadas para resolver los problemas.

## **Etapas en la resolución de problemas con computadora**

### **Análisis del problema.**

En esta primera etapa, se analiza el problema en su contexto del mundo real. Deben obtenerse los requerimientos del usuario. El resultado de este análisis es un modelo preciso del ambiente del problema y del objetivo a resolver. Un componente importante de este modelo son los datos a utilizar y las transformaciones de los mismos que llevan al objetivo.

### **Diseño de una solución.**

Suponiendo que el problema es computable, a partir del modelo se debe definir una estructura de sistema de hardware y software que lo resuelva. El primer paso en el diseño de la solución es la modularización del problema, es decir, la descomposición del mismo en partes que tendrán una función bien definida y datos propios (locales). A su vez, debe establecerse la comunicación entre los módulos del sistema de software propuesto.

### **Especificación de algoritmos.**

Cada uno de los módulos del sistema de software tiene una función que se puede traducir en un algoritmo. La elección del algoritmo adecuado para la función del módulo es muy importante para la eficiencia posterior del sistema de software. Una misma función sobre los mismos datos puede resolverse con una utilización de recursos muy diferente (memoria, tiempo) según el algoritmo elegido.

### **Escritura de programas.**

Un algoritmo es una especificación simbólica que debe convertirse en un programa real sobre un lenguaje de programación concreto. Este proceso de programación tiende a automatizarse en la medida que los lenguajes algorítmicos se acercan a los lenguajes reales de programación. A su vez, el programa escrito en un lenguaje de programación determinado (por ejemplo, Pascal, Ada, C, Java) debe traducirse al lenguaje de máquina de la computadora que lo vaya a ejecutar. Esta traducción, denominada compilación, permite detectar y corregir los errores sintácticos que se cometan en la escritura del programa.

### **Verificación.**

Una vez que se tiene un programa escrito en un lenguaje real y depurado de errores sintácticos, se debe verificar que su ejecución conduzca al resultado deseado, con datos representativos del problema real. Sería deseable poder afirmar que el programa es correcto, más allá de los datos particulares de una ejecución. Sin embargo, en los casos reales es muy difícil realizar una verificación exhaustiva de todas las posibles condiciones de ejecución de un sistema de software. La facilidad de verificación y la depuración de errores de funcionamiento del programa conducen a una mejor calidad del sistema y este es el objetivo central de la Ingeniería de Software.

Un error de funcionamiento de alguno de los programas del sistema puede llevar a la necesidad de rehacer cualquiera de las etapas anteriores, incluso volver a discutir los requerimientos.