



Elementos de Programación

UNIDAD 1. INTRODUCCION A LA PROGRAMACION

INDICE

INTRODUCCIÓN	2
1. ESQUEMA BÁSICO DE UNA COMPUTADORA - MODELO DE VON NEUMANN	2
2. EVOLUCIÓN DE LAS COMPUTADORAS.....	4
3. CARACTERÍSTICAS GENERALES.....	4
4. ¿QUÉ ES UNA COMPUTADORA Y QUE PROBLEMAS PUEDE RESOLVER?.....	5
5. PROGRAMACIÓN.....	6
5.1 ¿QUÉ ES PROGRAMAR?	6
5.2 LENGUAJES DE PROGRAMACIÓN.....	6
5.3 COMPILADORES E INTÉRPRETES	7
6. ETAPAS PARA SOLUCIÓN DE PROBLEMAS CON LA COMPUTADORA	8
7. CONSTRUCCIÓN DE ALGORITMOS	8
8. PROGRAMACIÓN ESTRUCTURADA	9

UNIDAD 1- INTRODUCCION A LA PROGRAMACION

OBJETIVOS: *Informar al Alumno del mundo Informático, la computadora, sus características y evolución. Exponer el ámbito de la programación con distintos tipos de programas y lenguajes. Transmitir los pasos para encarar un problema y como diseñar una posible solución al mismo.*

Introducción

Antes de comenzar se introducirá de forma sencilla algunos conceptos:

- **Computadora:** es un equipo electrónico capaz de realizar muchas operaciones muy rápidamente, con la particularidad de que no está diseñada para hacer una sola tarea, sino que es capaz de realizar tareas diferentes dándole las instrucciones de cómo hacerlo mediante un programa.
- **Programar:** es dar instrucciones a una computadora, decirle que es lo que queremos que haga.
- **Informática:** Las computadoras manejan datos y esos datos al procesarlos se transforman en información. El manejo de dicha información mediante un programa de computadora da origen a la informática haciendo que los datos e información puedan ser procesados en forma automática.
- **Digitalizar:** Hacer que los datos sean adaptados para ser procesados por un sistema informático.
- **Hardware:** Componentes físicos de la computadora.
- **Software:** Parte no tangible que se ejecuta en los equipos informáticos. Son los programas y/o aplicaciones.

Usted se encuentra en la era de la información. Cada vez más y más procesos cotidianos se encuentran digitalizados y los datos son manipulados por sistemas computacionales. Por ejemplo, al ir a un supermercado los productos son leídos con un lector de códigos de barra y procesados mediante un sistema que recupera su precio y va calculando el gasto realizado. También al subir a un colectivo y pasar la tarjeta SUBE, hay detrás un proceso informático que descuenta de nuestro saldo el viaje realizado. Estos son solo dos pequeños ejemplos, pero si piensa en cuantas cosas de la vida cotidiana dependen de un sistema informático se va a dar cuenta de que cada vez son más y más.

Las computadoras están presentes en la vida cotidiana y se presentan de muchas formas y tamaños: computadoras de escritorio, notebooks, tablets, teléfonos celulares, entre otros. Es decir, está rodeado de máquinas capaces de ejecutar distintos programas que permiten realizar tareas de una forma fácil y rápida.

1. Esquema Básico de una Computadora - Modelo de Von Neumann

John Von Neumann fue un matemático húngaro nacido en 1903, que diseñó una arquitectura general que define como deber estar organizada una máquina de propósito general (computadora).

Esta arquitectura fue tomada en cuenta a la hora de diseñar las primeras computadoras y sigue siendo la base fundamental de las computadoras actuales.

El diseño realizado por Von Neumann plantea la arquitectura en tres bloques principales:

- **CPU:** sigla que viene del inglés, Central Processing Unit (Unidad Central de Procesamiento), es la encargada de interpretar y ejecutar las instrucciones de los programas que se quieren ejecutar.
- **Memoria Principal:** Es un espacio donde se almacenan tanto los programas que se quieren ejecutar como los datos que dichos programas manipulan.
- **Unidad de Entradas y Salidas:** Es el bloque encargado de la comunicación con el mundo exterior, y, por lo tanto, con los usuarios. Por ejemplo, para mostrar información por una pantalla o recibir los datos ingresados por un mouse o teclado.

La figura 1 muestra los bloques principales de la arquitectura de Von Neumann.

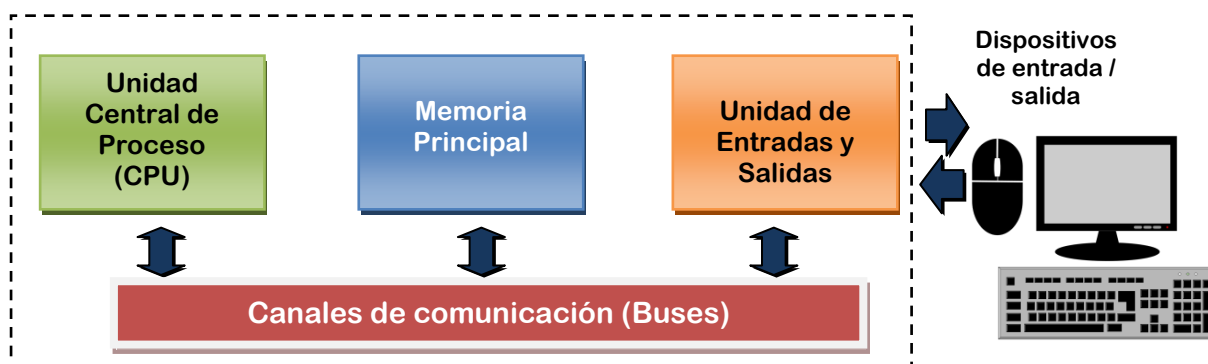


Figura 1: Modelo de Von Neumann

Este es un esquema general, ya que dentro de cada bloque hay varios componentes que trabajan en conjunto, para lograr que la computadora pueda ejecutar instrucciones y comunicarse con el usuario mediante dispositivos de entrada y salida como mouse, teclado, monitor, impresora, etc.

La encargada de ejecutar las instrucciones es la CPU, donde internamente incluye lo que se conoce como la unidad aritmética, que es capaz de realizar operaciones matemáticas y lógicas para poder procesar datos. Esas instrucciones se dan a la computadora mediante un programa que es almacenado en la memoria principal, y de allí se lleva a la CPU cada una de las instrucciones, una a una para ejecutarlas. En este esquema la memoria principal también guarda los datos con los que el programa trabaja y los resultados obtenidos. Esta memoria se divide en distintos "casilleros" donde cada uno puede guardar una cierta cantidad de información. Para acceder a cada casillero se utiliza una "dirección" diferente, por lo tanto, siempre que se recupera un dato o instrucción de la memoria se dice que accede a una dirección de memoria particular.

El bloque de entradas y salidas es un intermediario, que permite realizar la conexión con los dispositivos externos para permitir ingresar datos, o mostrar información. Este bloque también adecúa velocidades ya que los dispositivos externos son mucho más lentos que los componentes internos de la computadora.

Los bloques se comunican entre sí mediante distintos canales de comunicación llamados buses, donde por ellos pueden viajar direcciones de memoria, datos y señales de control.

2. Evolución de las computadoras

Las computadoras de hoy en día se presentan en una amplia variedad de formas, tamaños y precios. Las grandes computadoras de propósito general se utilizan en muchos negocios, universidades, hospitales y agencias gubernamentales para desarrollar sofisticados cálculos científicos y financieros.

Se conoce a estas grandes computadoras o grandes sistemas como **mainframes**, que necesitan de equipamiento y mobiliario especial, en cuanto a espacio, electricidad, humedad, etc. También existen otras computadoras más chicas en tamaño, como las computadoras personales que con la incorporación de las redes, permiten lograr recursos importantes y a un menor costo.

A **comienzo de los cincuenta** se disponía de grandes sistemas, aunque **muy poca gente sabía** cómo utilizarlos, lo hacían generalmente científicos, ingenieros y analistas financieros.

Durante **los sesenta**, fue cada vez **más frecuente que aprendieran a programar grandes sistemas los estudiantes en las universidades**, esto produjo la aparición de jóvenes profesionales provocando la desaparición del temor y recelo existentes hasta el momento, con la utilización de los grandes equipos.

A finales de los sesenta y comienzos de **los setenta**, se desarrollaron **minicomputadoras**, más pequeñas y menos caras, lo que permitió el **acceso** a las mismas de **comerciantes más pequeños** y en los **hogares**.

Hasta llegar a **nuestra época**, donde podríamos decir que es una **herramienta imprescindible** para cualquier profesional y comerciante para que pueda realizar sus actividades con eficiencia.

La **evolución de las computadoras es el resultado de la innovación de los componentes electrónicos**, donde con cada avance tecnológico nació una nueva generación de computadoras que mejoraban tanto en tamaño (haciéndose más pequeñas) como en velocidad (haciéndose más rápidas). Según la tecnología **se pueden identificar 4 generaciones** de computadoras:

- **1ra generación válvula de vacío**
- **2da generación Transistor**
- **3ra generación Circuito Integrado**
- **4ta generación Microprocesador**

Hoy en día si bien no hay un cambio radical en la tecnología, se avanza en la incorporación de varios núcleos de procesamiento en un mismo microprocesador, lo que permite ejecutar distintas instrucciones al mismo tiempo.

3. Características generales

Todas las **computadoras digitales**, independientemente de su tamaño, son básicamente **dispositivos electrónicos que pueden transmitir, almacenar y manipular información** (datos). Una computadora puede procesar distintos tipos de datos. Esto incluye datos numéricos, alfanuméricos (nombres, direcciones, etc.), datos gráficos (mapas, dibujos, fotografías, etc.), y sonido (música, lectura de textos, etc.). Desde el punto de vista del programador recién iniciado, los dos tipos de datos más familiares son los números y los caracteres, involucrando frecuentemente las aplicaciones de negocios el tratamiento de estos tipos de datos.

Para que la computadora procese un lote particular de datos es necesario darle un **conjunto apropiado de instrucciones** llamado **programa**. Estas instrucciones se introducen en la computadora y se almacenan en una parte de la memoria de la máquina.

Un programa almacenado se puede ejecutar en cualquier momento. La ejecución de un programa supone lo siguiente:

1. Un conjunto de datos, los datos de entrada, se introduce en la computadora (desde un teclado o un archivo), y se almacena en una porción de memoria de ésta.
2. Los datos de entrada se “procesarán” para producir ciertos resultados deseados, que son los datos de salida.
3. Los datos de salida, y probablemente algunos de los datos de entrada, se imprimirán en papel o se presentarán en un monitor.

Este procedimiento de tres pasos se puede repetir tantas veces como se desee. En cualquier caso, se debe tener presente que estos pasos, especialmente el 2 y el 3, pueden ser largos y complicados.

Ejemplo. Una computadora ha sido programada para calcular el área de un círculo utilizando la fórmula:

Área = $\pi \times r^2$, dando el valor numérico del radio como dato de entrada, es necesario dar los siguientes pasos para efectuar el cálculo:

1. Leer el valor numérico del radio del círculo.
2. Calcular el valor del área utilizando la fórmula anterior. Este valor se almacenará entre los datos de entrada en la memoria de la computadora.
3. Imprimir (mostrar en pantalla) los valores del radio y el área correspondiente.
4. Parar.

Cada uno de estos pasos requiere de una instrucción o más en un programa.

Lo anteriormente dicho ilustra dos características importantes de una computadora digital: memoria y capacidad de ser programada. Otras características importantes son su velocidad y fiabilidad.

4. ¿Qué es una computadora y que problemas puede resolver?

Para decidir cuáles y cómo se pueden resolver los problemas recordemos cómo funciona una computadora digital, comencemos por dar una definición:

“Una computadora es un sistema electrónico dedicado al procesamiento de datos (información), formado por varias unidades, cuyo funcionamiento viene dictado por el programa que se halla almacenado en su memoria principal”

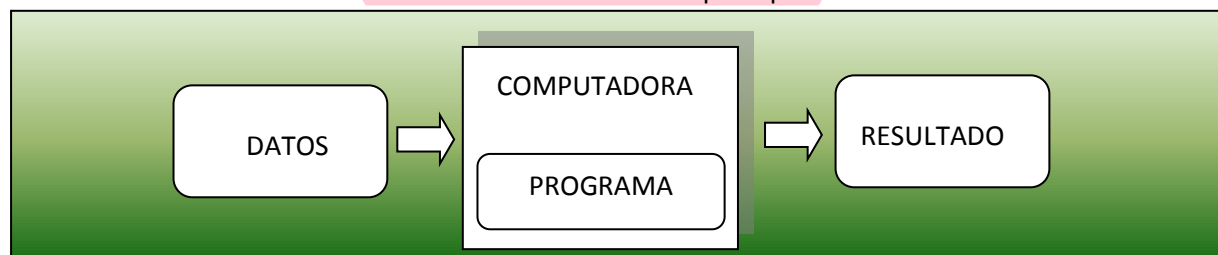


Figura 2: Resolución de problemas mediante una computadora

La computadora es un mecanismo complejo que solo opera sobre datos, resolviendo operaciones muy sencillas (sumas y comparaciones) a una elevadísima velocidad y con gran precisión.

Como la tarea es la resolución de problemas utilizando una computadora, se debe expresar una secuencia de operaciones lógicas, en un lenguaje suficientemente básico y preciso, en

instrucciones que la computadora sepa resolver, éstas serán las instrucciones que componen a cada lenguaje de programación.

Básicamente, la computadora puede realizar los siguientes tipos de operaciones:

- **Ingresar datos.** Pueden ser varios y de diferente tipo – numéricos y no numéricos.
- **Realizar cualquier tipo de operación algebraica y/o lógica,** efectuando comparaciones y generando ciclos definidos o condicionados.
- **Almacenar información** en forma transitoria (memoria principal), o permanente (memorias auxiliares)
- **Exhibir datos y/o resultados** en diferentes dispositivos como ser en una pantalla, una impresora, etc.

5. Programación

5.1 ¿Qué es programar?

Programar es la actividad de comunicar a la computadora lo que se tiene que hacer. Cualquiera que desee que la computadora haga algunas cosas, deberá acoplarse a este tipo de comunicación persona-computadora, utilizando un lenguaje concreto. Una manera de apreciar lo que es un programa, es considerarlo como si fuera un plan detallado para transformar información, aceptando datos y produciendo una nueva información. A las reglas de transformación se las denomina **algoritmos**.

El problema que se establece al escribir un programa es básicamente el de la comunicación entre un ser humano y una computadora. Probablemente el ser humano sabe cómo encarar el problema, o al menos sabe cómo resolver algunos problemas afines, pero si el problema se va a resolver con la ayuda de una computadora, los humanos deben transmitirle su propósito.

Los lenguajes humanos no son apropiados para la comunicación con las computadoras, principalmente porque acarrearán una notable serie de ambigüedades. De persona a persona la comunicación puede tolerar esta ambigüedad debido a acuerdos tácitos, y a que la comunicación no verbal está también presente. Pero en la comunicación con una máquina todo se lleva a cabo de forma literal, de modo que se necesita ser muy preciso y específico en describir la labor que se tenga que realizar. De ahí que la gente haya elaborado **lenguajes especializados**, denominados **lenguajes de programación**, para la comunicación entre la persona y la máquina.

5.2 Lenguajes de programación

Se necesita un sistema para decirle a la computadora lo que se quiere hacer. Los lenguajes de programación permiten realizar dicha tarea. Se trata de un lenguaje especial que no incluye ambigüedades. Estos lenguajes deben tener descripciones precisas de las tareas que se deben realizar.

Las computadoras actuales trabajan en sistema binario, es decir, con dos niveles de corriente que se representa con 0 y 1. Para un programador sería muy difícil escribir instrucciones de un programa como una secuencia de ceros y unos, por lo tanto, surgen los lenguajes de programación.

Cada CPU, cuyo componente físico en una computadora personal es el microprocesador, puede interpretar y ejecutar distintas instrucciones. Hay conjuntos de instrucciones estandarizadas que fueron evolucionando a lo largo del tiempo, por lo tanto, los microprocesadores en general soportan la ejecución de varios sets de instrucciones diferentes según su arquitectura.

El lenguaje de programación más cercano a la computadora es el Assembler, donde cada instrucción del lenguaje de programación se corresponde directamente con una instrucción que el microprocesador es capaz de ejecutar. Este lenguaje está muy ligado al hardware ya que solo funcionará con procesadores que entiendan dichas instrucciones. A este tipo de lenguajes se los conoce como lenguajes de bajo nivel por su cercanía al hardware.

Programar en Assembler no es una tarea fácil, ya que tareas sencillas muchas veces requieren de varias instrucciones para lograr realizarlas, como, por ejemplo, una multiplicación se hará sumando sucesivamente un número, y, además, el programa realizado va a funcionar en un hardware particular. Por este motivo, surgen lenguajes de programación más avanzados, donde una sola instrucción del lenguaje se traduce luego en varias instrucciones interpretables por el procesador haciendo la tarea del programador más sencilla. Además, estos lenguajes avanzados se independizan del hardware ya que el mismo programa puede ser traducido para ser ejecutado en distintos procesadores con distintos sets de instrucciones. Por eso se los llama lenguajes de Alto Nivel.

Todo lenguaje de programación está compuesto por:

- **Léxico:** que símbolos del lenguaje puedo utilizar para escribir mi programa. Letras, números y algunos símbolos conforman el léxico de la mayoría de los lenguajes.
- **Sintaxis:** son las reglas que indican de qué forma se combinan los símbolos para lograr escribir instrucciones válidas. La sintaxis también define palabras que están reservadas por el lenguaje y son necesarias para la escritura de las instrucciones. Por ejemplo, la palabra if, es reconocida en muchos lenguajes como el comienzo de una condición lógica, por lo tanto, esa palabra no puede utilizarse para otro fin.
- **Semántica:** cada instrucción válida del lenguaje de programación tiene un propósito particular. Indica algo que se quiere que la computadora haga. Ese sentido de cada instrucción es la semántica. La semántica, por lo tanto, es el significado de las instrucciones de un programa.

Un programa de computadora es básicamente una colección secuencial de “sentencias” siguiendo cada una de ellas las reglas sintácticas del lenguaje. Cada sentencia contiene información para la computadora. Usted puede considerar estas sentencias declarativas como si fueran absorbidas por la computadora una a una en un orden secuencial; entonces la computadora “ejecuta” cada sentencia, hace lo que la sentencia le pide que haga.

Sentencia 1	(primera en ejecutarse)
Sentencia 2	(segunda en ejecutarse)
---	-----
Sentencia n	(enésima en ejecutarse)

5.3 Compiladores e intérpretes

El programa en el lenguaje en que se escribe las instrucciones se llama programa fuente. Este programa fuente es un archivo de texto que la computadora no puede comprender directamente, y, por lo tanto, necesita que sea traducido al lenguaje máquina.

Hay dos mecanismos diferentes para realizar esa traducción, mediante un compilador o mediante un intérprete. El compilador toma el programa fuente, analiza si está bien escrito según la sintaxis del lenguaje de programación elegido y genera un programa en un lenguaje que la computadora puede interpretar llamado programa objeto. Ejemplos de lenguajes compilados son C, C++, Java, Fortran. En cambio, un intérprete va tomando una a una las instrucciones del programa fuente, las traduce y las ejecuta, es decir, no analiza todo el programa, sino que va traduciendo y ejecutando las instrucciones a medida que lee el programa fuente. El problema de los lenguajes interpretados es que, al no analizar el programa completo, puede ser que se comience a ejecutar un programa

hasta que se llegue a una instrucción que no pueda interpretarse por lo que se terminaría abruptamente la ejecución del programa. Un ejemplo de un lenguaje interpretado es `javascript`, utilizado para ejecutar instrucciones en una página web y agregarle funcionalidad.

6. Etapas para solución de problemas con la computadora

La solución de problemas mediante la computadora consta de ocho etapas, articuladas de tal forma que cada una depende de las anteriores, lo cual indica que se trata de un proceso complementario y, por lo tanto, cada paso exige el mismo cuidado en su elaboración. Las ocho etapas son:

1. Comprensión, definición y delimitación del problema a solucionar.

Para resolver el problema es fundamental **conocerlo y delimitarlo** por completo, determinar con qué datos se cuenta, la información a obtener. Es **importante tener claridad sobre el problema** a resolver para no dar soluciones incorrectas.

2. Estrategia - definir los algoritmos.

Disponer de los algoritmos necesarios para resolver los problemas. **Identificar en forma manual como resolvería sin la computadora el problema propuesto**, realizando todos los pasos necesarios en una hoja entendiendo bien el proceso.

3. Diagrama de flujo.

Es la primera etapa para realizar luego de tener en claro el problema y su manera de resolver. En esta etapa, es donde **se determinan los pasos e instrucciones que deben llevarse a cabo y el orden lógico de su ejecución**, para una eficiente solución al problema representado en forma gráfica con los símbolos de diagramas propuestos. Se puede decir que es aquí donde radica toda la dificultad para solucionar un problema a realizar.

4. Prueba de escritorio.

Consiste en hacer un seguimiento manual, a través, **de un lote de prueba que permita verificar los pasos que realiza el diagrama desarrollado**, esto permitirá reconocer a través del resultado si funciona correctamente o no.

5. Codificación.

Es la escritura de las instrucciones, determinadas en la etapa de la diagramación, **en un lenguaje de programación determinado**.

6. Compilación o interpretación del programa.

En esta etapa, la computadora chequea si todas las instrucciones están escritas **correctamente** en el código utilizado, respetando, simbología y sintaxis.

7. Ejecución del Programa.

Luego de la etapa anterior, si el compilador no dio errores, **el programa objeto es ejecutado**, para llegar a los resultados esperados.

8. Evaluación de los resultados.

Obtenidos los resultados, se **los evalúa para verificar si son correctos**. En caso contrario, se revisa las etapas anteriores para detectar la falla o error.

7. Construcción de Algoritmos

De las etapas mencionadas, ésta es la que traerá las mayores dificultades por lo cual será la principal del trabajo a realizar. Se debe destacar que el concepto de **"algoritmo"** es similar al de **"estrategia"**, pero con la diferencia que en lugar de hablar de "procesos o tareas", se hablará de **"instrucciones"**, lo cual **involucra un mayor grado de detalle**, donde estarán presentes **operaciones minuciosas y precisas**.

El algoritmo se lo puede describir en forma gráfica mediante un diagrama, y luego, codificarlo en un lenguaje de programación particular generando un programa.

La utilización de diagramas para pensar y diseñar los algoritmos antes de codificarlos tiene las siguientes ventajas:

1. permite una sencilla y rápida visualización total del problema y sus distintas alternativas.
2. es un verdadero medio de comunicación entre quien explica y quien aprende.
3. es un medio claro y conciso de documentación de los problemas.
4. facilita los posibles cambios y visualiza distintas alternativas posibles.

Existen varios métodos para encarar la construcción del algoritmo. A estas diferentes formas de encarar la construcción de una solución a un problema mediante la computadora, se las conoce como paradigmas de programación. Entre los principales paradigmas se puede mencionar: programación estructurada, orientada a objetos, funcional, lógica, etc. Un buen punto de partida para comenzar a comprender y dominar el diseño de algoritmos es utilizar el enfoque de la programación estructurada, ya que nos dará la base necesaria para poder luego utilizar otros paradigmas diferentes.

8. Programación estructurada

Este método tiene como soporte teórico al teorema fundamental de la programación estructurada, (Jacerina y Bohn) que afirma que todo algoritmo puede ser construido en forma correcta utilizando "únicamente" tres estructuras básicas:

- **SECUENCIAL:** las instrucciones del programa se ejecutan una a continuación de la otra, en formar ordenada, y siguiendo la secuencia que el programa indica.
- **SELECCIÓN:** permite tomar decisiones en el programa, abriendo distintos caminos haciendo que se ejecuten determinadas instrucciones solo si se cumple alguna condición particular. De esta forma el programa puede tener distintas variantes de ejecución y adaptarse a los datos que se están procesando.
- **ITERACION:** esta estructura permite repetir un conjunto de instrucciones, ya sea una cantidad de veces fija o dependiendo de alguna condición. Al utilizar instrucciones de iteración los programas quedan más cortos, y se pueden modificar más fácilmente que si se escribieran varias veces las mismas instrucciones.

También la programación estructurada indica que todo programa o módulo de un programa debe tener un comienzo y un único punto de finalización. Esto ayuda a la comprensión y diseño de los algoritmos haciendo que el proceso se ejecute en forma secuencial hasta llegar al final de mismo, existiendo un único punto donde el programa finaliza. Más adelante se verá que hay lenguajes que soportan que existan múltiples puntos de finalización de un programa. En este tipo de lenguajes para seguir la programación estructurada se recomendará unificar las salidas a un único punto para simplificar la lógica.

Una vez comprendidas las estructuras básicas de la programación estructuradas se verán distintas herramientas y estructuras de datos que permiten realizar programas mas complejos de forma más reducida y óptima.