#1 En complementos de aprendizaje:
Presentaciones - Videos - Ejercicios

Aprenda Practicando

2ª Edición

Introducción a la Programación

Algoritmos y su implementación en

VB.NET, C#, Java y C++

Felipe Ramírez

- Aprenda los elementos y técnicas de la lógica de programación
- Aprenda el proceso de análisis, abstracción y documentación de casos reales de negocios
- Desarrolle algoritmos, diagramas de flujo, miniespecificaciones y pruebas de escritorio
- Implemente algoritmos en Visual Basic.NET, C#, C++ Java y Raptor
- Aprenda los elementos de la programación orientada a objetos, usando Visual Basic

Alfaomega

En mi casa me espera
una rosa y un rosal.

Índice de contenido

Capítulo 4: La naturaleza de los datos

Capítulo 5: Operadores y reglas de precedencia

Capítulo 6: Algoritmos para el análisis de casos reales

Capítulo 7: Algoritmos orientados a datos y miniespecificaciones

Capítulo 10: Implementación de algoritmos en Visual Basic.NET

Capítulo 11: Implementación de algoritmos en C#

Capítulo 14: Fundamentos de la programación orientada a objetos

Índice de prácticas

Total de ejercicios: **62**

Prólogo

Iluso es pensar obtener resultados distintos,
haciendo las mismas cosas
Albert Einstein

Lo importante en un lenguaje es lo que se tiene que decir con él

En términos computacionales, supongo que pertenezco a lo que sería la tierra media. Una época en la cual si querías ver un cuadro dibujado en la pantalla, había que programarlo; una buena máquina tenía 20 MB en disco duro y un monitor que soportara 16 colores era un lujo que muy pocos privilegiados podían tener.

En esos días, como programador debías tener lógica de programación o te morías de hambre, dado que sin lógica era imposible desarrollar nada. Ahora las cosas han cambiado. Pasó el tiempo y las máquinas mejoraron, mejoraron las herramientas y los lenguajes también; ahora colocar un cuadro es tan fácil como arrastrar y colocar una figura en la pantalla, y el desarrollador no sabe qué sucede atrás, tan bambalinas. ¿Tengo algo en contra de la mejora continua de equipos, lenguajes y herramientas? Para nada. Vivo de ello.

Mi preocupación va en otro sentido. Pasan los años y como consultor ya me sé a la perfección los síntomas de una empresa con problemas de desarrollo de aplicaciones. Año con año sigo escuchando las mismas quejas: desfases en tiempo, desfases en presupuesto, y requerimientos mal entendidos pero bien desarrollados.

El común denominador es siempre un cliente descontento, antes como ahora.

Si las máquinas, los lenguajes y las herramientas de desarrollo han evolucionado tan radicalmente, y sin embargo seguimos teniendo los mismos problemas, entonces hemos estado buscando la solución a los problemas de desarrollo en el lugar equivocado.

Para identificar la solución a un problema basta con encontrar una situación libre de problemas, digámoslo así, un caso de éxito. Después analizamos nuestro caso de fracaso, y fácilmente detectamos las diferencias con el caso exitoso; corregimos el rumbo en nuestros casos de fracaso y corregimos la situación. Es una simple aplicación de mejores prácticas.

Hemos identificado que los desarrollos condenados al fracaso tienen el común denominador de ser desarrollos poco estructurados y estandarizados, caóticos, en donde el rol principal lo juegan las máquinas y herramientas, y no la mente humana. El problema es que los requerimientos no los hace una computadora, sino una persona a la que llamamos cliente.

No conozco un cliente que se queje por haber obtenido exactamente lo que pidió al precio justo. Para que ello suceda hay que analizar sistemáticamente lo que quiere, definir sus requerimientos y validarlos con él antes que nada; cuando ya ha aprobado los requerimientos, es necesario reducir el caso real a pasos que se puedan automatizar, construir un modelo abstracto, identificar los datos del modelo abstracto, diagramar el modelo y darle coherencia y secuencia, miniespecificar lo que han de ser las actividades que el programa debe desarrollar, codificar el modelo usando un lenguaje de programación, compilar los programas y depurarlos, probar que la aplicación hace lo que debe, y finalmente presentarlo al cliente para su aceptación. Esto funciona, pero no es fácil de hacer por personas que no han tenido práctica en ello.

Este libro enseña al lector a realizar todas las actividades anteriores, aumentando la capacidad de éxito del desarrollador. Sólo será necesario aplicar muchas habilidades mentales que nada tienen que ver con la computación, como lo son la capacidad de análisis, de abstracción y de conclusión.

No será una tarea fácil para algunos. En estos tiempos las nuevas generaciones ya han sido evangelizadas por la televisión, en donde todo es inmediato. Cualquier problema que toma más de cinco minutos de análisis mental ya es un fastidio, y esa circunstancia ciertamente no ayuda.

Las herramientas de computación cada día simplifican los procesos de producción de programas, pero no nos engañemos: la solución de los problemas está en la forma en que éstos se plantean, y es en la fase de análisis en donde se realiza dicha actividad. Finalmente la solución óptima de los programas está en nuestro cerebro y en su asombrosa capacidad de analizar, abstraer y concluir de forma sistemática. Las herramientas de desarrollo sólo permiten implementar más rápido lo que ya habita en nuestra imaginación.

En el 2003 escribí un artículo por invitación del Colegio Interamericano, en Guatemala. Decidí llamar al artículo “Notas particulares para el programador frustrado”, dado que trataba de razonar las causas de la falta de efectividad en el aprendizaje de la programación. En ese artículo afirmaba que la lógica de programación no podía enseñarse, dado que existen muchas formas de razonar. Sigo en lo mismo: la lógica de programación no puede enseñarse, sólo puede aprenderse. Sólo es cuestión de pensar lógicamente en términos generales, y será cuestión de tiempo en que nuestra forma de razonar toque todos los aspectos de nuestra vida, incluso la programación.

Espero que este libro sea una fiesta del conocimiento, y no un funeral de vocaciones. La programación no es un arte, es una técnica. Cualquiera que pueda pensar podrá analizar, cualquiera que pueda analizar podrá generar algoritmos, quien pueda generar algoritmos podrá generar modelos de datos, y quien pueda generar modelos de datos podrá miniespecificar, codificar y probar programas. ¿Usted puede

pensar? Si la respuesta es que sí, entonces no hay excusa. Sólo es necesario ponerse a trabajar.

Cuando Dios nos manda inspiración
es mejor que nos encuentre trabajando.
Pablo Picasso

Agradecimientos

Este libro es entrañable para mí, por muchas razones. Le debo tanto a la programación que 400 páginas son un homenaje injusto.

Con quien no debo ser injusto es con todos aquellos que forman un entorno en donde la escritura es posible.

A mi esposa Adriana, que me conoce a fondo y sabe sacar de mí la mejor persona que puedo ser, en todos los sentidos; agradezco además su ingenio para mantener a sana distancia a la pequeña Sara mientras escribo, lo cual es casi un deporte extremo.

A Isela Martínez, cuya ayuda incansable en la revisión editorial permite que escritos relativamente decentes puedan aspirar por un instante a ser escritos maravillosos y geniales.

A mis maestros, programadores de la vieja guardia: José Luis Ramos, Hugo Segundo y José Méndez; a mis amigos en la Universidad, Paco Salazar, Mary Araiza, Magda Madrigal, Eliud Palacios y Juan Enrique Saldaña; a Jorge Hernández y Elsa de la Fuente, del CSI en la Facultad de Ciencias Físico Matemáticas. A mis compañeros en el Gobierno Federal: Verónica López, Rey Manuel Martínez, y Aldo Avilés; a mis pupilos en la programación: Paco Guevara, Heriberto González, Maggy Córdoba, Antonio Ramírez, Carlos Eduardo Treviño, y muy especialmente a Víctor Manuel Ortega, que dio una revisada con ojos de alumno universitario al texto, evitando que el contenido tomara tintes filosóficos crípticos y poco entendibles (que vaya sí se me dan).

A los proveedores de música para programar: Diana Krall, The Gathering, Slayer, Iron Maiden, Tool, Mastodon. Qué tedioso programar sin música.

Gracias a Dios sobre todo, por dejarme ser feliz y no morir en el intento.

Monterrey, Nuevo León, Junio 2006.

¿Cómo utilizar este libro?

Este libro es una herramienta para la formación del pensamiento lógico y el razonamiento, aplicable a la función de programación de computadoras.

El libro está compuesto por material contenido en el curso de certificación 1008 de Aprenda Practicando: Introducción a la programación y su implementación en Visual Basic.NET, C#, Java y C++.

El contenido está pensado para que el lector, en un período de 4 a 6 meses, aprenda las técnicas de análisis y abstracción de casos reales, su representación de forma técnica y su codificación.

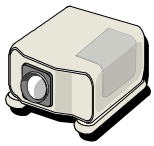
Si se es autodidacta y se dedica tiempo completo a la realización de los ejercicios del libro, se puede aprender el contenido del libro en 2 meses, con el apoyo de las herramientas de Aprenda Practicando en Línea.

El software que se requiere está disponible en línea, en las siguientes rutas:

.NET Framework Redistributable Package 2.0 .NET Framework SDK 2.0	msdn.microsoft.com/
Borland C++ Compiler 5.5	www.borland.com/downloads/index.html
J2SE 5.0 JDK	www.sun.com/java/

¿Cómo se componen los capítulos?

Los capítulos del presente libro tienen los siguientes elementos que fomentan el aprendizaje:



lp*xx*.ppt

- **Presentación profesional en Power Point.** Si se utiliza el presente libro para la exposición de cátedra o de cursos, la buena noticia es que no tendrá que desarrollar material de apoyo didáctico. En lugar de eso, podrá descargar de manera gratuita desde el sitio Web de Aprenda Practicando las presentaciones profesionales en Power Point que hemos desarrollado. Si las imprime como página de notas, adicionalmente obtendrá importante información para que la transmisión del conocimiento se lleve a cabo de manera efectiva. Cada capítulo señala, bajo el número de capítulo, el archivo que se deberá descargar para apoyar la exposición. Para descargar las presentaciones sólo es necesario ir al sitio **http://www.aprendapracticando.com**

- **Objetivo general y objetivos particulares.** Cada capítulo tendrá en su primera página un objetivo general, que representa la nueva habilidad que habremos adquirido al concluir el estudio del capítulo; además señalará los objetivos particulares, que representan en secuencia lógica los focos de atención para alcanzar el objetivo general.

- **Índice por capítulo.** Cada capítulo contiene su propio índice. Ubicar la información dentro de un capítulo resulta más fácil; además, el índice diferencia visualmente los temas de los ejercicios, a fin de que se pueda localizar a estos últimos de manera más fácil.

- **Notas informativas.** Cada capítulo contendrá notas adicionales, que profundizan en conocimiento general que no está disponible generalmente en la documentación del producto; además, se anotan aquí referencias históricas de los productos, así como consejos y advertencia que son para tomarse muy en cuenta, reduciéndose la curva de aprendizaje.

- **Videos explicativos.** Cada capítulo puede contener videos explicativos que abordan temas que es más fácil tratar de manera visual y dinámica, que textual y estática. Los videos están disponibles si se cursa el material en el entorno virtual de aprendizaje de Aprenda Practicando.

- **Gráficos ilustrativos.** Cada capítulo contendrá gráficos que permitirán ilustrar los temas tratados; a veces una imagen dice más que mil palabras.

- **Explicaciones textuales.** Cada capítulo contiene explicaciones profundas, que tratan el por qué de las cosas; esta es la principal diferencia con respecto a los manuales técnicos, que sólo explican cómo hacer las cosas, aunque el lector no entienda lo que está haciendo.

- **Ejercicios detallados, con explicaciones línea por línea y ejecución guiada.** La mayoría de los capítulos poseen ejercicios que permiten comprobar la teoría estudiada. Los ejercicios son desarrollados paso a paso; los programas pueden ser escritos aún sin disponer de herramientas visuales de desarrollo, y son ilustrados con líneas de código.

- **Terminología.** El texto introduce muchos conceptos nuevos que es necesario tener en mente. Cada capítulo tiene referido su propio índice de conceptos, por lo cual será muy fácil consultar el significado de los conceptos relevantes que es necesario saber para entender los temas.

- **Preguntas.** Cada capítulo incluye una serie de preguntas que tienen por objeto que el lector analice su propia opinión y experiencia respecto a lo que está aprendiendo. Las preguntas hacen referencias a los temas explicados en el capítulo, pero no cuestionan definiciones ni tópicos cuya respuesta está en el texto. Con las preguntas la idea es que piense respecto a lo que estudia.

- **Mapas mentales.** Los mapas mentales muestran en una sola imagen todos los temas ancla que se han cubierto en el capítulo. Actúan como resumen del capítulo, mismo que puede ser explorado de un vistazo.

- **Examen rápido.** Es la evaluación rápida que se debe resolver después de haber estudiado los temas, y de haber resuelto los ejercicios. Si se entendió cuáles eran los objetivos de estudio, y se contestaron las preguntas, se entendió la terminología, se realizaron los ejercicios y las prácticas individuales, es seguro que la evaluación siempre será contestada de una forma perfecta, con conocimiento de causa respecto a todo.

Sabemos de la enorme diversidad de personas. Cada persona tendrá su forma particular de aprender; algunos aprenden leyendo un resumen, otros viendo un video, otros leyendo, otros contestando preguntas, otros practicando, otros siendo cuestionados. Tratamos de brindar la mayor gama de herramientas, a fin de que cada quien aprenda como quiera, siempre y cuando aprenda.

En nuestro sitio Web encontrará además toda una plataforma de aprendizaje, denominado Aprenda Practicando en Línea que permitirá organizar el estudio y llevar un seguimiento del avance en el aprendizaje.

Estrategias de uso

Para el autodidacta...

La persona autodidacta es la que toma el aprendizaje en sus manos; no depende de las explicaciones de nadie, y por lo tanto tiene control total de su propio proceso de aprendizaje. A ese tipo de personas le sugerimos lo siguiente:

1. Todos los capítulos del 1 al 8 tienen secuencia, por lo que recomendamos su lectura uno tras otro.
2. Atienda de manera secuencial cada capítulo del tema.
    a. Lea con detenimiento los objetivos del capítulo, para que tenga presente que eso es lo que hay que saber al concluir el capítulo.
    b. Si interrumpe la lectura del capítulo, cada vez que reinicie la lectura, lea de nuevo los objetivos. Es muy importante que los tenga presentes.
    c. Realice los ejercicios en el momento en que los encuentra en el texto.

d. Responda el examen rápido; si no contesta correctamente la mayoría de las preguntas, vuelva a revisar el capítulo, hasta que quede satisfecho.

3. Recomendamos, si así lo desea, que pruebe los cuestionarios y juegos que en línea se encuentran en **www.aprendapracticando.com**, para comprobar el grado de experiencia que ha adquirido.

Para el uso con grupos...

Si el libro se utiliza para la capacitación de grupos, sugerimos que se tomen en cuenta las siguientes consideraciones para alcanzar los objetivos (suponemos que usted es el instructor / facilitador):

Al iniciar un tema...

1. El instructor / facilitador deberá dominar el tema a instruir; previo a la instrucción, deberá responder a las preguntas, realizar los ejercicios, realizar las prácticas individuales, y responder a la perfección el examen rápido.
2. Los equipos en los que se pretendan demostrar los ejercicios deberán estar preparados para el desarrollo las tecnologías que cubre el libro.
3. Las presentaciones profesionales en Power Point están disponibles en el sitio Web de Aprenda Practicando (**http://www.aprendapracticando.com**); descargue las versiones más actualizadas, antes de cada curso.
4. Antes de revisar una parte del libro, explique la importancia del aprendizaje del tema. Genere el interés por el tema; sin interés no hay aprendizaje.
5. Si puede citar su experiencia personal en el uso del conocimiento del tema, es lo ideal para demostrar que la tecnología estudiada es útil. En la medida en que la audiencia piensa que el tema es importante, dedicará una mayor atención.

Para revisar cada capítulo...

6. Solicite, antes de la sesión de instrucción, que los miembros del grupo lean el capítulo y resuelvan las preguntas al final del capítulo. Puede pedirlo como tarea escrita (de preferencia a mano, para contrarrestar el Copy - Paste), para garantizar que los participantes leyeron con anterioridad.
7. Al iniciar cada sesión deberá recordar los objetivos del capítulo que se estudiará.
8. En caso de que se inicie capítulo, después de ver los objetivos, solicite a los miembros del grupo que lean las respuestas a las preguntas (punto 6); de preferencia ceda la palabra de tal forma que todas las personas participen.
9. Proyecte la presentación profesional que acompaña a cada capítulo; para un mejor aprovechamiento, recomendamos que para su uso personal imprima la

presentación como página de notas, y lea las notas de cada una de las filminas, que le apoyarán con consejos y anotaciones relevantes.

10. Solicite a los alumnos que vean los videos que están disponibles en línea, y discuta su contenido.

11. En forma de exposición, solicite a los miembros del grupo que realicen los ejercicios hasta el final, cuando ya se haya expuesto toda la presentación.

12. Los miembros del grupo deberán realizar los ejercicios que el instructor / facilitador les pida. Se deberá especificar con claridad cuál es el tiempo máximo de realización; es importante que los tiempos marcados se respeten.

13. El instructor / facilitador, en caso de detectar dificultades en la realización de los ejercicios, deberá apoyar a los miembros del grupo, pero sin hacer el trabajo por ellos.

14. En caso de detectar un problema que sucede comúnmente en alguna práctica, es importante comentar con el grupo el problema y la solución.

15. Al concluir la realización de los ejercicios, el instructor / facilitador comentará las experiencias más relevantes que haya notado; repasará por última vez los objetivos del capítulo y comentará cómo fue que se cubrieron.

16. Al finalizar los ejercicios, el instructor / facilitador pedirá al grupo que contesten el examen rápido, de manera individual. Se darán 3 minutos para contestar; después, en voz alta planteará cada una de las preguntas, y revisará qué contestó el grupo. Después de oír las diferentes posturas, dirá cuál es la respuesta correcta y por qué. Los exámenes son una herramienta de aprendizaje más, no una base para el premio y el castigo.

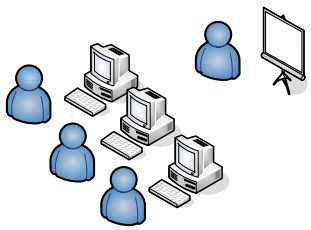
17. El instructor / facilitador podrá encargar las prácticas individuales como trabajo de refuerzo del conocimiento, ya sea para su desarrollo en sesión, o para que las personas desarrollen las prácticas en casa.

18. De ser posible, motive a los participantes a comprobar sus conocimientos en los diferentes cuestionarios y juegos que en línea se encuentran en **www.aprendapracticando.com**.

19. Si la infraestructura lo permite, puede solicitar a los miembros del grupo que vayan cubriendo la estrategia de aprendizaje de Aprenda Practicando en Línea.

Archivos complementarios del libro

En el sitio Web de Aprenda Practicando encontrará todos los archivos requeridos para el estudio de los temas del libro; además, podrá ver en línea los videos referidos en el texto. Si así lo desea, también podrá utilizar todas las herramientas de Aprenda Practicando En Línea (**APEL**), que ayudarán al aprendizaje estructurado del curso.



Todo el contenido de este libro, incluyendo los programas de implementación de los casos prácticos en Visual Basic.NET, C#, C++ y Java, pueden cubrirse en una semana, asistiendo al curso profesional:

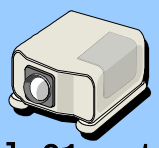
1008: Introducción a la Programación y su implementación usando Visual Basic.NET, C#, Java y C++.

Este curso es requerido para la certificación CAP-DNET (Certificación Aprenda Practicando – Desarrollador en Microsoft .NET), y puede cursarse en centros autorizados Aprenda Practicando.

Nota importante: Dependiendo de la demanda de los recursos educativos ofrecidos gratuitamente por Aprenda Practicando, y la respuesta de los lectores, estos pueden variar en su funcionamiento, agregarse o eliminarse, sin que sea necesaria notificación previa por escrito.

1

La computadora y los niveles de datos



lp01.ppt

Objetivos: **Identificará la composición básica de una computadora y los niveles de datos que se pueden manejar utilizándola.**

1. **Aprenderá los componentes generales de un sistema de cómputo.**
2. **Aprenderá cuáles son los dispositivos que se pueden controlar mediante la programación.**
3. **Aprenderá cuáles son los tipos de programa que se pueden desarrollar.**
4. **Aprenderá cuáles son los niveles de datos.**

Contenido

La computadora, como todas las herramientas, no piensa.

Imagine alguien que quiere ser cirujano y no se ha tomado la molestia de aprender bien cómo se compone el cuerpo humano. No obstante ello, se compra el herramental médico (bisturí, sierra, hilo y agujas) y se dispone a meterle cuchillo al primer paciente que se cruce en su camino. Queda claro que a medida que abra a unos cuantos pacientes irá aprendiendo cómo es el cuerpo humano, pero es seguro que no nos gustaría ser nosotros uno de sus pacientes.

Lo mismo sucede con muchas disciplinas, aunque sin los efectos drásticos del ejemplo anterior. La gente quiere lo que llamo yo "pensar con las manos", es decir, empezar a actuar sin saber qué está haciendo.

Precisamente uno de los principales problemas al aprender a programar es que se olvida el fin último de la programación, que es darle órdenes a una herramienta llamada computadora. Mucha gente no sabe qué es una computadora ni cómo se compone, y aún así se aventura a intentar programar; como es de esperarse el resultado es bastante mediocre.

Si además tomamos en cuenta que hoy en día las computadoras son muy variadas, más que nunca debemos hacernos a la idea que estamos tratando con muchas herramientas que debemos conocer de forma particular. Las computadoras se parecen en esencia, pero no todas son iguales. Médicamente hablando, los programadores de computadoras se parecerían más a un veterinario que a un médico de humanos, en el sentido que tendrán que aprender la composición de diferentes sujetos de trabajo, para luego poder manejarlos eficientemente.

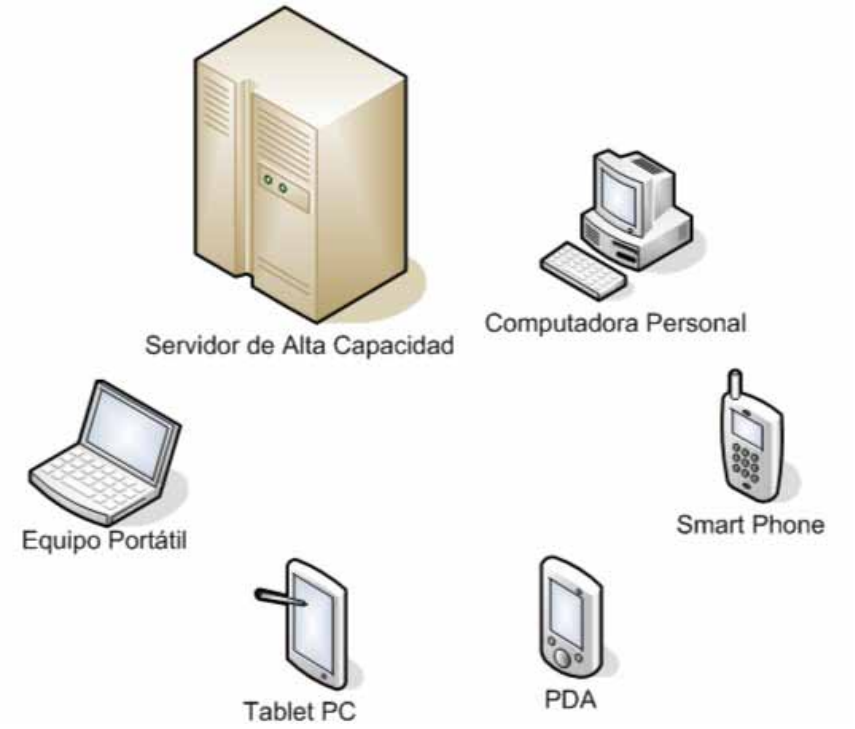
Computadora

Iniciamos definiendo a un *sistema* como el conjunto de elementos interrelacionados entre sí para alcanzar un mismo fin.

Podemos definir a una *computadora* como un sistema de dispositivos electrónicos, mecánicos y magnéticos que sirven para procesar y almacenar datos. Las computadoras reciben nombre comercial dependiendo de su tamaño o su uso, aunque en esencia son lo mismo. Tenemos las computadoras de escritorio, las computadoras portátiles, los PDA y celulares, etcétera.

Figura 01.01

Diferentes computadoras.



Una computadora es un sistema, dado que los dispositivos que la componen trabajan de forma interrelacionada, con el objetivo de apoyar el fin último de la computadora: procesar y almacenar datos.

Tipos de dispositivos

En su concepto básico, una computadora se forma de dispositivos de naturaleza electrónica, mecánica o magnética, y se clasifican de la siguiente manera.

- *Dispositivos de entrada*.

  Son los dispositivos que permiten proporcionar a la computadora los datos a procesar o almacenar, o bien, indicarle a la computadora la ejecución de acciones. Ejemplos típicos de este tipo de dispositivos son el teclado y el ratón.

- *Dispositivos de salida*.

  Son los dispositivos que permiten comunicar resultados de procesamiento al usuario de la computadora. El típico dispositivo de salida es la pantalla del monitor.

- *Dispositivos de procesamiento*.

  Son los dispositivos que se encargan del procesamiento de los datos. El principal dispositivo de procesamiento es lo que se conoce como *Unidad Central de Procesamiento* (*CPU* / *Central Processing Unit*), que contiene dos cosas muy importantes: la memoria RAM y el microprocesador.

  La *memoria RAM* (*Random Access Memory*) es de vital importancia, pues es el área de almacenamiento temporal en el que los programas deben ser cargados previamente a fin de que el microprocesador pueda trabajar con ellos. Nada puede trabajarse en la computadora si no está antes en la RAM. La memoria RAM se divide en *posiciones de memoria* a las que se puede tener acceso a través de *direcciones de memoria* (*Memory Address*), en las cuales se puede alojar (*allocate*) información en forma de datos de diferente tipo.

  Con respecto al *microprocesador*, es el que se encarga del trabajo de cómputo del equipo, a través de dos unidades internas: unidad aritmética y unidad lógica (de ahí que se le llame *unidad aritmético / lógica*).

  La unidad aritmética se encarga de las operaciones de suma, resta, división y multiplicación, mientras que la unidad lógica se encarga de resolver comparaciones y decisiones, conduciendo la información por los canales adecuados.

- *Dispositivos de almacenamiento permanente*.

  Son los dispositivos en los cuales podemos almacenar datos de manera persistente, es decir, que no se pierdan al momento de apagar el equipo.

- *Dispositivos periféricos*.

  Son los dispositivos de salida que no forman parte de la computadora, pero que pueden conectarse a ésta para ampliar su funcionalidad. Ejemplo de este tipo de dispositivo son las impresoras y los equipos biométricos.

Conocer los diferentes tipos de dispositivo es de vital importancia, ya que aprender a programar es aprender a controlar los dispositivos mediante instrucciones.

Al conjunto de dispositivos físicos que componen una computadora se le da el nombre de *hardware*.

Programas

El hardware por sí solo no sirve para nada. Es necesaria la existencia de programas que le indiquen a los dispositivos qué hacer.

Podemos identificar los siguientes tipos de programas controlando a una computadora.

- *Sistema operativo*.

  Es el conjunto de programas y datos que permiten reconocer, identificar y utilizar los dispositivos de una computadora. Generalmente, los otros tipos de programas se comunican con el sistema operativo, y es éste el que comunica las órdenes a los dispositivos.

- *Programas aplicativos*.

  Es el conjunto de programas y datos que utilizan de forma genérica las capacidades de la computadora, para realizar tareas específicas. Generalmente se trata de programas comerciales como Microsft Word, Macromedia Flash, etcétera; se caracterizan por permitir la creación de archivos autónomos de trabajo.

- *Sistemas de información*.

  Es el conjunto de programas y datos que permiten utilizar las capacidades de procesamiento y almacenamiento de la computadora, con el fin de generar, manipular y divulgar información. Generalmente se trata de programas desarrollados en la misma organización, o adquiridos a terceros. Se caracterizan por no producir archivos autónomos de trabajo, sino por consumir bases de datos.

A todos los elementos físicamente intangibles que participan en el funcionamiento de la computadora se les da el nombre de *software*.

A un determinado hardware funcionando a través de un determinado software, se le da el nombre de *entorno operativo* o *plataforma operativa*. Los entornos operativos son muy útiles al momento de verificar compatibilidad de los programas y los equipos; generalmente cuando usted requiera soporte le preguntarán qué equipo (marca y modelo) está usando, y qué sistema operativo (marca, versión, actualizaciones) es el que tiene instalado. En otras palabras, para dar soporte se requiere conocer el entorno operativo. También cuando compre software encontrará seguramente los requerimientos mínimos del programa, que generalmente harán referencia a una plataforma operativa específica.

Niveles de datos

Los datos manejados por una computadora pueden ser entendidos de dos maneras distintas: desde el punto de vista físico y desde el punto de vista de relevancia.

El punto de vista físico tiene que ver con el consumo de áreas de almacenamiento que los datos ocupan, mientras que desde el punto de vista de relevancia, lo que importa es el significado y el valor que el usuario le otorga a los datos.

En términos físicos

Internamente las computadoras manejan los datos en formato binario (0 y 1).

Las computadoras desconocen si lo almacenado es importante o no, e incluso desconocen el significado de lo que almacenan.

Bits. Se conoce como *bit* a los ceros y unos manejados internamente por la computadora. Los bits son la unidad básica de almacenamiento y comunicación de información dentro de la computación, y son procesados con suma eficiencia por el procesador de la computadora. Una operación tan simple como una suma o una comparación pueden consumir capacidad de procesamiento, de ahí que los procesadores actuales deben ser capaces de realizar millones de operaciones por segundo.

Se tienen los siguientes símbolos y *unidades de medida para los bits*.

kilobit	kbit	2^{10} bits
megabit	Mbit	2^{20} bits
gigabit	Gbit	2^{30} bits
terabit	Tbit	2^{40} bits
petabit	Pbit	2^{50} bits

Bytes. Un conjunto de 8 bits hacen un *byte*, que en esencia equivalen a un símbolo o caracter legible a los humanos.

¿Por qué son 8 bits los que componen un byte, y no 7 ó 9?

La respuesta es una cuestión de economía. En los orígenes de la computación, dado lo limitado de los recursos de las computadoras, había que decidir por el mínimo número de bits que compondrían a un símbolo, pero que al mismo tiempo permitiera un conjunto de combinaciones suficientes para manejar todos los símbolos utilizados en un idioma.

Dado el manejo binario de las computadoras, las opciones eran las siguientes:

$2^1=2$
$2^2=4$
$2^3=8$
$2^4=16$
$2^5=32$
$2^6=64$
$2^7=128$
$2^8=256$
$2^9=512$
$2^{10}=1024$
...

Se llegó al consenso que 2^8 proporcionaba un juego de caracteres de 256 símbolos, en los cuales era posible almacenar las letras mayúsculas (A, B, C, etc.), minúsculas (a, b, c, etc.), y símbolos especiales (¡, @, #, $, %, &, etc.) Esto al menos en su versión occidental.

A cada juego de 256 símbolos se le denomina *página de códigos*. Las páginas de códigos de un solo byte contienen un máximo de 256 valores. Una página de códigos con un límite de 256 caracteres no puede adaptarse a todos los idiomas porque para ello sería necesario utilizar mucho más de 256 caracteres. Por lo tanto, los distintos alfabetos utilizan páginas de códigos separadas. Hay una página de códigos para el griego, otra para el japonés, otra para el español, etcétera.

Aunque un código tan reducido puede ser suficiente para idiomas como el Hawaiano, que sólo tiene 12 caracteres, puede ser muy limitado para otros idiomas más complejos. Las páginas de códigos de un solo byte resultan insuficientes para la mayoría de los idiomas asiáticos, ya que éstos utilizan normalmente más de 5,000 caracteres basados en el Chino. Afortunadamente ya se han desarrollado páginas de códigos de doble byte, con el fin de obtener compatibilidad con estos idiomas.

NOTA

En la actualidad, dado que las aplicaciones pueden ser multilingües gracias a la globalización, 256 caracteres no son suficientes para representar todos los diferentes símbolos que se pueden utilizar. Afortunadamente el almacenamiento ya no es un problema tan grave, por lo que propuestas como *UNICODE* permiten asignar una codificación basada en 2 bytes, con lo que es posible almacenar 2^{16} combinaciones, lo que es bastante. Investiga más acerca de UNICODE.

Los bytes, tienen los siguientes símbolos y unidades de medida.

kilobyte	kB	2^{10} bytes
megabyte	MB	2^{20} bytes
gigabyte	GB	2^{30} bytes
terabyte	TB	2^{40} bytes
petabyte	PB	2^{50} bytes

En términos de relevancia

Es más importante la imaginación que el conocimiento

Albert Einstein

Por *valor* entendemos la utilidad o deleite que nos proporciona algo, al menos en nuestra percepción.

Somos las personas quienes le otorgamos a los datos el carácter de valiosos o irrelevantes. Los datos pueden tener diferente nivel de utilidad para las personas, y esta utilidad generalmente está asociada al significado que tienen para nosotros.

Podemos proponer los siguientes niveles de los datos, en cuanto a la relevancia que tienen para nosotros:

- *Dato.*

  Es el conjunto de representaciones simbólicas no significativas, dado que no tenemos la capacidad de reconocerles un significado más allá del que tienen los símbolos que componen el dato.

- *Información.*

  Es el conjunto de datos que en determinada cantidad y forma aumenta el conocimiento o reduce la incertidumbre respecto a un sujeto, evento o circunstancia. La información sólo podrá ser considerada información si tenemos capacidad de reconocerla.

- *Conocimiento.*

  Es la información, cuando tenemos capacidad para usarla en actividades del plano real y limitado. El conocimiento tiene naturaleza práctica.

- *Imaginación.*

  Es la información, cuando tenemos capacidad para usarla en actividades del plano irreal e ilimitado. La imaginación tiene naturaleza teórica.