

**Libro:** Como programar en C++, H.M. Deitel

**Capítulo:** 1. Introducción a las computadoras y C++

---

## 1.1 Introducción

---

C++ es uno de los lenguajes de desarrollo de software más populares de la actualidad. El explosivo crecimiento del mercado de aplicaciones para teléfonos inteligentes y tabletas está creando oportunidades significativas para programar aplicaciones de este tipo.

## 1.2 Ordenadores e Internet en la industria y la investigación

---

Muchas de las empresas más influyentes y exitosas de las dos últimas décadas son empresas tecnológicas. Algunas formas en las que los ordenadores mejoran la vida de las personas en la investigación, la industria y la sociedad:

- **Registros médicos electrónicos.** Tener acceso a una red segura que brinde el historial médico de un paciente reduce la probabilidad de error, al tiempo que incrementa la eficiencia del sistema de salud, controlando los gastos.
- **Proyecto del Genoma Humano.** Uso de programas computarizados para analizar los 20, 000+ genes del ADN humano.
- **Alerta AMBER.** Sistema de alerta, utilizado para encontrar infantes desaparecidos, que involucra señales computarizadas, internet y dispositivos inalámbricos, así como de redes sociales.
- **World Community Grid.** Donación de poder de procesamiento computacional para el uso de supercomputadoras.
- **Informática en la nube.** Permite utilizar el software, el hardware y la información almacenada en la "nube", en vez de en el ordenador.
- **Imagenología médica.** Optimización de los escaneos utilizados en estudios de imagenología médica; creación de imágenes 3D.
- **GPS.** Sistema de Posicionamiento Global que utiliza una red de satélites para recuperar la información basada en ubicación.
- **Robots.** Pueden utilizarse para tareas cotidianas, entretenimiento, combate militar, exploración de las profundidades del mar y del espacio, etc.
- **Correo electrónico, mensajería instantánea, videochat y FTP.** Los servidores basados en Internet soportan toda la mensajería en línea. El FTP (protocolo de transferencia de archivos) permite intercambiar archivos entre varios ordenadores a través de Internet.
- **TV por internet.** Permiten acceder a una enorme cantidad de contenidos en demanda y ayudan a garantizar que los contenidos se transmitan al televisor sin problemas.
- **Servicios de música en streaming.** Permite escuchar grandes catálogos de música a través de la web, crear "emisoras de radio" personalizadas y descubrir nueva música en función de los comentarios.
- **Programación de videojuegos**

## 1.3 Hardware y Software

---

Las computadoras personales pueden realizar billones de cálculos en un segundo, mientras que las supercomputadoras ejecutan miles de trillones de instrucciones por segundo. La

supercomputadora Tianhe-2 puede realizar 3 millones de cálculos por segundo por cada persona en el planeta. Y los límites de la supercomputación crecen rápidamente.

- **Programas de computadora.** Secuencias de instrucciones a partir de las cuales las computadoras procesan la información.
- **Programador.** Persona que especifica las instrucciones de los programas.
- **Software.** Programas que corren en una computadora.

Una computadora consiste de varios dispositivos conocidos como **hardware** (teclado, monitor, ratón, disco duro, memoria, puerto DVD y unidades de procesamiento). Los costos de la informática se están reduciendo drásticamente gracias a la rápida evolución de las tecnologías de hardware y software. La tecnología de los chips de silicio ha hecho que la informática se tan económica que los ordenadores se han convertido en una comodidad.

### 1.3.1 Ley de Moore

Aproximadamente, cada uno o dos años, las capacidades de los ordenadores se han duplicado económicamente. - Gordon Moore, co-founder of Intel (1960s). Se aplica especialmente a la cantidad de memoria que tienen los ordenadores para los programas, la cantidad de almacenamiento secundario (en disco) que tienen para mantener los programas y los datos por más tiempo, así como la velocidad de sus procesadores. Éstos hacen que los ordenadores sean más capaces, fomentando la innovación. Se dio un crecimiento similar en las comunicaciones: hay una intensa competencia en la demanda de ancho de banda. Dichas mejoras fomentan la Revolución de la Información.

### 1.3.2 Organización de la computadora

La computadora está organizada en varias **unidades lógicas** o secciones:

Unidad Lógica	Descripción
Unidad de entrada (Input unit)	Sección de "recepción" que obtiene la información (datos y programas informáticos) de los dispositivos de entrada y la pone a disposición de las demás unidades para su procesamiento.
Unidad de salida (Output unit)	Sección de "envío" que toma la información que el ordenador ha procesado y la coloca en varios dispositivos de salida para que esté disponible para su uso fuera del ordenador.
Unidad de memoria (Memory unit)	Sección de "almacén" de acceso rápido y capacidad relativamente baja que retiene la información que ha sido introducida a través de la unidad de entrada, haciéndola inmediatamente disponible para su procesamiento cuando sea necesario. También retiene la información procesada hasta que puede ser colocada en los dispositivos de salida por la unidad de salida. La información de la unidad de memoria es volátil (normalmente se pierde cuando se apaga el ordenador).
Unidad aritmética y lógica (ALU, Arithmetic and logic unit)	Sección de "fabricación" que realiza los cálculos (suma, resta, multiplicación y división). También contiene los mecanismos de decisión.
Unidad central de procesamiento (CPU, Central Processing Unit)	Sección "administrativa" que coordina y supervisa el funcionamiento de las demás secciones. Un procesador multinúcleo implementa varios procesadores en un único chip de circuito integrado.
Unidad de almacenamiento secundaria (Secondary storage unit)	Sección de "almacenamiento" a largo plazo y de gran capacidad. Los programas o datos que no son utilizados activamente por las otras unidades se colocan en dispositivos de almacenamiento secundario (disco duro) hasta que se vuelvan a necesitar. La información de los dispositivos de almacenamiento secundario es persistente (se conserva incluso cuando el ordenador está apagado). El acceso a la información del almacenamiento secundario es mucho más largo que el de la memoria primaria, pero su coste por unidad es mucho menor.

## 1.4 Jerarquía de datos

Los elementos de datos procesados por los ordenadores forman una **jerarquía de datos** que se hace más grande y más compleja en su estructura a medida que avanzamos de los elementos de datos más simples a los más ricos.

Data item	Description
Bit	El elemento de datos más pequeño de un ordenador que asume el valor 0 o el valor 1 (dígito binario)
Character	Dígitos, letras y símbolos especiales. Conjunto de caracteres: conjunto de todos los caracteres utilizados para escribir programas y representar elementos de datos (como un patrón de 1s y 0s).
Field	Grupo de caracteres o bytes que transmite un significado
Record	Grupo de campos (fields) relacionados
File	Grupo de registros (records) relacionados; contienen datos arbitrarios en formatos arbitrarios
Database	Colección de datos organizada para facilitar su acceso y manipulación. Modelo más popular: base de datos relacional (tablas), que incluye registros y campos
Big Data	Cantidades masivas de datos

### Mediciones de bytes

Unidad	Bytes	Cantidad exacta
1 kilobyte (KB)	1024 bytes	$10^3$
1 megabyte (MB)	1024 kilobytes	$10^6$
1 gigabyte (GB)	1024 megabytes	$10^9$
1 terabyte (TB)	1024 gigabytes	$10^{12}$
1 petabyte (PB)	1024 terabytes	$10^{15}$
1 exabyte (EB)	1024 petabytes	$10^{18}$
1 zettabyte (ZB)	1024 exabytes	$10^{21}$

## 1.5 Lenguajes de máquina, lenguajes ensambladores y lenguajes de alto nivel

Tipo de lenguaje	Descripción
Machine	Consiste de números (1s y 0s); engorroso para los humanos
Assembly	Abreviaturas en inglés para representar operaciones elementales; más claro para humanos, incomprensible para computadoras hasta que son traducidos por los <b>ensambladores</b> a lenguaje de máquina
High-Level	Instrucciones simples para realizar tareas importantes. Dichas instrucciones se parecen más al inglés cotidiano y contienen notaciones matemáticas de uso común

- **Compiladores.** Programas traductores que convierten los programas de lenguaje de alto nivel en lenguaje de máquina.
- **Interpreter programs**(Programas de interpretación). Ejecutan programas de lenguaje de alto nivel directamente, sin necesidad de compilación, aunque más lentamente que los programas compilados. Procesan **Scripting languages** (JavaScript, PHP).

**Tip.**Un programa interpretado puede comenzar a ejecutarse tan pronto como se descargue en la máquina del cliente, sin necesidad de ser compilado antes de poder ejecutarse; los scripts interpretados se ejecutan más lentamente y consumen más memoria que el código compilado.

## 1.6 C & C++

---

C fue implementado en 1972 por Dennis Ritchie en los Laboratorios Bell. Es conocido como el lenguaje de desarrollo del sistema operativo UNIX. C++ evolucionó a partir de C. La Organización Internacional de Normalización (ISO) para estandarizar el C en todo el mundo se unió a la Organización Internacional de Normalización (ISO) para estandarizar el C en todo el mundo (1990). C11 es el último estándar ANSI para el lenguaje de programación C. Evoluciona el lenguaje C para seguir el ritmo de un hardware cada vez más potente y de unos requisitos de usuario cada vez más exigentes. También hace que C sea más coherente con C++.

C++, extensión de C, fue desarrollada por Bjarne Stroustrup en 1979 en los Laboratorios Bell. Originalmente se llamaba "C con clases", y fue rebautizado como C++ (a principios de la década de 1980). C++ ofrece una serie de características que "arreglan" el lenguaje C; proporciona capacidades para la programación orientada a objetos que se inspiraron en el lenguaje de simulación Simula lenguaje de programación de simulación.

### C++ Standard Library

Los programas C++ se componen de piezas llamadas clases y funciones; la Biblioteca Estándar C++ contiene ricas colecciones de ellas. El aprendizaje del "mundo" de C++ consta de dos partes: el aprendizaje del propio lenguaje C++ (núcleo del lenguaje) y el aprendizaje del uso de las clases y funciones de la biblioteca estándar de C++.

### Observaciones

- **Reutilización del software.**Enfoque de "bloques de construcción" para crear programas. Utilizar piezas existentes siempre que sea posible. Práctica fundamental para una programación orientada a objetos eficaz.
- **Clases and funciones.** Al programar en C++, normalmente utilizará los siguientes bloques de construcción: clases y funciones de la biblioteca estándar de C++, creadas por usted y sus colegas, y de varias bibliotecas populares de terceros.

La ventaja de crear tus propias funciones y clases es que sabrás exactamente cómo funcionan. Desventaja: el tiempo y el complejo esfuerzo que supone diseñar, desarrollar y mantener nuevas funciones y clases que sean correctas y funcionen de forma eficiente.

### Tips

- El uso de funciones y clases de la biblioteca estándar de C++ en lugar de escribir sus propias versiones mejora el rendimiento del programa; también acorta el tiempo de desarrollo del programa.
- Utilizar las funciones y clases de la biblioteca estándar de C++ en lugar de escribir las propias mejora la portabilidad del programa, ya que están incluidas en todas las implementaciones de C++.

## 1.7 Lenguajes de programación

---

Algunos lenguajes de programación populares que soportan la programación orientada a objetos:

Lenguaje	Descripción
Fortran	(FORMula TRANslator) desarrollado por IBM Corporation (a mediados de la década de 1950) utilizado para aplicaciones científicas y de ingeniería que requieren cálculos matemáticos complejos.
COBOL	(COMmon Business Oriented Language) desarrollado a finales de la década de 1950 por los fabricantes de ordenadores, el gobierno de EE.UU. y los usuarios de ordenadores industriales, basado en un lenguaje desarrollado por Grace Hopper, utilizado para aplicaciones comerciales que requieren una manipulación precisa y eficiente de grandes cantidades de datos
Pascal	Programación estructurada (enfoque disciplinado para escribir programas más claros, más fáciles de probar y depurar y más fáciles de modificar que los programas producidos con técnicas anteriores). Desarrollado por el profesor Niklaus Wirth en 1971, es popular para la enseñanza de la programación estructurada.
Ada	basado en Pascal, desarrollado bajo el patrocinio del Departamento de Defensa de Estados Unidos (DOD) durante los años 70 y principios de los 80. Lleva el nombre de Lady Ada Lovelace: Se le atribuye la escritura del primer programa informático del mundo a principios del siglo XIX (para el dispositivo informático mecánico Analytical Engine diseñado por Charles Babbage)
Basic	desarrollado en los años 60 en el Dartmouth College para familiarizar a los novatos con las técnicas de programación.
Objective-C	Basado en C. Desarrollado a principios de los años 80 y posteriormente adquirido por NeXT, que a su vez fue adquirida por Apple. Lenguaje de programación clave para el sistema operativo OS X y todos los dispositivos con iOS.
Swift	2014, es el lenguaje de programación del futuro de Apple para el desarrollo de aplicaciones (apps) para iOS y OS X. lenguaje contemporáneo que incluye características populares de lenguajes de programación de Objective-C, Java, C#, Ruby, Python y otros. de código abierto (también se puede utilizar en plataformas que no sean de Apple).
Java	En 1991, Sun Microsystems financió un proyecto de investigación interno de la empresa dirigido por James Gosling, que dio lugar al lenguaje de programación orientado a objetos Java, basado en C++. Objetivo principal: permitir a los desarrolladores escribir programas que se ejecuten en una gran variedad de sistemas informáticos y dispositivos controlados por ordenador. "Escribir una vez, ejecutar en cualquier lugar". Se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de los servidores web, para proporcionar aplicaciones para dispositivos de consumo. lenguaje clave para desarrollar aplicaciones para smartphones y tabletas Android.
Visual Basic	Visual Basic de Microsoft se introdujo a principios de los años 90 para simplificar el desarrollo de aplicaciones de Microsoft Windows.

Lenguaje	Descripción
C#	Los tres principales lenguajes de programación orientados a objetos de Microsoft son C#, Visual C++ y Visual Basic. C# se desarrolló para integrar la web en las aplicaciones informáticas, y ahora se utiliza ampliamente para desarrollar aplicaciones empresariales y para el desarrollo de aplicaciones móviles.
PHP	Código abierto. Lenguaje de "scripting" apoyado por una comunidad de desarrolladores, utilizado por numerosos sitios web. Independiente de la plataforma.
Python	Lenguaje de scripting. Lanzado públicamente en 1991. Desarrollado por Guido van Rossum. Se basa en gran medida en Modula-3. "extensible" (puede ampliarse mediante clases e interfaces de programación)
JavaScript	Lenguaje de scripting más utilizado. Se usa principalmente para añadir programabilidad a las páginas web. se suministra con los principales navegadores web.
Ruby on Rails	Ruby -creado a mediados de la década de 1990 por Yukihiro Matsumoto- es un lenguaje de programación de código abierto orientado a objetos con una sintaxis sencilla similar a la de Python. Ruby on Rails combina el lenguaje de scripting Ruby con el framework de aplicaciones web Rails, desarrollado por la empresa 37Signals. Es bueno para desarrollar aplicaciones web con muchas bases de datos.
Scala	"Lenguaje escalable" diseñado por Martin Odersky. Lanzado en 2003. Utiliza paradigmas de programación orientada a objetos y de programación funcional; diseñado para integrarse con Java. reduce la cantidad de código en sus aplicaciones de forma significativa.

## 1.8 Introducción a la tecnología de objetos

Los objetos, o más exactamente -las clases de las que proceden los objetos-, son componentes de software reutilizables. Casi cualquier sustantivo puede representarse como objeto de software en términos de atributos (nombre, color, tamaño) y comportamientos (calcular, moverse). El enfoque modular de diseño e implementación orientado a objetos puede hacer que los grupos de desarrollo de software sean mucho más productivos de lo que era posible con técnicas anteriores.

- **Functions, Member Functions and Classes**(Funciones, funciones miembro y clases). Creamos una unidad de programa llamada clase para alojar el conjunto de funciones que realizan las tareas de la clase (funciones miembro de la clase). Una clase es similar a los planos de ingeniería de un coche, que albergan el diseño del pedal del acelerador, el pedal del freno, etc.
- **Instantation.** Construir un objeto a partir de una clase antes de que un programa pueda realizar las tareas que definen las funciones miembro de la clase. Un objeto se denomina entonces instancia de su clase.
- **Reuso.** La reutilización de las clases existentes al construir nuevas clases y programas ahorra tiempo y esfuerzo; ayuda a construir sistemas más fiables y eficaces: han pasado por extensas pruebas, depuración y ajuste de rendimiento. Las clases reutilizables son cruciales para la revolución del software que ha impulsado la tecnología de objetos.



- **Messages and Member-Function Calls** (Mensajes y llamadas a funciones miembro). Enviar mensajes a un objeto. Cada mensaje se implementa como una llamada a una función miembro que indica a una función miembro del objeto que realice su tarea.
- **Attributes and Data Members.** (Atributos y miembros de datos) Un objeto tiene atributos que lleva consigo cuando se utiliza en un programa. Estos atributos se especifican como parte de la clase del objeto. Los atributos son especificados por los miembros de datos de la clase.
- **Encapsulation.** Las clases encapsulan (envuelven) los atributos y las funciones miembro en objetos creados a partir de esas clases: los atributos y las funciones miembro de los objetos están íntimamente relacionados. Los objetos pueden comunicarse entre sí, pero no pueden saber cómo se implementan otros objetos: los detalles de implementación se ocultan dentro de los propios objetos. la ocultación de información es crucial para una buena ingeniería de software.
- **Inheritance.** Se puede crear una nueva clase de objetos de forma rápida y cómoda mediante la herencia: la nueva clase absorbe las características de una clase existente, personalizándolas y añadiendo características propias.

**Análisis y diseño orientado a objetos (OOAD: Object-Oriented Analysis and Design).** Código: instrucciones del programa. Para crear las mejores soluciones, hay que seguir un proceso de análisis detallado para determinar los requisitos de su proyecto (definir lo que debe hacer el sistema) y desarrollar un diseño que los satisfaga (cómo debe hacerlo el sistema). revisar el diseño por otros profesionales del software antes de escribir cualquier código. Si este proceso implica analizar y diseñar el sistema desde un punto de vista orientado a objetos, se denomina proceso de análisis y diseño orientado a objetos (OOAD). C++ está orientado a objetos. La programación en un lenguaje de este tipo, llamado programación orientada a objetos (POO), permite implementar un diseño orientado a objetos como un sistema de trabajo.

**Lenguaje de Modelado Unificado (UML: Unified Modeling Language).** Lenguaje gráfico único para comunicar los resultados de cualquier proceso OOAD. Es el esquema gráfico más utilizado para modelar sistemas orientados a objetos.

## 1.9 Entorno de desarrollo típico de C++

---

Los sistemas C++ constan de tres partes: entorno de desarrollo de programas, lenguaje y biblioteca estándar C++. Los programas C++ pasan por seis fases: editar, preprocesar, compilar, enlazar, cargar y ejecutar. Típico entorno de desarrollo de programas C++.

Fase	Descripción
1. Edición	El programador crea el programa en el editor, realiza las correcciones necesarias y lo guarda en el disco.
2. Preprocesamiento	El programa de preprocesamiento se ejecuta automáticamente antes de que comience la fase de traducción del compilador. El preprocesador C++ obedece a comandos directivas de preprocesamiento (indican que se deben realizar ciertas manipulaciones en el programa antes de la compilación). Estas manipulaciones incluyen otros archivos de texto que se van a compilar, y realizan diversas sustituciones de texto.
3. Compilación	El compilador traduce el programa C++ a código objeto (lenguaje de máquina) y lo almacena en el disco.
4. Linking (Enlazar)	Se enlaza el código objeto con las bibliotecas, creándose un archivo ejecutable que se almacena en el disco. Un enlazador enlaza el código objeto con el código de las funciones que faltan para producir un programa ejecutable (sin piezas que falten). Si el programa se compila y enlaza correctamente, se produce una imagen ejecutable.
5. Loading (cargar)	El cargador (loader) pone el programa en la memoria: toma la imagen ejecutable del disco y la transfiere a la memoria. También se cargan los componentes adicionales de las bibliotecas compartidas que soportan el programa.
6. Ejecución	La CPU toma cada instrucción y la ejecuta (una a la vez), almacenando nuevos valores de datos mientras se ejecuta el programa. Algunas arquitecturas de ordenador modernas ejecutan varias instrucciones en paralelo.

**Entornos de desarrollo integrado** (IDEs: Integrated development environments). Los IDEs proporcionan herramientas que apoyan el proceso de desarrollo de software, incluyendo editores para escribir y editar programas y depuradores para localizar errores lógicos - errores que causan que los programas se ejecuten incorrectamente. errores que causan que los programas se ejecuten incorrectamente.

#### Problemas que pueden surgir en el momento de la ejecución

- Operaciones ilegales para la aritmética de enteros en C++
- Algunas funciones de C++ toman su entrada de **cin** (flujo de entrada estándar; se pronuncia "see-in"), pero **cin** puede ser redirigido a otro dispositivo. Los datos salen por **cout** (flujo de salida estándar; se pronuncia "see-out", pantalla del ordenador), pero **cout** puede redirigirse a otro dispositivo.
- **Flujo de error estándar (cerr)**. El flujo **cerr** (conectado a la pantalla) se utiliza para reproducir mensajes de error.
- **Errores en tiempo de ejecución**. Se producen mientras se ejecuta un programa.
- **Errores de ejecución fatales**. Hacen que los programas terminen inmediatamente sin haber realizado con éxito su trabajo.
- **Errores de ejecución no fatales**. Permiten que los programas se ejecuten hasta el final, produciendo a menudo resultados incorrectos.

## 1.10 Prueba de una aplicación C++

Pasos:

1. Revisar la configuración
2. Iniciar Visual Studio
3. Crear un proyecto (grupo de archivos relacionados). Visual Studio organiza las aplicaciones en proyectos y soluciones, que contienen uno o más proyectos. Para crear un proyecto:
  1. Selecciona File > New > Project....
  2. En la parte izquierda del cuadro de diálogo New proyecto, seleccione la categoría Installed > Templates > Visual C++ > Win32
  3. En la sección central del diálogo New project, seleccione Win32 Console Application.
  4. Proporcione un nombre para su proyecto en el Name field. OK para mostrar el Win32 Application Wizard window. Next > para mostrar el paso de Configuración de la aplicación.
  5. Configure los ajustes para crear una solución que contenga un proyecto vacío y haga clic en Finalizar.
4. Agregar el archivo GuessNumber.cpp al Proyecto:
  1. Clic derecho del ratón en el folder Source Files y seleccionar Add > New Item... para mostrar un diálogo para añadir un nuevo archivo.
5. Compilación y ejecución del proyecto
  1. Debug > Start without debugging o teclear Ctrl + F5. Si el programa se compila correctamente, el IDE abre una ventana de símbolo del sistema y ejecuta el programa.
6. Introducir su primera estimación.
7. Introducir otra estimación.
8. Introducción de estimaciones adicionales
9. Volver a jugar o salir de la aplicación (y o n)

## 1.11 Sistemas operativos

---

Sistemas de software que hacen más cómodo el uso de los ordenadores. Proporcionan servicios que permiten que cada aplicación se ejecute de forma segura, eficiente y concurrente (en paralelo) con otras aplicaciones. **Kernel:** software que contiene los componentes principales del sistema operativo.

### 1.11.1 Windows - Sistema Operativo Propietario

A mediados de la década de 1980, Microsoft desarrolló el sistema operativo Windows: era una interfaz gráfica de usuario construida sobre DOS (Disk Operating System). Windows tomó prestados muchos conceptos (iconos, menús y ventanas) desarrollados por Xerox PARC y popularizados por los primeros sistemas operativos Macintosh de Apple. Es un sistema operativo propietario (controlado exclusivamente por Microsoft). Es el sistema operativo de escritorio más utilizado del mundo.

### 1.11.2 Linux—An Open-Source Operating System

El mayor éxito del movimiento de código abierto (individuos y empresas contribuyen con sus esfuerzos en el desarrollo, mantenimiento y evolución del software a cambio del derecho a utilizar ese software para sus propios fines, normalmente sin coste alguno; examinado por un público mucho más amplio que el software propietario, por lo que los errores se eliminan más rápido; fomenta la innovación).

Las rápidas mejoras en la informática y las comunicaciones, la disminución de los costes y el software de código abierto han hecho que sea mucho más fácil y económico crear una empresa basada en software ahora que hace una década.

El kernel de Linux es el núcleo del sistema operativo más popular de código abierto, de libre distribución y con todas las funciones, desarrollado por un equipo de voluntarios poco organizado y popular en servidores, ordenadores personales y sistemas integrados (en teléfonos y televisiones inteligentes, automóviles). El código fuente de Linux (el código del programa) está a disposición del público para su examen y modificación, y su descarga e instalación son gratuitas; los usuarios de Linux se benefician de una enorme comunidad de desarrolladores que depuran y mejoran activamente el núcleo, y de la posibilidad de personalizar el sistema operativo para satisfacer necesidades específicas.

### **1.11.3 Apple's OS X; Apple's iOS para iPhone®, iPad® e iPod Touch®**

- Apple es fundada por Steve Jobs y Steve Wozniak en 1976.
- Jobs y varios empleados de Apple, en 1979, visitaron el Xerox PARC (Centro de Investigación de Palo Alto) para conocer el ordenador de sobremesa de Xerox, que contaba con una interfaz gráfica de usuario (GUI), la cual sirvió de inspiración para el Apple Macintosh (1984).
- El lenguaje de programación Objective-C, creado por Brad Cox y Tom Love en Stepstone a principios de la década de 1980, añadió capacidades de programación orientada a objetos (POO) al lenguaje de programación C.
- Steve Jobs dejó Apple en 1985 y fundó NeXT Inc.
- En 1988, NeXT obtuvo la licencia de Objective-C de StepStone y desarrolló un compilador y bibliotecas de Objective-C que se utilizaron como plataforma para la interfaz de usuario de NeXTSTEP, así como Interface Builder (para interfaces gráficas de usuario).
- Jobs regresó a Apple en 1996, cuando ésta compró NeXT.
- OS X de Apple es descendiente de NeXTSTEP. iOS deriva del OS X de Apple y se utiliza en los dispositivos iPhone, iPad y iPod Touch. En 2014, Apple presentó Swift, que se convirtió en código abierto en 2015. La comunidad de desarrolladores de aplicaciones para iOS está pasando gradualmente de Objective-C a Swift.

### **1.11.4 Google's Android**

Android -sistema operativo para dispositivos móviles inteligentes de más rápido crecimiento- se basa en el núcleo de Linux y Java. Sus aplicaciones también pueden desarrollarse en C++ y C. Ventaja: apertura de la plataforma.

El sistema operativo es de código abierto y gratuito; fue desarrollado por Android, Inc. (adquirida por Google en 2005). En 2007, se formó la Open Handset Alliance™ para desarrollar, mantener y evolucionar Android. El sistema operativo se utiliza en numerosos teléfonos inteligentes, dispositivos de lectura electrónica, tabletas, quioscos con pantalla táctil en tiendas, coches, robots, reproductores multimedia y mucho más.

## **1.12 El Internet y la World Wide Web**

---

A finales de la década de 1960, ARPA (Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa de Estados Unidos) buscó conectar en red los principales sistemas informáticos de una docena de universidades e instituciones de investigación financiadas por ARPA; procedió a poner ARPANET (precursor de la actual Internet) en marcha. Aunque está permitió a los investigadores conectar sus ordenadores en red, su principal ventaja fue la capacidad de comunicación rápida y sencilla a través de correo electrónico.

El protocolo (conjunto de reglas) para comunicarse a través de ARPANET se conoce como Protocolo de Control de Transmisión (TCP). El TCP garantizaba que los mensajes, formados por paquetes (piezas numeradas secuencialmente), se dirigieran correctamente del emisor al receptor, llegaran intactos y se ensamblaran en el orden correcto.

### **Internet: Una red de redes**

Organizaciones de todo el mundo estaban implementando sus propias redes para la comunicación tanto intraorganizacional (dentro) como interorganizacional (entre organizaciones). Uno de los retos era permitir que estas diferentes redes se comunicaran entre sí. ARPA lo consiguió desarrollando el Protocolo de Internet (IP), que creó una "red de redes", la arquitectura de Internet. El conjunto de protocolos se denomina ahora TCP/IP.

Las empresas gastaron grandes cantidades de dinero para desarrollar y mejorar su presencia en Internet. Como resultado, el ancho de banda (capacidad de transporte de información de las líneas de comunicación) en Internet ha aumentado enormemente, mientras que los costes de hardware han caído en picada.

### **La World Wide Web: Hacer que Internet sea "fácil de usar"**

La World Wide Web ("la web") es un conjunto de hardware y software asociado a Internet que permite a los usuarios de ordenadores localizar y visualizar documentos basados en multimedia (texto, gráficos, animaciones, audios y vídeos) sobre cualquier tema.

En 1989, Tim Berners-Lee, del CERN (Organización Europea para la Investigación Nuclear), comenzó a desarrollar una tecnología para compartir información a través de documentos de texto "hipervinculados" (HTML: HyperText Markup Language). También escribió protocolos de comunicación: Protocolo de Transferencia de Hipertexto (HTTP) para formar la columna vertebral de su nuevo sistema de información de hipertexto (World Wide Web).

En 1994, Berners-Lee fundó el Consorcio de la World Wide Web (W3C), dedicado a desarrollar tecnologías web. Uno de sus principales objetivos es hacer que la web sea universalmente accesible para todo el mundo, independientemente de su discapacidad, idioma o cultura.

**Servicios web.** Son componentes de software almacenados en un ordenador a los que puede acceder una aplicación (u otro componente de software) en otro ordenador a través de Internet.

- Mashups. Permiten desarrollar rápidamente aplicaciones combinando servicios web complementarios, de múltiples organizaciones. Ej. 100 Destinations

**Ajax.** La tecnología Ajax ayuda a que las aplicaciones basadas en Internet funcionen como las de escritorio.

**Internet de las cosas.** "Cosa" es cualquier objeto con una dirección IP y la capacidad de enviar datos automáticamente a través de Internet.

## **1.13 Algunos términos clave del desarrollo de software**

---

Tecnología	Descripción
Agile software development (desarrollo ágil de software )	Conjunto de metodologías que tratan de conseguir que el software se implemente más rápidamente, utilizando menos recursos.
Refactoring (refactorización )	Reelaborar los programas para hacerlos más claros y fáciles de mantener, conservando su corrección y funcionalidad.
Design patterns (patrones de diseño)	Arquitecturas probadas para construir software orientado a objetos flexible y mantenible.
LAMP	Acrónimo de las tecnologías de código abierto utilizadas para crear aplicaciones web de forma económica (Linux, Apache, MySQL y PHP). MySQL es un sistema de gestión de bases de datos de código abierto; PHP es un popular lenguaje de "scripting" del lado del servidor para desarrollar aplicaciones web; Apache es el software de servidor web más popular. El equivalente para el desarrollo en Windows es WAMP (Windows, Apache, MySQL y PHP)
SaaS: Software as a Service (Software como servicio)	El software se ejecuta en servidores en otro lugar de Internet. Cuando ese servidor se actualiza, todos los clientes del mundo ven las nuevas capacidades, sin necesidad de instalación local (se accede al servicio a través de un navegador).
PaaS: Platform as a Service (Plataforma como servicio)	Proporciona una plataforma informática para desarrollar y ejecutar aplicaciones como un servicio a través de la web, en lugar de instalar las herramientas en el ordenador.
Cloud computing	Puede utilizar software y datos almacenados en la "nube" (a los que se accede en ordenadores o servidores) remotos a través de Internet y que están disponibles bajo demanda) en lugar de tenerlos almacenados localmente en su ordenador; permite aumentar / reducir recursos informáticos para satisfacer las necesidades en cualquier momento; ahorra dinero al trasladar al proveedor de servicios la carga de gestionar estas aplicaciones.
SDK: Software Development Kit (kits de desarrollo de software)	Incluyen herramientas y documentación que los desarrolladores utilizan para programar aplicaciones.

Cuando se desarrollan grandes productos de software, se ponen a disposición de las comunidades de usuarios como una serie de versiones, cada una más completa y pulida que la anterior.

<b>Versión</b>	<b>Descripción</b>
Alpha	Primera versión de un producto de software que todavía está en desarrollo activo. Tiene errores, es incompleta e inestable y se libera a un número relativamente pequeño de desarrolladores para probar nuevas características y obtener comentarios tempranos.
Beta	Se lanzan a un mayor número de desarrolladores en una fase posterior del proceso de desarrollo, una vez que se han corregido la mayoría de los errores importantes y se han completado casi por completo las nuevas funciones. Es más estable, pero sigue estando sujeto a cambios.
Release candidates (versiones candidatas)	Tienen todas las características, están libres de errores (en su mayoría) y están listas para ser utilizadas por la comunidad, lo que proporciona un entorno de pruebas diverso: el software se utiliza en diferentes sistemas, con distintas limitaciones y para una variedad de propósitos.
Final release	Los errores que aparecen en la versión candidata se corrigen y el producto final se pone a disposición del público. Las empresas de software distribuyen actualizaciones incrementales a través de Internet.
Continuous beta	No tiene números de versión, es alojada en la nube y está en constante evolución para que los usuarios tengan siempre la última versión.

## 1.14 C++11 and C++14: The Latest C++ Versions

C++11 fue publicado por ISO/IEC en 2011. C++11 amplió la biblioteca estándar de C++ y añadió varias funciones y mejoras para aumentar el rendimiento y la seguridad.

C++14, fue publicado por ISO/IEC en 2014. añadió varias características del lenguaje y mejoras en la Biblioteca Estándar de C++, y corrigió errores de C++11.

## 1.15 Bibliotecas Boost C++

Son bibliotecas gratuitas y de código abierto creadas por miembros de la comunidad C++. Están revisadas por pares y son portables a través de muchos compiladores y plataformas. Las expresiones regulares se utilizan para hacer coincidir patrones de caracteres específicos en el texto. Pueden utilizarse para validar datos y asegurarse de que tienen un formato determinado, para sustituir partes de una cadena por otra o para dividir una cadena.

- Poderosa capacidad de programación que C++ absorbió de C: los punteros inteligentes ayudan a evitar algunos errores clave asociados a los punteros tradicionales.

## 1.16 Mantenerse al día con las tecnologías de la información

Publicación	URL
AllThingsD	allthingsd.com
Bloomberg BusinessWeek	<a href="http://www.businessweek.com">www.businessweek.com</a>
CNET	news.cnet.com
Communications of the ACM	cacm.acm.org
Computerworld	<a href="http://www.computerworld.com">www.computerworld.com</a>
Engadget	<a href="http://www.engadget.com">www.engadget.com</a>
eWeek	<a href="http://www.eweek.com">www.eweek.com</a>
Fast Company	<a href="http://www.fastcompany.com">www.fastcompany.com</a>
Fortune	fortune.com
GigaOM	gigaom.com
Hacker News	news.ycombinator.com
IEEE Computer Magazine	<a href="http://www.computer.org/portal/web/computingnow/computer">www.computer.org/portal/web/computingnow/computer</a>
InfoWorld	<a href="http://www.infoworld.com">www.infoworld.com</a>
Mashable	mashable.com
PCWorld	<a href="http://www.pcworld.com">www.pcworld.com</a>
SD Times	<a href="http://www.sdtimes.com">www.sdtimes.com</a>
Slashdot	slashdot.org
Stack Overflow	stackoverflow.com
Technology Review	technologyreview.com
Techcrunch	techcrunch.com
The Next Web	thenextweb.com
The Verge	<a href="http://www.theverge.com">www.theverge.com</a>
Wired	<a href="http://www.wired.com">www.wired.com</a>