



## INFORMATICA APLICADA A PROCESOS

Curso: 5° "A" – 5° "C"

Prof. MIRA, Jorge L.

NAVARRO, Lucio E.

En este espacio el estudiante desarrolla capacidades para realizar procesos básicos de operación en distintas actividades pertinentes al perfil y sirve para aportar los conocimientos mínimos que faciliten el desarrollo de otros espacios posteriores en los que los conocimientos informáticos son de fundamento para aproximarse a modelos de procesamientos electrónicos aplicados a esos y a sistemas de control en lenguajes de alto nivel. Actuando individualmente o en equipo en el análisis de los procesos productivos, análisis sistémicos de las operaciones, manejar los sistemas de procesamiento de datos avanzados para la realización de representaciones, informes, estadísticas y gráficos.

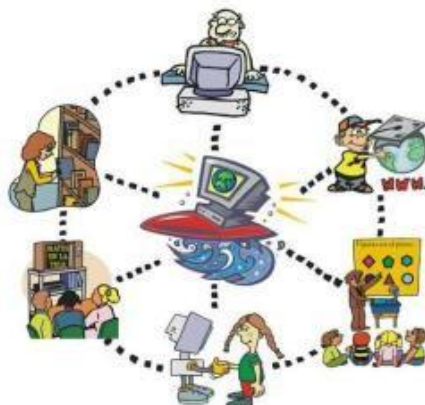
### OBJETIVOS

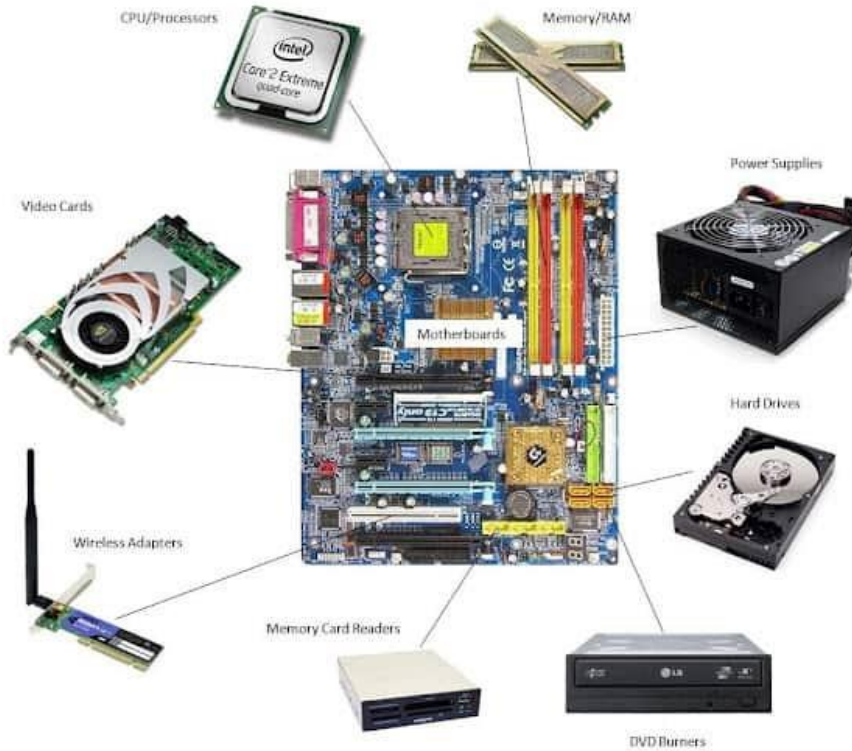
- Identificar los componentes de la computadora.
- Adquirir habilidades a través de la resolución de problemas para la conversión de unidades de almacenamiento de la información.
- Conocer el funcionamiento de un microprocesador.
- Evaluar las características más convenientes de una computadora a la hora de adquirir una computadora.
- Conocer ventajas y desventajas de distintos sistemas operativos.
- Conocer los diferentes virus que afectan el sistema operativo y como se utilizan diferentes programas de antivirus.



### EJE N° 1: INTRODUCCIÓN A INFORMÁTICA

Generalidades de Hardware y Software. Clasificación. Dato e información. Unidades de almacenamiento de la información. Conversión de unidades. Procesamiento de datos. Arquitectura de un ordenador, componentes físicos y lógicos. Identificación de los mismos. Funciones de componentes básicos: placa madre, puertos de comunicación, chipset. Microprocesadores: tipos, características y funcionamiento. Memorias: tipos y características. Periféricos de entrada, de salida, de comunicación y de almacenamiento.





En las siguientes páginas desarrollaremos lineamientos básicos para comprender y conocer **herramientas** que cada día son más **frecuentes y útiles** en nuestra vida cotidiana y laboral.

Sabemos que la posibilidad de estar en contacto con estas herramientas cada vez es más cercana. Seguramente alguno de ustedes conoce mucho, y en profundidad distintas herramientas o no. Sí, estamos convencidos que cada día en los distintos hogares o espacios laborales necesitamos de esta ayuda que nos simplifica las tareas.

Permanentemente nos encontramos frente a tecnologías informáticas y, cada día, están incorporadas a nuestras rutinas facilitando y resolviendo nuestras necesidades diarias. Las encontramos presentes en casi todos los elementos electrónicos y a veces escondidas como por ejemplo: en ascensores, videograbadoras, teléfonos inteligentes, tablets, cajeros automáticos en bancos, etc.



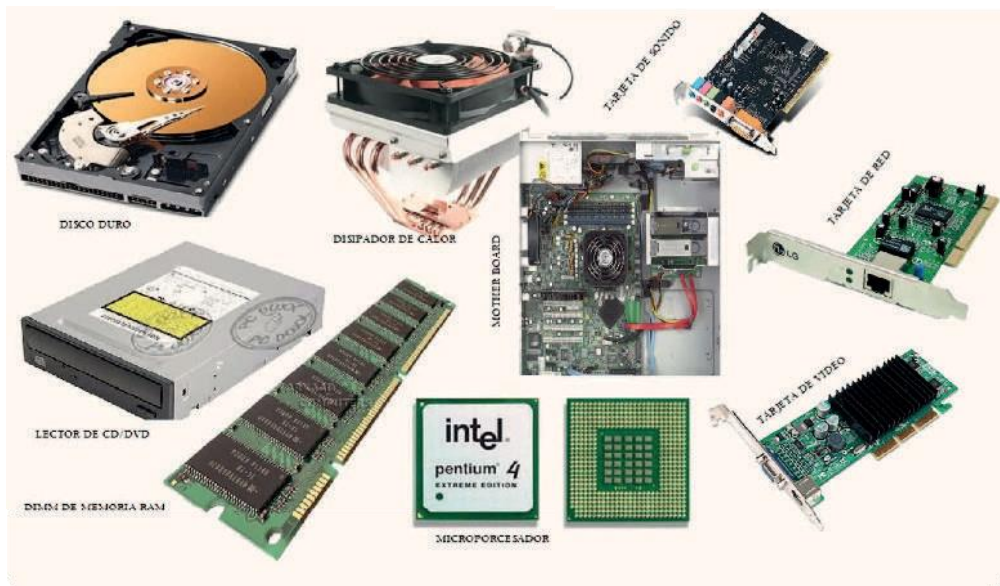
En la actualidad, operamos con tecnología informática permanentemente, lo que nos permite desenvolvemos con mayor participación en la sociedad a través de la comunicación y la información.

Al principio las computadoras se usaban para ayudar y facilitar las tareas que ya se hacían por otros medios, pero una vez introducidas en la sociedad, las computadoras han abierto nuevas posibilidades que las transformaron en indispensables; y dada su facilidad y comodidad, se han expandido a casi todas las actividades, tanto laborales como de estudio, de entretenimiento o de comunicación.

## ¿Qué es el Hardware?

Es un sistema compuesto por cinco elementos:

- **UCP:** (Unidad Central de Procesamiento) Esta unidad se encuentra construida en un circuito integrado (chip), llamado **MICROPROCESADOR**. Su función es interpretar y ejecutar las instrucciones de los programas, efectuar operaciones aritméticas y lógicas con los datos y comunicarse con las demás partes del sistema. Es el conjunto de dispositivos físicos, conectados entre sí, que componen el ordenador (unidad central, monitor, teclado, ratón, impresora, etc...).



- **Dispositivos periféricos de entrada:** Son el teclado, el mouse y el escáner, entre otros. Su función es permitir el ingreso de datos (números o letras).

- **Dispositivos periféricos de salida:** Entre los más comunes podemos mencionar el monitor y la impresora. Su función más importante es mostrarnos los datos ingresados y la información procesada.

- **Unidades de almacenamiento:** Están representados por la memoria principal y las memorias secundarias en las que se guardan los datos y los programas.

- **Buses:** Se expresan en la red de cables que enlaza todos los elementos internos de la computadora y la conecta con el mundo exterior.

- **Generalidades de Hardware y Software (clasificación).**

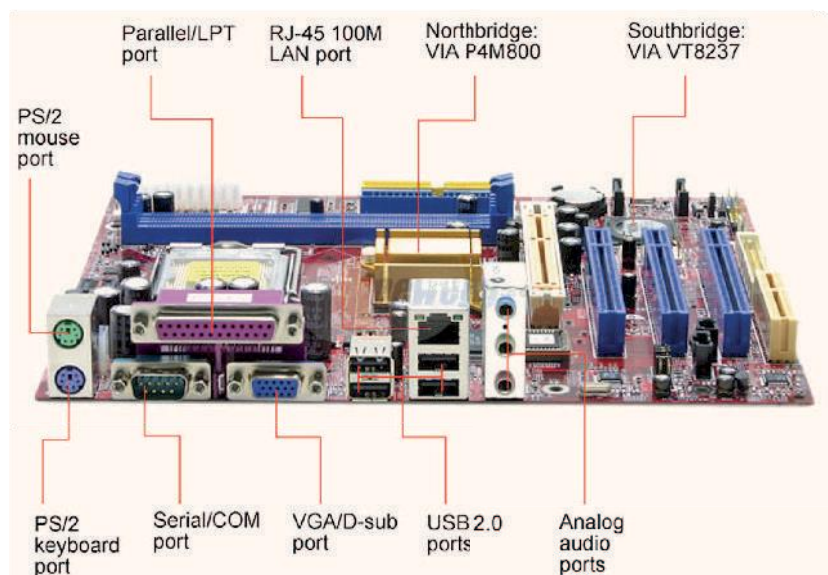
- Dato e información.
- Procesamiento de datos.

- **Arquitectura de un ordenador, componentes físicos y lógicos, identificación de los mismos.**

- Funciones de componentes básicos: placa base, chipset, buses, controladores, puertos, etc. Características, tecnologías, concepto.
- Microprocesadores: tipos, características y funcionamiento.
- Memorias: tipos y características.
- Unidades de almacenamiento externo. Tipos de Discos duros, velocidades, rendimiento etc. Tabla de particiones, tipos, formatos y sus características *MBR y GPT*. Pendrive (capacidades, diferencias y características de formatos) Unidades de almacenamiento (Hard Disk, unidad CD/DVD, Bluray, etc.)



- Dispositivos externos. Tarjetas: gráficas, sonido, controladoras específicas, etc. Sistemas y dispositivos Plug & Play, BIOS. Periféricos de entrada, de salida y, de entrada y salida.





- **Software**
  - Sistema operativo Windows



## **INTRODUCCIÓN**

Sin el software, una computadora es en esencia una masa metálica sin utilidad. Con el software, una computadora puede almacenar, procesar y recuperar información, encontrar errores de ortografía en manuscritos, tener aventuras e intervenir en muchas otras valiosas actividades.

El software para computadoras puede clasificarse en general en tres clases: los programas de sistema, que controlan la operación de la computadora en sí, los programas de aplicación, los cuales resuelven problemas para sus usuarios, y los lenguajes de programación o programas de desarrollo.

El programa fundamental de todos los programas de sistema es el sistema operativo, que controla todos los recursos de la computadora y proporciona la base sobre la cual pueden escribirse los programas de aplicación.

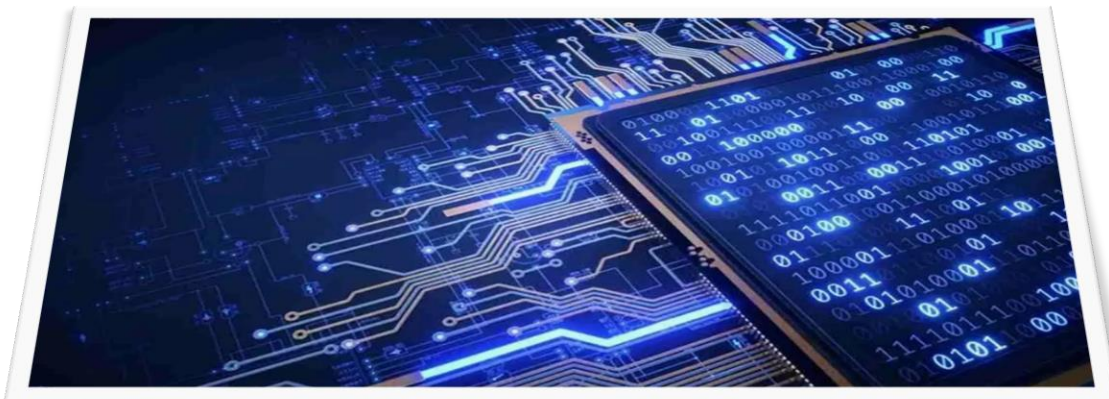
Un sistema de computación moderno consta de uno o más procesadores, cierta memoria principal (a menudo conocida como "memoria central"), relojes, terminales, discos, interfaces en una red y otros dispositivos de entrada/salida.

En fin, un sistema complejo. La escritura de programas que tengan un registro de todos estos componentes y los utilice en forma correcta, ya no digamos en forma óptima, es una labor en extremo difícil. Si cada programador tuviera que preocuparse por la forma de funcionamiento de las unidades de disco y con las docenas de cosas que podrían ir mal al leer un bloque de un disco, es poco probable que pudieran escribirse muchos programas.

Hace muchos años, quedó claro que debía determinarse una forma de proteger a los programadores de la complejidad del hardware. La forma en que esto ha evolucionado de manera gradual es colocando un nivel de software por arriba del simple hardware, con el fin de controlar todas las partes del sistema y presentar al usuario una interfaz o máquina virtual que facilite la comprensión del programa.

A finales de los 40's el uso de computadoras estaba restringido a aquellas empresas o instituciones que podían pagar su alto precio, y no existían los sistemas operativos.

En su lugar, el programador debía tener un conocimiento y contacto profundo con el hardware, y en el infortunado caso de que su programa fallara, debía examinar los valores de los registros y paneles de luces indicadoras del estado de la computadora para determinar la causa del fallo y poder corregir su programa, además de enfrentarse nuevamente a los procedimientos de apartar tiempo del sistema y poner a punto los compiladores, ligadores, etc.; para volver a correr su programa, es decir, enfrentaba el problema del procesamiento serial (serial processing)



## ¿Qué es un Sistema Operativo?

La mayoría de los usuarios de computadora tienen cierta experiencia con un sistema operativo, pero es difícil poder precisar la definición de éste. Parte del problema es que los sistemas operativos llevan a cabo dos funciones que, en esencia, no tienen relación entre sí, por lo que, según la persona que hable, usted podría oír más información acerca de una función u otra.



### AMBIENTE WINDOWS

Los sistemas operativos de interfaces gráficas vinieron a facilitar el uso de las computadoras, poniéndolas al alcance de todo mundo. Los usuarios ya no tienen que memorizar y utilizar una larga serie de comandos para ejecutar una función específica (como se hace con MS-DOS). Ahora, basta con que identifiquen el icono que representa la función deseada, opriman y suelten rápidamente un botón del ratón para que puedan ejecutarla.

Las características más importantes de los sistemas operativos de entorno gráfico son:

**Escritorio.** Como todos los sistemas operativos multitareas, permite mantener abiertas varias aplicaciones. El escritorio cuenta con gran cantidad de herramientas, útiles para todas las actividades que se puedan realizar en la computadora.

- Multitareas
- Menús desplegables
- Cuadros de diálogo
- Iconos.
- Uso del ratón (mouse)
- Con el ratón se realizan tareas como:
- Abrir programas o documentos
- Ejecuta comandos
- Se modifica el tamaño de las ventanas.
- Menú contextual.
- Arrastrar iconos y ventanas

- *Software libre*



## ¿Qué es el Software Libre?

«Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que **los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software**. Es decir, el «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre». En inglés, a veces en lugar de «free software» decimos «libre software», empleando ese adjetivo francés o español, derivado de «libertad», para mostrar que no queremos decir que el software es gratuito.

Puede haber pagado dinero para obtener copias de un programa libre, o puede haber obtenido copias sin costo. Pero con independencia de cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de [vender copias](#).



Promovemos estas libertades porque todos merecen tenerlas. Con estas libertades, los usuarios (tanto individualmente como en forma colectiva) controlan el programa y lo que este hace. Cuando los usuarios no controlan el programa, decimos que dicho programa «no es libre», o que es «privativo». Un programa que no es libre controla a los

usuarios, y el programador controla el programa, con lo cual el programa resulta ser [un instrumento de poder injusto](#).

El «Open source» (Código abierto) es algo distinto: su filosofía es diferente y está basada en otros valores. Su definición práctica también es diferente, pero de hecho casi todos los programas de código abierto son libres. Explicamos la diferencia [aquí](#).

## **Definición de software libre**

La definición de software libre estipula los criterios que se tienen que cumplir para que un programa sea considerado libre. De vez en cuando modificamos esta definición para clarificarla o para resolver problemas sobre cuestiones delicadas. Más abajo en esta página, en la sección [Historial](#), se puede consultar la lista de modificaciones que afectan la definición de software libre.

## **Las cuatro libertades esenciales**

Un programa es software libre si los usuarios tienen las cuatro libertades esenciales:

- La libertad de ejecutar el programa como se desee, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que se desee (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para ayudar a otros (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (libertad 3). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no son

libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, nosotros los consideramos contrarios a la ética a todos por igual.

En cualquier circunstancia, estas libertades deben aplicarse a todo código que pensemos utilizar hacer que otros utilicen. Tomemos por ejemplo un programa A que automáticamente ejecuta un programa B para que realice alguna tarea. Si se tiene la intención de distribuir A tal cual, esto implica que los usuarios necesitarán B, de modo que es necesario considerar si tanto A como B son libres. No obstante, si se piensa modificar A para que no haga uso de B, solo A debe ser libre; B no es relevante en este caso.

## **El software libre *puede* ser comercial**

«Software libre» no significa «no comercial». Por el contrario, un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. Esto es de fundamental importancia, sin ello el software libre no podría alcanzar sus objetivos.

Queremos invitar a todo el mundo a utilizar el sistema GNU, incluidas las empresas y sus trabajadores. Eso requiere que se permita el uso comercial. Esperamos que programas sustitutivos libres reemplacen a programas privativos similares, pero no podrán hacerlo si a las empresas se les prohíbe utilizarlos. Queremos que los productos comerciales que contengan software incluyan el sistema GNU, y eso representaría distribución comercial por un precio. Ya no es inusual el desarrollo comercial de software libre, y ese software libre comercial es muy importante. El respaldo profesional, remunerado, al software libre satisface una importante necesidad.



Así pues, excluir el uso comercial, la programación comercial y la distribución comercial sería un lastre para la comunidad del software libre y obstaculizaría su camino al éxito. Hemos de concluir que un programa con una licencia que contenga tales restricciones no puede considerarse software libre.

Un programa libre debe ofrecer las cuatro libertades a todo usuario potencial que obtenga una copia del software y haya respetado las condiciones de la licencia libre que cubre el software en cualquier distribución anterior del mismo. Privar de alguna de esas libertades a ciertos usuarios, o exigirles un pago en dinero o en especie para ejercerlos, equivale a no garantizarles las libertades en cuestión, lo que hace que el programa no sea libre.



## Delimitación de la frontera entre software libre y software privativo

En el resto del artículo explicamos con más precisión qué alcance debe tener cada una de estas libertades, en diversos asuntos, para que un programa sea libre.

### La libertad de ejecutar el programa como se desee

La libertad de ejecutar el programa significa que cualquier tipo de persona u organización es libre de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y finalidad, sin que exista obligación alguna de comunicarlo al programador ni a ninguna otra entidad específica. En esta libertad, lo que importa es el propósito del *usuario*, no el del *programador*. Usted como usuario es libre de ejecutar el programa para alcanzar sus propósitos, y si lo distribuye a otra persona, también esa persona será libre de ejecutarlo para lo que necesite; usted no tiene el derecho de imponerle sus propios objetivos a la otra persona.

La libertad de ejecutar el programa como se desee significa que al usuario no se le prohíbe o no se le impide ejecutarlo. Esto no tiene nada que ver con el tipo de funcionalidades que el programa posea, ni con su capacidad técnica de funcionar en un entorno dado, ni con el hecho de que el programa sea o no sea útil con relación a una operación computacional determinada.



Por ejemplo, si el código rechaza arbitrariamente ciertas órdenes significativas (o incluso falla sin motivo), lo que resta utilidad al programa o incluso lo vuelve completamente inútil, pero no se niega a los usuarios la libertad de ejecutar el programa, esto no entra en conflicto con la libertad 0. Si el programa es libre, los usuarios pueden restituirle la utilidad, ya que las libertades 1 y 3 permiten a usuarios y colectivos hacer y distribuir versiones modificadas en las que se haya eliminado el código problemático.

«Como se desee» incluye la opción «nada en absoluto», si es eso lo que desea. De modo que no hay necesidad de añadir una «libertad de no ejecutar el programa».

### La libertad de estudiar el código fuente y modificarlo

Para que las libertades 1 y 3 (realizar cambios y publicar las versiones modificadas) tengan sentido, usted debe tener acceso al código fuente del programa. Por consiguiente, el acceso al código fuente es una condición necesaria para el software libre. El «código fuente» ofuscado no es código fuente real y no cuenta como código fuente.

La libertad 1 incluye la libertad de usar su versión modificada en lugar de la original. Si el programa se entrega unido a un producto diseñado para ejecutar versiones modificadas por

terceros, pero rechaza ejecutar las suyas —práctica conocida como «tivoización» o «bloqueo», o (según la terminología perversa de quienes lo practican) «arranque seguro»—, la libertad 1 se convierte en una vana simulación más que una realidad práctica. Estos binarios no son software libre, aun cuando se hayan compilado a partir de un código fuente libre.



Una manera importante de modificar el programa es agregándole subrutinas y módulos libres ya disponibles. Si la licencia del programa especifica que no se pueden añadir módulos que ya existen y que están bajo una licencia apropiada, por ejemplo si requiere que usted sea el titular del copyright del código que desea añadir, entonces se trata de una licencia demasiado restrictiva como para considerarla libre.

Si una modificación constituye o no una mejora, es un asunto subjetivo. Si su derecho a modificar un programa se limita, básicamente, a modificaciones que alguna otra persona considera una mejora, el programa no es libre.

Un caso especial de la libertad 1 es la libertad de borrar código del programa para que su ejecución no tenga ningún efecto, o para que invoque algún otro programa. Así pues, la libertad 1 incluye la «libertad de eliminar el programa».

## **La libertad de redistribuir copias si así lo desea: requisitos básicos**

La libertad para distribuir (libertades 2 y 3) significa que usted tiene la libertad para redistribuir copias con o sin modificaciones, ya sea gratuitamente o cobrando una tarifa por la distribución, a [cualquiera en cualquier parte](#). Ser libre de hacer esto significa, entre otras cosas, que no tiene que pedir ni pagar ningún permiso para hacerlo.

También debe tener la libertad de hacer modificaciones y usarlas en privado para su propio trabajo o pasatiempo, sin siquiera mencionar que existen. Si publica sus cambios, no debe estar obligado a notificarlo a nadie en particular, ni de ninguna manera en particular.

La libertad 3 incluye la libertad de publicar sus versiones modificadas como software libre. Una licencia libre también puede autorizar otras formas de publicación; en otras palabras, no tiene que ser una licencia con [copyleft](#). No obstante, una licencia que requiera que las versiones modificadas no sean libres, no se puede considerar libre.

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente, tanto para las versiones modificadas como para las que no lo estén. (Distribuir programas en forma de ejecutables es necesario para que los sistemas operativos libres se puedan instalar fácilmente). Resulta aceptable si no existe un modo de producir un formato binario o ejecutable para un programa específico, dado que algunos lenguajes no incorporan esa característica, pero debe tener la libertad de redistribuir dichos formatos si encontrara o programara una forma de hacerlo.

## Copyleft

Ciertos tipos de reglas sobre la manera de distribuir software libre son aceptables, cuando no entran en conflicto con las libertades principales. Por ejemplo, el [copyleft](#), definido muy sucintamente, es la regla en base a la cual, cuando redistribuye el programa, no se puede agregar restricciones para denegar a los demás las libertades principales. Esta regla no entra en conflicto con las libertades principales, más bien las protege.

En el proyecto GNU usamos el copyleft para proteger legalmente las cuatro libertades para todos. Creemos que existen razones importantes por las que es mejor usar el copyleft. De todos modos, el software libre sin copyleft también es ético. Véase en [categorías del software libre](#) una descripción de la relación que existe entre el «software libre», «software con copyleft» y otros tipos de software.

- Virus y otros archivos dañinos
- Unidad de medida byte
- Definición y tabla de equivalencia con mayores dígitos.
- Tamaños de archivos de diferentes tipos (Documentos Word, Excel, etc. Tema de audio en formato mp3, video musical en formato mp4, video recital completo en formato mp4, renders de video o arquitectura)
- Situaciones problemáticas en bytes convirtiendo unidades en regla de 3 para resolver cuantos y/o cuales archivos pueden entrar en un pendrive de poca capacidad (para lograr conocer la diferencia entre KB, MB, GB y TB)

## Unidad de medida Byte

**Unidad de información digital** equivalente a cuatro bits (*binary digit*, dígito binario) originalmente y posteriormente como estándar se adoptó que 1 byte equivale a ocho bits.

-El byte se utiliza generalmente en las áreas de informática y telecomunicaciones, en esta última se denomina comúnmente **octeto**, que proviene del francés *octet*, derivado del latín *octo* y del griego *okto*, que significa ocho, diferenciando así el byte de 8 bits de otros bytes con diferente equivalencia de bits.

-El término byte se utiliza a menudo para especificar cantidad, por ejemplo, la cantidad de memoria de un determinado dispositivo o la capacidad de almacenamiento.  
Ejemplo: 16 GB (gigabyte).

-Cada byte representa un solo carácter de texto en un ordenador. El byte representa letras, símbolos, números, signos de puntuación, caracteres especiales, etc. y codifica diferentes informaciones en un mismo equipo, dependiendo de la cantidad. Por ejemplo, 1 B equivale a un carácter o letra, 10 B equivale a una o dos palabras, 100 B equivale a una o dos frases, 1 kB (1024 B) equivale a una historia muy corta, 10 kB equivale a una página de enciclopedia, tal vez con un dibujo simple, 100 kB equivale a una fotografía de resolución mediana, 1 MB (1024 kB) equivale a una novela, 10 MB equivale a dos copias de la obras completas de Shakespeare (poeta y actor inglés), 100 MB equivale a un estante de 1 metro de libros, 1 GB (1024 MB) equivale a una furgoneta llena de páginas de texto, 1 TB (1024 GB) equivale a 50000 árboles, 10 TB equivale a la colección impresa de la biblioteca del congreso de los EEUU.



**Para expresar cantidades de datos, se utilizan las siguientes medidas:**

- 1 byte = 8 bits
  - 1 kilobyte (kB o kbyte) = 1024 bytes
  - 1 megabyte (MB o Mbytes) = 1024 kilobytes
  - 1 gigabyte (GB o Gbytes) = 1024 megabytes
  - 1 terabyte (TB o Tbytes) = 1024 gigabytes
  - 1 petabyte (PB o Pbytes) = 1024 terabytes
  - 1 exabyte (EB o Ebytes) = 1024 petabytes
  - 1 zettabyte (ZB o Zbytes) = 1024 exabytes
  - 1 yottabyte (YB o Ybytes) = 1024 zettabytes
- 
- Documento de Word simple = 5 KB a 100 MB o mas
  - Tema musical en formato mp3 = 2,5 MB a 15 MB o mas
  - Video musical (1 tema) en formato mp4 = 50 MB o mas
  - Video musical (Recital completo) en formato mp4 = 1,5 GB o mas
  - Renders de video o arquitectura = muchos GB

