

I. Introducción

Para comenzar un estudio profundo de la figura del *software*, debemos tener presente la importancia de este producto, que con el paso del tiempo va en aumento, debido a la extraordinaria utilidad del mismo, su influencia en múltiples ámbitos, y como consecuencia de ello, el incremento de su valor económico.

Hasta las décadas de 1960 y 1970 el *software* era producido y distribuido por los mismos fabricantes de *hardware* (equipos físicos) quienes se encargaban de incluirlos en sus productos como accesorios de los mismos (por ej. IBM). En consecuencia y ante el incipiente desarrollo del *software*, estos no alcanzaban un valor superior al 30% del total del precio de un ordenador. Con el paso del tiempo y el avance de la ingeniería informática, se ha logrado separar al *software* del *hardware*, creándolo en forma independiente, como un bien jurídico en sí mismo, autónomo (en muchos casos, con un valor económico muy superior al *hardware*), susceptible de actos de comercio, de transacciones civiles y por ende merecedor de tutela jurídica (Tantaleán Odar)

Para colocar en números reales y graficar de alguna manera la importancia de la que hablamos respecto de la industria del *software* en el mundo moderno y específicamente en Argentina, según la agencia Telam, a inicios del año 2014, publicó que el año 2013 concluyó con 78.500 empleados, USD 900 millones en exportaciones y USD 3.750 millones facturados, repartidos en más de 4.000 empresas, en su gran mayoría nacionales y con una tasa de crecimiento promedio del 20 % en la última década (Tjor, 2014).

1. Concepto de *Software* y aspectos terminológicos.

Tal como nos enseña el Dr. Fernández Delpech (2014), podríamos definir al Derecho Informático como la nueva disciplina jurídica integrada por el conjunto de principios y normas que regulan los efectos jurídicos nacidos de la interrelación entre el derecho y la informática. Por su parte, la Informática es una ciencia que estudia métodos, procesos y técnicas, con el fin de almacenar, procesar y transmitir informaciones y datos en formato digital. A esos métodos, procesos y técnicas, los podemos llamar Sistemas Informáticos, dentro de los cuales encontramos dos elementos fundamentales que los componen y que son el *hardware* y el *software*.

En efecto, la palabra *hardware* hace alusión a los componentes físicos, materiales o tangibles que integran una computadora u ordenador, como por ejemplo, la unidad central de procesamiento (CPU), la placa madre, el microprocesador, el disco rígido, etc. Por otro lado, tenemos el *software*¹, que designa a todo el equipamiento o soporte lógico (inmaterial e intangible), incluyendo tanto a las aplicaciones como a los datos sobre los cuales ellas operan.

Por su parte, con una visión técnica, Hess Araya siguiendo a Valentín Carrascosa López dice que por aplicación o programa entendemos “Toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas directa o indirectamente en un sistema informático, para realizar una función o tarea o para obtener un resultado determinado, cualquiera que fuere su forma de expresión o fijación” (2007, p. 1).

En nuestra legislación, el artículo 5 de la ley N° 25922 de Promoción de la Industria del *Software* define al mismo diciendo que es “la expresión organizada de un conjunto de órdenes o instrucciones en cualquier lenguaje de alto nivel, de nivel intermedio, de ensamblaje o de máquina, organizadas en estructuras de diversas secuencias y combinaciones, almacenadas en medio magnético, óptico, eléctrico, discos, chips, circuitos o cualquier otro que resulte apropiado o que se desarrolle en el futuro, previsto para que una computadora o cualquier máquina con capacidad de procesamiento de información ejecute una función específica disponiendo o no de datos, directa o indirectamente”.

Gran aporte es el que realiza la Cámara de Apelaciones en lo Civil, Comercial, Minas, de Paz y Tributaria de Mendoza que en el caso de “Microsoft Corporation c/Samaco S.A. p/daños y perjuicios” expresó que “el programa de computación o *software*, es el componente inmaterial, es información que se traduce en una combinación de comandos e instrucciones sin los cuales la computadora o *hardware* es completamente inútil”; es decir, que se trata de la parte no visible del sistema de la que deriva, en esencia, el provecho que podemos obtener de cualquier computadora.

Básicamente, el *software* realiza tres tareas, primero el ingreso de datos (entrada), luego tratamiento de los mismos de acuerdo a las instrucciones del programa respectivo (procesamiento) y finalmente genera los resultados obtenidos (salida) (Hess Araya, 2007).

¹ Entendiendo actualmente al *software* no solo como programas utilizados en computadoras personales (PC), sino también en sistemas operativos y aplicaciones móviles, por ejemplo para teléfonos inteligentes (*smartphones*), televisores inteligentes (*Smart tv*), *tablets*, etc.

Piénsese, por ejemplo, en la preparación de una planilla de pagos salariales a los empleados de una empresa. Tenemos todos los datos correspondientes a los trabajadores, número de horas trabajadas, valor de la hora, descuentos de aportes, etc. (entrada); el sistema arrojará un cálculo del monto a pagar a cada trabajador (procesamiento); y por último producirá un reporte o incluso podrá imprimir los bonos de sueldo correspondientes (salida).

Finalmente, para ir terminando con la conceptualización del *software*, debemos tener en cuenta que el mismo se compone de dos elementos: el Código Fuente y el Código Objeto.

Claro está, que una computadora no es más que un conjunto de dispositivos sin capacidad de análisis o raciocinio propio, por ende todas las tareas que la misma sea capaz de llevar a cabo, desde las más sencillas a las más complejas, dependen de que se les haya suministrado previamente instrucciones precisas. A esto se lo denomina programar la computadora (Hess Araya, 2007).

Las computadoras no comprenden nuestro idioma. El lenguaje humano es infinitamente más complejo y sofisticado que el lenguaje de máquina (que es el único que la computadora puede manejar), ya que está lleno de abstracciones, sobreentendidos y factores culturales, etc. -resultado de millones de años de evolución- que están fuera del alcance de un sistema informático. El lenguaje de máquina, es verdaderamente más limitado y está compuesto solo por dos elementos: unos y ceros, que a su vez representan los estados de “encendido” y “apagado”, que son los únicos que pueden interpretar los componentes internos del *hardware*. En consecuencia para que un ordenador sea capaz de interpretar y ejecutar una orden o instrucción, esta deberá expresarse primero en código binario, es decir, en términos de unos y ceros (por ejemplo “10110011”) (Hess Araya, 2007).

El código fuente, es el que está escrito en un lenguaje de programación (por ej. Java o Visual BASIC de Microsoft) el cual les permite a los programadores escribir sus aplicaciones de una manera más sencilla o legible. Mientras que el código objeto o código binario es el que resulta de la conversión del lenguaje del código fuente al lenguaje de la máquina, que se expresa en alfabeto binario, es decir, se trata de una secuencia de instrucciones que se le presenta a la máquina como unos y ceros antes de que ésta sea capaz de ejecutarlo y por lo tanto es ininteligible para los seres humanos. Ambos elementos componen el *software* y están alcanzados por la protección que brinda nuestra ley de Propiedad Intelectual (Fernández Delpech, 2014).

2. Caracterización y Naturaleza jurídica del *Software*.

Como señalamos anteriormente, hasta la década del '60 y '70, la mayor parte del *software* era producido y distribuido por los propios fabricantes del *hardware* y era considerado como accesorio de aquel (por ej. IBM construía las computadoras y programaba la mayoría de las aplicaciones). Sin embargo, a mediados de la década del '60, como consecuencia de las acciones del Departamento de Justicia de los EE.UU. se forzó a la IBM a separar la distribución de *hardware* y *software*, por considerar, lo contrario, como anticompetitivo y especialmente a partir de la revolución de las computadoras en los años '80, el *software* adquirió entidad y autonomía propias. A partir de ahí comenzaron a surgir innumerables fabricantes de *software*, que hasta la actualidad producen programas diversos, destinados a distintos sistemas informáticos con gran éxito, como por ejemplo Microsoft, Apple y Oracle entre los más destacados (Hess Araya, 2007).

Actualmente el *software*, constituye un bien jurídico en sí mismo y autónomo, susceptible de relaciones civiles y comerciales, aunque todavía algunas clases de *software*, como por ejemplo los sistemas operativos, los controladores o *drivers* continúan siendo producidos por los fabricantes de *hardware*, tal es el caso de la empresa Apple en algunos de sus productos.

En un plano jurídico, y en lo que a su naturaleza concierne, podemos caracterizar entonces al *software* así:

- Es un bien mueble;
- Es inconsumible, ya que no se destruye con el uso;
- Es fungible, porque cada copia es intercambiable y exactamente equivalente a las demás.
- Es un bien indivisible, puesto que no se puede ejecutar un programa parcialmente, ni distribuirlo por partes ejecutables separadamente.²
- Posee carácter patrimonial, porque es pecuniariamente estimable y susceptible de figurar como objeto de transacciones civiles y comerciales (Hess Araya, 2007, p. 5).

El *software* es un bien inmaterial, intangible y volátil, resultado de la ejecución por parte del computador de la secuencia de instrucciones que lo componen. Siempre cabe distinguir al bien incorpóreo o ideal, del soporte físico sobre el que esté fijado. Así pues, el derecho de propiedad

² El hecho de que sea posible actualizar o modificar algunos elementos separadamente, no contradice el hecho de que todos ellos deban ser cargados y ejecutados unitariamente por el computador.

que se ejerza sobre el soporte carece de nexo jurídico inmediato con la cuestión de la titularidad del *software* en sí (Hess Araya, 2007).

Finalmente, para concluir con este apartado, vale aclarar, que las obras tecnológicas de *software* son creaciones del intelecto humano, consideradas como derechos de propiedad intelectual, equiparándolas a las producciones científicas, literarias y artísticas, y como ya veremos más adelante, protegidas en nuestro país, por el sistema de derechos de autor.

3. Clasificaciones del *Software*.

Como en toda disciplina científica, las clasificaciones siempre tienen una finalidad meramente didáctica, lo que nos permite estudiar un determinado tema en profundidad, separando y relacionando cada una de sus partes para comprender el todo de manera más sencilla y pormenorizada.

En lo que al *software* se refiere, si bien en algún punto pueden llegar a confundirse las distintas clasificaciones con las modalidades de licenciamiento del mismo, lo que será analizado en un capítulo posterior, creemos que es importante para el desarrollo y estudio del tema, tratar ambos por separado, por lo tanto intentaremos evitar dichas confusiones. Para ello, seguimos principalmente a Hess Araya (2007), quien ha propuesto dos mecanismos de clasificación, uno de índole técnico y otro jurídico, los cuales no se excluyen entre sí y son provechosos para nuestro objeto.

Vale aclarar, que dentro del llamado *software*, coexisten una multiplicidad de variantes que en ocasión resultan prácticamente contrapuestas, tanto en su filosofía de concepción como en el modo que se licencian, modifican e interactúan con otros programas, usuarios y desarrolladores. Por ello, dicha clasificación no es taxativa, ya que por lo vertiginoso del rubro nos impide generar encasillamientos más que temporales (Tjor, 2014).

3.1. Desde la perspectiva técnica

El *software* puede ser técnicamente agrupado en dos categorías generales: el *software* de sistema y el *software* de aplicación.

- Software de sistema: alude a aquellos programas y datos requeridos para el funcionamiento básico del equipo informático. En esencia, posibilitan la correcta operación del *hardware*, al establecer y regular la comunicación entre la unidad central de proceso (CPU) y todos los demás componentes físicos. El más importante de estos programas es el llamado sistema operativo, que la computadora debe cargar e iniciar desde el momento del encendido. Existen diversos sistemas operativos, cada uno diseñado específicamente para la clase de *hardware* sobre la que deban operar. Esto incluye productos comerciales (como el Windows de Microsoft y MacOS de Apple) como los no comerciales (ej. Linux).
- Software de aplicación: por su parte, reúne a todos aquellos programas de computación con los que trabajamos para obtener determinados resultados, como por ejemplo los procesadores de texto, planillas de cálculo, video juegos, navegadores de internet, reproductores de audio y video, etc. Estas aplicaciones corren encima del sistema operativo, solicitando y obteniendo de él diversos servicios relativos al *hardware*, por ejemplo, para imprimir documentos o guardar archivos en el disco.

3.2 Desde la perspectiva jurídica

En este plano, Hess Araya (2007) nos propone cuatro esquemas de clasificación complementarios del *software*, los cuales vamos a ir completando y modificando según nuestro criterio, de los que podemos extraer diversas consecuencias legales relevantes: a) según el grado de estandarización del *software*; b) según el grado de vinculación del *software* con el *hardware*; c) según la presencia o ausencia de contraprestación y sus modalidades; y d) según la modalidad de licenciamiento.

3.2.1 Según el grado de estandarización: *software* estándar o hecho a la medida.

- Software estándar: incluye innumerables productos fabricados para distribución masiva. Es estándar precisamente porque es igual para todos, sin perjuicio de que –una vez instalado- el usuario pueda personalizar algunas de sus opciones. Es erróneo confundir esta categoría con la de *software* “comercial”, ya que existe gran cantidad de programas estándar que son de distribución gratuita.

- Software hecho a la medida: es aquel producido específicamente contra pedido y para las necesidades concretas de un usuario o usuarios. Por ejemplo un programa de computación realizado especialmente para un comercio, con el cual puedan llevarse a cabo las operaciones de compraventa, de facturación, de presupuestos, contener una base de datos de la clientela y de los proveedores, consultar *stocks* de mercadería, precios, etc.

3.2.2 Según el grado de vinculación del *software* con el *hardware*: *firmware* o *software* autónomo.

- Firmware³: esencialmente hace alusión a la clase de *software* que se almacena en la memoria ROM⁴ de una computadora. El *firmware* es el responsable de las acciones iniciales que toma un sistema informático desde el instante del encendido, que esencialmente consisten en realizar un auto diagnóstico y luego buscar, cargar e iniciar la ejecución del sistema operativo. El *firmware* viene siempre pre-instalado en el equipo informático, por ende, es siempre accesorio del *hardware*.
- Software autónomo: por exclusión, es todo aquel que, por no venir integrado en el *hardware*, es necesario obtener separadamente e instalar, previo a su empleo. Por lo tanto, es un bien jurídico individualizado (mueble e ideal).

3.2.3 Según la presencia o ausencia de contraprestación y sus modalidades: *software* comercial, *freeware*, *shareware*, *adware*.

- Software comercial: es el que llega a los usuarios por canales establecidos de mercadeo (tiendas, catálogos, descargas en línea, etcétera) y es distribuido normalmente con ánimo de lucro. En otras palabras, es objeto de una transacción mercantil determinada, en virtud de la cual el usuario lo adquiere (por ej. vía licencia de uso) previo pago del precio.
- Freeware: se refiere a un modo de adquisición de *software*, caracterizado por la ausencia de contraprestación; es decir, por su gratuidad. El usuario es normalmente libre de

³ Resulta difícil traducir este término al castellano, es conocido como microcódigo.

⁴ ROM significa *read only memory* (memoria de lectura exclusiva o no regrabable). Se diferencia de la RAM o *random access memory* (memoria de acceso aleatorio), que es lo que normalmente conocemos como la memoria principal de una computadora.

instalarlo y usarlo en una o más computadoras, así como de distribuir copias a otras personas. El titular, renuncia a sus derechos de remuneración y reproducción, pero las restantes atribuciones patrimoniales y morales permanecen intactas, ya que por ejemplo el titular del derecho de autor puede establecer acciones legales que impidan la distribución de versiones modificadas del programa, o para reclamar contra el cobro indebido de sumas por su distribución a terceros.

Por su parte Federico Tjor (2014) hace algunas aclaraciones que vale la pena mencionar, explicándonos al respecto, que en este caso el desarrollador nada espera en retribución por su obra, aunque prohíbe su modificación y en consecuencia no libera el código fuente. Sobre este último, resulta fundamental comprender que no se equipara al *software* libre o *free software*, por lo tanto no deben confundirse uno con otro, ya que como veremos en párrafos posteriores, éste último no se refiere a la onerosidad o no del bien, sino a la libertad de utilización, copia, modificación, y distribución entre otras posibilidades.

¿Por qué querría alguien hacer un *software* gratuito? En realidad, hay varias y muy buenas razones:

- ✓ Algunas empresas fabricantes de *software* distribuyen gratuitamente versiones limitadas de sus productos comerciales, con el fin de atraer posibles interesados hacia estos últimos;
- ✓ Otras lo hacen con el fin de abrir un nicho de mercado que les permita crearse una reputación tecnológica y comercial, y así colocar sus otros productos y servicios.
- ✓ Otros prefieren regalar un producto, en la esperanza de obtener dinero mediante la prestación de servicios complementarios (soporte técnico, consultorías, etcétera) o a través de la venta de futuras actualizaciones (por ejemplo, en el caso del *software* antivirus) (Hess Araya, 2007, p. 9 y 10).
- Shareware⁵: es una modalidad especial, que se caracteriza por estar sujeta al examen temporal del *software* hasta el acaecimiento de una condición preestablecida, en cuyo momento el usuario debe decidir si desea conservarlo o no. En caso afirmativo, deberá satisfacer la contraprestación estipulada en el contrato por el licenciante.

⁵ Del inglés *Share*, compartir

Generalmente, no se provee del código fuente, por lo que el usuario no puede modificar el *software*.

La ventaja de esta alternativa es la libre circulación y prueba del programa, en ocasiones de manera limitada o con solo algunas opciones básicas habilitadas como herramienta de difusión y distribución para pequeños y medianos desarrolladores, quienes generalmente carecen de una marca reconocida y de una estructura publicitaria masiva (Tjor, 2014).

El *shareware*, es particularmente apto para productos que –por su novedad, escasa publicidad, etcétera- el fabricante estima que los usuarios posiblemente no adquirirían sin antes convencerse de sus funciones y utilidades. De esta manera han llegado a popularizarse desde juegos hasta programas de productividad. Un ejemplo de ello es el muy popular programa llamado WinZip para comprimir y descomprimir archivos.

De los conceptos enunciados anteriormente, podemos profundizar los siguientes elementos de interés con relación al *shareware*:

a) Examen o prueba del *software*: algunos programas tendrán funcionalidades limitadas total o parcialmente; otros no presentarán restricción alguna. En ciertos casos, el *software* marcará de alguna manera los archivos de datos, imágenes, etcétera, como por ejemplo mediante una leyenda que diga “Esta imagen ha sido generada por una copia no registrada de...” o alguna marca de agua sobre un video.

b) Condición preestablecida: lo más usual es que la condición consista en el vencimiento de un plazo determinado, que generalmente se suele fijar en 30 o 60 días. Sin embargo también existen otras modalidades, como por ejemplo estableciendo un número máximo de utilidades del *software*.

c) Contraprestación estipulada: una vez cumplida la condición, el usuario está obligado a desistir del empleo del producto y a desinstalarlo de su computadora, a menos que satisfaga dicha contraprestación predeterminada. En este último caso, el registro suele ser correspondido con la asignación al usuario de alguna contraseña o código que le permitirá acreditar que es titular legítimo del programa, así como activar las funcionalidades previamente limitadas del paquete, si las hubiere.

En consecuencia, de darse el caso contrario, es decir, que el usuario continúe usando el *software* sin satisfacer los términos de la contraprestación exigida, hará incurrir al infractor

en las responsabilidades civiles y penales propias de la violación de los derechos de autor (esto es, la llamada piratería del *software*).

- Adware: es un sistema más actual, que toma su nombre del término inglés “*advertising*” que significa publicidad. Este tipo de programas es de utilización y distribución gratuita respecto del usuario, y por su parte el desarrollador es retribuido a través de la exhibición automática y obligatoria de publicidad (carteles, videos, audios, animaciones, juegos, etcétera) ya sea durante la instalación, o bien la ejecución del *software*, obteniendo una ganancia relativa a la cantidad de vistas que cuente la publicidad y simultáneamente un monto porcentualmente mayor sobre aquellos que efectivamente se interesen en lo publicitado (medido no en número de exhibiciones del cartel, sino en *clicks* sobre el mismo o contratación de lo ofertado, siempre que resulte derivado de la publicidad inserta) (Tjor, 2014).

3.2.4 Según la modalidad de licenciamiento: *software* de dominio público y de dominio privado (*software* propietario y *software* libre).

Si bien este es un tema que analizaremos más profundamente al desarrollar los contratos relacionados al *software*, especialmente el de licencia de uso y reproducción, haremos aquí una breve referencia de cada uno para terminar de completar las clasificaciones.

El Dr. Fernández Delpech (2011) siguiendo a Martín Carranza Torres, nos habla de una clasificación desde el punto de vista del derecho de propiedad, y es la de *software* de dominio público y de dominio privado.

- Software de dominio público: en este caso el titular de los derechos de autor, renuncia voluntariamente a los derechos patrimoniales sobre la aplicación, y por ende, la licencia conferida al usuario es de carácter irrestricta. Por su parte, los derechos morales, permanecen intactos en cabeza del autor, en virtud de su carácter irrenunciable.

Tal como nos aclara Horacio Fernández Delpech (2011) la titularidad de los derechos recae sobre la sociedad en su conjunto, es decir, que ninguna persona o grupo de personas tiene derechos exclusivos sobre la obra, ya que no pueden apropiarse de la misma, ni pasarla a su dominio privado, así como tampoco monopolizar su utilización, salvo que se trate de

una obra derivada, conforme al artículo 29 del decreto 41.223/34, el cual reglamenta la Ley de Propiedad Intelectual.

- Software de dominio privado: es aquel que ha sido creado por un autor que lo ha registrado a su nombre, o aquel que, ha sido creado por una persona en ejercicio de sus funciones laborales en el ámbito de una empresa y consecuentemente, el *software* le pertenece a la empresa y se encuentra registrado a nombre de ésta (Fernández Delpech, 2011).

Hecha esta clasificación, ahora debemos distinguir dentro del *software* de dominio privado:

1) *Software* propietario

Es el creado por una o varias personas, que conservan la totalidad de los derechos de propiedad sobre la obra, pero autorizan a terceros, generalmente mediante un contrato de licencia a su uso, distribuyendo el *software* en código objeto, y reservándose el código fuente del mismo. Esto implica por lo general, que el usuario sólo tiene derecho a ejecutar el *software* y no dispone de acceso a su código fuente o aun teniendo acceso a él, no tiene derecho a modificarlo ni distribuirlo (Fernández Delpech, 2011, 2014).

Al respecto, Tjor (2014) nos aclara que son programas de código fuente cerrado, protegidos por el sistema del *Copyright* (derecho de autor), al efecto de garantizar legalmente a su creador la posibilidad de licenciar la utilización de su obra bajo ciertas condiciones, restringiendo todo tipo de modificación, estudio, copia, redistribución o ingeniería inversa. En su gran mayoría, se trata de programas licenciados de modo oneroso, preferido por las empresas dedicadas a su desarrollo por obvias razones de protección autoral, tal como sucede en los populares sistemas operativos de Microsoft (Windows), sus paquetes de oficina (Office), etc.

2) *Software* libre

Es un concepto que nace en el contexto de un movimiento filosófico -si se quiere- que se popularizó cuando Richard Stallman creó la *Free Software Foundation* en 1984.

Es el *software* creado por una o varias personas, donde sus titulares transfieren a terceros los derechos totales o parciales sobre la obra y/o autorizan a estos a su uso o modificación con mayor o menor alcance (Fernández Delpech, 2011).

Se lo conoce también como *open source* o código abierto y parte de la premisa de principio que el *software* debería ser libre y accesible a todos. Se caracteriza por distribuir libremente el código objeto y el código fuente del programa, con la idea de que otros programadores puedan estudiarlo e incluso modificarlo para fines de personalización o bien para introducirle mejoras. Los más conocidos ejemplos de esta categoría son el sistema operativo Linux y el paquete de *software* de oficina OpenOffice.org (Hess Araya, 2007).

Tjor (2014) nos dice que la *Free Software Foundation* (FSF) indica que *software* libre significa respetar la libertad de los usuarios y la comunidad, dándoles la libertad de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el *software*.

Contrariamente a lo que se cree, *software* libre no equivale a *software* gratuito, aunque ambas características puedan encontrarse en un mismo producto, nada impide que un *software* libre sea distribuido lucrativamente. Es una cuestión de libertad y no de precio (Hess Araya, 2007).

Cuando se habla de *software* libre o de *software* propietario, se está refiriendo en realidad a dos formas de licenciamiento del *software*, el licenciamiento libre y el licenciamiento propietario, ya que el *software* sea que esté en el dominio público o en el dominio privado, es uno solo, diferenciándose solamente en la forma de licenciamiento empleado (Fernández Delpech, 2014, p. 568).

II. Conclusiones

Finalizamos el primer capítulo con la certeza de haber desarrollado todos los elementos principales de lo que se considera *software* o programas de computación, dejamos en claro que junto con el *hardware* son los elementos fundamentales de lo que se denomina Sistema Informático, y éste a su vez, forma parte de la Informática (como ciencia). Por lo tanto, los efectos jurídicos que genera la interrelación que existe entre la Informática y el Derecho es lo que estudia el Derecho Informático, siendo esta última, una nueva rama del derecho que comprende el tema en examen.

Habiendo repasado los conceptos introductorios del mismo, sus características y naturaleza jurídica, pudimos diferenciar al *software* del *hardware*, y por último elaboramos una clasificación bastante amplia sobre el concepto, lo que nos permite proseguir con nuestro trabajo e introducirnos

de lleno en la problemática real y jurídica del *software*, estudiando paralelamente la protección civil, penal y administrativa vigente en nuestro país en dicha materia.