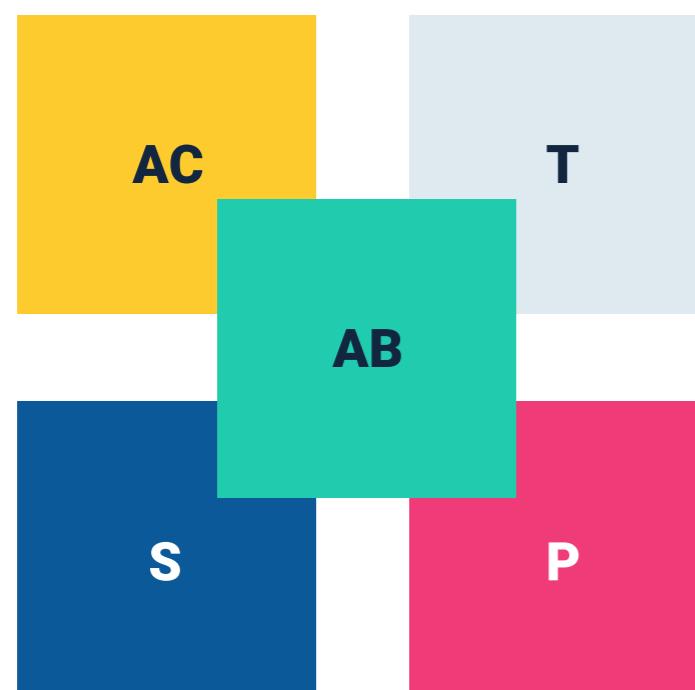


Contrato y planeación de propuesta técnico económica

En este recorrido temático, vamos a plantear una propuesta de manera técnico-económica de acuerdo con las necesidades del cliente y respetando los requerimientos. Lo anterior se hará teniendo en cuenta las normativas, legislaciones y condiciones contractuales de este proceso. Se realiza también una estimación de desarrollo, costos y el talento humano requerido para el proyecto en curso; además del uso de las herramientas para calcular los costos y generar recursos económicos monetizando un servicio o un producto.

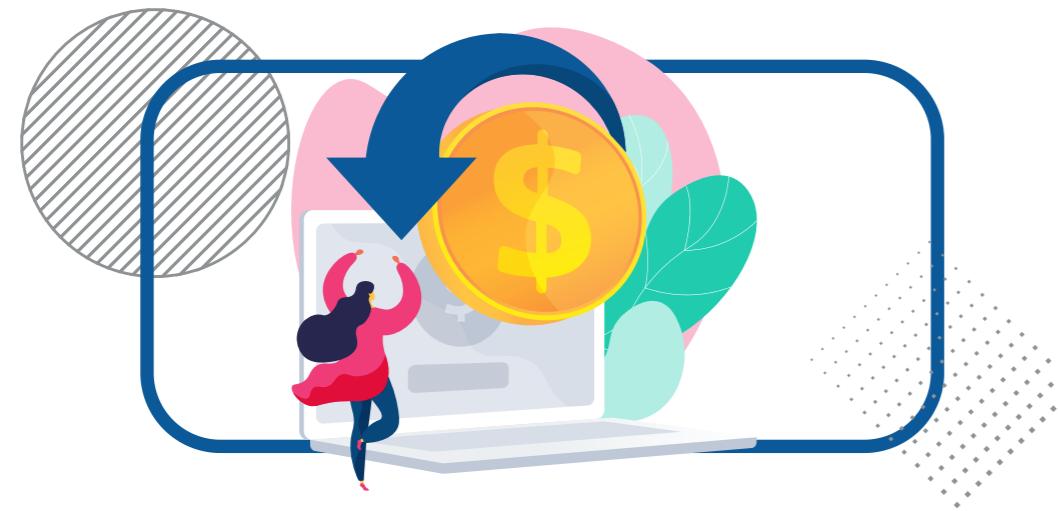


Iniciar >



	PRIMARIO	#EF3C79	ACENTO CONTENIDO	#FECB2F	CB	
	SECUNDARIO	#0B5999	ACENTO BOTONES	#21CBAE		
	NEUTRAL 1	#EFEFEF	NEUTRAL 2	#F9F7EC		

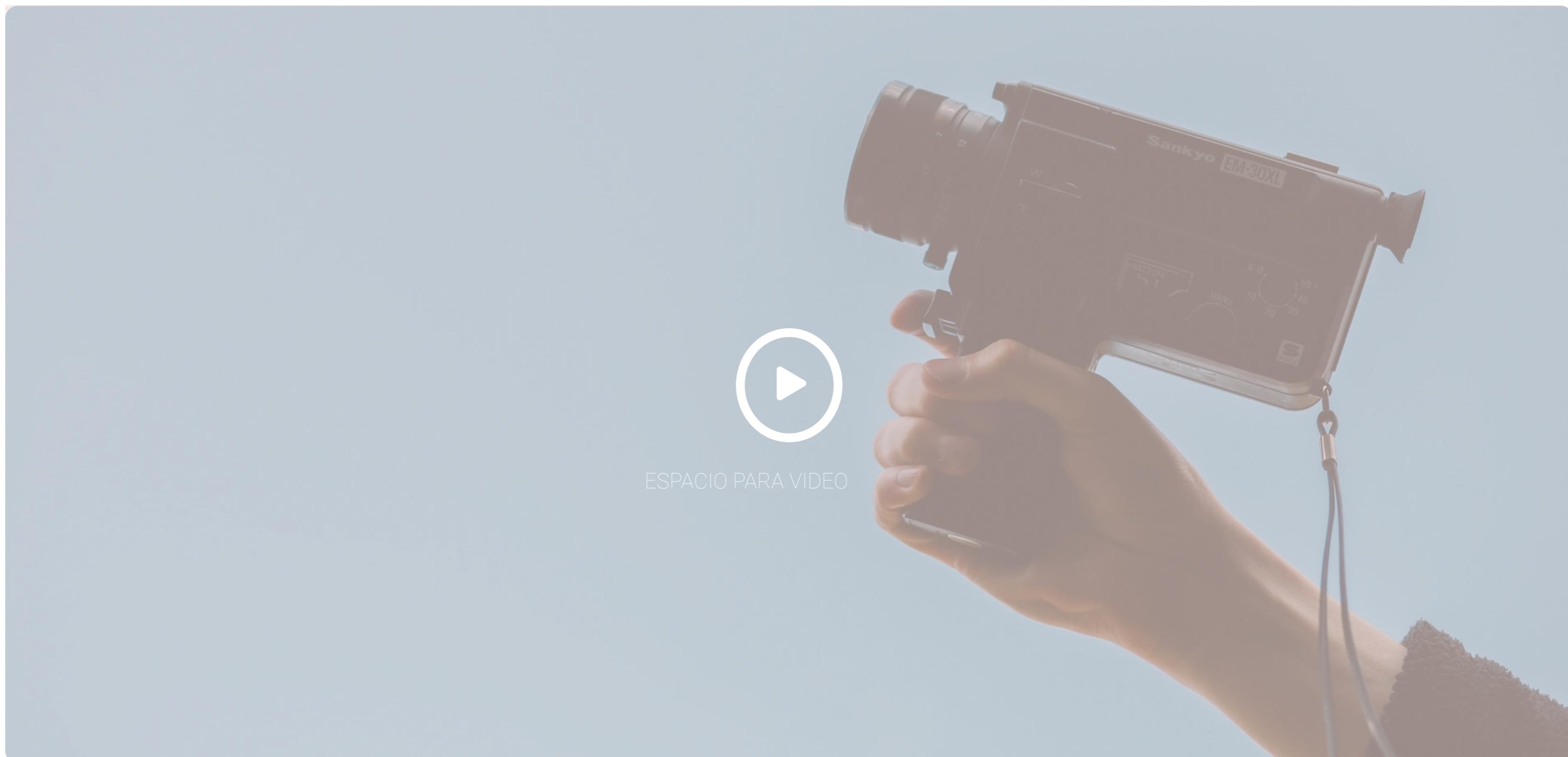
i Introducción



Bienvenido a una nueva experiencia de aprendizaje, en este componente se tiene como objetivo encontrar un punto de equilibrio entre la oferta del cliente, el pago al equipo de trabajo y las ganancias para la empresa, en el marco del desarrollo de un software web.

Este componente tendrá un carácter teórico, en el cual tendremos la oportunidad de explorar las normativas y legislaciones en un proyecto de software, así mismo, se podrá realizar una buena estimación en el equipo de desarrollo y el talento necesario para el desarrollo y la manera óptima de calcular el costo de un proyecto, diseñando un plan de ventas y de monetización en un proyecto de software.

Adicionalmente, se busca desarrollar en el aprendiz las principales habilidades para identificar y aplicar los conceptos básicos que aquí se exponen. Veamos a continuación el siguiente video para profundizar con detalle en los temas mencionados:

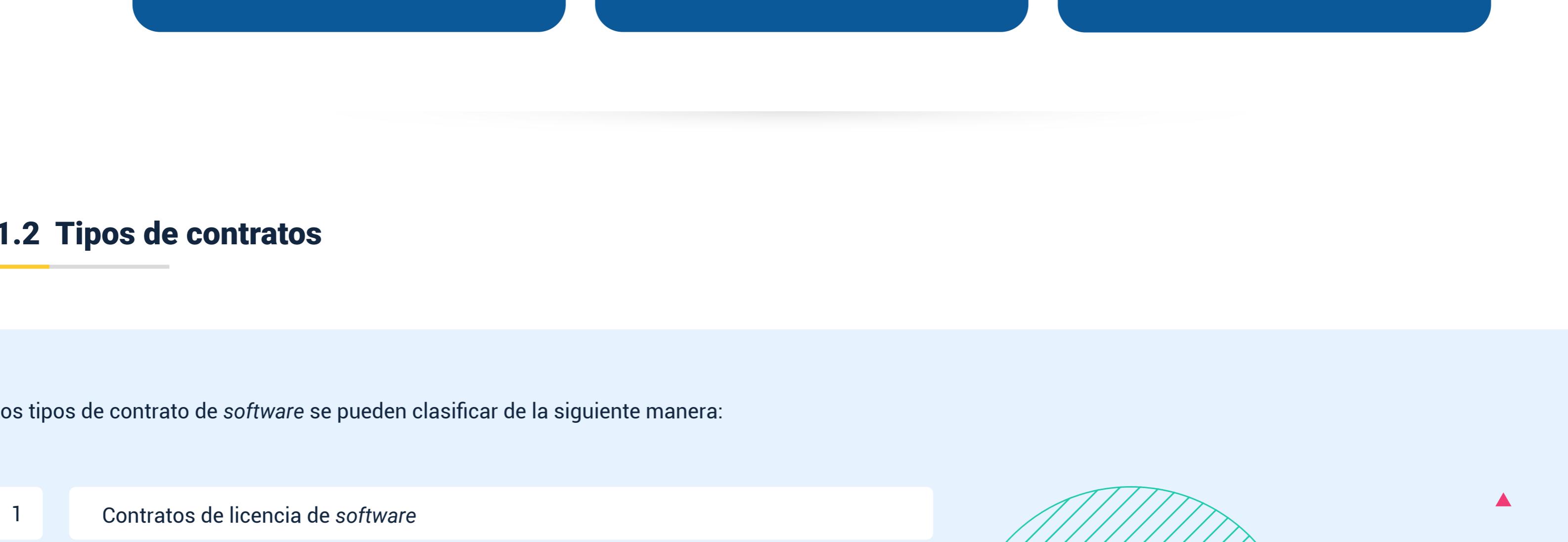


1 Contrato de software

El contrato de software es un tipo de contratación que permite que el propietario del programa conceda un derecho de uso a una persona o empresa a cambio de un intercambio económico. De esta manera, puede tratarse de un contrato de desarrollo de software, entre otros.

1.1 Aspectos mínimos en un contrato de software

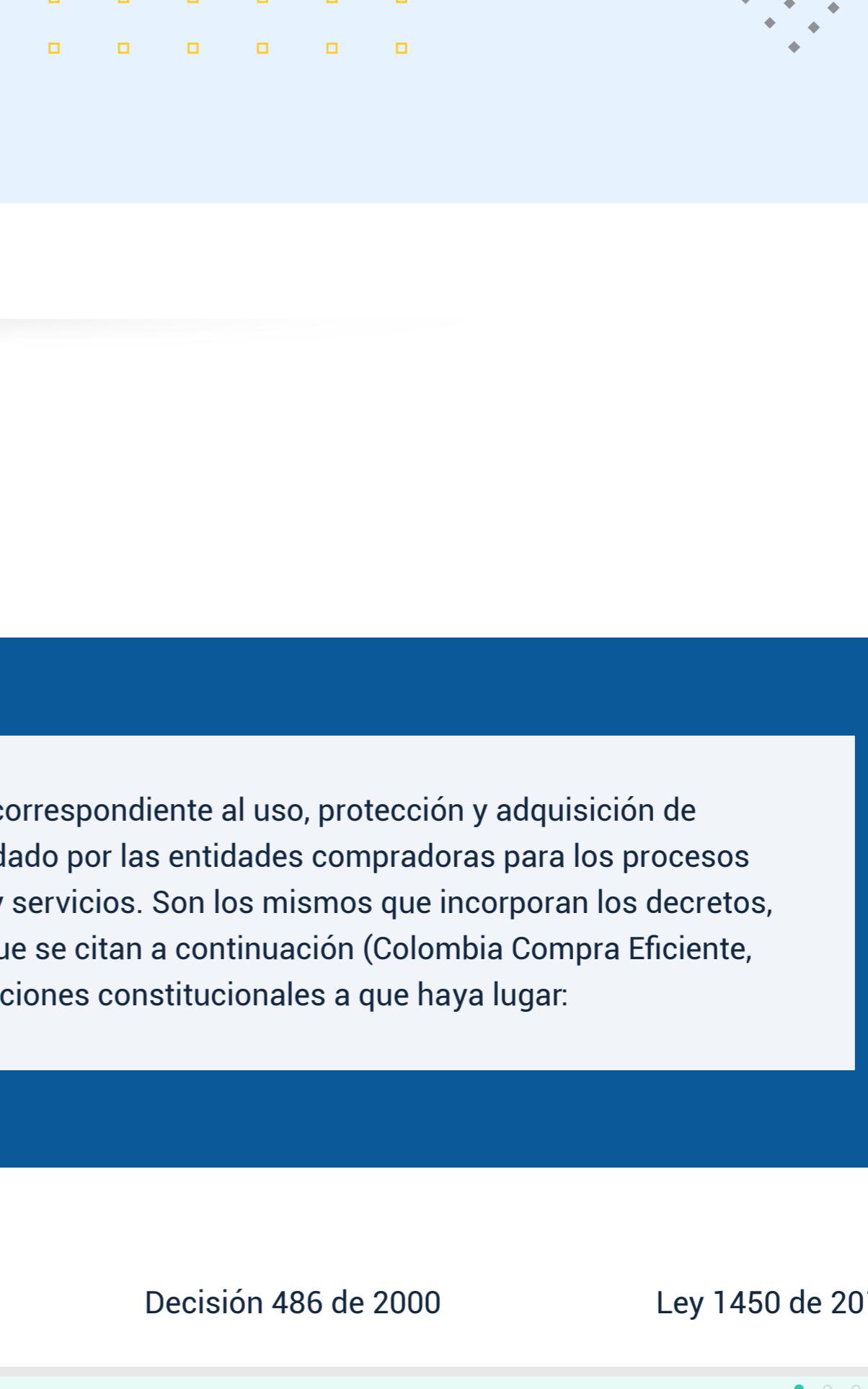
De acuerdo con los intereses de las partes y la modalidad de contrato, las cláusulas que se debe tener en cuenta en un contrato son las siguientes:



1.2 Tipos de contratos

Los tipos de contrato de software se pueden clasificar de la siguiente manera:

- 1 Contratos de licencia de software
- 2 Contratos de desarrollo de software
- 3 **Contrato de soporte y mantenimiento**
este tipo de contrato tiene un software de inicio (desarrollo o de licencia) y permite que el proveedor asegure el correcto funcionamiento del software, tanto en aspectos asociados al buen uso, como la verificación del correcto funcionamiento.
- 4 Contrato de Software As A Service (contrato SAAS)
- 5 Contrato de distribución de software



1.3 Normativa y legislaciones en la contratación de software

El marco normativo y legislativo correspondiente al uso, protección y adquisición de herramientas de software es validado por las entidades compradoras para los procesos de contratación de estos bienes y servicios. Son los mismos que incorporan los decretos, leyes, circulares y resoluciones que se citan a continuación (Colombia Compra Eficiente, 2020), sin perjuicio de las disposiciones constitucionales a que haya lugar:

Ley 23 de 1982 Decreto 1360 de 1989 Ley 565 de 2000 Decisión 486 de 2000 Ley 1450 de 2011

Ley 23 de 1982

La protección jurídica que se concede al software tiene su fundamento en la Ley 23 de 1982, mediante la cual, se establece que "los autores de obras literarias, científicas y artísticas gozarán de protección para sus obras". Ahora bien, la doctrina ha asociado el lenguaje de programación (código objeto y código fuente) a un idioma natural, por lo que el software como herramienta es asociada a una obra literaria que demanda protección como creación. No obstante, lo anterior desconoce componentes diferentes al mero lenguaje de programación evitando proteger de manera integral el software como invención o bajo el amparo de una patente.

Ley 23 de 1982 Decreto 1360 de 1989 Ley 565 de 2000 Decisión 486 de 2000 Ley 1450 de 2011

Decreto 1360 de 1989 Aunado a la Ley 23 de 1982

A través de este decreto se incorpora el concepto de software en la legislación colombiana, allí se define al software como una creación propia del dominio literario, y el Artículo 2 contempla que el software sujeto de protección debe comprender alguno de 3 aspectos: (i) el programa de computador, (ii) la descripción de programa y, (iii) el material auxiliar. Sin embargo, al igual que la generalidad prevista en la Ley 23, el Artículo 7 del decreto señala que "la protección otorgada al soporte lógico no excluye otras formas de protección" reiterando en la protección jurídica que puede predicarse de otros componentes relacionados con el software.

Ley 23 de 1982 Decreto 1360 de 1989 Ley 565 de 2000 Decisión 486 de 2000 Ley 1450 de 2011

Ley 565 de 2000

Por medio de la cual se aprueba el Tratado de la OMPI -Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor (WCT), adoptado en Ginebra, el veinte de diciembre de 1996, en la cual se establece expresamente que los programas de ordenador están protegidos como obras literarias en el marco de lo dispuesto en el Artículo 2 del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión. En el mismo sentido, se refiere a la protección que por derecho de autor debe darse a las compilaciones de bases de datos, sea cual sea su modalidad.

Ley 23 de 1982 Decreto 1360 de 1989 Ley 565 de 2000 Decisión 486 de 2000 Ley 1450 de 2011

Decisión 486 de 2000

La Decisión de la Comunidad Andina de Naciones refiriéndose a los requisitos de la patentabilidad establece en el Artículo 15 el listado para no considerar como invenciones sujetas a patente, "(e) Los programas de ordenador o el soporte lógico" derivando así en la imposibilidad de proteger jurídicamente mediante patente al software como creación. Así mismo, se evidencia una falencia en el desarrollo normativo que contradice acuerdos internacionales como el Acuerdo sobre los aspectos de los derechos de propiedad intelectual relacionados con el comercio, que establece en el Artículo 27, como materia patentable lo siguiente:

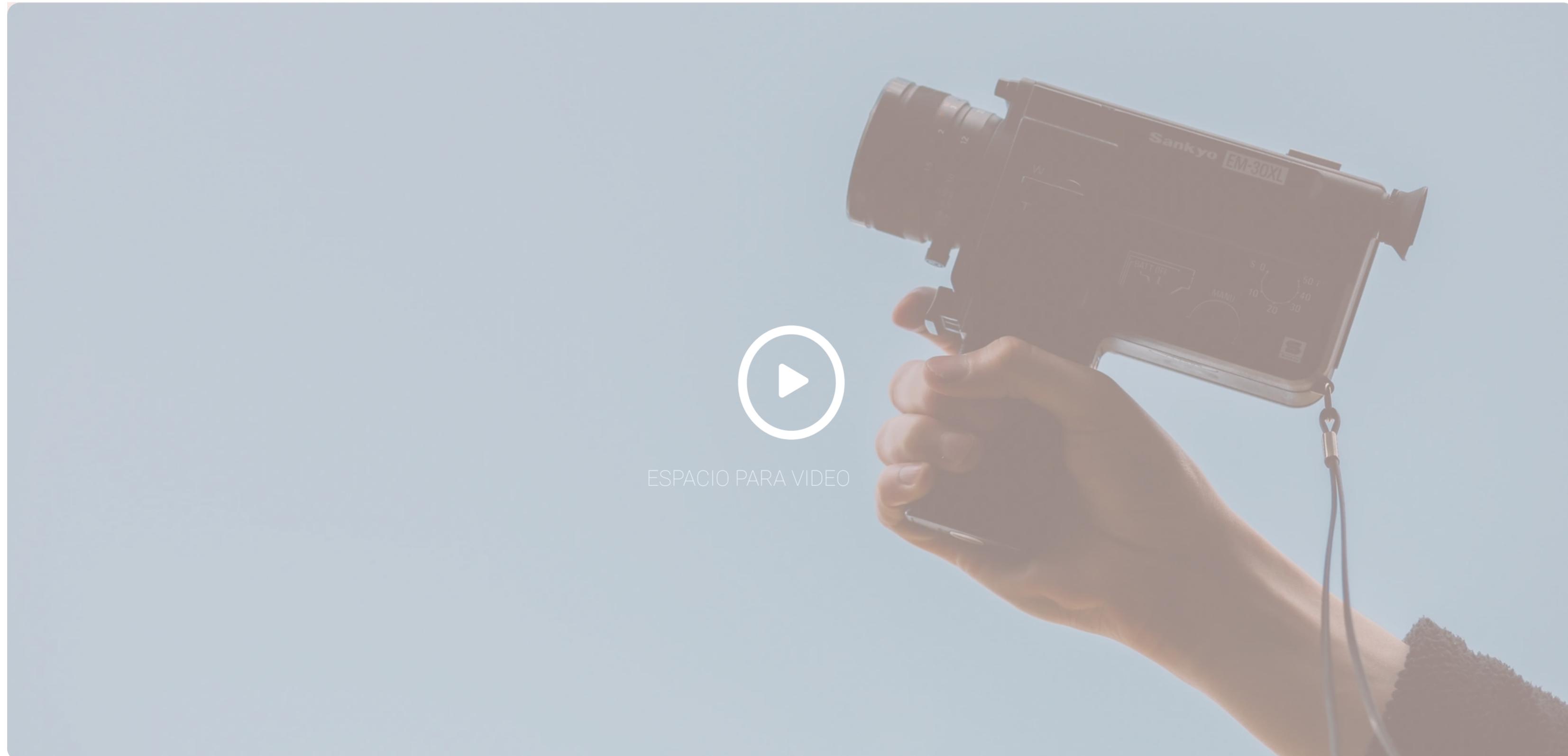
Sin perjuicio de lo dispuesto en los párrafos 2 y 3, las patentes podrán obtenerse por todas las invenciones, sean de productos o de procedimientos, en todos los campos de la tecnología, siempre que sean nuevas, entrañen una actividad inventiva y sean susceptibles de aplicación industrial.

Dicha limitación contraría la política de gobierno digital establecida por el Ministerio de las Tecnologías de la Información y las Telecomunicaciones pues impide la protección de nuevas creaciones en materia tecnológica.

Ley 23 de 1982 Decreto 1360 de 1989 Ley 565 de 2000 Decisión 486 de 2000 Ley 1450 de 2011

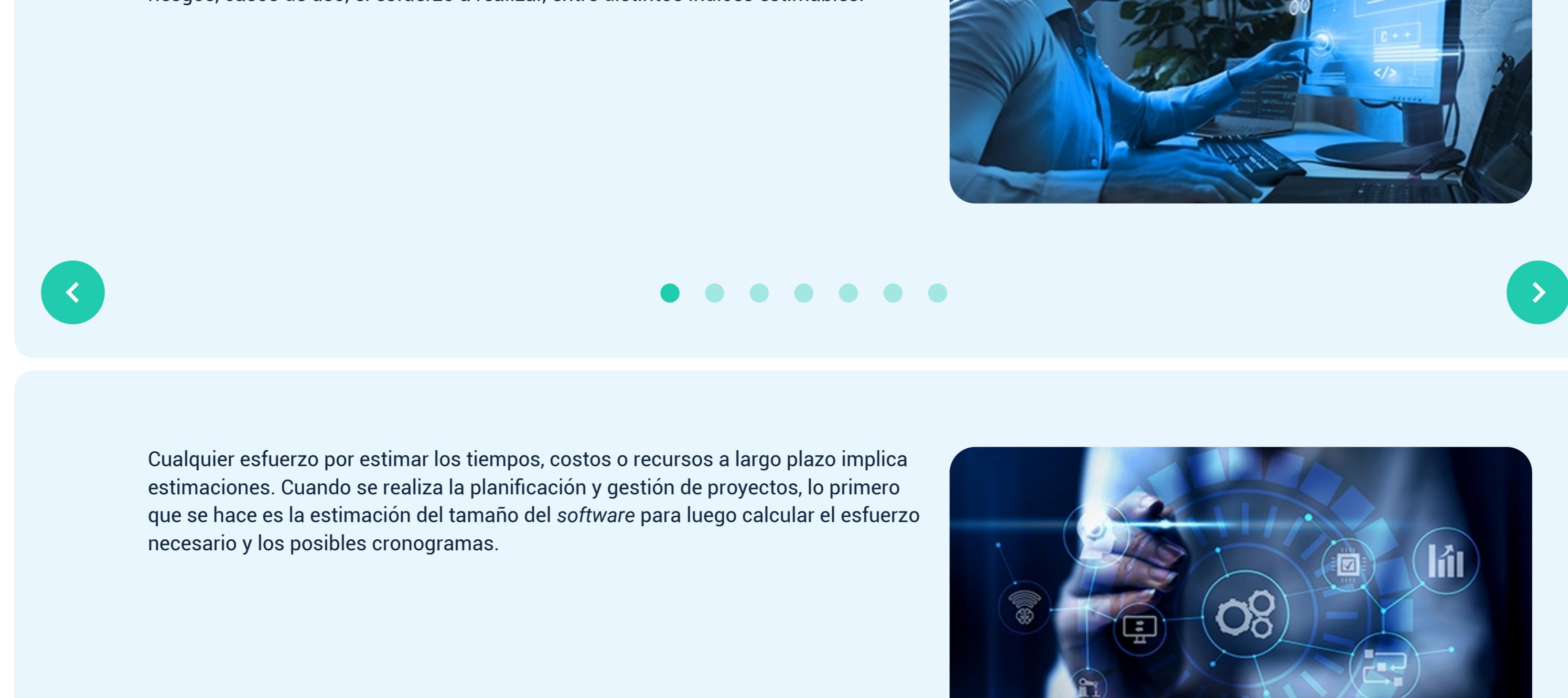
2 Cálculo de costos de software

Veamos el siguiente video para entrar en contexto con el siguiente tema: cálculo de costos de software



3 Técnicas de estimación de proyectos de software

El primer elemento de un proyecto de software es la planificación. En esta etapa del proceso, una de las tareas con mayor valor es la estimación, tanto de recursos, esfuerzos, costes y tiempos para el buen desarrollo del proyecto.



Dentro del entorno de un proyecto de software existen diferentes factores a estimar, por ejemplo: el cálculo del tiempo de ejecución de cada una de las actividades a realizar dentro del proyecto (análisis, gestión, desarrollo, pruebas, etc.) o del proyecto en general, así como las validaciones de los casos de prueba por los riesgos, casos de uso, el esfuerzo a realizar, entre distintos índices estimables.

Cualquier esfuerzo por estimar los tiempos, costos o recursos a largo plazo implica estimaciones. Cuando se realiza la planificación y gestión de proyectos, lo primero que se hace es la estimación del tamaño del software para luego calcular el esfuerzo necesario y los posibles cronogramas.

La estimación de costos es una de las primeras tareas que se hacen en el proyecto, casi siempre, después de señalar los requerimientos; no obstante, se sigue realizando con regularidad a lo largo de la ejecución del proyecto con el objetivo de adaptar la precisión en la estimación previa.

La estimación de por si es difícil, una forma adecuada para acercarse puede ser desarrollando de manera incremental los requerimientos, detallando las estimaciones a medida que se va encontrando más información. La estimación temprana puede ser riesgosa, dado que la cantidad de información que se tiene al principio del sistema, por lo general, es muy limitada, lo que dificulta el proceso.

Resulta útil tener muy claro los usos principales de la estimación de costos en la administración de proyectos:

- En la etapa de planteamiento indicar la cantidad de personas que se requieren para la ejecución del proyecto y establecer cronogramas correctos.
- Durante el desarrollo del proyecto permite conocer su alcance y calcular si el proyecto se está desarrollando según lo indicado en el cronograma. De lo contrario, se deberá crear un plan de acción a tiempo. Para lo anterior es necesario contar con métricas que permitan calcular el nivel de cumplimiento del desarrollo del software.

En el ámbito de la ingeniería de software, la estimación de costos se trata de validar el alcance del software, así como la cantidad de personas que se necesitará para el desarrollo del producto. Hay una notoria relación entre calidad, costo y cronograma. Estos tres aspectos están estrechamente vinculados entre sí.

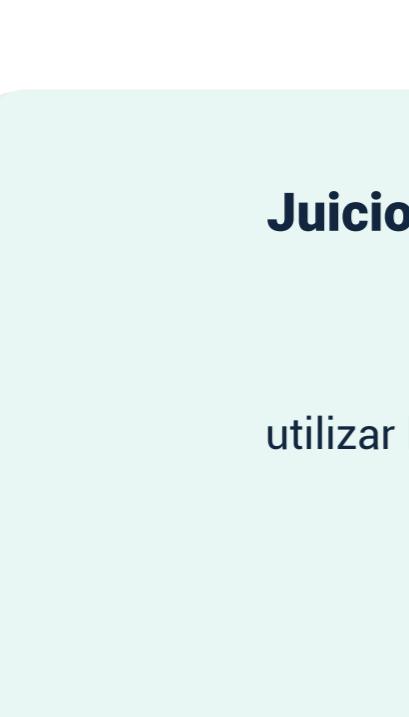
Una historia de usuario es una definición breve, informal y en lenguaje sencillo de lo que un usuario quiere hacer dentro de un producto de software para conseguir algo que le resulte valioso.

Una historia de usuario es necesario resaltar que el proceso de estimación es impredecible dado que se intenta medir elementos como el esfuerzo, tamaño y costos de un proyecto que aún no ha iniciado y del cual es muy probable que se tenga poca información. Es muy importante que el representante de la gestión del proyecto tenga buen conocimiento en el tema, que tenga en cuenta trabajos pasados para repetir los mismos problemas anteriores y sea una persona con características que le permitan prever posibles riesgos. Además, hay que tener presente que la estimación en horas tempranas es de mayor riesgo, por lo que hay que realizar constantemente validaciones y ajustes a las estimaciones ya existentes. Los usuarios suelen seguir el patrón rol - función - beneficio:

3.1 Técnicas de estimación del tamaño del software

Existen varias técnicas para medir el tamaño del software, entre las cuales podemos encontrar:

Medición de líneas de código fuente producidas (SLOC)



Es una de las técnicas más conocidas, pero requiere disponer del código lo que representa una restricción para realizar estimaciones tempranas, ya que la información que se obtiene en esos momentos y el tiempo que se dispone para las primeras estimaciones del proyecto son normalmente escasos. De todas las etapas en las cuales se revisa la factibilidad del proyecto, cuando menos datos/información se normaliza es en el momento inicial, cuando todavía se está revisando la factibilidad del proyecto. Desde el punto de vista del desarrollo y comercialización del software específico para terceros, esa estimación temprana tiene gran importancia. Sin embargo, la tarea de estimación no permite una mejor planificación, lo que falta del proyecto.

- 1 Entradas al sistema.
- 2 Salidas del sistema.
- 3 Consultas.
- 4 Grupos de datos lógicos del sistema.
- 5 Grupos de datos lógicos que no pertenecen al sistema, pero que el sistema utiliza.

Debe calcular cuántos acontecimientos de cada parámetro contiene un sistema, calificándolos según su grado de complejidad en: baja, media y alta. Cada parámetro para una complejidad dada tiene un tipo de peso determinado, estos son los puntos de función asignados a ese parámetro. Luego de haber realizado esta sucesión, los puntos de función se gradúan a los requerimientos del sistema.

La fórmula de Albrecht para calcular los puntos función, es la siguiente:

$$FP = UFP \times TCF$$

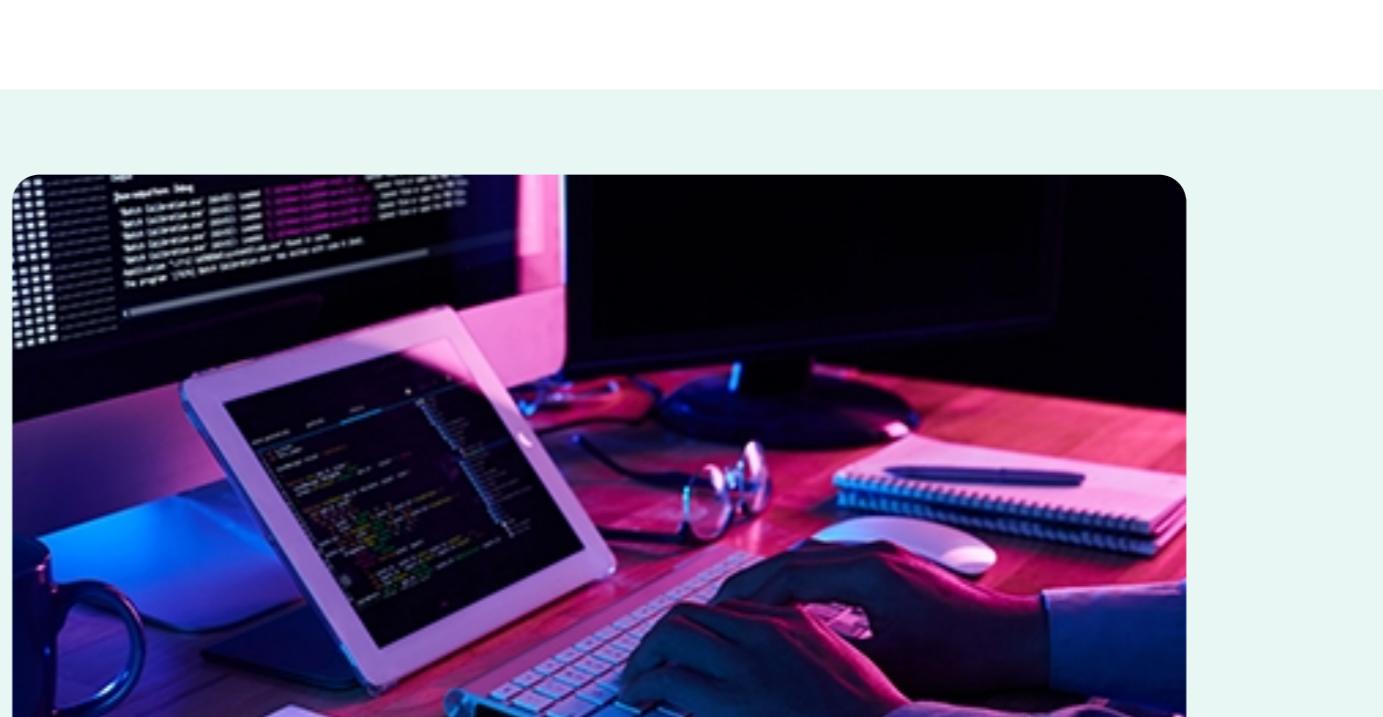
UFP: son los Puntos Función No Ajustados

TCF: Factor de Complejidad Técnica.

Los UFP son deducidos mediante la sumatoria de los pesos de todos los parámetros identificados.

Existen varios tipos de metodologías que estiman el tamaño del software; por ejemplo:

- IFPUG-FPA (Function Point Analysis).
- MKII (Mark II).
- FFP (Full Function Point).
- NESMA FPA (Netherlands Software Metrics Users Association Function Point Analysis).



Puntos de Casos de Usos (UCP)



El método UCP clasifica los casos de uso en simple, promedio y complejo con factores de peso 5, 10 y 15, respectivamente. La clasificación se realiza con base en el número de transacciones que contiene el caso, 1 a 3 para simple, 4 a 7 para promedio y 8 o más para complejos. Cabe aclarar que se describe una transacción como el evento que ocurre entre el sistema a ser modelado y un actor. Luego de haber especificado los casos de uso del sistema ya se tiene un mayor nivel de detalle para poder estimar, de manera más precisa, los puntos de función. Posteriormente, con las transacciones de puntos de función [Consultas Externas, Entradas Externas y Salidas Externas], los puntos de función sobre estos Casos de Uso con mayor detalle, la estimación del esfuerzo y tamaño es más puntual que la anterior. El FPA calcula el tamaño del software determinando la funcionalidad provista al usuario basándose solamente en las especificaciones funcionales y el diseño lógico. Este método consiste en analizar los puntos de función teniendo en cuenta:

- 1 Entradas al sistema.
- 2 Salidas del sistema.
- 3 Consultas.
- 4 Grupos de datos lógicos del sistema.
- 5 Grupos de datos lógicos que no pertenecen al sistema, pero que el sistema utiliza.

Debe calcular cuántos acontecimientos de cada parámetro contiene un sistema, calificándolos según su grado de complejidad en: baja, media y alta. Cada parámetro para una complejidad dada tiene un tipo de peso determinado, estos son los puntos de función asignados a ese parámetro. Luego de haber realizado esta sucesión, los puntos de función se gradúan a los requerimientos del sistema.

La fórmula de Albrecht para calcular los puntos función, es la siguiente:

$$FP = UFP \times TCF$$

UFP: son los Puntos Función No Ajustados

TCF: Factor de Complejidad Técnica.

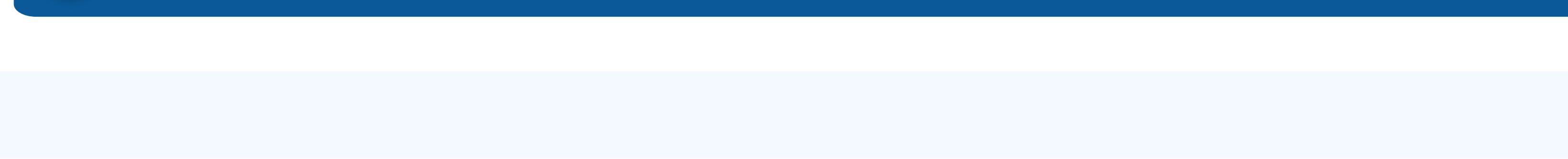
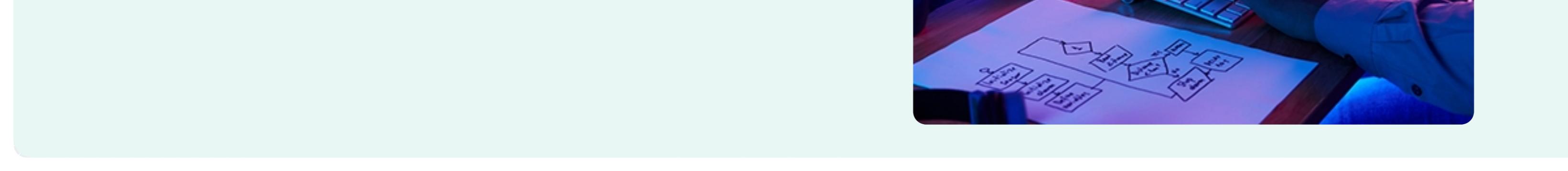
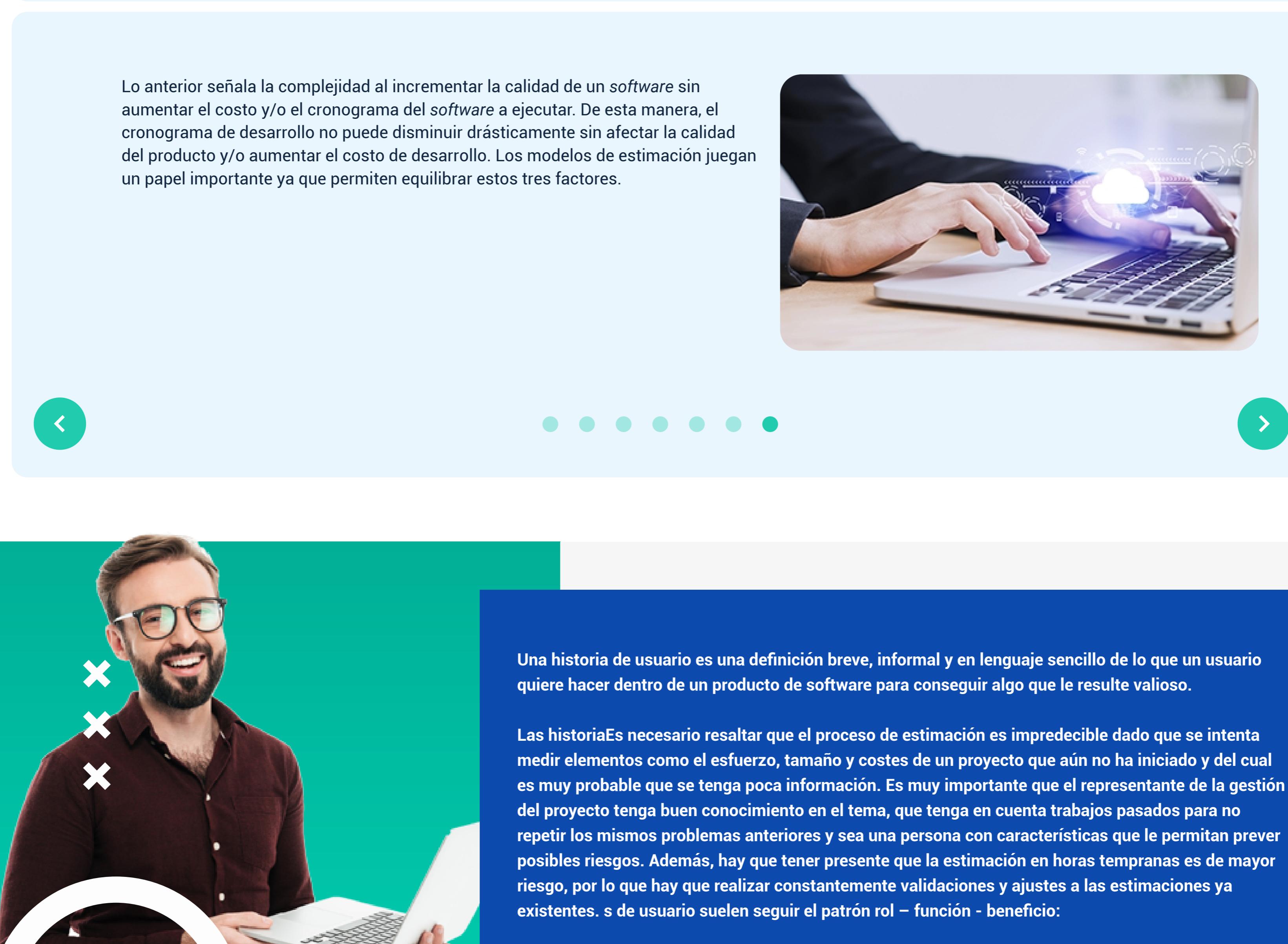
Los UFP son deducidos mediante la sumatoria de los pesos de todos los parámetros identificados.

Existen varios tipos de metodologías que estiman el tamaño del software; por ejemplo:

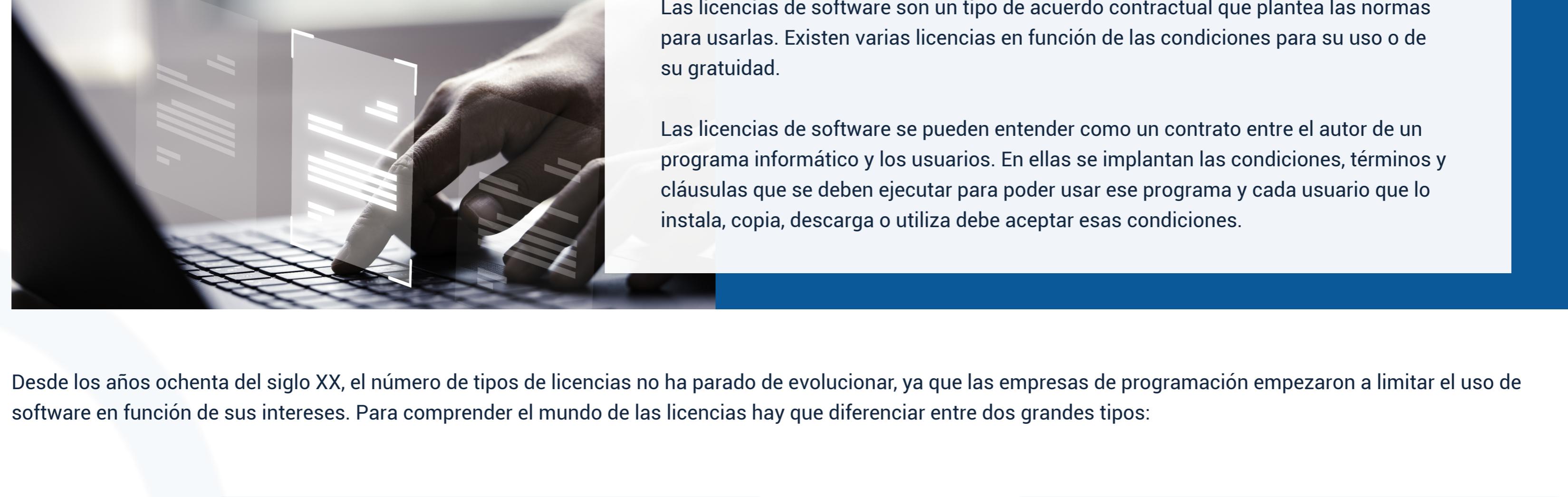
- IFPUG-FPA (Function Point Analysis).
- MKII (Mark II).
- FFP (Full Function Point).
- NESMA FPA (Netherlands Software Metrics Users Association Function Point Analysis).

3.2 Técnicas de estimación de esfuerzo (cantidad de personas necesarias)

Para cuantificar el posible esfuerzo en un proyecto de software se utiliza la medida de meses/persona (PM), lo que representa los meses de trabajo de una persona en el equipo de desarrollo del proyecto. Existen diferentes métodos de estimación del esfuerzo, entre esos están:



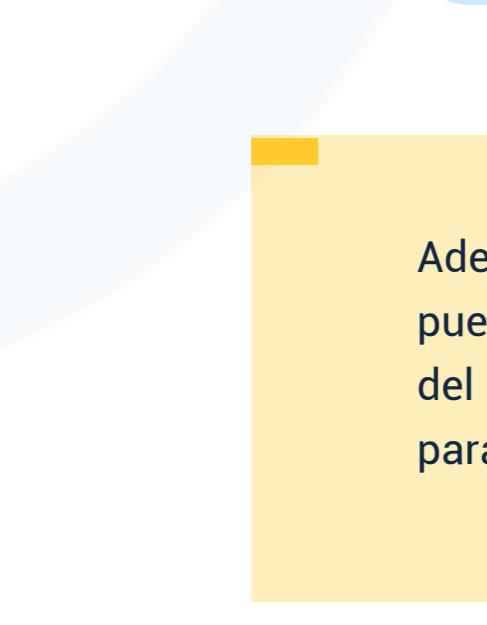
4 Licenciamiento



Las licencias de software son un tipo de acuerdo contractual que plantea las normas para usarlas. Existen varias licencias en función de las condiciones para su uso o de su gratuidad.

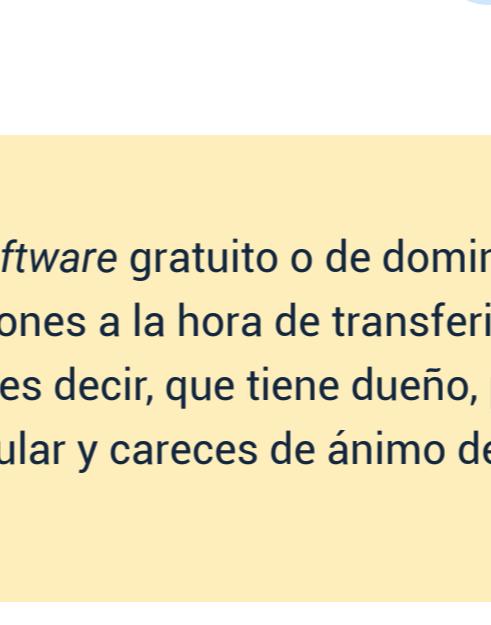
Las licencias de software se pueden entender como un contrato entre el autor de un programa informático y los usuarios. En ellas se implantan las condiciones, términos y cláusulas que se deben ejecutar para poder usar ese programa y cada usuario que lo instala, copia, descarga o utiliza debe aceptar esas condiciones.

Desde los años ochenta del siglo XX, el número de tipos de licencias no ha parado de evolucionar, ya que las empresas de programación empezaron a limitar el uso de software en función de sus intereses. Para comprender el mundo de las licencias hay que diferenciar entre dos grandes tipos:



El software libre carece de copyright

no tiene por qué ser gratis, pero el usuario tiene libertad sobre ese programa para modificarlo, usarlo, copiarlo y distribuirlo a su manera. Eso permite a los millones de desarrolladores del mundo aumentar sus posibilidades y expandir el conocimiento.



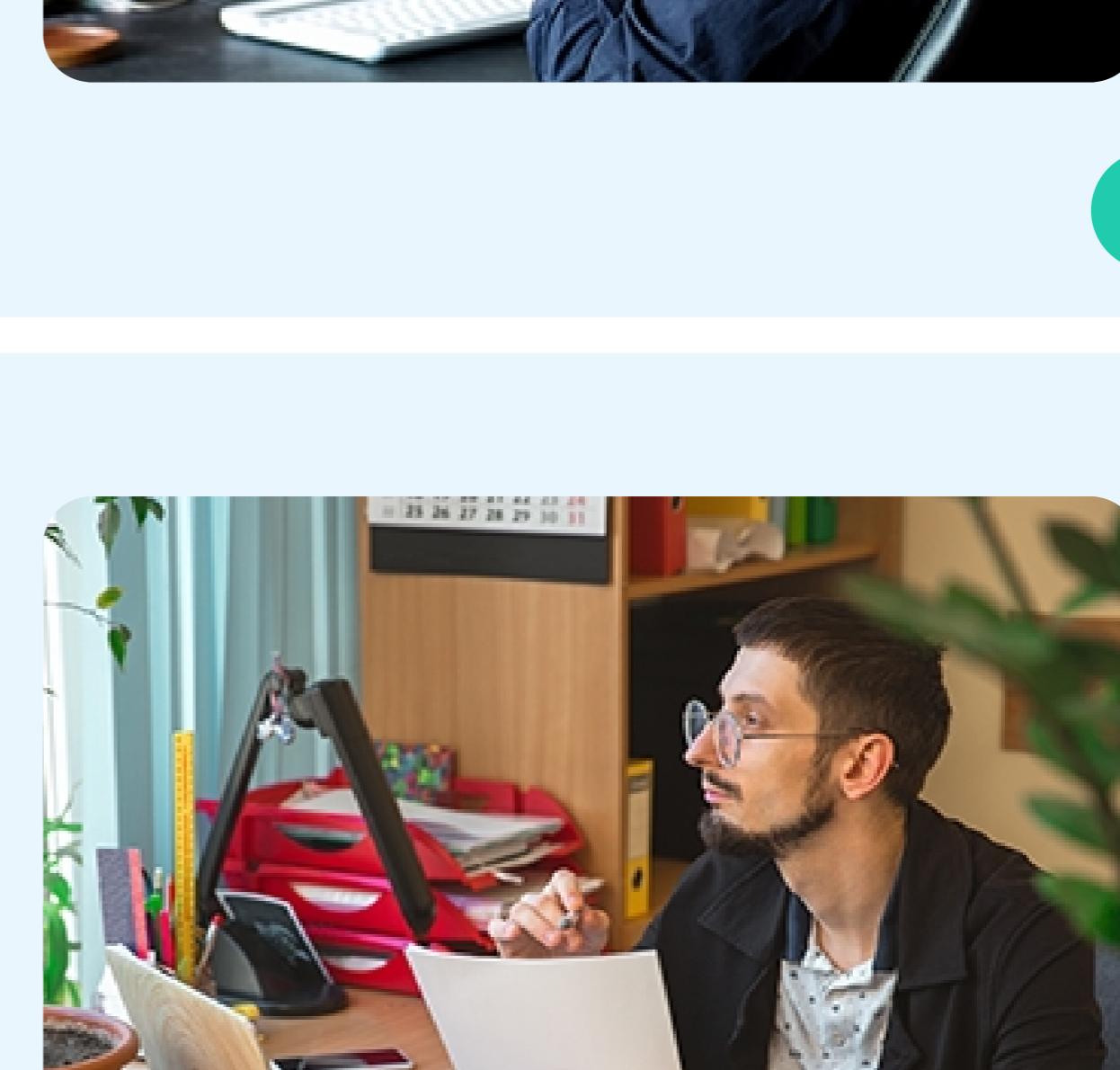
El software de propietario

es aquel que su creador/dueño legal prohíbe la copia o delimita su redistribución y modificación sin su autorización o sin un pago previo

Además de estos dos tipos de licencias, se puede encontrar el software gratuito o de dominio público, que carece de copyright y se puede utilizar libremente, aunque el autor puede imponer limitaciones a la hora de transferir el programa y los trabajos generados del mismo. Por otro lado, existe la licencia de software semilibre, es decir, que tiene dueño, pero el programa viene con permisos para distribuir, usar, copiar y modificar si eres una persona particular y careces de ánimo de lucro.

Cuando se toma un programa, sea bajo el tipo de licencia que sea, lo que realmente se adquiere es una licencia de uso, nunca el programa en sí y, como norma principal, se otorga una licencia por punto de uso o dispositivo. Veamos algunos tipos de licencias:

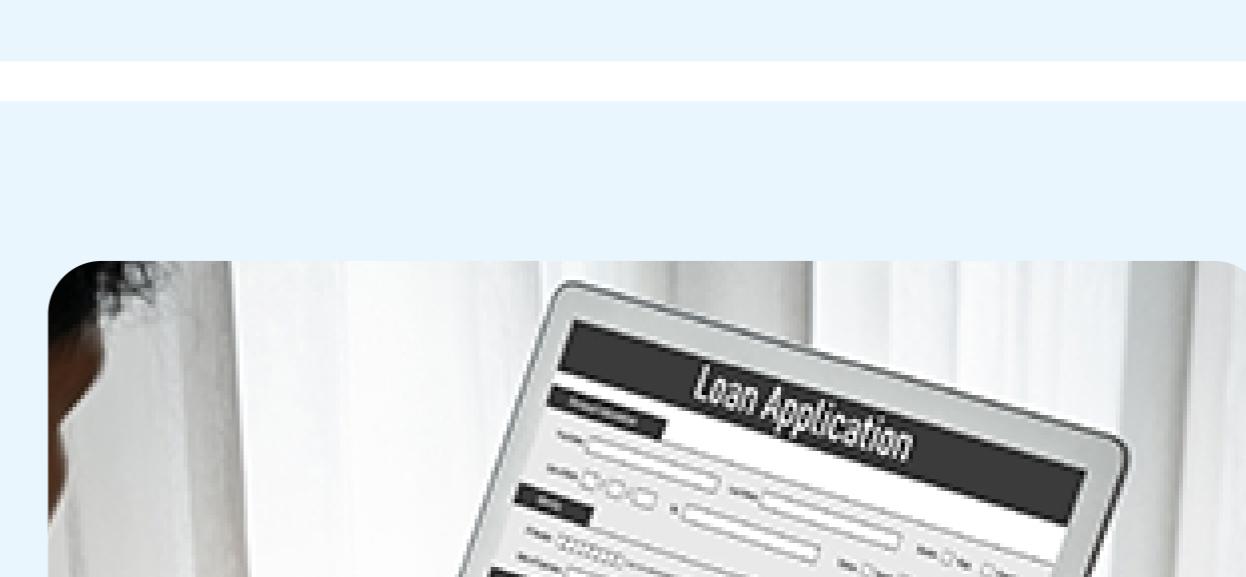
Licencias de software libre: las licencias de software libre permiten ejecutar el programa, estudiar su funcionamiento, adaptarlo a las necesidades de cada usuario, copiarlo y redistribuirlo y, por supuesto, mejorarlo y poner esas mejoras al alcance y servicio de toda la comunidad.



Licencias de software libre no protegido con copyleft: son aquellas cuyo autor autoriza para redistribuir, modificar y añadir restricciones adicionales a algunas copias o versiones modificadas, que pueden no ser libres de manera completamente y requieren un software pago para que ejecute algunos elementos.

Licencias de software libre protegidas con copyleft: en contraposición al copyright o copia reservada (derechos de autor), son software libre que no permiten a quienes las cambian o redistribuyen agregar ninguna restricción adicional. Así, cada copia del software, aun si ha sido cambiado, debe ser software libre. Copyleft es una práctica legal y común que consiste en el ejercicio del derecho de autor para motivar el libre uso y distribución de una obra y pide las mismas libertades al distribuir sus derivados y copias.

Licencia de software de GPL (GNU LGPL): es software libre y permite que se enlace con módulos no libres. Debe estar disponible y accesible para copias ilimitadas y a cualquier persona que lo solicite.



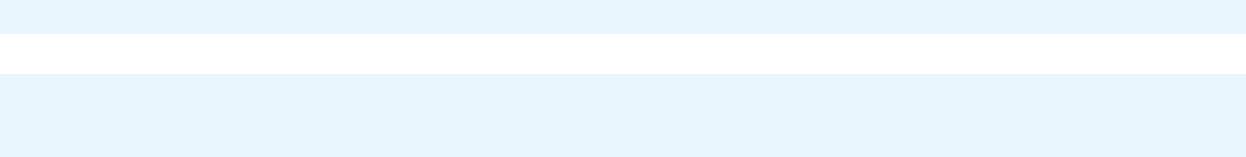
Licencia BSD: es una licencia permisiva, pues casi no impone condiciones sobre lo que un usuario puede hacer. Es la menos restrictiva para los desarrolladores. El software puede ser vendido y no hay responsabilidad de incluir el código fuente.

Licencias de MPL: son las de Mozilla Firefox, Mozilla Thunderbird y la mayoría de otros software de Mozilla. Son algo menos permisivas que las anteriores.

Licencia de software de Debian: su nombre completo es Debian Free Software Guidelines (DFSG) y de ella deriva la licencia de Open Source Initiative. Decreta en sus criterios de distribución la exigencia de la redistribución libre (también del trabajo derivado bajo la misma licencia del original), publicación del código fuente y que no puede excluir a grupo de personas o ninguna persona, ni el software ni ninguna forma de su utilización.



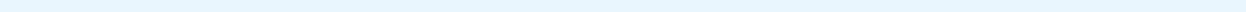
Licencia Open Source Initiative: una de las licencias más conocidas y utilizadas, llamada también fuentes abiertas o código abierto, que permite la libre distribución (también del código fuente), la variación del mismo, los desarrollos derivados y su redistribución en las mismas condiciones que el software original.



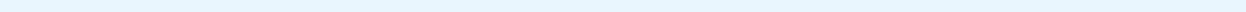
Licencia de software de X.org (X Windows System): distribuida por el Consorcio X (es la organización que controla su desarrollo), es usada en los sistemas operativos Linux y UNIX nacidos como una alternativa a la minoría de los sistemas gráficos como Microsoft. Es también software libre, aunque sin adherirse al copyleft.

Licencia de software Freeware: autoriza el uso del software de forma gratuita y libre, pero en general a particulares y no a empresas u organismos oficiales.

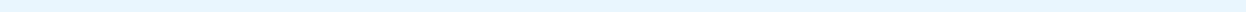
Licencia de software de Donationware: es similar a la licencia Freeware, pero solicita a cambio una donación, aunque no sea obligatorio hacerlo para poder utilizarla. La licencia de software de Postcardware es similar, pero, en lugar de una donación, suele solicitar el envío de una postal como confirmación de su utilización.



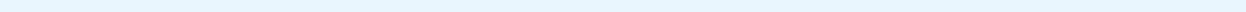
Licencias de prueba o Shareware: todos los usuarios se han encontrado alguna vez con este tipo de licencia. Suelen pedir un registro previo y autoriza el uso de un programa para que el usuario lo evalúe y, posteriormente, lo compre. La licencia tipo Demo es una parte de un programa para su evaluación. Son muy populares en el caso de software de juegos, etc.



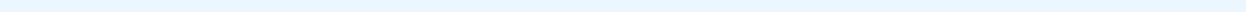
Licencias de software de propietario: con las licencias de software de código cerrado o software propietario o privativo (llamadas CLUFs o para usuario final), los propietarios establecen los derechos de modificación, uso, copia, distribución, redistribución, cesión y cualquier otra consideración necesaria. Los fabricantes de programas sometidos a este tipo de licencias ofrecen servicios de actualizaciones durante el tiempo de vida del producto y, además, soporte técnico. Este tipo de licencias no permiten que el software sea cambiado, copiado o distribuido, desensamblado; además, controlan el número de copias que pueden ser instaladas y los fines establecidos para los cuales puede ser utilizado.



Licencia de software de privativo: su uso, redistribución o modificación está prohibida, o requiere de la solicitud de una autorización.



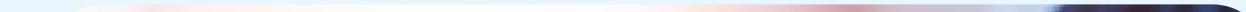
Licencia de software de comercial: es aquél que está desarrollado por una entidad que tiene la intención de monetizarlo por su uso.



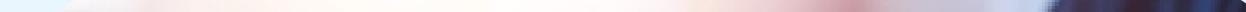
Licencia de software de OEM: supedita su venta a que ese software debe formar parte de un equipo nuevo (por ejemplo, el caso del sistema operativo Microsoft cuando nos compramos un nuevo ordenador).



Licencia de software de Retail: es software que se vende, el programa es cien por ciento propiedad del usuario nuevo y puede regalarlo o venderlo.



Licencia de software de volumen: su destino son los grandes usuarios (empresas) y el contrato de adquisición estipula un determinado número de equipos que pueden utilizar el mismo código.



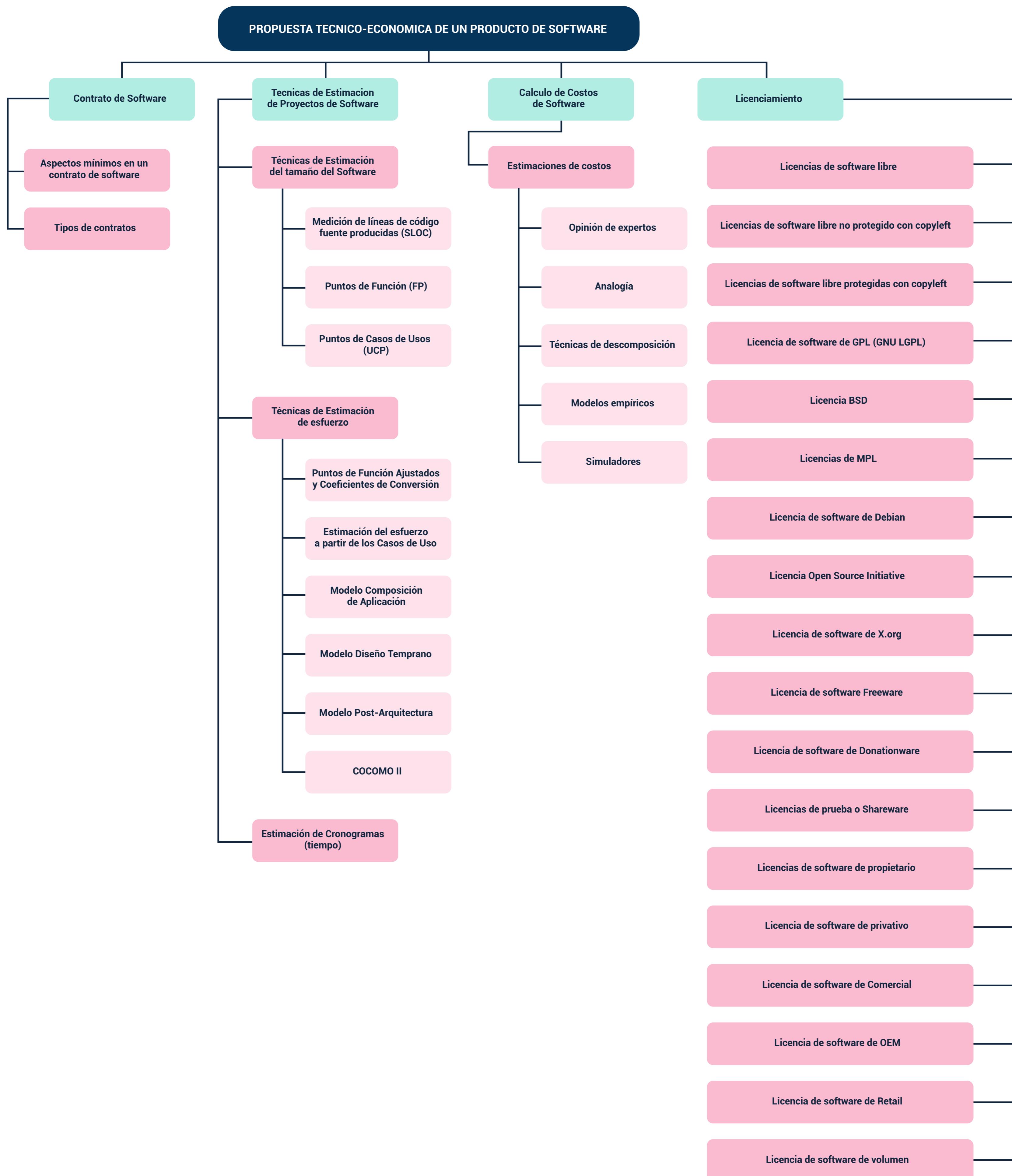
Recuerde explorar los demás recursos que se encuentran disponibles en este componente formativo; para ello, diríjase al menú principal, donde encontrará la síntesis, una actividad didáctica para reforzar los conceptos estudiados, material complementario, entre otros.

Desarrollo de aplicaciones web Full stack

Síntesis: Planeación y recolección de información según necesidades.



De acuerdo con las temáticas abordadas en este componente formativo, se observa que se ha realizado un barrido sobre los temas principales a tratar, a saber, la generación de una propuesta de manera técnico/económica de acuerdo con las necesidades del cliente, teniendo en cuenta las normativas, condiciones contractuales que del proceso, realizando el proceso de estimación de desarrollo, costos, además del uso de las fórmulas para calcular los costos del proyecto y generar recursos económicos monetizando un servicio o un producto dependiendo del tipo de licenciamiento que prefiere generar el proyecto.





Portada actividad

800 x 800



Portada actividad

800 x 800



Imagen acompañamiento Actividad

600 x 600



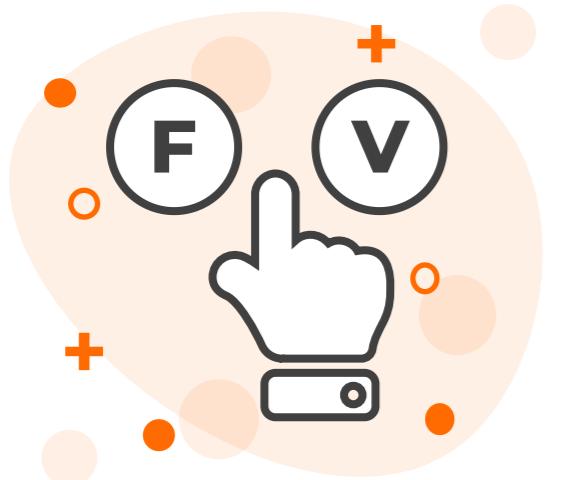
Imagen Resultado

900*600

i

Actividad didáctica

Verdadero y falso



Verdadero y falso

Teniendo en cuenta los conceptos del componente anterior recién vistos, validar las siguientes preguntas de verdadero y falso para así poder realizar la retroalimentación.

Realizar 