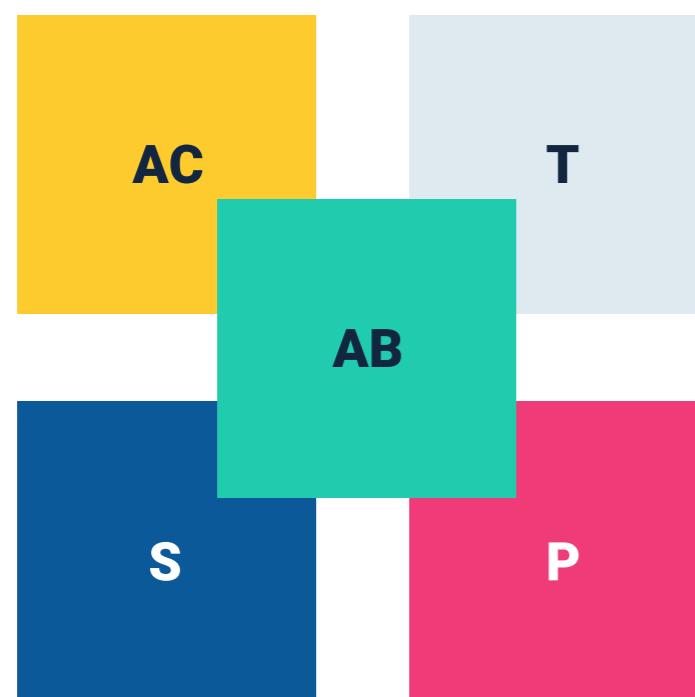


Contrato y planeación de propuesta técnico - económica

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.

- X
- X
- X

[Iniciar >](#)

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



Introducción

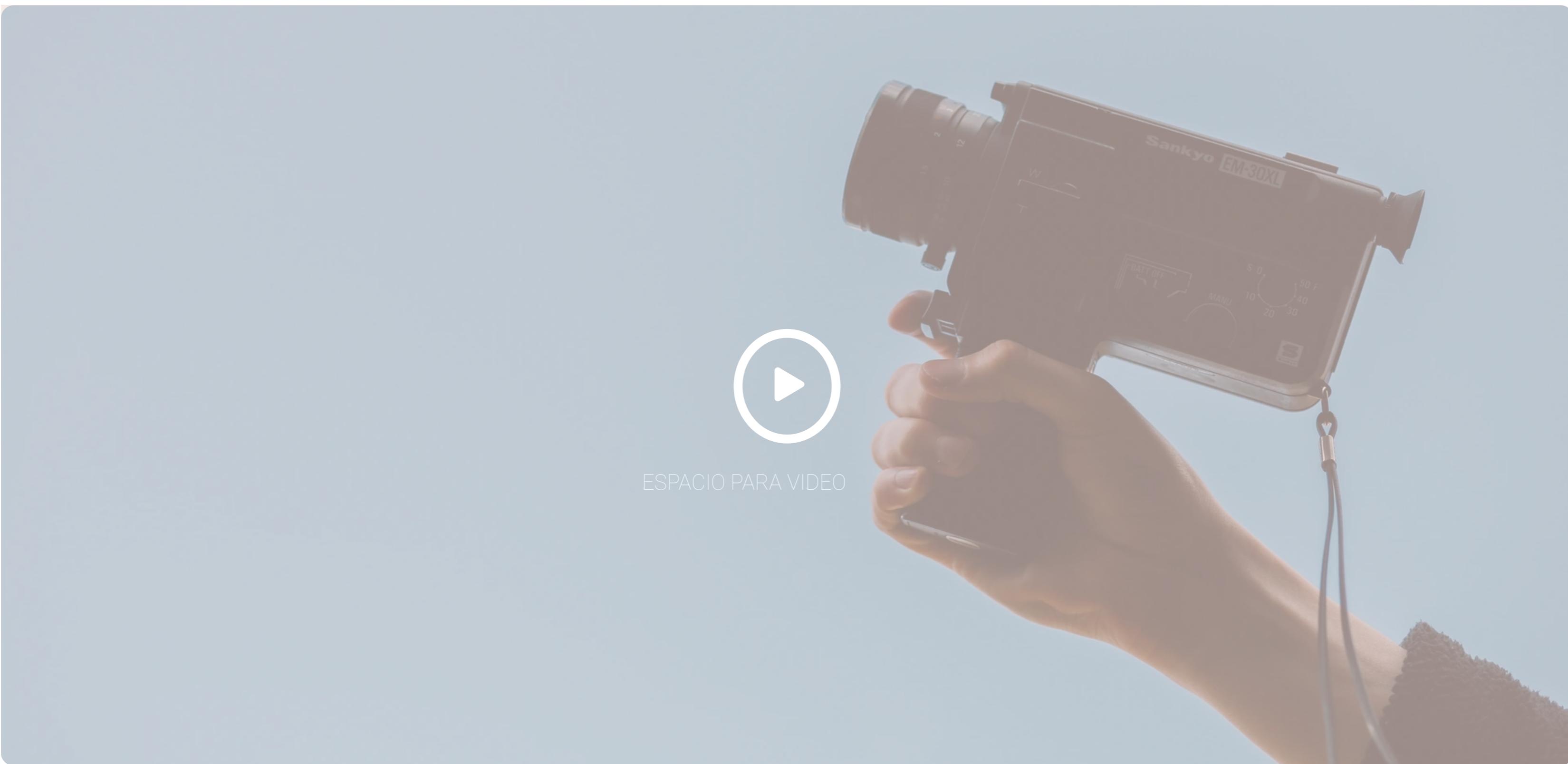


Apreciado aprendiz, bienvenido a este componente formativo, donde se estudiará el punto de equilibrio entre la oferta del cliente, el pago al equipo de trabajo y las ganancias para la empresa, en el marco del desarrollo de un software web. Afiance sus conocimientos y capacidades para realizar una buena estimación de un equipo y del talento necesarios; sepa calcular, de manera óptima, el costo del proyecto y diseñe con efectividad y acierto, un plan de ventas y monetización adecuados, para el proyecto de software.

En el siguiente video conocerá, de forma general, la temática que se estudiará a lo largo del componente formativo.



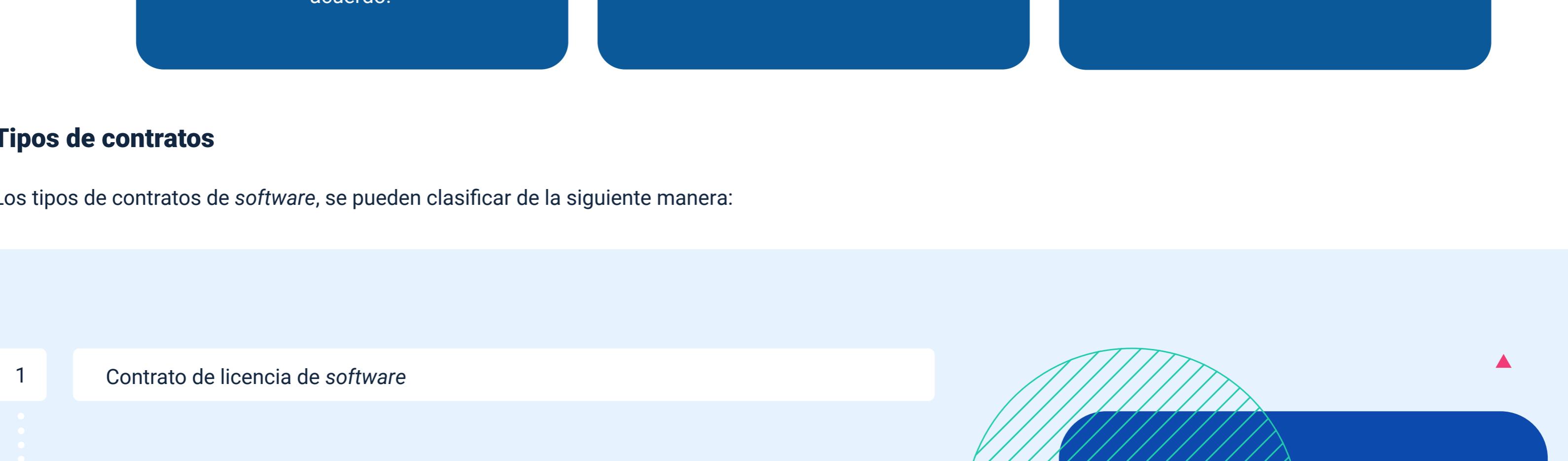
ESPACIO PARA VIDEO



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

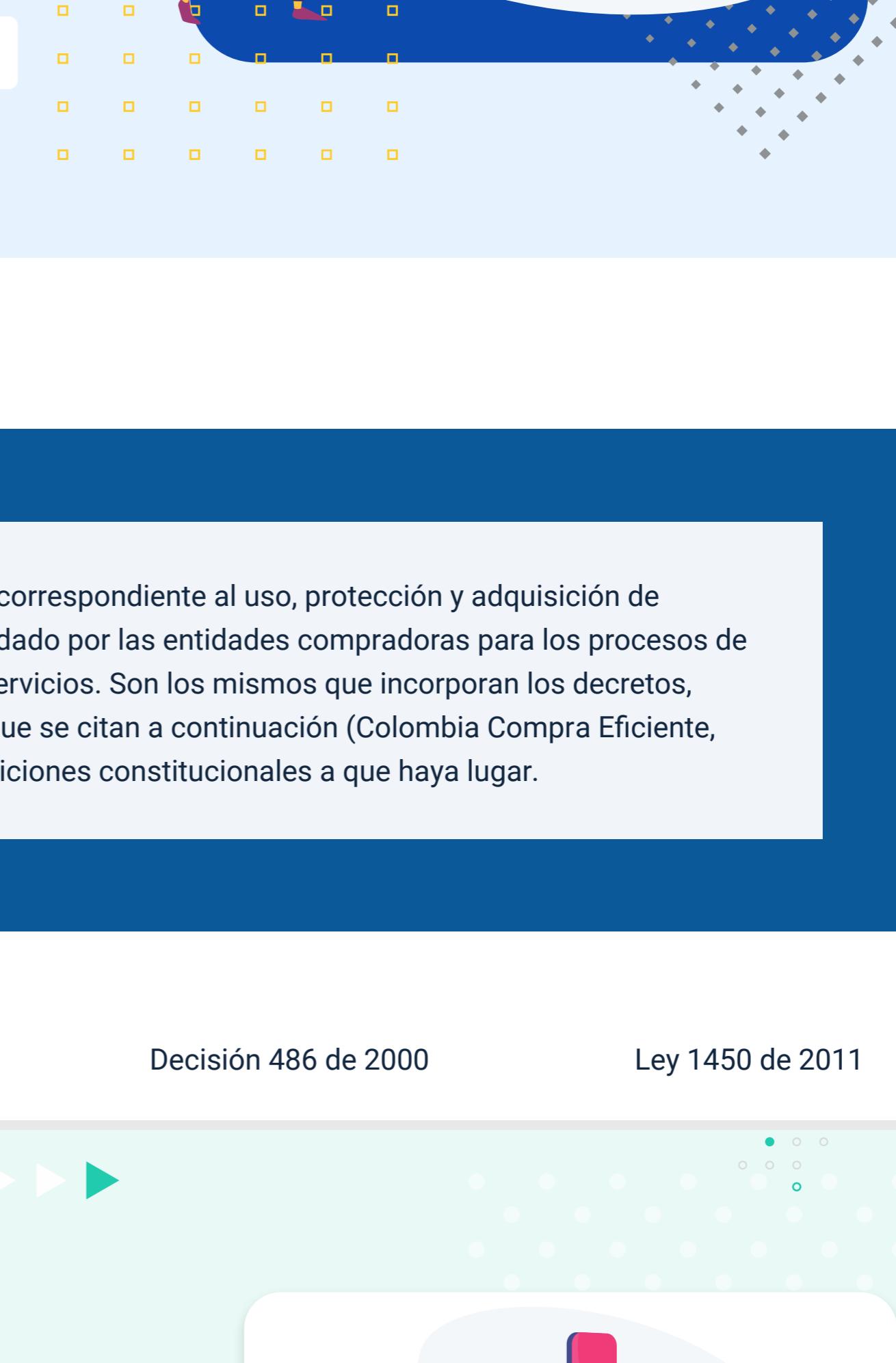


Tipos de contratos

Los tipos de contratos de software, se pueden clasificar de la siguiente manera:

- 1 Contrato 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 en la verificación del correcto funcionamiento.
- 4 Contrato de Software As A Service (contrato SAAS)
- 5 Contrato de distribución de software



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

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 asociado 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.

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; es ahí donde 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 estos 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 software no excluye otras formas de protección", reiterando en la protección jurídica que puede predicarse de otros componentes, relacionados con el software.



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.



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 ordenadores 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 contraría acuerdos internacionales, como el Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual, relacionados con el comercio y que establece, en el Artículo 27, como materia patentable, lo siguiente:



Ley 1450 de 2011

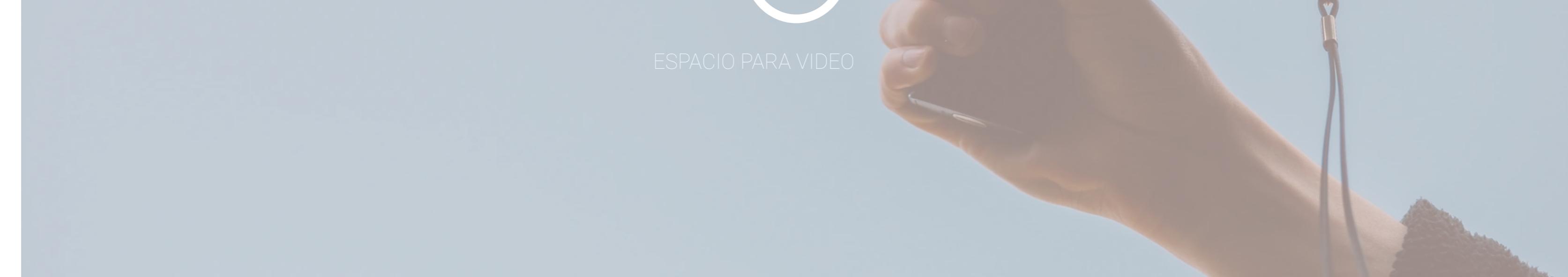
En relación con la naturaleza de los contratos de cesión de derechos patrimoniales de autor, el Artículo 30 de la Ley 1450 de 2011, establece que:

Los actos o contratos por los cuales se transfieren, parcial o totalmente, los derechos patrimoniales de autor o conexos, deberán constar por escrito como condición de validez. Todo acto por el cual se enajene, transfera, cambie o limite el dominio sobre el derecho de autor, o los derechos conexos, así como cualquier otro acto o contrato que implique exclusividad, deberá ser inscrito en el Registro Nacional del Derecho de Autor, para efectos de publicidad y oponibilidad ante terceros. Será inexistente toda estipulación, en virtud de la cual el autor transfera, de modo general o indeterminable, la producción futura, o se obligue a restringir su producción intelectual o a no producir.



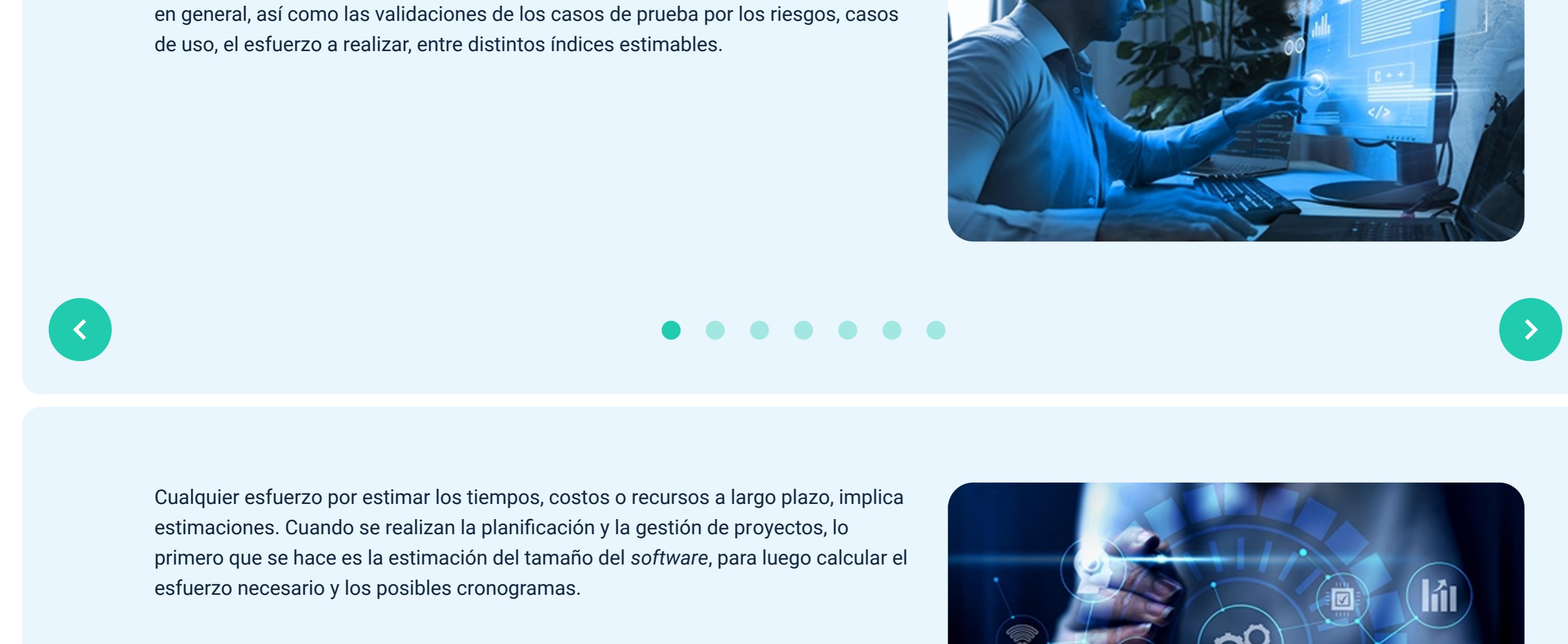
Cálculo de costos de software

En el siguiente video, se muestran aspectos claves para el **cálculo de costos de software**. Visualícelo atentamente:

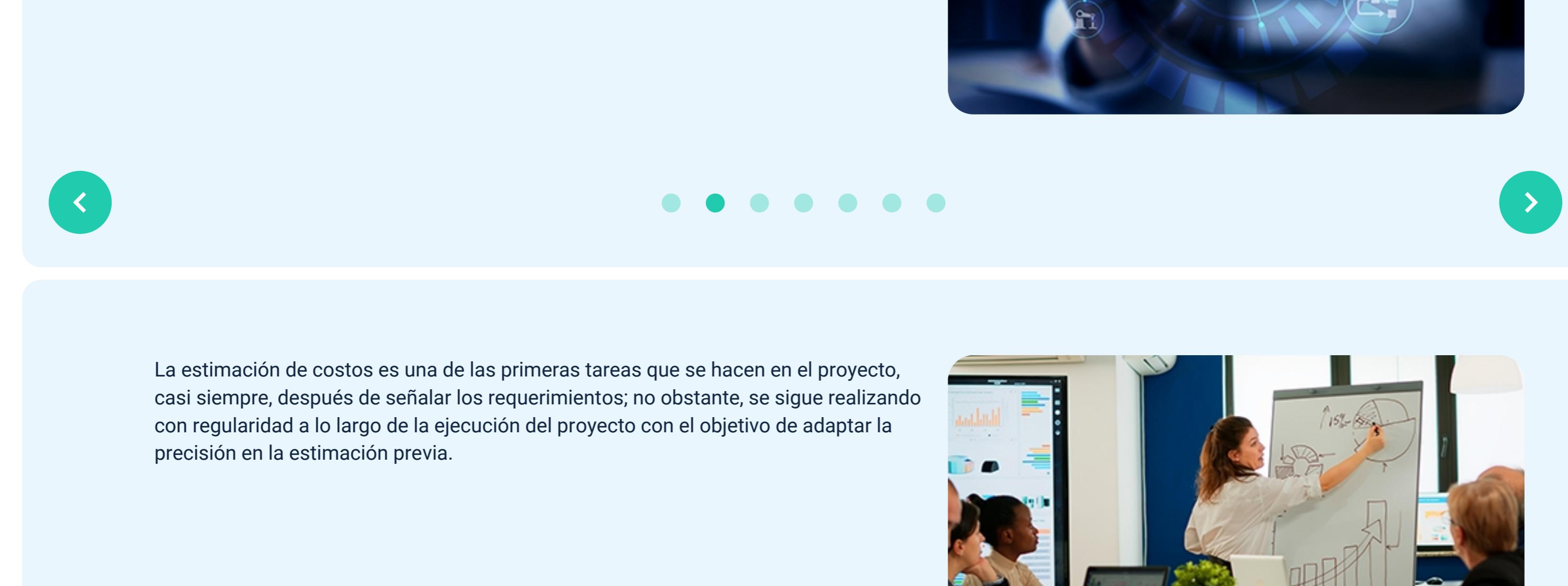


2 Técnicas de estimación de proyectos de software

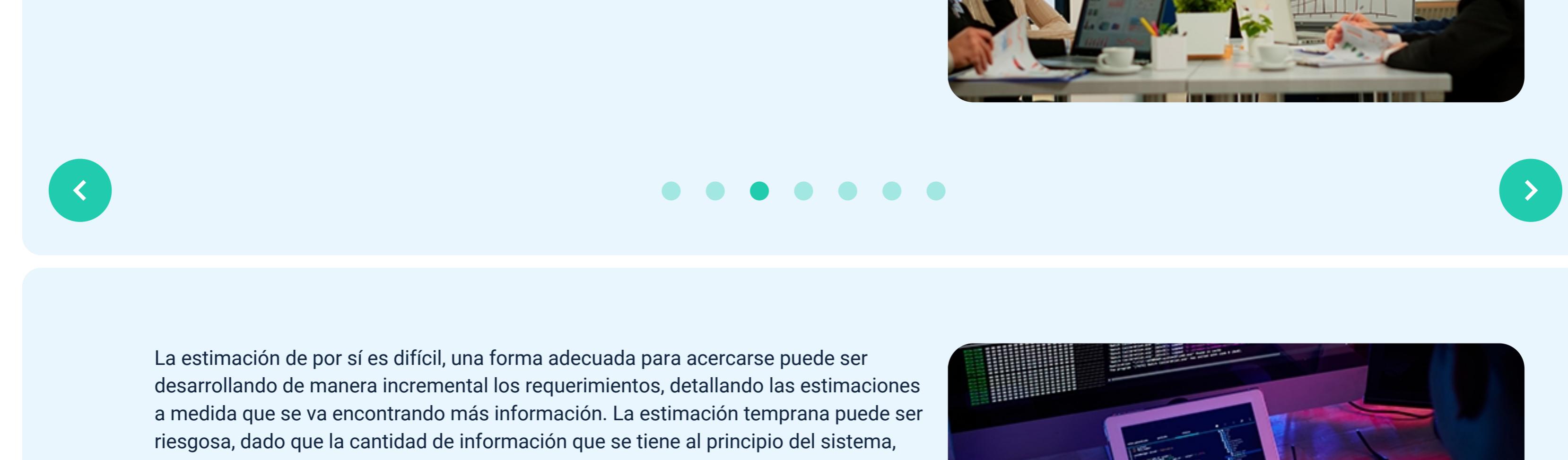
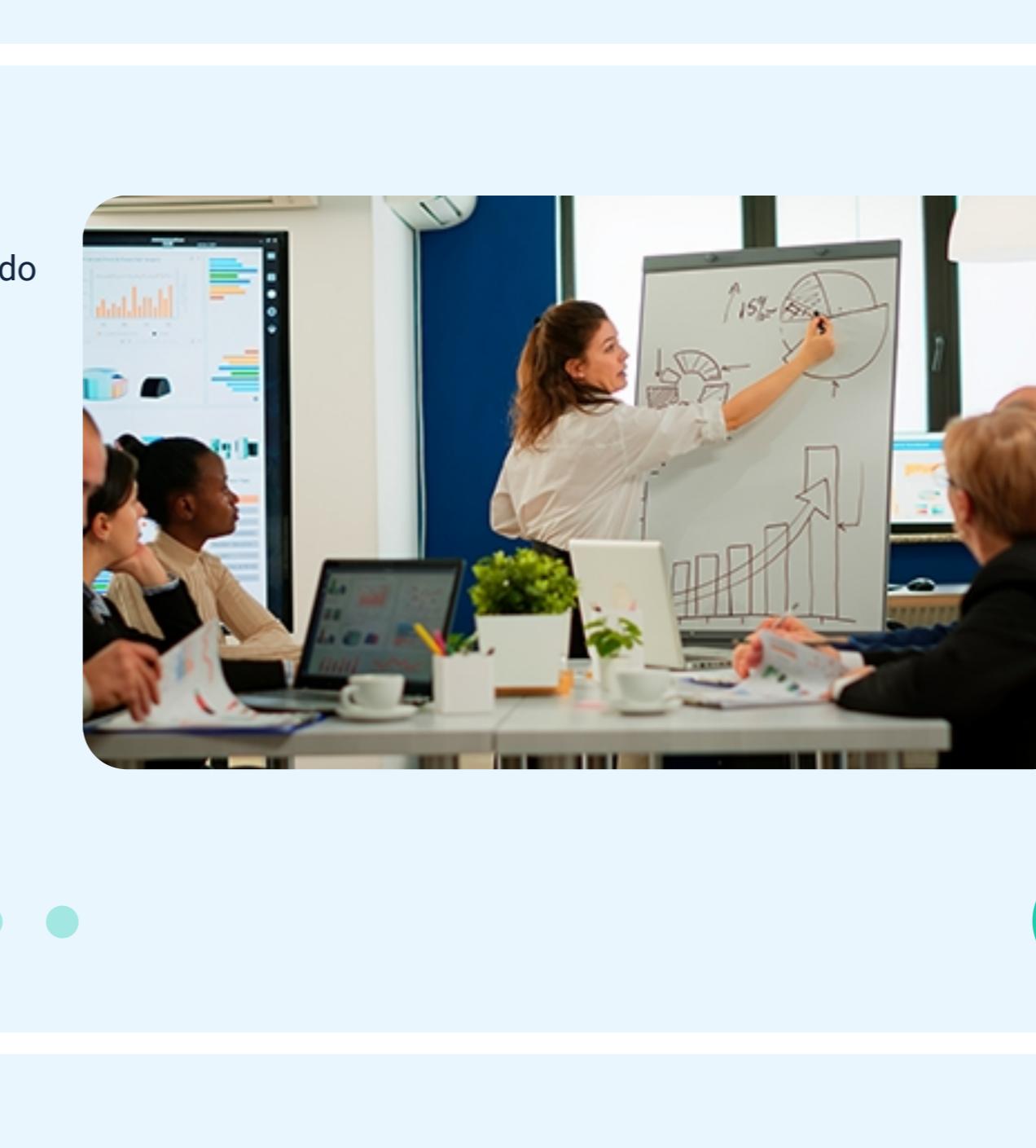
El primer elemento de un proyecto de software, es la planificación. En esta etapa del proceso, una de las tareas con mayor valor es la estimación, tanto de recursos, esfuerzos, costos y tiempos para el buen desarrollo del proyecto.



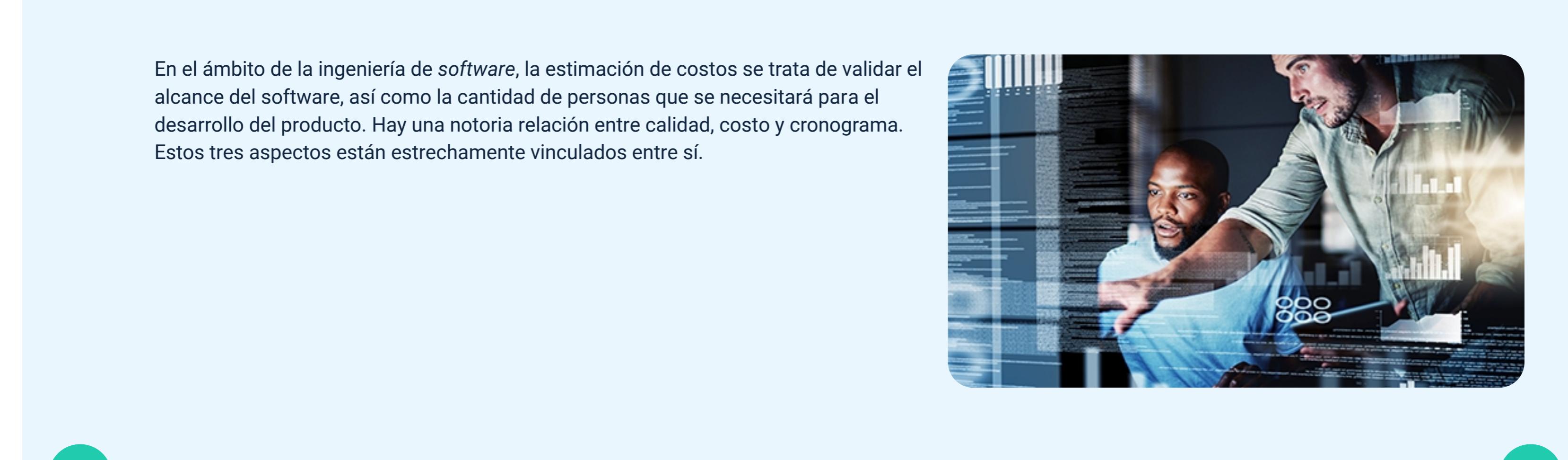
Evaluar los elementos que tienen relación con un proyecto (cronograma, esfuerzo, costo, personal, etc.), requiere conocer su alcance para estimar las posibles acciones, contrastar opciones y obtener cálculos de los costos, antes de decidirse por un enfoque determinado.



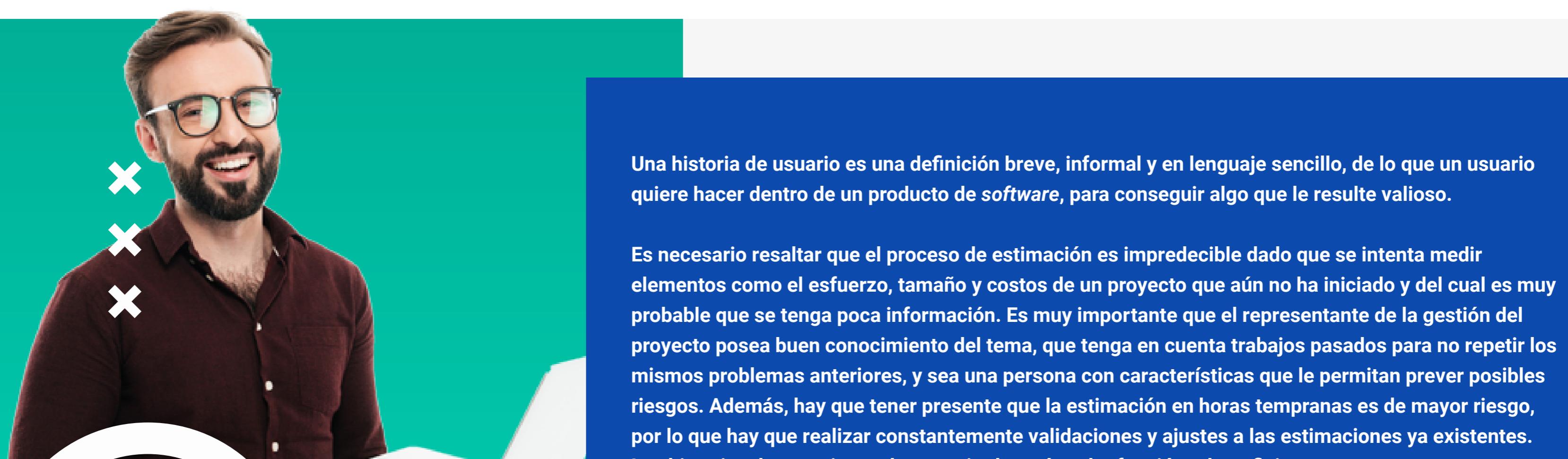
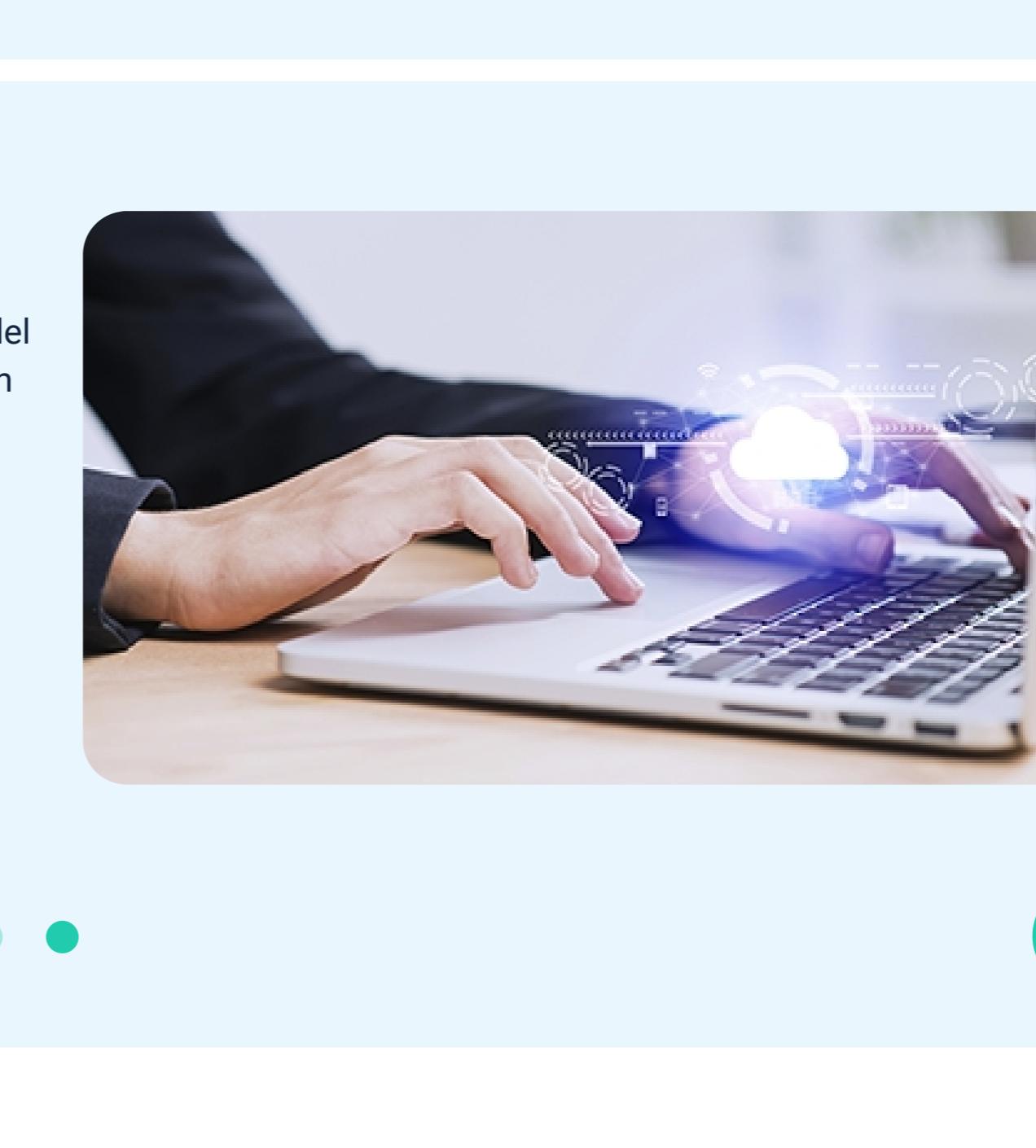
Dentro del entorno de un proyecto de software, existen diferentes factores a estimar; por ejemplo, el cálculo del tiempo de ejecución de cada una de las actividades a realizar dentro del proyecto (análisis, gestión, desarrollo, pruebas, etc.) o del proyecto en general, así como las validaciones de los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



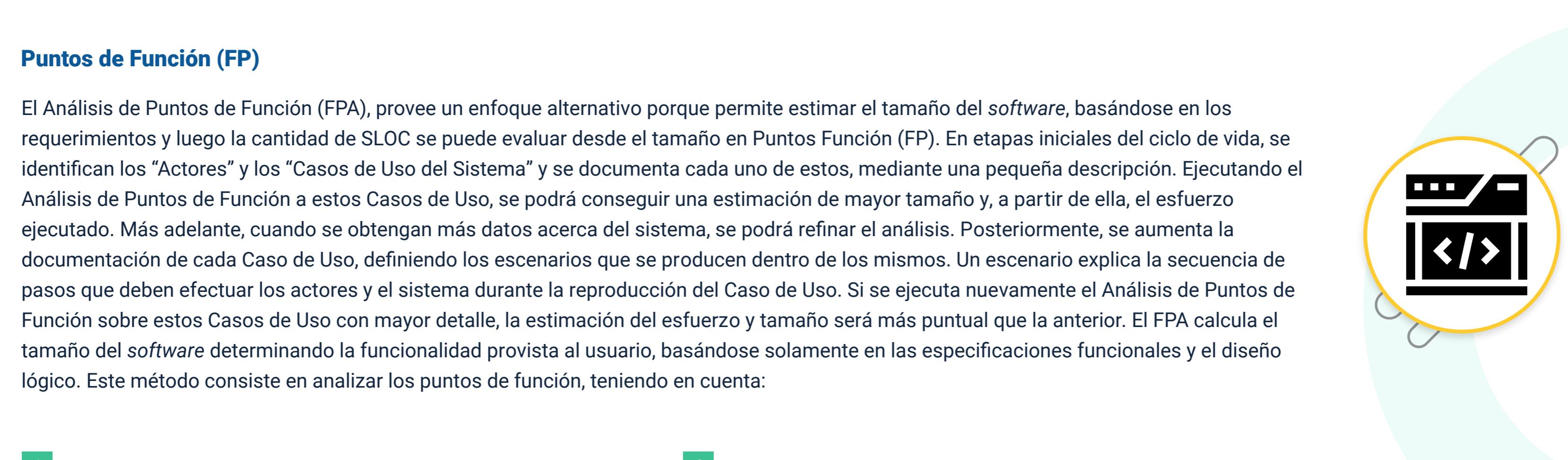
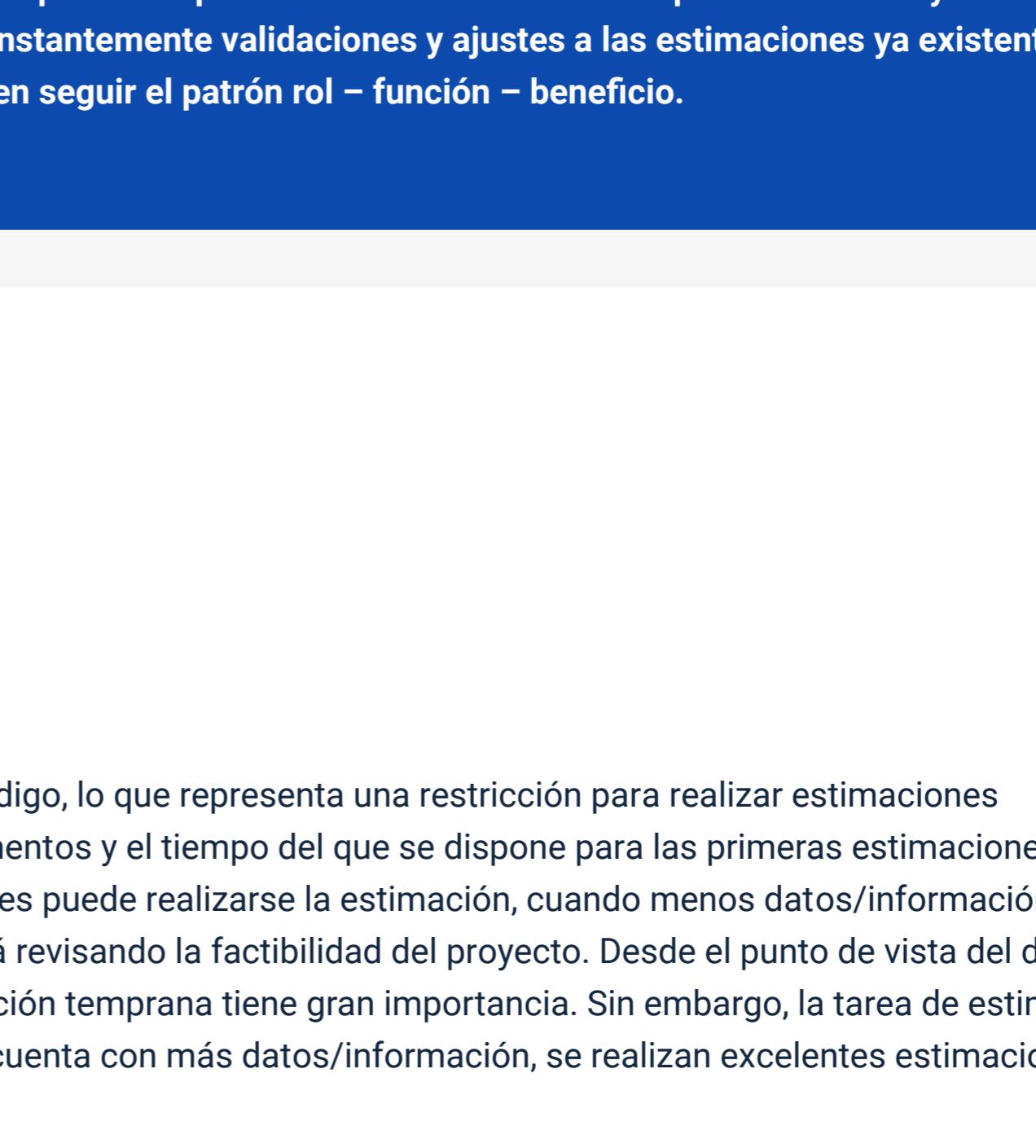
Cualquier esfuerzo por estimar los tiempos, costos o recursos a largo plazo, implica estimaciones. Cuando se realizan la planificación y la gestión de proyectos, lo primero que se hace es la estimación del tamaño del software, para luego calcular el esfuerzo necesario y los posibles cronogramas.



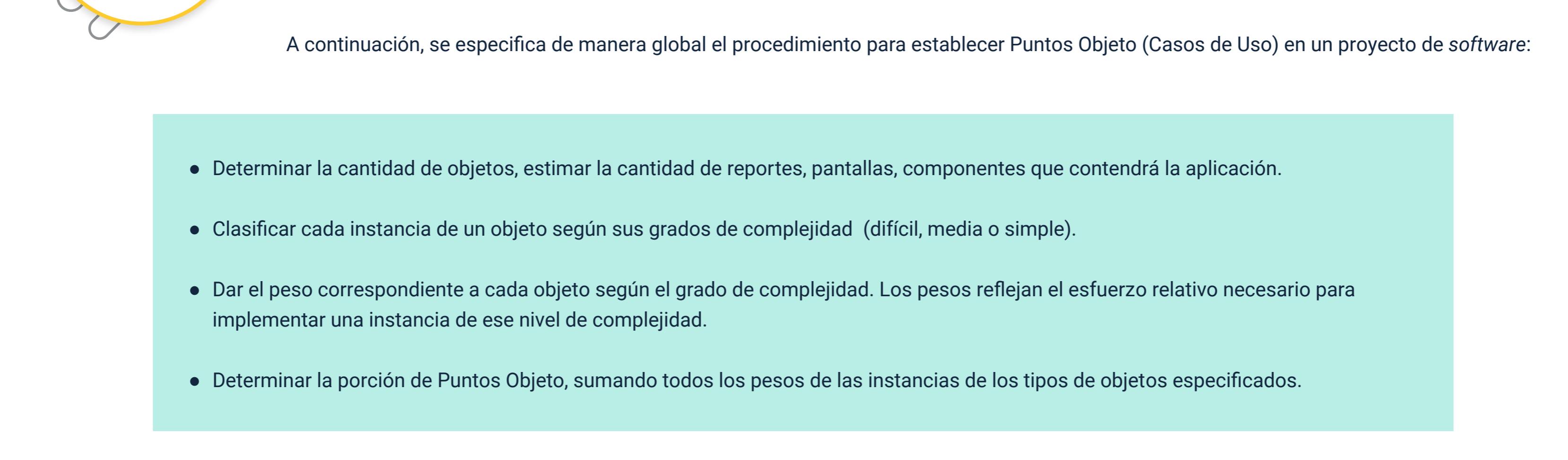
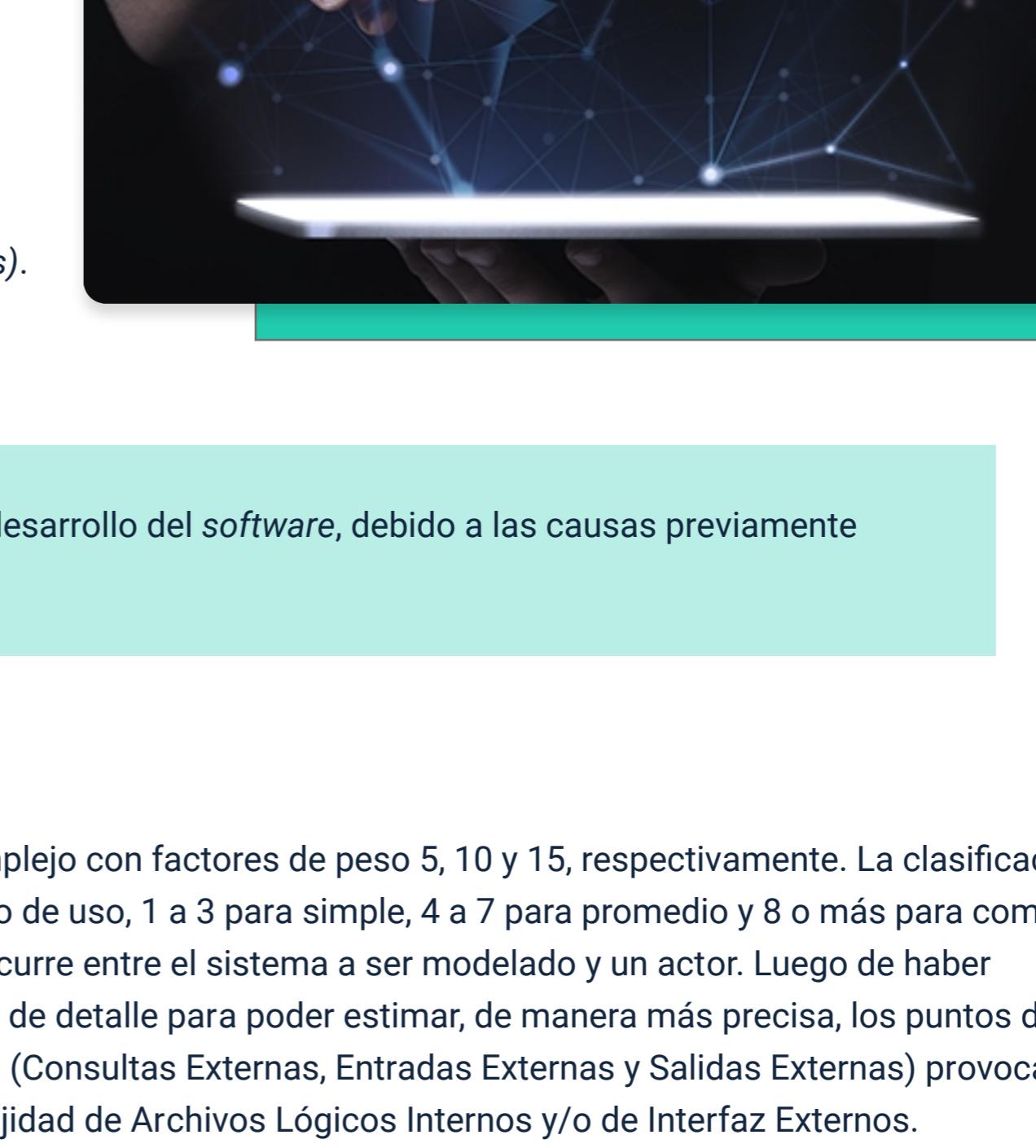
La estimación de costos es una de las primeras tareas que se hacen en el proyecto, casi siempre, después de señalar los requerimientos; no obstante, se sigue realizando con regularidad a lo largo de la ejecución del proyecto con el objetivo de adaptarla a la precisión en la estimación previa.



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.



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.



Técnicas de estimación del tamaño del software

Hay diferentes métodos para evaluar el tamaño del software, como por ejemplo:

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, debido a que la información que se obtiene en esos momentos y el tiempo del que se dispone para las primeras estimaciones del proyecto, son normalmente escasos. De todas las etapas en las cuales puede realizarse la estimación, cuando menos datos/información se tiene para realizarla, 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 se realiza una sola vez en el proyecto, sino que, a medida que se cuenta con más datos/información, se realizan excelentes estimaciones que permiten una mejor planificación de lo que falta del proyecto.



Puntos de Función (FP)

El Análisis de Puntos de Función (FPA), provee un enfoque alternativo porque permite estimar el tamaño del software, basándose en los requerimientos y luego la cantidad de SLOC se puede evaluar desde el tamaño en Puntos Función (FP). En etapas iniciales del ciclo de vida, se identifican los 'Actores' y los 'Casos de Uso del Sistema' y se documenta cada uno de estos, mediante una pequeña descripción. Ejecutando el Análisis de Puntos de Función a estos Casos de Uso, se podrá conseguir una estimación de mayor tamaño y, a partir de ella, el esfuerzo ejecutado. Más adelante, cuando se obtengan más datos acerca del sistema, se podrá refinar el análisis. Posteriormente, se aumenta la documentación de cada Caso de Uso, definiendo los escenarios que se producen dentro de los mismos. Un escenario explica la secuencia de pasos que deben efectuar los actores y el sistema durante la reproducción del Caso de Uso. Si se ejecuta nuevamente el Análisis de Puntos de Función sobre estos Casos de Uso con mayor detalle, la estimación del esfuerzo y tamaño será 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:



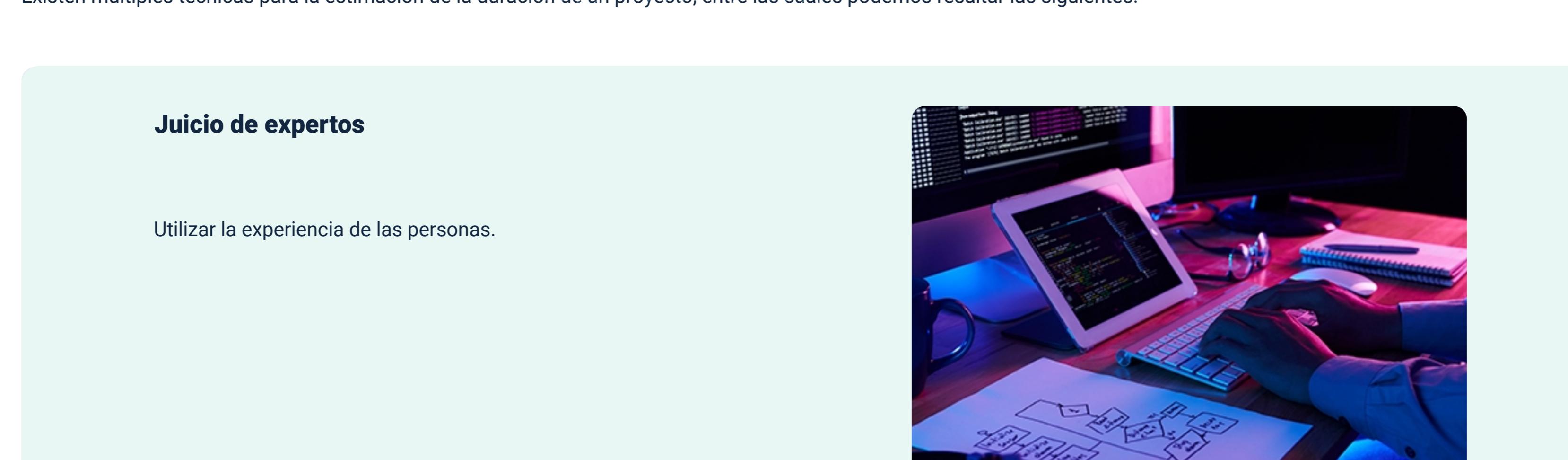
1 Entradas al sistema.
2 Salidas del sistema.
3 Consultas.
4 Grupos de datos lógicos del sistema.
5 Grupos de datos lógicos que no pertenecen al sistema, pero que el sistema utiliza.

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:

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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 de uso, 1 a 3 para simple, 4 a 7 para promedio y 8 o más para complejos. Cabe aclarar que un 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) provocadas por las secuencias que forman un escenario, es posible definir la complejidad de Archivos Lógicos Internos y/o de Interfaz Externos.

A continuación, se especifica de manera global el procedimiento para establecer Puntos Objeto (Casos de Uso) en un proyecto de software:

- Determinar la cantidad de objetos, estimar la cantidad de reportes, pantallas, componentes que contendrá la aplicación.
- Clasificar cada instancia de un objeto según sus grados de complejidad (difícil, media o simple).
- Dar el peso correspondiente a cada objeto según el grado de complejidad. Los pesos reflejan el esfuerzo relativo necesario para implementar una instancia de ese nivel de complejidad.
- Determinar la porción de Puntos Objeto, sumando todos los pesos de las instancias de los tipos de objetos especificados.

Se debe resaltar la característica de que los casos de uso no tienen un tamaño definido, por lo que un método basado en Casos de Uso debería tener en cuenta escenarios del mismo o el número de transacciones para determinar su peso. A medida que se van concluyendo las especificaciones de los Casos de Uso, se pueden ir depurando las estimaciones de los puntos de función.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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 FP (Netherlands Software Metrics Users Association Function Point Analysis).

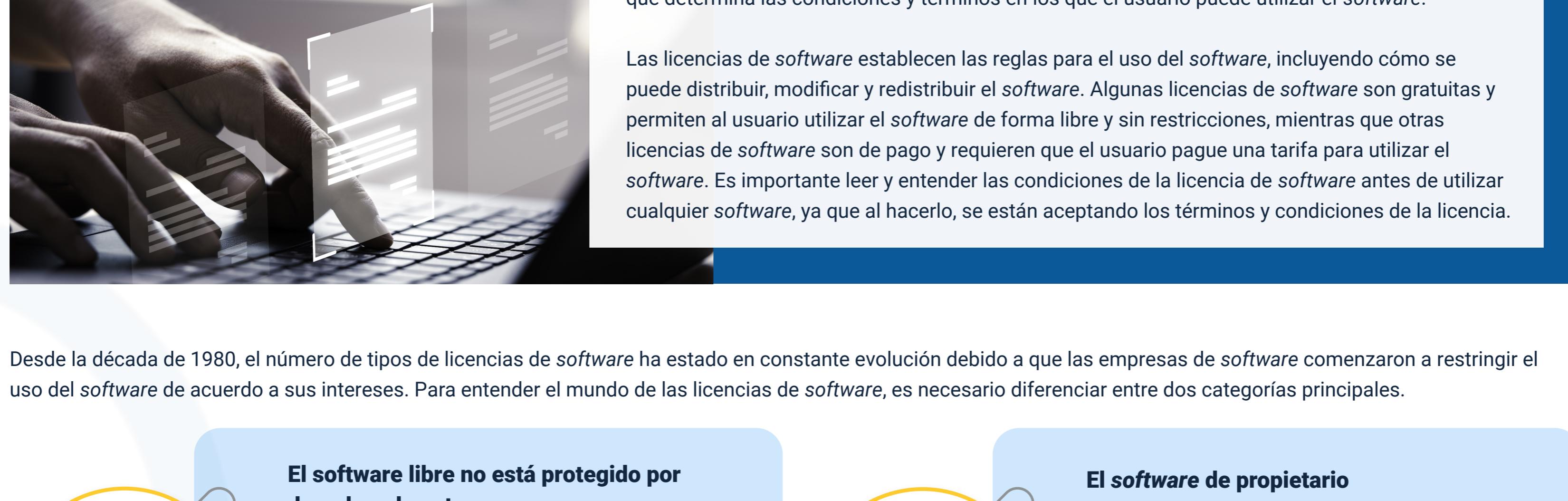
Este método también posee dificultades en su aplicación en etapas tempranas de desarrollo del software, debido a las causas previamente mencionadas de la escasez de información del sistema en esos momentos.

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

$FP = UFP \times TCF$ donde:
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.</

3 Licenciamiento



Las licencias de software son un acuerdo entre el autor o propietario del software y el usuario final que determina las condiciones y términos en los que el usuario puede utilizar el software.

Las licencias de software establecen las reglas para el uso del software, incluyendo cómo se puede distribuir, modificar y redistribuir el software. Algunas licencias de software son gratuitas y permiten al usuario utilizar el software de forma libre y sin restricciones, mientras que otras licencias de software son de pago y requieren que el usuario pague una tarifa para utilizar el software. Es importante leer y entender las condiciones de la licencia de software antes de utilizar cualquier software, ya que al hacerlo, se están aceptando los términos y condiciones de la licencia.

Desde la década de 1980, el número de tipos de licencias de software ha estado en constante evolución debido a que las empresas de software comenzaron a restringir el uso del software de acuerdo a sus intereses. Para entender el mundo de las licencias de software, es necesario diferenciar entre dos categorías principales.

El software libre no está protegido por derechos de autor

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 las licencias de software libre y de propiedad, existen otras opciones de licencias, como el software gratuito o de dominio público, que no está protegido por derechos de autor y se puede utilizar libremente. Sin embargo, el autor del software puede imponer ciertas restricciones en cuanto a la distribución y uso del software y de cualquier trabajo generado a partir del mismo. También existe la licencia de software semilibre, que pertenece a un dueño pero se permite la distribución, uso, copia y modificación del software por parte de individuos sin fines de lucro. Es importante leer y comprender las condiciones de cada tipo de licencia de software antes de utilizar el software.

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: permiten a los usuarios utilizar el software de acuerdo a sus necesidades, incluyendo ejecutar el programa, estudiar su funcionamiento, modificarlo para adaptarlo a sus necesidades, copiar y redistribuir el software y mejorarlo y compartir esas mejoras con la comunidad. Estas licencias ofrecen una gran libertad y flexibilidad para el uso del software y son una buena opción para aquellos que deseen colaborar en el desarrollo y mejora del software.

Licencias de software libre no protegido con copyright: son aquellas que permiten la redistribución y modificación del software, pero también permiten añadir restricciones adicionales a algunas copias o versiones modificadas del software. Esto significa que algunas copias o versiones modificadas del software pueden no ser completamente libres y pueden requerir un software de pago para poder ejecutar ciertos elementos. Estas licencias pueden ser útiles para aquellos que deseen distribuir software libre, pero también desean restringir ciertas formas de uso o modificación del software.

Licencias de software libre protegidas con copyright: son aquellas que no permiten a quienes modifican o redistribuyen el software agregar restricciones adicionales. Esto significa que cualquier copia o modificación del software debe ser libremente utilizable y distribuible. El copyright es una práctica legal y común que consiste en utilizar los derechos de autor para promover el uso y la distribución libres de una obra y exigir las mismas libertades al distribuir copias y derivados de la obra. Esto se utiliza para garantizar que el software libre siga siendo libre a medida que se modifica y se redistribuye.

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 de 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 copyright.

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 aquel 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.

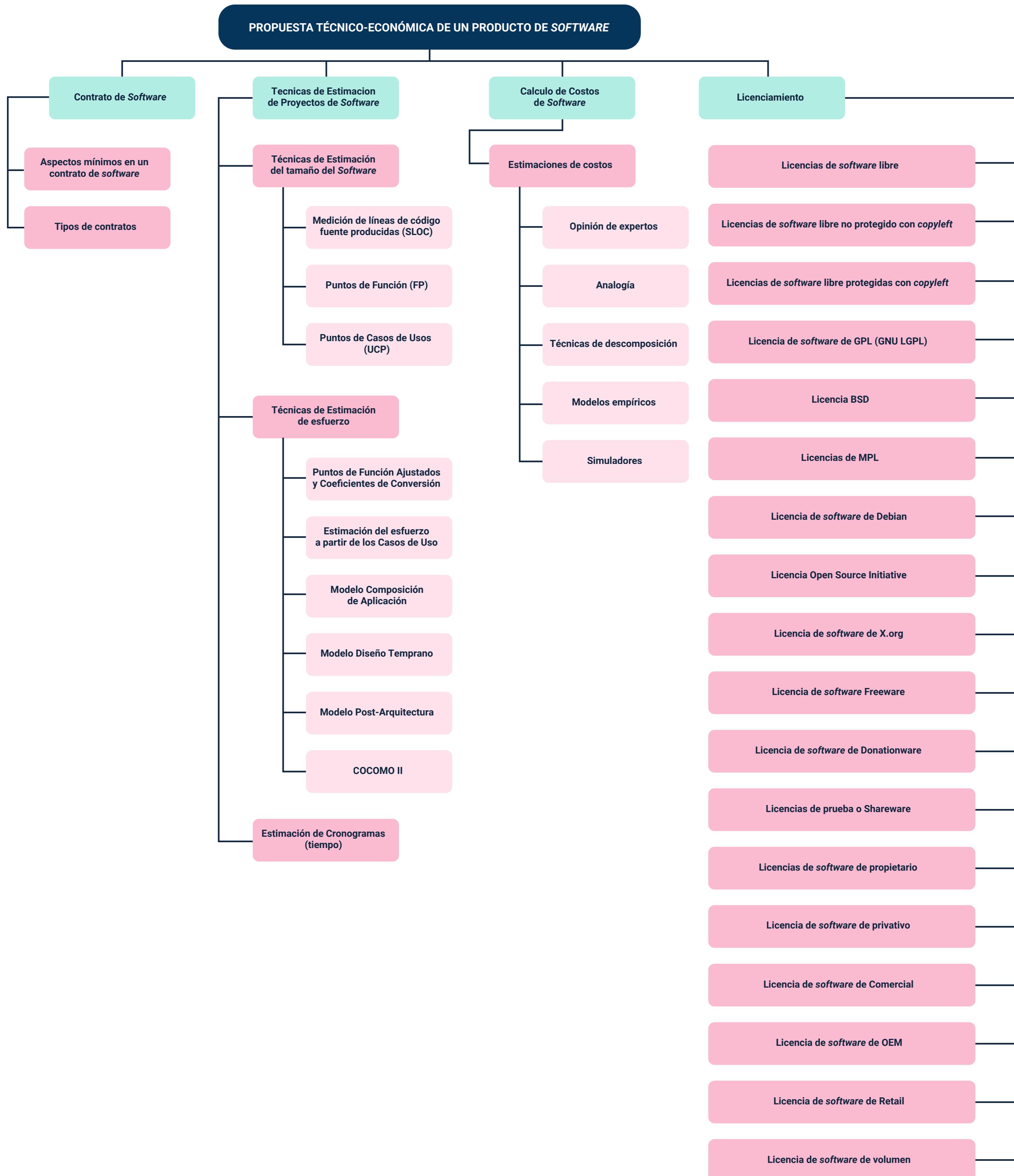
Licencia de software de volumen: su destino son los grandes usuarios (empresas) y el contrato de adquisición estipula un determinado número de equipos que pueden utilizar el mismo código.

Desarrollo de aplicaciones web Full stack

Síntesis: Planeación y recolección de información según necesidades.



El siguiente mapa integra los criterios y especificidades de los conocimientos expuestos en el presente componente formativo:





Portada actividad

800 x 800



Portada actividad

800 x 800



Imagen acompañamiento Actividad

600 x 600



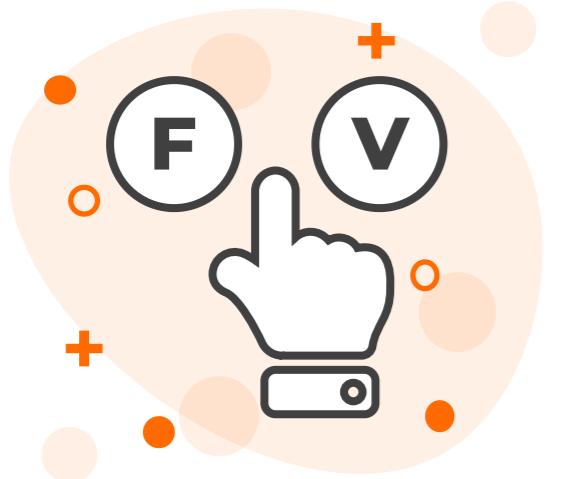
Imagen Resultado

900*600

i

Actividad didáctica

Verdadero y falso



Verdadero y falso

Apreciado aprendiz, a continuación, encontrará una serie de preguntas que deberá resolver, con el objetivo de evaluar la aprehensión de los conocimientos expuestos en este componente formativo.

Realizar 