

SEMINARIO DE CAPACITACIÓN PARA DESARROLLADORES DE SOFTWARE
24/02/2025

Parte II

La gestion y la explotación del software

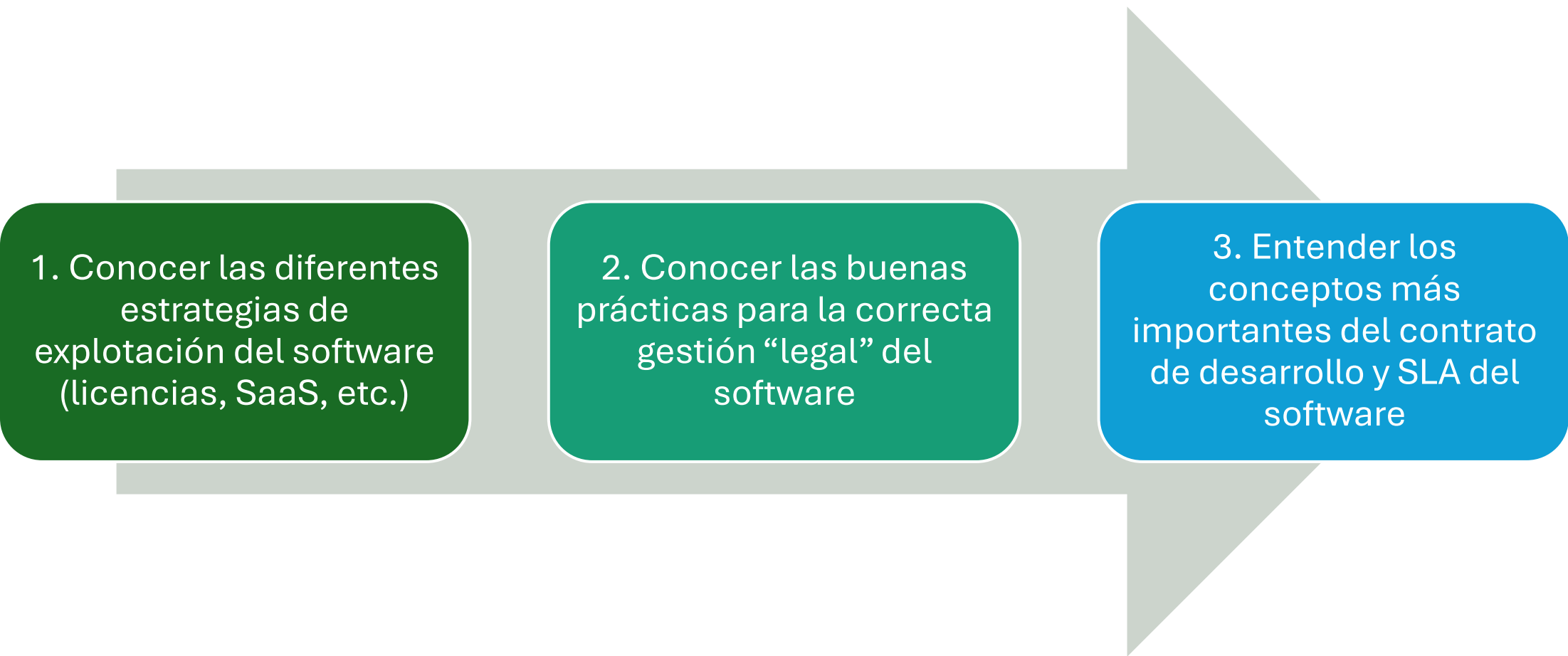
Malcolm Bain

Abogado, Across Legal, Barcelona

Recordatorio... ¿Qué hemos dejado para febrero?

- *Estrategias de licencia y comercialización*
- *Buenas practices de gestion de PI del software*
- *Cómo cumplir y exigir el cumplimiento de los derechos*
- *Defensa de los derechos y resolución de conflictos*
- *El contrato de desarrollo*

Objetivos de aprendizaje para hoy



1. Conocer las diferentes estrategias de explotación del software (licencias, SaaS, etc.)

2. Conocer las buenas prácticas para la correcta gestión “legal” del software

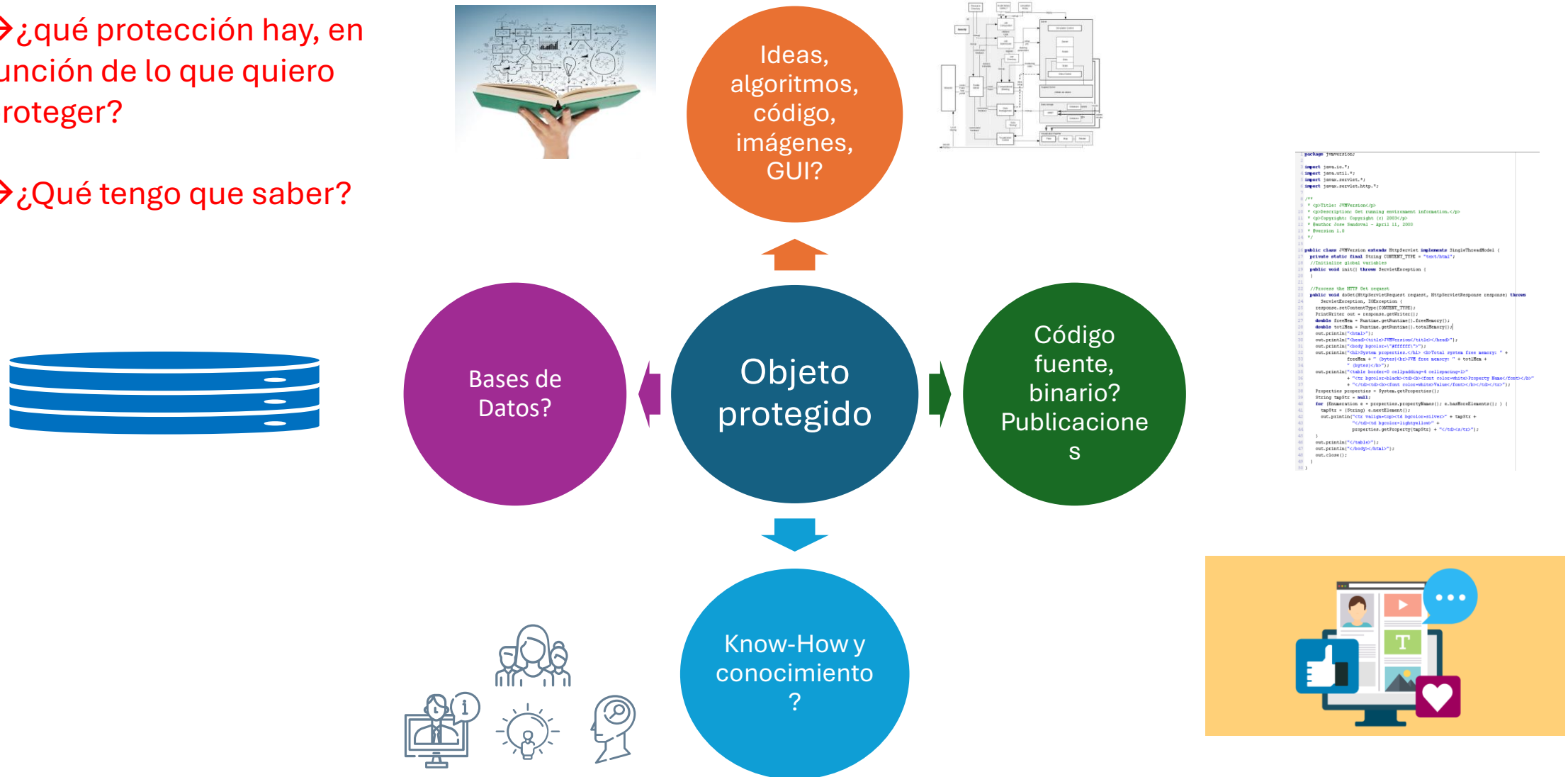
3. Entender los conceptos más importantes del contrato de desarrollo y SLA del software

Repaso en 5 *slides*

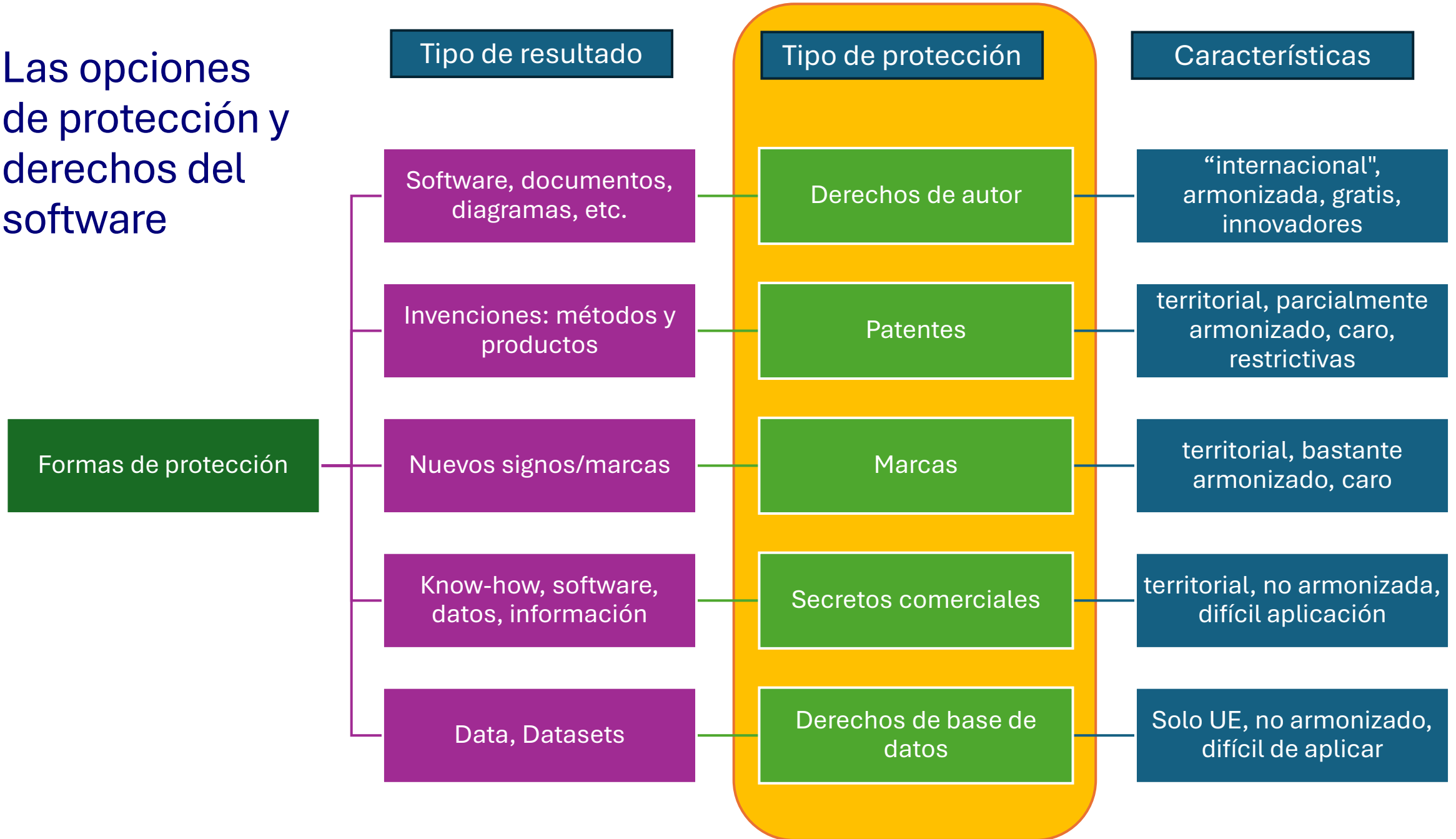
Protección legal en función de la naturaleza y tipología de la tecnología

→ ¿qué protección hay, en función de lo que quiero proteger?

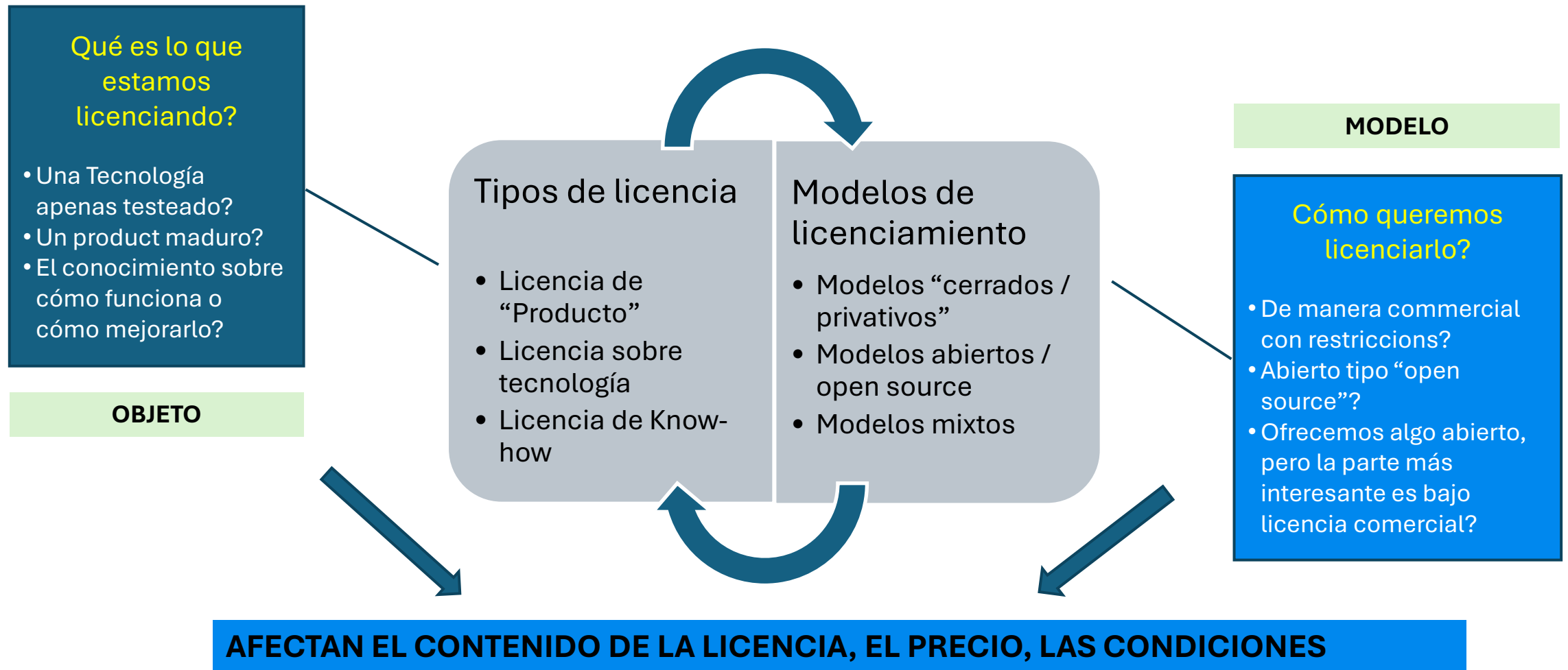
→ ¿Qué tengo que saber?



Las opciones de protección y derechos del software



Conceptos de licencia: el objeto licenciado, y el modelo de licencia



Tipos de licencia : Propietarias, libres, académicas

Licencia de software tradicional/propietario:

- Distribución binaria
- Proporciona un derecho a (instalar y) "usar" el software
- Restringe ciertos usos - prohíbe las modificaciones, distribución,
- No permite el acceso al código fuente
- Limita las garantías y responsabilidades
- Pago de \$\$ licencias en función de determinados criterios (número de usuarios, datos procesados, CPUs, etc.)
- "Código cerrado", por razones comerciales

Licencias tradicionales, cerradas

- Software comercial,
- Freeware y shareware

Licencia de software libre

- Distribución código fuente y binaria
- Proporciona todos los derechos de uso
- Permite modificar, distribuir,
- Limita las garantías y responsabilidades
- Código abierto para fomentar la colaboración

No hay ingresos por licencia

Modelos académicos

- Usos académicos no-comerciales

Modelos abiertos

- Open source, copyleft

Licencia "académica" – available source

- Distribución código fuente y binaria
- Proporciona todos los derechos de uso académicos y científico, salvo el uso comercial:
- Permite modificar, distribuir dentro de la comunidad científica
- Limita las garantías y responsabilidades
- Código abierto para fomentar la colaboración

NO hay ingresos por licencia

Un abanico de licencias open source

Licencias permisivas

Licencias	<ul style="list-style-type: none">• MIT, BSD, Apache, Perl, Python
Contenido	<ul style="list-style-type: none">• Todos derechos de uso• Sin condiciones• Publicidad, patentes, derº aplicable
Objetivos	<ul style="list-style-type: none">• Otorgar los derechos, sin condiciones• Proteger el autor / titular contra responsabilidades• Compatibles con otras licencias
Generaciones	<ul style="list-style-type: none">• Simples: BSD, MIT, etc• Complejas: Apache, AFL

Licencias “híbridas” - copyleft suave

Licencias	<ul style="list-style-type: none">• LGPL, MPL, OSL, Artistic, EUPL
Contenido	<ul style="list-style-type: none">• Licencia copyleft, efecto limitado al código originario + modificaciones (pero no a software enlazado)• Licencia de patente / defensiva
Objetivos	<ul style="list-style-type: none">• Mantener el “núcleo” FOSS + acceso CF• Permitir su integración en obras mayores
Generaciones	<ul style="list-style-type: none">• LGPLv2 (1991)• MPL (98) / OSL (2003) / LGPLv3 (2007)

Licencias copyleft

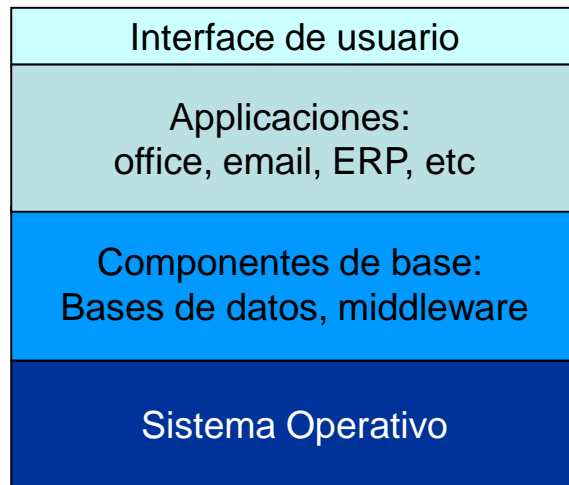
Licencias	<ul style="list-style-type: none">• GPL, AGPL
Contenido:	<ul style="list-style-type: none">• Copyleft: Si se distribuye el código o las obras derivadas y basadas en el software<ul style="list-style-type: none">• tienen que distribuirse bajo la misma licencia• hay que ofrecer acceso al código fuente
Objetivos:	<ul style="list-style-type: none">• Evitar privatizar el código• asegurar el acceso a nuevas versiones, construir comunidad
Generaciones	<ul style="list-style-type: none">• 1991: GPL2• 2007: GPL3 y AGPL3

Explotación del software:

Consideraciones previas

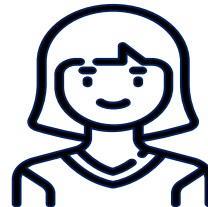
Factores esenciales para el modelo y la distribución de software

Modularidad: arquitectura del software

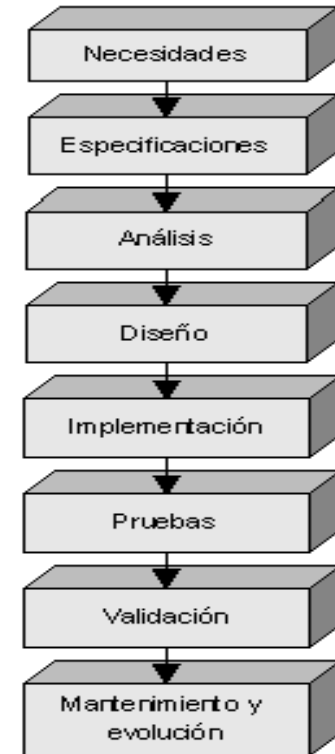


Qué estás distribuyendo?
Qué estás incluyendo en la licencia?

¿Cuáles son los factores que afectan la explotación del software?



Ciclo de vida del Software



Además del software,
¿qué necesidades tiene el cliente?

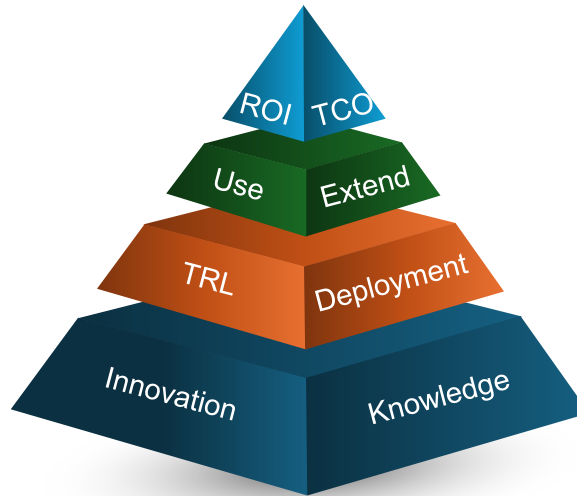
Variedad de factores que determinen la explotación del software

Factores tecnológicos

- Nivel de madurez
- Modo de despliegue
- Cumplimiento con estándares
- Niveles de seguridad
- Continuidad tecnológica

Objetivos propios / cliente

- Innovación digital
- Transferencia conocimiento
- Mejora de la capacidad técnica
- Ingresos



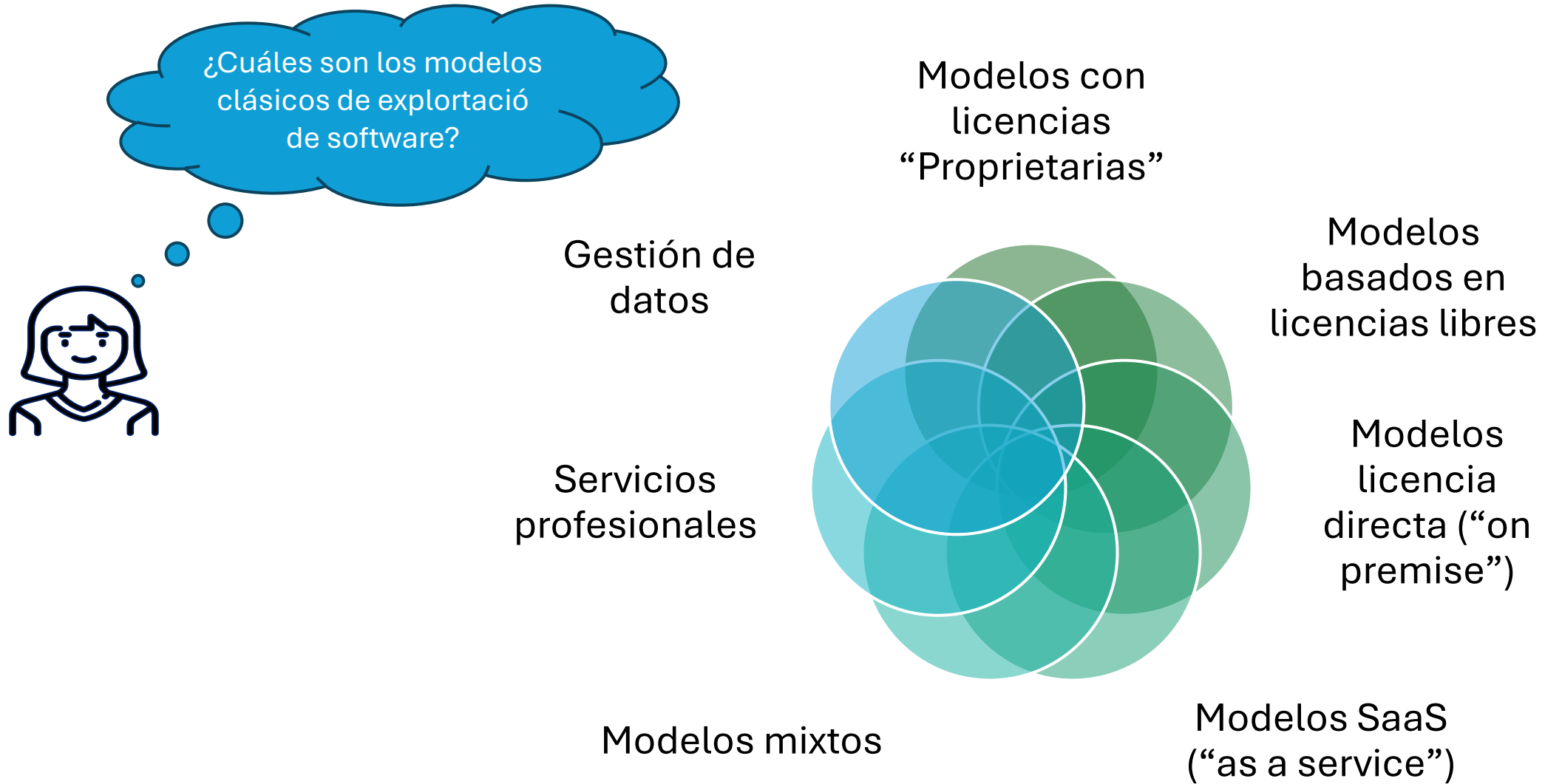
Factores económicos

- Modelo de ingresos
- Modelos de costos
- TCO / ROI

Factures de uso

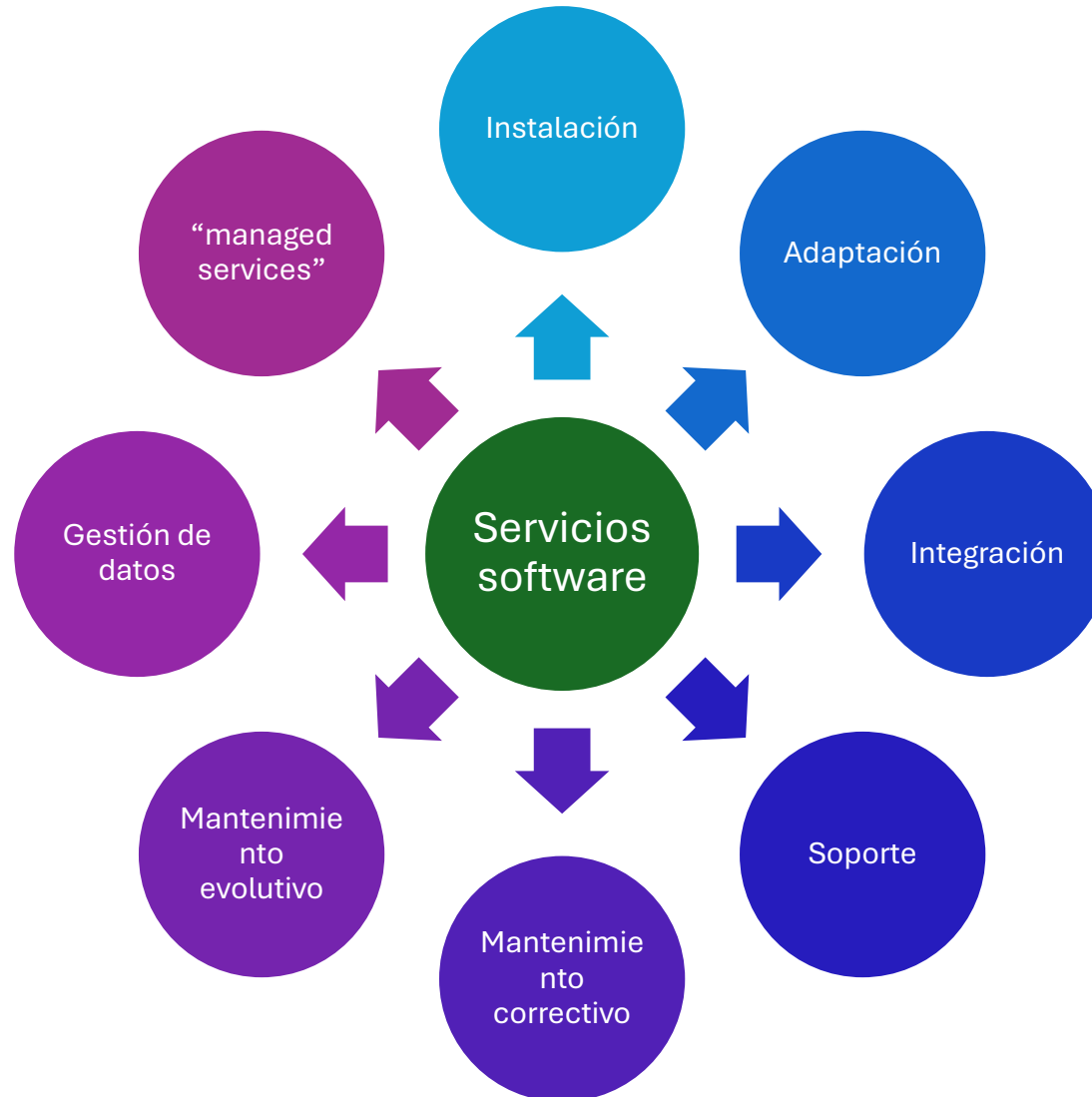
- On premise, SaaS, embebido
- Adaptación
- Redistribución
- Mejoras

Un gran abanico de modelos de licencia y explotación



Servicios técnicos: adaptación, soporte y mantenimiento

Características
<ul style="list-style-type: none">• Alcance segun descripción técnica• Tiempo acordado• Precio• Prestación de servicio diligente y professional• Niveles de servicio• Garantías de conformidad y corrección de errores

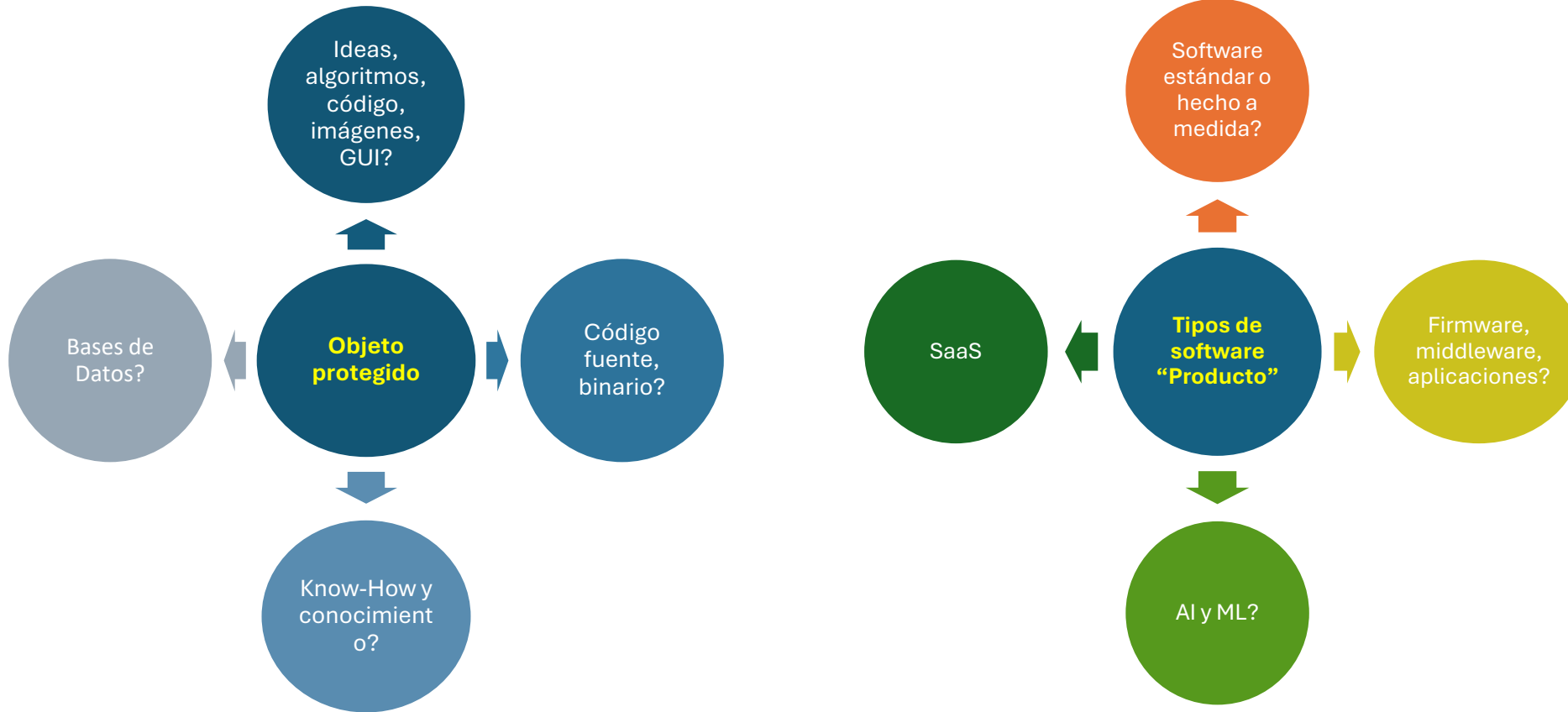
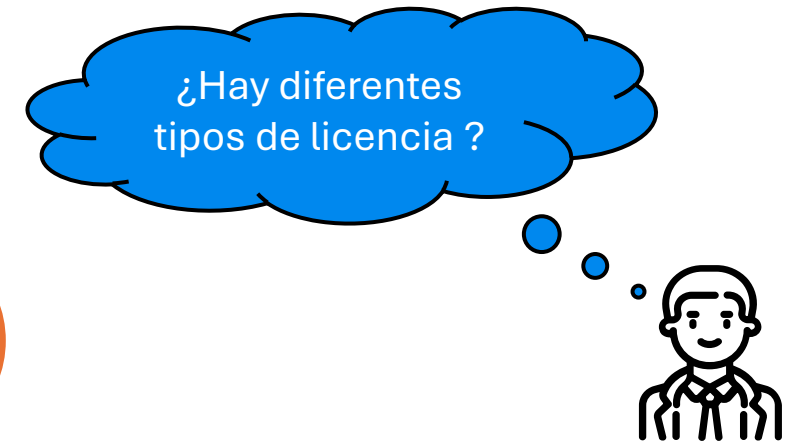


SLA - ANS
<ul style="list-style-type: none">• Garantías de nivel de servicio• Disponibilidad de plataforma o servicio• Servicio de soporte – niveles<ul style="list-style-type: none">• L1 – soporte al usuario• L2 - soporte técnicos• L3 - Corrección de errores• Tiempos de respuesta y resolución• Penalizaciones?

Explotación del software:

Estrategias de licencias

Tipos de licencia: context tecnológico, licenciatario, usos



EULA: la licencia de usuario final o consumidor (B2C)

B2B: la licencia profesional o "Enterprise"

OEM: la licencia para integrar en otras soluciones

SaaS / On premise

SDK/librerías

Modelos clásicos de explotación del software

On premise

- Licencia para instalación y uso local / interno
- Restricciones sobre copias y redistribución
- “EULA” (ofimática, etc.) – garantías consumidor
- “Enterprise” – garantías negociadas
- Reportes y auditoría
- Datos de cliente: en el cliente
- Servicios de soporte: remotos o en situ?

Cloud / SaaS

- Licencia de acceso por usuario, o via API
- Restricciones de uso / conexión
- Uso controlado/monitorizado por el proveedor
- Datos de cliente: en la nube (privacidad, seguridad, espacio alojamiento)
- Multi-tenant?
- Migración de cliente al final (portabilidad)

Binary / Source Code

- Licencia de instalación y uso de un binario: solamente para instalar y usar
- Licencia de acceso y uso del código fuente: Para modificar, o para garantía de continuidad
- Confidencialidad del código fuente (riesgos)
- Riesgos de re-ingenería del cliente?
- Escrow: depósito del código fuente con acceso condicional

Aspectos comunes: Garantías de conformidad y funcionamiento. Garantías de IPR. Soporte y “SLAs”. Precio y pago

Otros modelos de distribución

OEM

- Licencia para integración o “bundling” en otro producto y ventas
- Con o sin código fuente
- Restricciones sobre cantidades y redistribución “tal cual”
- Precio: único? Por venta?
- Reportes de ventas y derecho de auditoría
- Servicios de soporte: SLA tiene que corresponder al SLA del cliente

SDK/API

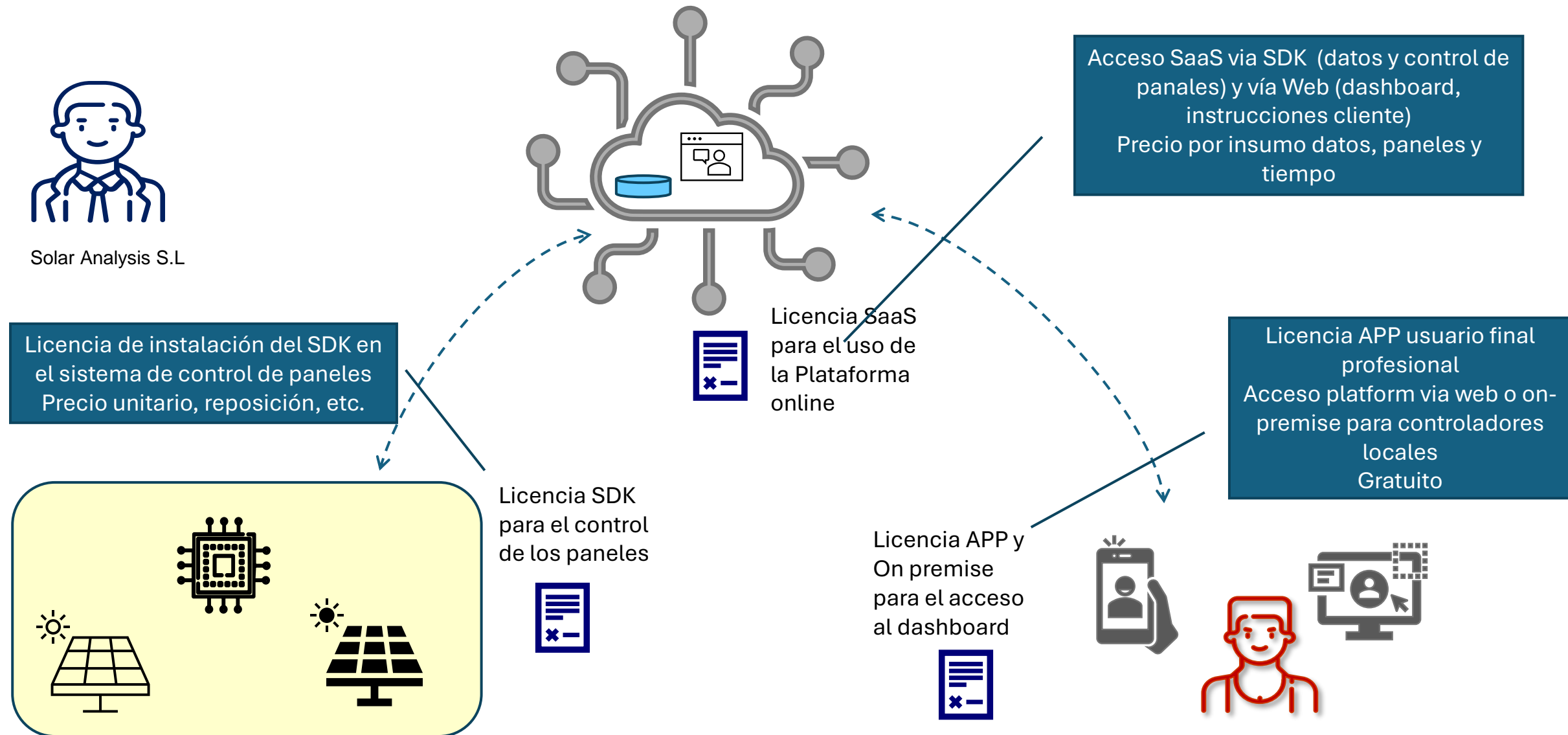
- Licencia uso de librería como SDK o acceso técnico API
- Con o sin código fuente
- Restricciones sobre cantidades y redistribución “tal cual”
- API: conectividad Cloud
- Precio: único? Por venta?
- Reportes de ventas y derecho de auditoría
- Servicios de soporte: SLA tiene que corresponder al SLA del cliente

Software a medida

- Contrato “llaves en mano”
- Servicios de desarrollo y entrega del software
- Especificaciones, desarrollo, despliegue, migración,
- Formación del usuario final
- IPR: cesión de todos los derechos?
- Entrega de código fuente
- Gestión de componentes existentes (software libre, software propio)

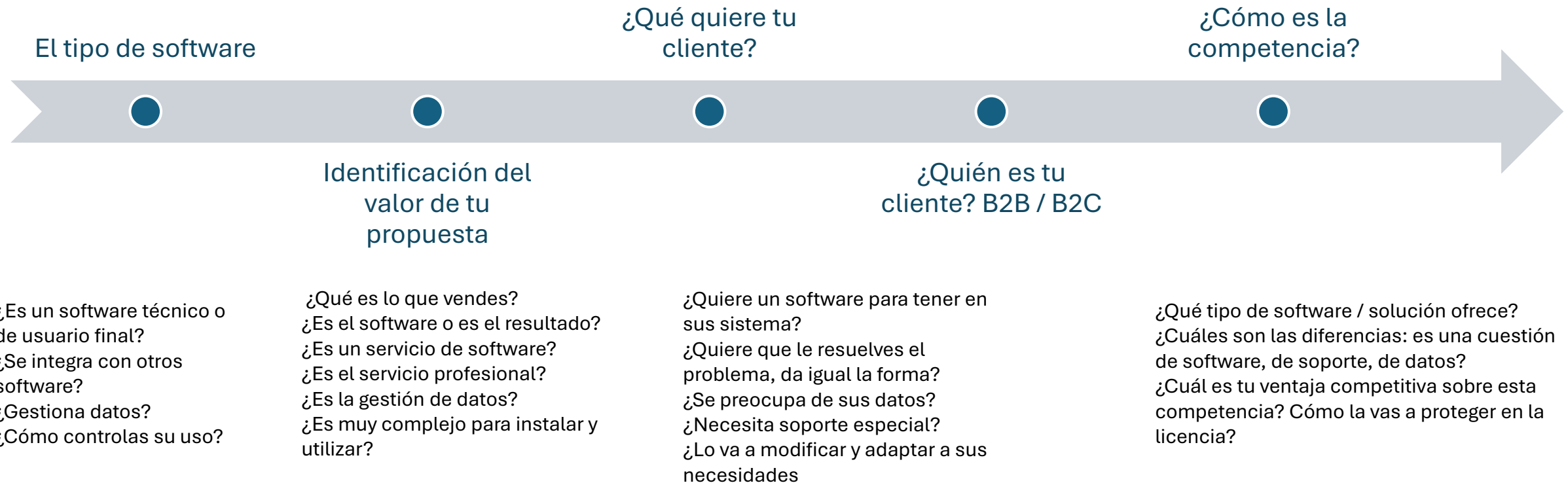
Aspectos comunes: Garantías de conformidad y funcionamiento. Garantías de IPR. Soporte y “SLAs”. Precio y pago

Ejemplo de la variedad de licencias



¿Qué licencia utilizo para mi negocio? ¿Cómo la elijo?

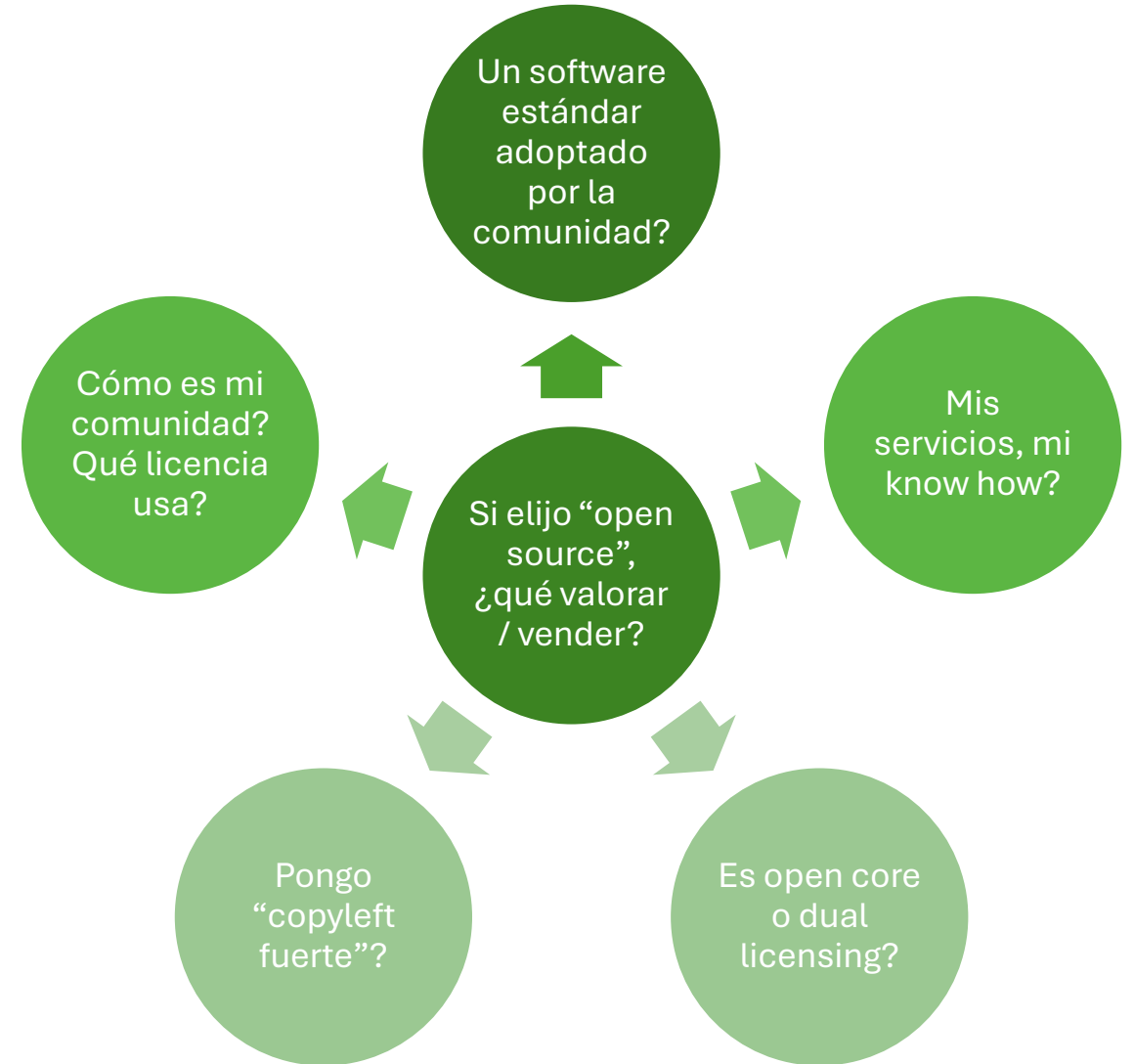
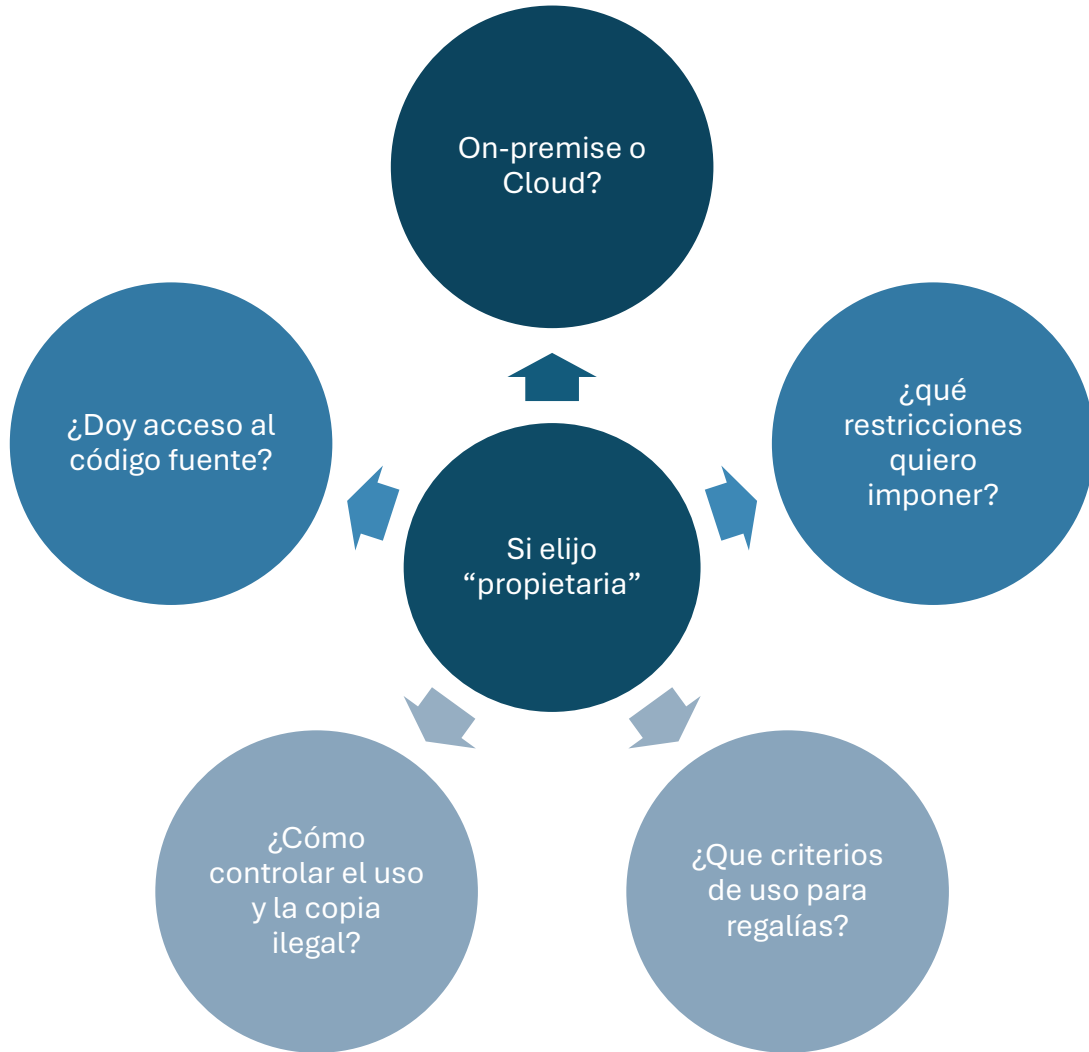
Pasos para elegir una licencia



Dos modelos “clásicos”

Tradicional /
“propietaria”

Open Source

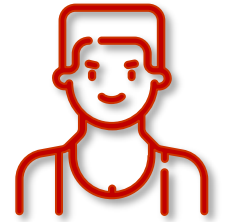


Decisiones sobre licencias



Bill y la licencia propietaria

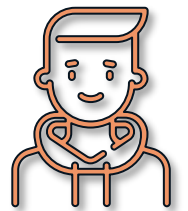
Marc y la licencia libre



4 Casos

Marta y los servicios SaaS

Ismael y el Producto mixto



Decisiones sobre licencias



Bill

Elijo una **licencia propietaria** con soporte nivel empresarial

Software empresarial complejo, con datos altamente sensibles.

El valor está en la seguridad e integración con infraestructura del cliente y los módulos especiales

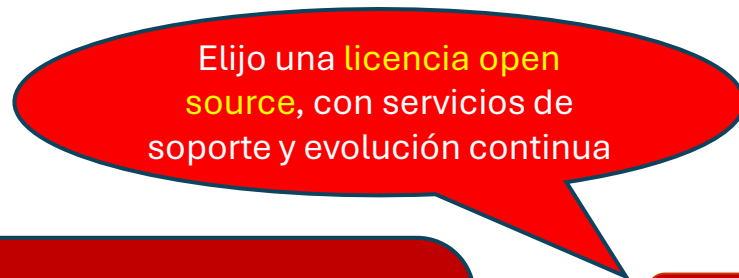


Marta

Elijo una **licencia SaaS** con alojamiento alta disponibilidad y APIs para integraciones

Software de Plataforma de e-commerce

El valor está en la rapidez de customización y implantación e integración con datos en la nube y pasarelas de pago

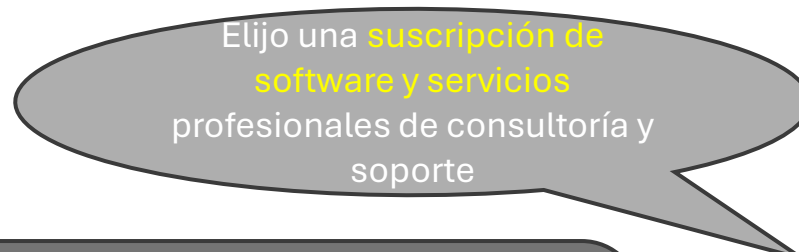


Marc

Elijo una **licencia open source**, con servicios de soporte y evolución continua

Software de AI, con motor y modelos.

El valor está en seguir trabajando con la comunidad científica, ofrecer libertad y continuidad de negocio al cliente, y en el entrenamiento continuo y soporte



Ismael

Elijo una **suscripción de software y servicios** profesionales de consultoría y soporte

Software para cálculos científicos y simulaciones

El valor está en la alta sofisticación científica, la aportación de un conocimiento técnico avanzado y análisis y solución para las necesidades de cliente



Bill

Software empresarial complejo, con datos altamente sensibles.

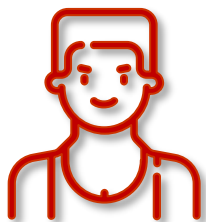
El valor está en la seguridad e integración con infraestructura del cliente y los módulos especiales

Precio por usuario o instalación
Restricciones de uso
Prohibición sobre la distribución
Garantías profesionales con SLA
Servicios adicionales: consultoría, instalación, integraciones
Clave de activación de licencia (control)
Derecho de auditoría
Opcionalmente: escrow o acceso al código fuente

• Temas legales que Bill tiene que tener en cuenta

- Asegurar la titularidad de todo el código
- Protección de funcionalidades (patentes?)
- Protección de la confidencialidad (código fuente, documentación)

- Licencia adecuada para prevenir la copia y distribución ilegal
- Auditorías de cumplimiento?
- Permitir al cliente modificar el código?
- Dar soporte y mantenimiento? SLA (acuerdo de nivel de servicio para el soporte)?
- Licencias evoluciones, o hacer pagar nueva licencia sobre las ampliaciones?



Marc

Software de AI, con motor y modelos.

El valor está en seguir trabajando con la comunidad científica, ofrecer libertad y continuidad de negocio al cliente, y en el entrenamiento continuo y soporte

No se cobra la licencia sino el servicio: asesoramiento, instalación, mantenimiento

Libertades de uso

Permite modificación y distribución

Garantías profesionales con SLA

Servicios adicionales: consultoría, adaptación, modelos o entrenamiento especiales

Actualizaciones rápidas, mejoras de la comunidad

Sin auditoría o control

Acceso al código fuente

Temas legales a tener en cuenta

- Asegurar la titularidad de su parte del código
- Gestionar contribuciones al proyecto abierto / libre
- Revisar y conocer las licencias libres sobre sus componentes

- Seleccionar la licencia libre de explotación
- Buscar el modelo de negocio basado en el software libre. Servicios? Marca? Extensiones?
- Gestionar la relación con la comunidad – buscar sostenibilidad



Marta

Software de Plataforma de e-commerce

El valor está en la rapidez de customización y implantación e integración con datos en la nbe y pasarelas de pago

Precio por usuario, datos u otra forma de uso online
Restricciones de conectividad
Prohibición sobre la reingeniería / ataque / hacking
Garantías profesionales con SLA sobre el servicio
Servicios adicionales: formación, integraciones, gestión datos
Control automático por autenticación en la plataforma
No necesita auditoría, etc.
Opcionalmente: pasar a “on premise”

Temas legales que Marta debe tener en cuenta

- Asegurar la titularidad de todo su código
 - Protección de funcionalidades (patentes?) y la confidencialidad (código fuente, documentación)
 - Definición del modelo SaaS (suscripción) y elegir el modelo de “licencia” (contrato de uso de la plataforma) adecuada para prevenir el uso ilegal (accesos no autorizados)
- Conectividad de los clientes: via web, APIs, SDK..? Condiciones de seguridad
 - Gestión de los datos de los clientes. privacidad, confidencialidad, seguridad,
 - Gestión de la infraestructura de la Nube: selección del hoste (Google, Azure, etc.) y modelo de seguridad, continuidad de negocio
 - Mantenimiento incluido - ¿Dar soporte? SLA (acuerdo de nivel de servicio para el soporte)?
 - Cliente: elasticidad? Libertad de cambio de plataforma? Opción “on-premise”?



Ismael

Software para cálculos científicos y simulaciones

El valor está en la alta sofisticación científica, la aportación de un conocimiento técnico avanzado y análisis y solución para las necesidades de cliente

Se cobra licencia (básica?) y sobre todo el servicio

Normalmente, amplios derechos de uso del cliente sobre el resultado

Permite modificación y pero a menudo, NO la distribución

Garantías profesionales pero pocas veces se da un SLA

Clave en los servicios : consultoría, adaptación, modelos o entrenamiento especiales

Mejoras en función de peticiones del cliente

Sin auditoría o control

Acceso al código fuente

Temas legales que Ismael debe tener en cuenta

- Asegurar la titularidad de todo el código
- Protección de funcionalidades (patentes?)
- Protección de la confidencialidad (código fuente, documentación)
- Protección del know-how en los servicios, retención del personal

- Definición de servicios de valor añadido
- Licencia y contrato de servicios adecuada alineado con los servicios prestados
- Gestión de las garantías sobre servicios, responsabilidades
- Condiciones de servicios de soporte y mantenimiento? SLA
- Propiedad intelectual sobre las customizaciones o resultados de servicios profesionales para el cliente

On-premise v. SaaS

La licencia “on premise”

Concesión de licencia de software

- Verdadera licencia: concesión de derechos de reproducción y uso
- licencia de instalación y uso en local
- Licencia por uso, por usuario, por instalación

Restricciones habituales

- Prohibición de hacer más copias
- Prohibición de redistribuir
- Prohibir la ingeniería inversa, descompilación

Garantías y soporte de un software

- Garantías de conformidad con la descripción técnica
- Garantía de funcionamiento y corrección de errores (SLA)

Control del usuario / policing el cumplimiento de la licencia

- Control de lugares de instalación (remoto: ping)
- Informes/Reporting periódica
- Derecho de auditoría y verificación

Otros aspectos

- Seguridad determinado por el cliente
- Datos almacenados y gestionados por el cliente
- Soporte de primer nivel: a menudo el servicio técnico del cliente
- Acceso al código fuente: directamente o bajo “escrow”
- Diferencias en EULA consumidor y Licencia “Enterprise” profesional: garantías, limitaciones, soporte,

La suscripción SaaS

Naturaleza

- No es licencia de PI. Es un contrato de prestación de servicio comercial
- Sin distribución de software – no hay “cesión” o “licencia” como tal.

Restricciones

- Conexión de usuarios no autorizados (identificación)
- Acceso a la tecnología (dashboard cliente?)
- No realizar usos ilegales

Garantías y soporte de un servicio

- Garantías de conformidad con la descripción técnica
- Garantía de funcionamiento y corrección de errores (SLA)
- Garantía de disponibilidad y conectividad de la plataforma

Gestión de datos cliente y datos personales

- Titularidad del cliente, restricciones de uso proveedor
- Reutilización para entrenamiento y mejora?
- Niveles de seguridad
- Obligaciones de protección de datos personales

Gestión de la infraestructura Cloud

- Subcontratación nube (Google Cloud, AWS)
- Niveles de seguridad y protección datos
- Continuidad servicio y replicas

Otros aspectos

- Control del proveedor sobre la gestión de datos del cliente (falta de independencia)
- Soporte primer nivel: usualmente el proveedor
- Profesional: Opciones de llevar todo “on-premise” con acceso a código fuente?
- Diferencias Consumidor (estándar, tipo Google) y Profesional (negociado, dimensionado, garantías negociadas, soporte formación)

Tabla comparativa

	On-premise	SaaS
Naturaleza	Licencia de software, de cesión de derechos de autor	Servicio tecnológico, con acceso remoto mediante internet
Restricciones	Copia, modificación, distribución Ingeniería inversa	Accesos no autorizados o ilegales Sin acceso al software
Precio	Todos modos (un tanto alzado, una regalía variables, cuotas fijas o variables anuales, etc.)	Suele ser “por uso”, con criterios de tiempo, de usuarios y de datos o conexiones
Garantías	Conformidad de software y correcciones de errores (parches entregados bajo SLA) Titularidad del software frente a reclamaciones de terceros	Conformidad y corrección de errores (en sistemas propios) Disponibilidad (conectividad de tipo 99.99%) Replicas y copias de seguridad de datos
Control	Informes y auditorías del cliente	Controles de acceso online
Otros	No implica temas de datos: los datos están bajo el control y en los sistemas del cliente Puede incluir servicios de instalación y configuración, mantenimiento evolutivo	Protección de la confidencialidad de los datos de cliente y controles sobre datos de carácter personal Aplicación de niveles y medidas de seguridad tecnológica y organizativas Gestión de la infraestructura / hosting

Gestión (legal) del desarrollo y explotación

Gestión del desarrollo - Aspectos “internos”

Titularidad, derechos y limitaciones

- Asegurar que el grupo/la entidad/la empresa es titular del concepto, know-how y código desarrollado
- Identificar autores y revisar sus contratos (relación laboral, becario, PDI, etc.)
- Todos los derechos de Propiedad intelectual e industrial deben estar cedidos al proyecto (entidad)

Protección jurídica

- Asegurar los derechos de propiedad intelectual e industrial, y proteger la confidencialidad
- Elegir las medidas formales de protección: registro del código? Registro de marca? patentabilidad)?
- Confidencialidad: definir perímetro y medidas técnicas, contratos con implicados
- Know-how: confidencialidad, pero también retención de personal

Patentabilidad

- Analizar y estudiar la patentabilidad de las funciones: ¿es una solución técnica a un problema técnico?
- Verificar si otros han patentado la idea
- “Estudio de patentabilidad” y “Freedom to operate”

Gestión del desarrollo - Aspectos Externos

Uso de componentes de terceros

- Opensource: identificar y hacer inventario de componentes y licencias, estudio de compatibilidad, analizar impacto sobre licenciamiento
- Librerías o frameworks propietarias: SDK/OEM: negociar licencia adecuada para la explotación (derechos de uso, precio, reporting, etc.)

Trabajar con terceros

- Socios / partners y subcontratas o consultores
- Aspectos legales: titularidad de resultados, acceso y uso de su “background” y “foreground”, confidencialidad, derecho a patentar, reutilización académica, garantizar posibilidad de transferir los resultados

Gestión de la explotación del software

Licencia y control

- Selección en función de criterios : on-premise, SaaS, SDK, etc.
- Valorar la propuesta de valorar y definir el tipo de licenciamiento (privativo, open source, B2B, B2C, etc.)
- Proceso de aceptación y firma del documento de licencia (Orden de Compra, Condiciones generales...)
- Controles sobre el uso: libre (open source) o con restricciones (tipo de controles, reporting, auditorías)

Entrega

- On premise: condiciones de instalación/despliegue, pruebas, aceptación, niveles de mantenimiento y actualizaciones
- SaaS: gestionar todos los aspectos relativos a la plataforma de hosting (AWS, Gcloud, Azure..): SLAs sobre todo, pero también datos personales, conectividad, replicas, etc.
- Código: asegurar los avisos de © Año, Titular + mención de la licencia

Servicios

- Definición y alcance, calendario, pagos, dependencias (de cliente, p.e.)
- Proceso de prestación y entrega y aceptación metodología (tradicional, agile..)
- Garantías de diligencias y profesionalismo
- SLAs sobre servicios
- Derechos de propiedad intelectual sobre los resultados? Del cliente o reutilizables?

7 aspectos esenciales

Propiedad:

- Autoría / titularidad (cesión IP empleados y freelancers)

- Qué diferencia entre “Autor” y “Titular”
- Si soy investigador, el software es mío o de la Universidad?
- Qué derechos tengo, si soy autor del algoritmo y software? Y los datos?
- Si contrato un desarrollador, cómo aseguro que yo sea el titular del resultado?

Cumplimiento:

- Licensing sobre componentes de 3º

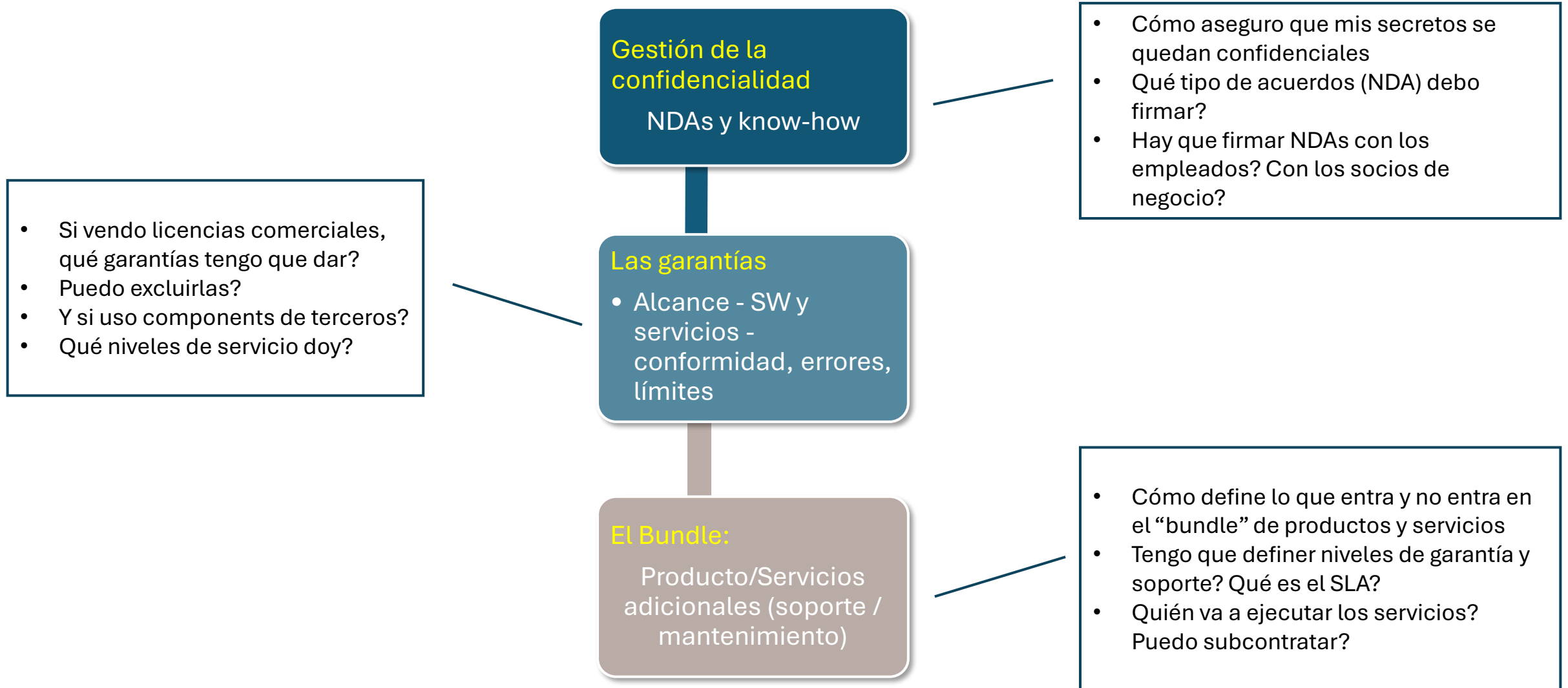
- Qué components estoy usando? Mis proveedores me lo tienen que indicar?
- Identificar los componentes y sus licencias
- Analizar las licencias, entender los impactos (copyleft?)

Selección de licencia:

- Propietaria v/s liberar v/s SaaS (licencia SW)

- En función del modelo de explotación, entender qué tipo de licencia, y sus características, será el mejor para su negocio
- Qué pongon la licencia propietaria?
- Qué licencia libre será mejor? Con copyleft? GPL? MIT?
- Puedo vender los dos modelos: On premise y Cloud?

7 aspectos esenciales



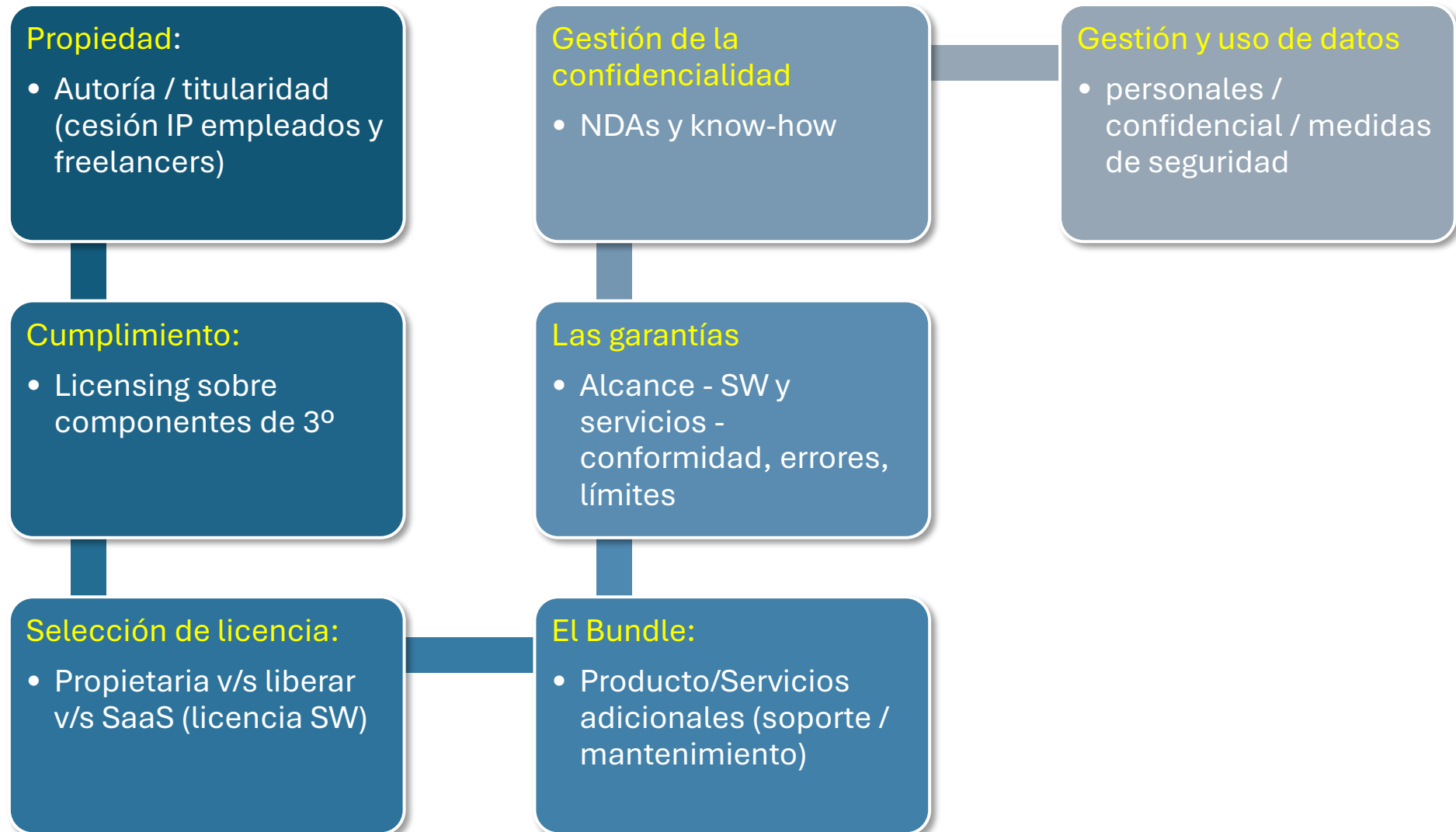
7 aspectos esenciales

- De quién son los datos en el sistema?
- Y los metadatos?
- Puedo usar los datos de cliente para mejorar mis sistemas?
- Tengo que anonimizar?
- Y los datos personales?
- Qué medidas de seguridad tengo que implementar?

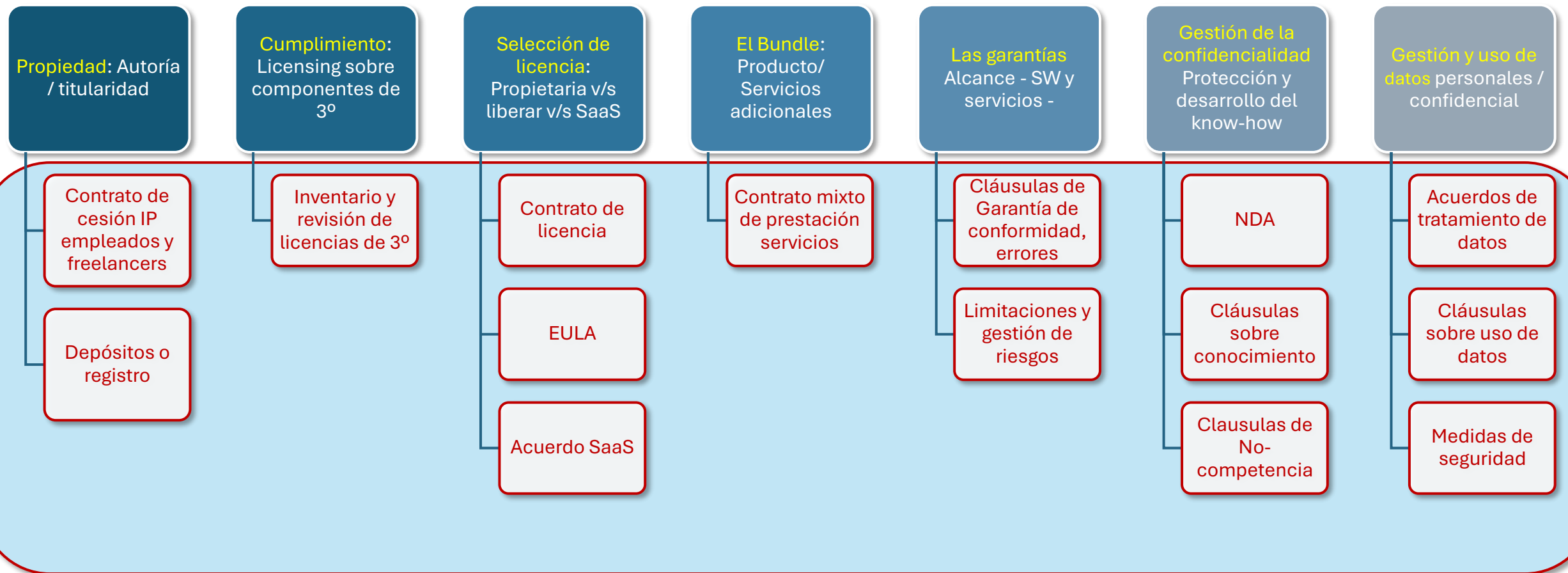
Gestión y uso de datos

- Datos personales
- Datos IA
- Datos confidenciales
- Medidas de seguridad

7 temas esenciales



La gestión legal del negocio de software – la documentación legal



Hot topics



SBOM: software bill of materials



IA: generación de código con IA



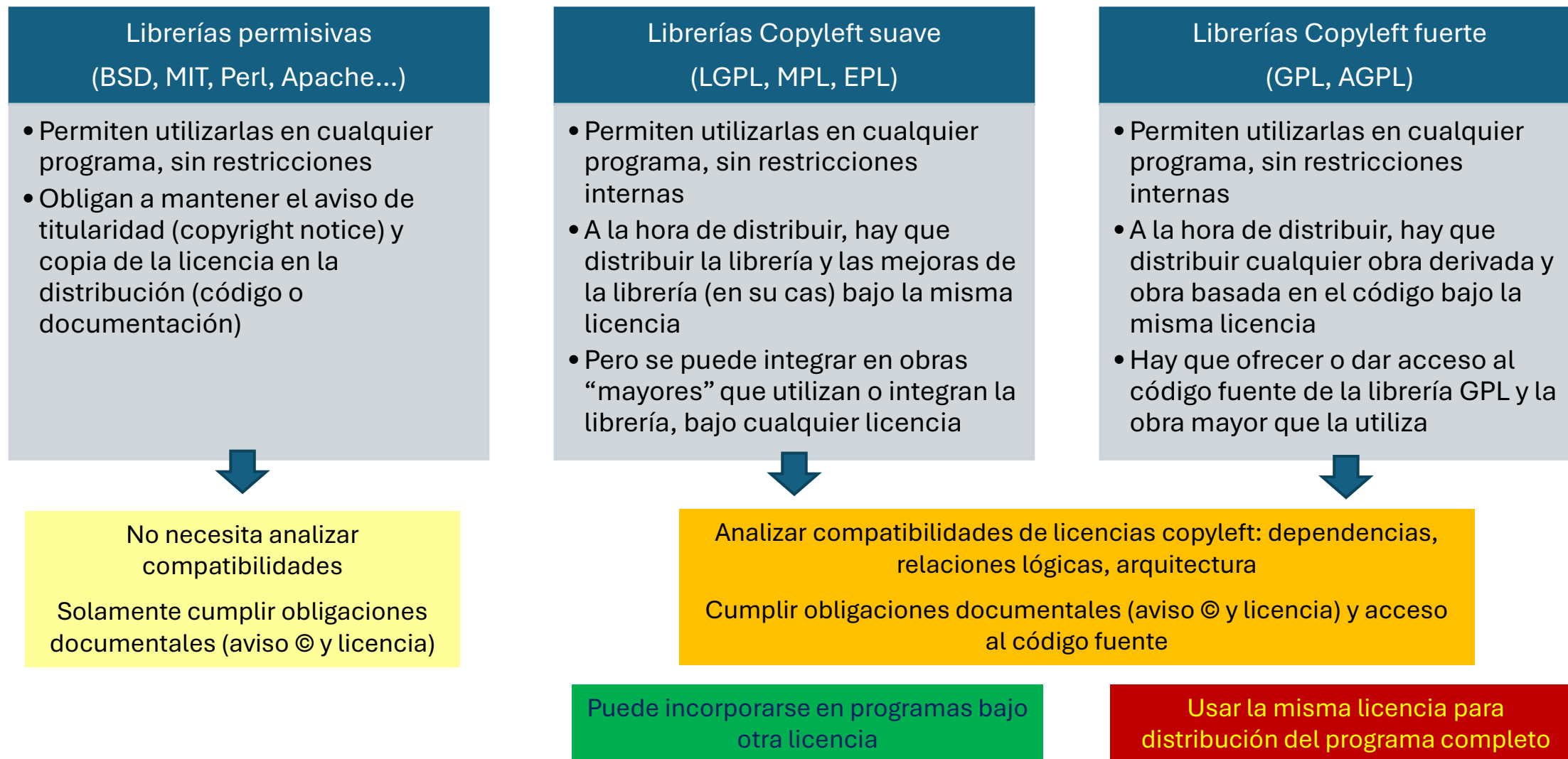
IA: integración de modelos de terceros



Datos: obtención, protección, uso para IA, distribución?

Guia breve de gestión del open source

Utilizar librerías Open Source



Compatibilidad de licencias open source

El análisis de compatibilidad depende de la relación lógica y arquitectura de los componentes

- Relaciones solamente de acumulación / uso a través de APIs estándares → no afectado por copyleft
- Relaciones de arquitectura (“Stack”) → no afectado (por ejemplo, ejecutarse sobre Linux, o entorno OpenJDK)
- Relaciones lógicas íntimas/fuertes, gran nivel de dependencia → se ha creado una obra “basada” en el componente, y hay que cumplir las obligaciones sobre modificación y distribución: mantener la misma licencia para la obra distribuida

Licencias permisivas son compatibles con todas licencias (excepto la BSD-4, incompatible con GPLv2)

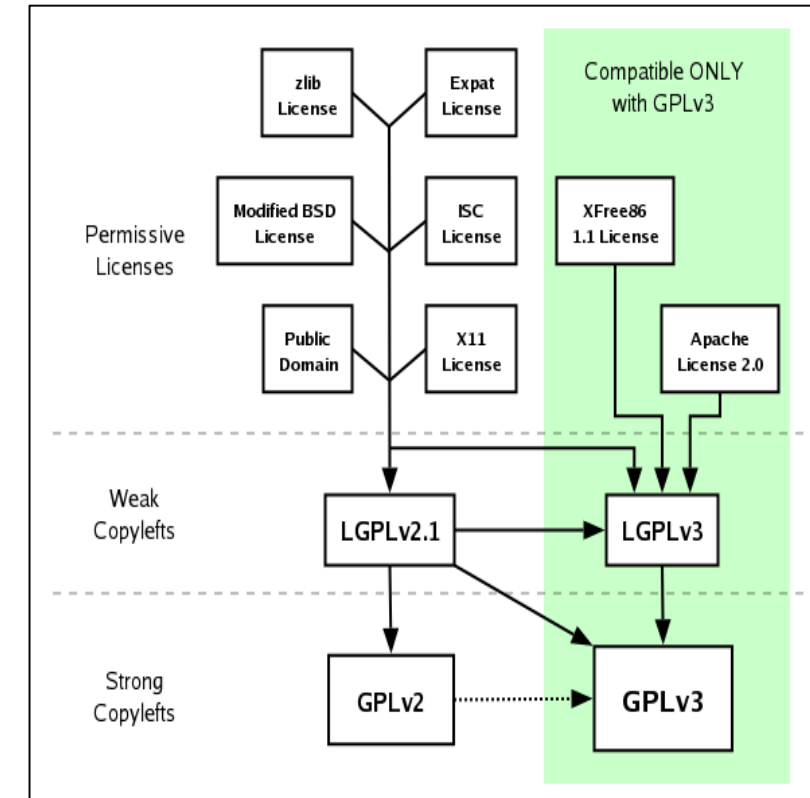
- Se puede integrar en obras permisivas, o con copyleft (porque permiten la redistribución con cualquier licencia)

Licencias copyleft fuerte (GPL/AGPL) son incompatibles con componentes bajo copyleft de cualquier tipo,

- SALVO que se indique el contrario en la licencia (LGPL → GPL, o MPL → LGPL, GPL, AGPL...)

El conjunto de licencias sobre componentes tiene que ser compatible con la licencia de distribución

- Ningun problema para licencias permisivas y copyleft suave (permiten incorporación en software distribuido bajo cualquier licencia)
- Componentes copyleft fuerte: lo “todo” tiene que distribuirse bajo la misma licencia



El contrato de desarrollo de software

Objetivo y naturaleza del contrato

Objetivo

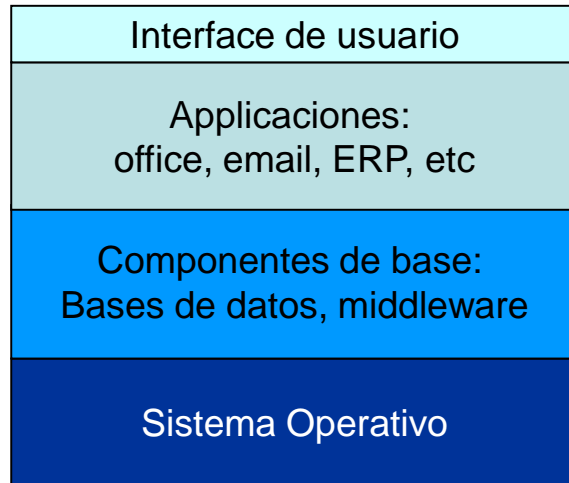
- Creación de un software a medida, desde cero o desde un modelo
- Adaptado a las necesidades del cliente
- Soporte y mantenimiento evolutivo y correctivo

Naturaleza del contrato

- **Para el cliente:** entrega de una obra, “llaves en mano”
 - Garantías de adecuación a necesidades, funcionamiento correcto, corrección de errores
- **Para el desarrollador:** prestación de un servicio... sin todas las llaves?
 - Garantías de calidad del desarrollo ... pero no necesariamente sobre el resultado

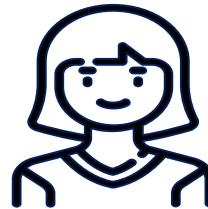
Factores esenciales para el modelo y la distribución de software

Modularidad: arquitectura del software

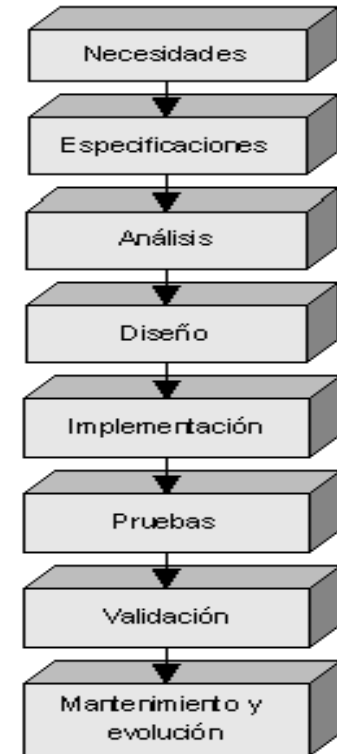


Qué estás entregando?
Qué estás incluyendo en la licencia?

¿cuáles son los
elementos esenciales
del desarrollo de
software?



Ciclo de vida del Software



Además del software, ¿qué
necesidades tiene el cliente?

Puntos claves del contrato de desarrollo

1. Definición del proyecto

- Qué es lo que se entrega – Anexo(s) técnico(s)
- Sobre qué parte nos hacemos responsables
- Qué es lo que NO hacemos (obligaciones cliente, terceros, dependencias)

2. Entregas y aceptación

- Plan de proyecto, fechas y formas de entrega
- Pruebas previas de validación
- Proceso de aceptación y documento de proyecto – definición de los criterios de aceptación

3. Gestión de modificaciones

- Flexibilidad frente a los cambios (cliente, entorno, tecnología)
- Proceso formal, documentado

4. Método de desarrollo: Waterfall or Agile

- Waterfall: entrega progresiva y final, con “hitos” (milestones) predefinidos → similar a entrega de una obra
- Agile: desarrollo iterativo a partir de “user stories” evolutivo (→ backlogs para sprints) → similar a la prestación de un servicio

Puntos claves del contrato de desarrollo

5. Propiedad intelectual

- Cesión de derechos: el cliente es “titular” de los derechos del software – ¿tiene independencia del proveedor?
- Licencia de uso: el cliente solamente tiene un derecho de uso (exclusivo o no...), ¿podemos reaprovechar?

6. Garantías

- De servicio: calidad de servicio, competencia del equipo, entregas en tiempo
- De producto: funcionamiento correcto, adecuación a necesidades, corrección de errores

7. SLA – soporte y mantenimiento

- Muchos tipos de SLA, desde “esfuerzos mínimos” hasta “24/7 con todo incluido”
- Consideraciones esenciales: objeto del soporte, proceso, criticidad y tiempos de respuesta, resolución, exclusiones

8. Precio y pago...

- Gestión del riesgo / esfuerzos / *cashflow*
 - Desarrollador: Pago por adelantado y pago por hito/iteración
 - Cliente: pago una vez todo entregado y verificado, retención de un % en garantía

Puntos transversales

1. Gestión de proyecto

- 90% de los problemas vienen de una gestión inadecuada de proyecto tecnológico
- Definición del alcance (“scope”), interacción con cliente, colaboración / input del cliente, desviaciones

2. Resolución de conflictos

- Fase previa: resolución amigable
- Fase formal: notificación y establecimiento de posiciones
- Fase conflictiva: tribunales o arbitraje / mediación

3. Tribunales / Arbitraje / Mediación

- Tribunales: más presión, más formal, proceso establecido por ley
- Arbitraje: resolución más flexible, confidencial, también vinculante. Considerar servicio de Arbitraje / Mediación de la OMPI

4. Gestión del open source

- Inventario y documentación
- Compatibilidades de licencia y “cumplimiento” en la entrega

5. “Escrow” del código y documentación

- Medida para asegurar el acceso del cliente a los resultados totales o particales
- Medida para proteger contra la desaparición del proveedor
- Deposito del código fuente y la documentación preparatoria / técnica
- Acceso bajo condiciones: insolvencia, resolución contrato soporate, incumplimiento graves, otros casos particulares

Conclusiones y repaso

Objetivos de aprendizaje

1. Conocer las diferentes estrategias de explotación del software (licencias, SaaS, etc.)

- Licencias privadas u open source
- On premise, SaaS, SDK, a medida

2. Conocer las buenas prácticas para la correcta gestión “legal” del software

- 7 pasos esenciales
- Documentación “legal”
- Gestión open source

3. Entender los conceptos más importantes del contrato de desarrollo y SLA del software

- Puntos clave
- Aspectos más generales / transversales

Fin!

Gracias por vuestra atención

Malcolm Bain
Across Legal, Barcelona
mbain@acrosslegal.com