



**SERVICIO NACIONAL DE APRENDIZAJE SENA
CENTRO DEL DISEÑO Y MANUFACTURA DE CUERO – COMPLEJO SUR**

NORMA / UNIDAD DE COMPETENCIA

Estructurar propuesta técnica de servicio de tecnología de la información según
requisitos técnicos y normativa

CÓDIGO NORMA DE COMPETENCIA LABORAL

220501094

NOMBRE DE LA COMPETENCIA

ELABORACIÓN DE LA PROPUESTA TÉCNICA DEL SOFTWARE.

RESULTADO DE APRENDIZAJE RAP.3

220501094-01- Definir especificaciones técnicas del software de acuerdo con las
características del software a construir

EVIDENCIA.

Evidencia GA2- 220501094-AA1-EV01: Foro temático: Licenciamiento de software

AUTOR

Grupo II

JAVIER ANDRES NARVAEZ GALIANO

ANDRES FELIPE SAENZ ARENAS

JOSE MANUEL VALENCIA MORENO

INSTRUCTOR

WILMAR ALEXANDER ZAPATA



- **¿Qué es software?**

- Conjunto de programas, datos, reglas y documentación asociada que le dicen a un ordenador qué hacer. En términos más sencillos, es la parte "intangible" de un sistema informático, en contraposición al hardware, que es la parte física.



El software además se encarga de dar funcionamiento y utilidad al hardware, según las características de las reglas y las funcionalidades de los programas del software se puede clasificar en diferentes tipos de software:

- **Software de sistema:** Controla el hardware del ordenador y proporciona una plataforma para ejecutar otro software. Ejemplos: Sistemas operativos (Windows, macOS, Linux), controladores de dispositivos.
- **Software de aplicación:** Permite a los usuarios realizar tareas específicas. Ejemplos: Procesadores de texto (Word), navegadores web (Chrome, Firefox), juegos.
- **Software de programación:** Se utiliza para crear otro software. Ejemplos: Compiladores, IDEs (Entornos de Desarrollo Integrados).

- **¿Qué es una licencia de software?**

- contrato legal entre el titular de los derechos de autor del software (generalmente el desarrollador o la empresa) y el usuario final. Esta licencia concede al usuario el derecho a utilizar el software bajo ciertos términos y condiciones. En esencia, te da permiso para usar el software, pero no te da la propiedad del mismo.
- No confundir el usuario final con el stake holder:



Usuario Final: Es la persona que interactúa directamente con el software para realizar una tarea específica. Por ejemplo, si se crea un software de contabilidad, el usuario final sería el contador o el empleado que utiliza el software para llevar las cuentas.



Stakeholder (Interesado): Es una persona o grupo que tiene interés en el éxito del proyecto de software, pero no necesariamente lo usa directamente. El cliente que solicitó la creación del software es un *stakeholder* clave. Otros *stakeholders* podrían ser:

- **Inversores que financian el proyecto**
- **El equipo de desarrollo que crea el software**
- **Departamentos de la empresa que se ven afectados por el software.**





- **Tipos de licencia de software.**
 - **Software de Código Abierto (Open Source):**
 - **Características:**
 - Un conjunto de principios para el desarrollo de software que promueven la colaboración, la transparencia y el acceso al código fuente.
 - Similar al software libre, pero con un enfoque más práctico y menos ideológico.
 - Las licencias de código abierto deben cumplir con la Open Source Definition (OSD).
 - En la práctica, la mayoría de las licencias consideradas "software libre" por la FSF también se consideran "código abierto" por la Open Source Initiative (OSI).
 - **Ejemplos:**
 - Linux (GPL)
 - Firefox (MPL)
 - Android (Apache License 2.0)
 - Python (Python Software Foundation License)
 - Node.js (MIT License)
 - **Diferencias Clave:**
 - En la práctica, hay poca diferencia entre software libre y software de código abierto. La principal diferencia radica en la filosofía detrás del movimiento.
 - El software de código abierto se enfoca en los beneficios prácticos de la colaboración y la transparencia.
 - El software libre se enfoca en la libertad del usuario.
 - **Software Propietario (también llamado "Software de Código Cerrado"):**
 - **Características:**
 - El código fuente no está disponible o está estrictamente controlado por el propietario.
 - El usuario tiene derechos limitados sobre el software, generalmente solo el derecho a usarlo de acuerdo con los términos de la licencia.
 - Normalmente requiere el pago de una licencia para su uso.
 - El propietario conserva todos los derechos de autor y controla las modificaciones y la distribución.
 - **Ejemplos:**
 - Microsoft Windows
 - Adobe Photoshop
 - Software de Autodesk (AutoCAD, etc.)
 - Muchos juegos comerciales



- **Diferencias Clave:**
 - El usuario **no** puede ver, modificar ni distribuir el código fuente (generalmente).
 - El usuario **debe** aceptar los términos de la licencia para usar el software.
 - El propietario del software generalmente ofrece soporte técnico.
- **Freeware:**
 - **Características:**
 - Software que se puede usar de forma gratuita, pero generalmente con restricciones.
 - El código fuente no está disponible.
 - El autor retiene los derechos de autor y controla la distribución.
 - Puede haber restricciones sobre el uso comercial, la modificación o la redistribución.
 - **Ejemplos:**
 - Algunos programas antivirus gratuitos (con funcionalidades limitadas)
 - Algunas herramientas de productividad
 - Algunos reproductores multimedia
 - **Diferencias Clave:**
 - Es gratuito para usar, pero no es necesariamente de código abierto o libre.
 - El usuario no tiene las libertades de software libre.
 - A menudo viene con términos de uso restrictivos.
- **Shareware:**
 - **Características:**
 - Software que se distribuye de forma gratuita durante un período de prueba.
 - Después del período de prueba, el usuario debe pagar una licencia para seguir usándolo.
 - El código fuente no está disponible.
 - **Diferencias Clave:**
 - Es gratuito por un tiempo limitado.
 - El usuario debe pagar para seguir usándolo después del período de prueba.
 - No es de código abierto ni libre.



- **Dominio Público:**
 - **Características:**
 - Software que no está protegido por derechos de autor.
 - Se puede usar, modificar y distribuir libremente sin restricciones.
 - No tiene autor o el autor ha renunciado a todos los derechos.
 - **Diferencias Clave:**
 - **No tiene restricciones** de uso, modificación o distribución.
 - Es el tipo de licencia más permisiva posible.
 - Es raro encontrar software nuevo que se libere directamente al dominio público.

Característica	Dominio Público	Shareware	Freeware	Código Cerrado (Propietario)	Open Source (Código Abierto)
Código Fuente	Disponible	No Disponible	No Disponible	No Disponible	Disponible
Modificación	Permitida	No Permitida	No Permitida	No Permitida	Permitida
Distribución	Permitida	Generalmente Restringida	Generalmente Restringida	Restringida	Permitida
Pago	No	Después del Periodo de Prueba	No	Generalmente	Opcional
Libertades	Completas	Limitadas	Limitadas	Limitadas	Completas
Uso Comercial	Permitido	Depende de la Licencia	Depende de la Licencia	Depende de la Licencia	Permitido (Generalmente)
Soporte Técnico	No Garantizado	No Garantizado (a menos que se compre)	No Garantizado	Generalmente Ofrecido	Depende del Proyecto/Comunidad
Ejemplo	Imágenes/Textos antiguos (sin derechos)	WinZip (prueba)	Adobe Acrobat Reader (gratis)	Microsoft Windows	Linux

- **¿Cuáles son las más adecuadas y por qué?**
 - La elección de una licencia depende de diversos factores, tales como los objetivos del creador del software, el modelo de negocio, el público objetivo, y si se desea fomentar la colaboración abierta o mantener un control estricto sobre el uso del software.



Ahora siendo así como ejemplo las siguientes licencias podrían ser las “mejores” si se cumplen ciertas condiciones:



■ Licencias propietarias o cerradas

- Adecuadas cuando:
 - El desarrollador o empresa quiere controlar el uso y la distribución del software.
 - Se busca generar ingresos a través de la venta de licencias o suscripciones.
 - El software está destinado a un público específico que requiere soporte técnico o características exclusivas.



■ Licencias de código abierto u open source

- Adecuadas cuando:
 - El objetivo es fomentar la colaboración y el desarrollo comunitario.
 - Se desea compartir el código y permitir que otros contribuyan o utilicen el software de manera libre.
 - El software tiene como objetivo la educación o el beneficio social.





CONCLUSIÓN

Es cierto que los tipos de licencia de software son muchas y variadas, con el fin de cubrir las necesidades y requisitos de un proyecto seguramente se irán generando nuevos modelos para el licenciamiento de software, a partir de los ya existentes, nos gustaría resaltar dos tipos de licencia que en nuestra consideración son los más importantes y relevantes a día de hoy y estos serían:

La licencia de propietario o cerrada y la licencia de código abierto u open source son fundamentales en el desarrollo tecnológico actual. El software propietario, como Microsoft Windows o Adobe Photoshop, ofrece soluciones integradas y controladas por sus desarrolladores, garantizando estabilidad y soporte dedicado. Por otro lado, las licencias de código abierto, como la Licencia Pública General de GNU (GPL), permiten a los usuarios acceder, modificar y distribuir el código fuente, fomentando la colaboración y la innovación comunitaria.

Un ejemplo reciente en el ámbito de la inteligencia artificial es la competencia entre modelos como ChatGPT de OpenAI y DeepSeek de la startup china DeepSeek. Mientras que ChatGPT opera bajo una licencia propietaria, DeepSeek ha adoptado un enfoque de código abierto. La aparición de DeepSeek ha provocado reacciones en el mercado, incluyendo una caída en el valor de las acciones de Nvidia y la reconsideración de estrategias por parte de grandes empresas tecnológicas y gobiernos. Este desarrollo subraya cómo las licencias de software, ya sean propietarias o de código abierto, pueden influir en la dinámica competitiva y en la adopción de tecnologías emergentes como la IA.

En resumen, las licencias de software propietario y de código abierto continúan siendo fundamentales en la configuración del ecosistema tecnológico actual, especialmente en áreas emergentes como la inteligencia artificial.

Bibliografía:

[DeepSeek has fundamentally redefined who controls the future of AI](#)