

# Software Libre vs Software Proprietario en Sistemas Operativos





# CONTENIDO

1. Definición de software libre y software propietario

2. Principales licencias

3. Ventajas del software libre

4. Ventajas del software propietario



# PART. 01

Definición de software libre y  
software propietario





## Características fundamentales

El software libre se caracteriza por permitir a los usuarios ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Estas libertades fomentan la colaboración y la transparencia.

## Ejemplos de software libre

Algunos ejemplos de software libre son Linux, Apache, LibreOffice y GIMP. Estas herramientas son ampliamente utilizadas y se pueden modificar según las necesidades del usuario.



# Software propietario

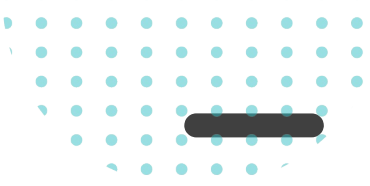
## Características fundamentales

El software propietario es aquel cuyo uso y distribución están restringidos por su propietario. Los usuarios no tienen acceso al código fuente y deben aceptar términos específicos de licencia.

## Ejemplos de software propietario

Ejemplos de software propietario incluyen Microsoft Windows, Adobe Photoshop y Oracle Database. Estos productos requieren una compra o licencia para su uso y no permiten modificaciones.





# PART. 02

## Principales licencias



## Definición y características

La Licencia Pública General de GNU (GPL) es una licencia que permite a los usuarios ejecutar, estudiar, compartir y modificar el software, garantizando que las modificaciones también sean libres.



## Casos de uso

Comúnmente utilizada en proyectos de software libre, la GPL es adecuada para desarrolladores que desean asegurar que su software y las contribuciones sean igualmente accesibles para todos.



## Definición y características

La Licencia MIT es una de las licencias más permisivas, permitiendo prácticamente cualquier uso del software, siempre que se incluya la misma licencia en el software redistribuido.

## PART 01

## Casos de uso

Ideal para desarrolladores que buscan maximizar la adopción de su software, la MIT es popular en proyectos de código abierto, permitiendo flexibilidad en el uso y la integración.

## PART 02



## 01.

---

### Definición y características

La Licencia BSD permite redistribuir y modificar el software con pocas restricciones, enfocándose en la libertad de uso. La EULA (Acuerdo de Licencia de Usuario Final) es un contrato que especifica cómo se puede usar el software.

## 02.

---

### Comparación con otras licencias

Comparada con la GPL, la BSD es menos restrictiva, permitiendo que el software sea cerrado, mientras que la EULA varía en condiciones, buscando proteger los derechos del desarrollador.



# PART. 03

Ventajas del software libre



## Comparación de precios

El software libre generalmente no tiene costos de licencia, lo que lo convierte en una opción económica en comparación con las soluciones propietarias.

## Accesibilidad

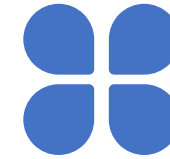
El software libre puede ser descargado y utilizado por cualquier persona, lo que facilita su acceso en entornos educativos y para estudiantes con recursos limitados.





## Personalización

Los usuarios pueden modificar el software libre para adaptarlo a sus preferencias y necesidades, lo que es ideal para proyectos escolares y personales.



## Adaptación a necesidades específicas

El software libre se puede ajustar para satisfacer requisitos específicos de cursos o programas, facilitando el aprendizaje y la enseñanza en diversas disciplinas.





## Soporte comunitario

Los usuarios pueden acceder a foros y grupos de discusión donde pueden obtener ayuda y compartir experiencias, creando un ambiente de aprendizaje colaborativo.



## Contribución y colaboración

El software libre fomenta la colaboración entre estudiantes, permitiendo a los usuarios contribuir al desarrollo del software, lo que enriquece su experiencia educativa.



# PART. 04

Ventajas del software propietario





## Asistencia técnica

La asistencia técnica garantiza que los usuarios puedan recibir ayuda inmediata para resolver problemas o dudas, mejorando así la experiencia general del software.



## Actualizaciones y mantenimiento

Las actualizaciones regulares son fundamentales para corregir errores y agregar nuevas funcionalidades, asegurando que el software esté siempre optimizado y seguro.



# Compatibilidad empresarial



## Integración con sistemas existentes

El software propietario suele diseñarse para trabajar sin problemas con otros sistemas de la empresa, facilitando la continuidad operativa y la reducción de costos.



## Estándares del mercado

Cumplir con los estándares del mercado es esencial para la interoperabilidad. El software propietario está diseñado para satisfacer las normas de la industria, garantizando su uso en diversas plataformas.



# Desarrollo controlado

## Gestión de calidad



La gestión de calidad en el software propietario implica estrictos procesos de pruebas y validación, lo que asegura que el producto final cumpla con las expectativas del usuario.

## Innovación controlada

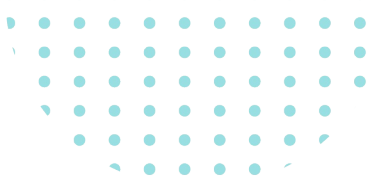


La innovación controlada permite a los desarrolladores implementar nuevas características de manera planificada, evitando cambios abruptos que podrían afectar la usabilidad del software.

# Ventajas y Limitaciones de Cada Modelo (resumen)

| Característica | Software Libre  | Software Propietario  |
|----------------|---|---|
| Ventajas       | <p>Costo: Generalmente gratuito, eliminando altos costos de licenciamiento.</p> <p>Flexibilidad: El acceso al código fuente permite una personalización total.</p> <p>Comunidad: Una comunidad global corrige errores y mejora el software constantemente. Es transparente y auditable.</p> | <p>Soporte Oficial: Cuentas con una empresa responsable que ofrece soporte técnico garantizado (SLA).</p> <p>Compatibilidad: Suele tener mayor compatibilidad con hardware y software comercial.</p> <p>Facilidad de Uso: A menudo tiene una curva de aprendizaje más suave y una experiencia de usuario más unificada.</p> |
| Limitaciones   | <p>Soporte: El soporte depende de la comunidad (foros) o de terceros (con costo).</p> <p>Curva de Aprendizaje: Puede requerir conocimientos técnicos más avanzados.</p> <p>Compatibilidad: A veces puede tener problemas con hardware muy específico o software muy especializado.</p>      | <p>Costo: Las licencias pueden ser muy caras, especialmente a gran escala.</p> <p>Dependencia (Vendor Lock-in): Estás atado a las decisiones, precios y futuro de una sola empresa.</p> <p>Falta de Flexibilidad: No puedes modificarlo ni auditar su código fuente en busca de fallos.</p>                                 |





# Thanks



MGS. JOSE FRANCO LAMILLA

