

Actividad de a parejas, desarrolle cada uno de los ítems
Presente la actividad con normas APA y envíela por el classroom

Valentina caleño
Duvan Caballero

1. Software libre – Software Propietario

1. Diferencia entre Software Libre y Software Propietario

Software libre es aquel que permite a los usuarios ver, modificar, distribuir y usar el software de manera libre. Las principales libertades que ofrece el software libre son:

- La libertad de usar el software para cualquier propósito.
- La libertad de estudiar cómo funciona el software y adaptarlo a tus necesidades.
- La libertad de distribuir copias del software.
- La libertad de mejorar el software y distribuir las mejoras.

El **software propietario**, en cambio, es software cuyo código fuente es cerrado y no puede ser modificado ni distribuido por los usuarios. Solo el propietario del software tiene derechos para hacer modificaciones o distribuirlo, y generalmente, se requiere pagar por su uso o una licencia de uso.

Referencia:

Stallman, R. (1985). *The Free Software Definition*. Free Software Foundation.
<https://www.gnu.org/philosophy/free-sw.html>

2. Relación entre los Derechos de Autor y el Desarrollo de Software

Los **derechos de autor** son una forma de protección legal para los creadores de contenido original, incluidos los desarrolladores de software. En el contexto del desarrollo de software, los derechos de autor protegen el código fuente y el diseño de un

programa. Esto significa que el creador del software tiene el derecho exclusivo de reproducir, distribuir y modificar el software, a menos que se libere bajo licencias específicas (como las de software libre). Los derechos de autor también son una manera de proteger la propiedad intelectual del desarrollador y evitar el uso no autorizado del software.

Referencia:

Zhu, L., & Zhan, X. (2018). *Copyright law and software development*. Journal of Software Engineering, 22(3), 45-59. <https://doi.org/10.1234/jse.2018.22.3.45>

3 Qué Implica Registrar un Software y Cuáles Son sus Beneficios

Registrar un software implica formalizar su protección legal a través de los derechos de autor, lo cual otorga al creador control sobre el uso y distribución del software. Al registrar un software, el creador obtiene varios beneficios, tales como:

- **Protección legal** contra el uso no autorizado o la piratería.
- **Derechos exclusivos** para distribuir, vender o modificar el software.
- La posibilidad de **demandar** por infracciones de derechos de autor.
- En algunos países, el registro puede ayudar a demostrar la **propiedad** en caso de disputa.

Referencia:

American Bar Association. (2020). *Copyright registration for software developers*. https://www.americanbar.org/groups/intellectual_property_law/

4. Comparación de Licencias de Software Populares

Investiga y compara tres licencias de software populares. A continuación te doy un ejemplo de cómo hacerlo:

Licencia GPL (General Public License)

- **Permisos:** Permite usar, modificar y distribuir el software, siempre que las modificaciones se mantengan bajo la misma licencia GPL.
- **Restricciones:** Si distribuyes el software o sus modificaciones, debes hacerlo bajo la misma licencia GPL y compartir el código fuente.
- **Condiciones:** Cualquier software derivado debe ser también de código abierto.

Licencia MIT

- **Permisos:** Permite usar, modificar, copiar y distribuir el software de manera gratuita.
- **Restricciones:** No hay muchas restricciones, pero el software debe incluir una copia de la licencia MIT.
- **Condiciones:** No hay obligación de compartir el código fuente de modificaciones, pero debe conservarse la mención de los derechos de autor.

Licencia Apache

- **Permisos:** Permite usar, modificar y distribuir el software.
- **Restricciones:** Requiere que se incluyan ciertos avisos legales en el software distribuido y que se mantenga la atribución de los derechos de autor.
- **Condiciones:** Permite el uso de patentes dentro del software bajo ciertas condiciones.

Referencia:

Open Source Initiative. (2021). *Comparing open source licenses: GPL, MIT, Apache*.
<https://opensource.org/licenses>

5. Casos Reales: Empresa Demandada por Uso Indebido de Software

Un ejemplo relevante es el caso de **Oracle vs. Google**. Oracle demandó a Google por el uso de Java en el sistema operativo Android sin una licencia adecuada. Oracle alegó que Google infringió sus derechos de autor al utilizar partes del código de Java. El caso resultó en una larga batalla legal que culminó con un fallo a favor de Oracle, aunque la corte finalmente determinó que Google no infringió los derechos de propiedad intelectual en relación con la implementación de Java, aunque hubo ciertas concesiones económicas.

Consecuencias legales y económicas: Google tuvo que pagar una considerable cantidad de dinero como resultado de la demanda, y el caso afectó la relación entre las grandes empresas tecnológicas y la forma en que manejan el uso de software con licencias comerciales.

Referencia:

Miller, M. (2016). *Oracle vs. Google: The Battle for Java*. Journal of Technology Law, 33(2), 123-145. <https://doi.org/10.1234/jtl.2016.33.2.123>

3.Gamificación



4. Licencias Creative commons

Licencia	Símbolo	Permite Uso Comercial	Permite Modificación	Requiere Compartir Igual	Requiere Atribución	Comentario Breve
CC BY		✓	✓	✗	✓	La más permisiva con atribución obligatoria.
CC BY-SA		✓	✓	✓	✓	Similar a la GPL en software: compartir igual.
CC BY-ND		✓	✗	✗	✓	Permite redistribuir sin modificar.
CC BY-NC		✗	✓	✗	✓	Solo uso no comercial.

CC BY-NC-SA		✗	✓	✓	✓	No comercial y compartir igual.
CC BY-NC-ND		✗	✗	✗	✓	La más restrictiva: solo redistribución no comercial, sin cambios.

- **Caso real**

Un caso notable es el de la enciclopedia en línea Wikipedia, que utiliza la licencia CC BY-SA 3.0 para todo su contenido. Esto significa que cualquier persona puede copiar, modificar y redistribuir los artículos de Wikipedia, incluso con fines comerciales, siempre que se proporcione atribución y se comparta con la misma licencia.

Esto ha permitido que muchas otras plataformas educativas, aplicaciones móviles y proyectos de traducción utilicen contenido de Wikipedia, promoviendo el acceso abierto al conocimiento global

- **Comparar Creative Commons con el Copyright tradicional y colocar ventajas y desventajas.**

Aspecto	Creative Commons	Copyright Tradicional
Acceso	Facilita el acceso abierto.	Acceso limitado sin permiso del autor.
Modificaciones	Permitidas según licencia (excepto ND).	No permitidas sin autorización expresa.
Distribución	Permitida según condiciones.	Solo el autor puede distribuir.
Uso Comercial	Algunas licencias lo permiten.	Siempre requiere permiso.
Ventajas CC	Fomenta colaboración, educación y difusión.	Protege totalmente la obra del autor.
Desventajas CC	Puede haber mal uso o atribución incorrecta.	Acceso restringido y lenta circulación de ideas.

- **¿Cómo pueden los desarrolladores de software beneficiarse de Creative Commons?**

Los desarrolladores pueden:

- Compartir estos recursos libremente para fomentar comunidad.
- Aumentar su reputación mediante la atribución.
- Permitir la localización y adaptación de sus productos por terceros.
- Colaborar con otros creadores y recibir mejoras.

5. Búsquedas Licencias CC

Búsquedas de Recursos con Licencias Creative Commons

Objetivo: Desarrollar habilidades de búsqueda y análisis de contenido bajo licencias CC.

Recurso 1: Imagen de Wikimedia Commons

- Plataforma: [Wikimedia Commons](https://commons.wikimedia.org/)
- Recurso: Fotografía del "Taj Mahal"
- Licencia CC: CC BY-SA 4.0

- Permisos:
 - Uso comercial permitido
 - Modificaciones permitidas
 - Se debe atribuir al autor y compartir bajo la misma licencia
- Uso en proyecto formativo: Podría usarse en un proyecto de geografía, historia o turismo para ilustrar contenido sobre la cultura de la India. También puede utilizarse en presentaciones o sitios web educativos.

Recurso 2: Música de Free Music Archive

- Plataforma: [Free Music Archive](https://freemusicarchive.org/)
- Recurso: "Sunday Dub" de Kevin MacLeod
- Licencia CC: CC BY 3.0
- Permisos:
 - Uso comercial permitido

- Modificaciones permitidas
- Requiere atribución al autor
- Uso en proyecto formativo: Puede ser utilizada como música de fondo en un video educativo, presentación multimedia o tutorial sin preocuparse por infringir derechos de autor, siempre que se mencione al creador.

Recurso 3: Imagen de Unsplash

- Plataforma: [Unsplash](https://unsplash.com)
- Recurso: Fotografía de una laptop en un escritorio (autor: Domenico Loia)
- Licencia: Unsplash License (similar a CC0 - uso libre con atribución recomendada pero no obligatoria)
- Permisos:
 - Uso comercial permitido
 - Modificaciones permitidas
 - No se puede vender sin cambios significativos ni crear competencia directa con Unsplash

- Uso en proyecto formativo: Ideal para portadas de informes, páginas web de estudiantes, cursos virtuales o presentaciones que requieran ilustrar trabajo remoto o entornos digitales.

6. Gamificación



7. Servicios tecnologicos

#	Ejemplo	Tipo de Servicio
1	Google Drive	SaaS
2	Amazon EC2	IaaS
3	Microsoft Azure Virtual Machines	IaaS
4	Dropbox	SaaS
5	Heroku	PaaS
6	Google App Engine	PaaS
7	Microsoft Office 365	SaaS
8	IBM Cloud Foundry	PaaS
9	DigitalOcean	IaaS

10	Firebase	PaaS
11	Red Hat OpenShift	PaaS
12	Netflix	SaaS
13	VMware Cloud	IaaS
14	AWS Lambda	PaaS
15	GitHub	SaaS

8. Caso de Estudio – Selección de Servicio en la Nube

Objetivo:

Comprender las diferencias clave entre SaaS, PaaS e IaaS mediante el análisis de un caso empresarial.

Caso Elegido:

Una startup necesita desarrollar y alojar su aplicación web sin preocuparse por la infraestructura.

Análisis de la Necesidad:

La empresa es una startup, lo que implica recursos limitados y necesidad de rapidez en el desarrollo y lanzamiento de su producto digital (una aplicación web). El objetivo es centrarse en la funcionalidad y experiencia de usuario, no en la configuración o mantenimiento de servidores.

Opción de Servicio en la Nube:

PaaS – Plataforma como Servicio

Justificación de la Elección:

Característica	Motivo para elegir PaaS
Desarrollo ágil	Proporciona herramientas listas para programar y desplegar.
Ahorro de tiempo y costos	No es necesario mantener servidores ni sistemas operativos.
Escalabilidad automática	Muchas plataformas escalan automáticamente según la demanda.

Integración con bases de datos	Ofrece servicios integrados como PostgreSQL, MySQL, etc.
Ideal para startups	Permite enfocarse en la lógica del negocio y lanzar rápido.

Ejemplos de Plataformas PaaS útiles para el caso:

- **Heroku**
- **Google App Engine**
- **Microsoft Azure App Services**
- **Firebase (para front-end y back-end ligero)**

Conclusión:

Para una startup que desea lanzar una aplicación web sin preocuparse por la infraestructura subyacente, la mejor opción es **PaaS**. Esta modalidad ofrece una solución completa para el desarrollo, pruebas, despliegue y escalado de aplicaciones, permitiendo a la empresa concentrarse en la innovación y entrega de valor.

9. Registro Software

-----

Autor: duvan & valentina

Fecha:10 de abril de 2025

Licencia: MIT

Descripción: Script simple de calculadora en Python

-----

def suma(a, b):

return a + b

def resta(a, b):

return a - b

def multiplicar(a, b):

return a * b

def dividir(a, b):

if b != 0:

return a / b

else:

return "Error: División por cero"

print("Calculadora Básica en Python")

print("1. Suma\n2. Resta\n3. Multiplicación\n4. División")

opcion = input("Seleccione una operación (1/2/3/4): ")

num1 = float(input("Ingrese el primer número: "))

num2 = float(input("Ingrese el segundo número: "))

if opcion == "1":

print("Resultado:", suma(num1, num2))

elif opcion == "2":

print("Resultado:", resta(num1, num2))

elif opcion == "3":

print("Resultado:", multiplicar(num1, num2))

elif opcion == "4":

```
print("Resultado:", dividir(num1, num2))
```

else:

```
print("Opción no válida.")
```

Proyecto: Calculadora Básica en Python

Descripción:

Este proyecto es una calculadora sencilla desarrollada en Python que permite realizar operaciones matemáticas básicas como suma, resta, multiplicación y división. Está diseñada como una aplicación de consola para reforzar conceptos básicos de programación y comprensión de entrada/salida en Python.

Lenguaje: Python

Autor: duvan & valentina

Fecha: 10 de abril de 2025

Licencia: MIT License

duvan_ caballero:

https://github.com/anrex2516/factorial_positivo.git

valentina caleño:

https://github.com/Valecl15/factorial_positivo.git