Guía de Proyectos Colaborativos Abiertos

Implementando la innovación abierta en el aula

> ***Serie:*** *Software Libre para la Comunidad Educativa*
>
> **Guía complementaria del Episodio 05 — Comunidad y Colaboración: El modelo de innovación abierta*
>
> ***Licencia:*** *CC BY-SA 4.0* · ***Autor:*** *Jared Lopez Leaño* · ***Producción:*** *OpenIT Academy* · ***Aval:*** *Núcleo Linux Bolivia*

1. Propósito de esta guía

Esta guía ofrece una ruta práctica para que docentes de cualquier área y nivel puedan **implementar proyectos colaborativos abiertos en el aula**, inspirándose en el modelo de desarrollo del Software Libre y la innovación abierta.

No se trata de convertir a todos los estudiantes en programadores. Se trata de que experimenten, en primera persona, cómo se construye conocimiento en red: **compartiendo, revisando el trabajo de otros, proponiendo mejoras y documentando lo aprendido.**

Al finalizar la implementación, los estudiantes habrán desarrollado habilidades como:

- Trabajo en equipo y comunicación efectiva.
- Revisión por pares y retroalimentación constructiva.
- Resolución colaborativa de problemas.
- Documentación y transparencia.
- Uso de herramientas libres y formatos abiertos.
- Comprensión del valor de compartir el conocimiento.

2. ¿Qué es un proyecto colaborativo abierto?

Un **proyecto colaborativo abierto** es una actividad educativa en la que un grupo de estudiantes trabaja de manera conjunta, distribuida y transparente para crear un producto o resolver un problema, siguiendo principios similares a los de las comunidades de Software Libre:

Principio	¿Qué significa en el aula?
Apertura	El proceso y los resultados son visibles para todos los participantes.
Colaboración	Cada estudiante aporta desde sus habilidades; el resultado es colectivo.
Revisión por pares	El trabajo de cada uno es revisado y mejorado por otros.

Transparencia	Se documentan las decisiones, los cambios y los errores.
Meritocracia	Las ideas se valoran por su calidad, no por el cargo o la nota.
Reutilización	El conocimiento generado puede ser usado y adaptado por otros.
Licencia abierta	Los productos se comparten con licencias que permiten su uso y modificación.

El mito del genio solitario

Durante mucho tiempo se nos enseñó que la innovación es un acto individual: un genio en un garaje, una empresa con secretos industriales, un inventor solitario. El Software Libre demuestra lo contrario: **los grandes avances tecnológicos son el resultado de comunidades enteras colaborando en red.**

Proyectos como **GNU/Linux** (que hace funcionar gran parte de internet) o **Moodle** (usado por millones de estudiantes) no fueron creados por una sola persona ni por una sola empresa. Son el resultado del trabajo de miles de personas que comparten un objetivo común.

> ***La lección para el aula:*** *el conocimiento no se construye en aislamiento, sino en red. El error es una oportunidad de mejora colectiva. Cada aporte, por pequeño que sea, fortalece al todo.*

3. Conceptos clave del modelo colaborativo abierto

Antes de implementar un proyecto, conviene familiarizarse con estos conceptos. No es necesario dominarlos técnicamente, pero sí entender su lógica.

Concepto	Definición adaptada al aula
Repositorio	El espacio común donde se guarda el proyecto y su historial de cambios. Puede ser una carpeta compartida, un documento en línea o un repositorio en Git.
Control de versiones	Sistema que registra quién cambió qué, cuándo y por qué. Permite volver atrás si algo sale mal.
Git	Herramienta libre de control de versiones, usada en los grandes proyectos de Software Libre.
Fork (bifurcación)	Copia personal de un proyecto para trabajar sin afectar la versión principal.
Pull request (solicitud de cambios)	Propuesta de mejora que un estudiante hace al proyecto común; otros la revisan antes de aceptarla.
Issue (incidencia)	Registro de un problema, error o tarea pendiente en el proyecto.
Revisión por pares	Proceso en el que otros estudiantes leen, comentan y sugieren mejoras al trabajo de un compañero.
Documentación	Registro escrito de lo que se hizo, cómo se hizo y por qué.
Licencia abierta	Permiso legal que autoriza a otros a usar, adaptar y compartir el trabajo (por ejemplo, CC BY-SA 4.0).
README	Archivo que explica de qué trata el proyecto, cómo usarlo y cómo contribuir.

> ***Recomendación:*** *no es necesario usar Git desde el primer día. Se puede empezar con herramientas más simples (documentos*

> *compartidos, carpetas en la nube) e ir incorporando complejidad según el nivel del grupo.*

4. Beneficios para los estudiantes

Los proyectos colaborativos abiertos no solo enseñan contenidos curriculares. Desarrollan habilidades transversales fundamentales para el siglo XXI:

4.1 Habilidades técnicas

- Uso de herramientas libres y formatos abiertos.
- Comprensión básica de control de versiones.
- Documentación técnica.
- Investigación y resolución de problemas.

4.2 Habilidades colaborativas

- Comunicación efectiva en equipo.
- Revisión por pares y retroalimentación constructiva.
- Distribución de tareas y responsabilidades.
- Respeto por el trabajo de otros.

4.3 Habilidades cognitivas

- Pensamiento crítico al evaluar el trabajo propio y ajeno.
- Descomposición de problemas complejos en tareas pequeñas.
- Tolerancia a la iteración y al error.
- Metacognición: reflexión sobre el propio aprendizaje.

4.4 Habilidades ciudadanas

- Comprensión del valor de compartir el conocimiento.
- Respeto por las licencias y la autoría.
- Participación en comunidades.
- Ética del trabajo colaborativo.

> ***Frase para recordar:*** *en las comunidades de Software Libre, las ideas se valoran por su calidad, no por el cargo de quien las propone. Esta es una excelente preparación para el mundo profesional y para la vida en sociedad.*

5. Tipos de proyectos colaborativos abiertos para el aula

A continuación se presentan **siete tipos de proyectos**, ordenados de menor a mayor complejidad técnica. Cada docente puede elegir el que mejor se adapte a su área, nivel y contexto.

5.1 Traducción y localización

¿En qué consiste? Traducir al español (o a una lengua local) la interfaz, la documentación o los materiales de un proyecto de Software Libre.

Áreas curriculares: Lengua, idiomas, comunicación.

Ejemplo: Traducir las cadenas de texto de una aplicación libre (como GIMP, LibreOffice o Moodle) o los tutoriales de un proyecto.

Habilidades desarrolladas: Precisión lingüística, trabajo colaborativo, uso de glosarios, revisión por pares.

Nivel de dificultad: ☆ Bajo

5.2 Documentación de proyectos

¿En qué consiste? Crear o mejorar la documentación de un proyecto: manuales, tutoriales, guías de uso, preguntas frecuentes.

Áreas curriculares: Lengua, comunicación, cualquier área técnica.

Ejemplo: Escribir un tutorial para usar una herramienta libre en el aula, con capturas de pantalla y explicaciones paso a paso.

Habilidades desarrolladas: Redacción clara, síntesis, empatía con el usuario, organización de información.

Nivel de dificultad: ☆☆ Bajo-Medio

5.3 Reporte de errores (issues)

¿En qué consiste? Identificar y reportar errores o mejoras en un proyecto libre, de forma clara y estructurada.

Áreas curriculares: Informática, ciencias, cualquier área que use software.

Ejemplo: Documentar un fallo en una aplicación libre con pasos para reproducirlo, resultado esperado y resultado obtenido.

Habilidades desarrolladas: Observación, comunicación técnica, pensamiento lógico, empatía con el equipo de desarrollo.

Nivel de dificultad: ☆☆ Bajo-Medio

5.4 Diseño gráfico y experiencia de usuario

¿En qué consiste? Crear íconos, ilustraciones, plantillas o mejorar la interfaz de un proyecto libre.

Áreas curriculares: Arte, diseño, tecnología.

Ejemplo: Diseñar un set de íconos para una aplicación educativa libre, o proponer mejoras de usabilidad.

Habilidades desarrolladas: Creatividad, comunicación visual, empatía con el usuario, trabajo con retroalimentación.

Nivel de dificultad: ☆☆☆ Medio

5.5 Creación de contenidos educativos abiertos (REA)

¿En qué consiste? Crear materiales educativos (apuntes, infografías, videos, presentaciones) con licencias abiertas, para que otros puedan usarlos y adaptarlos.

Áreas curriculares: Todas.

Ejemplo: Producir una infografía sobre un tema del currículo con licencia CC BY-SA 4.0, publicarla en un repositorio y permitir que otros la reutilicen.

Habilidades desarrolladas: Investigación, síntesis, producción multimedia, comprensión de licencias.

Nivel de dificultad: ☆☆☆ Medio

5.6 Contribución a Wikipedia o Wikidata

¿En qué consiste? Editar, mejorar o crear artículos en Wikipedia, o agregar datos estructurados en Wikidata.

Áreas curriculares: Lengua, historia, ciencias, cualquier área con contenido enciclopédico.

Ejemplo: Mejorar el artículo de Wikipedia sobre un tema local (una especie, un personaje histórico, un lugar) con fuentes verificables.

Habilidades desarrolladas: Investigación, verificación de fuentes, redacción enciclopédica, trabajo colaborativo, ética de la información.

Nivel de dificultad: ☆☆☆ Medio

5.7 Contribución a un proyecto de Software Libre

¿En qué consiste? Aportar código, pruebas, traducciones o mejoras a un proyecto real de Software Libre.

Áreas curriculares: Informática, programación, tecnología.

Ejemplo: Corregir un error sencillo en un proyecto libre, agregar una funcionalidad pequeña o mejorar una prueba automatizada.

Habilidades desarrolladas: Programación, uso de Git, comunicación en comunidades, revisión de código, trabajo distribuido.

Nivel de dificultad: ☆☆☆☆ Alto

> ***Recomendación:*** *para estudiantes que recién comienzan, se sugiere empezar por los tipos 5.1, 5.2 o 5.5, que no requieren conocimientos técnicos avanzados y permiten experimentar el modelo colaborativo desde el primer día.*

6. Implementación paso a paso

La siguiente ruta está pensada para un proyecto de **4 a 8 semanas**, adaptable a distintos niveles y áreas. Se puede implementar completa o por fases, según el tiempo disponible.

Fase 0 — Preparación docente (1 semana antes)

Paso	Acción	Herramientas sugeridas
1	Elegir el tipo de proyecto (sección 5) y el tema.	—
2	Definir los objetivos de aprendizaje y el producto final esperado.	Planificación docente
3	Seleccionar las herramientas libres que se usarán.	LibreOffice, Git, Moodle, Nextcloud, etc.
4	Preparar un ejemplo o modelo del producto esperado.	—
5	Diseñar la rúbrica de evaluación (sección 9).	—
6	Verificar el acceso a internet y a los equipos necesarios.	—
7	Informar a estudiantes y familias sobre el proyecto.	—

Fase 1 — Introducción y encuadre (1 sesión, 45–60 min)

Paso	Acción
1	Presentar el proyecto y su propósito: ¿qué van a crear? ¿para quién? ¿por qué es importante?
2	Explicar el modelo colaborativo abierto con ejemplos (GNU/Linux, Moodle, Wikipedia).
3	Introducir los conceptos clave (sección 3) de forma sencilla y con ejemplos.
4	Mostrar ejemplos de proyectos similares ya realizados.
5	Acordar las reglas de convivencia y colaboración: respeto, revisión constructiva, transparencia.
6	Formar los equipos (4–6 estudiantes) y asignar roles iniciales (sección 7).
7	Presentar la rúbrica de evaluación para que sepan qué se espera.

Fase 2 — Organización y planificación (1 sesión, 45 min)

Paso	Acción
1	Cada equipo define su objetivo específico dentro del proyecto común.
2	Descomponer el objetivo en tareas pequeñas y asignarlas.
3	Crear el espacio de trabajo compartido: carpeta, documento en línea o repositorio.
4	Escribir el **README** del proyecto: de qué trata, quiénes participan, cómo se organiza.
5	Establecer un calendario con hitos y fechas de entrega.
6	Acordar la licencia abierta que tendrá el producto final (por ejemplo, CC BY-SA 4.0).

Fase 3 — Desarrollo colaborativo (2–4 semanas)

Paso	Acción
1	Los equipos trabajan en sus tareas, documentando avances y decisiones.
2	Cada cierto tiempo, se realizan **revisiones por pares**: otros equipos leen y comentan el trabajo.
3	Se registran los problemas encontrados (issues) y se proponen soluciones.
4	Se incorporan las mejoras sugeridas, documentando los cambios.
5	El docente actúa como facilitador: aclara dudas, da retroalimentación y asegura la participación de todos.
6	Se realizan **reuniones breves de equipo** (10 min) al inicio o final de cada sesión para sincronizar.

Consejos de facilitación:

- Circular entre los equipos y escuchar sin intervenir demasiado.
- Fomentar que los estudiantes resuelvan sus propios conflictos.
- Recordar que el error es parte del proceso: documentarlo es una oportunidad de mejora.
- Asegurar que todos los integrantes participen activamente.

Fase 4 — Integración y revisión final (1 semana)

Paso	Acción
1	Integrar los aportes de todos los equipos en el producto común.
2	Realizar una **revisión final colaborativa**: todos revisan el conjunto.
3	Corregir errores, mejorar la documentación y pulir el producto.
4	Preparar una **presentación breve** (5 min) del proyecto para compartir con el resto del grupo.
5	Publicar el producto final con su licencia abierta.

Fase 5 — Presentación y cierre (1 sesión, 45–60 min)

Paso	Acción
1	Cada equipo presenta su aporte y el producto final común.
2	Se comparten los aprendizajes: ¿qué aprendieron? ¿qué fue difícil? ¿qué harían diferente?
3	Se realiza una **autoevaluación y coevaluación** con la rúbrica.
4	Se publica el producto en un repositorio o sitio web, con su licencia abierta.
5	Se invita a otros grupos, docentes o instituciones a reutilizarlo.
6	Se celebra el logro colectivo.

7. Roles en la comunidad de aula

Para que un proyecto colaborativo funcione, es útil asignar roles claros. Estos roles pueden rotar entre los estudiantes a lo largo del proyecto.

Rol	Responsabilidad
Coordinador/a de equipo	Organiza las reuniones, distribuye tareas, vela por el cumplimiento del calendario.
Documentador/a	Registra las decisiones, los avances y los cambios en el README y otros documentos.
Revisor/a de calidad	Lee el trabajo de sus compañeros y propone mejoras constructivas.
Gestor/a de incidencias	Registra los problemas encontrados y les da seguimiento.
Comunicador/a	Presenta el proyecto al resto del grupo y a la comunidad educativa.
Enlace con la comunidad externa	Contacta a otros docentes, proyectos o comunidades que puedan aportar.
Facilitador/a docente	Acompaña, aclara, da retroalimentación y asegura la participación de todos.

> ***Consejo:*** *los roles no deben ser rígidos. En las comunidades de Software Libre, las personas aportan desde sus habilidades y van rotando según las necesidades del proyecto.*

8. Herramientas libres recomendadas

8.1 Para la organización del proyecto

Necesidad	Herramienta libre	Alternativa simple
Documento compartido	Nextcloud, Etherpad, CryptPad	Documento en línea compartido
Control de versiones	Git, GitLab, Gitea, Codeberg	Carpeta compartida con historial
Gestión de tareas	Nextcloud Deck, Kanboard, Focalboard	Tablero en la pizarra o en un documento
Incidencias	GitLab Issues, Gitea Issues	Documento de "pendientes"
Comunicación	Mattermost, Element	Correo electrónico o grupo de mensajería
Publicación del producto	Nextcloud, GitLab Pages, Codeberg Pages	Blog del aula o sitio web escolar

8.2 Para la producción de contenidos

Necesidad	Herramienta libre
Documentos de texto	LibreOffice Writer
Hojas de cálculo	LibreOffice Calc
Presentaciones	LibreOffice Impress
Imágenes vectoriales	Inkscape
Imágenes raster	GIMP, Krita
Audio	Audacity
Video	Kdenlive
Libros electrónicos	Calibre, Sigil
Infografías	Inkscape, Scribus

8.3 Para la publicación y el intercambio

Necesidad	Herramienta libre
Repositorio de código	GitLab, Codeberg, Gitea
Repositorio de REA	Moodle, Nextcloud, OER Commons
Wiki	MediaWiki (Wikipedia)
Licencias abiertas	Creative Commons, GNU GPL

9. Rúbrica de evaluación sugerida

La siguiente rúbrica puede adaptarse según el nivel, el área y los objetivos específicos del proyecto. Se recomienda compartirla con los estudiantes al inicio.

Criterio	Excelente (4)	Bueno (3)	En proceso (2)	Inicial (1)
Participación y colaboración	Participa activamente, apoya a sus compañeros y asume su rol con	Participa con regularidad y cumple sus tareas.	Participa de forma irregular y necesita recordatorios.	Participa poco o de forma pasiva.

	responsabilidad.			
Calidad del aporte	Su aporte es significativo, bien fundamentado y mejora el producto colectivo.	Su aporte es correcto y cumple con lo esperado.	Su aporte es incompleto o necesita correcciones.	Su aporte es mínimo o no se integra al proyecto.
Revisión por pares	Revisa con detalle, hace comentarios constructivos y propone mejoras concretas.	Revisa y comenta, aunque con poco detalle.	Revisa de forma superficial o con comentarios poco útiles.	No revisa el trabajo de sus compañeros.
Documentación	Documenta de forma clara, completa y ordenada; el README está actualizado.	Documenta lo esencial, aunque con vacíos.	Documenta poco o de forma desordenada.	No documenta su trabajo.
Uso de herramientas abiertas	Usa herramientas libres y formatos abiertos de forma autónoma y correcta.	Usa las herramientas indicadas con algo de apoyo.	Usa las herramientas con dificultad o de forma incompleta.	No usa las herramientas indicadas.
Comunicación	Se expresa con claridad, escucha y respeta las ideas de otros.	Se comunica bien, aunque con algunos problemas.	Se comunica con dificultad o interrumpe.	No se comunica o no respeta a sus compañeros.
Reflexión y metacognición	Reflexiona profundamente sobre su aprendizaje y propone mejoras para el futuro.	Reflexiona sobre lo aprendido con algo de profundidad.	Su reflexión es superficial.	No reflexiona sobre su aprendizaje.
Ética y licencias	Comprende y respeta las licencias; cita las fuentes y comparte con licencia abierta.	Respeta las licencias, aunque con algunos descuidos.	Comete errores en el uso de licencias o citas.	No respeta las licencias ni cita las fuentes.

Puntaje total: ____ / 32

10. Ejemplos de proyectos listos para adaptar

Ejemplo 1: "Traducimos juntos" (nivel bajo)

Área: Lengua / idiomas

Duración: 3 semanas

Producto: Traducción colaborativa de una aplicación libre al español o a una lengua local.

Pasos:

1. Elegir una aplicación libre pequeña (por ejemplo, un juego educativo o una extensión de navegador).
2. Dividir las cadenas de texto entre los equipos.
3. Traducir, revisar por pares y unificar criterios con un glosario.
4. Enviar la traducción al proyecto original (o publicarla en un repositorio).

5. Presentar el proceso y los aprendizajes.

Ejemplo 2: "Manuales que ayudan" (nivel bajo-medio)

Área: Lengua / comunicación / informática

Duración: 4 semanas

Producto: Manual de uso de una herramienta libre para la comunidad educativa.

Pasos:

1. Elegir una herramienta libre que se use en la escuela (LibreOffice, GIMP, Scratch, etc.).
2. Investigar su funcionamiento y detectar las dudas más frecuentes.
3. Escribir un manual con pasos, capturas de pantalla y consejos.
4. Revisar por pares y con usuarios reales (otros estudiantes o docentes).
5. Publicar con licencia CC BY-SA 4.0.

Ejemplo 3: "Wikipedia en el aula" (nivel medio)

Área: Historia / ciencias / lengua

Duración: 5 semanas

Producto: Artículos de Wikipedia mejorados o creados sobre temas locales.

Pasos:

1. Elegir un tema local con poca información en Wikipedia (una especie, un personaje, un lugar).
2. Investigar con fuentes verificables.
3. Redactar el artículo siguiendo las convenciones de Wikipedia.
4. Revisar por pares y con la comunidad de Wikipedia.
5. Publicar y dar seguimiento.

Ejemplo 4: "Infografías abiertas" (nivel medio)

Área: Ciencias / arte / comunicación

Duración: 4 semanas

Producto: Infografías educativas con licencia abierta.

Pasos:

1. Elegir un tema del currículo.
2. Investigar y sintetizar la información.
3. Diseñar la infografía con Inkscape o Scribus.
4. Revisar por pares y con usuarios reales.
5. Publicar en un repositorio de REA con licencia CC BY-SA 4.0.

Ejemplo 5: "Primer aporte a un proyecto libre" (nivel alto)

Área: Informática / programación

Duración: 6–8 semanas

Producto: Contribución real a un proyecto de Software Libre.

Pasos:

1. Elegir un proyecto libre amigable para principiantes (con etiquetas como "good first issue").
2. Leer la documentación y la guía de contribución.
3. Hacer un fork del repositorio.
4. Resolver una incidencia sencilla (error, mejora, documentación).
5. Enviar un pull request y responder a la revisión.
6. Documentar el proceso y compartir la experiencia.

11. Preguntas frecuentes

¿Necesito saber programar para implementar un proyecto colaborativo abierto?

No. La mayoría de los proyectos (traducción, documentación, REA, Wikipedia) no requieren programación. Solo los proyectos de contribución a Software Libre (sección 5.7) requieren conocimientos técnicos.

¿Cuánto tiempo requiere implementar un proyecto?

Depende del tipo y la profundidad. Un proyecto sencillo puede durar 3 semanas; uno más complejo, 6–8 semanas. Se recomienda empezar con algo pequeño y gradual.

¿Qué pasa si los estudiantes no tienen acceso a internet en casa?

Se puede trabajar con herramientas que funcionen sin conexión (LibreOffice, GIMP, Audacity) y sincronizar en clase. También se puede usar un servidor local (Nextcloud, Moodle) en la escuela.

¿Cómo evalúo un proyecto colaborativo?

Con la rúbrica de la sección 9, que combina evaluación del proceso (participación, revisión por pares, documentación) y del producto. Se recomienda incluir autoevaluación y coevaluación.

¿Qué hago si un estudiante no participa?

Conversar con él o ella para entender las causas. A veces es timidez, falta de acceso o desconocimiento. Se puede ajustar el rol o la tarea, o acompañar de cerca. Si el problema persiste, documentarlo y buscar apoyo con el equipo de orientación.

¿Debo publicar el producto final en internet?

No es obligatorio. Se puede publicar en un repositorio interno o en un sitio web de la escuela. Lo importante es que el producto tenga una licencia abierta y que la comunidad educativa pueda reutilizarlo.

¿Cómo manejo los conflictos en los equipos?

Fomentar la comunicación abierta y el respeto. Si surge un conflicto, mediar sin imponer soluciones. Recordar que en las comunidades abiertas los desacuerdos son parte del proceso y se resuelven con argumentos, no con autoridad.

¿Qué licencia debo usar para el producto final?

Se recomienda **CC BY-SA 4.0** para contenidos educativos, y **GNU GPL** o **MIT** para código. Ambas permiten el uso, la modificación y la redistribución, siempre que se reconozca la autoría y se comparta con la misma licencia.

12. Lista de verificación para el docente

Antes de empezar

- Elegí el tipo de proyecto y el tema.
- Definí los objetivos de aprendizaje y el producto final.
- Seleccioné las herramientas libres que se usarán.
- Preparé un ejemplo o modelo del producto esperado.
- Diseñé la rúbrica de evaluación.
- Verifiqué el acceso a internet y a los equipos necesarios.
- Informé a estudiantes y familias sobre el proyecto.

Durante el proyecto

- Presenté el modelo colaborativo abierto con ejemplos.
- Formé los equipos y asigné roles iniciales.
- Acompañé el trabajo de los equipos con retroalimentación.
- Promoví la revisión por pares y la documentación.
- Aseguré la participación de todos los estudiantes.
- Registré los avances y los problemas encontrados.

Al finalizar

- Integré los aportes en el producto final.
- Realicé una revisión final colaborativa.
- Publiqué el producto con su licencia abierta.
- Facilité la presentación y la reflexión final.
- Apliqué la rúbrica con autoevaluación y coevaluación.
- Compartí la experiencia con otros docentes.
- Celebré el logro colectivo.

13. Recursos para profundizar

Sobre el modelo colaborativo abierto

- **Episodio 05 completo** — *Comunidad y Colaboración: El modelo de innovación abierta* (repositorio de la serie).
- **The Cathedral and the Bazaar** — Eric S. Raymond (texto clásico sobre modelos de desarrollo).
- **Producing Open Source Software** — Karl Fogel (guía libre sobre cómo funcionan los proyectos abiertos).
- **Open Source Guides** — opensource.guide (guías para contribuir a proyectos libres).

Sobre proyectos educativos abiertos

- **Moodle** — moodle.org
- **Wikipedia** — www.wikipedia.org
- **Wikimedia Education** — outreach.wikimedia.org
- **Creative Commons** — creativecommons.org
- **OER Commons** — www.oercommons.org

Sobre herramientas libres

- **LibreOffice** — www.libreoffice.org
- **Git** — git-scm.com
- **Codeberg** — codeberg.org
- **GitLab** — gitlab.com
- **Inkscape** — inkscape.org
- **GIMP** — www.gimp.org

- **Audacity** — www.audacityteam.org
- **Kdenlive** — kdenlive.org

Sobre comunidades educativas

- **Núcleo Linux Bolivia** — nucleolinux.org
- **OpenIT Academy** — recursos de formación abierta.
- **Free Software Foundation** — www.fsf.org
- **Document Foundation** — www.documentfoundation.org

14. Créditos y licencia

Campo	Valor
Serie	Software Libre para la Comunidad Educativa
Guía	Guía de Proyectos Colaborativos Abiertos
Episodio relacionado	05 | Comunidad y Colaboración: El modelo de innovación abierta
Autor	Jared Lopez Leaño
Producción	OpenIT Academy
Aval	Núcleo Linux Bolivia
Licencia	CC BY-SA 4.0
Hashtags	#Comunidad #Colaboración #InnovacionAbierta #SoftwareLibre #Educación #Moodle #Linux #NúcleoLinuxBolivia #OpenITAcademy

> ***El conocimiento se multiplica cuando se comparte.***
>
> *📥 Comparte esta guía, adáptala a tu contexto y comienza a construir en red con tus estudiantes.*
>
> ***Aprender con libertad es aprender mejor.***