



Fundamentos de GitHub. Infraestructuras OpenAIRE y Zenodo.

Anexo técnico para el entregable *D3.7 PoliRural GitHub Account*

Proyecto	PoliRural	
Título del proyecto:	Desarrollo de políticas de colaboración orientadas al futuro para las zonas rurales y sus habitantes	
Subvención	818496	
Página web:	www.polirural.eu	
Contacto:	info@polirural.eu	
Versión:	1.0	
Date:	Septiembre de 2022	
Responsable :	TRAGSA	
Nivel de difusión:	Público	X
	Confidencial - sólo miembros del consorcio y servicios de la Comisión Europea	
Palabras clave:	GitHub, OpenAIRE, Open Science, Zenodo, acceso abierto, intercambio de datos, datos abiertos, metadatos.	

Este proyecto ha recibido financiación del programa de investigación e innovación Horizonte 2020 de la Unión Europea en virtud del acuerdo de subvención nº 818496.

Historial de revisiones

Nº de revisión	Fecha	Autor	Organización	Descripción
0.x	Marzo-agosto 2022		TRAGSA	Estructura básica y borradores sucesivos
1.0	Septiembre de 2022		TRAGSA	Versión final

Los autores son los únicos responsables de la información y opiniones contenidas en esta publicación.

Se ha hecho todo lo posible para garantizar que todas las declaraciones e información contenidas en este documento sean exactas, pero los socios del proyecto PoliRural no aceptan ninguna responsabilidad por cualquier error u omisión.

Índice

Índice de materias	3
Lista de cuadros	4
Lista de figuras	5
Glosario.....	6
1 Introducción.....	8
2 Fundamentos de GitHub.....	9
2.1 Git	9
2.2 Interfaces gráficas de usuario (GUI) para Git	10
2.3 GitHub y GitLab.....	11
2.4 Funcionalidades de GitHub	11
2.4.1 Conceptos clave	11
2.4.2 Archivos clave en un repositorio.....	12
2.4.3 Seguridad y privacidad en GitHub	13
2.4.4 Licencias de repositorios en GitHub	14
2.4.5 Flujo de trabajo de proyectos basado en GitHub.....	15
3 OpenAIRE	16
3.1 Zenodo	17
3.2 Principios justos de Zenodo	18
3.3 Interacción de Zenodo con GitHub.....	19
3.4 Otros servicios de OpenAIRE.....	23
4 Conclusiones.....	24
4.1 Ventajas	24
4.2 Desventajas	24
5 Anexos	25
5.1 Anexo I: Puntos de referencia de las plataformas de sistemas de control de versiones.....	25
5.1.1 Características principales comparativas.....	25
5.1.2 Régimen de precios.....	26
5.1.3 Popularidad.....	26
5.2 Anexo II: Concesión de licencias	27
5.3 Anexo III: Recursos	28

Lista de cuadros

TABLA 1 - LICENCIAS DE GITHUB DISPONIBLES POR DEFECTO.....	15
TABLA 2 - SERVICIOS DE OPENAIRE (FUENTE)	23
CUADRO 3 - PRINCIPALES CARACTERÍSTICAS Y PROVEEDORES	25
TABLA 4 - PLANES Y PRECIOS DE GITHUB, GITLAB Y BITBUCKET	26

Lista de figuras

FIGURA 1 OPERACIONES HABITUALES DE GIT. VÍA WIKIMEDIA COMMONS	10
FIGURA 2 ESTRUCTURA DE UN PROYECTO BASADO EN GITHUB	15
FIGURA 3 MIEMBROS DE OPENAIRE A.M.K.E.	16
FIGURA 4 SECCIÓN DE ZENODO PARA IMPORTAR EL REPOSITORIO DE GITHUB	19
FIGURA 5 CARGA MANUAL DE ARCHIVOS EN ZENODO	20
FIGURA 6 SELECTOR DE TIPO DE ARCHIVO DE ZENODO	20
FIGURA 7 INFORMACIÓN BÁSICA REQUERIDA	20
FIGURA 8 EL DERECHO DE ACCESO Y LA LICENCIA SON METADATOS OBLIGATORIOS.....	21
FIGURA 9 OTROS METADATOS RECOMENDADOS Y OPCIONALES DISPONIBLES.....	21
FIGURA 10 CAMPOS DE METADATOS DE FINANCIACIÓN EN PROYECTOS CE / OPENAIRE	21
FIGURA 11 GITHUB VS. GITLAB VS. BITBUCKET GOOGLE TRENDS	26
FIGURA 12 EJEMPLO DE ESPECTRO BÁSICO DE LICENCIAS (ADAPTADO DE PERMISSIVE SOFTWARE LICENSE WIKI).....	27
FIGURA 13 COMPATIBILIDAD CON LA LICENCIA FLOSS (FUENTE: WIKIMEDIA COMMONS)	27
FIGURA 14 COMPATIBILIDAD DE LICENCIAS DE SOFTWARE PARA OBRAS DERIVADAS (FUENTE: WIKIMEDIA COMMONS)	28

Glosario

- **La Ciencia Abierta** es un enfoque del proceso científico que se centra en la difusión del conocimiento tan pronto como está disponible utilizando tecnología digital y colaborativa. Es una prioridad política de la Comisión Europea y el método de trabajo estándar de sus programas de financiación de la investigación y la innovación, ya que mejora la calidad, la eficiencia y la capacidad de respuesta de la investigación.
- **Software de código abierto (OSS)** se refiere a todo aquel software cuyo código fuente original se pone a libre disposición y puede redistribuirse y modificarse. Para otros productos distintos del software se utiliza genéricamente **Open Source (OS)** cuando la gente puede modificarlo y compartirlo porque su diseño es de acceso público. Es frecuente ver ambos conceptos intercambiados.
- **El acceso abierto (AA)** hace referencia al acceso gratuito a la información y al uso sin restricciones de los recursos electrónicos para todo el mundo.
- **Principios FAIR:** Los principios FAIR (Findability, Accessibility, Interoperability and Reusability) describen cómo deben organizarse los resultados de la investigación para que sea más fácil acceder a ellos, comprenderlos, intercambiarlos y reutilizarlos.
- **European Open Science Cloud (EOSC)** está reconocida por el *Consejo de la Unión Europea* entre las 20 acciones de la agenda política 2022-2024 del *Espacio Europeo de Investigación (EEI)* con el objetivo específico de profundizar en las prácticas de ciencia abierta en Europa. Su objetivo es desarrollar una "red de datos y servicios FAIR" para la ciencia en Europa sobre la que pueda construirse una amplia gama de servicios de valor añadido. Estos servicios van desde la visualización y el análisis hasta la conservación de la información a largo plazo o el seguimiento de la adopción de prácticas de ciencia abierta.
- **OpenAIRE:** OpenAIRE es un proyecto europeo de apoyo a *la Ciencia Abierta*. Por un lado, OpenAIRE es una red de expertos en ciencia abierta que promueven e imparten formación en este campo. Por otro, es también una infraestructura técnica que recopila resultados de investigación de proveedores de datos conectados.
- **Zenodo:** Se trata de un repositorio de investigación abierto y polivalente. Fue creado por OpenAire y el CERN ("Conseil Européen pour la Recherche Nucléaire", o "o Consejo Europeo de Investigación Nuclear") para ofrecer a los investigadores un lugar donde depositar publicaciones, conjuntos de datos y otros artefactos de investigación como códigos, carteles, presentaciones, etc.
- **DOI (Identificador de objetos digitales):** Cadena de números, letras y símbolos utilizada para identificar de forma permanente y única un artículo o documento, y enlazar con él en la web.
- **Concesión de licencias:** Formalmente, la *concesión de licencias* es el acto de otorgar a las personas un permiso oficial para hacer, tener o vender algo. En el mercado del código abierto, existen alrededor de cien licencias que cubren muchos casos de uso.
- **RGPD:** El Reglamento General de Protección de Datos (RGPD) es una legislación que actualizó y unificó las leyes de privacidad de datos en toda la Unión Europea (UE). El GDPR fue aprobado por el Parlamento Europeo el 14 de abril de 2016 y entró en vigor el 25 de mayo de 2018. El GDPR sustituye a la Directiva de Protección de Datos de la UE de 1995.
- **Metadatos:** Conjunto de datos que describen y dan información sobre otros datos.

Los metadatos nos ayudan a comprender la estructura, naturaleza y contexto de los datos. Los metadatos facilitan la búsqueda y recuperación de datos.

- **Sistemas de control de versiones (VCS):** Una categoría de herramientas de software capaz de registrar los cambios realizados en los archivos mediante el seguimiento de las modificaciones en el código.

- **Repositorio (o Repo):** Lugar donde se almacenan datos, archivos y paquetes de software.
- **Git:** un *sistema de control de versiones* distribuido, gratuito y de código abierto, diseñado para gestionar con rapidez y eficacia desde proyectos pequeños a muy grandes.
- **GitHub:** Un servicio de alojamiento web para proyectos de desarrollo de software que utiliza el sistema de control de revisiones Git, además de control de acceso, seguimiento de errores, solicitudes de características de software, gestión de tareas, integración continua y wikis para cada proyecto.
- **GUI:** Una *interfaz gráfica de usuario (GUI)* es una interfaz a través de la cual, los usuarios interactúan con dispositivos electrónicos como ordenadores y smartphones mediante iconos, menús y otros indicadores o representaciones visuales (gráficos).
- **IDE:** Un *entorno de desarrollo integrado (IDE)* es un software para crear aplicaciones que combina herramientas comunes para desarrolladores en una única interfaz gráfica de usuario (GUI). Suele incluir un editor de código fuente, automatización, depurador y una biblioteca coherente de plugins especializados.
- **API:** Una *interfaz de programación de aplicaciones (API)* es un conjunto de protocolos, rutinas, funciones y/o comandos utilizados para facilitar la interacción entre distintos servicios de software sin tener que saber cómo están implementados.
- **JSON (JavaScript Object Notation)** es un formato de intercambio de datos ligero para almacenar y transportar datos, fácil de leer y escribir para los humanos, independiente del lenguaje de programación y fácil de analizar y generar para las máquinas. JSON se utiliza a menudo cuando se envían datos desde un servidor a una página web.

1 Introducción

Este documento recopila alguna información básica sobre GitHub y su interacción con la infraestructura europea OpenAIRE, especialmente a través del repositorio Zenodo. No se trata de un análisis en profundidad de estos temas, pero si se necesitaran más detalles, se proporcionan algunos recursos a lo largo del texto; de todos modos el informe ha sido diseñado para ser coherente sin necesidad de visitar los enlaces.

Como ocurre con toda la industria moderna de Hardware y Software, todo puede cambiar a varios niveles y velocidades, incluyendo partes relevantes, flujos de trabajo, precios, licencias, etc. La mayoría de los enlaces, documentación y referencias se han obtenido según estaban en la primavera de 2022.

GitHub es el repositorio de software más popular para proyectos Open Source, y en este documento veremos que podría considerarse la opción natural para compartir no sólo código, si no también datos, documentos, esquemas, gráficos, etc.

En el Capítulo 2, revisamos los fundamentos de GitHub, los temas principales incluyen una comparación con su competidor directo hoy en día, es decir, GitLab.

El capítulo 3 está dedicado a la infraestructura de OpenAire y Zenodo.

Las ventajas e inconvenientes de utilizar estas metodologías se enumeran en el Capítulo 4.

Por último, se han añadido tres anexos. El primero es una rápida revisión de las principales soluciones actuales de *alojamiento de repositorios*, incluyendo características y precios. El segundo muestra una síntesis visual de las opciones de licencias de Código Abierto. Finalmente, el último ofrece una recopilación de recursos interesantes en Internet.

2 Fundamentos de GitHub

GitHub es principalmente un sistema colaborativo *de control de versiones* basado en [git](#)¹, un sistema *distribuido de control de versiones* de código abierto y gratuito ([GNU General Public License version 2](#)²).

La funcionalidad central de este tipo de herramientas es la gestión de cambios durante los procesos de desarrollo de software, pero también tiene algunas funciones *secundarias* que pueden ser adecuadas para los no programadores:

- cada proyecto tiene un control de acceso por usuario/equipo
- capacidades de colaboración
- herramientas de gestión de tareas
- seguimiento de problemas, no sólo errores, sino también ideas, comentarios o tareas
- publicación fácil (semiautomática) y segura de páginas web y wikis
- herramientas de redes sociales, comunicación y difusión fuera de la caja
- fácil integración en el flujo de trabajo mediante interfaces gráficas de usuario (GUI)

En el contexto de la infraestructura *OpenAIRE* (y específicamente en relación con los repositorios de Zenodo) GitHub está perfectamente integrado en ellos, y por lo tanto es una solución *por defecto* para muchos usuarios de la infraestructura.

Este capítulo muestra los fundamentos y las principales características actuales de esta herramienta.

2.1 Git

La descripción que figura en la [documentación oficial](#)³ es:

Git es un sistema de control de revisiones rápido, escalable y distribuido, con un conjunto de comandos inusualmente rico que proporciona tanto operaciones de alto nivel como acceso completo a las funciones internas.

Git funciona desde la *línea de comandos*, puede ejecutarse localmente desde cualquier ordenador, o en otro entorno como Internet o en cualquier red, y puede instalarse para los principales sistemas operativos visitando la [página de descargas de git](#).⁴ Además, Git está disponible gratuitamente en Internet. Git está bajo la *licencia de código abierto GPL*, lo que significa que puedes obtener su código fuente y modificarlo para adaptarlo a tus necesidades.

Básicamente, Git supervisa una carpeta local del proyecto, registra todos los cambios y, eventualmente, los exporta a un repositorio remoto (*Push*). Otros desarrolladores pueden colaborar en el proyecto importándolo (*Fork* o *Clone*), colaborando en algún área, y proponiendo incluir esos

¹ <https://git-scm.com/>

² <https://opensource.org/licenses/GPL-2.0>

³ <https://git-scm.com/docs/git.html>

⁴ <https://git-scm.com/downloads>

cambios a través de *Pull request*. Además, se pueden colocar diferentes versiones del software, o prototipos, etc. en diferentes *ramas*.

A continuación se muestra un esquema del uso típico de Git:

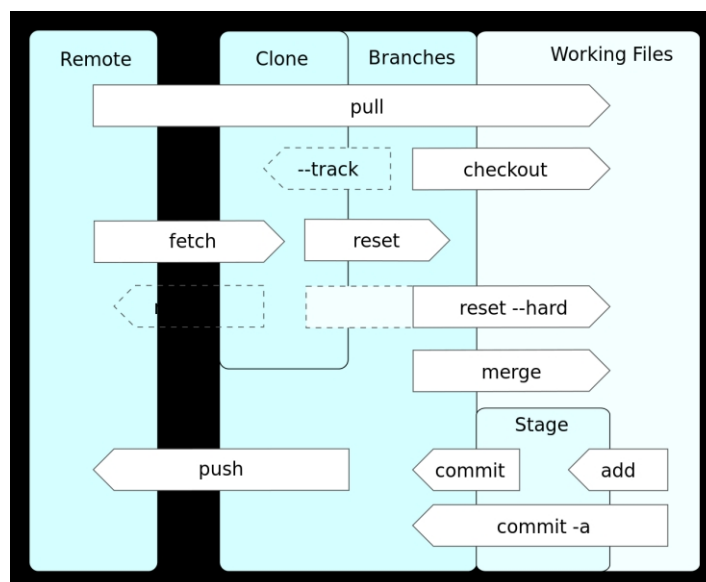


Figura 1 Operaciones habituales de Git. Vía Wikimedia Commons.⁵

Git es una herramienta poderosa pero tiene dos limitaciones críticas:

- Archivos binarios y grandes. Git se diseñó para procesar principalmente archivos de texto plano relativamente pequeños. Se necesita una *extensión de Git* para superar esta limitación: [Git Large File Storage \(LFS\)](https://git-lfs.github.com/).⁶
- Git se basa en comandos, lo que en general hace que el flujo de trabajo no sea amigable para las funciones de los no programadores.

2.2 Interfaces gráficas de usuario (GUI) para Git

El propio Git viene con un par de herramientas GUI para confirmar ([git-gui](https://git-scm.com/docs/git-gui)⁷) y explorar ([gitk](https://git-scm.com/docs/gitk)⁸) repositorios. Dado que Git se publica bajo una *licencia de código abierto* ([GNU GPLv2](https://www.gnu.org/licenses/gpl-3.0.html)⁹), muchas empresas, tanto de *código abierto* como *privadas*, han estado diseñando y manteniendo interfaces gráficas que ofrecen a los usuarios un uso más visual e intuitivo de Git.

⁵ https://commons.wikimedia.org/wiki/File:Git_operations.svg

⁶ <https://git-lfs.github.com/>

⁷ <https://git-scm.com/docs/git-gui>

⁸ <https://git-scm.com/docs/gitk>

⁹ <https://opensource.org/licenses/GPL-2.0>

Algunas *soluciones propietarias* populares son, por ejemplo, GitKraken, Sublime Merge, Sourcetree, etc.

En el ámbito del *código abierto*, la solución más popular es [GitHub Desktop](#)¹⁰.

Además, la mayoría de los *editores de código fuente* y *Entornos de Desarrollo Integrado* (IDE) son compatibles con Git (y otras) herramientas de control de versiones.

El [sitio](#) oficial [de Git enumera las GUIs más populares](#)¹¹ acortadas por sistema operativo. Además, Wikipedia ofrece una extensa lista de GUIs [Git de escritorio](#)¹² también.

2.3 GitHub y GitLab

GitHub y Gitlab son hoy en día los principales proveedores de repositorios nativos de *alojamiento en la nube con control de versiones* integrado a través de Git. Ambos son muy similares en sus *funcionalidades, esquema de precios y licencias*, y ambos servicios superan (al menos parcialmente) las limitaciones de Git.¹³

Además, ambas plataformas añaden otras *funcionalidades clave* para los desarrolladores modernos (*CI/CD, DevOps, seguimiento de errores...*), y otras herramientas multidisciplinarias (gestión de proyectos, edición wiki, desarrollo web y alojamiento seguro, redes sociales...) adecuadas también para otros roles, como directivos, científicos de datos, desarrolladores web, comunicación corporativa, edición de libros, almacenamiento de recursos, networking profesional, etc.

La elección de uno u otro debe hacerse caso por caso; sin embargo, en el escenario de la interacción con *OpenAIRE / Zenodo*, GitHub surge como una elección natural dado que ya está integrado en Zenodo.

En caso de que se necesite GitLab, no surgen mayores problemas, y el editor podría elegir entre subir manualmente los archivos a través de la plataforma Zenodo, o incluso importar el repositorio de GitLab a GitHub y luego proceder con la herramienta de importación de repositorios GitHub de Zenodo.

De hecho, la mayor parte de lo que se dice en el capítulo siguiente (2.4) sobre GitHub podría decirse también de GitLab.

2.4 Funcionalidades de GitHub

2.4.1 Conceptos clave

A continuación se recogen los conceptos clave para entender los procesos GitHub:

¹⁰ <https://desktop.github.com/>

¹¹ <https://git-scm.com/downloads/guis/>

¹² https://en.wikipedia.org/wiki/Comparison_of_Git_GUIs

¹³ Para una comparativa completa de ambos sitios (actualizada a 2022),
visite: <https://slashdot.org/software/comparison/GitHub-vs-GitLab/>

- **Repositorio (o Repo).** - Una ubicación central de almacenamiento de archivos supervisada por un *control de versiones* para almacenar múltiples versiones de archivos. Puede configurarse en un equipo local para una única carpeta, en un servidor o en la nube. Los usuarios pueden crear dos tipos de repositorios en *GitHub*:
 - *Repositorios públicos.* - Cualquiera puede acceder a ellos y utilizar el contenido (de acuerdo con la *licencia* proporcionada en el repositorio).
 - *Repositorios privados.* - Sólo el *propietario / miembros del equipo* pueden interactuar con él.
- **Confirmar.** - Cuando cualquier archivo cambia en el repositorio, el usuario puede *confirmar* ese cambio en el repositorio *local* o *remoto*. Después de eso, los repositorios pueden ser actualizados en ambas direcciones:
 - *Tirar.* - Transfiere *los commits* realizados o almacenados en un repositorio remoto a un repositorio local.
 - *Empujar.* - Transfiere *los commits* realizados o almacenados localmente a un repositorio remoto.
- **Recuperar.** - Añade cambios de un repositorio remoto a la rama de trabajo local sin confirmarlos.
- **Rama.** - Una *rama* permite trabajar aislado de los archivos actuales o de producción. Si los cambios realizados funcionan, el usuario puede enviar una *Pull request* al propietario y colaboradores que podrán *fusionar* los cambios con la *rama principal*.
- **Fork.** - Cualquiera con acceso a un repositorio puede *bifurcarlo*, es decir, hacer una copia del repositorio original en *GitHub* (y luego localmente si es necesario) para que el usuario pueda trabajar en él, y luego enviar al propietario los cambios a través de *Pull request*. Si los cambios son aprobados, el propietario puede integrarlos en la *rama ascendente* del proyecto.
- **Clonar.** - Similar al *fork*, hace una copia del repositorio original en una máquina local. De esta forma, los colaboradores pueden sincronizar fácilmente los cambios tanto en el repositorio local como en el remoto.
- **Gist.** - Puede considerarse como un repositorio de *un solo archivo*, pueden ser fragmentos de código, o un archivo de configuración, o una tabla de datos o incluso una lista de tareas. Por lo demás, tiene las principales características de un repositorio ordinario (control de versiones integrado, bifurcación y clonación, Gists públicos y secretos, flujos de trabajo similares, etc.).
- **Markdown.** - Aunque no es obligatorio, *Markdown* es la sintaxis por defecto en la plataforma *GitHub*. Puedes consultar los detalles en la *página de documentación: [Acerca de la escritura y el formato en GitHub](#)*¹⁴.
- **GitHub Marketplace.** - Es fácil ampliar las *funcionalidades de GitHub* mediante

[todo tipo de aplicaciones](#)¹⁵. Para más información básica, *GitHub Docs* ofrece un [glosario completo de GitHub](#)¹⁶.

2.4.2 Archivos clave de un repositorio

La mayoría de los repositorios contienen algunos archivos singulares que el usuario debe **c o n o c e r**, por ser de especial importancia:

¹⁴ <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/about-writing-and-formatting-on-github>

¹⁵ <https://github.com/marketplace>

¹⁶ <https://docs.github.com/en/get-started/quickstart/github-glossary>

- **README.** - El archivo README.md (si se utiliza el formato markdown, aunque se admite cualquier texto plano, por ejemplo README.txt -Lo mismo ocurre con la mayoría de los archivos que aparecen a continuación-), es la primera información que pueden ver los visitantes del repositorio. Por lo tanto, se utiliza a menudo para explicar el proyecto, mostrar su grado de madurez, señalar archivos o carpetas de documentación, o cualquier información que el propietario pueda considerar relevante para los visitantes o usuarios. Además, un archivo README puede añadirse al perfil personal/de la organización, convirtiéndose en una portada para el proyecto, producto o empresa.
- **LICENCIA.** - El archivo LICENSE muestra la licencia legal del repositorio, una información importante cuando el repositorio es público y es necesario preservar algunos derechos, o cuando su proyecto contiene trabajo de otras personas con su propia licencia. En el *capítulo 2.4.4* y en el *anexo II* se ofrece más información.
- **CITATION.cff.** - Explica cómo quieren los autores que se cite su proyecto. Aunque también es un archivo de texto plano, se necesita algún formato para que sea legible por máquina. Consulta los detalles en los documentos de GitHub: [Acerca de los archivos CITATION](#).¹⁷
- **.gitignore.** - Este archivo es muy útil cuando el usuario está trabajando localmente, contribuyendo al proyecto a través de pull-requests, pero necesita mantener ciertos archivos y/o carpetas fuera del control de versiones, es decir, ignorarlos no incluyéndolos en el repositorio de GitHub. Un ejemplo se puede ver en el *capítulo 2.4.3*.
- **CÓDIGO_DE_CONDUCTA.** - Cuando el repositorio está orientado a la comunidad y muchas personas interactúan en el proyecto, un texto claro sobre el comportamiento esperado de los participantes puede ser fundamental.

Dependiendo de la complejidad del proyecto, o del número de participantes, etc. otros archivos pueden ser también importantes: **AUTORES, CHANGELOG, CONTRIBUYENTES, AGRADECIMIENTOS,** etc.

2.4.3 Seguridad y privacidad en GitHub

GitHub cuenta con buenas políticas de privacidad (incluido el GDPR de la UE), seguridad, auditoría y cumplimiento. Puede encontrar amplia documentación sobre esta dimensión de los servicios ofrecidos en la [página de seguridad de GitHub](#).¹⁸

¹⁷ <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-citation-files>

¹⁸ <https://github.com/security>

Se puede añadir otra capa de seguridad configurando [el protocolo SSH](#)¹⁹, que permite autenticar a los usuarios y darles acceso a un repositorio determinado (privado o público). Esta opción debería usarse en casos en los que haya datos sensibles implicados.

En términos de privacidad, el factor humano es central porque las mayores vulnerabilidades provienen del mal uso de la gestión de archivos y/o datos sensibles, algo que escala rápidamente cuando muchas personas tienen acceso a uno o varios repositorios públicos. Los repositorios Git disponen de un método para evitar que determinados archivos y carpetas puedan ser terminados supervisados por la versión de control y/o, posteriormente, ser compartidos a repositorios remotos. Consiste simplemente en excluir archivos, carpetas o determinados patrones mediante un único archivo de texto plano (*.gitignore*) que lista todas las carpetas/archivos a excluir.

Por ejemplo:

```
## Archivos temporales de Word
*.tmp
~$*.doc*

## Archivos/carpetas
privados Aux*.*

/no compartido/
/Docs/embargoed

## gitignore itself
.gitignore
```

Para más información sobre *.gitignore*, visita por ejemplo la [documentación de git](#)²⁰.

Puede consultar una colección de plantillas *.gitignore* en este repositorio²¹.

2.4.4 Licencias de repositorios en GitHub

Aunque el propietario del repositorio puede optar por cualquier licencia, Github incluye automáticamente el texto legal apropiado para una serie de licencias populares y, por defecto, asigna la autoría al propietario del repositorio.

Un método típico para clasificar las licencias está relacionado con lo permisivo que es el software (u otros productos digitales), y de esta forma se presenta en la siguiente tabla.

¹⁹ <https://docs.github.com/en/github-ae@latest/authentication/connecting-to-github-with-ssh/about-ssh>

²⁰ <https://git-scm.com/docs/gitignore>

²¹ <https://github.com/github/gitignore>

Dominio público

- Creative Commons Zero v1.0 Universal
- Sin licencia

Permisivo	Menos permisivos
• Licencia Apache 2.0	• Licencia Pública General de GNU v3.0
• Licencia MIT	• Licencia pública general Affero de GNU v3.0
• Licencia BSD "simplificada" de 2 cláusulas	• Licencia Pública General GNU v2.0
• Licencia BSD de 3 cláusulas "nueva" o "revisada"	• Licencia Pública General Reducida de GNU v2.1
• Licencia de software Boost 1.0	• Licencia pública Eclipse 2.0
• Licencia pública Mozilla 2.0	

Tabla 1 - Licencias de GitHub disponibles por defecto

La tarea de elegir la licencia correcta no siempre es fácil, GitHub mantiene un sitio que puede ayudar a elegir la correcta: [Elegir una licencia de código abierto | Elegir una licencia](#)²²

En el Anexo II: *Licencias* se puede consultar un rápido repaso a la situación actual de las licencias de código abierto.

2.4.5 Flujo de trabajo del proyecto basado en GitHub

El uso de GitHub en un proyecto supone en cierto modo un *cambio de paradigma*, ya que en el flujo de trabajo podrían integrarse varios actores, entornos y roles, incluidos usuarios y comunidades, editores, revisores, público en general, etc. El informe "Ten Simple Rules for Taking Advantage of Git and GitHub"²³ - CC BY 4.⁰²⁴ muestra un ejemplo de trabajo con GitHub.

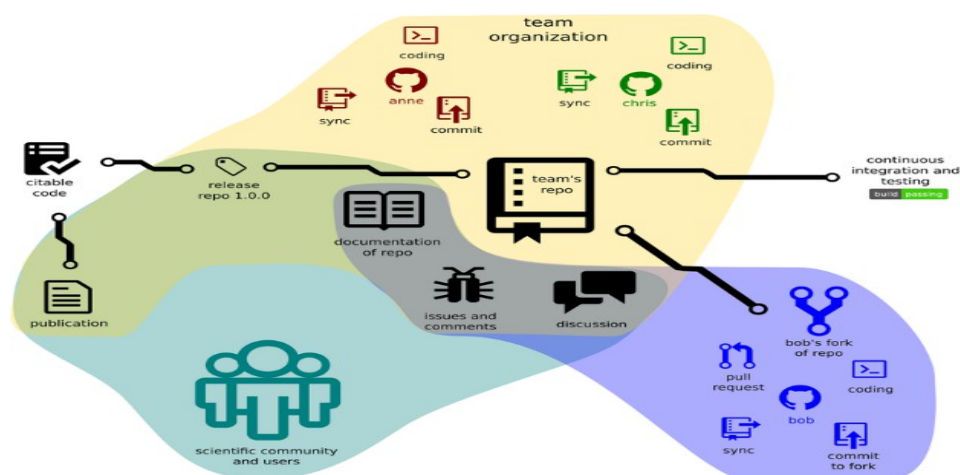


Figura 2 Estructura de un proyecto basado en GitHub

²² <https://choosealicense.com/>

²³ <https://doi.org/10.1371/journal.pcbi.1004947>

²⁴ <https://creativecommons.org/licenses/by/4.0/>

3 OpenAIRE

En la [página Acerca de](#)²⁵ podemos leer:

OpenAIRE es una asociación sin ánimo de lucro, establecida en 2018 como una entidad legal, OpenAIRE A.M.K.E, para garantizar una infraestructura de comunicación académica abierta permanente para apoyar la investigación europea.

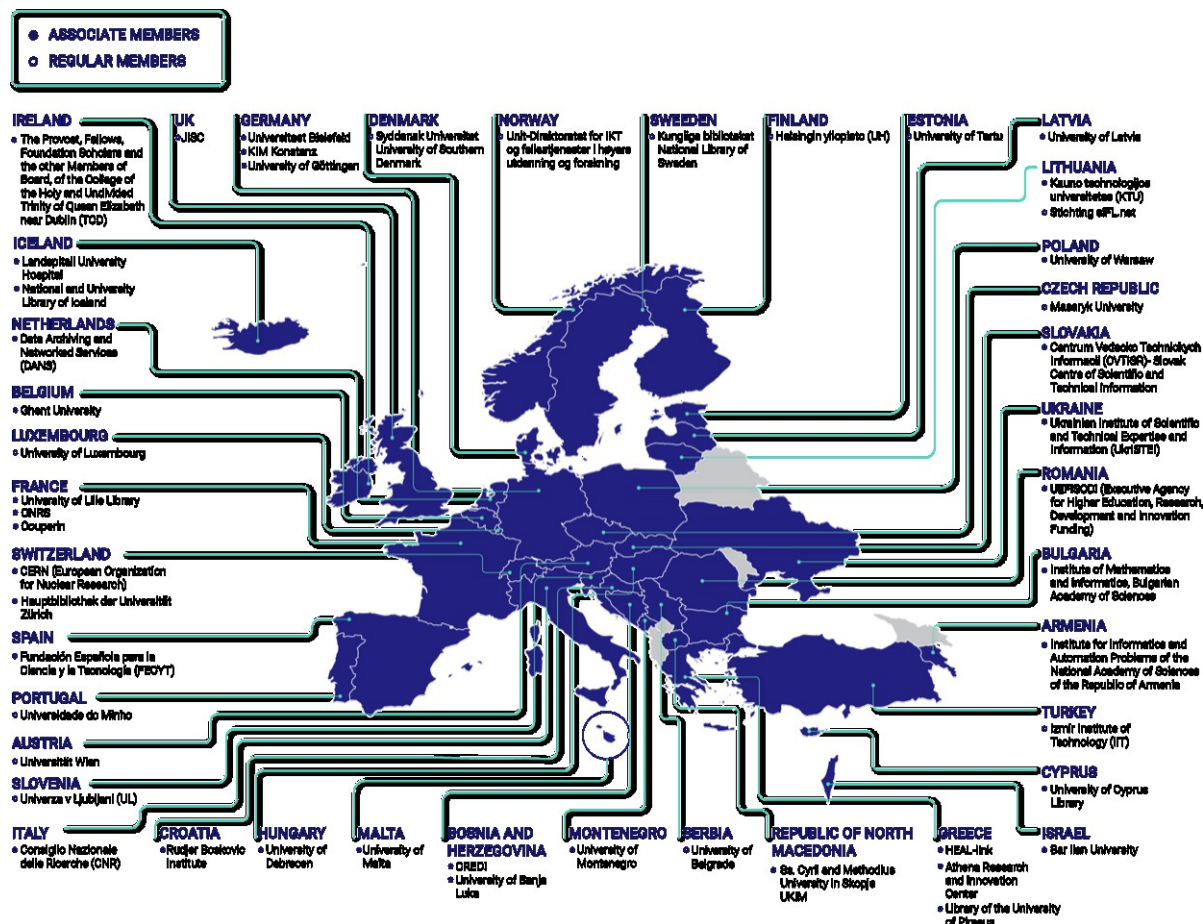


Figura 3 Miembros de OpenAIRE A.M.K.E.

La [misión de OpenAIRE](#)²⁶ se describe así:

La misión de OpenAIRE está estrechamente vinculada a la misión de la Comisión Europea: proporcionar un acceso abierto, ilimitado y sin barreras a los resultados de la investigación financiada con fondos públicos en Europa. OpenAIRE cumple sustancialmente la visión de la EOSC, ya que sus operaciones ya proporcionan el pegamento para muchas de las funcionalidades impulsadas por los usuarios y la investigación, ya provengan de la larga cola de la ciencia (repositorios y soporte local) o de comunidades de investigación disciplinadas por dominios o Infraestructuras de Investigación.

²⁵ <https://www.openaire.eu/about>

La [Nube Europea de la Ciencia Abierta \(EOSC\)](#)²⁷ comparte de hecho con OpenAIRE objetivos como el acceso sin fisuras, la gestión FAIR (Findability, Accessibility, Interoperability and Reusability) y la reutilización fiable de los datos de investigación y todos los demás objetos digitales producidos a lo largo del ciclo de vida de la investigación (por ejemplo, métodos, software y publicaciones).

OpenAIRE contribuye activamente a través de su amplia red, apoyando las políticas de Código Abierto y ayudando a través de los [National Open Access Desks \(NOADs\)](#)²⁸. Puede consultar más información sobre la contribución de OpenAIRE a EOSC en el documento "Achieving Open Science in the European Open Science Cloud: Setting out OpenAIRE's vision and contribution to EOSC" ([DOI | 10.5281/zenodo.3610132](#))²⁹.

3.1 Zenodo

Zenodo es un repositorio de datos abierto que funciona con el *software de código abierto Invenio*³⁰, creado y mantenido por el CERN y que permite a los investigadores compartir y preservar cualquier resultado de investigación de cualquier tamaño, formato y de cualquier ciencia / campo.

Dado que Zenodo forma parte de la infraestructura de OpenScience.eu, veamos algunos extractos de OpenScience.eu sobre Zenodo³¹:

Zenodo se puso en marcha en el marco del proyecto OpenAIRE, encargado por la Comisión Europea para facilitar el acceso abierto a los resultados de la investigación financiada con fondos públicos en Europa.

[...]

Como socio de OpenAIRE, la Organización Europea para la Investigación Nuclear (CERN) creó Zenodo en 2013 como un servicio de repositorio de datos "general" para los investigadores de la UE. Cualquiera que tenga resultados de investigación puede depositarlos (por ejemplo, artículos, conjuntos de datos, imágenes, carteles, software).

[...]

Zenodo acepta archivos de gran tamaño (hasta 50 GB) sin restricción de formato. A los datos se les asigna un Identificador de Objeto Digital (DOI), lo que significa que no necesita publicar en una revista para obtener un DOI único y permanente, lo que hace que su publicación sea más fácilmente citable según las normas internacionales.

²⁷ https://ec.europa.eu/info/research-and-innovation/strategy/strategy-2020-2024/our-digital-future/open-science/european-open-science-cloud-eosc_es

²⁸ <https://www.openaire.eu/openaire-and-eosc>

²⁹ <https://zenodo.org/record/3610132>

³⁰ <https://invenio-software.org/>

³¹ <https://openscience.eu/zenodo/>

3.2 Principios justos de Zenodo

Zenodo se ajusta a la definición de **los Principios FAIR** tal y como se recoge en *Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data 3:160018 doi: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18) (2016)*³².

A continuación se enumeran las políticas clave de estos principios, que pueden consultarse en la página de [los Principios de Zenodo](https://about.zenodo.org/principles/)³³:

- **Ser localizable.** - Los principios FAIR promueven un uso extensivo de *metadatos* siguiendo los términos recomendados del [Esquema de Metadatos de DataCite](https://datacite.org/schema/)³⁴, incluyendo el DOI como campo de *metadatos obligatorio*, siendo entonces registrado e indexado, y por lo tanto inmediatamente localizable directamente en el motor de búsqueda de Zenodo tras su publicación.
- **Ser accesibles.** - Para ser accesibles, los metadatos y los datos deben ser comprensibles tanto para las personas como para las máquinas, y los datos deben almacenarse en un repositorio de confianza. Los metadatos de registros individuales y colecciones pueden obtenerse por dos métodos:
 - mediante el [protocolo OAI-PMH](https://www.oai.org/)³⁵ por el identificador del registro y el nombre de la colección
 - a través de la [API REST](https://zenodo.org/api/) pública³⁶
- **Ser interoperables.** - Para ser *interoperables*, los metadatos deben utilizar un lenguaje formal, accesible, compartido y ampliamente aplicable para la representación del conocimiento, como los vocabularios controlados acordados. Zenodo utiliza el [esquema JSON](https://datacite.org/schema/)³⁷ como representación interna de los metadatos, y está preparado para la exportación a otros formatos de *esquemas internacionales*.
- **Ser reutilizable.** - En este aspecto se incluyen algunas características y políticas:
 - La licencia es uno de los términos obligatorios en los metadatos de Zenodo, y se refiere a una licencia de [Definición Abierta](https://creativecommons.org/licenses/by/4.0/)³⁸ (sucintamente, "Los datos y contenidos abiertos pueden ser utilizados, modificados y compartidos libremente por cualquiera para cualquier propósito").
 - Los datos descargados por los usuarios están sujetos a la licencia especificada en los metadatos por el propietario.
 - Todos los datos y metadatos cargados son trazables hasta un *usuario* registrado de Zenodo.

³² <https://doi.org/10.1038/sdata.2016.18>

³³ <https://about.zenodo.org/principles/>

³⁴ <https://schema.datacite.org/>

³⁵ <https://zenodo.org/oai2d>

³⁶ <https://developers.zenodo.org/>

³⁷ <https://json-schema.org/>

³⁸ <https://opendefinition.org/>

- Zenodo no es un repositorio de dominio específico, pero, gracias a su conformidad con el esquema de metadatos de DataCite³⁹, cumple una de las normas más amplias disponibles para todos los dominios.

Los datos de metadatos y los propios metadatos se alojarán durante toda la vida útil del repositorio. Ésta es actualmente la vida útil del laboratorio anfitrión, el CERN, que tiene un programa experimental definido para los próximos 20 años como mínimo. Los metadatos se almacenan en servidores de bases de datos de alta disponibilidad en el CERN, que están separados de los propios datos.

3.3 Interacción de Zenodo con GitHub

GitHub es hoy la principal plataforma para desarrolladores de software (más de 70 millones de usuarios) y en ella se alojan muchos proyectos importantes *de código abierto*, como [los proyectos de la Comisión Europea](#)⁴⁰, o los repositorios [OpenAIRE](#)⁴¹ y [Zenodo](#)⁴².

En este escenario es perfectamente natural que un repositorio como Zenodo haga todo lo posible para facilitar la compatibilidad con GitHub. Por ejemplo, un usuario puede registrarse fácilmente en Zenodo a través de cuentas de GitHub (o alternativamente a través de ORCID)⁴³, y luego importar fácilmente a Zenodo sus repositorios públicos de GitHub.⁴⁴

Una vez que el usuario/editor ha iniciado sesión, Zenodo puede acceder fácilmente a la [sección del importador de repositorios de Github](#)⁴⁵.

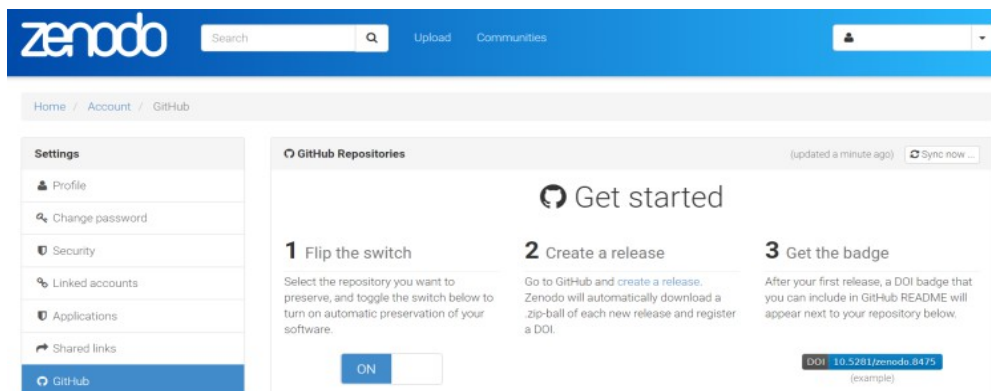


Figura 4 Sección de Zenodo para importar el repositorio de Github

³⁹ <https://schema.datacite.org/meta/kernel-4.4/>

⁴⁰ <https://github.com/ec-europa>

⁴¹ <https://github.com/openaire>

⁴² <https://github.com/zenodo>

⁴³ <https://www.zenodo.org/login/>

⁴⁴ <https://zenodo.org/account/settings/github/> - Se necesita una cuenta de Zenodo

⁴⁵ <https://www.zenodo.org/account/settings/github/> - Se necesita una cuenta de Zenodo

Alternativamente, el editor puede hacerlo [cargando manualmente](#)⁴⁶ los archivos pertinentes.

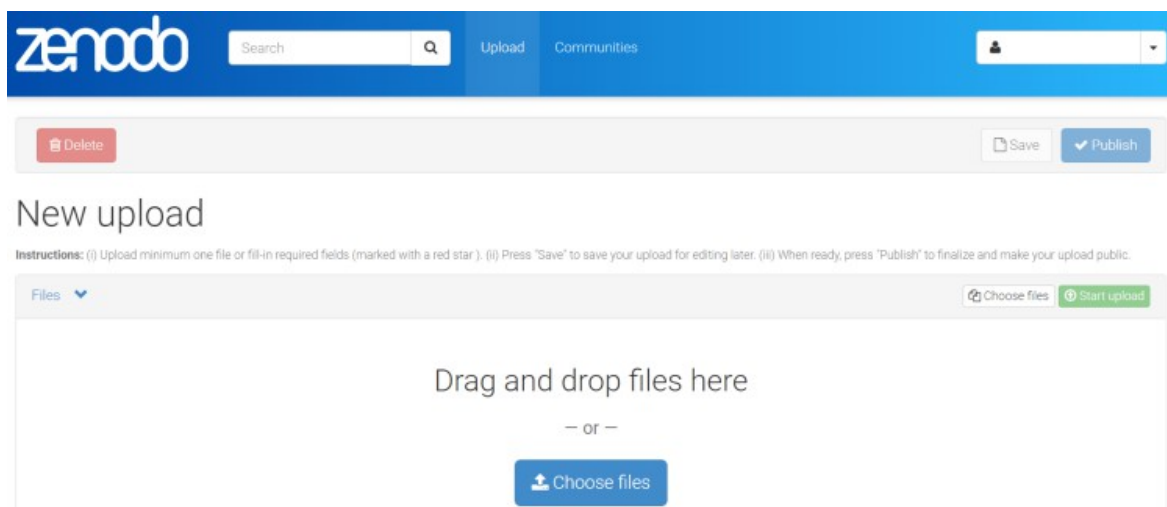


Figura 5 Carga manual de archivos en Zenodo

En este caso, el usuario también debe cargar manualmente los metadatos, pero el sistema Zenodo facilita la tarea mediante menús visuales. A continuación se muestran algunas capturas del proceso.

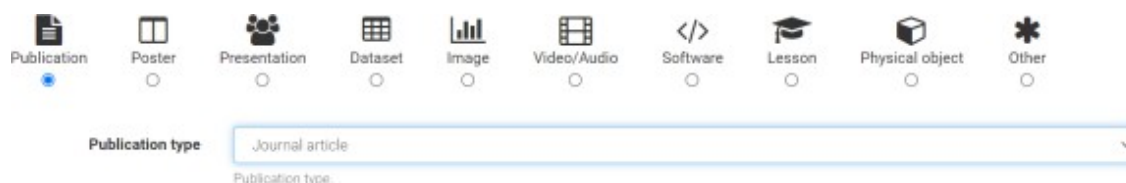


Figura 6 Selector de tipo de archivo de Zenodo

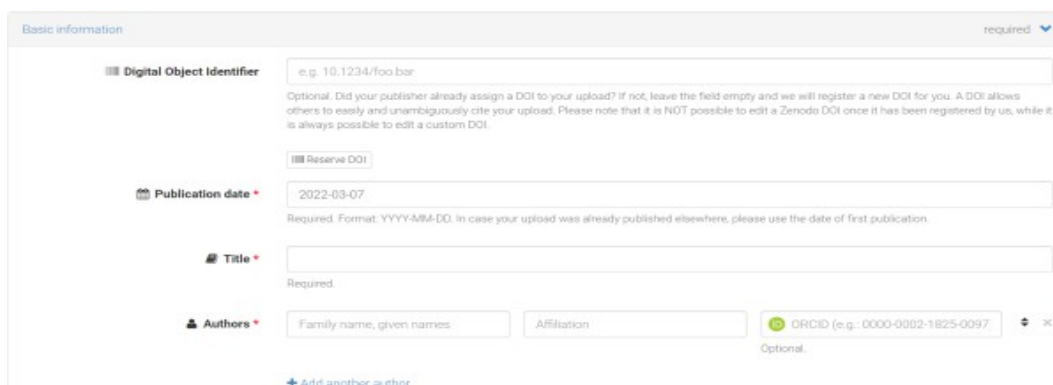


Figura 7 Información básica requerida

⁴⁶ <https://www.zenodo.org/deposit/new> - Se necesita una cuenta de Zenodo

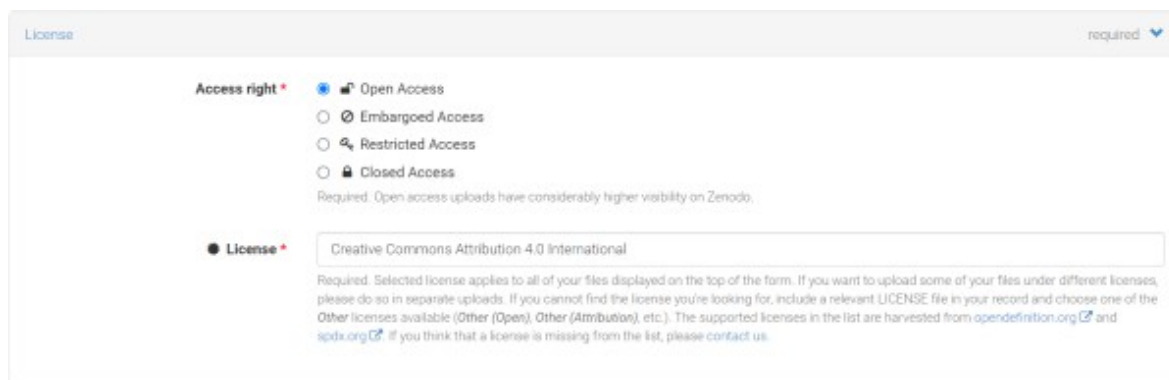


Figura 8 El derecho de acceso y la licencia son metadatos obligatorios

Aparte de los campos de metadatos obligatorios, Zenodo ofrece algunos campos recomendados y opcionales, que pueden mejorar la localización de los documentos en los motores de búsqueda de Zenodo y OpenAIRE. Dependiendo del tipo de documento, puede ser aconsejable rellenar tantos campos como sea posible.



Figura 9 Otros metadatos recomendados y opcionales disponibles

Por ejemplo, en el contexto de los proyectos de la Comisión Europea (a través de OpenAIRE), probablemente la *sección de financiación* podría ser requerida aún no siendo obligatoria para Zenodo.

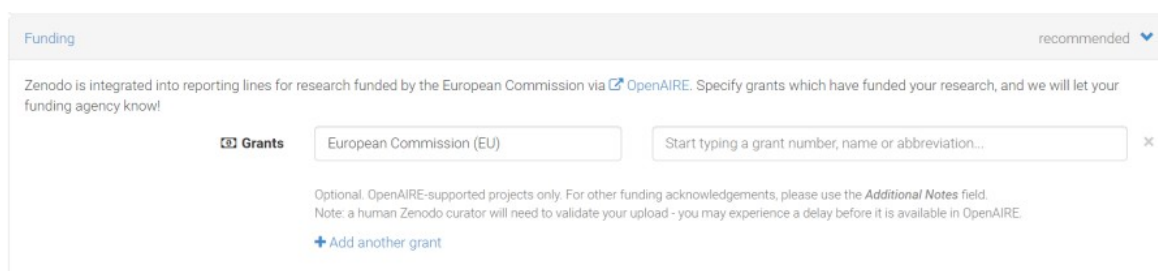


Figura 10 Campos de metadatos de financiación en proyectos CE / OpenAIRE

Los metadatos de GitHub se extraen automáticamente, pero si se necesita alguna personalización, debe incluirse un archivo (*.zenodo.json*) en la raíz del repositorio.⁴⁷ Se trata de un sencillo archivo de texto plano fácilmente legible y/o editable por humanos. Abajo y ejemplo arbitrario.

```
{
  "creadores": [
    {
      "orcid": "0000-0002-1825-0097",
      "afiliación": "Instituto de Investigación
Felina", "nombre": "Field, Gar"
    },
    {
      "orcid": "0000-0002-1825-0097",
      "afiliación": "Instituto de investigación
felina", "nombre": "Gato, Félix"
    }
  ],
  "licencia": "Apache-2.0",
  "título": "Red-Dot: Detección vectorial láser potenciada por ML",
  "related_identifiers": [
    {
      "esquema": "doi",
      "identificador": "10.1234/software.paper.5678",
      "relation": "isDocumentedBy",
      "resource_type": "publication-article"
    }
  ],
  "keywords": ["Gatos", "Láser", "Comportamiento"],
  "comunidades": [
    {"identificador": "software-cats"}
  ],
  "subvenciones": [{"id": "777541"}]
}
```

Esta funcionalidad facilita la gestión de la información y metadatos apropiados en cada repositorio a enlazar con Zenodo. A la hora de incluir el archivo *.zenodo.json* se debe comprobar la [documentación de depósito de entidades](https://developers.zenodo.org/#entities)⁴⁸, y utilizar alguna herramienta para comprobar el formato del archivo, por ejemplo [JSON Formatter & Validator](https://developers.zenodo.org/#github)⁴⁹.

⁴⁷ <https://developers.zenodo.org/#github>

⁴⁸ <https://developers.zenodo.org/#entities>

⁴⁹ <https://jsonformatter.curiousconcept.com/>

3.4 Otros servicios de OpenAIRE

Zenodo es sólo una parte de la infraestructura OpenAIRE, una lista de otros servicios gratuitos orientados a adoptar fácilmente la Ciencia Abierta, se puede ver a continuación:

Servicio	Lema/Descripción
<ul style="list-style-type: none"> • Argos: para administradores ⁵⁰ 	Acerque sus planes de gestión de datos al lugar donde se generan, analizados y almacenados.
<ul style="list-style-type: none"> • OpenCitations ⁵¹ 	Organización independiente sin ánimo de lucro dedicada a la publicación de material bibliográfico y de investigación abierto. datos de citas mediante el uso de tecnologías de la web semántica (Linked Data).
<ul style="list-style-type: none"> • Gráfico de investigación ⁵² 	Recurso abierto que agrega una colección de propiedades de datos de investigación (metadatos, enlaces) disponibles dentro de la infraestructura de OpenAIRE Open Science para que financiadores, organizaciones, investigadores, comunidades de investigación y editores interconecten información utilizando una semántica enfoque de base de datos gráfica.
<ul style="list-style-type: none"> • Proporcione ⁵³ 	El Panel de control de proveedores es un servicio integral en el que los proveedores de contenidos interactúan con OpenAIRE y se convierten en parte integrante de un OpenAIRE global. Comunidad investigadora. Una puerta de acceso a la Nube Europea de la Ciencia Abierta.
<ul style="list-style-type: none"> • Amnesia ⁵⁴ 	Realice investigaciones y comparta sus resultados que cumplan las directrices del GDPR utilizando algoritmos de anonimización de datos.
<ul style="list-style-type: none"> • Conectar ⁵⁵ 	Cree un portal de acceso a los resultados de la investigación abierta y vinculada de su comunidad
<ul style="list-style-type: none"> • Recuentos de uso ⁵⁶ 	Recopilación de datos de uso de los repositorios de proveedores de contenidos de Open Science, y otras fuentes de datos científicos.
<ul style="list-style-type: none"> • Monitor ⁵⁷ 	Colabore con nosotros para construir indicadores que se adapten a sus necesidades y para ver, comprender y visualizar estadísticas e indicadores de investigación.
<ul style="list-style-type: none"> • Zenodo ⁵⁸ 	Un repositorio abierto y fiable para toda la comunidad científica, que permite a los investigadores de todas las disciplinas compartir y preservar los resultados de sus investigaciones, independientemente de su tamaño o formato. Zenodo hace que los resultados científicos de todo tipo sean citables, compartibles y descubribles. a largo plazo.

Tabla 2 - Servicios de OpenAIRE ([fuente](#)) ⁵⁹

⁵⁰ <https://argos.openaire.eu/>

⁵¹ <https://opencitations.net/>

⁵² <https://graph.openaire.eu/>

⁵³ <https://provide.openaire.eu/>

⁵⁴ <https://amnesia.openaire.eu/>

⁵⁵ <https://connect.openaire.eu/>

⁵⁶ <https://usagecounts.openaire.eu/>

⁵⁷ <https://monitor.openaire.eu/>

⁵⁸ <https://www.zenodo.org/>

⁵⁹ <https://www.openaire.eu/factsheets>

4 Conclusiones

El uso de GitHub y Zenodo en el contexto de OpenAIRE garantiza los estándares y políticas europeas de Open Science respecto a los resultados de la investigación. Su aplicación en proyectos actuales (o futuros) puede presentar pros y contras que se pueden resumir en los siguientes puntos.

4.1 Ventajas

- Asignación gratuita de DOI a registros públicos e indexación automática en OpenAIRE
- Versionado automático de código, conjuntos de datos, documentos, informes, etc.
- Citaciones automáticas
- Mejor y más fácil difusión de contenidos
- Estadísticas de uso de repositorios
- Cumplimiento de las normas de seguridad y privacidad
- 20 años de persistencia de datos y metadatos en el Centro de Datos del CERN

4.2 Desventajas

- Es necesario afrontar algunos cambios de paradigma en el flujo de trabajo, y la curva de aprendizaje podría ser escalonada para determinadas funciones
- Carga de trabajo adicional para desarrolladores, autores y gestores
- Los repositorios y, por lo tanto, los contenidos quedan expuestos a Internet en la naturaleza
- Necesidad de algún tipo de mantenimiento de los repositorios
- La elección de licencias puede ser conflictiva y llevar mucho tiempo
- Precaución adicional para el [cumplimiento del GDPR](#)⁶⁰.






















⁶⁰ <https://gdpr.eu/>

5 Anexos

5.1 Anexo I: Puntos de referencia de las plataformas de sistemas de control de versiones

La información relevante de las tres soluciones más populares hoy en día en el ámbito del alojamiento de código y control de versiones mediante *Git* son **GitHub**, **GitLab** y **BitBucket**.

5.1.1 Características principales comparativas

Características	GitHub	GitLab	BitBucket
Número de usuarios	73 millones de euros	30 millones	10 millones
Repositorios privados gratuitos			
Repositorios públicos gratuitos			
Usabilidad de la navegación			
Interfaz gráfica de usuario (GUI)	 Escritorio GitHub gratuito y de código abierto ⁶¹		 Árbol de fuentes gratuito pero patentado ⁶²
Almacenamiento de archivos de gran tamaño			
Código abierto			
3 rd herramientas de terceros Integración			
Alojamiento de sitios estáticos	Páginas de GitHub ⁶³	Páginas de GitLab ⁶⁴	Bitbucket Cloud ⁶⁵ (con limitaciones), o mediante un plugin comercial: Páginas para Bitbucket Server ⁶⁶

Cuadro 3 - Principales características y principales proveedores (2022)

⁶¹ <https://desktop.github.com/>

⁶² <https://www.sourcetreeapp.com/>

⁶³ <https://pages.github.com/>

⁶⁴ <https://about.gitlab.com/stages-devops-lifecycle/pages/>

⁶⁵ <https://support.atlassian.com/bitbucket-cloud/docs/publishing-a-website-on-bitbucket-cloud/>

⁶⁶ <https://marketplace.atlassian.com/apps/1212525/pages-for-bitbucket-server>

5.1.2 Régimen de precios

Planes actuales - Primavera de 2022



Planes	GitHub	GitLab	BitBucket
Gratis			
Medio	<i>Equipo</i> 44 \$ por usuario/año	<i>Premium</i> 19 \$ por usuario/mes	<i>Estándar</i> 3 \$ por usuario/mes Desde 15 \$ /mes
Top	<i>Empresa</i> 231 \$ por usuario/año	<i>Lo último en</i> 99 \$ por usuario/mes	<i>Premium</i> 6 \$ por usuario/mes Desde 30 \$ /mes
Autoalojado / Autogestionado	<i>Servidor de empresa</i> 231 \$ por usuario/año	<i>Para todos los niveles</i> Comparación de funciones autogestionadas ⁶⁷	<i>Centro de datos</i> 2300 \$ por 25 usuarios/año
Detalles	Precios - Planes para cada promotor ⁶⁸	Precios de ^{GitLab} ⁶⁹	Bitbucket - Precios ⁷⁰

Tabla 4 - Planes y precios de GitHub, GitLab y BitBucket

5.1.3 Popularidad

Google trends muestra el interés o popularidad utilizando como métrica el número de búsquedas:



Figura 11 GitHub vs. GitLab vs. BitBucket [Google Trends](#)⁷¹

⁶⁷ <https://about.gitlab.com/pricing/self-managed/feature-comparison/>

⁶⁸ <https://github.com/pricing>

⁶⁹ <https://about.gitlab.com/pricing/>

⁷⁰ <https://www.atlassian.com/software/bitbucket/pricing>

⁷¹ <https://trends.google.es/trends/explore?cat=13&date=all&q=GitHub,GitLab,BitBucket>

5.2 Anexo II: Concesión de licencias

En la actualidad existen alrededor de cien licencias de código abierto en el mercado, y elegir una no es fácil, de hecho puede convertirse en una tarea crítica que podría comprometer todo el proyecto. En la medida en que un análisis en profundidad del proceso de concesión de licencias está fuera del alcance de este documento, este anexo está diseñado para obtener una visión rápida del estado actual de la técnica en este campo.

Este espectro de licencias enfocado a la privatización se explica en la siguiente imagen:

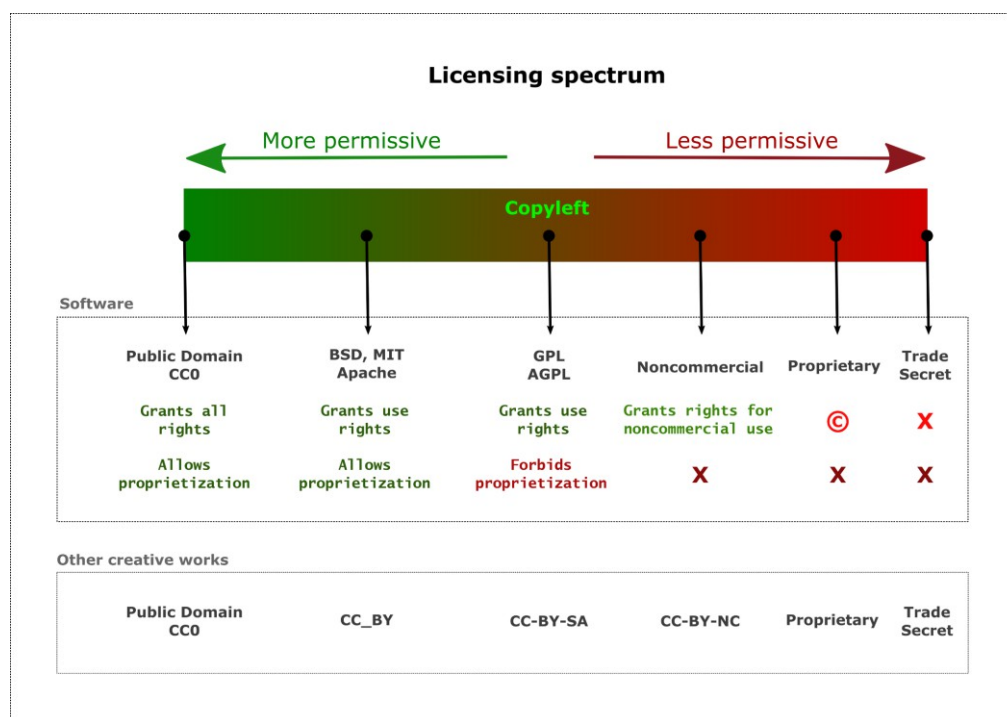


Figura 12 Ejemplo de espectro básico de licencias (Adaptado de [Permissive software license wiki](https://en.wikipedia.org/wiki/Permissive_software_license))⁷²

Ejemplo de esquema de compatibilidad de licencias de *software* libre y de código abierto (FOSS):

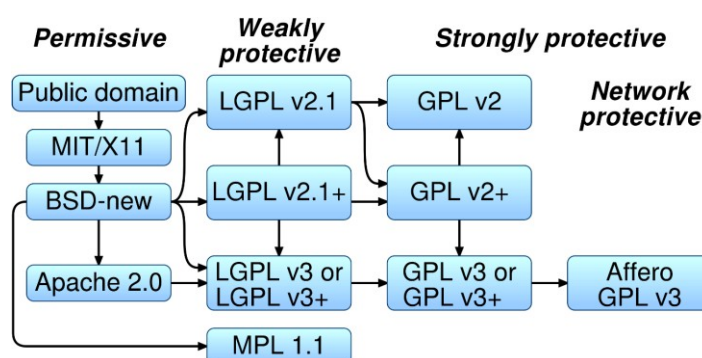


Figura 13 Compatibilidad con la licencia Floss (Fuente: [Wikimedia Commons](https://commons.wikimedia.org/wiki/File:FOSS_license_compatibility.png))⁷³

⁷² https://en.wikipedia.org/wiki/Permissive_software_license

⁷³ <https://commons.wikimedia.org/wiki/File:Floss-license-slide-image.svg>

En cuanto a los resultados técnicos que dependen de otros componentes que pueden ser (o no) más o menos permisivos, debemos tener en cuenta la compatibilidad con versiones anteriores.

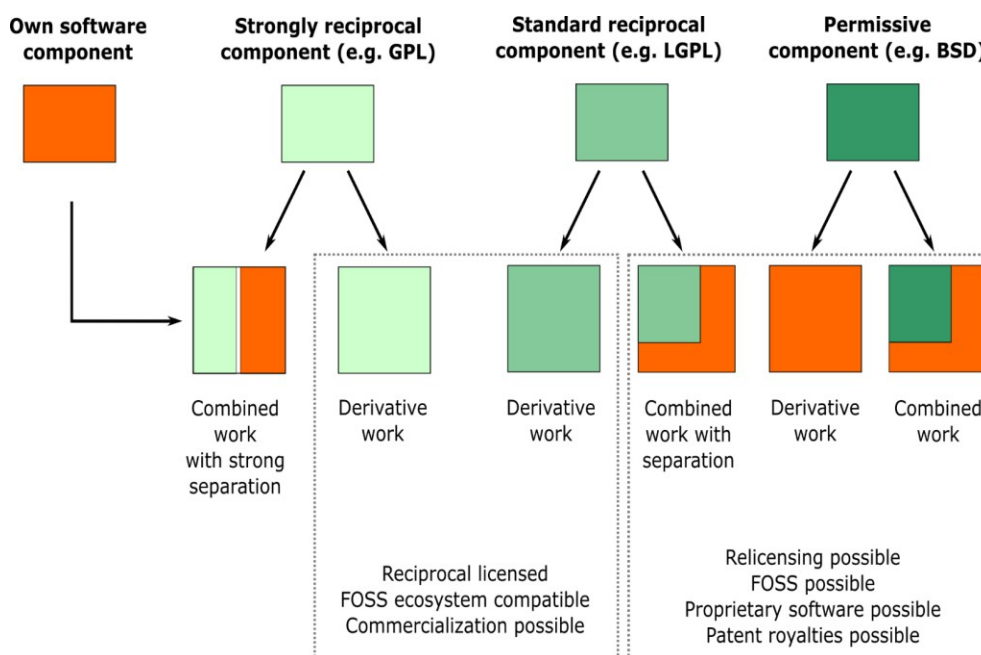


Figura 14 Compatibilidad de licencias de software para obras derivadas (Fuente: [Wikimedia Commons](#))⁷⁴

5.3 Anexo III: Recursos

Proyectos

- [PoliRural](#) - Desarrollo colaborativo de políticas orientadas al futuro para las zonas y la población rurales
- [Digital Innovation Hub](#) - Espacio social para regiones inteligentes
- [PoliRural GitHub Organization](#) - WP3 - Integración final, publicación y licencias

Principales sitios relacionados con este documento

- [GitHub](#) / [Documentos de GitHub](#)
- [OpenAIRE](#)
- [OpenAIRE en EOSC](#)
- [Zenodo](#)

Documentos, informes y sitios informativos

- [Guía de datos de investigación 2022](#) - Biblioteca EUI
- [Licencias de código abierto y datos abiertos en la era de las infraestructuras inteligentes: Revisión y marcos de selección de licencias](#) | [SpringerLink](#) - Pay walled
- [Calidad de los metadatos](#) | [Colecciones de naturaleza](#)

⁷⁴ <https://commons.wikimedia.org/wiki/File:Software-license-compatiblity-graph.svg>

- [FAIRsharing.org](https://fairsharing.org) - Normas, bases de datos y políticas
- [Cartilla de licencias de código abierto para IA/ML empresarial | Hacia la ciencia de datos](#)
- [Cómo aplicar una licencia a su proyecto de software de código abierto | FOSSA](#)
- [Repositorios disciplinarios](#) - Open Access Directory at Simmons University
- [Compatibilidad de licencias | Wikipedia](#)
- [Licencia de software permisiva | Wikipedia](#)

Herramientas

- [Elegir una licencia de código abierto | Elegir una licencia](#) - por GitHub
- [Zona de juegos JSON-LD](#)
- [Formateador y validador JSON \(curiousconcept.com\)](#)
- [Formato de archivo de citas \(CFF\) \(citation-file-format.github.io\)](#)
- [ScienceOpen](#) - Un entorno de descubrimiento interactivo