



## E2.5. Informe Indicadores de Software v 1.0



MINISTERIO  
DE CIENCIA, INNOVACIÓN  
Y UNIVERSIDADES



Instituto de Salud Carlos III



Infraestructura de Medicina de Precisión  
asociada a la Ciencia y la Tecnología

<b>Programa</b>	IMPacT: Infraestructura de Medicina de Precisión asociada a la Ciencia y la Tecnología		
<b>Nombre Proyecto</b>	IMPACT-Data: Programa de Ciencia de Datos de IMPaCT		
<b>Expediente</b>	IMP/00019		
<b>Duración</b>	Enero 2021 – Diciembre 2024		
<b>Página web</b>	<a href="http://impact-data.bsc.es">impact-data.bsc.es</a>		
<b>Paquete Trabajo</b>	WP2 – INFRAESTRUCTURA COMPUTACIONAL EN LA NUBE PARA LA GESTION E INTEGRACION DE DATOS.		
<b>Tarea</b>	T2.2 - Adopción e implantación de protocolos para el uso de contenedores software y flujos de trabajo siguiendo las recomendaciones de ELIXIR, EOSC-Life y GA4GH		
<b>Entregable</b>	E2.5. Informe Indicadores de Software		
<b>Versión</b>	1.0		
<b>Fecha Entrega</b>	30/06/2024	<b>Fecha Aprobación</b>	
<b>Responsable</b>	BSC (Organización)		
<b>Nivel Diseminación</b>	X	PU	Público
		CO-IMP	Confidencial, sólo participantes de los pilares de IMPaCT, incluyendo la comisión de evaluación de IMPaCT.
		CO-DATA	Confidencial, sólo participantes de IMPaCT-Data, incluyendo la comisión de evaluación de IMPaCT.

<i>Autores</i>		
<i>Organización</i>	<i>Nombre</i>	<i>Rol</i>
BSC-CNS	Pablo Martínez	Autor
BSC-CNS	Eva Martín	Autora
CNB	Laura del Cano	Revisora
BSC	Aina Jene	Revisora

<i>Historial de versiones</i>			
<i>Nro.</i>	<i>Fecha</i>	<i>Descripción</i>	<i>Autor</i>
<b>v 0.0</b>	22/04/2024	Documento creado	Pablo Martínez (BSC-CNS)
<b>v 0.1</b>	30/05/2024	Primera versión índice	Pablo Martínez (BSC-CNS)
<b>v 0.2</b>	01/07/2024	Análisis y contenido	Eva Martín (BSC-CNS)
<b>v 0.3</b>	12/07/2024	Revisión interna	Pablo Martínez (BSC-CNS)
<b>v 0.4</b>	30/09/2024	Revisión de secciones	Eva Martín (BSC-CNS)
<b>v 1.0</b>	07/10/2024	Versión final	Pablo Martínez (BSC-CNS)

# Contenido

Contenido	3
Tablas	4
Figuras	4
Resumen Ejecutivo	5
Introducción	6
Audiencia	6
Ámbito	6
Relación con otros Entregables	6
Estructura Entregable	6
1    Motivación	7
2    Buenas prácticas para el desarrollo de software de investigación de código abierto (4OSS)	7
2.1    Recomendación: Código Fuente Abierto (OSS)	8
2.2    Recomendación: Software Localizable	9
2.3    Recomendación: Escoger Licencia Apropriada	11
2.4    Recomendación: Definir Comunicación, Gobernanza y Contribución	13
3    Calidad del Software	14
4    Implementación de 4OSS en IMPaCT-Data	15
5    Conclusiones	18
Referencias	19
Anexo A.    Relación de herramientas seleccionadas para Infraestructura IMPaCT-Data	20
Anexo B.    Seminario bio.tools and EDAM	24
Anexo C.    Seminario de encapsulado de componentes software en contenedores	29
Anexo D.    Seminario Workflows - Uso de gestores de workflows, registro y compartición	35

## Tablas

Tabla 1. Combinación de licencias OSS.....	12
Tabla 2. Repositorios para el catálogo de software de IMPaCT-Data .....	17
Tabla 3. Comentarios sobre la ontología EDAM para anotación de herramientas software. 18	

## Figuras

Figura 1. Registro bio.tools .....	10
Figura 2. Clasificación concepto Relation extraction en la ontología EDAM.....	11
Figura 3. Seminarios para aplicar 4OSS en IMPaCT-Data .....	15
Figura 4. Asistencia a seminarios/tutoriales .....	16
Figura 5. Uso de licencias OSS para las herramientas software.....	17
Figura 6. Uso de la ontología EDAM en El registro en bio.tools .....	17

# Resumen Ejecutivo

Este documento es el Informe final sobre los principales indicadores medidos para el conjunto del software utilizado por el programa de Ciencia de Datos de IMPaCT.

En el entregable E2.4 (Guía de Buenas Prácticas para el Desarrollo y Mantenimiento de Software) se definieron varias recomendaciones para el desarrollo de software de código abierto. Esas cuatro recomendaciones son: (1) hacer público desde el inicio el código fuente del programa en un sistema/plataforma con control de versiones, (p. ej. GitHub, GitLab, etc.); (2) hacer el software localizable a través de su publicación en registros utilizados por la comunidad para tal fin (p.ej. Bio.tools); (3) escoger una licencia reconocida por la comunidad, con preferencia por licencias de software libre, teniendo en cuenta las licencias de los componentes del que depende el software en cuestión; y (4) definir los mecanismos de comunicación, gobernanza y colaboración más allá de los propios desarrolladores del software de investigación, como por ejemplo disponer de un sistema de contribución al software, accesible a colaboradores externos y donde internamente se documenten los desarrollos realizados y donde se expliciten los proceso de modificación del software (p.ej. El reporte de errores).

En el mismo entregable (E2.4), se definieron las acciones a implementar en el marco de IMPaCT-Data para facilitar la adopción de dichas recomendaciones. Estas acciones incluyen: sesiones de trabajo y seminarios para facilitar acceso a las plataformas, creación de documentos y guías de buenas prácticas y ponerlas a disposición de los miembros de IMPaCT-Data. Los seminarios impartidos incluyeron los siguientes temas: (1) registro de herramientas en el registro bio.tools y el uso de la ontología EDAM para describir el software de investigación; (2) guía de buenas prácticas para el desarrollo de software recogiendo ejemplos prácticos de cada una de las recomendaciones; (3) encapsulado de software de investigación de forma modular en contenedores; y (4) uso de gestores de flujos de trabajo, popularmente conocidos como workflows o pipelines; el registro de dichos workflows y su publicación para el potencial uso por parte de la comunidad científica.

Como resultado de las acciones descritas, al cierre del E2.4, se identificaron 75 herramientas dentro del consorcio, de las cuales 57 se han clasificado como herramientas software, 15 como workflows y 3 como bases de datos. De las herramientas identificadas, 51 tienen licencia OSS y 58 han sido incluidas en el registro de ELIXIR bio.tools.

En el presente documento se actualizará el catálogo de herramientas identificadas en el E2.4 y se procederá a su análisis a través de los principios FAIR para evaluar el grado de implantación de las recomendaciones y el éxito de las acciones implementadas.

# Introducción

## Audiencia

Este documento está destinado a todos los participantes del proyecto IMPaCT-Data, para que dispongan de un catálogo actualizado de herramientas y de la evaluación de las mismas, de cara a poder proseguir con sus desarrollos y completar aquellos parámetros en los que la evaluación indique que tienen posibilidades de mejora.

## Ámbito

Las evaluaciones que se describen en este documento servirán como indicador del grado de desarrollo de las herramientas y de los puntos de mejora a tratar en futuras iteraciones de las mismas.

## Relación con otros Entregables

Este entregable tiene relación con el E2.4 Guía de Buenas Prácticas para el Desarrollo y Mantenimiento de Software, pues actualizará el catálogo de herramientas indicado en él y se obtendrá una evaluación de las herramientas disponibles al final del proyecto.

También se vincula con el E2.3 Informe de la Infraestructura y la Implementación de Referencia (IR) descrita en él. Ya que las herramientas descritas en el E2.4 y en este informe se podrán usar en los servicios descritos en la IR.

## Estructura del Entregable

En la Sección 1 se explica la motivación detrás de las evaluaciones que forman parte de este documento. Los procesos de las evaluaciones y su metodología se explican en la Sección 2. La Sección 3 incluye el catálogo de herramientas y componentes software que se han analizado, incluyendo como se ha obtenido dicho listado y las limitaciones encontradas [el listado de herramientas se puede consultar en el anexo A]. En la Sección 4 se han incluido los resultados de las evaluaciones de las herramientas. Y, finalmente, en la Sección 5 se incluyen las conclusiones.

En los anexos de este documento se incluye la lista de herramientas identificadas para el catálogo de IMPaCT-Data (Anexo A) y el material de soporte para implementar las evaluaciones. El Anexo B incluye detalles sobre el registro de herramientas software en el registro de ELIXIR bio.tools.



# 1 Motivación

Hoy en día el software es una parte crítica en la investigación en el campo de las ciencias de la vida. De hecho, ciertas investigaciones no se podrían realizar si no existieran las herramientas software que lo permitieran, por ejemplo compilando o analizando ciertos tipos de datos. Cuando el software se desarrolla como parte de las herramientas que dan soporte a la investigación, habitualmente no se considera como un resultado de la investigación, por lo que no se le aplican los mismos procesos para asegurar su calidad, referencia, reproducibilidad y reusabilidad.

La producción y el uso de software de código abierto (OSS, por sus siglas en inglés para Open Source Software) son unos de los mecanismos clave que permiten aplicar los principios de ciencia abierta a los componentes software. OSS es aquel software para el cual el código fuente es públicamente accesible para cualquier persona que quiera inspeccionar, usar, modificar y mejorar dicho código. Disponer del código de las herramientas software que se utilizan para la investigación contribuye al reconocimiento de los desarrolladores del mismo, a crear una comunidad alrededor de dichos desarrollos, así como a aumentar la confianza sobre el mismo, al permitir la posibilidad de inspeccionar, modificar y mejorar dicho código. Es importante señalar que, dependiendo de las licencias de los componentes utilizados en el desarrollo de software, puede ser necesario utilizar el mismo tipo de licencias, p. ej. familia de licencias GPL.

Como parte de los desarrollos generados en el marco del proyecto IMPaCT-Data se han generado herramientas y programas para facilitar y permitir la investigación en el ámbito de la salud a través del uso secundario de datos de pacientes. En este entregable se exponen, explican y detallan las metodologías de evaluación del software y los resultados de dichas evaluaciones.

## 2 Evaluaciones y Metodología

Las recomendaciones de desarrollo de software, tal como se definieron en el E2.4, son cuatro y están propuestas en la guía de buenas prácticas definida por el grupo de trabajo *Software development best practices for Life Sciences* de la organización ELIXIR [1]. El objetivo de este grupo de trabajo es mejorar la calidad y sostenibilidad del software desarrollado por investigadores en el dominio de ciencias de la vida. Las cuatro recomendaciones son:

1. **Disponer de un Código Fuente Abierto (OSS)**
2. **Que el software sea Localizable**
3. **Escoger una Licencia Apropiaada**
4. **Definir Comunicación, Gobernanza y Contribución**

Estas recomendaciones no pretenden substituir las guías de buenas prácticas existentes para el desarrollo de software, sino complementarlas. Estas recomendaciones están basadas en los valores del código fuente abierto, y se han escrito para mejorar el software desarrollado



por investigadores permitiendo que sea más localizable (fácil de encontrar), re-usable y transparente.

De hecho, estas recomendaciones están alineadas con los principios FAIR (siglas en inglés para los términos **F**indable, **A**ccesible, **I**nteroperable y **R**eusable) definidos en el contexto de gestión de datos en el ámbito científico, y en la actualidad trasladados a distintos tipos de objetos digitales, incluido el software de investigación.

Para cada una de estas recomendaciones se han definido varias prácticas o indicadores asociados que nos permiten hacer una evaluación pormenorizada (ver Anexo A). Las prácticas y/o indicadores de cada recomendación se definen en las siguientes secciones.

La evaluación de las herramientas de software se llevó a cabo utilizando una combinación de métodos automáticos y manuales para medir los indicadores asociados a las cuatro recomendaciones anteriores. Los indicadores se obtuvieron de forma manual o programática, según el caso (ANEXO C). A continuación explicamos en qué consiste cada alternativa de obtención de indicadores:

- **Obtención programática:** Se utilizaron scripts en Python para extraer datos de [bio.tools](https://bio.tools), una plataforma que centraliza información sobre software biomédico. Mediante la API de bio.tools, los scripts analizaron la disponibilidad y características de algunos indicadores para cada una de las herramientas. El listado de herramientas evaluadas se explica en la sección 3 y se puede encontrar en el Anexo A..
- **Obtención manual:** Debido a varios factores, algunos de los indicadores se obtuvieron y evaluaron manualmente. Esto implicó la revisión directa de los repositorios donde se almacenan/registrar las herramientas y el registro manual de dichos indicadores en una tabla.

En conjunto, un total de siete indicadores se midieron automáticamente. Dos de ellos podían evaluarse automáticamente para una mayoría de casos y requirieron de evaluación manual en los restantes (ver Anexo C, Tabla 3). Los otros once indicadores se evaluaron manualmente.

## 2.1 Recomendación: Código Fuente Abierto (OSS)

Para maximizar la reproducibilidad, reusabilidad y la colaboración se recomienda tener el código fuente en un repositorio abierto (de acceso público) con control de versiones, por ejemplo, GitHub o GitLab. Tener el código fuente disponible públicamente también ayuda a mejorar la visibilidad y la confianza en el componente software.

Los beneficios de esta recomendación son:

- Promueve la confianza del componente software en concreto y del proyecto en el que se está desarrollando en general.
- Facilita encontrar proyectos existentes en los que se está desarrollando software.
- El uso de un sistema de control de versiones proporciona acceso al historial de las contribuciones, lo cual ayuda a dar reconocimiento a los contribuidores de código.
- Incentiva la contribución de la comunidad.
- Incrementa las oportunidades de colaboración y re-uso.
- Expone el trabajo a la evaluación de la comunidad, posibilitando sugerencias y validaciones del código.
- Incrementa la transparencia para el escrutinio de la comunidad.
- Incentiva a los desarrolladores a pensar e implementar buenas prácticas al desarrollar código.
- Facilita la reproducción de los resultados científicos generados por las versiones anteriores del software.
- Incentiva a los desarrolladores a generar documentación, incluyendo un manual de usuario y comentarios en el código fuente.
- Garantiza el almacenamiento a largo plazo del código fuente.

Una buena práctica al crear un repositorio de código en un repositorio abierto es incluir el archivo *README.md*, en el directorio raíz, incluyendo una descripción del componente para que los usuarios o potenciales contribuidores puedan entender su propósito y/o funcionalidad, así como las políticas de contribución al código.

Los indicadores asociados a esta recomendación son:

- **1.1. Repositorio accesible.** Fue evaluado programáticamente utilizando la API de bio.tools para obtener los registros completos de cada herramienta y verificar el acceso al repositorio correspondiente, si este estaba disponible.
- **1.2. Código fuente en repositorio.** Fue evaluado manualmente mediante la revisión directa del repositorio para confirmar la presencia del código fuente.
- **1.3. Fichero README en raíz del repositorio.** Fue evaluado de forma programática de manera directa, verificando la presencia del archivo README en la raíz del repositorio.
- **1.4. Fichero README contiene descripción del software.** Fue comprobado manualmente de forma directa, revisando el contenido del README para verificar que incluyera una descripción del software.

## 2.2 Recomendación: Software Localizable

Consiste en registrar el software en un registro que sea popular en el dominio y al registrarlo incluir los suficientes metadatos para maximizar las posibilidades que sea encontrado en las búsquedas.

Los beneficios de esta recomendación son:

- Incrementa la visibilidad del proyecto, del software, su uso, sus éxitos, referencias y quienes contribuyen.
- Incentiva a los desarrolladores de software a pensar que metadatos describen mejor el software y cómo exponerlo.
- Ayuda a exponer los metadatos en un formato entendible por las máquinas (p. ej. Buscadores) a través del registro.
- Incrementa las opciones de colaboración, re-uso y mejora.

Dependiendo del tipo de software, hay que buscar el registro adecuado para darle la máxima visibilidad en el dominio más adecuado. Por ejemplo, en el contexto de IMPaCT-Data, para las herramientas software hemos escogido el registro bio.tools y para los flujos de trabajo, el registro WorkflowHub. Estos dos recursos forman parte del ecosistema de ELIXIR, lo cual se alinea con las directivas fijadas por IMPaCT. Cada registro tiene sus normas para la anotación de sus entradas, por ejemplo en el registro bio.tools se utiliza la ontología EDAM<sup>1</sup> para la anotación de los recursos que contiene (ver Anexo B), que contiene más de 3.500 conceptos clasificados y relacionados entre ellos. De hecho, en EDAM los conceptos están clasificados como:

- Tema (traducción de *Topic*<sup>2</sup>): Una categoría que denota un dominio o campo de interés, de estudio, aplicación, trabajo, datos o tecnología. Los temas no tienen fronteras claramente definidas entre sí.
- Operación (*Operation*<sup>3</sup>): Una función que procesa un conjunto de entradas y da como resultado un conjunto de salidas, o asocia los argumentos (entradas) con los valores (salidas).
- Datos (*Data*<sup>4</sup>): Información, representada en un artefacto de información (registro de datos) que es “comprensible” por herramientas computacionales dedicadas que pueden usar los datos como entrada o producirlos como salida.
- Formato (*Format*<sup>5</sup>): Una forma definida o diseño de representar y estructurar datos en un archivo de ordenador, blob, string, mensaje o en cualquier otro lugar.

Las anotaciones en el registro bio.tools utilizan los conceptos asociados a las categorías *Topic* y *Operation*. En la Figura 1 se muestran dos de las herramientas de IMPaCT-Data registradas en bio.tools, después de la descripción se ven los conceptos relacionados con el *Topic* en la primera línea y con el *Operation* en la segunda, las etiquetas de la tercera línea son otras clasificaciones fuera de EDAM (tipo herramienta, licencia, colección, etc.).

---

<sup>1</sup> <https://edamontology.org>

<sup>2</sup> [https://bioportal.bioontology.org/ontologies/EDAM?p=classes&conceptid=topic\\_0003](https://bioportal.bioontology.org/ontologies/EDAM?p=classes&conceptid=topic_0003)

<sup>3</sup>

[https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Foperation\\_0004](https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Foperation_0004)

<sup>4</sup>

[https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Fdata\\_0006](https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Fdata_0006)

<sup>5</sup>

[https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Fformat\\_1915](https://bioportal.bioontology.org/ontologies/EDAM/?p=classes&conceptid=http%3A%2F%2Fedamontology.org%2Fformat_1915)

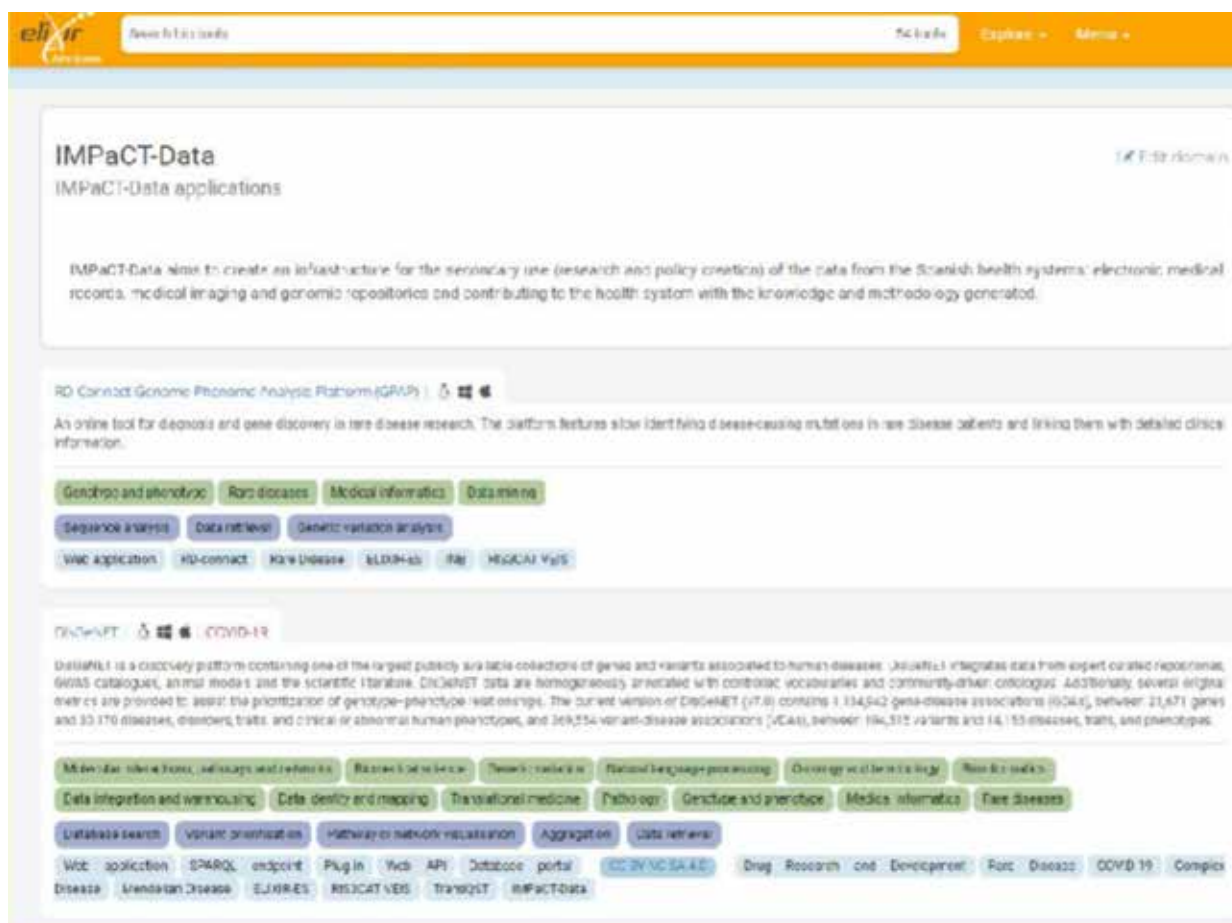


Figura 1. Registro bio.tools

Para la herramienta *RD-Connect Genome-Phenome Analysis Platform (GPAP)* de la Figura 1, los conceptos de EDAM son:

- Topic: *Genotype and phenotype* (de la clasificación *Biology/Genetics*), *Rare diseases* (*Medicine/Pathology*), *Medical Informatics* (*Informatics classification*), y *Data Mining* (*Computer Science*)
- Operation: *Data Retrieval* (clasificaciones<sup>6</sup> *Analysis/Text mining/Information Retrieval* and *Data Handling/Query and Retrieval*), *Information Extraction* (*Analysis/Text mining* y *Prediction and recognition/Text mining*), y *Relation Extraction* (*Analysis/Text mining* y *Prediction and recognition/Text mining*).

The indicators associated to this recommendation are:

- **2.1. Software registrado en bio.tools.** Dado que el catálogo de herramientas consideradas se obtuvo de bio.tools, este indicador es redundante y siempre será positivo. El cumplimiento de este indicador garantiza que las herramientas evaluadas cumplen con un estándar previo de registro y visibilidad en la comunidad científica.

---

<sup>6</sup> Un concepto puede estar clasificado en varias clasificaciones

- **2.2. Software descrito con *EDAM topic* y *EDAM operation*.** Fue evaluado programáticamente utilizando la API de bio.tools para extraer los registros completos de cada herramienta y verificar si incluían la descripción correspondiente a EDAM topic y EDAM operation.

## 2.3 Recomendación: Escoger Licencia Apropriada

A la hora de desarrollar software de investigación, es importante asignar una licencia a dicho software. Siguiendo las directrices marcadas por IMPaCT, el software desarrollado debe estar disponible bajo licencias de código abierto (Open Source). Escoger la licencia Open Source adecuada permite clarificar los términos y condiciones de uso, modificación y redistribución del código fuente, así como fija responsabilidades de los autores del software. En el caso que el software utilice otros componentes disponibles bajo licencias de código abierto, se debe comprobar que se están cumpliendo los términos y condiciones de la licencia de cada componente y si la licencia es compatible con la que hemos escogido para el nuestro.

Los beneficios de esta recomendación son:

- Clarifica las responsabilidades y derecho de terceros que quieran utilizar, copiar, redistribuir, modificar o re-usar el código fuente.
- Permite utilizar el código fuente en jurisdicciones en las que “código sin licencia” significa que no se puede utilizar de ninguna manera.
- Protege la propiedad intelectual del software.
- Proporciona sostenibilidad a largo plazo permitiendo contribuciones y reutilización en el contexto de organizaciones legales con financiación.

Una buena práctica al crear un repositorio en un repositorio abierto es incluir el archivo *LICENSE.md*, en el directorio raíz, incluyendo el texto de la licencia escogida para el componente.



En cuanto al uso de licencias de código abierto, se recomienda escoger una licencia aprobada por la Open Source Initiative (OSI), excepto que se necesiten condiciones especiales. Esto se debe a que las licencias OSI Approved garantizan que el software cumple con los estándares internacionales de código abierto, promoviendo la transparencia, la colaboración y la reutilización del software. Además, en la web de OSI se puede encontrar la lista de las licencias aprobadas y los criterios que deben cumplir, facilitando la elección de una licencia reconocida y ampliamente aceptada en la comunidad de código abierto.

Las licencias OSS se distinguen entre *copyleft* y permisivas. Las licencias *copyleft* se caracterizan porque los trabajos derivados deben ser puestos a disposición de la comunidad obligatoriamente, lo que significa que el software se debe redistribuir utilizando la misma licencia o una más restrictiva. Las licencias permisivas no incluyen restricciones en los trabajos derivados, siempre que se de crédito a los autores originales, lo cual permite su potencial uso comercial. Hay un tercer grupo de licencias que está entre los dos tipos anteriores, el código en sí está disponible bajo una licencia *copyleft* pero los componentes (normalmente librerías) se pueden combinar con otro tipo de licencias sin la obligación de distribución bajo el mismo tipo de licencia (las llamaremos *copyleft permisivas*).

Si queremos distribuir un componente OSS, la licencia que se puede escoger para la distribución depende de dos factores: (1) la licencia original del componente y (2) el tipo de distribución que se quiere hacer. En la siguiente tabla se definen los tipos de licencia que se pueden aplicar (celdas) a la distribución de un software dependiendo de la licencia original (columnas) y de su tipo de distribución (filas). Se ha considerado la distribución como trabajo derivado (se incluyen modificaciones en el código fuente), si se distribuye combinado con otros componentes y ambos.



Tabla 1. Combinación de licencias OSS

	<b>Copyleft</b>	<b>Copyleft permisiva</b>	<b>Permisiva</b>
<b>Derivado</b>	Sólo si la licencia es copyleft	Dependiendo de cómo se enlace (linking) el componente dentro del proyecto	Cualquier tipo de licencia
<b>Combinado</b>	Solo si la licencia es copyleft	Cualquier tipo de licencia	Cualquier tipo de licencia
<b>Derivado y combinado</b>	Solo si la licencia es copyleft	Dependiendo de cómo se enlace (linking) el componente dentro del proyecto	Cualquier tipo de licencia

The indicators associated to this recommendation are:

- **3.1. Hay licencia.** La existencia de la licencia del software fue verificada mediante una comprobación programática en el registro de software correspondiente para ver si los autores habían especificado una licencia. En caso negativo, se realizó una revisión manual del repositorio del proyecto para buscar la presencia de un archivo LICENSE o LICENSE.md.
- **3.2. La licencia es OSI approved<sup>7</sup>.** La evaluación se realizó programáticamente para verificar si la licencia correspondía a una identificación SPDX o a los sinónimos definidos. En caso de no lograr el mapeo, se llevó a cabo una revisión manual para determinar si la licencia cumplía con los criterios de aprobación de la OSI.

## 2.4 Recomendación: Definir Comunicación, Gobernanza y Contribución

Definir los procesos de comunicación, gobernanza y contribución no significa que el software deba ser desarrollado colaborativamente. Sin embargo, hay que definir de manera clara la estrategia de colaboración y contribución teniendo un modelo transparente de gobernanza y canales de comunicación.

Los beneficios de esta recomendación son:

- Incrementa la transparencia en cómo el proyecto y el desarrollo del software se gestiona.
- Ayuda a definir responsabilidades y procesos de toma de decisiones.
- Ayuda a la comunidad a colaborar, contribuir y comunicarse con el proyecto.

<sup>7</sup> OSI Approved se refiere a las licencias de software que han sido validadas por la Open Source Initiative (OSI) para cumplir con los estándares del software de código abierto (OSS). Estas licencias garantizan que el software puede ser utilizado, modificado y distribuido libremente.

La comunicación se refiere a establecer los mecanismos necesarios para el trabajo colaborativo dentro del proyecto, incluyendo la puesta a punto de distintos canales de comunicación. De hecho, los canales de comunicación dependerán de diversos factores teniendo en la cantidad de peticiones de mejora esperadas y del tamaño del equipo de desarrollo. Estos mecanismos se deben implantar desde el inicio del proyecto. Por ejemplo, el uso de listas de distribución, plataformas de mensajería como Slack<sup>8</sup> o rocket chat<sup>9</sup> y los *issues* en los propios repositorios de código. Para comunicaciones importantes, como por ejemplo la publicación de una nueva versión, estas se pueden hacer a través de un blog asociado al proyecto u otro espacio web y a través de los canales de los contribuidores del código.

La gobernanza se refiere a los procesos de toma de decisiones, idealmente se debe definir las diferentes responsabilidades (roles) y reconocimiento. Es importante no dejar responsabilidades sin definir dado que esto puede crear incertidumbre a medida que el proyecto software avanza y crece.

Para incentivar la colaboración de agentes externos, la inclusión de las guías de contribución es muy útil, por ejemplo, añadir el archivo *CONTRIBUTING.md* en el directorio raíz repositorio. Esta guía debe incluir como mínimo:

- Reconocimiento/bienvenida a las personas que pretenden contribuir.
- Descripción de las diferentes maneras de contribución (secciones “How-to”). Por ejemplo, como reportar un error (*bug*), como contribuir con código, como sugerir mejoras. En el caso de permitir la contribución con código, habría que incluir un enlace a las convenciones de codificación y la guía de estilo.
- La manera de contactar.

En el curso producido por el grupo de trabajo de ELIXIR *Software development best practices for Life Sciences*<sup>10</sup> (sección 4) se incluye una lista de verificación que incluye aspectos que se deben tener en cuenta para hacer las contribuciones más fáciles, claras y transparentes.

También es importante definir un código de conducta, que se puede incluir en el repositorio raíz como un archivo separado (*CODE\_OF\_CONDUCT.md*) o en una sección del archivo *CONTRIBUTING.md*. Las comunidades Open Source no solo son espacios en los que se comparte tecnología, tienen un componente social de comunidad. El código de conducta define lo que se espera de los miembros de la comunidad en sus interacciones, lo que ayuda a mantener el bienestar de la comunidad. En el código de conducta se espera encontrar:

- Comportamiento esperado
- Comportamiento no aceptado

---

<sup>8</sup> <https://slack.com/intl/es-es/>

<sup>9</sup> <https://www.rocket.chat/>

<sup>10</sup> <https://softdev4research.github.io/4OSS-lesson/>

- Consecuencias en caso de comportamiento no aceptado
- Protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros

Los indicadores asociados a esta recomendación son:

- **4.1. Existencia de un archivo llamado CONTRIBUTING.md en directorio raíz.** Fue evaluado programáticamente de manera directa, verificando automáticamente la presencia de dicho archivo en la raíz del repositorio.
- **4.2. El archivo CONTRIBUTING.md contiene reconocimiento/bienvenida a las personas que pretenden contribuir.** Fue evaluado manualmente, revisando el contenido del archivo para verificar que incluyera un mensaje de reconocimiento o bienvenida a los futuros contribuyentes
- **4.3. El archivo CONTRIBUTING.md contiene descripción de las diferentes maneras de contribución (secciones “How-to”). Por ejemplo, como reportar un error (*bug*), como contribuir con código, como sugerir mejoras. En el caso de permitir la contribución con código, habría que incluir un enlace a las convenciones de codificación y la guía de estilo.** Fue evaluado manualmente, revisando el archivo para comprobar si incluía información detallada sobre cómo contribuir, como reportar errores, aportar código o sugerir mejoras. Además, se verificó la presencia de un enlace a las convenciones de codificación y la guía de estilo en caso de permitir la contribución con código.
- **4.4. El archivo CONTRIBUTING.md contiene la manera de contactar.** Fue evaluado manualmente, revisando el archivo para verificar que incluyera información sobre cómo contactar al equipo o a los mantenedores del proyecto
- **4.5.1. Existencia de un archivo llamado CODE\_OF\_CONDUCT.md en directorio raíz.** Fue evaluado programáticamente de manera directa, comprobando automáticamente la presencia de este archivo en la raíz del repositorio.
- **4.5.2. El código de conducta se encuentra dentro del CONTRIBUTING.md.** Fue evaluado manualmente, revisando el archivo para verificar que incluyera una sección con el código de conducta.
- **4.6. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el comportamiento esperado.** Fue evaluado manualmente, revisando el contenido del archivo pertinente para verificar que incluyera una descripción clara del comportamiento esperado de los contribuyentes.
- **4.7. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el comportamiento no aceptado.** Fue evaluado manualmente, revisando el contenido del archivo pertinente para verificar que incluyera una descripción clara del comportamiento no aceptado.
- **4.8. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene las consecuencias en caso de comportamiento no aceptado.** Fue evaluado manualmente, revisando el

- contenido del archivo pertinente para verificar que incluyera una descripción clara de las consecuencias en caso de comportamiento no aceptado.
- **4.9. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros.** Fue evaluado manualmente, revisando el contenido del archivo pertinente para verificar que incluyera un protocolo claro para reportar comportamientos no aceptados.

### 3 Catálogo de herramientas

En un entregable previo (E2.4), se identificaron bio.tools, bioconda y workflowhub como plataformas donde se puede registrar el software y de las cuales se extraería el conjunto de herramientas a analizar. Sin embargo, en este entregable, se decidió considerar únicamente las herramientas de bio.tools. Esta decisión se tomó porque bio.tools contiene la forma más sencilla de software, centrada principalmente en herramientas individuales, mientras que bioconda incluye no solo software, sino también entornos completos, y workflowhub se enfoca en flujos de trabajo (workflows) más complejos.

Dado que el objetivo es analizar las prácticas aplicadas en el desarrollo de software biomédico, tiene sentido comenzar por las formas más simples antes de abordar las más complejas. Además, es probable que los desarrolladores e investigadores que utilizan bioconda y workflowhub sean usuarios más avanzados con mayores conocimientos de desarrollo, lo que puede influir en la adopción de ciertas prácticas. Por lo tanto, empezar con bio.tools permite obtener una visión más básica y generalizada de las prácticas de desarrollo antes de analizar escenarios más avanzados y complejos.

El listado de herramientas utilizadas para este entregable se obtuvo de bio.tools, específicamente de aquellas registradas por sus autores como parte de la colección 'IMPACT- Data' (con 'collectionID'). Para ello, se utilizó la API de bio.tools (<https://bio.tools/api/tool/>), lo que permitió extraer toda la información registrada de cada herramienta, recopilando así todos los detalles relevantes asociados a esta colección. [Ver Anexo A].

## 4 Evaluaciones

A continuación se presentan los resultados para cada una de las recomendaciones.

### 4.1. Código Fuente Abierto (OSS)

En esta subsección se presentan los resultados relacionados con los siguientes indicadores:

- 1.1. Repositorio accesible.
- 1.2. Código fuente en repositorio.
- 1.3. Fichero README en la raíz del repositorio.
- 1.4. Fichero README contiene descripción del software. Estos indicadores fueron previamente explicados en la sección anterior y aquí se detallan los hallazgos obtenidos.

#### 1.1. Repositorio accesible:

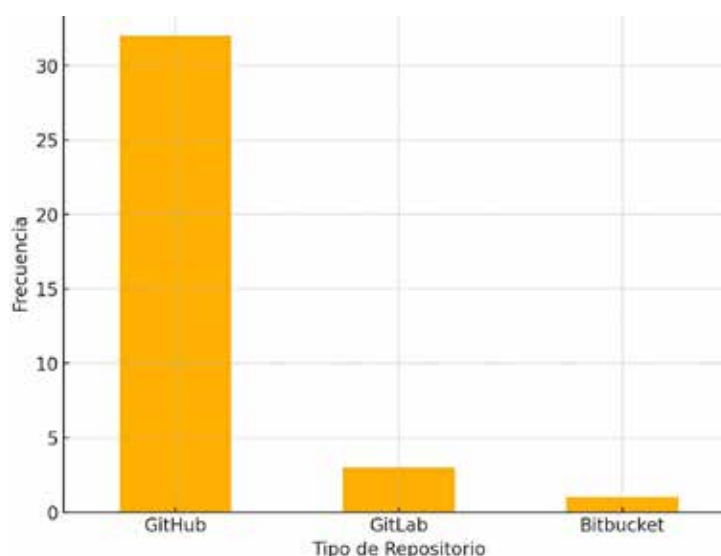
Un **41,67%** de las herramientas evaluadas tienen un repositorio público. Una única herramienta tiene registrado en bio.tools un repositorio no accesible. Las plataformas más utilizadas son GitHub, GitLab y Bitbucket, siendo GitHub la más predominante [ver Figura 1].

#### 1.2. Código fuente en repositorio:

Todos los repositorios accesibles contienen el código fuente de la herramienta.

#### 1.3. Fichero README en raíz del repositorio:

De las herramientas que tienen un repositorio accesible, el **94.1%** (32/34) incluyen un archivo README, lo cual supone el **39,4%** de las herramientas evaluadas.



**Figura 1:** Distribución de tipos de repositorio

#### 1.4. Fichero README contiene descripción del software

El **88,2%** (31/34) de los archivos README contienen la descripción de la herramienta.

#### Conclusiones y recomendaciones:

El cumplimiento de las prácticas relacionadas con el código fuente abierto es moderado. Esto sugiere que la comunidad de desarrollo valora la transparencia y la accesibilidad del código fuente, pero se podría trabajar en asegurar que más herramientas tengan repositorios accesibles y con buena documentación.

## 4.2. Software Localizable

En esta subsección se presentan los resultados relacionados con los siguientes indicadores:

- 2.1. Software registrado en bio.tools.
- 2.2. Software descrito con EDAM topic y operation.

Estos indicadores fueron previamente explicados en la sección anterior, y aquí se detallan los resultados obtenidos a partir de su evaluación.

#### 2.1. Software registrado en bio.tools:

Todas las herramientas están registradas en bio.tools, ya que este fue el punto de partida para seleccionar el conjunto de herramientas a evaluar. Esto mejora su localización y accesibilidad, facilitando a los investigadores y desarrolladores encontrar, evaluar e integrar estas herramientas en sus proyectos. Al estar centralizadas en una plataforma reconocida, se garantiza su visibilidad y un estándar mínimo de documentación, lo que contribuye a su difusión y uso en la comunidad biomédica.

#### 2.1. Software descrito con EDAM topic y operation:

El **89.3%** de las herramientas evaluadas están descritas con términos EDAM. Esto es positivo para la estandarización y facilita la búsqueda y categorización de estas herramientas en plataformas como bio.tools.

#### Conclusiones y recomendaciones:

El hecho de que un 89.3% de las herramientas evaluadas estén descritas con términos EDAM es un resultado muy positivo. Esto indica que la mayoría de los desarrolladores están siguiendo estándares de descripción semántica.



## 4.3. Escoger Licencia Apropriada

En esta subsección se presentan los resultados relacionados con los siguientes indicadores:

- 3.1. Hay licencia.
- 3.2. Licencia es OSI approved.

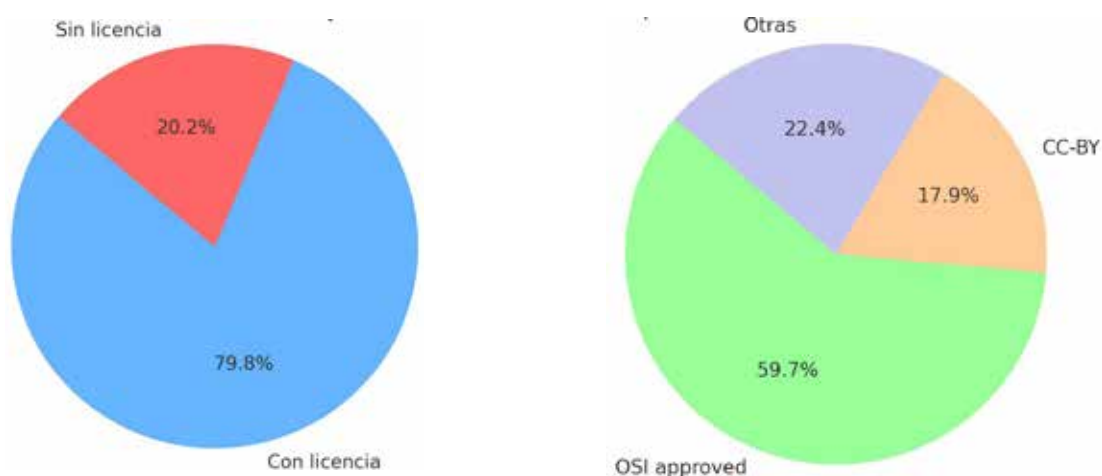
Estos indicadores fueron previamente explicados en la sección anterior, y aquí se detallan los resultados obtenidos a partir de su evaluación.

### 3.1. Hay licencia:

Un porcentaje significativo de herramientas (20.2%) no tiene licencia, lo que plantea riesgos en términos de uso y distribución.

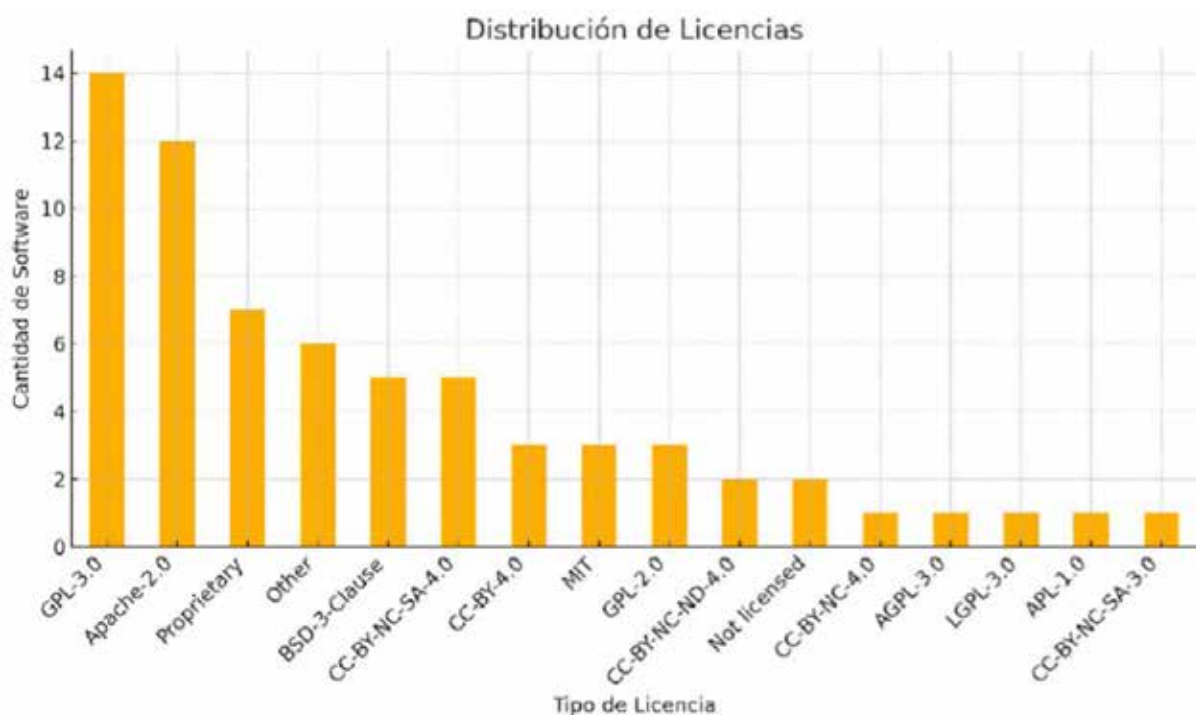
### 3.2. Licencia es OSI approved:

Entre las herramientas con licencia, la mayoría utiliza licencias aprobadas por la OSI (59.7%), lo que es positivo para la claridad y seguridad legal. Las licencias GPL-3.0 son las más comunes entre las aprobadas por la OSI. Conviene destacar que un número considerable de herramientas (17.9%) utiliza licencias CC-BY, que son más apropiadas para datos y contenidos que para software.



**Figura 2:** Izquierda: Distribución de herramientas con y sin licencia. Derecha: Distribución de tipos de licencia entre las herramientas con licencia.





**Figura 3:** Distribución de licencia

### Conclusiones y recomendaciones:

Aunque muchas herramientas tienen una licencia, la adopción de licencias aprobadas por la OSI podría ser mayor. Es fundamental que las herramientas sin licencia definan una, y se recomienda optar por licencias aprobadas por la OSI para asegurar compatibilidad y claridad legal. Las herramientas actualmente usando licencias CC-BY deberían considerar cambiar a una licencia específica para software.

## 4.4. Definir Comunicación, Gobernanza y Contribución

En esta subsección se presentan los resultados relacionados con los siguientes indicadores:

- 4.1. Existencia de un archivo llamado CONTRIBUTING.md en directorio raíz.
- 4.2. El archivo CONTRIBUTING.md contiene reconocimiento/bienvenida a las personas que pretenden contribuir.
- 4.3. El archivo CONTRIBUTING.md contiene descripción de las diferentes maneras de contribución (secciones “How-to”). Por ejemplo, como reportar un error (*bug*), como contribuir con código, como sugerir mejoras. En el caso de permitir la contribución con código, habría que incluir un enlace a las convenciones de codificación y la guía de estilo.
- 4.4. El archivo CONTRIBUTING.md contiene la manera de contactar.
- 4.5.1. Existencia de un archivo llamado CODE\_OF\_CONDUCT.md en directorio raíz.
- 4.5.2. El código de conducta se encuentra dentro del CONTRIBUTING.md.
- 4.6. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el comportamiento esperado.
- 4.7. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el comportamiento no aceptado.
- 4.8. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene las consecuencias en caso de comportamiento no aceptado.
- 4.9. El código de conducta, ya esté en archivo CODE\_OF\_CONDUCT.md o dentro de CONTRIBUTING.md, contiene el protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros.

Estos indicadores fueron previamente explicados en la sección anterior, y aquí se detallan los resultados obtenidos a partir de su evaluación.

### 4.1. Archivo CONTRIBUTING.md en directorio raíz:

Cuatro herramientas, el **4,8%** de las herramientas evaluadas incluyen un archivo CONTRIBUTING.md en el directorio raíz. Esta es una de las prácticas menos adoptadas. Pocas herramientas incluyen un archivo CONTRIBUTING.md, lo que puede dificultar las contribuciones de nuevos desarrolladores.

### 4.2. CONTRIBUTING.md contiene reconocimiento/bienvenida a las personas que pretenden contribuir

Solo una de los cuatro archivos CONTRIBUTING.md contiene un reconocimiento/bienvenida a las personas que pretenden contribuir. Lo más habitual es pasar directamente a especificar directamente cómo contribuir.

**CONTRIBUTING.md** contiene descripción de las diferentes maneras de contribución (secciones “How-to”). Por ejemplo, como reportar un error (bug), como contribuir con código, como sugerir mejoras. En el caso de permitir la contribución con código, habría que incluir un enlace a las convenciones de codificación y la guía de estilo

Todos los archivos CONTRIBUTING.md contienen una descripción de las diferentes maneras de contribución.

#### **4.3. CONTRIBUTING.md contiene la manera de contactar**

Solo una de los cuatro archivos CONTRIBUTING.md contiene la manera de contactar. Este CONTRIBUTING.md es el mismo que también tiene el reconocimiento/bienvenida a las personas que quieren contribuir (4.2).

#### **4.4. Archivo CODE\_OF\_CONDUCT.md en directorio raíz (4.5.1) o dentro de CONTRIBUTING.md (4.5.2):**

Una herramienta evaluada incluye un archivo CODE\_OF\_CONDUCT.md en el directorio raíz y otra contiene el Código de Conducta dentro de del archivo CONTRIBUTING.md. La presencia de un archivo CODE\_OF\_CONDUCT.md es excepcionalmente baja entre las herramientas evaluadas.

#### **5.6. El código de conducta, ya esté en archivo “CODE\_OF\_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el comportamiento esperado.**

En ambos casos el Código de Conducta contiene el comportamiento esperado de los participantes.

#### **5.7. El código de conducta, ya esté en archivo “CODE\_OF\_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el comportamiento no aceptado.**

En ambos casos el Código de Conducta contiene el comportamiento no aceptado de los participantes.

#### **4.8. El código de conducta, ya esté en archivo “CODE\_OF\_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene las consecuencias en caso de comportamiento no aceptado.**

Solo uno de los Códigos de Conducta contiene las consecuencias en caso de comportamiento no aceptado.

#### **4.9. El código de conducta, ya esté en archivo “CODE\_OF\_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros.**

Ambos Códigos de Conducta contienen el protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros.

### **Conclusiones y recomendaciones:**

La baja adopción de estas prácticas indica un área clara de mejora. Incluir archivos CONTRIBUTING.md y CODE\_OF\_CONDUCT.md podría fomentar un entorno de desarrollo más colaborativo y profesional. En el caso concreto de los archivos CONTRIBUTING.md, cuando se incluyen, suelen tener un contenido insuficiente, ya que no suelen contener ni bienvenida ni contacto. En cuanto al Código de Conducta, en los pocos casos en los que existe, suele contener la información necesaria.

### **Resumen de los resultados:**

- Las prácticas relacionadas con la disponibilidad del código fuente en repositorios públicos y la inclusión de archivos README son todavía insuficientemente adoptadas, ya que solo el 41,67% de las herramientas evaluadas tienen un repositorio con el código fuente.
- El registro en bio.tools es común por la metodología aplicada, y la descripción con EDAM es muy extendida (89,3%), el cual es un resultado muy positivo.
- Aún hay un pequeño conjunto de herramientas que no tienen una licencia definida (20,2%), y no todas las licencias son aprobadas por la OSI (59,7%).
- La inclusión de archivos CONTRIBUTING.md y CODE\_OF\_CONDUCT.md es extremadamente baja (2%).

## **5 Conclusiones**

Existe una polarización clara entre las herramientas evaluadas, con algunas cumpliendo muchas de las prácticas recomendadas (ver ejemplos de herramientas en ANEXO B), mientras que otras cumplen con muy pocas. Esta disparidad sugiere una diferencia significativa en la adopción y quizá conocimiento de mejores prácticas dentro de la comunidad. Como consecuencia, las herramientas que cumplen con muchas buenas prácticas tienden a ser más robustas y mejor documentadas, facilitando su uso, mantenimiento y contribución. Por otro lado, las herramientas que no cumplen muchas de estas prácticas pueden enfrentar desafíos en términos de uso, mantenimiento y colaboración, lo que puede limitar su impacto y sostenibilidad a largo plazo.

En términos de áreas de mejora, es evidente que se deben promover prácticas que actualmente tienen baja adopción. Es esencial que se adopten prácticas fundamentales como el mantenimiento de repositorios públicos, algo que solo cumple el 40% de las herramientas evaluadas, y la inclusión de una licencia clara, presente en solo el 80% de los casos. La ausencia de un repositorio público dificulta la revisión, reutilización y mejora del software por parte de la comunidad, mientras que la falta de una licencia explícita genera incertidumbre sobre los términos de uso y contribución.

La inclusión de archivos CONTRIBUTING.md y CODE\_OF\_CONDUCT.md en los repositorios de software es una práctica importante que está subutilizada. Estos archivos son cruciales para guiar a los nuevos colaboradores y establecer normas claras de conducta, lo que puede mejorar significativamente la colaboración y la salud del proyecto. Sin embargo, puede resultar difícil redactar estos documentos para desarrolladores que no tengan experiencia en proyectos de software libre, ya que estos proyectos suelen ser muy heterogéneos en sus enfoques y prácticas.

Para facilitar este proceso, se recomienda el uso de plantillas y guías ya existentes como punto de partida. Por ejemplo, GitHub ofrece una [guía](#) sobre CONTRIBUTING.md y la [Contributor Covenant](#) proporciona un modelo de CODE\_OF\_CONDUCT.md que se ha convertido en un estándar en la comunidad de software libre. Estas referencias pueden ayudar a los desarrolladores a redactar documentos efectivos y coherentes, incluso si no tienen experiencia previa en este ámbito.

Además, un archivo CONTRIBUTING.md puede ser útil incluso si no se colabora fuera del propio grupo. Puede servir como una guía fundamental para que los nuevos miembros del equipo sepan claramente cómo comenzar a trabajar en el proyecto, a quién dirigirse para obtener ayuda, y cuáles son las normas y procedimientos internos. Esto ahorra tiempo y reduce la curva de aprendizaje para los recién llegados, facilitando su integración y contribución efectiva al proyecto desde el principio.

Al promover la creación de estos archivos, se establece una cultura de colaboración más sólida y profesional. Al proporcionar guías claras para contribuir y un marco de comportamiento esperado, se fomenta un entorno más inclusivo y organizado, lo que puede atraer a más colaboradores y mejorar la calidad del software, ya sea en un contexto abierto o dentro de un equipo cerrado.

Para abordar la falta de buenas prácticas básicas estamos considerando proponer una encuesta dirigida a los desarrolladores y autores de estas herramientas en un futuro. El objetivo de esta encuesta es comprender las razones detrás de la no adopción de ciertas buenas prácticas que podrían incrementar la reutilización de su trabajo. Queremos averiguar si esta carencia se debe a una falta de conocimiento sobre la importancia de estas prácticas o a la ausencia de incentivos claros para implementarlas. Los resultados de la encuesta nos ayudarían a identificar posibles barreras y a desarrollar estrategias más efectivas.

Dependiendo de los resultados de la encuesta, se podrían implementar programas de educación y soporte para desarrolladores. Estos programas podrían incluir talleres, guías detalladas y recursos de apoyo, enfocados en las áreas identificadas como problemáticas o desconocidas. Al basar estas iniciativas en los hallazgos de la encuesta, se garantiza que los esfuerzos estén alineados con las necesidades reales de los desarrolladores, facilitando la adopción de mejores prácticas. Esto mejoraría la calidad y sostenibilidad de las herramientas de software, beneficiando a la comunidad en general.



## Referencias

[1] Jiménez, R. C., Kuzak, M., Alhamdoosh, M., Barker, M., Batut, B., Borg, M., ... & Crouch, S. (2017). Four simple recommendations to encourage best practices in research software. *F1000Research*, 6.

[2] Orviz, P., Lopez, A., Duma, D. C., David, M., Gomes, J., & Donvito, G. (2021). *A set of Common Software Quality Assurance Baseline Criteria for Research Projects*. Manubot.

## Anexo A: Relación de herramientas seleccionadas para Infraestructura IMPaCT-Data y evaluación del cumplimiento de buenas prácticas.

En este anexo se incluye la lista de 84 herramientas identificada en el contexto de IMPaCT- Data, las cuales todas están registradas en el registro bio.tools siguiendo las recomendaciones incluidas en este documento (ver sección 2.2)

[Relación de herramientas seleccionadas para Infraestructura IMPaCT-Data](#)

## ANEXO B: Tipo de obtención de los distintos indicadores

Indicador	Forma de obtención
1.1. Repositorio y accesible	Automática
1.2. Código fuente en repositorio	Manual
1.3. Fichero README en raíz del repositorio	Automática
1.4. Fichero README contiene descripción del software	Manual

**Tabla 1:** Tipo de obtención de cada uno de los indicadores asociados a la recomendación 1. Código Fuente Abierto (OSS).

Indicador	Forma de obtención
2.1. Software registrado en bio.tools	Manual
2.2. Software descrito con EDAM <i>topic</i> y <i>operation</i>	Automático

**Tabla 2:** Tipo de obtención de cada uno de los indicadores asociados a la recomendación 2. Software Localizable.

Indicador	Forma de obtención
3.1. Hay licencia repositorio)	Automático (en bio.tools) / Manual (en
3.2. Licencia es <i>OSI approved</i>	Automático (si mapea a SPDX y sinónimos que tenemos actualmente) / Manual (si no logramos mapearlo)

**Tabla 3:** Tipo de obtención de cada uno de los indicadores asociados a la recomendación 4. Escoger Licencia Apropiada.

Indicador	Forma de obtención
4.1. Archivo “CONTRIBUTING.md” en directorio raíz	Automático
4.2. “CONTRIBUTING.md” contiene reconocimiento/bienvenida a las personas que pretenden contribuir	Manual
4.3. “CONTRIBUTING.md” contiene descripción de las diferentes maneras de contribución (secciones “How-to”). Por ejemplo, cómo reportar un error ( <i>bug</i> ), como contribuir con código, como sugerir mejoras. En el caso de permitir la contribución con código, habría que incluir un enlace a las convenciones de codificación y la guía de estilo	Manual
4.4. “CONTRIBUTING.md” contiene la manera de contactar	Manual
4.4.1. Archivo “CODE_OF_CONDUCT.md” en directorio raíz	Automático
4.4.2. Código de conducta dentro del “CONTRIBUTING.md”	Manual
4.5. El código de conducta, ya esté en archivo “CODE_OF_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el comportamiento esperado.	Manual
5.7. El código de conducta, ya esté en archivo “CODE_OF_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el comportamiento no aceptado.	Manual
4.8. El código de conducta, ya esté en archivo “CODE_OF_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene las consecuencias en caso de comportamiento no aceptado.	Manual

4.9. El código de conducta, ya esté en archivo

“CODE\_OF\_CONDUCT.md” o dentro de “CONTRIBUTING.md”, contiene el protocolo para reportar un comportamiento no aceptado por parte de alguno de los miembros. Manual

---

**Tabla 4:** Tipo de obtención de cada uno de los indicadores asociados a la recomendación 4. Definir Comunicación, Gobernanza y Contribución.

## ANEXO C: Ejemplos Representativos de Herramientas Evaluadas

En este anexo se presentan ejemplos representativos de las herramientas evaluadas, con el fin de ofrecer al lector una visión más concreta del proceso de evaluación y los criterios aplicados. Para cada herramienta, se enumeran los indicadores evaluados. Aunque no se seleccionaron por ser necesariamente destacadas, estas herramientas ilustran las prácticas comunes y áreas de mejora dentro del conjunto evaluado. Las observaciones finales proporcionan un análisis breve de los puntos fuertes y debilidades identificados en cada caso, ayudando a comprender la aplicación práctica de los indicadores definidos en el estudio.

### FAIR4Health Data Privacy Tool

- **Repositorio y accessible:** [GitHub](#)
- **Código fuente en repositorio:** Sí
- **Software registrado en bio.tools:** Sí
- **Software descrito con EDAM topic y operation:** Sí ([http://edamontology.org/operation\\_3283](http://edamontology.org/operation_3283))
- **Licencia:** CC-BY-4.0
- **Licencia es OSI approved:** No
- **Fichero README en raíz del repositorio:** Sí
- **Archivo CONTRIBUTING.md en directorio raíz:** No
- **Archivo CODE\_OF\_CONDUCT.md en directorio raíz:** No

**Observaciones:** FAIR4Health Data Privacy Tool se destaca por la existencia de repositorio público de código fuente en GitHub y su descripción con EDAM en bio.tools. Sin embargo, falta la inclusión de archivos [CONTRIBUTING.md](#) y [CODE\\_OF\\_CONDUCT.md](#), lo que podría mejorarse. Además, su licencia no es una licencia apropiada para software, ya que no aborda aspectos específicos como la distribución de código fuente o las garantías.

### FAIR4Health Data Curation Tool

- **Repositorio y accessible:** [GitHub](#)
- **Código fuente en repositorio:** Sí
- **Software registrado en bio.tools:** Sí
- **Software descrito con EDAM topic y operation:** Sí ([http://edamontology.org/topic\\_0089](http://edamontology.org/topic_0089), [http://edamontology.org/operation\\_3432](http://edamontology.org/operation_3432))
- **Licencia:** CC-BY-4.0
- **Licencia es OSI approved:** No
- **Fichero README en raíz del repositorio:** Sí
- **Archivo CONTRIBUTING.md en directorio raíz:** No
- **Archivo CODE\_OF\_CONDUCT.md en directorio raíz:** No

**Observaciones:** Similar a la herramienta anterior, FAIR4Health Data Curation Tool cumple con muchas prácticas clave pero carece de archivos [CONTRIBUTING.md](#) y

**CODE\_OF\_CONDUCT.md.** Al igual que la anterior, su licencia no es una licencia apropiada para software, ya que no aborda aspectos específicos como la distribución de código fuente o las garantías.

### perSVade

- **Repositorio y accessible:** [GitHub](#)
- **Código fuente en repositorio:** Sí
- **Software registrado en bio.tools:** Sí
- **Software descrito con EDAM topic y operation:** Sí ([http://edamontology.org/topic\\_3175](http://edamontology.org/topic_3175), [http://edamontology.org/operation\\_3227](http://edamontology.org/operation_3227))
- **Licencia:** GPL-3.0
- **Licencia es OSI approved:** Sí
- **Fichero README en raíz del repositorio:** Sí
- **Archivo CONTRIBUTING.md en directorio raíz:** No
- **Archivo CODE\_OF\_CONDUCT.md en directorio raíz:** No

**Observaciones:** perSVade cumple con la mayoría de las prácticas recomendadas, incluyendo el uso de una licencia aprobada por la OSI. Sin embargo, también carece de archivos **CONTRIBUTING.md** y **CODE\_OF\_CONDUCT.md**.

### TRIFID

- **Repositorio y accessible:** [GitLab](#)
- **Código fuente en repositorio:** Sí
- **Software registrado en bio.tools:** Sí
- **Software descrito con EDAM topic y operation:** Sí ([http://edamontology.org/topic\\_0080](http://edamontology.org/topic_0080), [http://edamontology.org/operation\\_3421](http://edamontology.org/operation_3421))
- **Licencia:** GPL-3.0
- **Licencia es OSI approved:** Sí
- **Fichero README en raíz del repositorio:** Sí
- **Archivo CONTRIBUTING.md en directorio raíz:** No
- **Archivo CODE\_OF\_CONDUCT.md en directorio raíz:** No

**Observaciones:** TRIFID es otra herramienta que se destaca por su cumplimiento de la mayoría de las prácticas, incluyendo una licencia OSI approved. La falta de archivos **CONTRIBUTING.md** y **CODE\_OF\_CONDUCT.md** sigue siendo una área de mejora.

### PlasmidID

- **Repositorio y accessible:** [GitHub](#)
- **Código fuente en repositorio:** Sí
- **Software registrado en bio.tools:** Sí
- **Software descrito con EDAM topic y operation:** Sí ([http://edamontology.org/topic\\_3301](http://edamontology.org/topic_3301), [http://edamontology.org/operation\\_3225](http://edamontology.org/operation_3225))



- **Licencia:** GPL-3.0
- **Licencia es OSI approved:** Sí
- **Fichero README en raíz del repositorio:** Sí
- **Archivo CONTRIBUTING.md en directorio raíz:** No
- **Archivo CODE\_OF\_CONDUCT.md en directorio raíz:** No

**Observaciones:** PlasmidID cumple con muchas prácticas, incluyendo el uso de una licencia aprobada por la OSI y una descripción con EDAM. La inclusión de archivos **CONTRIBUTING.md** y **CODE\_OF\_CONDUCT.md** podría mejorar aún más esta herramienta.