

ALTa Ciencia Abierta - Encuentro 4

Código Abierto

Jesica Formoso, Patricia Loto, Paz Míguez, Driselda
Patricia Sánchez-Aguirre, Julián Buede



Referencia sugerida:

<https://doi.org/10.5281/zenodo.12752353>



Nos presentamos

Jesica Formoso: Psicóloga, Dra de la UBA, investigadora de CONICET. Organizadora de RLadies BA y coordinadora del área de Medición de Impacto en MD.

Patricia Loto: Lic. en Sistemas por UNNE. Actualmente contribuyo en el área de Formación, Comunidad y Accesibilidad de MD.

Acompañan: Julián Buede, Francisco Palm, Mariela Rajngewerc

ALTa Ciencia Abierta



Encuentro 1

Encuentro 2

Encuentro 3

Encuentro 4

Encuentro 5

Encuentro 6

Qué, por qué y cómo de la Ciencia Abierta

Cómo usar herramientas populares de Ciencia Abierta

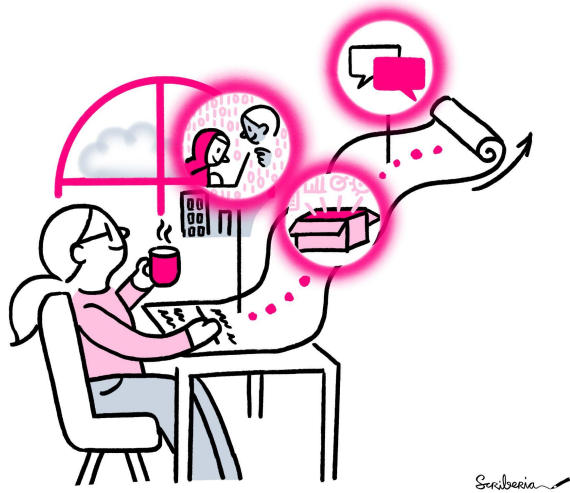
Cómo usar, crear y compartir Datos Abiertos

Cómo usar, crear y compartir Código Abierto

Cómo usar, crear y compartir Resultados Abiertos

Plenario de Iniciativas de Ciencia Abierta

¿Qué es el Software de Código Abierto?



- El **código** es un conjunto de instrucciones en un lenguaje de programación que una máquina u ordenador puede interpretar.
- El **software** es una colección de código, datos y otra información que una máquina utiliza para realizar tareas específicas.

¿Qué es el Software de Código Abierto?

Es gratuito y editable

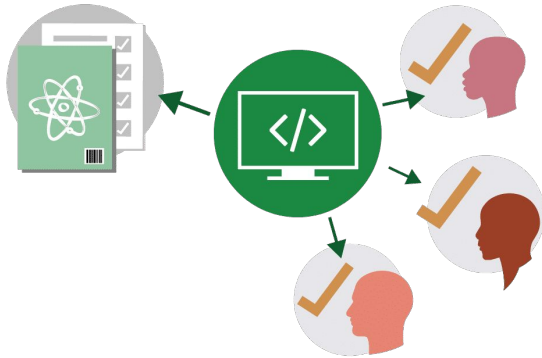
Puede tener distintos niveles de apertura

Se desarrolla colaborativamente

¿Por qué es importante compartir el Código?

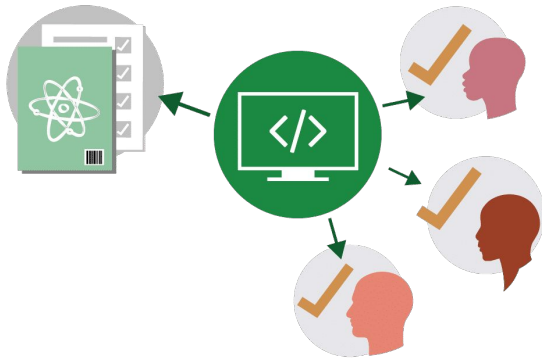
- Facilita la **reproducción de los resultados**.
- Contribuye a **validar** hallazgos
- Reduce recursos necesarios para replicar experimentos
- Abre nuevas oportunidades de **colaboración**

Principios del Código Abierto



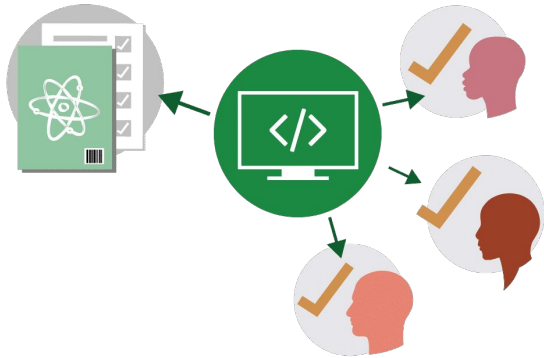
- **Transparencia**
- Colaboración
- Compartir pronto y con frecuencia
- Inclusividad
- Comunidad

Principios del Código Abierto



- Transparencia
- **Colaboración**
- Compartir pronto y con frecuencia
- Inclusividad
- Comunidad

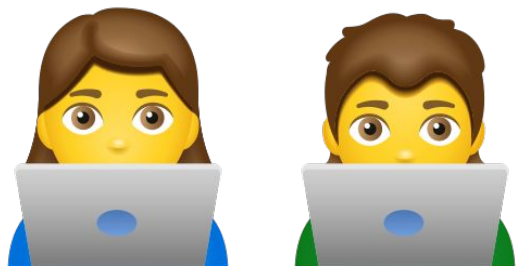
Principios del Código Abierto



- Transparencia
- Colaboración
- **Compartir pronto y con frecuencia**
- Inclusividad
- Comunidad

Ejercicio 1:

Responda la encuesta de Zoom

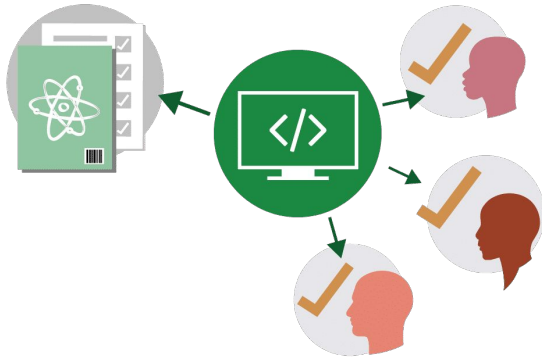


Duración: 5 minutos

¿Cuál es la definición que aplica a software de Código Abierto?

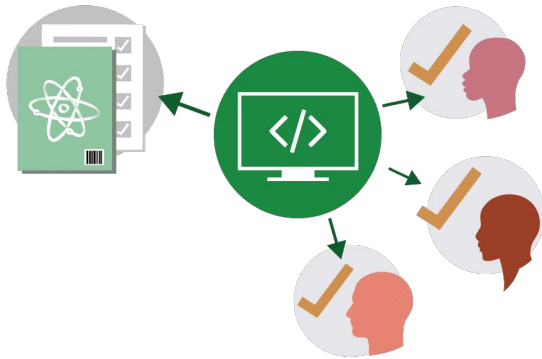
- A. Se distribuye con su código fuente, permitiendo su uso, modificación y distribución, aunque con costo.
- B. Se distribuye con su código fuente de forma gratuita, permitiendo su uso, modificación y distribución con los mismos derechos originales.
- C. Se distribuye sin su código fuente, pero sin costo alguno, permitiendo que otros lo utilicen libremente.

Principios del Código Abierto



- Transparencia
- Colaboración
- Compartir pronto y con frecuencia
- **Inclusividad**
- Comunidad

Principios del Código Abierto



- Transparencia
- Colaboración
- Compartir pronto y con frecuencia
- Inclusividad
- **Comunidad**

Beneficios



- Acelera la ciencia.
- Ahorra tiempo y dinero.
- Aumenta usuarios y desarrolladores.
- Mejora la calidad y confianza en el software.
- Aumenta la visibilidad y mejora las oportunidades laborales y profesionales de los desarrolladores.

Desafíos

- Costo y esfuerzo
- Competencia
- Interpretación errónea
- Ausencia de citación
- Código demasiado sensible
- Percepción de poca utilidad

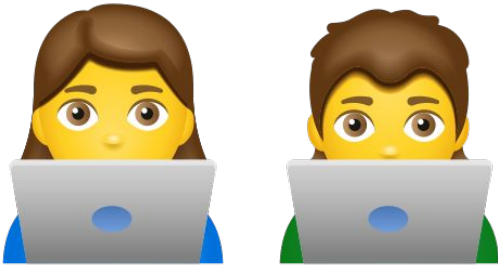
Cómo mitigarlos

- Trabajar en comunidad
- Citación y licencia
- Buena documentación
- ¡No empeorar el problema!
- Controlar el acceso
- ¡Nunca se sabe!



Ejercicio 2

Intercambio en salas de grupo



Duración: 15 minutos

1. Cada grupo **elige una persona para moderar** la conversación y **otra para sintetizar**.
2. Tomen 5 minutos para presentarse
3. Luego, compartan: ¿Has reusado código de otras personas? ¿Escribes código propio? ¿Lo compartes? ¿Qué plataformas usas? ¿Qué dificultades has encontrado?

Recapitulando



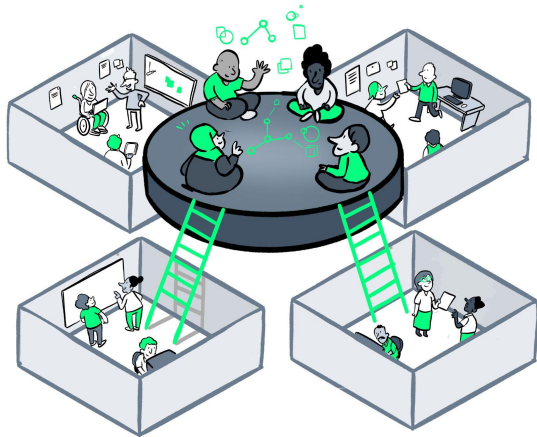
- El software es de código abierto cuando **su código fuente está disponible**: cualquiera puede verlo, modificarlo y distribuirlo como desee.
- Cuando los materiales son accesibles, podemos **construir sobre las ideas y descubrimientos de los demás**.
- Por medio de la colaboración, podemos **resolver problemas juntos** que nadie puede resolver solo.

Volvemos en 5 minutos



No te desconectes pero sí aléjate de pantallas

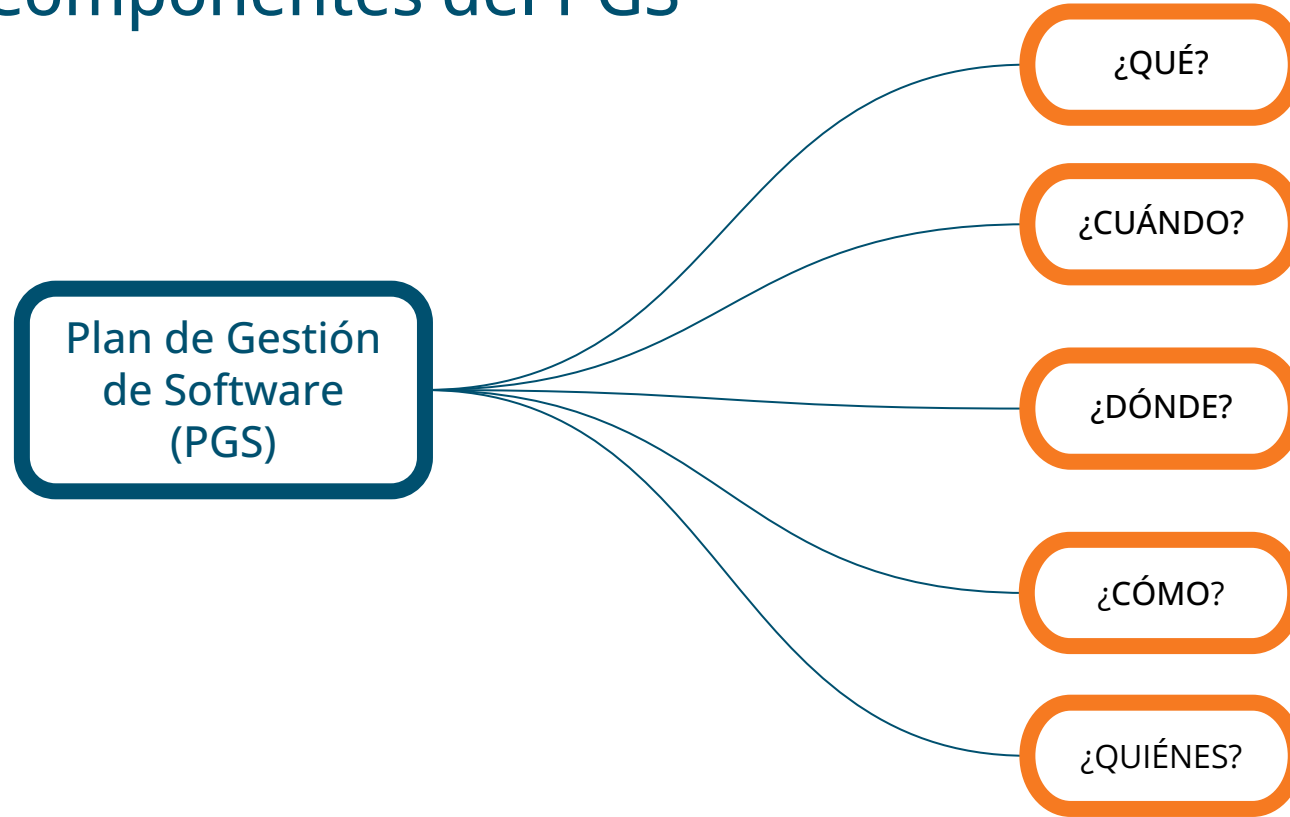
Plan de Gestión de Software (PGS)



Scriberia 

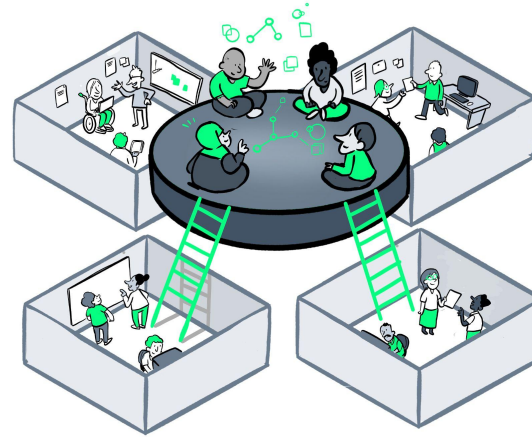
- El objetivo y el alcance del proyecto.
- Las características principales y posibles limitaciones.
- La audiencia.
- El o los lenguajes de programación que se usarán.
- Los recursos necesarios.
- Cómo se administrará el proyecto.

Componentes del PGS



EJEMPLO DE PGS

Ejemplo práctico



Scriberia 

¿Qué Compartir?

- ¿Cuál es la finalidad de compartir?
- ¿Hay que considerar políticas de la agencia de financiamiento o regulaciones locales?
- ¿Hay problemas de seguridad relacionados con el código?

¿Cuándo Compartir?

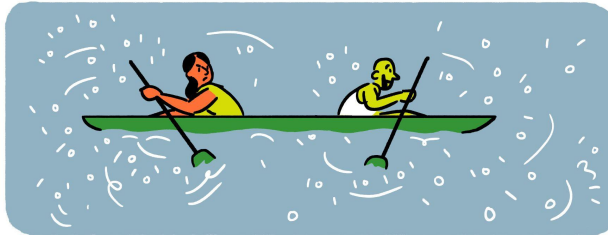
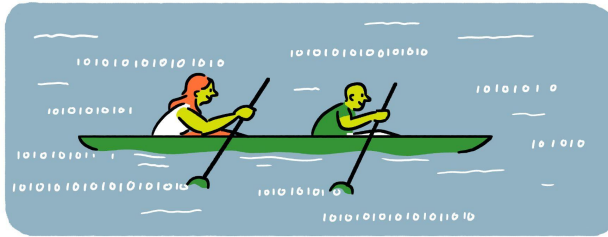
Planifica compartir tu código desde el inicio.



Scriberia

¡IMPORTANTE!

Cuando **no** compartir el código



Scriberia

- El código incluye secretos militares de un país.
- El código contiene propiedad intelectual patentada.
- El código está restringido por políticas institucionales o regulaciones organizacionales.

¿Dónde Compartir?

- Revistas de acceso abierto/software abierto y conferencias.
- Repositorios de software.
- Comunidades para la revisión por pares.
- Repositorios públicos a largo plazo.



¿Dónde Compartir?

Una **plataforma de control de versiones** permite:

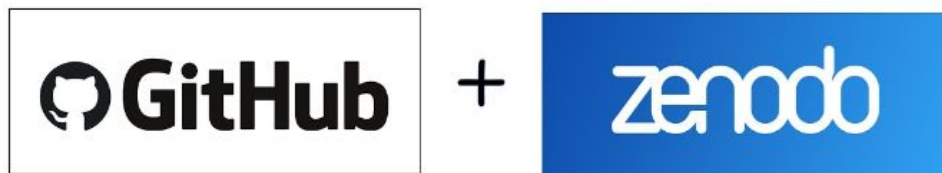
- Seguimiento de los cambios.
- Seguimiento de las contribuciones.
- Revertir cambios no deseados o errores.



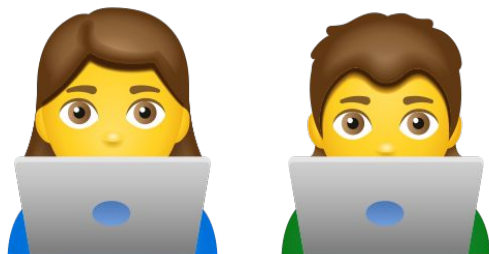
¿Dónde Compartir?

Zenodo permite:

- Crear una captura estática del código.
- Se crea una nueva captura con cada nuevo **lanzamiento.**



Demostración: GitHub y Zenodo

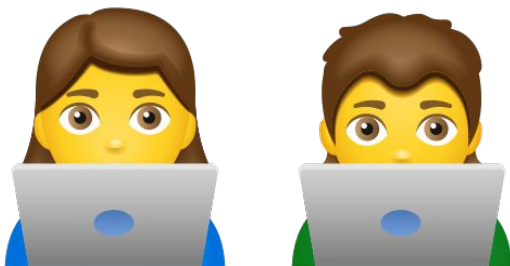


Pasos:

1. Conectar Zenodo (<https://sandbox.zenodo.org/>) a GitHub.
2. Habilitar el repositorio que queremos archivar.
3. Generar un nuevo lanzamiento.

Ejercicio 3:

Responda la encuesta de Zoom



Duración: 5 minutos

¿Por qué usar control de versiones en proyectos de Código Abierto?

- A. Mejora automáticamente la calidad del código escrito.
- B. Resuelve automáticamente todos los conflictos que surgen de ediciones simultáneas de código
- C. Permite hacer un seguimiento de los cambios en el código y la documentación a lo largo del proyecto.

¿Cómo Compartir?

Reutilización
de código

Archivo **README.md**

Debe contener:

- Nombre y breve descripción del proyecto(propósito del código).
- Instrucciones de instalación.
- Dependencias de código.
- Ejemplos de uso del código.

¿Cómo Compartir?

Reutilización
de código

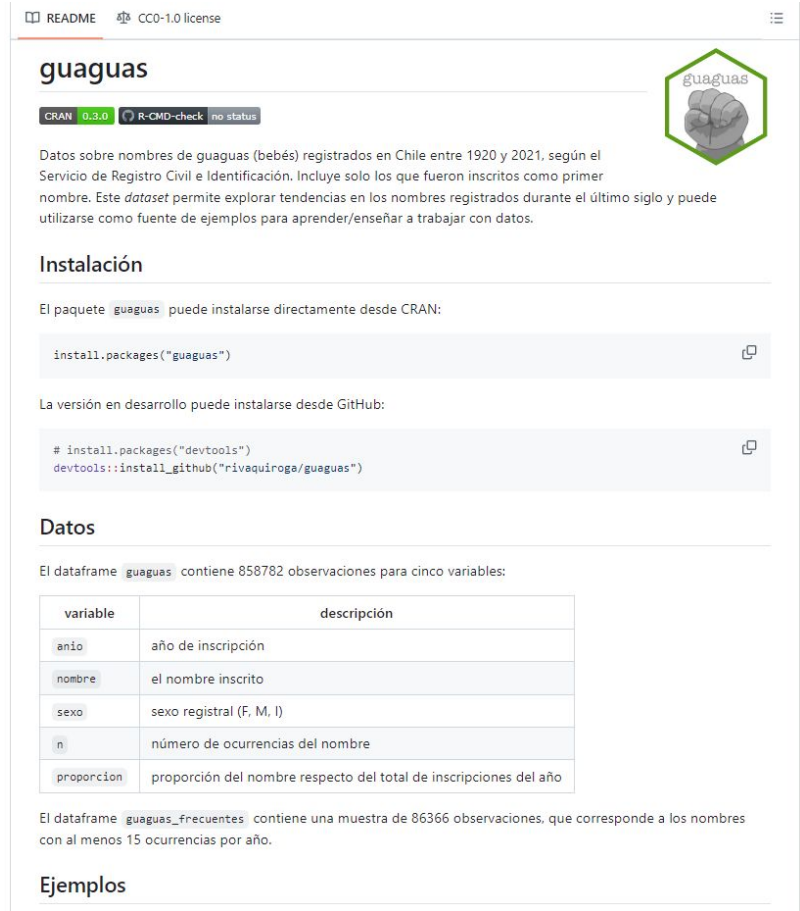
- Licencia: **LICENSE.md**
- Identificador Persistente(DOI)
- Información sobre cómo citar:
 - **CITATION.cff**
- Guías para colaboradores:
 - **CONTRIBUTING.md**
 - **CODE_OF_CONDUCT.md**

Proyecto Guaguas

Ejemplo del **Cómo Compartir**

Quiroga, R. (2022). *guaguas: Nombres Inscritos en Chile (1920 - 2021) (Version 0.3.0) [Data set]*.


<https://github.com/rivaquiroga/guaguas>



README CC0-1.0 license

guaguas

CRAN 0.3.0 R-CMD-check no status



Datos sobre nombres de guaguas (bebés) registrados en Chile entre 1920 y 2021, según el Servicio de Registro Civil e Identificación. Incluye solo los que fueron inscritos como primer nombre. Este *dataset* permite explorar tendencias en los nombres registrados durante el último siglo y puede utilizarse como fuente de ejemplos para aprender/enseñar a trabajar con datos.

Instalación

El paquete `guaguas` puede instalarse directamente desde CRAN:

```
install.packages("guaguas")
```

La versión en desarrollo puede instalarse desde GitHub:

```
# install.packages("devtools")
devtools::install_github("rivaquiroga/guaguas")
```

Datos

El dataframe `guaguas` contiene 858782 observaciones para cinco variables:

variable	descripción
<code>año</code>	año de inscripción
<code>nombre</code>	el nombre inscrito
<code>sexo</code>	sexo registral (F, M, I)
<code>n</code>	número de ocurrencias del nombre
<code>proporción</code>	proporción del nombre respecto del total de inscripciones del año

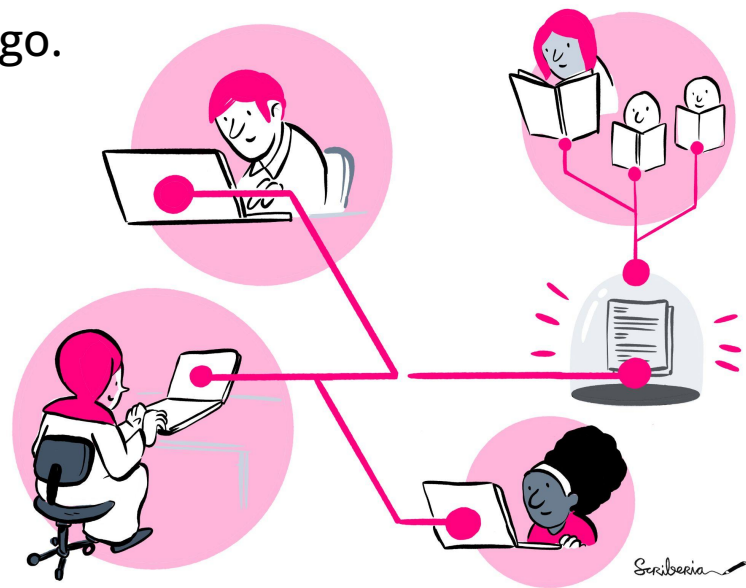
El dataframe `guaguas_frecuentes` contiene una muestra de 86366 observaciones, que corresponde a los nombres con al menos 15 ocurrencias por año.

Ejemplos

¿Quiénes?

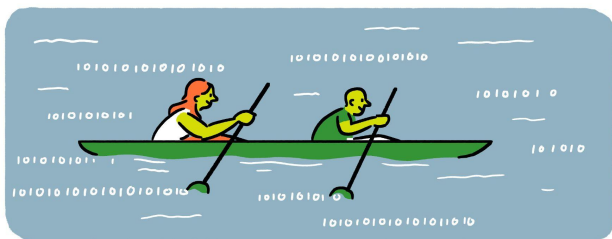
Roles

Comprende los **roles y responsabilidades** necesarios para compartir y (si corresponde) mantener tu código.



¡IMPORTANTE!

Acerca del mantenimiento del código



Scriberia

- Informemos a los usuarios en la documentación.
- Respondamos a las solicitudes de funciones nuevas y correcciones tanto como podamos.
- Consideremos ceder la propiedad si no podemos mantenerlo.

Ejercicio 4:

Responda la encuesta de Zoom



Duración: 5 minutos

Selecciona cuáles de las siguientes son formas de ayudar a otros a que reutilicen tu código:

- A. Asignar una licencia apropiada.
- B. Abrir el código desde el primer momento.
- C. Agregar una guía para contribuidores.
- D. Agregar información sobre cómo citar el código.

Recapitulando



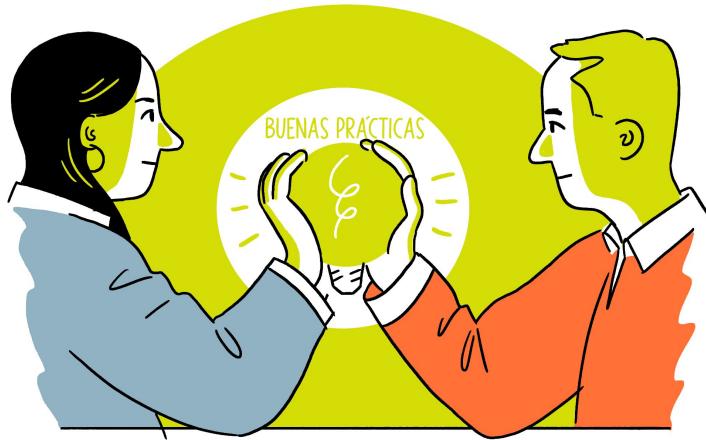
- Escribe el **Plan de Gestión de Software** antes de iniciar el proyecto.
- **¡Usa control de versiones!** Permite hacer un seguimiento de los cambios en el código y la documentación de un proyecto.
- Siempre **agrega un README.md** descriptivo.

Volvemos en 5 minutos



No te desconectes pero sí aléjate de pantallas

Buenas prácticas



Scriberia

- **Buscar** Código Abierto.
- **Evaluar** Código Abierto.
- **Citar** Código Abierto.
- **Contribuir** a proyectos de Código Abierto.
- **Participar** de comunidades.

Cómo buscar Código Abierto



- Definir un propósito claro.
- Determinar las tareas que se esperan que el código lleve a cabo.
- Utilizar palabras clave relacionadas con el propósito de programación.
- Identificar dónde buscar: Foros, literatura científica, repositorios abiertos, colegas con experiencia.

Cómo evaluar Código Abierto



- **Funcionalidad:** ¿Será útil para el problema científico?
- **Interoperabilidad:** ¿Qué tan difícil será de usar?
- **Seguridad:** ¿Es seguro? ¿Usar el software crearía un riesgo de seguridad?
- **Licencias + restricciones:** ¿Puede usarse? ¿Es legal usar este código en un proyecto?

Cómo citar

[AAS Software](#)
[Citation](#)
[Suggestions](#)

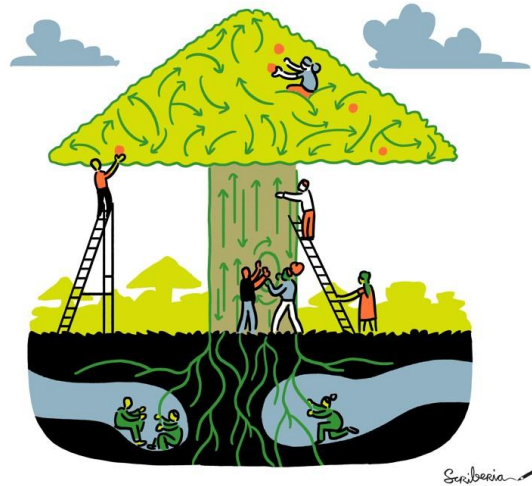
- Citar cualquier código que haya contribuido a la investigación.
- Consultar instrucciones específicas sobre cómo citar software en la revista en que se publica.
- Algunas licencias de software requieren un reconocimiento en las referencias de la publicación.

Cómo citar



- Utilizar y citar código archivado en un repositorio a largo plazo con un DOI permanente.
- Sigue el formato de citación preferido proporcionadas en el repositorio en README o CITATION.
- Los paquetes pueden ofrecer una forma de citar versiones individuales para reproducibilidad.

Cómo contribuir a proyectos de Código Abierto



- Mejorar usabilidad.
- Corregir errores.
- Informar problemas.
- Agregar documentación.
- Crear tutoriales o casos de uso.
- Refactorizar: mejorar diseño, automatización o estructura.
- Organizar/asistir a reuniones comunitarias.
- Revisar código.

Beneficios de contribuir



- Mejorar las habilidades para escribir código.
- Colaborar con personas de todo el mundo.
- Aumentar la visibilidad profesional.
- Construir una red.
- Contribuir a la comunidad.

Comunidades de Código Abierto

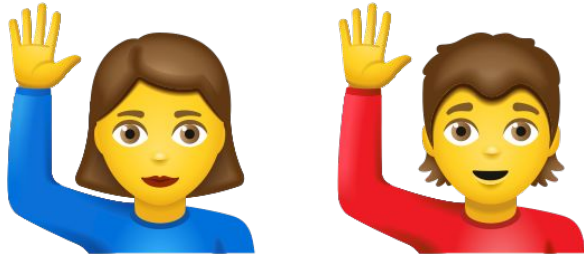


Espacios de aprendizaje social que ofrecen:

- Punto de entrada bajo para aprender.
- Compartir experiencias.
- Construir cultura y mantenerse actualizado/a.
- Comunidad no jerárquica.

Ejercicio 5:

Contesta en el chat



Duración: 3 minutos

*¿Conoces o has participado de alguna
comunidades de Código Abierto?*

Compartí por el chat el nombre y el
enlace.

Comunidades de Código Abierto



PANGEO





Uso basico del paquete censo2017

Mauricio Vargas S.

2024-06-02

Source: [vignettes/censo2017.Rmd](#)



Introduccion

Este paquete se integra perfectamente con el tidyverse, con el que daremos un ejemplo muy basico para mostrar sus principales funciones.

Aproximacion de la poblacion con el grado de doctor en la Region del Bio Bio.

Se procedera a obtener una aproximacion usando dplyr ya que, puede haber personas que no sean de la comuna y aparezcan censadas en ella. Sin embargo, cabe aclarar que para esta ocasion no haremos el filtro que corrige esto. Nuestra idea es mantener el ejemplo lo mas simple posible.

Primero que todo, se cargan los paquetes necesarios.

- censo2017: Proporciona los datos censales para poder generar las tablas y graficos de este ejemplo,
- dplyr: Facilita filtrar datos en una tabla, unir distintas tablas y en general todas las tareas de limpieza y transformacion de datos.
- ggplot2: Nos permite graficar usando el concepto de "gramatica de graficos", es decir que podemos ir creando graficos incrementales y controlar los ejes, titulos y demas elementos por separado.
- chilemapas: Nos entrega mapas terrestres con






On this page

Introduccion

Aproximacion de la poblacion con el grado de doctor en la Region del Bio Bio.

Ejercicios para el usuario

Pachá Merge pull request #16 from FvD/T1 200f860 · last year 116 Commits

 .github	Create FUNDING.yml	2 years ago
 R	Update zzz.R	2 years ago
 data-raw	cod. variables en sql	3 years ago
 data	cod. variables en sql	3 years ago
 inst	saca borrador rechazado en CRAN	4 years ago
 man	actualiza enlaces	3 years ago
 tests	better downloading function	3 years ago
 vignettes	actualiza enlaces	3 years ago

About

Una Base de Datos de Alto Rendimiento del Censo 2017 de Chile.

docs.ropensci.org/censo2017/

[rstats](#) [demographics](#) [chile](#) [census](#)
[censo](#) [demografia](#) [redatam](#) [duckdb](#)

 Readme

 CC0-1.0 license

 Code of conduct

 Activity

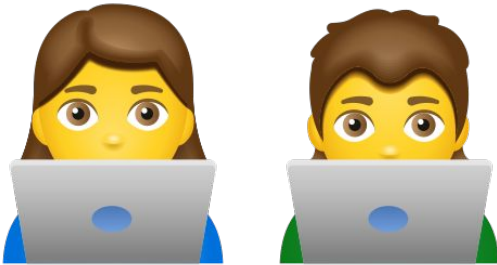
 Custom properties

 27 stars

 3 watching

Ejercicio 6:

Intercambio en salas de grupo



Duración: 10 minutos

Teniendo en cuenta lo visto hasta el momento, compartan en grupos:

- *¿Qué prácticas podemos incorporar para facilitar que otras personas contribuyan a nuestro código o lo reusen?*
- *¿Cómo hacer que nuestro proyecto sea más inclusivo?*

Recapitulando



- **Buenas prácticas** para buscar, evaluar y citar Código Abierto.
- **Cómo contribuir a proyectos de Código Abierto** y sus beneficios.
 - El desarrollo comunitario mejora el producto, su uso y su impacto.
- **Participar de comunidades de Código Abierto:** permite compartir experiencias y aprendizajes, identificar obstáculos comunes y mejorar el conocimiento de forma iterativa.

Próximos pasos

- Encuesta valoración
- Evaluación para la certificación
- Próximo encuentro



Crítica constructiva

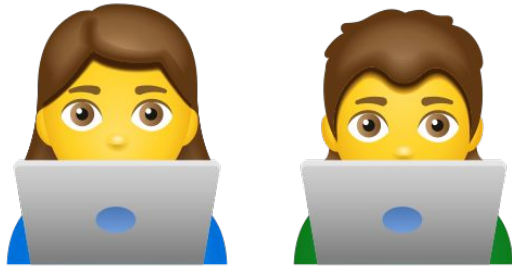
1 - Positiva

2 - Específica

**3 - Sugiere
próximos pasos**

Valoramos tu opinión:

Completa nuestra encuesta anónima

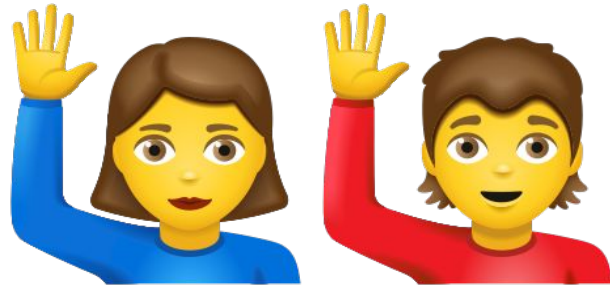


Duración: 5 minutos

http://tiny.cc/ALTa_CA_Encuesta4

Cuando termines, avisanos por el chat

Certificación NASA: Evaluación del módulo



Duración: 15 minutos

http://tiny.cc/ALTa_CA_Eval4

¿Dudas? Puedes consultar
levantando la mano o por el chat.

¿Terminaste? Puedes salir de la
reunión, ¡hasta la próxima!

Próximo encuentro



Encuentro 1

Encuentro 2

Encuentro 3

Encuentro 4

Encuentro 5

Encuentro 6

Qué, por qué y cómo de la Ciencia Abierta

Cómo usar herramientas populares de Ciencia Abierta

Cómo usar, crear y compartir Datos Abiertos

Cómo usar, crear y compartir Código Abierto

Cómo usar, crear y compartir Resultados Abiertos

Plenario de Iniciativas de Ciencia Abierta

Lecturas útiles



- NASA TOPS OS101 - Módulo 4

🚨 ¡Aviso! Estos materiales en español están en proceso de contextualización y aún no son la versión final. Te invitamos a que si encuentras errores o tienes sugerencias, los publiques como un "issue" en GitHub. ¡Dar devolución abierta es una excelente manera de contribuir a un proyecto!

- Software Carpentry - El control de versiones con GIT

Recursos útiles



- Ejemplo de proyecto software abierto: <https://github.com/numpy/numpy>
- Joss: <https://joss.theoj.org/>
- Cran: <https://cran.r-project.org/>
- Pypi: <https://pypi.org/>
- rOpenSci: <https://ropensci.org/>
- PyOpenSci: <https://www.pyopensci.org/>
- Zenodo de prueba: <https://sandbox.zenodo.org/>
- Citación: <https://citation-file-format.github.io/>
- Guía de cómo contribuir en un proyecto de Código Abierto:
<https://opensource.guide/es/how-to-contribute/> (En español)

Recursos útiles



- Practical guide to Software Management Plans: [Martinez-Ortiz, C., Martinez Lavanchy, P., Sesink, L., Olivier, B. G., Meakin, J., de Jong, M., & Cruz, M. \(2022\). Practical guide to Software Management Plans \(1.0\). Zenodo. https://doi.org/10.5281/zenodo.7248877](https://doi.org/10.5281/zenodo.7248877)
- Software Management Plans: <https://www.esciencecenter.nl/national-guidelines-for-software-management-plans/>
- Writing and using a software management plan: <https://www.software.ac.uk/guide/writing-and-using-software-management-plan>
- Software Management Plan (SMP): <https://library.maastrichtuniversity.nl/research/rdm/research-software-management/software-management-plan-smp/#:~:text=The%20goal%20of%20an%20SMP,be%20recognized%20as%20research%20output.>
- Template para software management: <https://documents.library.maastrichtuniversity.nl/open/6402486e-d88c-4ecb-9f2f-c257bf245f0c>

¡Muchas gracias!



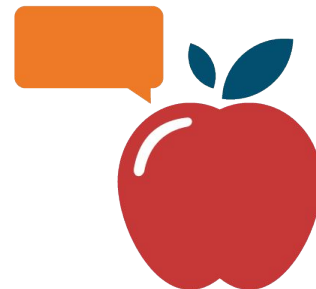
Suggested citation:

<https://doi.org/10.5281/zenodo.12752353>

Este encuentro fue posible gracias a:



Chan
Zuckerberg
Initiative 



MetaDocencia



@metadocencia

CS&S