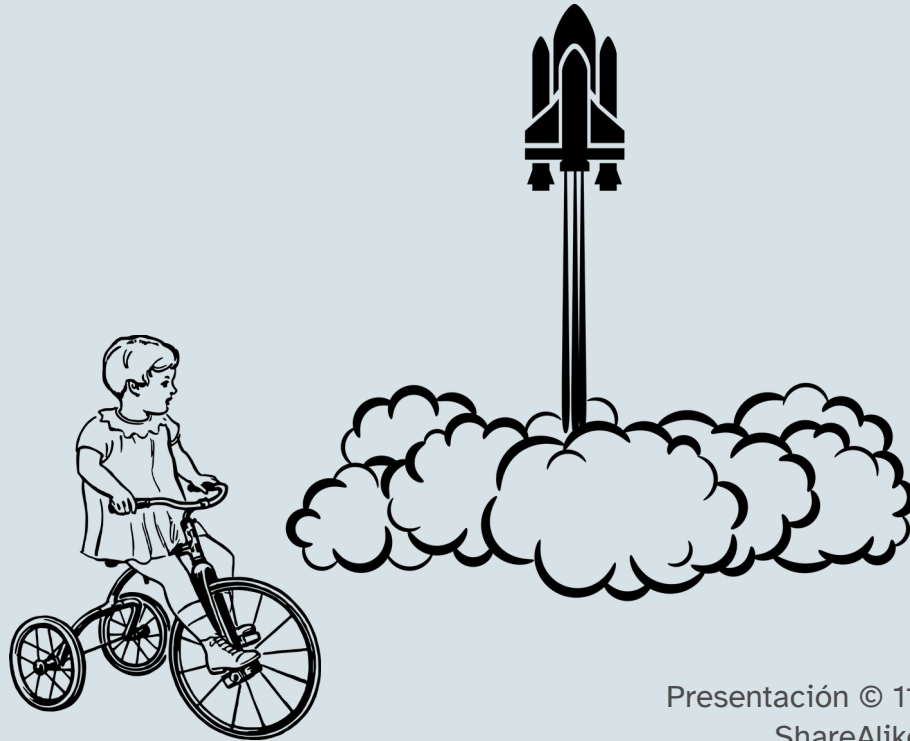


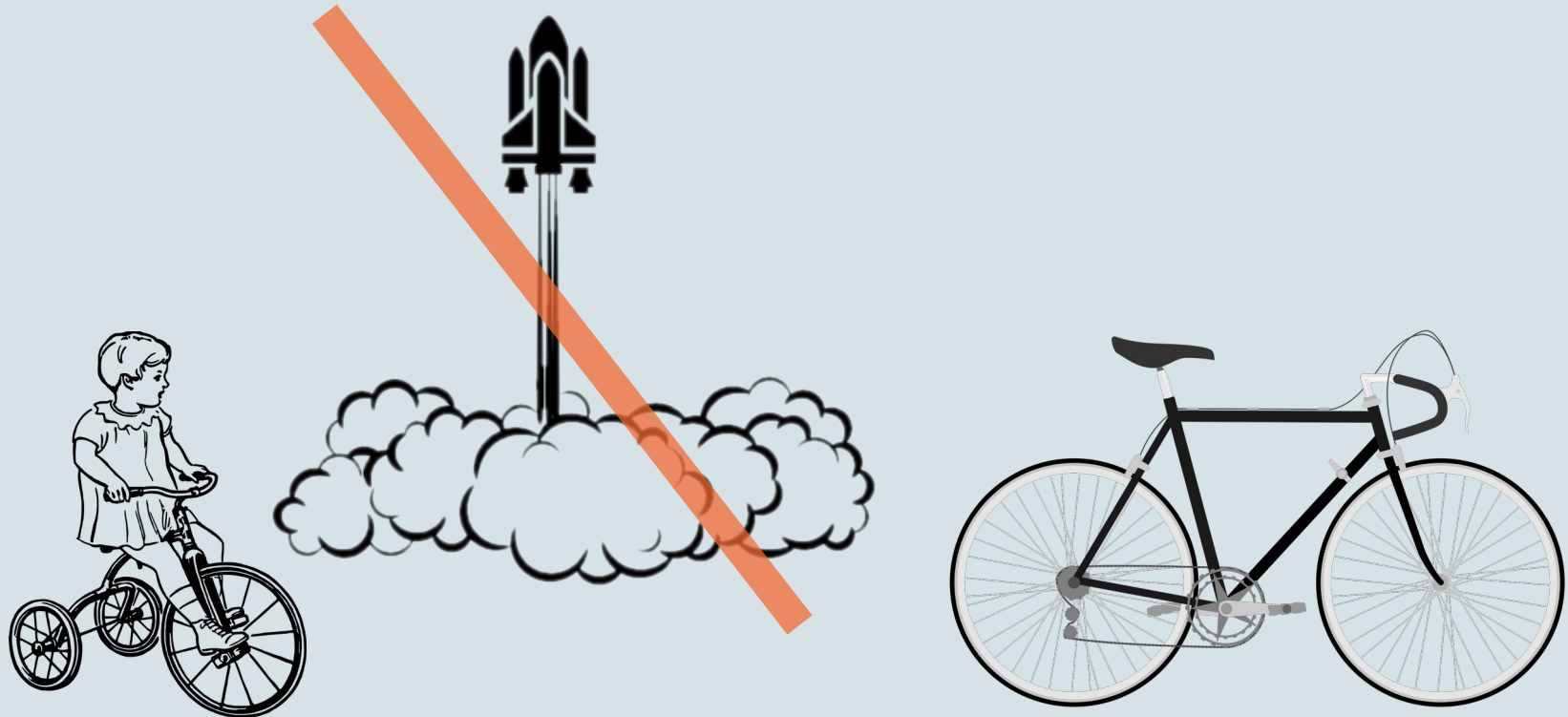
# LA INGENIERÍA DEL SOFTWARE EN LA INVESTIGACIÓN GEOGRÁFICA



Rubén Béjar  
16 de octubre de 2024



# LA INGENIERÍA DEL SOFTWARE EN LA INVESTIGACIÓN GEOGRÁFICA



# El software de investigación

Código fuente, algoritmos, scripts, flujos de trabajo computacionales y ejecutables creados durante el proceso de investigación o con el propósito de llevar a cabo la investigación

(Gruenpeter et al., 2021)

# El software **en la** investigación

Componentes de software (sistemas operativos, bibliotecas, dependencias, paquetes...) **usados para la investigación pero no creados durante la misma o sin una clara intención de ser investigación**

(Gruenpeter et al., 2021)

# Objetivos del software de investigación

Modelizado, simulación y análisis de datos

# Modelizado, simulación y análisis de datos (geográficos)

- Scripts y notebooks (p.ej., Jupyter) con análisis de datos espacio-temporales
- Simulaciones basadas en agentes, autómatas celulares o métodos numéricos
- Visualización científica
- Entrenamiento, inferencia o ajuste fino de modelos de aprendizaje automático para clasificación, predicción o generación (IA)

# Objetivos del software de investigación

Modelizado, simulación y análisis de datos

Prototipos y pruebas de concepto

# Prototipos y pruebas de concepto (geográficos)

- Diseño de nuevos algoritmos y optimización de algoritmos existentes
- Diseño de sistemas de recuperación de información geográfica, almacenes de datos, búsquedas espaciales...



# Objetivos del software de investigación

Modelizado, simulación y análisis de datos

Prototipos y pruebas de concepto

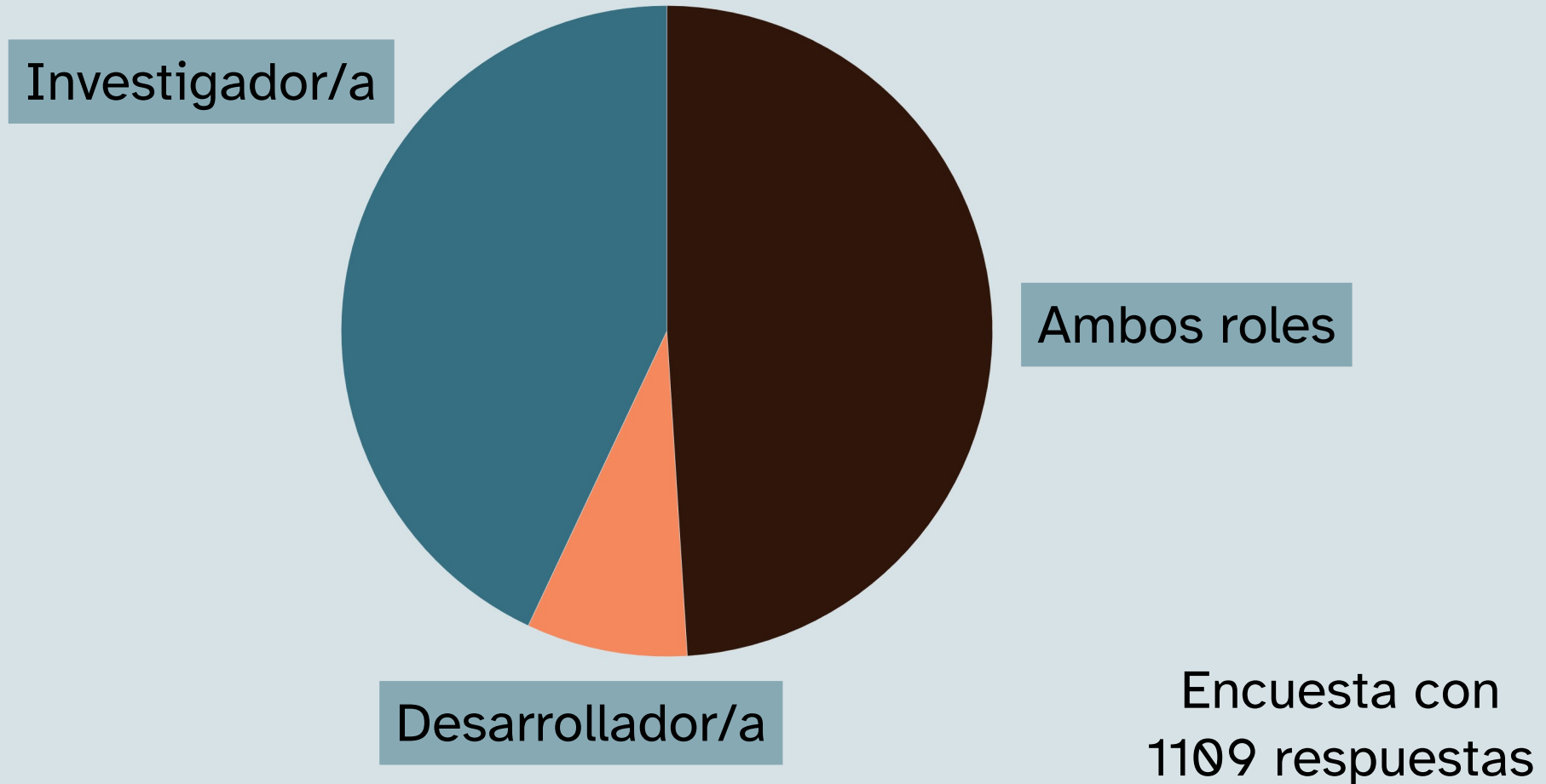
Infraestructuras de investigación

# Infraestructuras de investigación (geográfica)

- Aplicaciones para la captura de datos sobre el terreno, el trabajo colaborativo, el análisis de riesgos, la toma de decisiones...
- Aplicaciones para el control y la monitorización de espacios naturales
- Cuadernos de laboratorio interactivos (p.ej., Jupyter notebooks) con flujos de trabajo científico reusables
- Servicios web/API web, bibliotecas, aplicaciones de línea de comandos y/o con interfaces gráficas...

# La investigación sobre el software de investigación

# EE.UU. (Carver et al., 2022)



# EE.UU. (Carver et al., 2022)

- El software de investigación es crítico
- No se piensa en su sostenibilidad a largo plazo
- Necesidad de
  - Oportunidades y tiempo para formación
  - Financiación específica para el software de investigación
  - Reconocerlo como mérito de investigación

# Oak Ridge National Laboratory (Malviya-Thakur et al., 2023)

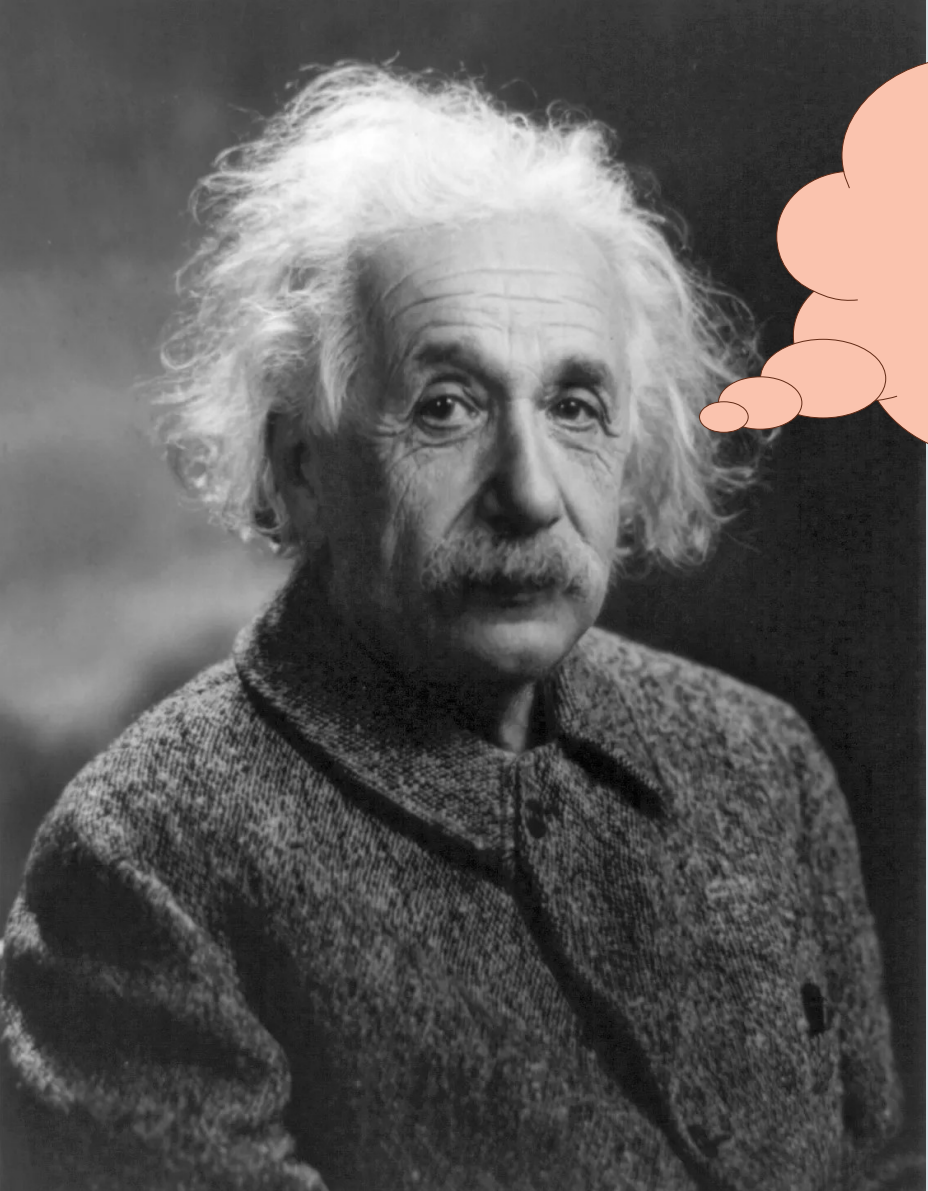
- La ciencia en el ORNL requiere mucho software grande y de vida larga
  - A menudo desarrollado por equipos multidisciplinares y multinstitucionales (*Geoinformatics Engineering, Multimodal Sensor Analytics, Transportation Analytics...*)
- Mucha gente en estos equipos se identifica como ingeniero/a de software de investigación
- La ingeniería del software de investigación es una disciplina científica, y una actividad esencial, en el ORNL

La ingeniería del software de  
investigación

*Research software engineering*  
(RSE)

Aplicación de buenas prácticas de  
ingeniería del software al software  
de investigación





¿Por qué  
ingeniería  
del  
software?

El software de investigación es **crítico**

El software de investigación es **crítico**

No se piensa en su **sostenibilidad a largo plazo**

El software de investigación es **crítico**

No se piensa en su **sostenibilidad a largo plazo**

La ciencia en el ORNL requiere mucho **software grande y de vida larga**

A menudo desarrollado por **equipos multidisciplinarios y mult institucionales**

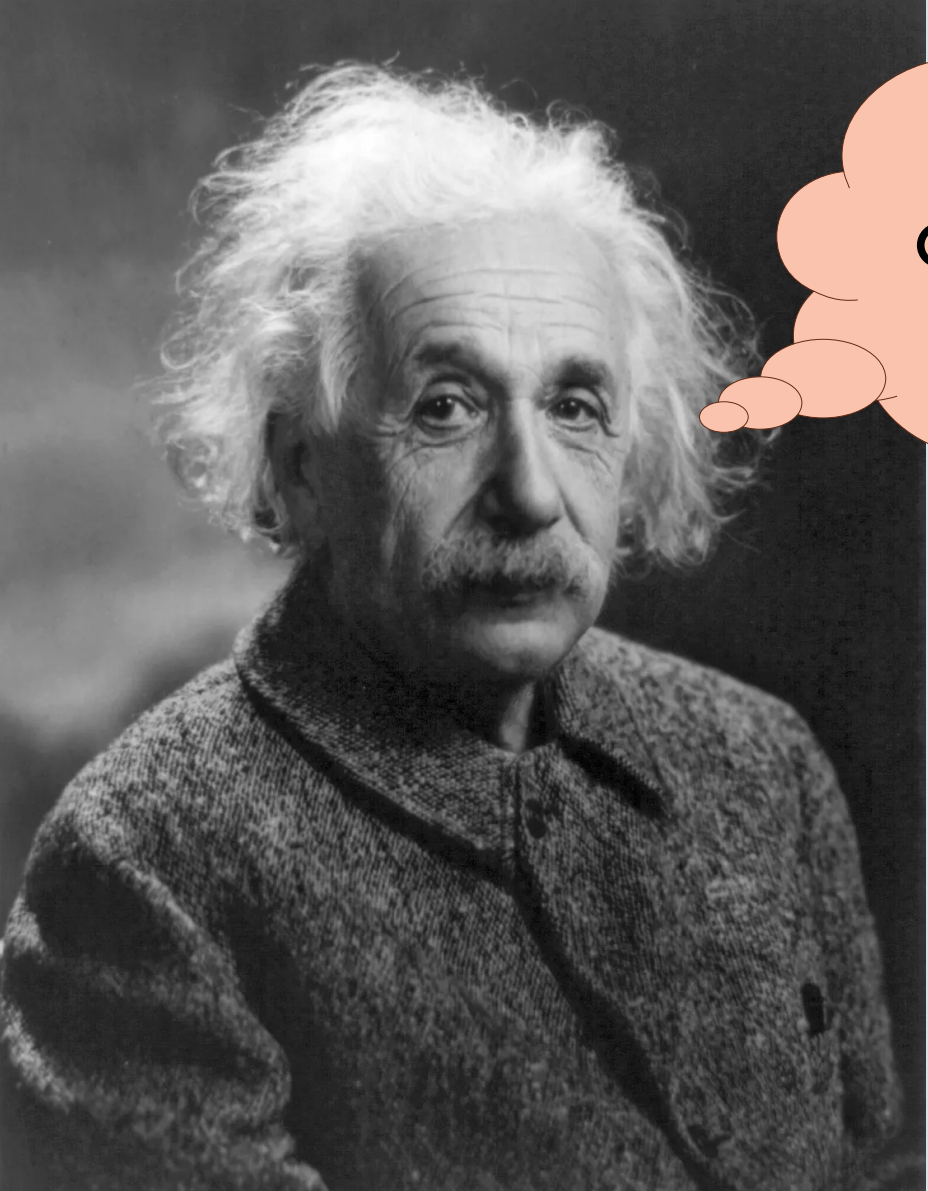
El software de investigación es **crítico**

No se piensa en su **sostenibilidad a largo plazo**

La ciencia en el ORNL requiere mucho **software grande y de vida larga**

A menudo desarrollado por **equipos multidisciplinarios y mult institucionales**

Aplicación de **buenas prácticas** de ingeniería del software



¿Para qué  
quiero saber  
esto?

Elegir actividades de formación científica

Orientación de la carrera profesional

Publicación de resultados científicos específicos más allá del *paper*

Reproducibilidad de experimentos basados en software

Contratación e integración de ingenieros/as de software en equipos de investigación



¿Y yo?



# RSE e información geográfica – University of Nottingham

- *“The Digital Research team are working with the School of Geography to trial a new Research Software Engineering (RSE) service. Our RSE has been across a range of different research projects...”*
- Análisis de *big data* (LiDaR)
  - Automatización y uso de la nube para sustituir a las soluciones HPC
- Modelado del efecto del clima en patrones ecológicos y evolutivos
  - Facilitar el acceso a HPC y escalabilidad en la nube
- Análisis del cierre de bancos en Reino Unido
  - Modelizado basado en agentes con simulación de miles de escenarios en la nube

# RSE e información geográfica – University of Alabama

- *“The goal of the Research Software Engineering (RSE) effort at the University of Alabama is to support the growing need to develop high-quality research software”*
- Modelizado basado en agentes de redes de narcotráfico
- Geovisor de datos de humedad del suelo creados con un modelo de aprendizaje automático tipo *random forest* a partir de datos de satélite
- Aplicación web para mostrar datos geolocalizados sobre el abuso en el uso de opioides

# La ingeniería del software de investigación en la práctica

## Lecciones aprendidas

Dggstools es una biblioteca y herramienta de línea de comandos desarrollada en Python para manipular datos geográficos sobre la malla global discreta (DGGS)  
rHEALPix

La herramienta dggstools es parte del proyecto de I+D+i PID2020-113353RB-I00, financiado por el Ministerio de Ciencia e Innovación de España (MCIN/AEI/10.13039/501100011033/) y del proyecto T59\_23R financiado por el Gobierno de Aragón.

# Disponible en PyPI

## dggstools 0.1.1

```
pip install dggstools
```

[Versión más reciente](#)

Publicación: 17 may 2024

A Python library and command line tool to manipulate raster and vector GIS data in a DGGS (rHEALPix for now).

### Navegación

[Descripción de proyecto](#)[Histórico de versiones](#)[Archivos de descarga](#)

### Verified details

These details have been [verified by PyPI](#)

### Responsables



[rubejar](#)

## Descripción de proyecto

### DGGSTools

DOI [10.5281/zenodo.10659071](https://doi.org/10.5281/zenodo.10659071)

A python library (see the [latest version API docs](#) and command line tool to manipulate raster and vector GIS data in the spatial framework provided by a DGGS (rHEALPix for now).

### Requirements

- Python 3.10 or higher
- `pip` for installing Python packages

### Install the package and the command line tool

# Archivada en Zenodo (repositorio digital) con DOI

May 21, 2024 (v0.1.1a202405211013)

Software

 Open


**dggstools**

Bejar, Ruben 

Tools to manipulate raster and vector GIS data in the spatial framework provided by a DGGS

- Código en GitHub
  - Control de versiones
  - Contribuciones individuales transparentes
  - El README explica lo que hay que hacer para usar la herramienta
  - Publicación de versiones inestables
    - Lo último de lo último siempre está disponible ahí, aunque sea una versión intermedia que no se publica en PyPI
  - Gestor de incidencias (*issues*) para mejoras, cambios y corrección de errores


# La documentación de la API está publicada

  
Search...  

## API Documentation

- `is_a_subarea_rhealpix()`
- `optimize_cuids_rhealpix()`
- `is_optimal_cuids_rhealpix()`
- `class RHEALPixAUIDGenerator`
  - `RHEALPixAUIDGenerator()`
  - `n_side`
  - `generate_auid_hash_b64()`
  - `cuids_from_auid_b64()`

DGGTools version main

built with 

## dggstools.auids.rhpx

Generation of area unique identifiers (AUIDs) for rHEALPix.

[► View Source](#)

```
def is_a_subarea_rhealpix(auid1_comp_b64: str, auid2_comp_b64: str, nil_name: str = '$') -> bool:
```

[► View Source](#)

Takes two AUIDs and returns True if the first one is of an area which is spatially contained in the area defined by the second. If both AUIDs are generated from optimal sets of cuids, this always works (unless some bugs remain). If they are not optimal, it might give some false negatives. E.g.: lets assume N1, N2, N3 and N4 are the same area as N. If we generate an auid for N1, N2, N3 and N4, and another one for N, this function will return a false negative because the first list of cuids is not optimal (N1, N2, N3, N4 has an optimal version which is N).

TODO: This is RHEALPix specific, for now at least; and it has just been lightly tested. It is a work in progress. TODO: Check optimality and maybe give a warning of potential false negatives if inputs are not optimal. Or make it explicit as a parameter to manifest if the inputs are optimal or not.

```
def optimize_cuids_rhealpix(cuids: Sequence[str], n_side: int) -> Sequence[str]:
```

[► View Source](#)

Take a Sequence of cuids and returns an optimal version of it (optimal in the sense explained in the manuscript referred above). This is important because AUIDs are better (shorter, more "unique"...) when we know that they have been generated with optimal sets of cuids.

This is RHEALPix specific, for now at least; and it has just been lightly tested. It is a work in progress.

This becomes slow pretty fast as the resolution of the cuids increases. For a real administrative unit of some 50.000 km<sup>2</sup> at resolution 8 (n\_side 3), it takes about 30 s, while at resolution 9 is som 350 s.



- Entrega continua (*continuous delivery*, CD) con GitHub Actions para automatizar:
  - Pruebas + actualización de la documentación + generación de versión descargable + publicación automática en Zenodo (de versiones estables)

The screenshot displays the GitHub Actions interface for the repository 'IAAA-Lab / dggstools'. The workflow 'Python build and test' is shown, triggered by a push to the 'main' branch. The run is successful and took 5m 14s. The workflow steps are: 'build-and-unit-tests' (41s), 'data-tests / data-tests' (3m 16s), and 'pack-and-release' (55s). The left sidebar shows the 'Summary' tab with a list of jobs: 'build-and-unit-tests', 'data-tests', and 'pack-and-release', all marked as successful. The right panel shows the 'cl.yml' workflow file with the trigger 'on: push'.

IAAA-Lab / dggstools

Type to search

<> Code Issues 25 Pull requests Discussions Actions Projects Wiki Security Insights Settings

← Python build and test

✓ Made get\_gdf\_attrs... method more robust if called on a non-rhealpix ... #56

Summary

Jobs

- ✓ build-and-unit-tests
- ✓ data-tests
- ✓ pack-and-release

Run details

- Usage
- Workflow file

Triggered via push 4 months ago

rbejar pushed → e9adae1 main

Status: Success

Total duration: 5m 14s

Artifacts: -

cl.yml

on: push

build-... / build-and-unit-tests 41s

data-tests / data-tests 3m 16s

pack-and-release 55s

Hay una ponencia de congreso en la que explicamos los objetivos de la herramienta, damos ejemplos de uso etc.

# Article



Articles / Volume 5 / AGILE-GISS, 5, 40, 2024

<https://doi.org/10.5194/agile-giss-5-40-2024>  
© Author(s) 2024. This work is distributed under the Creative Commons Attribution 4.0 License.

Article

Metrics

Related articles

30 May 2024



## DGGSTools: An Open Source Python Package for the Manipulation of Vector and Raster Datasets in the rHEALPix Discrete Global Grid System

[Sergio Martin-Segura](#), [Rubén Béjar](#), and [F. Javier Zarazaga-Soria](#)

**Keywords:** discrete global grid system, rhealpix, raster, vector

**Abstract.** DGGSTools is a Python package and command-line application to process geospatial datasets within the spatial reference provided by the rHEALPix Discrete Global Grid System (DGGs). Its main objective is to provide the tools necessary to transform geospatial datasets to equivalent ones following the rHEALPix DGGs, and to allow its users to work with both raster and vector versions of those datasets as preferred. Once these equivalent datasets have been produced with DGGSTools, the users can obtain the benefits of working with a DGGs, intended to facilitate the efficient integration and retrieval of heterogenous global data, while using their preferred GIS tools and workflows for the analysis of those datasets.

**Download**

[Article \(1099 KB\)](#)  
[Metadata XML](#)  
[BibTeX](#)  
[EndNote](#)

**Share**



# Lecciones

- Los metadatos en Zenodo tienen algunos errores y se publicaron demasiadas versiones preliminares
  - La publicación en Zenodo se automatizó a partir de la construcción en GitHub, pero hubo un poco de fricción hasta que acertamos con el procedimiento adecuado

# Lecciones

- Generar una versión que se despliegue y funcione sin problemas en los entornos más habituales (Windows, MacOS, Linux) requiere probar en todos
  - Y en distintas versiones de los mismos

# Lecciones

- Generamos documentación, pero también falta documentación
  - Es difícil mantener actualizada la documentación de un software en evolución
  - Parte de la documentación (por ejemplo, ejemplos de uso) está en la ponencia de congreso que publicamos, que no es el formato ideal

# Lecciones

- El pipeline de entrega continua debería llegar hasta la publicación en PyPi de versiones estables
  - Esto requeriría probar automáticamente el paquete final en distintas plataformas. Es factible, pero cuesta esfuerzo ponerlo en marcha
  - También deberíamos documentar los cambios concretos de cada versión que se publique (*release notes*)

# Lecciones

- Cuando tienes una buena batería de pruebas automáticas y un pipeline de entrega continua, optimizar las pruebas es importante
  - La mayor parte del tiempo de ejecución del pipeline con Python estará en la ejecución de las pruebas

# La enseñanza de la ingeniería del software de investigación

## Lecciones aprendidas



# Cursos Extraordinarios

**Verano 2024**

## **Introducción a la ingeniería del software para la investigación**



Grupo de Sistemas de  
Información Avanzados  
**Universidad Zaragoza**



Vicerrectorado de  
Cultura y Proyección Social  
**Universidad Zaragoza**

- 1ª edición como curso extraordinario de verano de UNIZAR 23/24
- Abierto a cualquiera, en esta edición tuvimos estudiantes de doctorado de física, ingeniería y arquitectura
  - En algún caso trabajando con información geográfica
- Al ser una primera experiencia las conclusiones que podemos sacar son anecdóticas
  - Pero sí son aspectos que consideraremos en futuras ediciones

- Buen dominio de la línea de comandos, mejor de lo esperado
  - Independientemente de las herramientas existentes, parece que desarrollar esta habilidad es importante si

- Interés en la automatización de pruebas
  - Es una herramienta crítica para mejorar la calidad del software, y se usa mucho menos de lo que se debería

- Interés en mejorar las habilidades de programación
  - Una formación básica seguida de un auto-aprendizaje “bajo demanda” normalmente descuida algunos fundamentos (modularización, mantenibilidad, documentación...)

- Interés en las interfaces gráficas (GUI), aunque esto no nos parecía fundamental
  - Mucho software de investigación no tiene interfaces gráficas (notebooks de Jupyter, herramientas de línea de comandos, bibliotecas...); esto limita su alcance
  - Los cursos de fundamentos de programación no suelen incluir el desarrollo de GUI

- La computación de altas prestaciones no está normalmente bajo el paraguas de la ingeniería del software para la investigación, pero está claro que se beneficiaría de aplicar algunas buenas prácticas de esta última
  - De esto tampoco contamos nada, pero para cierto público sería interesante

- Algunos grupos de investigación tienen pipelines avanzados de construcción, prueba y despliegue automáticos
  - Los doctorandos que entran se encuentra una infraestructura moderadamente compleja y ya establecida, pero que no sabrían replicar si tuvieran que empezar de cero
  - Se podrían diseñar actividades de aprendizaje sobre infraestructuras existentes además de ejercicios básicos partiendo de cero



¿Qué puedo hacer como  
asesor, financiador, revisor de  
proyectos...?

# La declaración de Amsterdam

- Sobre la financiación de la sostenibilidad del software de investigación
- Firmada por organizaciones de más de 20 países, así como por varias regionales y globales
- Hace 12 recomendaciones

(Research Software Alliance, 2024)

# La declaración de Amsterdam

- Estimula la documentación, licenciamiento, distribución como **software libre** y la **accesibilidad** del software de investigación
  - Habilita la reproducibilidad de la investigación
- Incentiva la **reutilización** y la **mejora** del software de investigación existente
- Incluye el software de investigación en las políticas de **ciencia abierta**
  - Para asegurar que es un resultado de investigación valioso y con impacto
  - *As open as possible, as closed as necessary*

# La declaración de Amsterdam

- Estimula el desarrollo y mantenimiento de un **ecosistema** de software de investigación para garantizar su **sostenibilidad**
  - Gente, comunidades e infraestructura
- Comparte información y trabaja de manera **coordinada**
  - El ecosistema de software de investigación sobrepasa cualquier frontera institucional o nacional
- Crea instrumentos de financiación adecuados para la **sostenibilidad** y la **innovación**
  - Mantenimiento y desarrollo a largo plazo

# La declaración de Amsterdam

- Estimula la formación, contratación y financiación de **personal investigador y técnico capaz** de reutilizar, desarrollar y mantener software de investigación sostenible
- Facilita el reconocimiento y la compensación adecuada para habilitar una **carrera profesional**
  - Para toda la gente involucrada en la creación y mantenimiento del software de investigación
- Exige que las **citas** del software de investigación reconozcan las contribuciones sustanciales en todos los aspectos del software

# La declaración de Amsterdam

- Estimula el uso responsable de indicadores adecuados para la **continuidad**, **reusabilidad** e **impacto** del software de investigación
- Considera el **impacto social y medioambiental** del uso del software de investigación
- Reconoce que la **diversidad**, la **equidad** y la **inclusividad** son factores significativos para hacer que el software de investigación sea sostenible

¿En qué me tengo que formar?

¿Qué habilidades tengo que  
incorporar a mi equipo?

*“Software Engineering Practices in  
Academia: Promoting the 3Rs—  
Readability, Resilience, and Reuse”*

(Connolly et al., 2023)



# Buenas prácticas de ingeniería del software - Control de versiones

- Por tu cuenta
  - Repositorios de código fuente (p.ej., en GitHub), mensajes de commit con sentido, backups de tus repositorios locales
- En tu grupo de investigación
  - *Feature commits*, historias de commit limpias (rebase vs. merge)
- Para una comunidad de investigación
  - Documenta tus políticas para la revisión y aceptación de commits, creación de ramas etc.

# Buenas prácticas de ingeniería del software - Diseño/programación de software

- Por tu cuenta
  - Discute con tus colegas, convenciones de nombrado de objetos de código consistentes
- En tu grupo de investigación
  - Modularización (p.ej., orientación a objetos), revisa la experiencia de uso con más gente, revisiones de código entre pares, consistencia en el estilo del código gestionada automáticamente (p.ej., <https://editorconfig.org>), construcción basada en scripts (p.ej., <https://python-poetry.org>)
- Para una comunidad de investigación
  - Especificaciones funcionales (casos de uso, desarrollo dirigido por el comportamiento), especificación de API, diagramas de diseño

# Mejora UI del login y dashboard de usuario del cliente SITMUN 3



## Aspectos generales

**Estimación horas de desarrollo:** 40 horas

**Entidad solicitante:** Comisión técnica SITMUN

**Prioridad:** ALTA

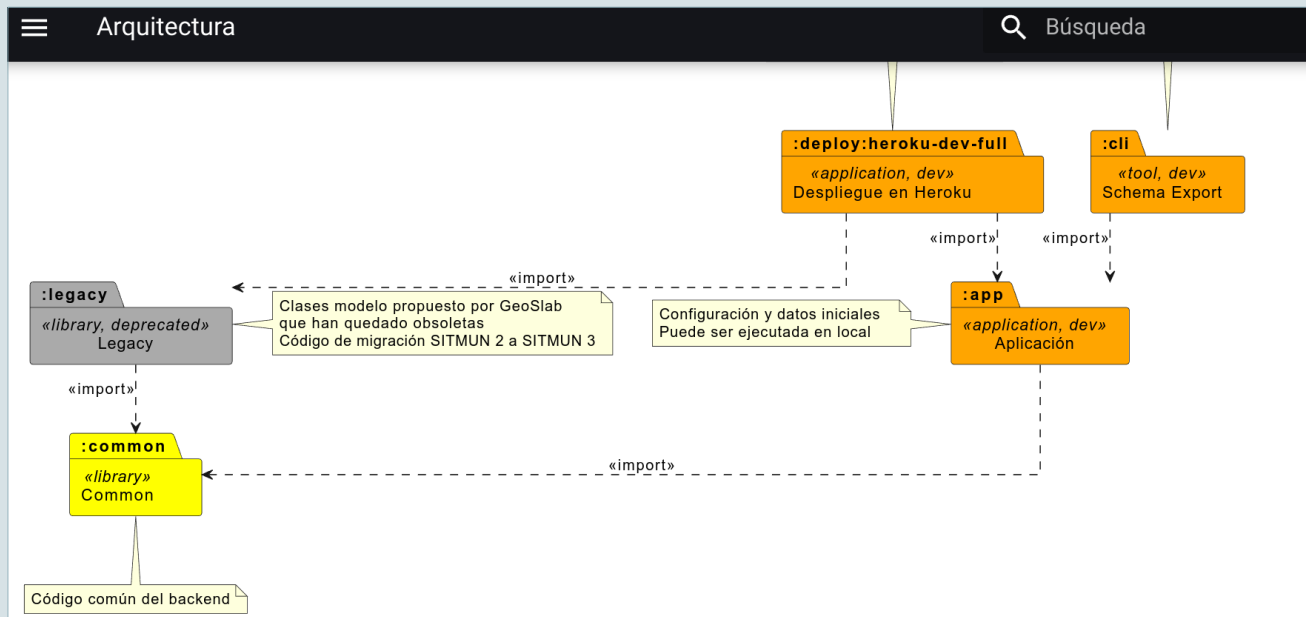
**Persona o entidad de referencia o contacto:** Consell Insular de Menorca

## Objetivo

Este proyecto busca mejorar el diseño y la funcionalidad de la interficie de acceso y selección de los clientes SITMUN disponibles para un usuario concreto, dotándolas de un diseño más amigable, claro y funcional. Se busca además ampliar la funcionalidad actual.

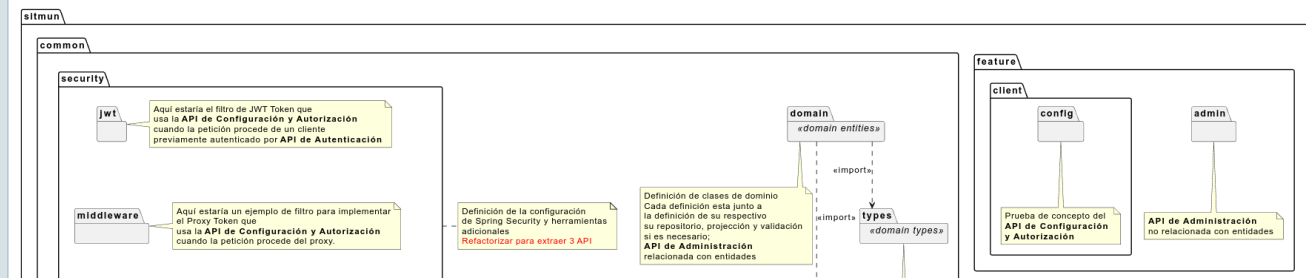
## Contexto y casos de uso

En las imágenes siguientes se puede ver la actual ventana de login y selección de aplicación del actual cliente SITMUN 3.



## Diagrama de paquetes

Es la biblioteca que contiene la definición del modelo de datos de **SITMUN 3**, la API de autenticación, la API de configuración y autorización y la política de seguridad.



# Buenas prácticas de ingeniería del software - Aseguramiento de la calidad

- Por tu cuenta
  - Pruebas unitarias automáticas, ejecución automática de las mismas localmente
- En tu grupo de investigación
  - Amplia cobertura de pruebas, incluye pruebas de integración/sistema, integración/despliegue continuos (p.ej., con GitHub Actions)
- Para una comunidad de investigación
  - Escribe pruebas automáticas para cualquier error que haya ocurrido en alguna versión anterior en producción (pruebas de regresión), prueba aspectos no funcionales (prestaciones, seguridad...), prueba en todas las plataformas soportadas, facilita el feedback de las personas que usan tu software



HEROKU

Jump to Favorites, Apps, Pipelines



sitmunteam



sitmun-backend-core

Overview

Resources

Deploy

Metrics

Activity

Access

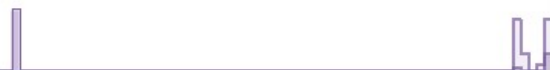
Settings

Metrics (last 24hrs)

[All Metrics](#) ➔

Response Time

703 ms



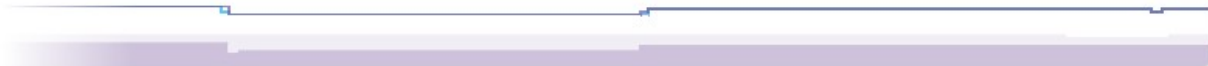
Throughput

< 1 rps



Memory

160 %



# Buenas prácticas de ingeniería del software - Despliegue y documentación para usuarios/as

- Por tu cuenta
  - Etiquetado de versiones con tags, empaqueta y despliega en tu repositorio (p.ej. GitHub releases), comentarios de código, ejemplos de uso ejecutables (p.ej., un notebook de Jupyter)
- En tu grupo de investigación
  - Versionado semántico, genera la documentación de tus API (p.ej., <https://www.sphinx-doc.org/en/master/>). genera documentación de uso (visión general, ejemplos, opciones...), publica en los índices de paquetes habituales (p.ej., <https://pypi.org>)
- Para una comunidad de investigación
  - Documenta los cambios en cada nueva versión pública, automatiza la generación de la documentación de tus API, publica la documentación de tus API (p.ej., <https://about.readthedocs.com>), documenta el despliegue con contenedores, escribe tutoriales, publica información para facilitar las contribuciones (onboarding)

# SITMUN 3 - API de Autenticación

0.8.0

OAS3

[/v3/api-docs-auth.yaml](#)

La **API de Autenticación** expone vía una API Web mecanismos para interactuar con el sistema de seguridad de SITMUN. Esta API se ha creado para que la **aplicación de administración** y los **visores de mapas** obtengan, tras pasar las credenciales de usuario, el JSON Web Token necesario para operar con el resto de las API.

## La documentación de esta API está en desarrollo

El contenido seguirá mejorando (aunque serán cambios menores) en los próximos días para ayudar a los desarrolladores que están trabajando en el proyecto SITMUN.

[Comité técnico SITMUN - Website](#)

[EURL 1.2](#)

### Servers

<https://sitmun-backend-core.herokuapp.com> - Servidor de desarrollo

## Autenticación

POST

**/api/authenticate** Accede con correo electrónico / contraseña.



# Buenas prácticas de ingeniería del software - Gestión

- Por tu cuenta
  - No te hace falta mucha gestión si trabajas por tu cuenta
- En tu grupo de investigación
  - Planifica cada lanzamiento (fecha, características que se incluirán), o al menos haz una priorización (p.ej. MoSCoW: must, should, could, won't) y estima dificultades/tiempos de desarrollo, organiza y reparte tareas en tu equipo (p.ej., GitHub Issues), presenta los avances a tu equipo
- Para una comunidad de investigación
  - Planifica y asigna tareas (si lideras la comunidad), diseña y publica un roadmap (incluyendo planes de desarrollo, pero también eventos, avisos de cambios futuros etc.), reuniones regulares para compartir progresos y solucionar bloqueos, revisión/experimentación de tecnologías para incluir nuevas dependencias en el proyecto, define una estructura de gobernanza de la comunidad, establece canales de feedback para priorizar nuevas funcionalidades y corrección de errores

¡Bienvenido a la hoja de ruta de SITMUN 3! Estas son las prioridades del equipo SITMUN.

El objetivo de esta hoja de ruta es ofrecer una visión general del estado del proyecto SITMUN 3 y las próximas funcionalidades a desarrollar.



#### Aspectos generales

**El Comité Técnico de SITMUN** es el órgano funcional encargado de revisar la descripción o, en su defecto, describir el alcance funcional de cada uno de los puntos. También de proponer la priorización de cada uno de los proyectos o funcionalidades.

Durante la **Asamblea Ordinaria Anual de socios del proyecto SITMUN** se vota y aprueba la priorización.

## Proyectos en desarrollo

Proyecto - Funcionalidad	Descripción	Estado
Cliente SITMUN3 - API SITNA	Creación de un cliente SITMUN3 genérico a partir de la integración de <a href="#">API SITNA</a> y con el layout definido para la IDE Menorca	Revisión de bugs y documentación

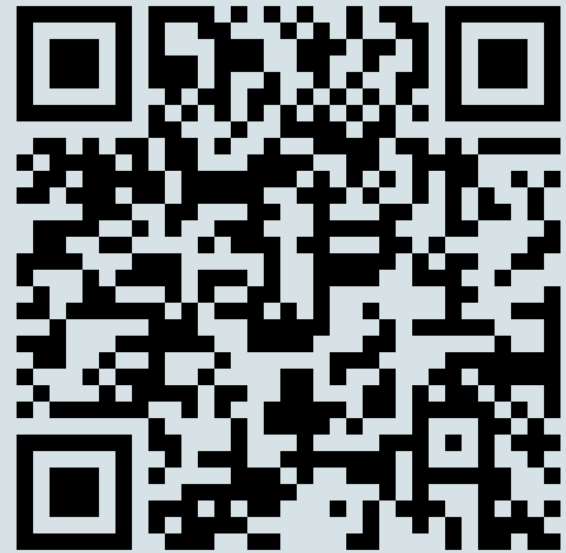
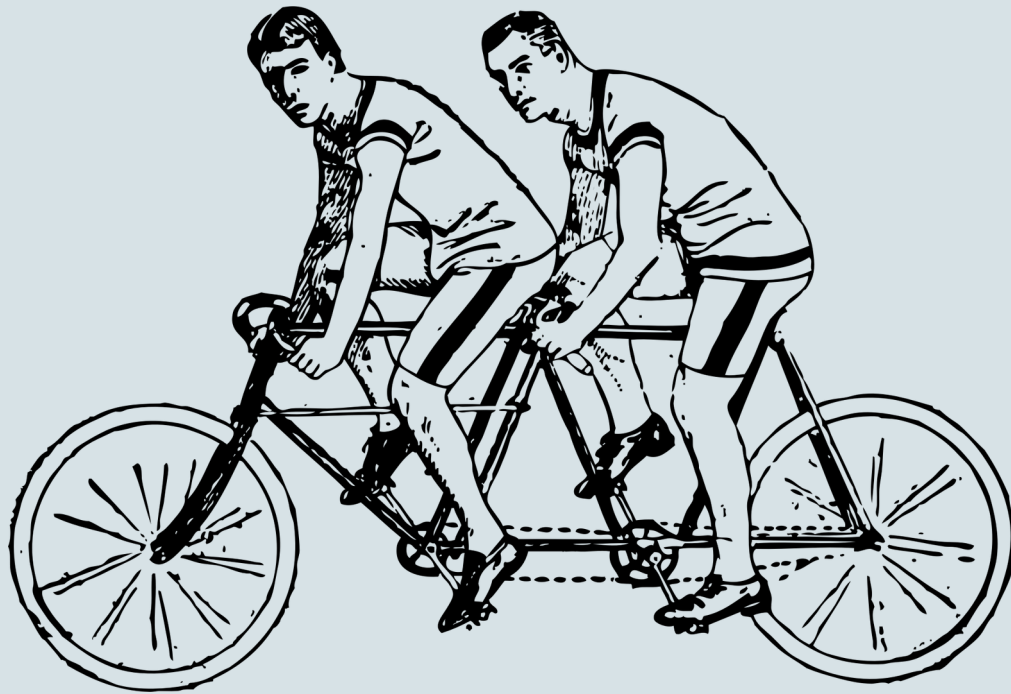
## Proyectos prioritarios

Proyecto - Funcionalidad	Descripción	Prioridad
-----------------------------	-------------	-----------

# Referencias

- ADORE.software Toolkit, 2024, <https://adore.software/toolkit/>
- Carver et al. A survey of the state of the practice for research software in the United States. 2022, <http://dx.doi.org/10.7717/peerj-cs.963>
- Connolly, A., Hellerstein, J., Alterman, N., Beck, D., Fatland, R., Lazowska, E., Mandava, V., & Stone, S. (2023). Software Engineering Practices in Academia: Promoting the 3Rs—Readability, Resilience, and Reuse. Harvard Data Science Review, 5(2). <https://doi.org/10.1162/99608f92.018bf012>
- European Commission: Directorate-General for Research and Innovation, Scholarly infrastructures for research software – Report from the EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS, Publications Office, 2020, <https://data.europa.eu/doi/10.2777/28598>
- Gruenpeter et al. Defining Research Software: a controversial discussion. 2021, <https://doi.org/10.5281/zenodo.5504016>
- Malviya-Thakur et al. Research Software Engineering at Oak Ridge National Laboratory. 2023, <https://www.doi.org/10.1109/MCSE.2023.3260211>
- Research Software Alliance. (2024). Amsterdam Declaration on Funding Research Software Sustainability (1.1). Zenodo. <https://doi.org/10.5281/zenodo.13735888>
- Web del Proyecto SITMUN, (2024). <https://sitmun.org/Contingut.aspx?idPub=8437>

# LA INGENIERÍA DEL SOFTWARE EN LA INVESTIGACIÓN GEOGRÁFICA



<https://www.iaaa.es/blog/>