

# ESUG2024ShowUsYourProject

31st July 2024 at 9:14am

"Minor spoilers ahead!"

`self` spoil: #OwnTalk

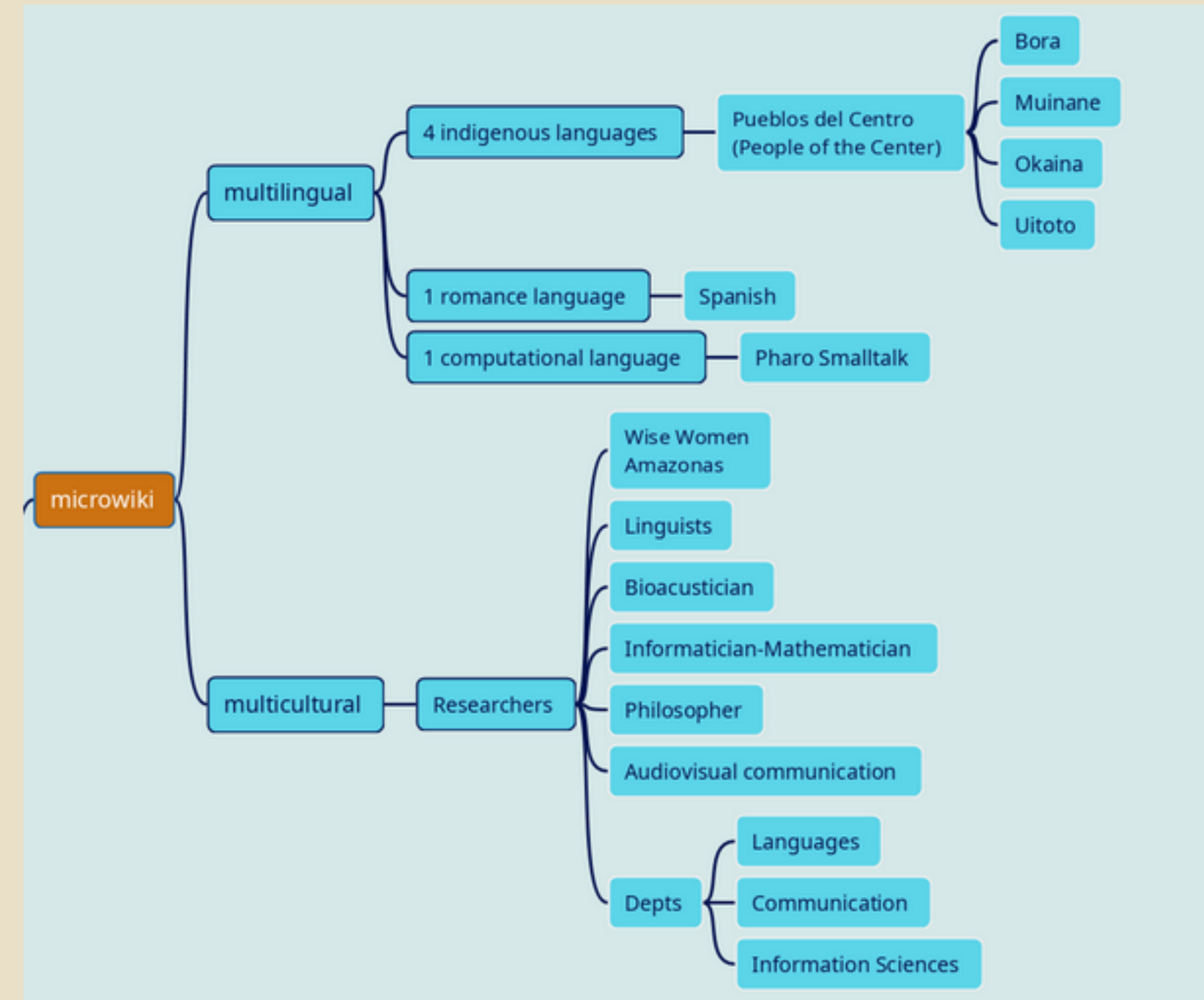
"At least a little bit... while leaving the most for the `_last talk_` of the `_last day_`"

In data stories with Pharo, we will introduce the story of a multilingual/multicultural microwiki.

See this talk as an interactive online page at: <https://is.gd/ESUG2024quick>

# MicrowikiCartofonias

9th July 2024 at 5:03am



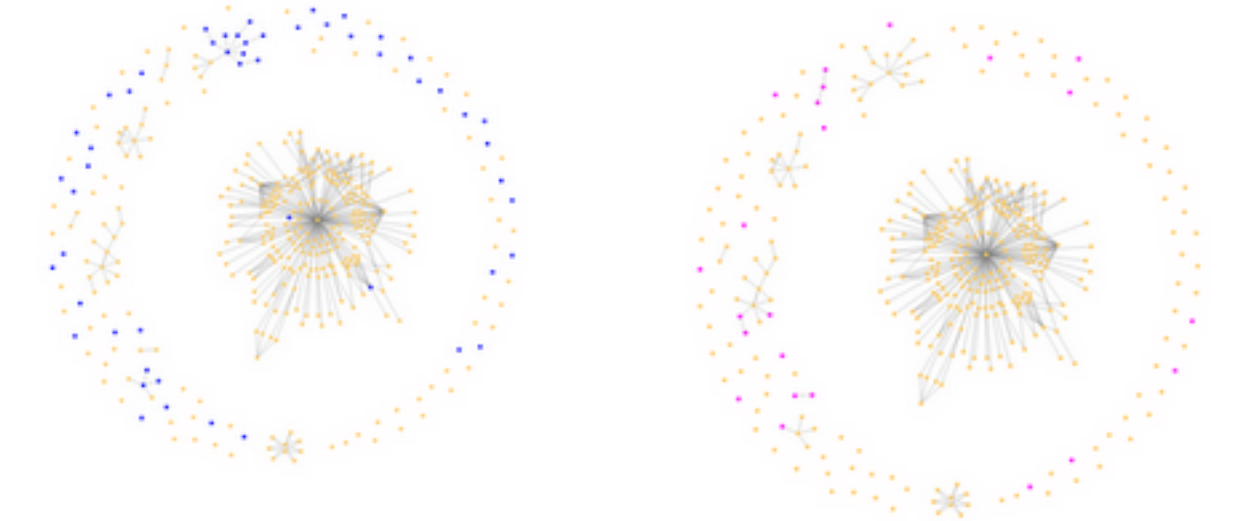
## Notebooks

# Notebooks

9th July 2024 at 7:11am

We bridge what we do via notebooks [like this one](#):

Si reunimos las gráficas anteriores, los tiddlers cuyos históricos queremos versionar son los que aparecen en siguientes gráficas, coloreados en tanto en azul como en magenta y que están almacenados en la variable `ourTiddlers`:



Definamos el lugar del disco duro donde guardaremos el microwiki:

```
microwiki
  folder: FileLocator documents / 'U/Javeriana/AmanecerLaPalabra/wikis'.
```

Ahora guardemos cada uno de los tiddlers creados para el proyecto en la subcarpeta `tiddlers/` en dicha localización:

```
ourTiddlers do: [:tiddler | tiddler exportSTONFile ].
microwiki folder / 'tiddlers'.
```

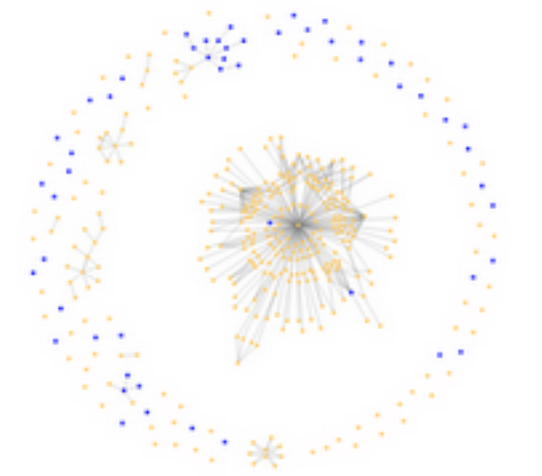
Icon	Name	Size	Creation
-	-	--	2024-06-22 13:49:46
📁	AlterItem-igCrJZFhdpH8.ston	265 B	2024-06-22 13:50:05
📁	AmanecerLaPalabra-TJecu08fuaat.ston	1.10 kB	2024-06-22 13:49:46
📁	BotonNuevoCartofonia-mNvX3N48Qz.ston	335 B	2024-06-22 13:49:47
📁	CajaDeHerramientas-mm0nT20vXkY.ston	496 B	2024-06-22 13:49:47
📁	CantoLagartja-TGCP0w0PpZa.ston	928 B	2024-06-22 13:49:47
📁	CartofoniasMacros-ICM21cJFk2r.ston	1.26 kB	2024-06-22 13:49:50
📁	Cartofonia-Duablo-Bora-W0d4E3DRhPL.ston	434 B	2024-06-22 13:49:48

# Notebooks

9th July 2024 at 7:11am

We bridge what we do via notebooks like

Si reunimos las gráficas anteriores, los tiddlers que aparecen en siguientes gráficas, coloreados en t almacenados en la variable ourTiddlers:

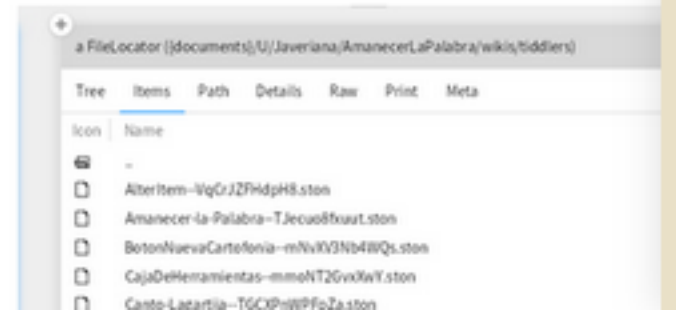


Definamos el lugar del disco duro donde guardare

```
microwiki
folder: FileLocator documents / 'U/Javer
```

Ahora guardemos cada uno de los tiddlers creados en dicha localización:

```
ourTiddlers do: [:tiddler | tiddler exportST
microwiki folder / 'tiddlers'.
```



# NotebooksPainPointsAndOpportunities

9th July 2024 at 7:36am

But notebooks have a lot of pain points, and so opportunities.

## What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities

Souti Chattopadhyay<sup>1</sup>, Ishita Prasad<sup>2</sup>, Austin Z. Henley<sup>3</sup>, Anita Sarma<sup>1</sup>, Titus Barik<sup>2</sup>  
Oregon State University<sup>1</sup>, Microsoft<sup>2</sup>, University of Tennessee-Knoxville<sup>3</sup>  
{chattops, anita.sarma}@oregonstate.edu, {ishita.prasad, titus.barik}@microsoft.com, azh@utk.edu

**ABSTRACT**  
Computational notebooks—such as Azure, Databricks, and Jupyter—are a popular, interactive paradigm for data scientists to author code, analyze data, and interleave visualizations, all within a single document. Nevertheless, as data scientists incorporate more of their activities into notebooks, they encounter unexpected difficulties, or pain points, that impact their productivity and disrupt their workflow. Through a systematic, mixed-methods study using semi-structured interviews ( $n = 20$ ) and survey ( $n = 156$ ) with data scientists, we catalog nine pain points when working with notebooks. Our findings suggest that data scientists face numerous pain points throughout the entire workflow—from setting up notebooks to deploying to production—across many notebook environments. Our data scientists report essential notebook requirements, such as supporting data exploration and visualization. The results of our study inform and inspire the design of computational notebooks.

**Author Keywords**  
Computational notebooks; challenges; data science; interviews; pain points; survey

**CCS Concepts**  
•Human-centered computing → Interactive systems and tools; Empirical studies in HCI; Software and its engineering → Development frameworks and environments;

**INTRODUCTION**  
Computational notebooks are an interactive paradigm for combining code, data, visualizations, and other artifacts, all within a single document [21, 36, 32, 30]. This interface, essentially,

Azure,<sup>1</sup> Databricks,<sup>2</sup> Colab,<sup>3</sup> Jupyter,<sup>4</sup> and nteract.<sup>5</sup> While originally intended for exploring and constructing computational narratives [29, 31], data scientists are now increasingly orchestrating more of their activities within this paradigm [33]: through long-running statistical models, transforming data at scale, collaborating with others, and executing notebooks directly in production pipelines. But as data scientists try to do so, they encounter unexpected difficulties—pain points—from limitations in affordances and features in the notebooks, which impact their productivity and disrupt their workflow.

To investigate the pain points and needs of data scientists who work in computational notebooks, across multiple notebook environments, we conducted a systematic mixed-method study using field observations, semi-structured interviews, and a confirmation survey with data science practitioners. While prior work has studied specific facets of difficulties in notebooks [24, 17], such as versioning [18, 19] or cleaning unused code [13, 34], the central contribution of this paper is a taxonomy of validated pain points across data scientists' notebook activities.

Our findings identify that data scientists face considerable pain points through the entire analytics workflow—from setting up the notebook to deploying to production—across many notebook environments. While our participants reported workarounds, these were ad hoc, required manual interventions, and were prone to errors. Our data scientist report their key needs are support for deploying notebooks to production and scheduling time-consuming batch executions as well as under-the-hood software engineering support for managing code and history. Our findings further our understanding of requirements for supporting data scientists' day-to-day ac-

Table 2: Summary of Pain Points in Computational Notebooks

PAIN POINT	DESCRIPTION	EXAMPLE
<b>Setup</b>	Loading and cleaning data from multiple sources and platforms is a tortuous, multi-step, manual process.	"If you do a lot of data loading and pre-processing always re-loading the data is time consuming" (IP2).
<b>Explore and Analyze</b>	An unending cycle of copy-paste and tweaking bits of code made worse by feedback latency and kernel crashes.	"I need immediate feedback, like when I am testing slight changes in the model. I don't want to execute everything again" (IP1).

# GrafoscopioPortingIntro


9th July 2024 at 9:04am

Grafoscopio (2014-2019): a computational notebook for data storytelling in Pharo

## Grafoscopio

reproducible research & publishing ⊕ interactive documentation ⊕ flexible & agile data visualization


**Write, structure & unify**



Different types of documents all them in a single site. Group, give hierarchy and visualize what you want and create different views of the same informaion.

[LOOK AN EXAMPLE >](#)

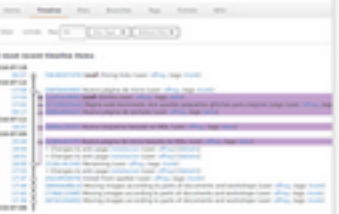
**Compute & visualize**



Connect several & heterogeneous data sources and integrate them through powerfull and flexible custom visualizations.

[LOOK AN EXAMPLE >](#)


**Collaborate & publish**



Templates and the historical control change, will ease the individual and collective work. Also publish in PDF and HTML and share the last version with your audience.

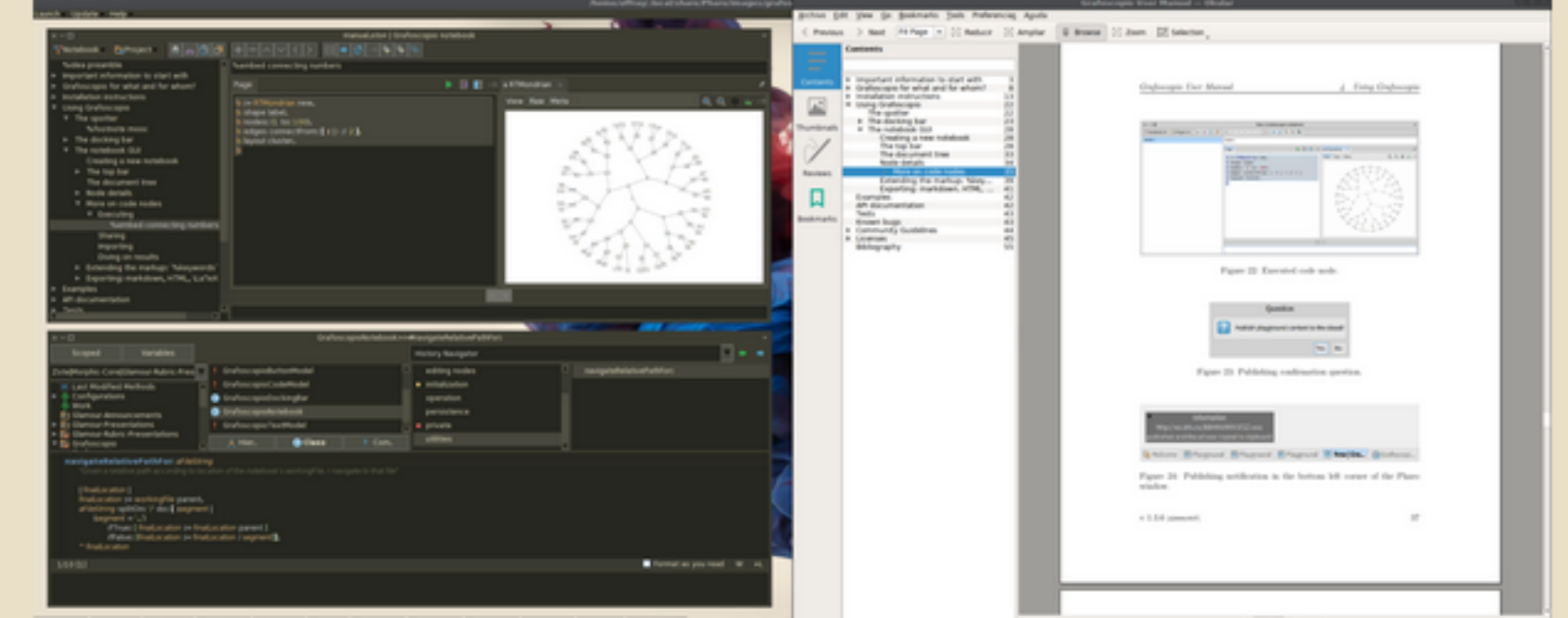
[LOOK AN EXAMPLE >](#)

**Explore & modify**



Use the powerful and interactive integrated development environment to see how the tool is made and to adapt it to different needs.

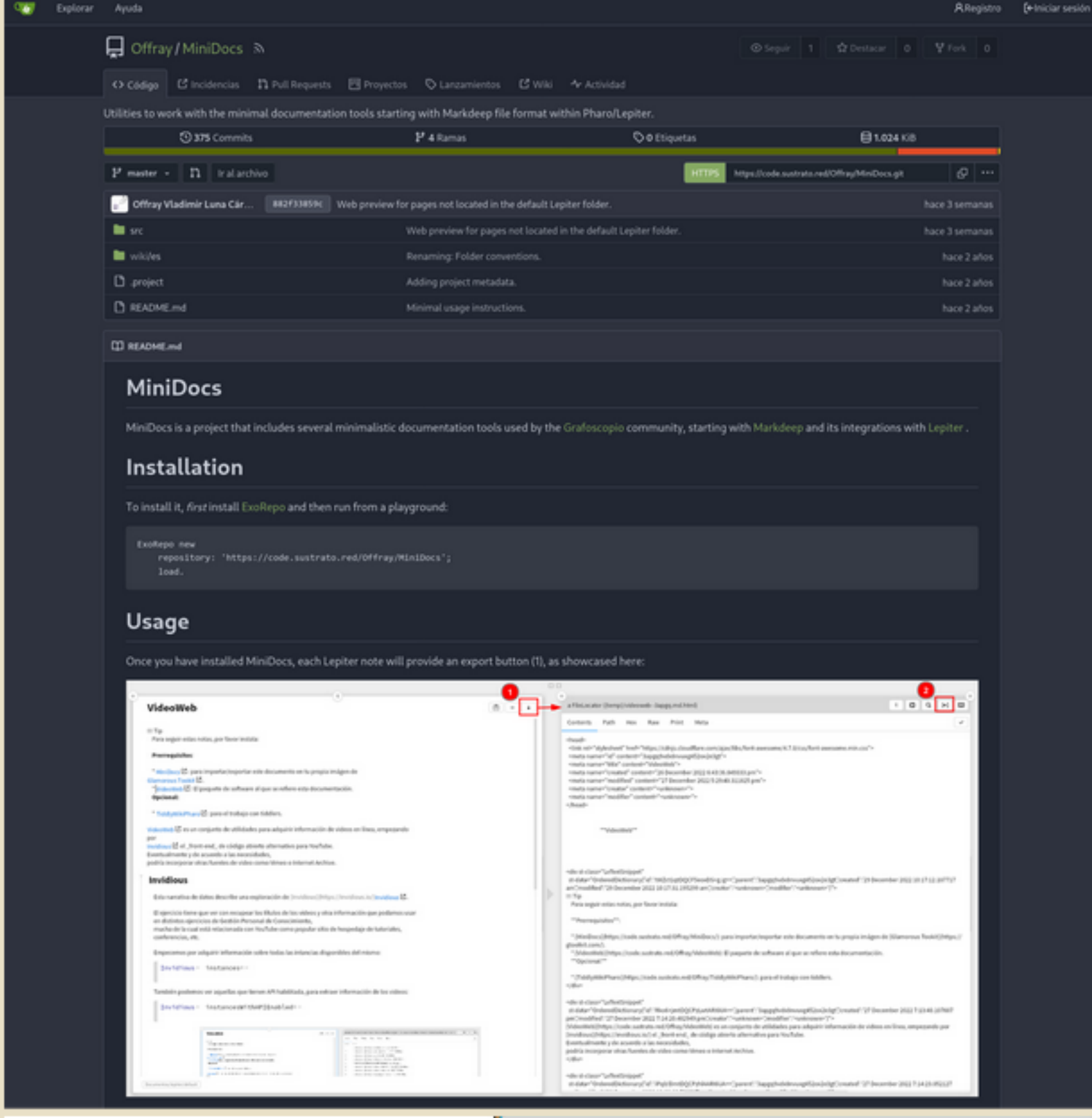
[MORE INFO >](#)



# MiniDocs

9th July 2024 at 9:08am

MiniDocs ports the Grafoscopio lessons from "plain" Pharo to GT/Lepiter.



Offray / MiniDocs

Utilities to work with the minimal documentation tools starting with Markdeep file format within Pharo/Lepiter.

375 Commits 4 Ramas 0 Etiquetas 1.024 KiB

Offray Vladimir Luna Cár. #82F3859C Web preview for pages not located in the default Lepiter folder. hace 3 semanas

src Web preview for pages not located in the default Lepiter folder. hace 3 semanas

wikiles Renaming Folder conventions. hace 2 años

project Adding project metadata. hace 2 años

README.md Minimal usage instructions. hace 2 años

### MiniDocs

MiniDocs is a project that includes several minimalistic documentation tools used by the *Grafoscopio* community, starting with *Markdeep* and its integrations with *Lepiter*.

### Installation

To install it, first install *ExoRepo* and then run from a playground:

```
ExoRepo new
repository: 'https://code.sustrato.red/Offray/MiniDocs';
load.
```

### Usage

Once you have installed MiniDocs, each Lepiter note will provide an export button (T), as showcased here:

