

# LA Documentation and Issue Management

ALA-LAs Workshop (Madrid, March 2025)

PART 2

Vicente J. Ruiz Jurado (@vjrj), 2025/03/11



# Introduction

When I started this job six years ago, my contract with GBIF explicitly stated that I had to **support** the community—through platforms like Slack or email—and **maintain our documentation**.

Since ALA, if I'm not wrong, heavily relies on **internal documentation** and communication (with some exceptions in README files or migration pages), this is no simple task. It wasn't easy when I first arrived, and it remains a challenge today.

I am not an ALA developer. I am just another member of the LA community—perhaps with the difference that I am solely responsible for maintaining our GBIF.es portal.

# A Methaphor

During this time, I've often felt like the bartender in a railway café (\*), constantly being asked all sorts of questions by the passengers, as if I were part of the railway company (ALA). But that's not the case—I don't have direct communication with the engineers.

However, by being in the café, talking to everyone, and trying to help, I have gathered bits and pieces of information from different sources.

BTW: Dave was a great help when he was at ALA, but even with his guidance, this has always been a difficult task.

(\*) Incidentally, this was an actual job I had during my studies...

# Documentation in Other FLOSS Projects

In many FLOSS projects, documentation is a collective effort among developers and contributors, often managed through wikis or Git and pull requests.

In contrast, ALA's approach is different.

# Documentation during this workshop

Let's review how we have contributed to ALA's documentation over the past few days and the wiki history.

Yes, the first part of this presentation was a data trap ... 😊

Anyway, a good part of our wiki was documented using the Google Doc that Dave prepared for the Paris Workshop 2019.

# Dependencies

Over time, I have tried to document the dependencies between ALA modules—something that is crucial in projects like this one.

For example, see the [Elasticsearch support matrix](#), which provides clear guidance on version compatibility.

Unfortunately, within ALA, this information is not shared (at least publicly).

When developing software, it's easy to document API changes and specify which modules require a particular API version.

But sadly, we learn about this breaking changes and incompatibilities in production.

# Dependencies (II)

Everything I have managed to collect so far is available at:

- [ALA Documentation Wiki - Dependencies](#)
- [LA Toolkit Backend - Dependencies YAML](#)

The **LA Toolkit** integrates the last, a self-collected dependency list as a basis for its recommendations.

However, it would be far better if ALA shared this information officially, rather than relying on individual efforts like mine.

Otherwise, the only alternative is trial and error when installing components, which is far from ideal.

# Conclusion

Comments, discussions, opinions?

Let's talk!