



SOFTWARE ARCHITECTURE

2024-25

Jose Emilio Labra Gayo
Pablo González
Irene Cid Rico

Lab 1

- Intro to labs
- Teams organization
- Git
- GitHub



Escuela de
Ingeniería
Informática



Universidad de Oviedo

● Intro to labs

What are we going to do in these sessions?

Design and develop an online game of questions and answers that are automatically generated using Wikidata.

Resources?

- <http://arquisoft.github.io>: course documents.
- [Virtual Campus](#)
- WIQ Lab [specification](#)
- Project github [repositories](#) .

Lab assessment?

70% - Teamwork 

30% - Individual work 

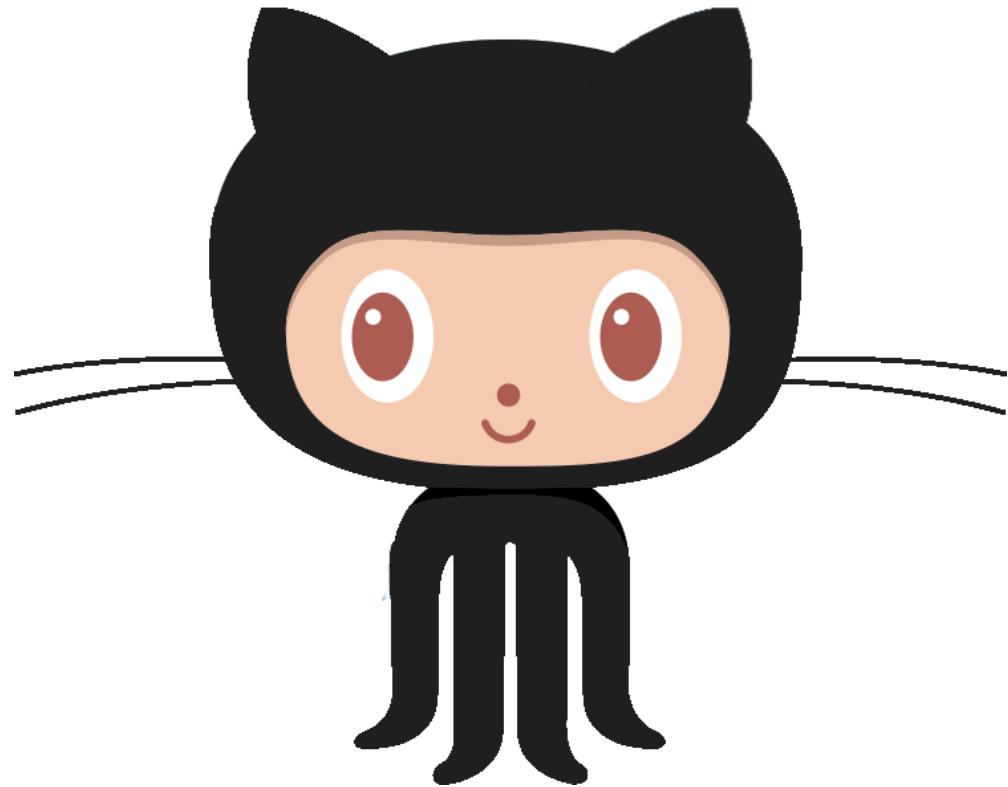
● Team work

Meeting minutes

- Each lab session == group meeting.
 - Other meetings allowed out of lab sessions
- **Mandatory** to create minutes of each meeting
- Wiki section of each repository will be used to record minutes
- Minimal mandatory format :
 - Date 
 - Participant list 
 - Agreements about work assignment for next session (open issues) 
 - Review state of tasks from past meetings
 - Links to **Issues** and **Pull requests** 
 - Short description of decisions taken
 - Preferable to include links architecture decision records (<https://adr.github.io/>)

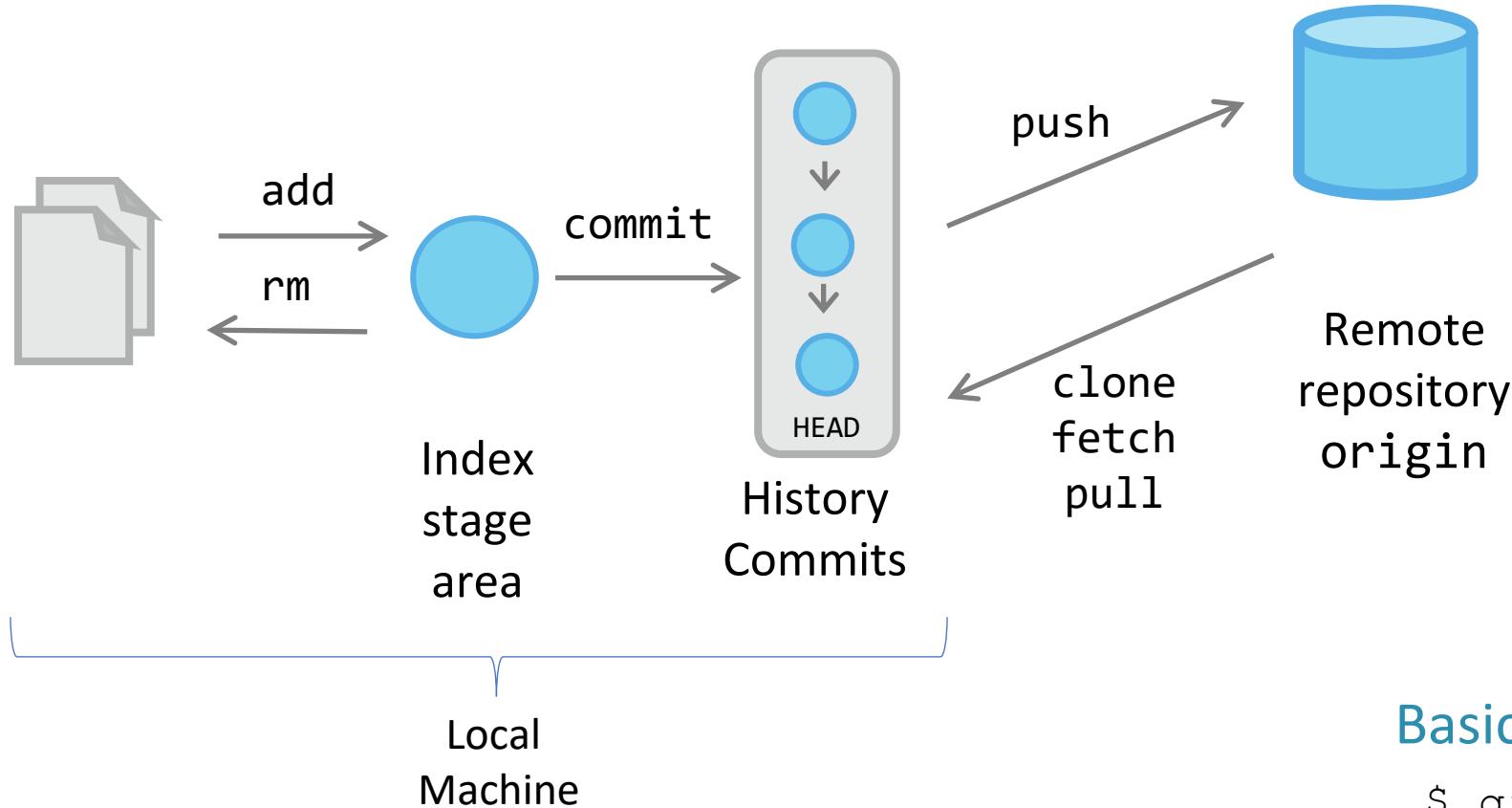


Git





Git



Basic workflow

```
$ git init  
$ git clone urlRepository  
$ git add .  
$ git commit -m "message"  
$ git push origin master
```



Git

Working with branches

Create a branch:

```
$ git checkout -b branch1
```

Check our current branch:

```
$ git branch
```

Change to another branch:

```
$ git checkout master
```

See differences from branches

```
$ git diff --stat master branch1
```

Merge branch:

```
$ git checkout master
$ git merge --no-ff branch1
```

Remove branch:

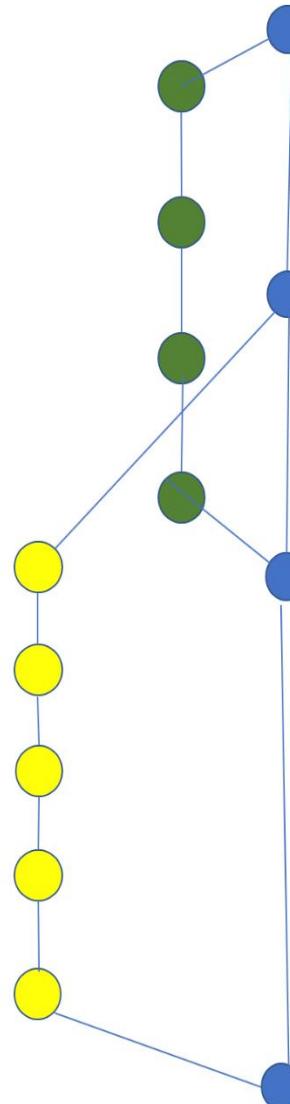
```
$ git branch -d branch1
```

Create the develop branch:

```
$ git checkout -b develop
```

Push it to the remote repository:

```
$ git push origin develop
```





Git

Branching strategies

- Several strategies whose success depends on several factors.
 - See: <https://martinfowler.com/articles/branching-patterns.html>
- Some popular patterns:
 - Git-flow, by Vincent Driessen, 2010: [A successful Git branching model](#)
 - [GitHub Flow](#) strategy
 - Trunk based development: <https://trunkbaseddevelopment.com/>



Pull Request

Steps

New branch

```
$ git flow feature start RE1 develop #option 1  
$ git checkout -b feature-RE1 develop #option 2
```

Add your name inside the *Collaborator* section in your *README.md* file

Send your local changes

```
$ git add .  
$ git commit
```

Submit your changes to remote

```
$ git push --set-upstream origin feature-RE1
```

Go to github and **ask for a pull request**

The screenshot shows a GitHub repository profile with the following details:

- 5 commits
- 3 branches
- 0 packages
- 0 releases

Your recently pushed branches:

develop (about 1 hour ago)	Compare & pull request
feature/RE1 (1 minute ago)	Compare & pull request



Pull request

The screenshot shows the GitHub interface for creating a pull request. The top navigation bar includes 'Code', 'Issues 4', 'Pull requests 2', 'Actions', 'Projects', 'Wiki', 'Security 1', 'Insights', and 'Settings'. The main section is titled 'Open a pull request' with the sub-instruction: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)'. The pull request creation form has the following steps:

1. 'base: master' dropdown.
2. 'compare: feature-new-cool-thing' dropdown, with a note: 'Able to merge. These branches can be automatically merged.'
3. User profile picture and commit message: 'Fix problem when creating an user'.
4. 'Write' tab selected, showing the commit message content:

```
# Changes

Use uuid v4 instead a manual generated id with Math.random to avoid problems when using the id in some places.

# Fixed issues

Fixed #30 on generating new users
```
5. 'Create pull request' button.

Annotations with arrows point to specific elements:

- 'Target branch' points to the 'base' dropdown (step 1).
- 'Branch we want to merge' points to the 'compare' dropdown (step 2).
- 'Add more context' points to the commit message area (step 4).
- 'Create pull request' points to the final button (step 5).

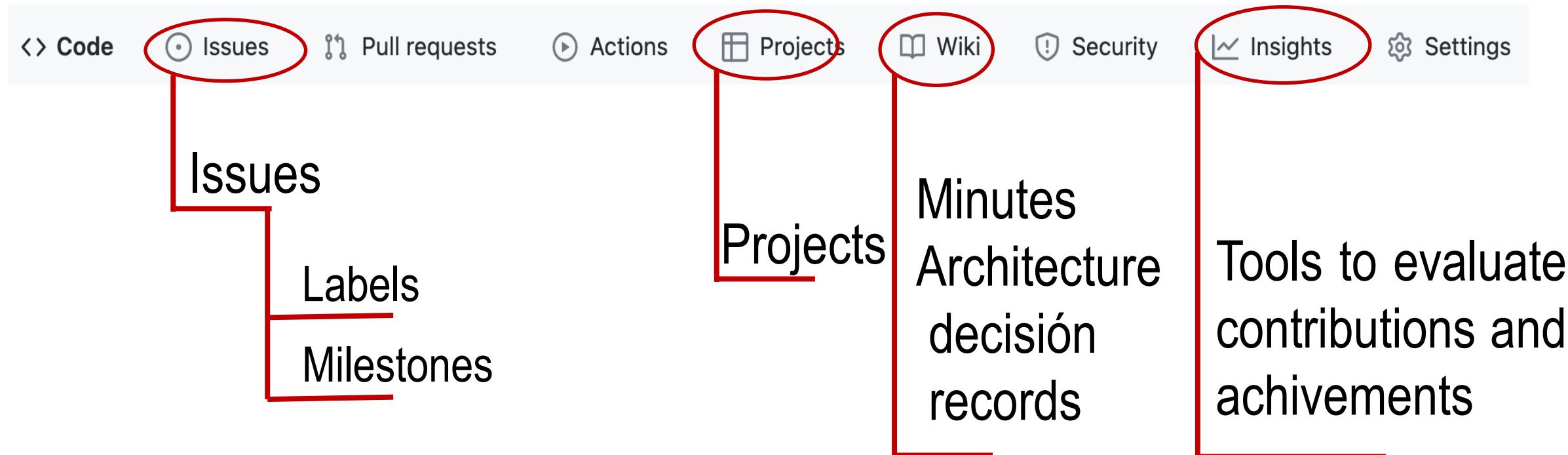
● GitHub as a Project management tool

Advantages of Project management

- Project planning (future)
- More control over current project's state (present)
 - Detect bottlenecks.
 - Share work load between team members.
 - Current problems.
- Report achievements (Past).
 - Evaluate each member's contributions to the project



Some tools provided by github for Project management



GitHub

Sprint: Projects > Mapbox-Development

Filter: + Add issue · × Delete issues · More

+ Add column

Backlog	To Do	In progress	Done
<ul style="list-style-type: none">Create Docker image for project (magibox-admin-server#1) opened by jfayet · enhancement priority lowThrow errors for missing dependencies (magibox-aggregates#10) opened by jfayet · enhancement hotfixCurling results in empty response, server crashes (magibox-admin-server#11) opened by jfayet · hotfix priority critServe aggregations by admin level (magibox-core-api#1) opened by miketabBase API in Swagger Spec Has Extra / (magibox-openapi#1) opened by shertoobUse arguments to overwrite default configurations (magibox-aggregate-mobility#2) opened by carlosdias · enhancementReturn hasProbData in API responses for schools (magibox-core-api#3) opened by ayman#17expired token's warning gets overridden by middleware library and return html instead of json (magibox-openapi#4) opened by miketab	<ul style="list-style-type: none">Colombia shapefile (mpis.json) should be fetched from api rather than be required in react app. (magibox-map#1) opened by miketabFix "No_Units" issue in react-app/src/public/Rib.js (magibox-map#2) opened by jfayetComment and refactor code in actions-select-country.js (magibox-map#3) opened by miketabAdd disclaimer popup on load (magibox-map#4) opened by miketabResarfFigure logic code to work in production (magibox-map#5) opened by miketab	<ul style="list-style-type: none">Add contributing guidelines (placeholder) (closes #20) (magibox-project#1) opened by jfayet · enhancementDetermine where to begin writing tests and refactoring (magibox-map#6) opened by miketabMap relationships between apps, maps, and azure resources. (magibox#1) opened by miketab · in progress internal organization priority highMobility: Map should display data for mobility being displayed (magibox-map#2) opened by miketab	<ul style="list-style-type: none"> Rename test suite (run-tests.sh), add comments. (magibox-project#1) opened by jfayet · enhancementRefactor WebglLayer.js (magibox-map#1) opened by jfayet · bugCreate production/staging/dev deployment slots for Mobility Mapping (magibox#1) opened by miketab · enhancement internal organization priority highSet up automatic CI testing for new pull requests (magibox#1) opened by jfayet · enhancement priority highAdd loading spinner to mobility map (magibox-map#17) opened by miketabAdd spinner mobility (magibox-map#1) opened by carlosdias · enhancementAdd CodeClimate maintainability + test coverage checks to CI (magibox-openapi#1) opened by jfayet · enhancementAdd TravisCI and CodeClimate tests (magibox-openapi#2) opened by jfayet · enhancement

Automated on To do · Manage · Automated on In progress · Manage · Automated on Done · Manage



Project tab

- It is possible to create KanBan projects
- Automate workflow managing also issues and pull requests
- It is possible to create different dashboards (docs, backend,...)

Code Issues Pull requests Actions Projects ...

Projects ▾

Q is:open

0 Open 0 Closed Sort ▾

There aren't any projects yet

Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Project board name

Description (optional)

Project template

Save yourself time with a pre-configured project board template

Template: None ▾

Create project

Automation options



Issues



They are like post-it in a **Kanban** project

👉 Each problem, task or even question related with the Project can have a issue.

They use **Markdown**.

Recommended to write only the necessary to understand the issue. You can add pictures or links

It is the main part of a Project and can be used to understand the state of the project.

An Issue can be related with other entities in a github repo.

People

Labels

Milestones



We only assess information that appears in the github repo



GitHub



Investigate to what extent HTTPS should be mandatory #1091

[New issue](#)

Issue status ←

Open

RubenVerborgh opened this issue on 9 Dec 2021 · 2 comments

Problem explanation

Comments



RubenVerborgh commented
on 9 Dec 2021

Member



...

In some previous testing, I have come across preliminary evidence that some Solid-related functionality only works over HTTPS. In particular, when running Mashlib as the on-server UI, authentication seems to break because the server is not running over HTTPS.

Whereas this is actually a question for the bigger Solid ecosystem, we can test some assumptions on CSS and turn them into recommendations.

If the answer is that some functionality only works over HTTPS, then we might want to make CSS start over HTTPS out of the box (e.g., by auto-generating `localhost` certificates etc.).



RubenVerborgh



task



task



label



label



label



label



label



label



label



label



label



label



label



label

Assignees

RubenVerborgh

→ Assigned to

Labels

task

Projects

None yet

Milestone

No milestone

→ Milestones

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

→ Issue log

Corresponding
pull request

● GitHub

Labels

- Labels can be used to catalog issues.
- Generic labels from GitHub
- You can create your own labels (backend, frontend, bbdd, hierarchies)
- You can personalize color and even use emojis .



Code Issues 105 Pull requests 13 Discussions Actions Projects 1

Labels Milestones Search all labels

32 labels

Label name	Short description	Number of labelled issues
worker threads	Making the server multithreaded	5
bug	Something isn't working	17
dependencies	Pull requests that update a dependency file	
developer experience		5
difficulty:high		2
difficulty:low		13
difficulty:medium		9

Short description

Number of labelled issues

Common hierarchies priorities
use to give

You can use arquisoft FAQ for questions

<https://github.com/Arquisoft/faq/issues>

- Share questions about the course
- It is allowed to add any issue that describes some question in either English or Spanish,
 - Anyone can contribute answering the question or adding any comment.
 - Contributors must follow a code of conduct that respects the ethical considerations from a University of Oviedo Course.
- The teachers can remove any issue or contribution that they consider inappropriate for the course.

Additional links

- Introduction to git [Git](#).
- [Quick reference](#) from Pablo Gonzalez
- Short [introducción a git](#) (Hugo)
- Git: the simple guide
 - <https://rogerdudler.github.io/git-guide/index.html>
- [learngitbranching.js.org](#)

Additional Doc

- [Miguel Angel Durán](#) channel and his initial guide [Aprendiendo Git](#).
- Youtube
- [Git explanation in 15 minutes](#)
 - [Tutorial](#) for beginners
 - [GitHub vs GitLab](#)

Lab assignment of this year



Description:

<https://arquisoft.github.io/course2526.html#labs>

1st deliverable

- Documentation, version 0.1
- 1st Proof of concept
- App deployed
 - WebApp invoking GameY
 - WebApp accessing Database

