# UNIT 2: Control Version Systems. GIT

# Lab 3.GitHub.

## 0. INTRODUCTION

**GitHub** is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

There are other platforms, such as **Bitbucket or GitLab.** They each have their share of fans, though GitHub is by far the most-used of the three. Of the three, only GitLab is open source, though all three support open source projects. GitHub offers free public repositories; Bitbucket also offers free private repositories; GitLab offers a Community Edition which is entirely free.

We will learn about Github.

## 1. Signing up for a new GitHub account.

GitHub offers free accounts for users and organizations working on public and open source projects, as well as paid accounts that offer unlimited private repositories and optional user management and security features.

If you haven’t done before, visit https://github.com/ and sign up for an account on the free plan.

## 2. Starting with GitHub

To learn basics about GitHub follow the guide “Hello World” in https://guides.github.com. **When you finish you must send me an email with the link to your “Hello World” repository**

You will learn about:

- Create a Repository
- Create a Branch
- Make and commit changes
- Open a pull request
- Merge your Pull Request

In this guide, you have been working directly with GitHub, this is a way, but the most common way is to work in local as we did in Laboratory II. GitHub will be our server, so to work with GitHub repositories we will clone them and after that, we will upload the changes to GitHub.

## 3. Issues and Milestones

Issues are a great way to keep track of tasks, enhancements, and bugs for your projects. Milestones are groups of issues that correspond to a project, feature, or time period.
A typical issue on GitHub has:

- A **title** and **description** describe what the issue is all about.
- Color-coded **labels** help you categorize and filter your issues (just like labels in email).
- A **milestone** acts like a container for issues. This is useful for associating issues with specific features or project phases (e.g. *Weekly Sprint 9/5-9/16* or *Shipping 1.0*).
- One **assignee** is responsible for working on the issue at any given time.
- **Comments** allow anyone with access to the repository to provide feedback.

After receiving your “Hello world” repository link I will add you as a collaborator to the “DAW_2023-24” repository. We all are going to collaborate to create an index of the module. For achieve this purpose I have created a set of issues, you must check them an resolve the task that has been assigned to you. You have to

- Go to DAW_2023-24 repository
- Find out your task and add a comment to the issue you are fixing
- Clone the repository
- *In order to carry out your task, you have to c*reate a new branch called “yourname”and move to the “your_name” branch (on your local machine)
- Complete your task in “yourname” branch
- Upload your changes, that is you must add your new branch, make sure you don’t overwrite master branch.
- Open a pull request, asking the owner to check and add the changes to the master branch just in case she agrees about your solution
- Close the issue
- Add a comment to some classmates’ pull requests
- Add a new issue and assign it to the owner of the repository

As we all are collaborators, we can clone the repository, work in local and upload the changes(R/W). The section “Forking project” shows another point of view.

## 3. Project Boards

Project Boards on GitHub help you organize and prioritize your work. You can create project boards for specific feature work, comprehensive roadmaps, or even release checklists. With project boards, you have the flexibility to create customized workflows that suit your needs.

GitHub Project Boards are base on the kanban/scrum methodologies. These methodologies allow for large and complex tasks to be broken down and completed efficiently. They place a high value on continual improvement, optimization of the work and the process and focus on a highly visible work flow that keeps all team members in the loop on Work In Progress and What’s To Come/To Do.

## 4. Forking project

If you want to start a project based on another project or contribute to someone else's project you will need fork the former project.

Creating a "fork" is producing a personal copy of someone else's project. You can submit *Pull Requests* to help make other people's projects better by offering your changes up to the original project. Forking is at the core of social coding at GitHub. To learn about "fork" complete the "Forking project" guide using the "Fork_DAW_2023-24" instead of the Spoon-Knife project.

## 5. Let me know you have finished

Go to the virtual classroom and let me know that you have completed the Lab3, task "I have completed Lab3"