What are the difference between Git and GitHub?
Git is a specific open-source version control system which could be installed to a local machine. It is a distributed version control system, which means that the entire codebase and history is available on every developer's computer, which allows for easy branching and merging.
GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. It allows to clone, pull and make changes on codes written. It also saves the history or a particular code: when it was modified, how it was modified and who modified it. To understand exactly what GitHub is, one needs to know two connected principles: Version control and Git
GitHub is a platform from where we could clone, pull and make changes on codes written. It allows to save the history or particular code, when it was modified, how it was modified and who modified it.

What is Git Workflow?
A Git Workflow is a recipe or recommendation for how to use Git to accomplish work in a consistent and productive manner. Git workflows encourage users to leverage Git effectively and consistently. Git offers a lot of flexibility in how users manage changes. Given Git's focus on flexibility, there is no standardized process on how to interact with Git. When working with a team on a Git managed project, it's important to make sure the team is all in agreement on how the flow of changes will be applied. To ensure the team is on the same page, an agreed upon Git workflow should be developed or selected.

How many types of version control systems are there?
There are two types of Version Control Systems: Centralized and distributed.

Centralized version control
With centralized version control systems, you have a single "central" copy of your project on a server and commit your changes to this central copy. You pull the files that you need, but you never have a full copy of your project locally. Some of the most common version control systems are centralized, including Subversion (SVN) and Perforce.

Distributed version control
With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project's files. Instead, you clone a copy of a repository locally so that you have the full history of the project. Two common distributed version control systems are Git and Mercurial.
While you don't have to have a central repository for your files, you may want one "central" place to keep your code so that you can share and collaborate on your project with others. That's where Bitbucket comes in. Keep a copy of your code in a repository on Bitbucket so that you and your teammates can use Git or Mercurial locally and to push and pull code.