In this module, you will learn about three popular tools used in data science: GitHub, Jupyter Notebooks, and RStudio IDE. You will become familiar with the features of each tool, and what makes these tools so popular among data scientists today.

**Objetivos de aprendizagem**

* Describe the fundamentals of Jupyter Notebooks and the JupyterLab platform and why they are popular for data science projects.
* Cite some examples of how Jupyter is used in data science for recording experiments and projects.
* Use Jupyter notebooks to experiment with code cells, create data visualization slides, and reporting results using Markdown.
* Create a Jupyter notebook with Python and use it to run code, change kernels, and save your work.
* Define the R programming language and discuss its application in data science.
* Use RStudio packages to experiment with plotting.
* Explain versioning, branching and why they’re vital to the software development process.
* Create, edit. and upload new files in GitHub.
* Use GitHub to create a branch, commit changes, and initiate a pull request.

# **Overview of Git/GitHub**

In this video, you’ll get an overview of Git and GitHub, which are popular environments

among developers and data scientists for performing version control of source code files and projects

and collaborating with others.

You can’t talk about Git and GitHub without a basic understanding of what version control is.

Reproduza o vídeo começando em ::30 e siga a transcrição0:30

A version control system allows you to keep track of changes to your documents.

This makes it easy for you to recover older versions of your document if you make a mistake,

and it makes collaboration with others much easier.

Here is an example to illustrate how version control works.

Let’s say you’ve got a shopping list and you want your roommates to confirm the things

you need and add additional items.

Without version control, you’ve got a big mess to clean up before you can go shopping.

With version control, you know EXACTLY what you need after everyone has contributed their ideas.

Reproduza o vídeo começando em :1:9 e siga a transcrição1:09

Git is free and open source software distributed under the GNU General Public License.

Git is a distributed version control system, which means that users anywhere in the world

can have a copy of your project on their own computer; when they’ve made changes, they

can sync their version to a remote server to share it with you.

Git isn’t the only version control system out there, but the distributed aspect is one

of the main reasons it’s become one of the most common version control systems available.

Version control systems are widely used for things involving code, but you can also version

control images, documents, and any number of file types.

You can use Git without a web interface by using your command line interface, but

GitHub is one of the most popular web-hosted services for Git repositories.

Others include GitLab, BitBucket, and Beanstalk.

There are a few basic terms that you will need to know before you can get started.

The SSH protocol is a method for secure remote login from one computer to another.

A repository contains your project folders that are set up for version control.

A fork is a copy of a repository.

A pull request is the way you request that someone reviews and approves your changes

before they become final.

A working directory contains the files and subdirectories on your computer that are associated

with a Git repository.

There are a few basic Git commands that you will always use.

When starting out with a new repository, you only need create it once: either locally,

and then push to GitHub, or by cloning an existing repository by using the command "git init".

Reproduza o vídeo começando em :3:3 e siga a transcrição3:03

"git add" moves changes from the working directory to the staging area.

"git status" allows you to see the state of your working directory and the staged snapshot

of your changes.

"git commit" takes your staged snapshot of changes and commits them to the project.

"git reset" undoes changes that you’ve made to the files in your working directory.

"git log" enables you to browse previous changes to a project.

"git branch" lets you create an isolated environment within your repository to make changes.

"git checkout" lets you see and change existing branches.

"git merge" lets you put everything back together again.

To learn how to use Git effectively and begin collaborating with data scientists around

the world, you will need to learn the essential commands.

Luckily for us, GitHub has amazing resources available to help you get started.

Go to try.github.io to download the cheat sheets and run through the tutorials.

In the following modules, we'll give you a crash course on setting up your local environment

and getting started on a project.

In this video, you’ll get an overview of Git and GitHub, which are popular environments among de: adicionado à seleção. Pressione [CTRL + S] para salvar como anotação

# **GitHub - Getting Started**

In the previous video, you learned about Git  and GitHub. Before you continue with this video,

register for a GitHub account and log in.  Let’s start by creating a new repository.

Click + then click New Repository. To  create a new repository, you need to provide

these details: give your new repository a name;  optionally, add a description of your repository;

choose the repository visibility - whether you  want it to be public or private; and choose the option

to Initialize this repository with readme file. Then click Create Repository.

Reproduza o vídeo começando em ::54 e siga a transcrição0:54

You will now be redirected to the  repository you have created.

The root folder of your repository is listed  by default and it has just one file ReadMe.md.

Reproduza o vídeo começando em :1:8 e siga a transcrição1:08

Now, it’s time to edit the readme. You can do this  in your browser. Just click the pencil to open the

online editor and you can change the text of the  readme. To save your changes to the repository,

you must commit them. After you have made your  changes, scroll down to the Commit changes

section. Add a commit message and optionally add  a description, then click Commit changes. The

"commit changes" is used to save your changes  to the repository. Go back to the home screen by

clicking the repository name link. Note that the  readme file is updated and verify your changes.

Reproduza o vídeo começando em :1:49 e siga a transcrição1:49

Let’s learn how to create a new file using the  built-in web editor provided by GitHub which

runs in the browser. Click Add File, then  click Create New File to create the new file.

Reproduza o vídeo começando em :2:4 e siga a transcrição2:04

To create a python file called firstpython.py.

First, provide the file name. Next, add a comment  that describes your code, then add the code.

Reproduza o vídeo começando em :2:17 e siga a transcrição2:17

Once finished, commit the change to the  repository. You can see that your file is

now added to the repository and the repository  listing shows when the file was added or changed.

When you need to change the  file, you can edit it again.

Click the file name, and then click the pencil  icon, make your edits and commit the changes.

Reproduza o vídeo começando em :2:40 e siga a transcrição2:40

You can also upload a file from your  local system into the repository.

From the home screen of the repository, click  Add File and choose the Upload files option.

Reproduza o vídeo começando em :2:53 e siga a transcrição2:53

Click Choose Your Files and select the files  you want to upload from your local system.

Reproduza o vídeo começando em :2:59 e siga a transcrição2:59

The file upload process may take a short time,

depending on what you are uploading.  Once the files finish uploading,

click Commit Changes. The repository now reflects  the files that were uploaded. In this video,

you learned how to create a repository, edit  files, and commit changes using the web interface.