Python:
Una herramienta multipropósito

CONTROL DE VERSIONES

¿Qué es control de versiones?

Control de versiones es el método de gestionar los cambios y versiones de un archivo o distintos archivos.

Por ejemplo, una manera muy básica de controlar versiones es simplemente crear varios:

Version1.py, version_1.1.py, version1_estasi.py, version_1_final.py, version1_finalfinal.py

¿Qué es control de versiones?

Como podrán darse cuenta esto no es lo más eficiente, especialmente cuando estamos trabajando en un proyecto que requiere cambios constantes a muchos archivos y quizás por varias personas a la vez.

Sistema de Control de Versiones

Un sistema de control de versiones o VCS es una herramienta que permite de manera más eficiente gestionar la vida de un proyecto y sus archivos.

Mediante estos sistemas comúnmente operamos con el concepto de repositorio.

Un repositorio puede contener uno o más proyectos (aunque comúnmente gestionamos uno solo).

Mediante un VCS podemos manejar varios repositorios!

GIT

Git es un sistema de versiones distribuido que permite que operemos con diferentes archivos, registremos cambios individuales, mantengamos un log de cambios, y podamos regresar a versiones anteriores con mucha facilidad.

GIT: VCS Distribuido

Git es distribuido debido a que cuando trabajamos en un repositorio, cada participante del proyecto opera en una copia local independiente, facilitando trabajar de manera separada.

Cuando un cambio se integra al repositorio principal, **git** solo rastrea la historia de cambios en sus archivos, haciendo el proceso más eficiente.

GIT: Comandos

git clone

Nos permite crear un clon local del repositorio que queremos. Podremos operar en este independientemente de las demás personas.

git add

Cuando hayamos cambiado o agregado un archivo al repositorio, add nos permite registrarlo como algo que queremos sea incluido en siguiente “commit” o “compromiso”.

git commit

Este comando nos permite registrar los cambios oficialmente en el log de cambios del repositorio.

git push

En el contexto distribuido, este comando nos permite “publicar” los cambios actualmente presentes en nuestro log e incorporarlos en el log del remoto de interés, este puede ser el repositorio principal u otro.

GIT: Remoto

Un remoto es simplemente un vínculo a otro repositorio

Remotos

Un remoto es simplemente un vínculo a otro repositorio de interés.

Este puede ser el repositorio original del cual creamos un clon, o el repositorio de otro miembro del equipo, un repositorio remoto de otro servidor, etc.

git remote add nombre urldelrepositorio

Github

Github y git son dos conceptos diferentes.

Git es un sistema de control de versiones

Github es una plataforma en la nube que permite gestionar repositorios de git a manera de servicio, y además ofrece opciones comunitarias para colaboración, rastreo de funcionalidades, publicación y automatización de chequeos y mucho más.

GitHub

Github

La mayor parte de repositorios git de código abierto existen en Github.

Sin embargo, existen alternativas como Gitlab o Bitbucket que también permiten operar con git.

Fork

Fork es un concepto único de Github, esencialmente crea un nuevo proyecto basado en otro y permite colaboración independiente o la separación completa del trabajo.

Creando un fork podemos crear un proyecto completamente diferente que se base en otro.

Pull request

El concepto que permite tener un proceso de aprobación y revisión a los cambios entre las ramas de un repositorio o entre forks.

Mediante esto podemos colaborar y tener conversaciones cuando existan cambios al código de fuente de un proyecto.

Aprenderemos como automatizar chequeos que se realicen cuando creemos pull requests en la siguiente clase.

Aprender más de Git y Github

Git vs GitHub – ¿Qué es el Control de Versiones y Cómo Funciona? (freecodecamp.org)

Ejercicio

- Creando un repositorio en Github
- Creando un paquete en Python
- Agregando tipos al paquete
- Publicando el paquete al repositorio de Github

Conclusión

Preguntas?

Siguiente clase

- Plataformas de desarrollo para Python
- Testing en Python
- Chequeos
- Integración Continua y CircleCI