

INVESTIGACIÓN #1- PYTHON

¿Qué es Git?

Git es un Sistema de Control de Versiones Distribuido (DVCS) utilizado para guardar diferentes versiones de un archivo (o conjunto de archivos) para que cualquier versión sea recuperable cuando lo desee. (Castellanos, 2021)

Distribuido

“El término "distribuido" significa que cuando le instruyes a Git que comparta el directorio de un proyecto, Git no sólo comparte la última versión del archivo. En cambio, distribuye cada versión que ha registrado para ese proyecto. {...} Bueno, entonces "distribuido" significa distribuir todas – no solo algunas seleccionadas – versiones de los archivos del proyecto que Git haya registrado” (Castellanos, 2021)

¿Qué es un Sistema de Control de Versiones?

Un Sistema de Control de Versiones (VCS) se refiere *al método* utilizado para guardar las versiones de un archivo para referencia futura.

De manera intuitiva muchas personas ya utilizan control de versiones en sus proyectos al renombrar las distintas versiones de un mismo archivo de varias formas como:

- `blogScript.js`,
- `blogScript_v2.js`,
- `blogScript_v3.js`,
- `blogScript_final.js`,

Pero esta forma de abordarlo es propenso a errores y inefectivo para proyectos grupales. Además, con esta forma de abordarlo, rastrear qué cambió, quién lo cambió y por qué se cambió, es un esfuerzo tedioso. Esto resalta la importancia de un sistema de control de versiones confiable y colaborativo como Git.

Estados de los archivos en Git

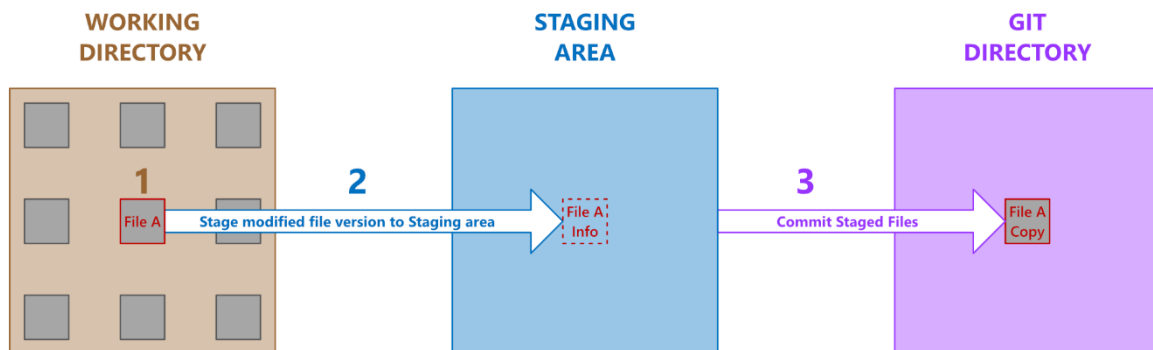
- Estado modificado: Un archivo en el estado modificado es un archivo revisado – pero no acometido (sin registrar). En otras palabras, archivos en el estado modificado son archivos que has modificado pero no le has instruido explícitamente a Git que controle.
- Estado preparado: Archivos en la etapa preparado son archivos modificados que han sido seleccionados – en su estado (versión) actual – y están siendo preparados para ser guardados (acometidos) al repositorio **.git** durante la próxima instantánea de

confirmación. Una vez que el archivo está preparado implica que has explícitamente autorizado a Git que controle la versión de ese archivo.

- Estado confirmado: Archivos en el estado confirmado son archivos que se guardaron en el repositorio **.git** exitosamente. Por lo tanto un archivo confirmado es un archivo en el cual has registrado su versión preparada en el directorio (carpeta) Git.

Ubicación de archivos

- Directorio de trabajo: El directorio de trabajo es una carpeta local para los archivos de un proyecto. Esto significa que cualquier carpeta creada en cualquier lugar en un sistema es un directorio de trabajo.
- Zona de preparación: La zona de preparación – técnicamente llamado “index” en lenguaje Git – es un archivo normalmente ubicado en el directorio **.git**, que guarda información sobre archivos próximos a ser acometidos en el directorio **.git**.
- Directorio Git: El directorio **.git** es la carpeta (también llamada "repositorio") que Git crea dentro del directorio de trabajo que le has instruido para realizar un seguimiento.



GIT HUB

GitHub es una plataforma basada en la web donde los usuarios pueden alojar repositorios Git. Facilita compartir y colaborar fácilmente en proyectos con cualquier persona en cualquier momento. GitHub también fomenta una participación más amplia en proyectos Código Abierto al proporcionar una manera segura de editar archivos en repositorios de otros usuarios. (Castellanos, 2021)

MARK DOWN (Peters, 2014)

Markdown es un lenguaje de marcado ligero creado por John Gruber y Aaron Swart que permite la escritura rápida de textos. Literalmente, hace posible que no se tengan que despegar los dedos del teclado para que podamos concentrarnos en lo esencial: la escritura.

Esta característica lo convierte en un gran aliado a la hora de documentar nuestros proyectos Python (o escritos en otros lenguajes); un trabajo que siempre es necesario abordar pero que suele resultar, a veces, monótono y poco gratificante.

Para mejorar notablemente la productividad Markdown se vale de una serie de caracteres y sentencias que se introducen sutilmente entre el texto con una doble utilidad:

- Hacer que los textos sean más legibles y publicables.
- Aplicar un formato rudimentario a los textos que es aprovechado cuando son convertidos a otros formatos conocidos (.pdf, .odt, .rtf, .doc y .html). Por ejemplo, si en un texto Markdown hemos marcado una palabra en cursiva al convertirlo al formato .odt se le aplicará el formato cursiva a la misma palabra o si se convierte a .html se incluirán las etiquetas de cursiva de ese lenguaje para conseguir el mismo resultado.

Los ficheros de Markdown son de texto plano, un formato muy apropiado para mantener la documentación que no suele acompañar al propio código de los programas. Además, es utilizado en software de publicación web como wikis, blogs, etc. y hay editores de texto Markdown (y plugins) que nos facilitarán la tarea de escribir:

- Linux: gedit, ReText, UberWriter
- Windows: HippoEDIT, MarkdownPad

• Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	**bold text**
Italic	<i>*italicized text*</i>
Blockquote	> blockquote
Ordered List	1. First item 2. Second item 3. Third item
Unordered List	- First item - Second item - Third item
Code	`code`
Horizontal Rule	---
Link	[title](https://www.example.com)
Image	![alt text](image.jpg)

Extended Syntax

.



Element	Markdown Syntax
Table	<pre> Syntax Description ----- ----- Header Title Paragraph Text </pre>
Fenced Code Block	<pre>``` { "firstName": "John", "lastName": "Smith", "age": 25 } ```</pre>
Footnote	<p>Here's a sentence with a footnote. [¹]</p> <p>[¹]: This is the footnote.</p>
Heading ID	<pre>### My Great Heading {#custom-id}</pre>
Definition List	<pre>term : definition</pre>
Strikethrough	<pre>~~The world is flat.~~</pre>
Task List	<pre>- [x] Write the press release - [] Update the website - [] Contact the media</pre>
Emoji (see also Copying and Pasting Emoji)	<p>That is so funny! :joy:</p>
Highlight	<pre>I need to highlight these ==very important words==.</pre>
Subscript	<pre>H~2~0</pre>
Superscript	<pre>X^2^</pre>