



Fundamentos de Programación 101

By Ernie

Ernesto José Canales Guillén

Círculos de estudio UCA

Ciclo Virtual 01/2021



Preparando nuestro entorno de Desarrollo

Git, GitHub, Compiler and VS Code



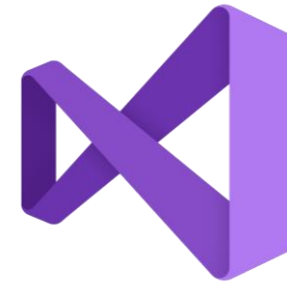
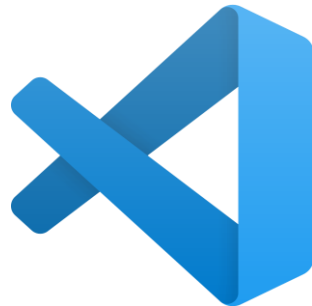
IDE vs Editor de texto

- Editores de texto:
 - Los editores de texto son herramientas más minimalistas, lo que quiere decir que son mucho más simples y compactas. Proporciona un entorno de desarrollo simple y que por lo general parecería que estamos utilizando una especie de Notepad avanzado. Los editores de texto tienen la peculiaridad de trabajar con archivos de texto y carpetas, es decir, abres una carpeta y trabajas con todo lo que hay dentro.
- Integrated Development Environment (IDE):
 - Son herramientas mucho más poderosas. Los IDEs tiene como principal diferencia, que no trabajan con archivos y carpetas, en su lugar, emplean el concepto de Proyectos. el IDE puede crear archivos adicionales al código para optimizar la experiencia del usuario. En estos archivos puede tener configuraciones de ejecución, deploy, tipo de proyecto, etc...

IDEs e Editores de texto en C++

Editor de texto

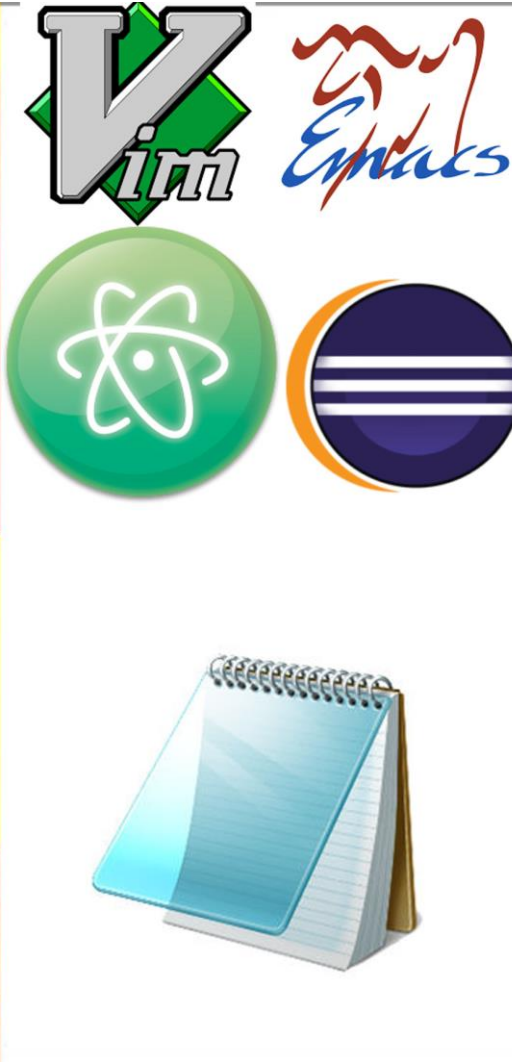
- Notepad
- Notepad ++
- Visual Studio Code



IDE

- Code::Blocks
- Visual Studio Community







Git y GitHub

- GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git.
- Git es un sistema de control de versiones. Un sistema de control de versiones nos va a servir para trabajar en equipo de una manera mucho más simple y optima cuando estamos desarrollando software.



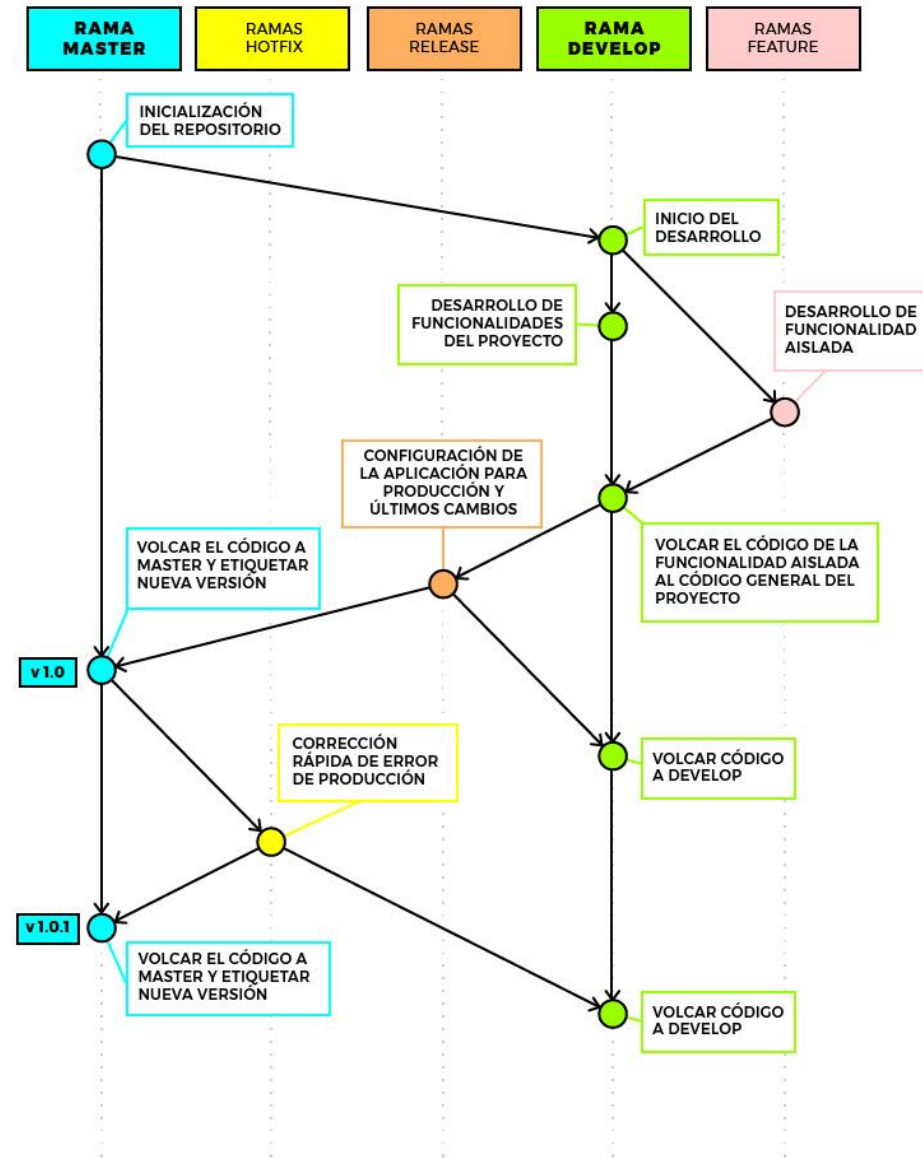
YouTube

Curso: Git y GitHub 1



YouTube

Curso: Git y GitHub 2





Git Cheat Sheet



Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new_branch

```
$ git branch new_branch
```

Delete the branch called my_branch

```
$ git branch -d my_branch
```

Merge branch_a into branch_b

```
$ git checkout branch_b
```

```
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$ git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

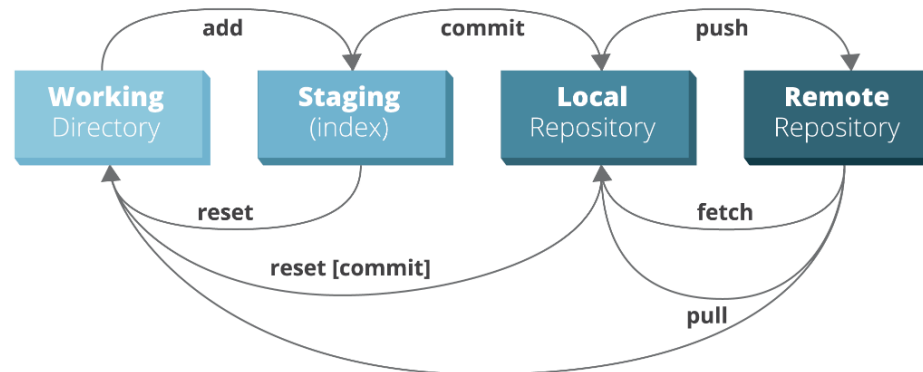
```
$ git push
```

Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit <https://training.github.com/> for official GitHub training.



Posted in r/git by u/spite77





Instalando Software...

S14_Installing_Compiler.pdf

S14_Installing_VS_Code.pdf

S14_Installing_Git_GitHub.pdf



Bibliografía

- L. J. Aguilar, Programación en C++. Algoritmos, estructuras de datos y objetos, Aravaca (Madrid): McGRAW-HILL, 2006.
- L. J. Aguilar, FUNDAMENTOS DE PROGRAMACIÓN: Algoritmos, estructura de datos y objetos, 28023 Aravaca (Madrid): McGRAW-HILL, 2008.
- D. Malik, C++ Programming: From Problem Analysis to Program Design, Boston, MA: Cengage Learning, 2003.
- Scott Chacon, Por Git: Everything you need to know about Git, 2nd edition, Apress, 2014.
- Reddit