

# **GIT, GITHUB Y MARKDOWN**

**ADOLFO SANZ DE DIEGO**

**OCTUBRE 2015**

**1 ACERCA DE**

# 1.1 AUTOR

- **Adolfo Sanz De Diego**
  - Blog: [asanzdiego.blogspot.com.es](http://asanzdiego.blogspot.com.es)
  - Correo: [asanzdiego@gmail.com](mailto:asanzdiego@gmail.com)
  - GitHub: [github.com/asanzdiego](https://github.com/asanzdiego)
  - Twitter: [twitter.com/asanzdiego](https://twitter.com/asanzdiego)
  - LinkedIn: [in/asanzdiego](https://in/asanzdiego)
  - SlideShare: [slideshare.net/asanzdiego](https://slideshare.net/asanzdiego)

## 1.2 LICENCIA

- Este obra está bajo una licencia:
  - Creative Commons Reconocimiento-CompartirIgual 3.0

## 1.3 FUENTE

- Las slides y sus fuentes las podéis encontrar en:
  - <https://github.com/asanzdiego/curso-git-github-markdown-2015>

# 2 INTRODUCCIÓN

## 2.1 OBJETIVOS

1. Conocer las **características de Git** y ser capaz de instalar y configurarlo.
2. Conocer y ser capaz de usar los **comandos de Git**.
3. Conocer las **características de GitHub** y ser capaz de crear una cuenta y configurarla.
4. Ser capaz de **crear y clonar repositorios** en GitHub.
5. Conocer y ser capaz de **usar las principales características de GitHub**.
6. Conocer la **sintaxis del lenguaje Markdown**.

## 2.2 INDICE

- **Bloque 1**
  - Introducción a Git
  - Introducción a GitHub
  - Introducción Markdown
- **Bloque 2**
  - Uso básico de Git
  - Uso básico de GitHub
  - Sintaxis básica de Markdown
- **Bloque 3**
  - Uso avanzado de Git
  - Uso avanzado de GitHub
  - Sintaxis avanzada de Markdown



## 2.3 ENLACES DE INTERÉS

- Aprender GIT... y de camino GitHub:
  - <https://github.com/oslugr/curso-git>
- Página oficial de Git:
  - <https://git-scm.com/>
- Pro GIT:
  - <https://git-scm.com/book/es/v2>
- Minitutorial de GIT:
  - <https://try.github.io/>
- Página oficial de GitHub:
  - <https://github.com/>
- Chuleta de la sintaxis de Markdown:
  - <http://warpedvisions.org/projects/markdown-cheat-sheet>

# 3 INTRODUCCIÓN A GIT

# 3.1 SISTEMA CONTROL DE VERSIONES

*"Sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante."*

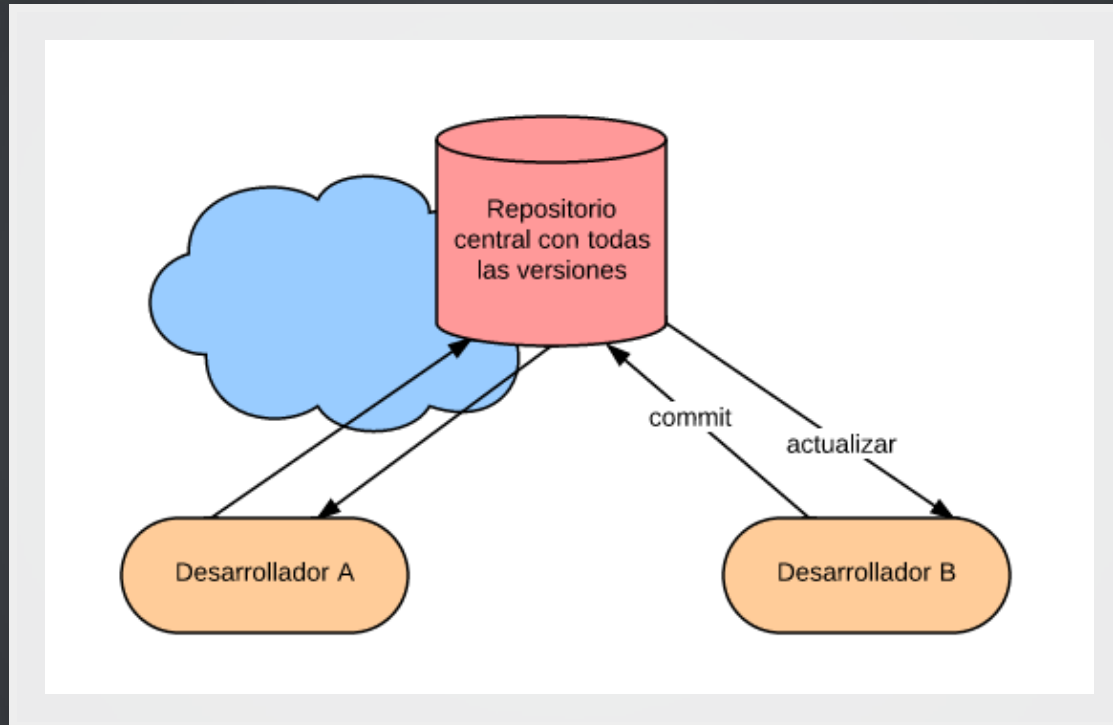
<https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>

## 3.2 VCS LOCALES

- **Lo más simple:** hacer copias de directorios.
- Aparecieron **BD en local** que guardan el registro de los cambios realizados a los archivos.

## 3.3 VCS CENTRALIZADOS

- Un **servidor central** que guarda los cambios.



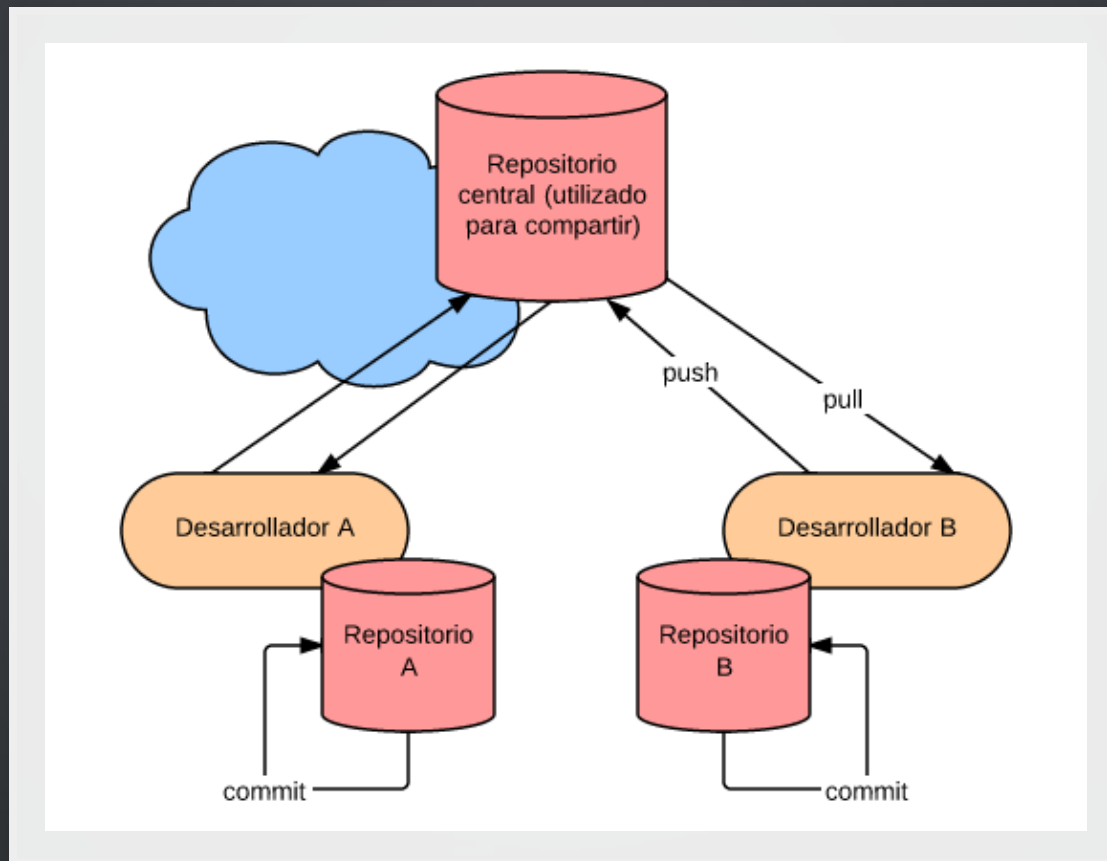
VCS Centralizado

## 3.4 PROS Y CONTRAS VCS CENTRALIZADOS

- **Pros:** más colaborativo que el local.
- **Contras:** dependes de un servidor central.

## 3.5 VCS DISTRIBUIDOS

- Cada cliente **no solo descarga la última copia, sino todo el repositorio.**



VCS Distribuido

## 3.6 VENTAJES VCS DISTRIBUIDOS

- Puedes seguir trabajando aunque el repositorio remoto esté caído.
  - **más autonomía**
- La información está más replicada.
  - **menos vulnerable**
- Permite pruebas en local y subir solo lo relevante.
  - **más limpieza**



## 3.7 CARACTERÍSTICAS DE GIT

- Creado por Linux Torvalds, líder del equipo del kernel Linux.
- Objetivos cuando se creó:
  - **Rápido**
  - **Sencillo**
  - **Multi rama**
  - **Distribuido**
  - **Grandes proyectos**

## 3.8 INSTALACIÓN

- Windows: <https://git-scm.com/download/win>
- Mac: <https://git-scm.com/download/mac>
- Linux: <https://git-scm.com/download/linux>

## 3.9 CONFIGURACIÓN INICIAL

```
git config --global user.name "Nombre que quieras mostrar"
```

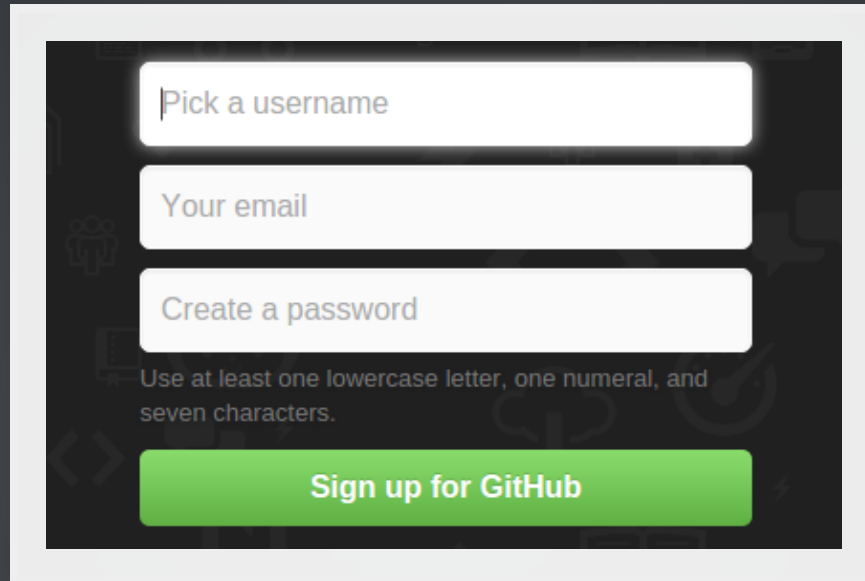
```
git config --global user.email "correo@electronico.es"
```

# 4 INTRODUCCIÓN A GITHUB

# 4.1 CARACTERÍSTICAS DE GITHUB

- **Plataforma de desarrollo colaborativo**, que utiliza Git.
- Los **repositorios son públicos**, salvo con cuenta de pago.
- Tiene facetas de **red social** (perfil público, seguidores, estrellas, etc.)
- Nos permite **gestionar organizaciones y equipos**.
- **Gestión de proyectos** (wiki, releases, incidencias, gráficos, etc.)
- **Servidor web**.

## 4.2 CREAR CUENTA

A screenshot of the GitHub sign-up form. The form is centered on a dark background and consists of three white input fields stacked vertically. The first field is labeled 'Pick a username', the second 'Your email', and the third 'Create a password'. Below the password field, there is a small text requirement: 'Use at least one lowercase letter, one numeral, and seven characters.' At the bottom of the form is a prominent green button with the text 'Sign up for GitHub' in white. The entire form is enclosed in a white rectangular border.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

Crear cuenta en GitHub

## 4.3 CREAR REPOSITORIO

[Pull requests](#) [Issues](#) [Gist](#)

**Adolfo Sanz De Diego**  
asanzdiego

Madrid

<http://about.me/asanzdiego>

Joined on 13 Apr 2012

30

Followers

10

Starred

74

Following

## Organizations



+ Contributions

Repositories

Public activity

New repository

New organization

profile

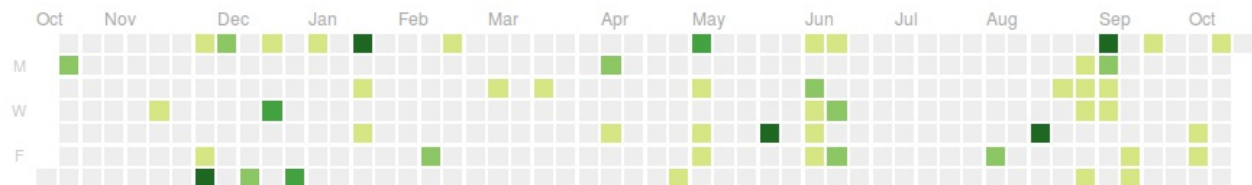
### Popular repositories

<a href="#">markdownslides</a>	47 ★
<a href="#">curso-interfaces-web-2014</a> Curso de Diseño de Interfaces Responsive We...	4 ★
<a href="#">casperjs-examples</a>	3 ★
<a href="#">deck.js-pandoc-slides</a>	3 ★
<a href="#">curso-javascript-avanzado-2015</a>	3 ★

### Repositories contributed to

<a href="#">HackathonLovers/HackathonLov...</a>	0 ★
<a href="#">LogLock/LogLock.github.io</a> .	0 ★
<a href="#">pelitweets/pelitweets-backend</a> Backend of pelitweets project	0 ★
<a href="#">HackathonLovers/snaphack-nod...</a>	0 ★
<a href="#">pronoide/pronoide.github.io</a> pronoide web site	0 ★

### Contributions



Summary of pull requests, issues opened, and commits. [Learn how we count contributions.](#)

Less More

Contributions in the last year

205 total

Oct 18, 2014 – Oct 18, 2015

Longest streak

5 days

September 5 – September 9

Current streak

0 days

Last contributed 8 days ago



# Crear un repositorio

# 5 INTRODUCCIÓN MARKDOWN

# 5.1 QUÉ ES MARKDOWN

*"Es un lenguaje de marcado ligero que trata de conseguir la máxima legibilidad y 'publicabilidad' usando texto plano."*

<https://es.wikipedia.org/wiki/Markdown>

## 5.2 CARACTERÍSTICAS PRINCIPALES

- Texto plano
- Sintaxis sencilla
- Legibilidad
- Publicabilidad
- Exportabiliad

## 5.3 EDITOR ONLINE

- <https://jbt.github.io/markdown-editor/>

## 5.4 CHULETA DE MARKDOWN:

- <http://warpedvisions.org/projects/markdown-cheat-sheet>

## 5.5 MARDOWNSLIDES

- <https://github.com/asanzdiego/markdownslides>

# 6 USO BÁSICO DE GIT

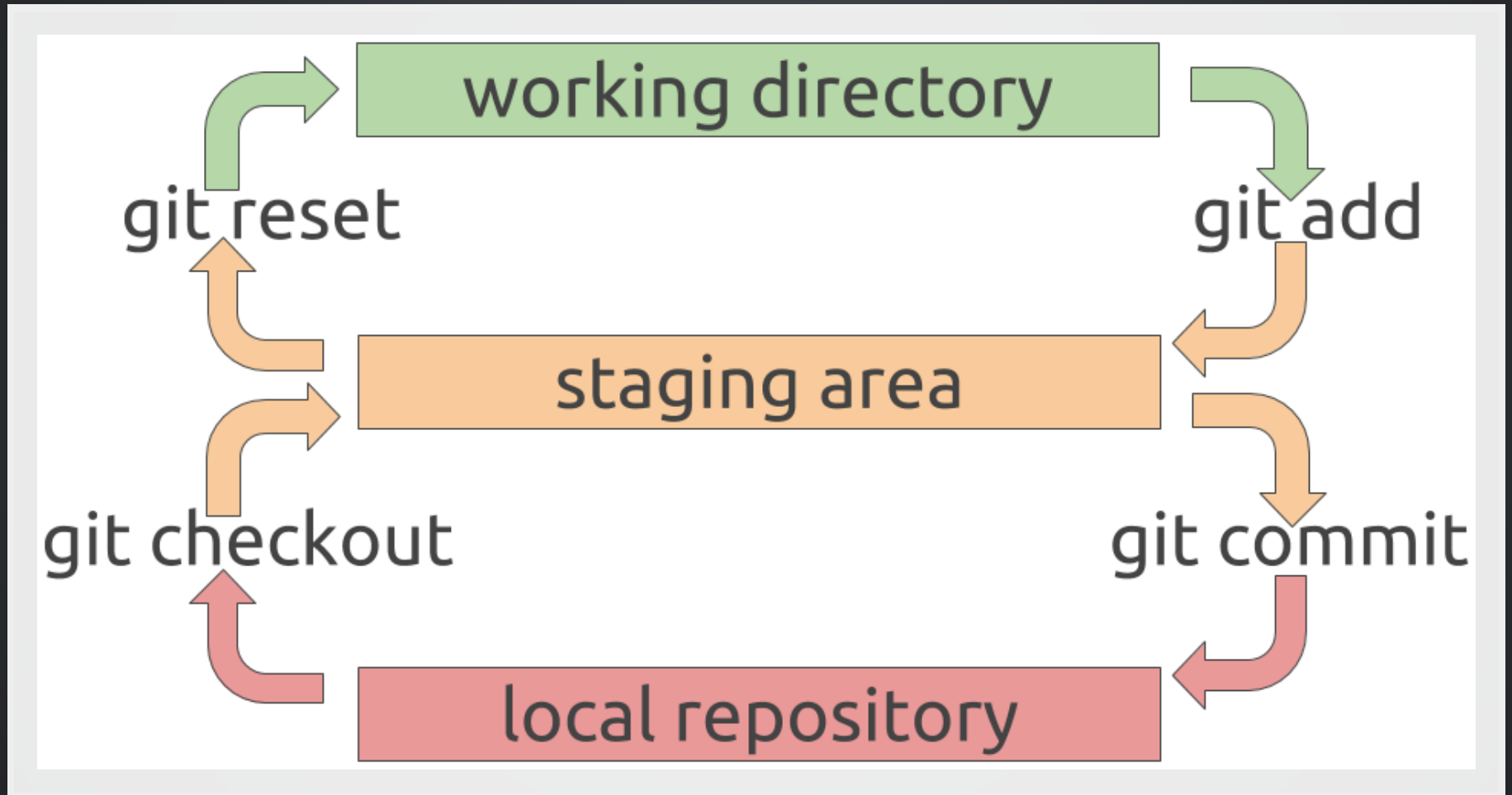


# 6.1 INIT

- **Inicializar** un repositorio.
- Crea el subdirectorio .git con archivos de git para gestionar repositorio.

```
git init
```

## 6.2 STAGING AREA



Staging Area

## 6.3 STATUS

- Para saber el **estado** de los archivos.

```
git status
```

## 6.4 DIFF

- Para ver las **diferencias** entre el área de staging y el área de trabajo.

```
git diff
```

## 6.5 ADD

- Para **añadir** los cambios de un fichero (o varios) al área de staging (desde el área de trabajo).

```
git add nombre-del-fichero
```

```
git add *.extension
```

## 6.6 RESET

- Para **resetear** los cambios de un fichero (o varios) al area de trabajo (desde el area de staging).

```
git reset nombre-del-fichero
```

## 6.7 COMMIT

- Para **subir** los cambios realizados al repositorio (desde el área de staging).

```
git commit -m "mensaje corto descriptivo con los cambios"
```

## 6.8 CHECKOUT

- Para **deshacer** los cambios de un fichero (o varios) al area de staging (desde el repositorio).

```
git checkout nombre-del-fichero
```



## 6.9 LOG

- Para ver el **listado de los commits** realizados en el repositorio.

```
git log
```

# 7 USO BÁSICO DE GITHUB

# 7.1 CONFIGURAR CLAVES (I)

- Nos permite gestionar repositorios **mediante SSH** sin tener que estar poniendo siempre nuestra contraseña.
- Se genera una **clave privada** que se guarde en nuestro ordenador y una **clave pública** que es la que tenemos que guardar en nuestra cuenta.

## 7.2 CONFIGURAR CLAVES (II)

- La podemos usar pues **solo con un ordenador.**
- Instrucciones:
  - <https://help.github.com/articles/generating-ssh-keys/>

## 7.3 CLONAR REPOSITARIOS

- Además de con git init, podemos empezar a trabajar con repositorios clonándolos:

```
git clone git@github.com:asanzdiego/markdownslides.git
```

## 7.4 USO SOCIAL

- Seguir a gente.
- Seguir proyectos (watch).
- Premiar proyectos (start).
- Forquear proyectos (fork).
- Pull-requests.
- Issues.
- Crear organizaciones.

# **8 SINTAXIS BÁSICA DE MARKDOWN**

# 8.1 ENCABEZADOS (I)

- <h1>, <h2>, <h3>

```
# Encabezado de primer nivel
```

```
## Encabezado de segundo nivel
```

```
### Encabezado de tercer nivel
```



## 8.2 ENCABEZADOS (II)

- Equivalente a lo anterior.

```
Encabezado de primer nivel
```

```
=====
```

```
Encabezado de segundo nivel
```

```
-----
```

```
### Encabezado de tercer nivel ###
```

## 8.3 LISTAS NO NUMERADAS

- No enumeradas:
  - se puede usar el asterísico
  - se puede usar el menos
  - se puede usar el más

```
* se puede usar el asterísico  
- se puede usar el menos  
+ se puede usar el más
```

# 8.4 LISTAS NO NUMERADAS

- Enumeradas:
  1. Primer elemento
  2. Segundo elemento
  3. Tercer elemento

```
1. Primer elemento  
1. Segundo elemento  
1. Tercer elemento
```

## 8.5 FORMATO (NEGRITA, CURSIVA, SUBRAYADO, TACHADO)

- Texto en cursiva con *un asterisco* o con *un guión bajo*.
- Texto en negrita con **dos asteriscos** o con **dos guiones bajos**.
- Texto ~~tachado~~.

```
- Texto en cursiva con *un asterisco* o con _un guión bajo_.  
- Texto en negrita con **dos asteriscos** o con __dos guiones bajos__.  
- Texto ~~tachado~~.
```

# 9 USO AVANZADO DE GIT

## **9.1 .GITIGNORE, MV Y RM**

## **9.2 BRANCH, CHECKOUT Y MERGE**

## **9.3 REBASE, STATUS Y DIFF**



# 10 USO AVANZADO DE GITHUB

# **10.1 GESTIONAR EQUIPOS**

## **10.2 PULL-REQUESTS**

## 10.3 GIT-PAGES

# **11 SINTAXIS AVANZADOS DE MARKDOWN**

# 11.1 IDS

## 11.2 TABLAS

## **11.3 BLOQUES (CITAS, CÓDIGO)**



## **11.4 ENLACES E IMÁGENES**