

Tabla de contenido

Introducción	1.1
Información sobre NodeJS	1.2
NodeJS	1.2.1
Información sobre Git	1.3
Git	1.3.1
Información sobre Github	1.4
Github	1.4.1
Información sobre Gitbook	1.5
Gitbook	1.5.1
Información sobre Gh-pages	1.6
Gh-pages	1.6.1
Información sobre Markdown	1.7
Markdown	1.7.1
Información sobre Pandoc	1.8
Pandoc	1.8.1
Información sobre Atom	1.9
Atom	1.9.1
Información sobre Cloud9	1.10
Cloud9	1.10.1

Práctica 1: Tareas iniciales

Introducción

En esta práctica se ha realizado un tutorial sobre los pasos necesarios para instalar diferentes herramientas, como también capturas de la utilización de las mismas (NodeJS, Github, Gitbook, Pandoc, Atom, etc).

- [Enlace a la publicación en gh-pages](#)
- [Enlace a la publicación en Gitbook](#)
- [Página personal de Diego](#)
- [Página personal de Alberto](#)

Información sobre nodejs

- ¿Qué es nodejs?
- Instalación de nodejs
 - Ubuntu
 - Windows

Node JS



¿Qué es nodejs?

Node es un intérprete Javascript del lado del servidor que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en una sólo una máquina física.

Instalación

Para la instalación dependerá del sistema en el que queramos que se realice la instalación, en primer lugar mostraremos los procedimientos para sistemas **Ubuntu**, seguido de los necesarios para los sistemas **Windows**.

Instalación nodejs Ubuntu

En este caso al estar bajo una distribución de Ubuntu, éste nos proporciona un repositorio de paquetes que son de fácil instalación mediante un único comando:

```
$ sudo apt-get install nodejs
```

Tras la ejecución del anterior comando, nodejs se habrá instalado correctamente.

```
$ node  
> 4+4  
8  
>
```

A su vez también es conveniente la instalación del gestor oficial de nodejs, **npm**, con el que se podrán instalar dependencias.

```
$ sudo apt-get install npm
```

Para demostrar su correcta instalación, realizaremos por medio de npm la descarga e instalación del paquete **express** de nodejs.

```
npm install express
```

```
alber7rp@alber7rp-X551CA:~$ npm install express
/home/alber7rp
  express@4.14.1
    accepts@1.3.3
      mime-types@2.1.14
        mime-db@1.26.0
      negotiator@0.6.1
      array-flatten@1.1.1
      content-disposition@0.5.2
      content-type@1.0.2
      cookie@0.3.1
      cookie-signature@1.0.6
      debug@2.2.0
        ms@0.7.1
      depd@1.1.0
      encodeurl@1.0.1
      escape-html@1.0.3
      etag@1.7.0
      finalhandler@0.5.1
        statuses@1.3.1
        unpipe@1.0.0
      fresh@0.3.0
      merge-descriptors@1.0.1
      methods@1.1.2
      on-finished@2.3.0
        ee-first@1.1.1
      parseurl@1.3.1
      path-to-regexp@0.1.7
      proxy-addr@1.1.3
        forwarded@0.1.0
        ipaddr.js@1.2.0
      qs@6.2.0
      range-parser@1.2.0
      send@0.14.2
        destroy@1.0.4
        http-errors@1.5.1
          inherits@2.0.3
            setprototypeof@1.0.2
          mime@1.3.4
          ms@0.7.2
        serve-static@1.11.2
        type-is@1.6.14
          media-typer@0.3.0
        utils-merge@1.0.0
      vary@1.1.0

npm WARN enoent ENOENT: no such file or directory, open '/home/alber7rp/package.json'
npm WARN alber7rp No description
npm WARN alber7rp No repository field.
npm WARN alber7rp No README data
npm WARN alber7rp No license field.
```

Otro ejemplo que podemos ver también es la instalación del módulo **gh-pages**, el cual se explicará su funcionamiento en el capítulo de gh-pages.

```
npm install gh-pages --save-dev
```

```

alber7rp@alber7rp-X551CA:~/Descargas/pr$ npm install gh-pages --save-dev
pr@1.0.0 /home/alber7rp/Desktop/pr
+-- gh-pages@0.12.0
| +-- async@2.1.2
| | +-- lodash@4.17.4
| +-- commander@2.9.0
| | +-- graceful-readlink@1.0.1
| +-- globby@6.1.0
| | +-- array-union@1.0.2
| | +-- array-uniq@1.0.3
| +-- glob@7.1.1
| | +-- fs.realpath@1.0.0
| | +-- inflight@1.0.6
| | +-- wrappy@1.0.2
| +-- inherits@2.0.3
| +-- minimatch@3.0.3
| | +-- brace-expansion@1.1.6
| | | +-- balanced-match@0.4.2
| | | +-- concat-map@0.0.1
| | +-- once@1.4.0
| | +-- path-is-absolute@1.0.1
| | +-- object-assign@4.1.1
| | +-- pify@2.3.0
| | +-- pinkie-promise@2.0.1
| | | +-- pinkie@2.0.4
| +-- graceful-fs@4.1.10
| +-- q@1.4.1
| +-- q-io@1.13.2
| | +-- collections@0.2.2
| | | +-- weak-map@1.0.0
| | +-- mime@1.3.4
| | +-- mimeparse@0.1.4
| | +-- qs@1.2.2
| | +-- url2@0.0.0
| +-- rimraf@2.5.4

npm WARN pr@1.0.0 No description
npm WARN pr@1.0.0 No repository field.
alber7rp@alber7rp-X551CA:~/Descargas/pr$ ls
node_modules  package.json

```

Por último, y como dato importante sobre npm, describimos la utilización de npm para desarrollar un supuesto proyecto.

```
npm init
```

Dicho comando inicializa por decirlo de alguna manera la estructura de nodejs de nuestro proyecto, básicamente crea el fichero **package.json**, en el cual se especificarán aspectos referentes a la aplicación, como puede ser la versión, dependencias, descripción, autor, scripts, repositorio de git, motor de node etc. Un ejemplo podría ser:

```
alber7rp@alber7rp-X551CA:~/Descargas/pr$ cat package.json
{
  "name": "pr",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "gh-pages": "^0.12.0"
  },
  "dependencies": {
    "express": "^4.14.1"
  },
  "description": ""
}
```

Instalación nodejs Windows

En este caso, para su instalación será necesario visitar la página web oficial de nodejs ([Aquí](#)), y descargar y posteriormente instalar la versión que deseamos.



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo.

Descargar para Windows (x64)

v6.9.5 LTS

Recomendado para la mayoría

v7.5.0 Actual

Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#) [Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

O revise la [Agenda de LTS](#).

 **LINUX FOUNDATION** COLLABORATIVE PROJECTS [Reporte un problema con Node.js](#) | [Reporte un problema con el sitio](#) | [Consiga ayuda](#)

© 2017 Node.js Foundation. All Rights Reserved. Portions of this site originally © 2017 Joyent.

Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark Guidelines of the Node.js Foundation.

Linux Foundation is a registered trademark of The Linux Foundation.

Linux is a registered trademark of Linus Torvalds.

[Node.js Project Licensing Information](#).

Información sobre Git

- Sistema de control de versiones
- ¿Qué es Git?
- Instalación de Git
 - Ubuntu
 - Windows

Git



Sistema de control de versiones

Git, es un software de control de versiones diseñado por [Linus Torvalds](#). La pregunta es ¿qué es control de versiones? Pues bien, se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración.

¿Qué es git?

Git fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente, es decir Git nos proporciona las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.

- Características de Git:
 - Rapidez en la gestión de ramas, debido a que Git nos dice que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente.
 - Gestión distribuida; Los cambios se importan como ramas adicionales y pueden ser fusionados de la misma manera como se hace en la rama local.
 - Gestión eficiente de proyectos grandes.
 - Realmacenamiento periódico en paquetes.
- Comandos más usuales
 - Iniciar un repositorio vacío en unas carpeta específica.

```
git init
```

- Añadir un archivo específico.

```
git add "nombre_de_archivo"
```

- Añadir todos los archivos del directorio

```
git add .
```

- Confirmar los cambios realizados. El “mensaje” generalmente se usa para asociar al commit una breve descripción de los cambios realizados.

```
git commit -m "mensaje"
```

- Revertir el commit identificado por "hash_commit"

```
git revert "hash_commit"
```

- Subir la rama(branch) “nombre_rama” al servidor remoto.

```
git push origin "nombre_rama"
```

- Mostrar el estado actual de la rama(branch), como los cambios que hay sin hacer commit.

```
git status
```

Instalación de Git

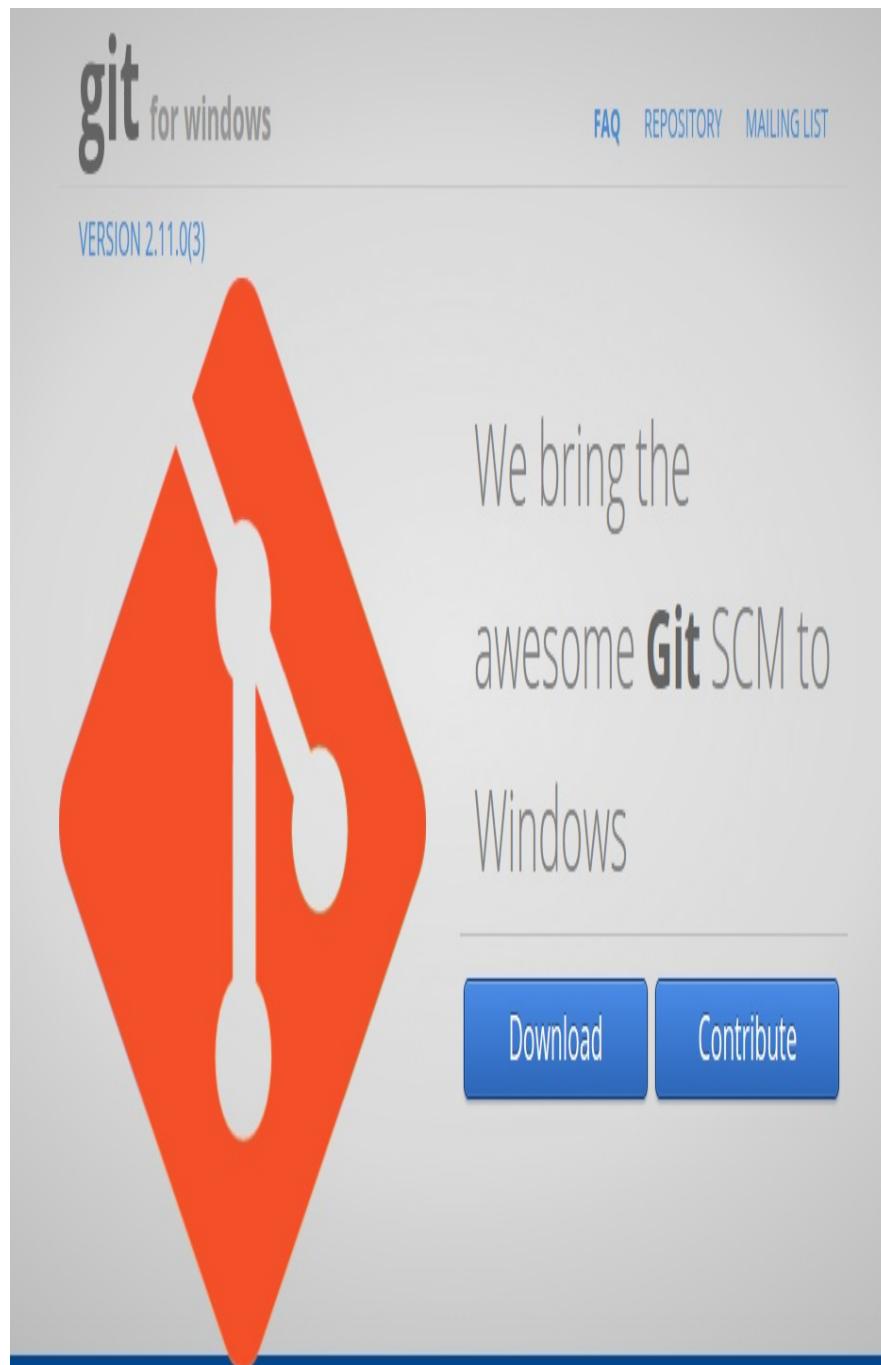
Instalación de Git en Ubuntu

Para proceder a su instalación bastará con teclear el siguiente comando en la terminal, el cual se encargará de buscar dicho paquete entre los repositorios que ofrece Ubuntu para instalarlo.

```
$ sudo apt-get install git
```

Instalación de Git en Windows

En el caso de Windows, para su instalación primero se deberá visitar la siguiente página web ([Aqui](#)) y descargar el software de instalación que ofrece.



Tras su instalación estará a nuestra disposición una versión de git a través de una shell como también mediante una interfaz gráfica.

- Version Bash



The screenshot shows a terminal window titled "MINGW32:~/git". The window contains the following text:

```
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

- Version GUI

Git Gui (git) C:/Users/Bacon/git

Repository Edit Branch Commit Merge Remote Tools Help

Current Branch: master

Unstaged Changes Staged for commit File: compat/winansi.c

```
@@ -70,11 +70,12 @@ static void warn_if_raster_font(void)
    if (!(fontFamily & TMF_TRUETYPE)) {
        const wchar_t *msg = L"\nWarning: Your console font probably "
            L"doesn't support Unicode. If you experience strange "
            L"characters in the output, consider switching to a "
            L"TrueType font such as Lucida Console!\n";
        WriteConsoleW(console, msg, wcslen(msg), NULL, NULL);
-       DWORD dummy;
+       WriteConsoleW(console, msg, wcslen(msg), &dummy, NULL);
    }
}

static int is_console(int fd)
{
@@ -112,16 +113,17 @@ static int is_console(int fd)

static void write_console(unsigned char *str, size_t len)
{
    /* only called from console_thread, so a static buffer will do */
    static wchar_t wbuf[2 * BUFFER_SIZE + 1];
+   DWORD dummy;

    /* convert utf-8 to utf-16 */
    int wlen = xutf8towcs(wbuf, (char*) str, ARRAY_SIZE(wbuf), len);

    /* write directly to console */
-   WriteConsoleW(console, wbuf, wlen, NULL, NULL);
+   WriteConsoleW(console, wbuf, wlen, &dummy, NULL);
}
```

Commit Message: New Commit Amend Last Commit

Rescan
Stage Changed
Sign Off
Commit
Push

Ready.

Información sobre Github

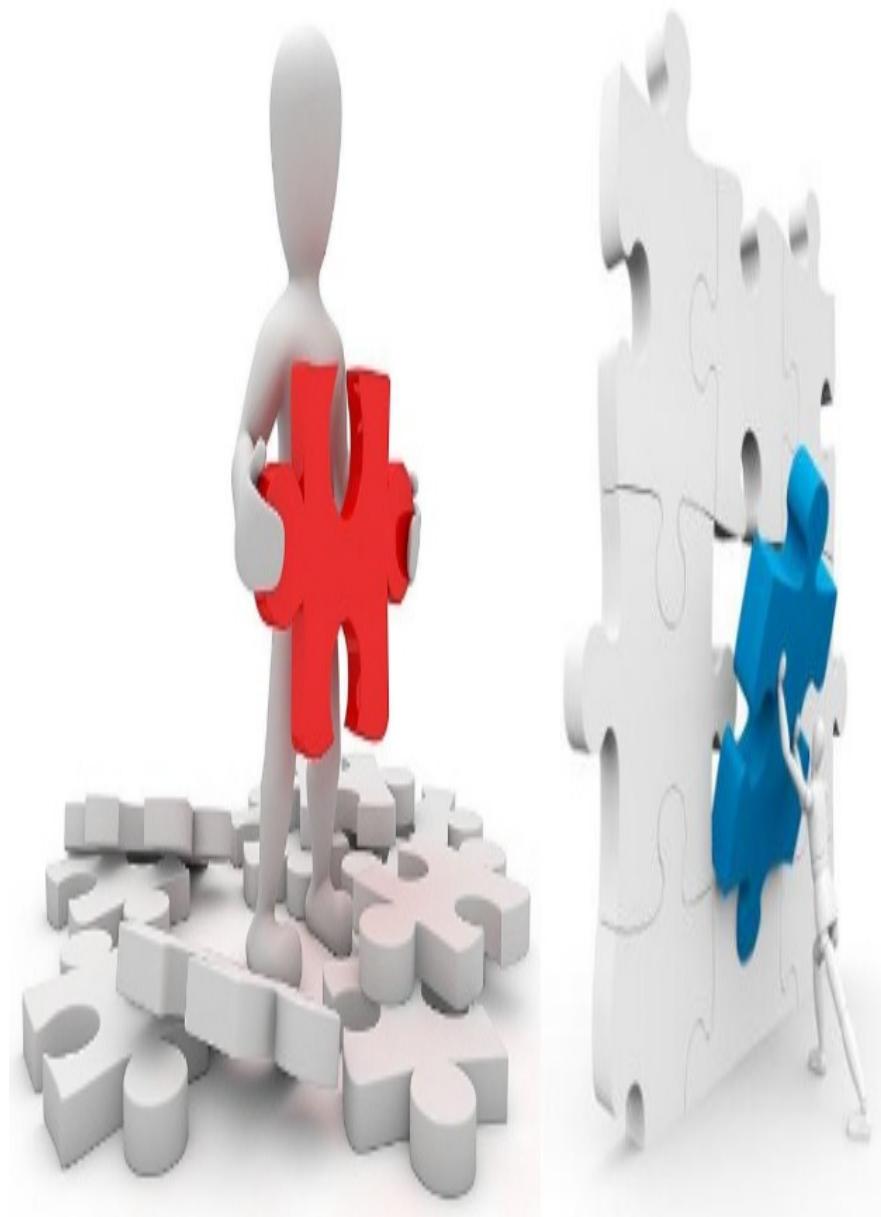
- ¿Qué es Github?
- Github desktop

Github



¿Qué es Github?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.



Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

Github desktop para Windows

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Github desktop es una versión de Github de escritorio para Windows y Mac. Ésta ofrece diferentes ventajas, como puede ser la rapidez de contribución como una mejor visibilidad de la evolución de los proyectos.

Para su instalación bastará con visitar la siguiente página web ([Aquí](#)) y descargar e instalar el software proporcionado, sincronizar con tu cuenta de github, etc.

GitHub Desktop

Overview Release Notes | Help

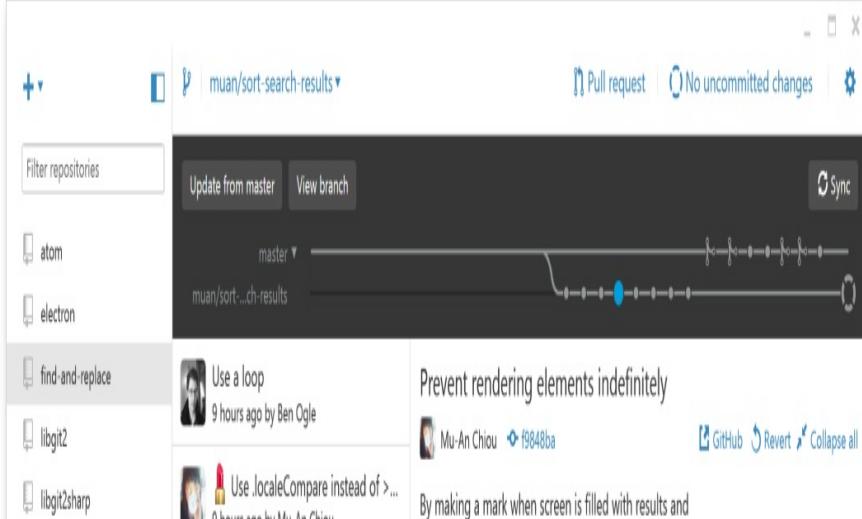
Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Available for [Mac](#) and [Windows](#)

[Download GitHub Desktop](#)
Windows 7 or later

By clicking the Download button you agree to the [End-User License Agreement](#)



De esta manera, una vez instalado se podrá acceder a todas aquellas herramientas que están disponibles en la versión web, en la siguiente imagen se puede ver un ejemplo de un proyecto donde se visualiza la diferencia de un código con respecto a nuevas instantáneas (commit).

3 changes

styles\ui-variables.less

<input type="checkbox"/> styles\panels.less	@@ -3,10 +3,10 @@
	3	3	@text-color-subtle: #999;
<input type="checkbox"/> styles\tabs.less	4	4	@text-color-highlight: #333;
<input type="checkbox"/> styles\ui-variables.less	5	5	@text-color-selected: #fff;
	6		- @text-color-info:#389DD4;
	7		- @text-color-success:#28BD3E;
	8		- @text-color-warning:#E79536;
	9		- @text-color-error:#FF6158;
	6		+ @text-color-info:#4BB8F3;
Modify info text color	7		+ @text-color-success:#6DCC50;
	8		+ @text-color-warning:#F9A646;
Tweak the info text color to be more like the OS default. Fixes #145	9		+ @text-color-error:#FF625C;
	10	10	
	11	11	// Background colors
	12	12	@background-color-info:@text-color-info;

Commit to change-colors

Hub

¿Qué es hub?

git + hub = github

hub is a command line tool that wraps git in order to extend it with extra features and commands that make working with GitHub easier.

Básicamente *hub* es una herramienta que nos facilita la interacción desde la línea de comandos con *github*, ésta es capaz de hacer todos aquellos procesos que haríamos desde la interfaz web de *github*, como por ejemplo crear un repositorio, eliminarlo, etc. Además nos provee de una sintaxis de fácil uso, ejemplo:

```
$ hub clone rtomayko/tilt  
  
# expands to:  
$ git clone git://github.com/rtomayko/tilt.git
```

Para instalar esta herramienta deberemos ejecutar los siguientes comandos:

```
$ git clone https://github.com/github/hub.git  
$ cd hub  
$ make install prefix=/usr/local
```

Información sobre Gitbook

- ¿Qué es Gitbook?
- Inicio de sesión en Gitbook
- Creación de un libro en Gitbook

Gitbook



¿Qué es Gitbook?

Gitbook es una plataforma que permite al usuario publicar la documentación de su proyecto utilizando diferentes plantillas . Las plantillas que posee Gitbook para crear una documentación son:

- Documentación tipo Libro.
- Documentación de una API.
- Base de conocimientos.

Gitbook está disponible mediante dos soportes, además de gitbook online,también está disponible [Gitbook Editor](#), que es una herramienta de creación de libros de manera local.

Inicio de sesión en Gitbook

Para el inicio de sesión de manera online es necesario que se tenga previamente una cuenta en alguno de éstos soportes: [Twitter](#), [Facebook](#), [Google+](#) o [Github](#).

Log in to Your Account



Username or Email

Password

If you don't have an account, [sign up](#). Forgot your password? [Reset it](#) with your email.

Sign in

En mi caso es recomendable el inicio de sesión mediante Github, ya que su sincronización directa con los repositorios puede dar lugar a una mayor facilidad de uso y ventajas.

Creación de un libro en Gitbook

Cuando se inicia sesión por primera vez, aparecerá la siguiente imagen con la que se puede escoger que tipo de libro queremos crear.

Create a new book

START BY SELECTING A TEMPLATE.



BOOK & MANUAL
A barebones template with all the essentials to get you started.



API DOCUMENTATION
A template setup with a perfect API documentation theme.



KNOWLEDGE BASE
A template to publish a FAQ or Knowledge base using GitBook.

v

7 Alberto G Ruiz Pérez ▾

Create Book

Existen varias maneras de crear e introducir contenido en un libro gitbook:

- Mediante **importación de archivos** (pdf, epub, html, etc).

alu0100825510 > prueba

Created a few seconds ago

ABOUT 0 DISCUSSIONS 0 UPDATES TRAFFIC SETTINGS

★ Star 0  Unsubscribe 1

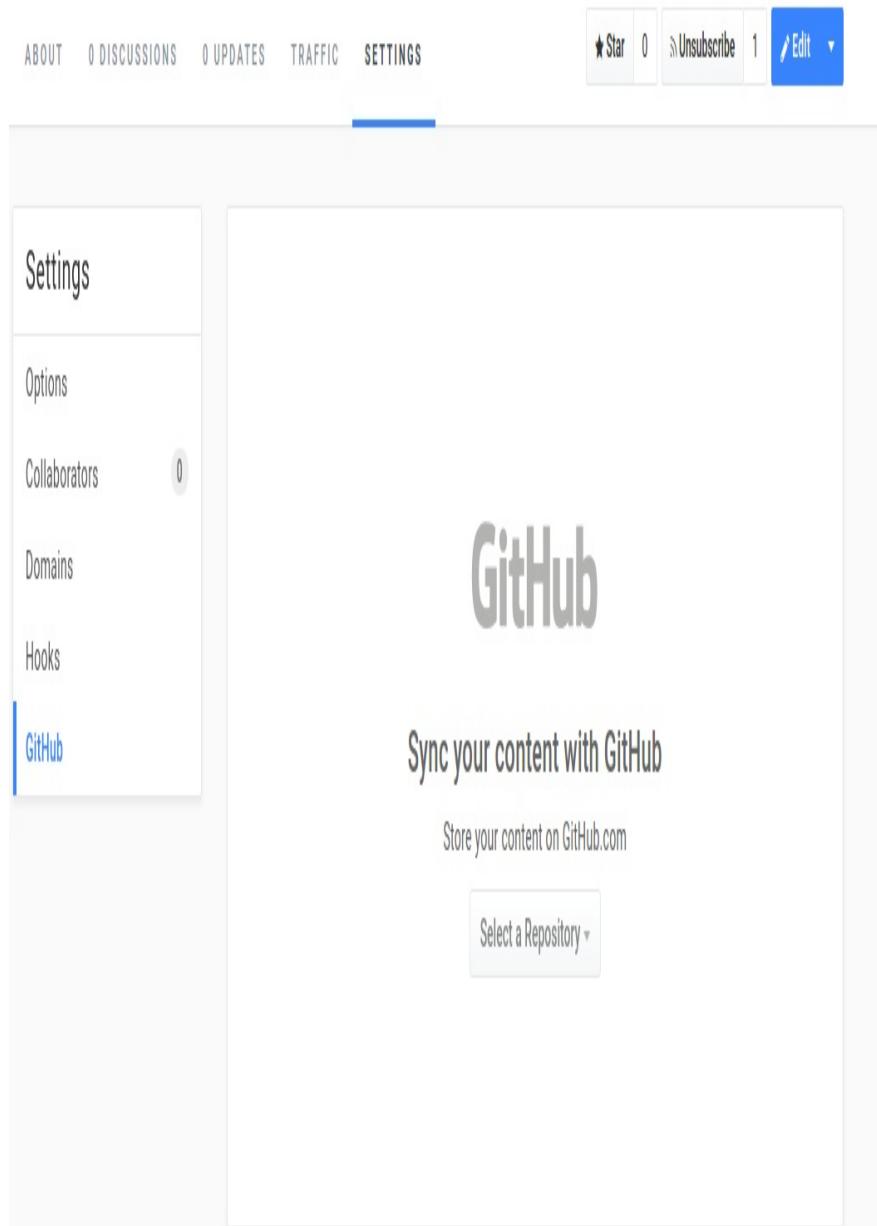


Write and publish a first draft

Or [import a document](#)

 [Edit your book](#) ▾

- Mediante sincronización de repositorio con Github.



[Learn more about the GitHub integration.](#)

- Mediante **Creación de un repositorio gitbook**. Otra de las posibilidades para el despliegue de un libro en gitbook es la creación de un repositorio en el mismo, en el que subiremos todos los archivos remotamente. Para ello será necesaria la instalación de la herramienta **gitbook** en Linux, ésta puede ser instalada de la siguiente forma:

```
sudo npm install gitbook-cli -g
```

Una vez instalados los recursos, se podrá inicializar un libro con el formato gitbook:

```
gitbook init <nombre_libro>
```

Ésto nos creará una estructura de archivos, básicamente el README.md y SUMMARY.md.

```
alber7rp@alber7rp-X551CA:~/Descargas/p/pps$ gitbook init libro
warn: no summary file in this book
info: create README.md
info: create SUMMARY.md
info: initialization is finished
alber7rp@alber7rp-X551CA:~/Descargas/p/pps$ ls libro/
README.md  SUMMARY.md
alber7rp@alber7rp-X551CA:~/Descargas/p/pps$
```

Además gitbook nos proporciona el siguiente comando para visualizar el progreso de nuestro libro en modo servidor por el puerto correspondiente a través de nuestro navegador:

```
gitbook serve
```

```
alber7rp@alber7rp-X551CA:~/Descargas/p/pp/libro$ gitbook serve
Live reload server started on port: 35729
Press CTRL+C to quit ...

info: 7 plugins are installed
info: loading plugin "livereload"... OK
info: loading plugin "highlight"... OK
info: loading plugin "search"... OK
info: loading plugin "lunr"... OK
info: loading plugin "sharing"... OK
info: loading plugin "fontsettings"... OK
info: loading plugin "theme-default"... OK
info: found 1 pages
info: found 0 asset files
info: >> generation finished with success in 1.7s !

Starting server ...
Serving book on http://localhost:4000
```

Por último, gitbook también nos ofrece la posibilidad de convertir nuestro libro en diferentes formatos, html, pdf, ebook, etc.

```
gitbook build
```

En el caso de la creación del formato html, se nos creará un directorio adicional denominado `_book`, donde se alojarán todos los ficheros de la website.

```

alber7rp@alber7rp-X551CA:~/Descargas/p/pp/libro$ gitbook build
info: 7 plugins are installed
info: 6 explicitly listed
info: loading plugin "highlight"... OK
info: loading plugin "search"... OK
info: loading plugin "lunr"... OK
info: loading plugin "sharing"... OK
info: loading plugin "fontsettings"... OK
info: loading plugin "theme-default"... OK
info: found 1 pages
info: found 0 asset files
info: >> generation finished with success in 1.0s !
alber7rp@alber7rp-X551CA:~/Descargas/p/pp/libro$ ls _book/
gitbook index.html search_index.json
alber7rp@alber7rp-X551CA:~/Descargas/p/pp/libro$ ls
_book README.md SUMMARY.md
alber7rp@alber7rp-X551CA:~/Descargas/p/pp/libro$ █

```

Por lo que si queremos finalmente subir nuestro repositorio local a uno remoto (en gitbook en este caso), y publicarlo con la metodología de Markdown, se deberá añadir la dirección de dicho repositorio remoto para luego empujarlo hacia gitbook:

```

git remote add <direccion_repo_remoto_gitbook>
git push <direccion_repo_remoto_gitbook> master

```

En gitbook, al igual que en github también consta de ramas y confirmaciones. Esta información está disponible en el apartado **update** del libro en cuestión, a continuación se puede ver un ejemplo de ello.

[ABOUT](#) [0 DISCUSSIONS](#) [0 UPDATES](#) [TRAFFIC](#) [SETTINGS](#) [Star 0](#) [Unsubscribe 1](#) [Edit ▾](#)

Branch: master ▾



Nothing to show!

There is currently no updates on this branch.

Información sobre gh-pages

- ¿Qué es gh-pages?
- Creación de una gh-page
- Módulo gh-pages despliegue automático

Gh-pages



¿Qué es gh-pages?

Las gh-pages o Github pages son páginas web que Github pone a nuestra disposición para nuestros proyectos, con la particularidad de que estas deben de estar alojadas en los repositorios propios de Github. Con el simple gesto de actualizar nuestro repositorio, también estaremos haciendo lo propio con la gh-page.

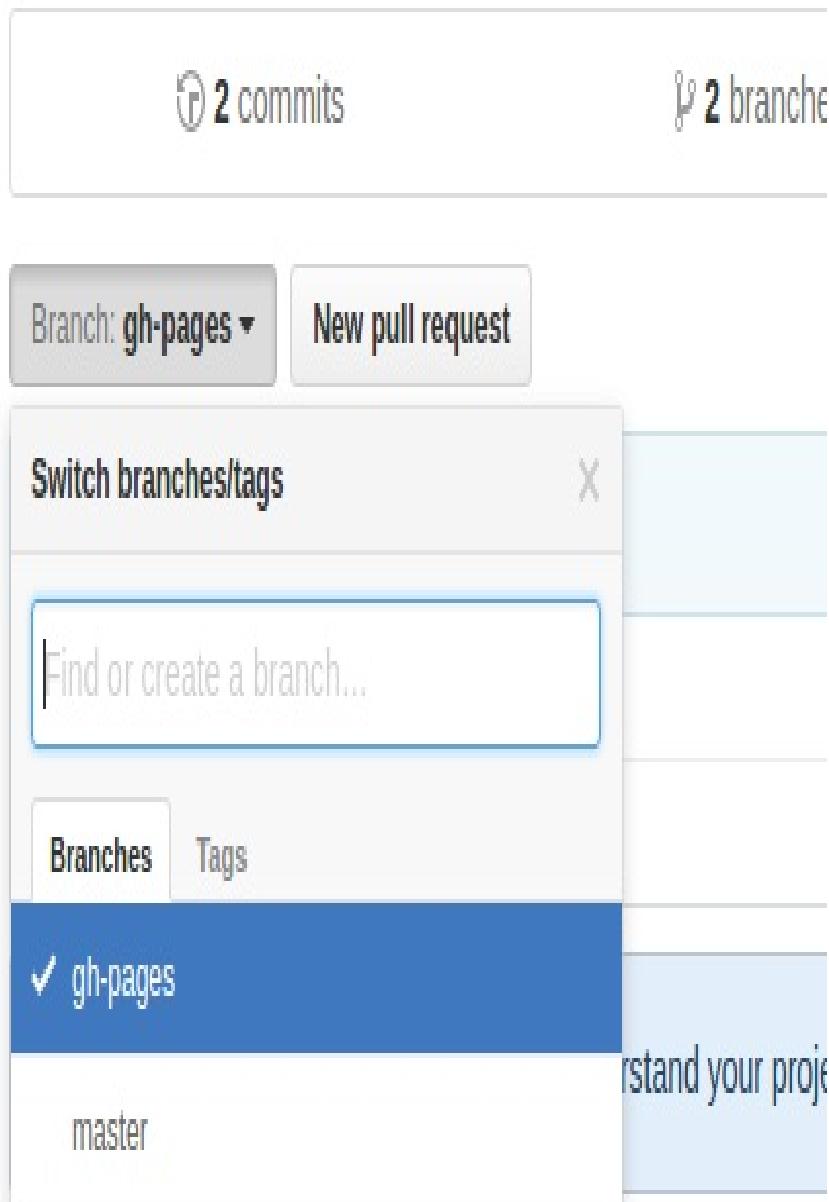
Creación de una gh-page

Suponiendo que tenemos un código correspondiente a una página web, para incluirlo en una gh-page se puede hacer de varias maneras:

- **Mediante Interfaz Gráfica:** Github proporciona por cada repositorio una opción en la que indicamos la activación de una gh-page y hacia qué rama queremos que apunte.



- **Mediante la creación directa de una rama gh-pages:** Otra manera de publicar directamente una gh-page es crear una rama llamada **gh-pages** en la que alojemos ahí nuestro código de la página web.



Creación de la rama *gh-pages* en el repo local, donde colocaremos todos los archivos perteneciente a la website que queremos alojar:

```
git branch gh-pages
```

Nos posicionamos localmente en la rama *gh-pages* creada, y subimos ahora directamente todos los archivos de esa rama, a la rama *gh-pages* del repo remoto, si no está creada, la crea automáticamente:

```
git push origin gh-pages
```

Módulo *gh-pages* despliegue automático

Este script de Node.js usa el módulo *gh-pages* como interfaz para interactuar con el repositorio. Previamente se debe instalar dicho módulo para ser utilizable.

```
npm install gh-pages --save-dev
```

Su uso básico es el siguiente:

```
var ghpages = require('gh-pages');
var path = require('path');

ghpages.publish(path.join(__dirname, 'dist'), function(err) { ... });
```

Calling this function will create a temporary clone of the current repository, create a gh-pages branch if one doesn't already exist, copy over all files from the base path, or only those that match patterns from the optional src configuration, commit all changes, and push to the origin remote.

If a gh-pages branch already exists, it will be updated with all commits from the remote before adding any commits from the provided src files.

Básicamente tras la ejecución de este script se realizará un despliegue de gh-pages en el repositorio que le indiquemos a la función **publish()**, si dicha rama **gh-pages** no existe, la creará de forma automática.

Veamos un ejemplo real:

- **Paso 1:** Iniciamos el proyecto creando el fichero *package.json* que describirá y mantendrá nuestra aplicación:

```
npm init
```

- **Paso 2:** Instalamos el modulo de gh-pages para incluirlo en nuestro proyecto:

```
npm install gh-pages --save-dev
```

```

alber7rp@alber7rp-X551CA:~/Descargas/pr$ npm install gh-pages --save-dev
pr@1.0.0 /home/alber7rp/Desktop/pr
+-- gh-pages@0.12.0
| +-- async@2.1.2
| | +-- lodash@4.17.4
| +-- commander@2.9.0
| | +-- graceful-readlink@1.0.1
| +-- globby@6.1.0
| | +-- array-union@1.0.2
| | +-- array-uniq@1.0.3
| +-- glob@7.1.1
| | +-- fs.realpath@1.0.0
| | +-- inflight@1.0.6
| | +-- wrappy@1.0.2
| | +-- inherits@2.0.3
| | +-- minimatch@3.0.3
| | | +-- brace-expansion@1.1.6
| | | | +-- balanced-match@0.4.2
| | | | +-- concat-map@0.0.1
| | +-- once@1.4.0
| | +-- path-is-absolute@1.0.1
| | +-- object-assign@4.1.1
| | +-- pify@2.3.0
| | +-- pinkie-promise@2.0.1
| | | +-- pinkie@2.0.4
| | +-- graceful-fs@4.1.10
| | +-- q@1.4.1
| | +-- q-io@1.13.2
| | | +-- collections@0.2.2
| | | | +-- weak-map@1.0.0
| | | +-- mime@1.3.4
| | | +-- mimeparse@0.1.4
| | | +-- qs@1.2.2
| | | +-- url2@0.0.0
| | +-- rimraf@2.5.4

npm WARN pr@1.0.0 No description
npm WARN pr@1.0.0 No repository field.
alber7rp@alber7rp-X551CA:~/Descargas/pr$ ls
node_modules  package.json

```

- **Paso 3:** Creamos un fichero *js*, por ejemplo *deploy.js*, en el que introduciremos el código que incluirá llamadas al módulo de *gh-pages* para hacer la subida a github de forma automática:

```

var ghpages = require('gh-pages');
var path = require('path');

ghpages.publish(path.join(__dirname, '.'), {
repo: 'https://github.com/alu0100825510/prueba.git'}, function(err) {
console.log(err);
console.log("Se ha subido a gh-pages correctamente");
});

```

```
alber7rp0a1ber7rp-x551ca@~/Descargas/pr$ touch deploy.js
alber7rp0a1ber7rp-x551ca@~/Descargas/pr$ nodejs deploy.js
Username for 'https://github.com': aluo100825510
Password for 'https://aluo100825510@github.com':
undefined
Se ha subido a gh-pages correctamente
alber7rp0a1ber7rp-x551ca@~/Descargas/pr$
```

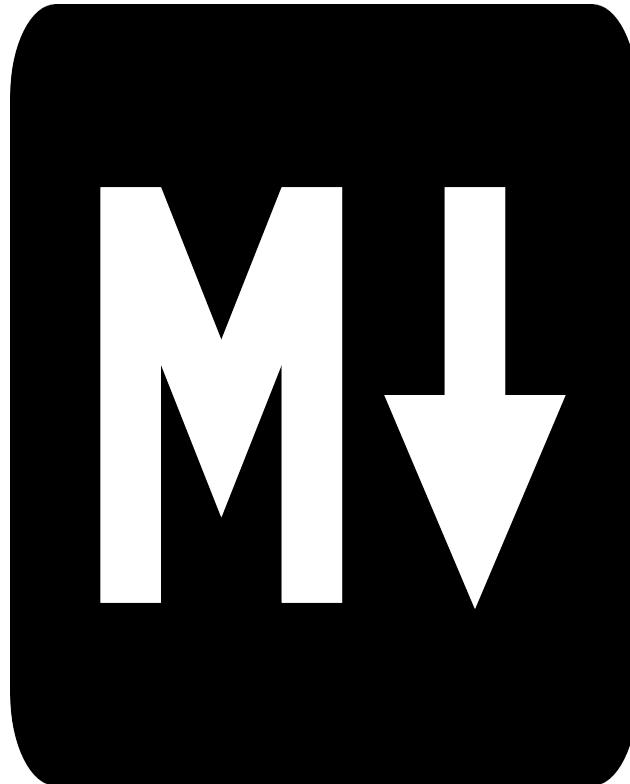
Tras la ejecución de éste último paso, podemos ver en nuestro repositorio remoto de github que se han subido correctamente los archivos a la rama *gh-pages*:



Información sobre Markdown

- ¿Qué es Markdown?
- Sintaxis de Markdown
- Implementaciones

Markdown



¿Qué es Markdown?

Es un lenguaje de marcado ligero creado por John Gruber que trata de conseguir la máxima legibilidad y facilidad de publicación tanto en su forma de entrada como de salida.

- Se distribuye bajo licencia BSD y se distribuye como plugin (o al menos está disponible) en diferentes sistemas de gestión de contenidos (CMS).
- Markdown convierte el texto marcado en documentos XHTML utilizando html2text creado por Aaron Swartz.
- Fue implementado originalmente en Perl por Gruber, pero desde entonces se ha traducido a multitud de lenguajes de programación, incluyendo PHP, Python, Ruby, Java y Common Lisp.
- No hay una norma definida para Markdown, aparte de la implementación original de John Gruber, que algunos consideran obsoleta. Esto ha conducido a fragmentación, pues distintos proveedores han escrito sus propias variantes de Markdown

Sintaxis de Markdown:

En este apartado descubrirás la sintaxis Markdown y los conceptos básicos para escribir utilizando este lenguaje de marcado. De hecho, al final del mismo estarás perfectamente capacitado para empezar a utilizar este lenguaje en tus escritos y publicaciones.

- **Encabezados:** se generan cuando se encuentra una almohadilla antes de texto. Encabezado h1 vendrá precedido de un símbolo "#", Encabezado h2 vendrá precedido de dos símbolos "##" y así sucesivamente.

```
# Encabezado h1
## Encabezado h2
### Encabezado h3
#### Encabezado h4
##### Encabezado h5
###### Encabezado h6
```

- **Saltos de línea:** se generan cuando se encuentran dos espacios juntos

```
"Quien fue a Sevilla,  
perdió su silla"
```

- **Citas:** Para citar solo es necesario escribir una cuña antes del texto

```
> La vida es muy corta para aprender alemán. -Tad Marburg
```

- **Texto con énfasis:** Debe agregarse un asterisco para cursiva y dos para negrita

```
*éñfasis* (cursiva)  
**éñfasis fuerte** (negrita)
```

- **Listas**

```
* Un elemento en una lista no ordenada  
* Otro elemento en una lista
```

```
1. Elemento en una lista numerada u ordenada.  
2. Otro elemento
```

- **Enlaces**

```
[Texto del enlace aquí](URL "Título del enlace")
```

- **Imágenes**

```
![Texto alternativo](URL "Título de la imagen")
```

Implementaciones:

Hay implementaciones de Markdown para muchos lenguajes de programación; también algunas plataformas y frameworks soportan el uso de Markdown.

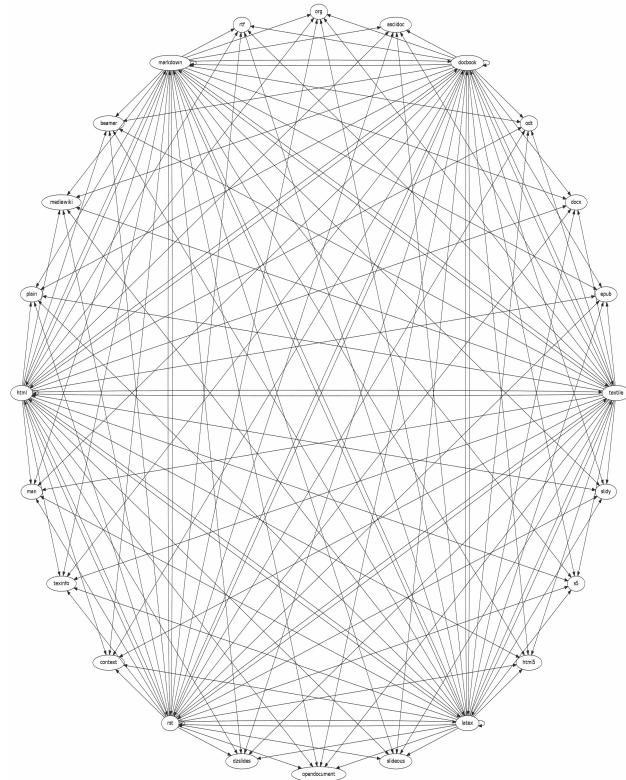
Mientras Markdown es muy sencillo y fácil de leer y escribir con un editor de texto plano (como el Bloc de Notas o gedit), hay editores especializados en Markdown (como Haroopad) disponibles para muchos sistemas operativos, como Linux, Mac OS, Windows, Android e iOS. Estos editores también tienen una ventana para previsualizar nuestro documento como si se tratase de un documento WYSIWYG. Ejemplos:

- PHP: PHP Markdown
- Python: markdown.py (de webpy.org)
- Python: (de python.org)
- Perl: Text::Markdown
- C#: Markdown.NET
- Ruby: BlueCloth
- Java: markdownj, MarkdownPapers
- JavaScript: Showdown
- Lua: markdown.lua
- MultiMarkdown, una versión en Perl de Markdown con ciertas características extra, así como otras herramientas relacionadas con Markdown.

Información sobre Pandoc

- ¿Qué es Pandoc?
- Instalación de Pandoc en Windows
- Instalación de Pandoc en Ubuntu

Pandoc

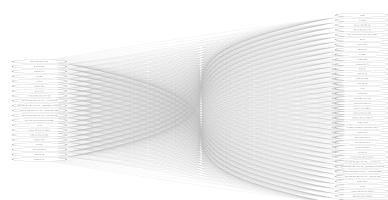


¿Qué es Pandoc?

Si necesita convertir archivos de un formato de marcado a otro, pandoc es su herramienta perfecta. Pandoc puede convertir documentos en markdown, reStructuredText, textile, HTML, DocBook, LaTeX, MediaWiki markup, TWiki markup, OPML, Emacs Org-Mode, Txt2Tags, Microsoft Word docx, LibreOffice ODT, EPUB, o Haddock markup a:

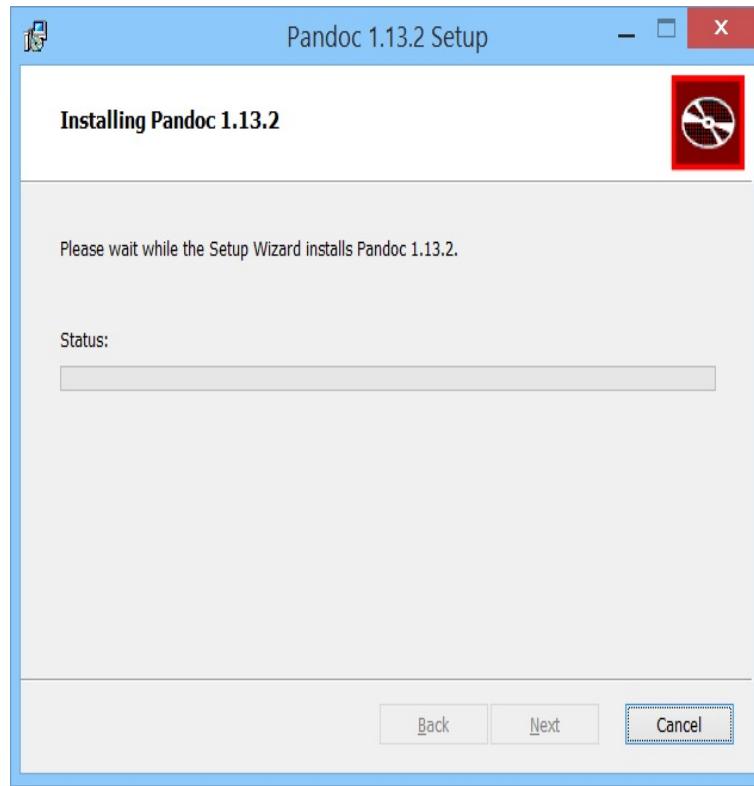
- HTML formats: XHTML, HTML5, and HTML slide shows using Slidy, reveal.js, Slideous, S5, or DZSlides.
- Word processor formats: Microsoft Word docx, OpenOffice/LibreOffice ODT, OpenDocument XML Ebooks: EPUB version 2 or 3, FictionBook2
- Documentation formats: DocBook, TEI Simple, GNU TexInfo, Groff man pages, Haddock markup
- Page layout formats: InDesign ICML
- Outline formats: OPML
- TeX formats: LaTeX, ConTeXt, LaTeX Beamer slides
- PDF via LaTeX
- Lightweight markup formats: Markdown (including CommonMark), reStructuredText, AsciiDoc, MediaWiki markup, DokuWiki markup, Emacs Org-Mode, Textile
- Custom formats: custom writers can be written in lua.

Pandoc entiende una serie de sintaxis de markdown muy útil, incluyendo metadatos del documento (título, autor, fecha); Notas a pie de página; mesas; Listas de definición; Superíndice y subíndice; Listas ordenadas mejoradas; Listas de ejemplos corrientes; Bloques de código delimitados con resaltado de sintaxis; Citas inteligentes, guiones y elipses; Markdown dentro de los bloques HTML; Y LaTeX en línea. Si se desea una compatibilidad de marcado estricta, todas estas extensiones se pueden desactivar.

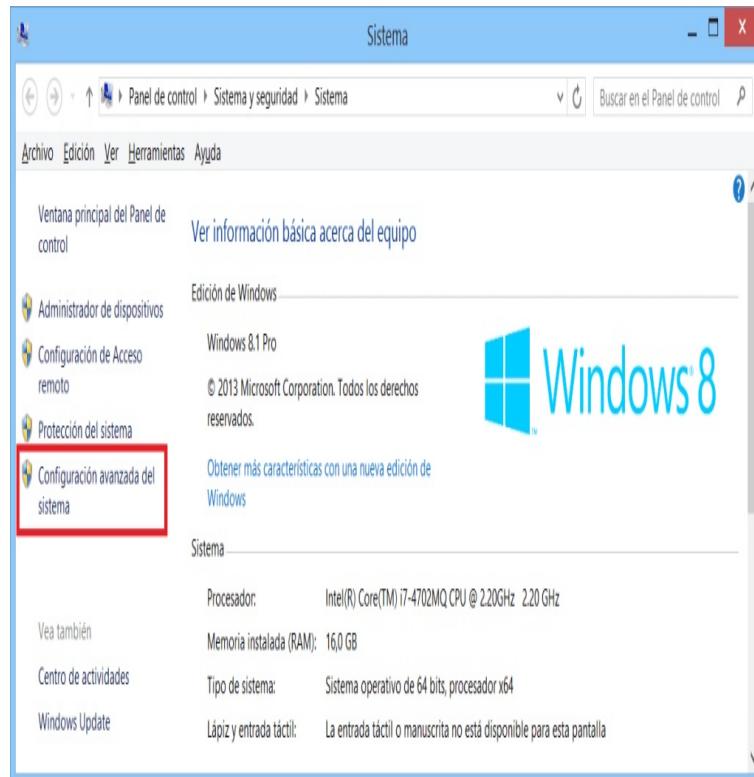


Instalación de Pandoc en Windows

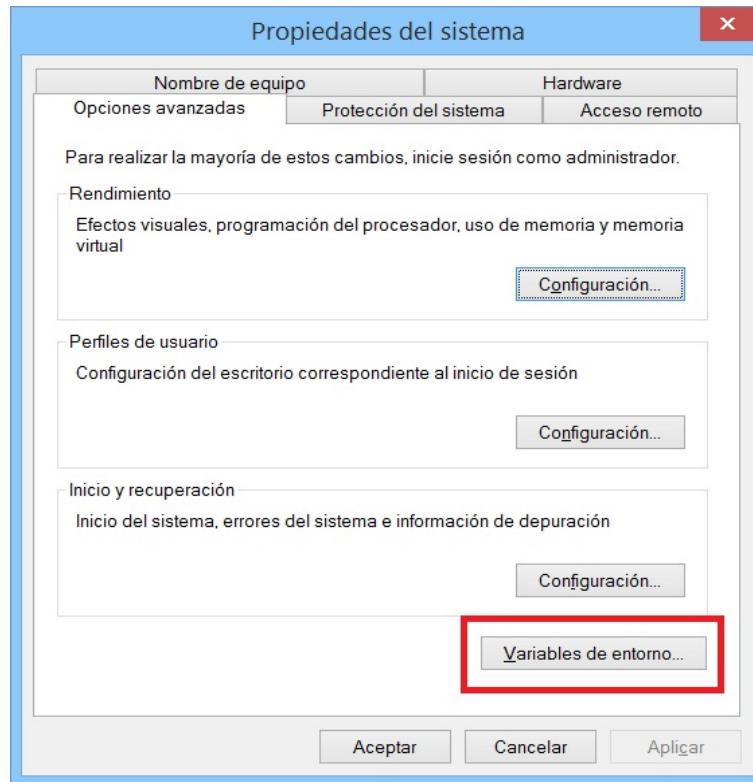
- Paso 1. Descargar el programa desde [aquí](#)
- Para la generación de PDF recomendamos [Miktex](#)



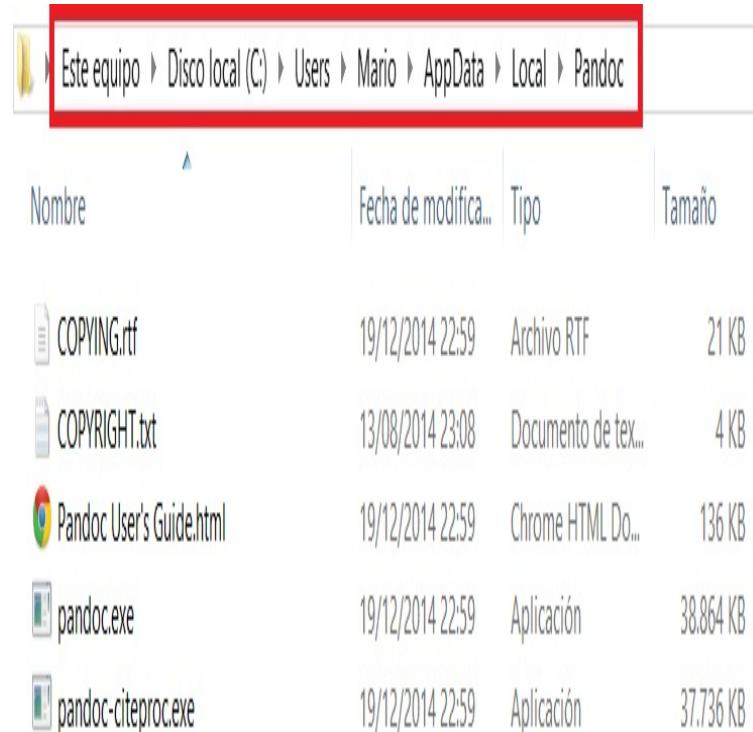
- Paso 2. Cuando esté Pandoc instalado, debemos añadirlo a las variables de entorno de Windows, ya que si no lo hacemos, Sublime Text no podrá ejecutar el comando. Para ello, vamos a la configuración avanzada del sistema:



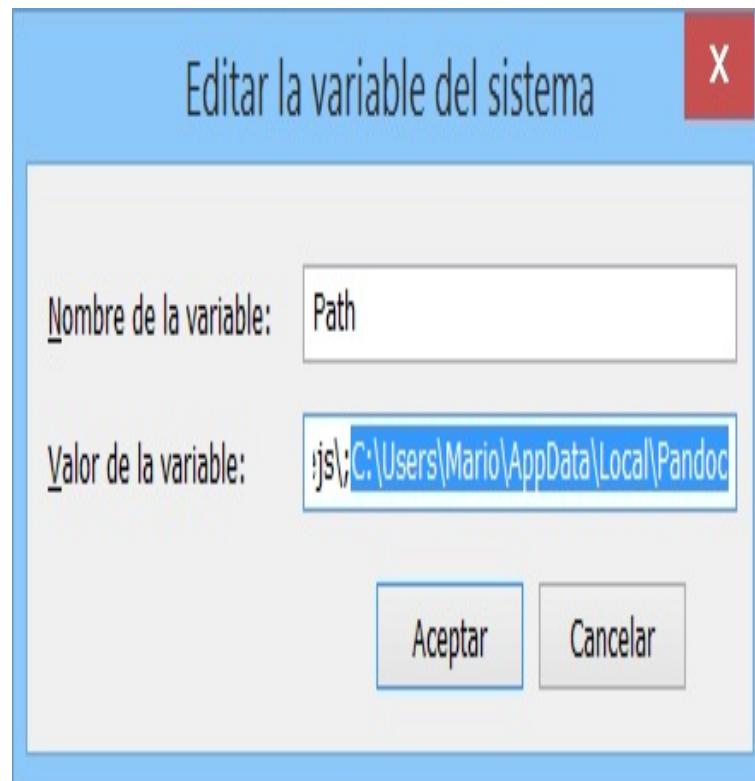
- Paso 3. Se abrirá una ventana y tenéis que hacer click en "Variables de entorno":



- Paso 4. Ahora tenéis que copiar como texto (click derecho, "copiar como texto") la ruta del directorio de instalación de Pandoc:



- Paso 5. La dirección que habeis copiado la teneis que introducir en la variable **Path**



Una vez llegueis a este punto, todo **listo**.

Instalación de Pandoc en Ubuntu

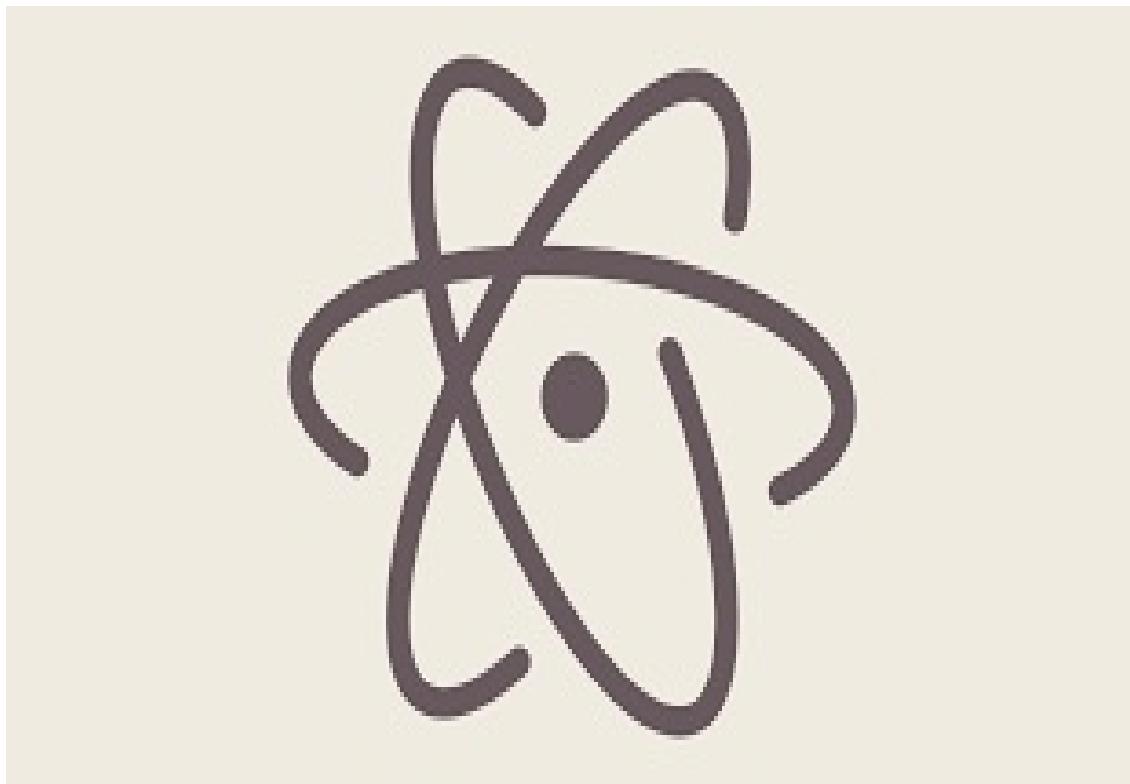
Hay que introducir el siguiente comando por consola y ya estará todo listo. Tarda un poco, así que paciencia.

```
sudo apt-get install pandoc texlive
```

Información sobre Atom

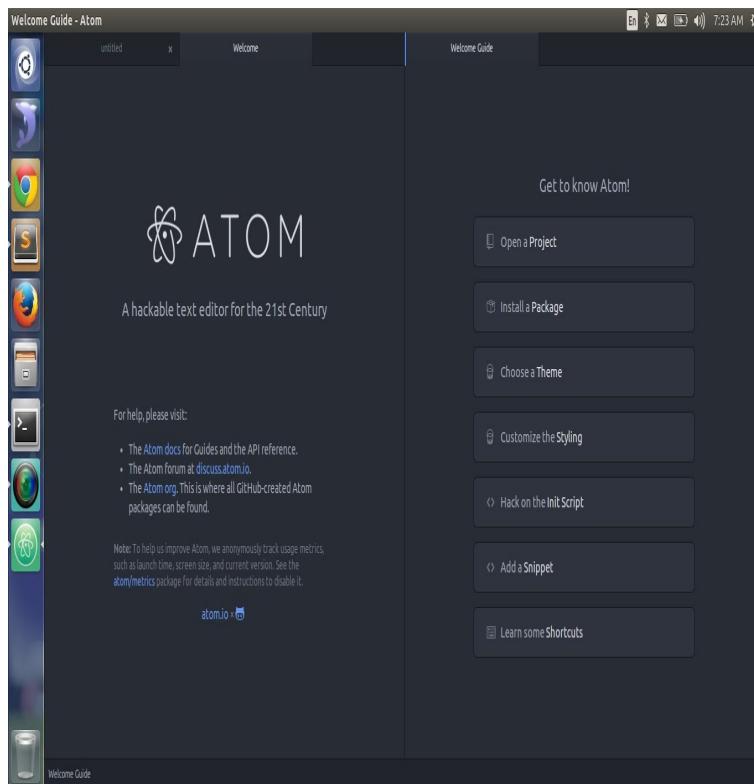
- ¿Qué es Atom?
- Instalar Atom en Linux
- Instalar Atom en Windows

Atom



¿Qué es Atom?

Atom es un editor de código de fuente abierta para macOS, Linux, y Windows con soporte para plug-ins escrito en Node.js. Incrustando Git Control, desarrollado por GitHub. Es una aplicación de escritorio construida utilizando tecnologías web. Está basado en Electrón (Anteriormente conocido como Atom Shell), un framework que permite aplicaciones de escritorio multiplataforma usando Chromium y Node.js. También puede ser utilizado como un entorno de desarrollo integrado (IDE). Atom libero su beta en la versión 1.0. Sus desarrolladores lo llaman un "Editor de textos hackable para el siglo XXI".



Instalar Atom en Linux

Atom es el editor de texto "hackable" de GitHub, totalmente basado en tecnologías web. Para poder instalarlo comenzamos con las dependencias básicas:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential git libgnome-keyring-dev fakeroot
```

También instalaremos node.js ya que Atom depende de este entorno de programación en javascript:

```
curl -sL https://deb.nodesource.com/setup | sudo bash -
sudo apt-get install -y nodejs
```

Para saber la versión de Node.js:

```
node -v
```

Node.js instalará npm automáticamente, para configurarlo:

```
npm config set python /usr/bin/python2 -g
```

Para instalar Atom abrimos un terminal y clonamos el código de Git:

```
git clone https://github.com/atom/atom
```

Cambiamos al directorio de Atom:

```
cd atom
git fetch -p
git checkout $(git describe --tags `git rev-list --tags --max-count=1`)
```

Construimos Atom:

```
script/build
```

Ahora ya podemos ejecutar Atom. Otra opción es crear un archivo ejecutable .deb, lo hacemos con el siguiente comando:

```
script/grunt mkdeb
```

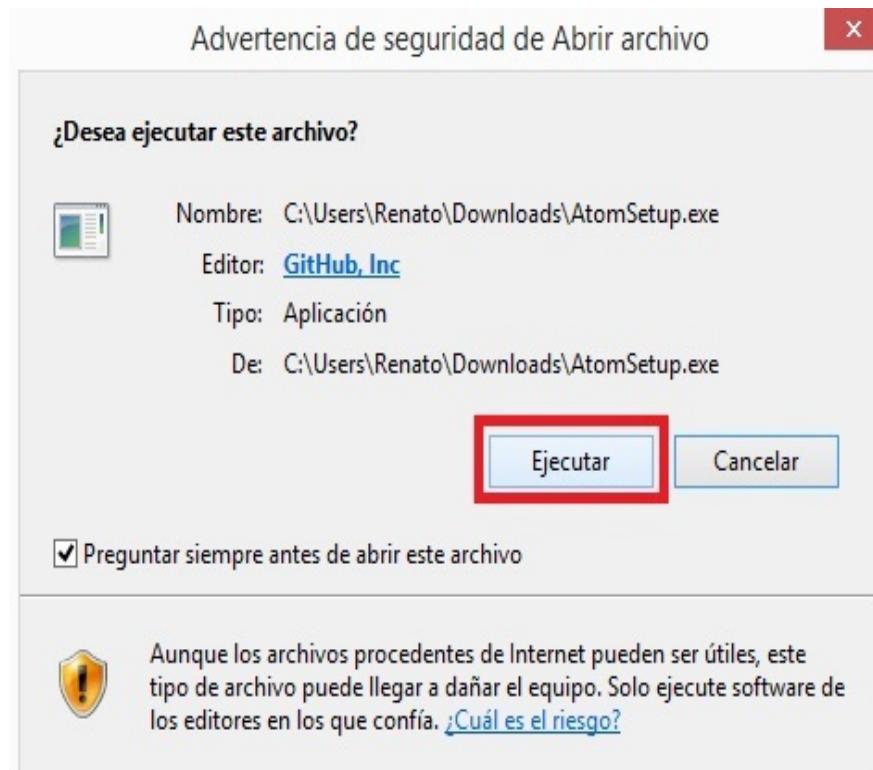
Instalar Atom en Windows

Podemos encontrar el instalador del editor Atom en el siguiente link: [Atom](#)

- Podrán encontrar un botón de descarga "Download Windows Installer".



- Una vez que hayamos descargado el programa ejecutamos el instalador.



El instalador tardará unos cuantos minutos para descargarse y tendremos listo el programa

Información sobre Cloud9

- ¿Qué es Cloud9?
- Instalación y uso.



Cloud9 IDE

Your code anywhere, anytime

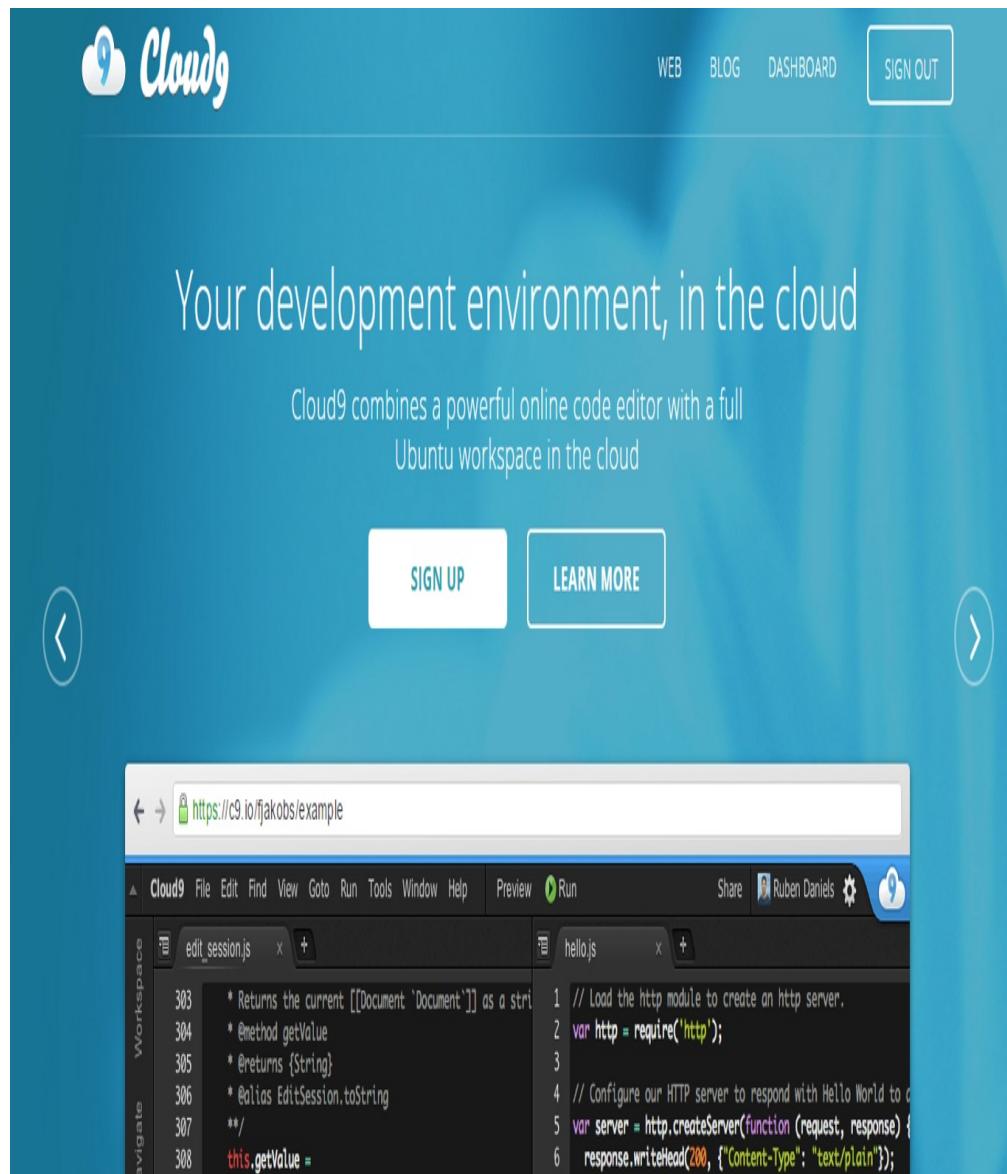
¿Qué es Cloud9?

Cloud9 (c9) es un entorno de desarrollo online que nos permite no solo editar código, también administrar un servidor virtual. Básicamente lo que tienes aquí es un programa que se ejecuta sobre el navegador pero que a pesar de ese limitante tiene las funcionalidades más destacadas de un IDE en el que además tenemos un entorno de trabajo real donde podemos poner nuestros programas en ejecución, ya sean sitios web o programas ejecutables.

Instalación y uso

Como es un servicio completamente online **no tendremos que instalar nada en nuestra máquina**. Debemos seguir unos sencillos pasos a continuación:

- **Paso 1.** Diríjase al siguiente enlace: <http://c9.io>



- **Paso 2.** Cree una cuenta (usando BitBucket, Github o una nueva cuenta de Cloud9) o inicie sesión si ya tiene una cuenta para ver su Panel de Control.



Sign up for Cloud9 (free)

Sign up with Github, Bitbucket or Email



Desired username

Your email

Your email, once more

Choose a password

Receive our monthly newsletter

This is included in the free plan:

- 1 Private Workspace
- 512 MB RAM
- 1.5 GB disk space
- 1 FTP Workspace

With a *Premium* account you get:

- 6 Private Workspaces
- 1 GB RAM
- 5 GB disk space
- Unlimited FTP Workspaces
- Connect to your own VM

All plans include full Shell Access/Terminal,
plus unlimited Public Workspaces &
Collaborators.

Switch to a Premium account?

- **Paso 3.** Entre a su Dashboard.

Blog

Workspaces

Filter...

Brady Dowling

Menu

- Workspaces
- Shared With Me
- Repositories

Create a new workspace

demo-project Cloned from ajaxorg/demo-project This project has no description yet. Updated a few seconds ago.	wordpress wordpress This is a wordpress environment that has the Advance EyeCare theme installed on it. Updated 2 days ago.	topazandsapphire default This is the Digital Ocean server with topazandsapphire.com installed on it. Handle with care. Updated 21 days ago.	dev-machine default This project has no description yet. Updated 14 hours ago.
1 CPU 512MB RAM 5GB HDD	1 CPU 512MB RAM 1GB HDD	ssh://root@45.55.182.171	ssh://ubuntu@10.240.144.45

wordpress-playground wordpress This is a place where you can toy around with Wordpress and do whatever you'd like to. Updated 2 months ago.	ghost nodejs This project has no description yet. Updated a month ago.	packthisstuff.github_io Cloned from packthisstuff/packthisstuff... This project has no description yet. Updated an hour ago.	blog Cloned from c9/blog This project has no description yet. Updated 25 days ago.	advanceicare-live default This is the Digital Ocean server that advanceicare.com is on. Updated 4 days ago.
1 CPU 512MB RAM 1GB HDD	1 CPU 1GB RAM 10GB HDD	1 CPU 512MB RAM 1GB HDD	1 CPU 512MB RAM 5GB HDD	ssh://root@107.170.108.166

profile-styleguide Cloned from c9/profile-styleguide This project has no description yet. Updated 20 hours ago.	cloud9 Cloned from c9/newclient Code for the Cloud9 IDE Updated 20 days ago.	whatstuprovo Cloned from whatstuprovo/whatstuprovo... The homepage for What's Up Provo Updated a day ago.	telescope Cloned from TelescopeJS/sample-project Testing out the telescope framework for upvoting stuff. Updated 5 days ago.	telescope-new meteor This project has no description yet. Updated 5 days ago.
1 CPU 512MB RAM 1GB HDD	1 CPU 2GB RAM 20GB HDD	1 CPU 512MB RAM 1GB HDD	1 CPU 512MB RAM 1GB HDD	1 CPU 768MB RAM 1GB HDD

telecope-test **hackhands-test** **testing-new-tab**

- **Paso 4.** Una vez en nuestra dashboard deberemos crear un nuevo workspace. Este puede ser público o privado. Podremos predefinir una plantilla diferente según el lenguaje de programación que vayamos a utilizar si conocemos esta información como se muestra en las dos imágenes siguientes.

The screenshot shows the Cloud9 IDE interface. On the left, there's a sidebar with a blue header containing a user icon with a '9' and the name 'omgjolthahaha'. Below the header are menu items: MENU, Workspaces, Shared With Me, and Repositories. The main area has a title 'Workspaces' and a search bar with a 'Filter...' placeholder and a magnifying glass icon. A large gray button on the left says '+ Create a new workspace'. A white modal window is open, titled 'Demo Project' with the subtitle '(Cloned from ajaxorg/demo-project)'. It contains text: 'We provide this demo workspace in Cloud9. It contains sample projects for Node.js, PHP, Python, Ruby, and HTML5.' At the bottom of the modal is a green 'Open' button. A red arrow points from the text 'Click here' to the 'Open' button.

9

omgjolthahaha

MENU

Workspaces

Shared With Me

Repositories

Workspaces

Filter...

+ Create a new workspace

Demo Project
(Cloned from ajaxorg/demo-project)

We provide this demo workspace in Cloud9. It contains sample projects for Node.js, PHP, Python, Ruby, and HTML5.

Open

Click here

© 2015 Cloud9 IDE Inc.

Blog Support Status

Choose a template



Custom



HTML5



Node.js



Meteor



PHP, Apache & MySQL



Python



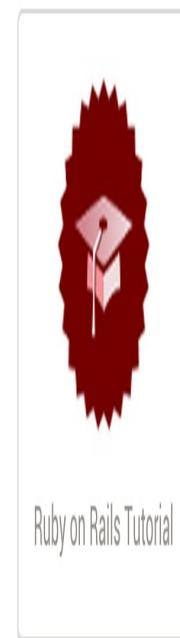
Ruby



C++



Wordpress



Ruby on Rails Tutorial

Create workspace

- **Paso 5.** Una vez creado el workspace podremos empezar a editarlo creando subcarpetas, archivos y teniendo a nuestra disposición una terminal como se muestra a continuación.

File Edit Selection Find View Goto Tools Share Help Preview Run

README.md

WORKSPACE FILES

mfc

hello-world.php

php.ini

README.md

1

2

3

4

5

6

7

8

9

10 Hi there! Welcome to Cloud9 IDE!

11

12 To get you started, we have created a small hello world application.

13

14 1) Open the hello-world.php file

15

16 2) Follow the run instructions in the file's comments

17

18 And that's all there is to it! Just have fun. Go ahead and edit the code,

19 or add new files. It's all up to you!

20

21 Happy coding!

22 The Cloud9 IDE team

23

24

25 # Support & Documentation

26

Terminal Output

mfcdev@mfc:~/mfc\$

```
user_name:~/workspace $
```