

Nombre del instructor	Indicar el nombre del instructor Adolfo Centeno Tellez		
Área temática	Full Stack		
Rol			
Competencia	Full Stack		
Subcompetencia	Front End		
Objetivo Módulo Conocimiento	Crear el ambiente de desarrollo local que permita ejecutar aplicaciones usando el framework Angular		
Módulo Conocimiento (5h en total )			
	Duración	Objetivos	Subtemas
Tema 1 - Creación y configuración del ambiente de trabajo para el framework Angular	30m	Explorar conceptos esenciales de Angular, como componentes, plantillas y servicios. Además, de temas como estructuras de aplicaciones modulares, servicios en Angular, además del enfoque de desarrollo dirigido por pruebas y finalmente la puesta en producción de una plataforma web.	1.- Concepto de Angular y sus principales características  2.- Instalación y configuración del gestor de versiones de Node llamado nvm  3.- Instalación de Node 20.0 y Angular 18.0

### TEMA 1: Creacion y configuracion del ambiente de trabajo para el framework Angular

#### Introducción al tema

En la actualidad el desarrollo de plataformas web es una tecnología en constante evolución, influenciado por tendencias tecnológicas, cambios en el comportamiento de los usuarios y mejoras en la experiencia de usuario.

El cómputo en la nube ha cambiado la forma en que se desarrollan y se implementan aplicaciones web. Las tecnologías como AWS, Azure y Google Cloud permiten a los desarrolladores crear aplicaciones escalables y flexibles sin la necesidad de invertir en infraestructura física.

Existen muchas opciones para crear aplicaciones web, que van desde HTML puro, o una combinación de HTML, CSS y JavaScript. JavaScript es el lenguaje de programación más popular para el desarrollo web, y se han creado múltiples frameworks que facilitan el proceso de desarrollo tales como:

**Angular:** Creado por Google, es un framework robusto y maduro para desarrollo web frontend. Utiliza TypeScript y se enfoca en la creación de aplicaciones web escalables y complejas. Angular es un marco de trabajo robusto, maduro, compatible con TypeScript, enfocado en la creación de aplicaciones web escalables y complejas.

**React:** Desarrollado por Facebook, es una *biblioteca* JavaScript popular para crear interfaces de usuario (UI). React es conocido por capacidad para reutilizar componentes es compatible con JavaScript y enfocado en la creación de interfaces de usuario interactivas.

**Vue.js:** Un framework frontend moderado y flexible, conocido por su curva de aprendizaje leve y su capacidad para crear aplicaciones web de una sola página (SPA). Vue.js es compatible con TypeScript y se enfoca en la creación de interfaces de usuario personalizables.

**Ember.js:** Un framework frontend maduro y estable, conocido por su arquitectura basada para crear aplicaciones web complejas y escalables. Ember.js es compatible con TypeScript y se enfoca en la creación de aplicaciones web robustas y mantenibles.

Existen otras alternativas menos populares en la actualidad como : jQuery, Qooxdoo, Dojo Toolkit, entre otros.

Adicionalmente el mercado exigen crear plataformas web muy robustas y escalables, con soporte de tecnologías emergentes muy marcados como las siguientes:

- El diseño responsivo y la accesibilidad son fundamentales para garantizar que las aplicaciones web sean compatibles con diferentes dispositivos y usuarios con discapacidades.
- La integración de machine learning y inteligencia artificial en el desarrollo web ha mejorado la personalización de la experiencia del usuario y la capacidad de las aplicaciones para aprender y mejorar con el tiempo.
- Las APIs y los microservicios han permitido a los desarrolladores crear aplicaciones más escalables y flexibles, permitiendo la integración de diferentes sistemas y servicios.
- Las bases de datos NoSQL como Firestore, MongoDB, Cassandra y Redis han reemplazado a las bases de datos relacionales tradicionales, ofreciendo mayor flexibilidad y escalabilidad.
- La automatización y el DevOps han mejorado la velocidad y la eficiencia en el desarrollo y el despliegue de aplicaciones web, permitiendo a los desarrolladores centrarse en la creación de valor para el usuario.
- Algunas tendencias emergentes en el desarrollo web incluyen la utilización de tecnologías como Blockchain, Internet of Things (IoT) y realidad aumentada (AR) para crear experiencias innovadoras y disruptivas.

Al finalizar este tema serás capaz crear un ambiente de desarrollo basado en Nodejs y el framework angular para crear aplicaciones web altamente escalables. Para lograr este objetivo primero aprenderás como instalar y configurar desde cero el gestor de versiones nvm, administrar diferentes versiones de node e instalar la versión 20.0 que usaremos en los proyectos del curso, instalaras el framework Angular en la versión 18.0 la cual es compatible con la versión 20.0 de nodejs.

### SUBTEMA 1. Concepto de Angular y sus principales características

**Angular** es un framework o marco de trabajo de software libre para el desarrollo de aplicaciones web, está basado en el lenguaje TypeScript.

*TypeScript (TS) es un lenguaje de programación construido a un nivel superior de JavaScript (JS). Esto quiere decir que TypeScript dota al lenguaje de varias características adicionales que hacen que podamos escribir código con menos errores, más sencillo, coherente y fácil de probar, en definitiva, más limpio y sólido.*

Angular es un **framework** ya que ofrece una estructura de soporte conceptual y tecnológica definida, generalmente, con artefactos o módulos de software específicos, que pueden servir como base para la organización y desarrollo de software, esto ofrece muchas ventajas sobre el desarrollo desde cero con HTML, CSS y Javascript.

Como framework, Angular ofrece una plantilla, esquema o estructura conceptual basada en tecnología que permite trabajar de una forma mucho más sencilla. De esta forma, se evitan posibles errores de programación.

Por tanto el framework Angular es un conjunto de herramientas y módulos que se pueden reutilizar para diferentes proyectos. Facilitando en varios aspectos el desarrollo, mejorando el tiempo, esfuerzo, organización.

Angular es desarrollado por Google y por una comunidad de programadores. El ecosistema de Angular está compuesto por aproximadamente 1.7 millones de programadores.

De acuerdo a “The Stack Overflow Developer Survey”, Angular es una de los frameworks de frontend más usados, al igual que React y Flutter.

Principales características del framework Angular.

#### **Arquitectura basada en componentes**

Angular usa una arquitectura basada en componentes, lo que permite a los desarrolladores construir elementos encapsulados y reusables. Cada

componente aísla su HTML, CSS y TypeScript, haciendo más fácil administrarlo y probar de forma individual cada elemento.

### **Servicios**

Los servicios son objetos que proporcionan funcionalidad común a varios componentes. Pueden ser utilizados para manejar la autenticación, la comunicación con un servidor o la gestión de datos.

### **Módulos**

Los módulos son paquetes de componentes, directivas y servicios que se utilizan para organizar la estructura de una aplicación. Cada módulo tiene su propio espacio de nombres y puede ser cargado o descargado según sea necesario.

### **Inyección de dependencias**

Angular utiliza una inyección de dependencias para proporcionar servicios y objetos a los componentes. Esto permite a los desarrolladores escribir código más modular y reutilizable.

### **Templates**

Los templates (plantillas) en Angular son archivos HTML que contienen directivas y expresiones que se evalúan dinámicamente. Los templates se utilizan para definir la estructura y el contenido de los componentes.

### **Router**

El router de Angular es responsable de manejar la navegación entre las diferentes páginas y rutas de una aplicación. Permite a los desarrolladores definir rutas y manejar la navegación programáticamente.

### Angular CLI

Angular CLI (Command Line Interface) proporciona un conjunto de herramientas para crear, construir, probar y desplegar aplicaciones de Angular. Permite el desarrollo rápido de las aplicaciones y facilita las tareas de desarrollo de software.

#### **Sala de inspiración**

Piensa en tus plataformas web favoritas como instagram, youtube, amazon, google docs, entre otras.

- o Analiza en qué framework crees estén construidas
- o Como desarrollador que framework usarías para crear una plataforma como amazon.
- o Si usaras Angular, en cuantos componentes dividirías la plataforma de youtube.

Pendiente

#### **Sala de pruebas**

Dadas las conclusiones a las que han llegado, ahora usarás Github como herramienta para administrar el código de tus proyectos. Pensando en este modelo:

**Pregunta 1:** *Si te piden realizar una página web muy simple, donde no tengas que instalar ninguna librería adicional, que usarías?*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b>  Jquery	Incorrecta:  Jquery es una librería de javascript, pero debe ser instalada o bien ser insertada mediante un CDN (Content Delivery Network)
<b>Opción b:</b>	

React	<p>Incorrecta:</p> <p>React es una librería de Javascript que debe instalarse manualmente</p>
<p><b>Opción c:</b></p> <p>Angular</p>	<p>Incorrecta:</p> <p>Angular es un framework para crear aplicaciones web, que debe instalarse adicionalmente.</p>
<p><b>Opción d:</b></p> <p><b>HTML, CSS y Javascript</b></p>	<p><b>Correcta:</b></p> <p>El stack tecnológico mínimo con HTML, CSS y Javascript puede ser ejecutado directamente por el navegador sin necesidad de recursos adicionales.</p>

**Pregunta 2:** ¿Angular por default usa el lenguaje de programación?

Opciones de respuesta	Retroalimentación
<p><b>Opción a:</b></p> <p>Javascript</p>	<p>Incorrecta:</p> <p>Javascript es el lenguaje de programación que puede ser interpretado directamente por el navegador, pero angular no está basado en Javascript.</p>
<p><b>Opción b:</b></p> <p><b>Typescript</b></p>	<p><b>Correcta:</b></p> <p>Angular está basado en Typescript, una versión mejorada de javascript con tipos de datos fuertemente tipados y características básicas de orientación a objetos.</p>
<p><b>Opción c:</b></p> <p><b>HTML</b></p>	<p>Incorrecta:</p>

	HTML no es un lenguaje de programación, sino solo un lenguaje de marcado de hipertexto.
<b>Opción d:</b>  <b>CSS</b>	Incorrecta:  CSS son hojas de estilo para mejorar la apariencia del HTML

**Pregunta 3:** En angular, permite crear, construir, probar y desplegar aplicaciones de Angular

Opciones de respuesta	Retroalimentación
<b>Opción a:</b>  Router	Incorrecta:  Router es una característica de Angular que permite redireccionar el control hacia otros componentes.
<b>Opción b:</b>  <b>Services</b>	Incorrecta:  Services es una característica de Angular que permite tener conectividad con APIs externos a la aplicación de Angular.
<b>Opción c:</b>  <b>Componentes</b>	Incorrecta:  Los componentes son unidades funcionales de Angular, compuestas por un HTML, un CSS, un archivo Typescript y uno de pruebas.
<b>Opción d:</b>  <b>CLI</b>	<b>Correcta:</b>  Angular CLI es el cliente angular en línea de comandos que permite automatizar tareas comunes como crear un proyecto, ejecutarlo en modo front, modo test, hacer build, entre muchos otros.





### SUBTEMA 2. Instalación y configuración del gestor de versiones de Node llamado nvm

En el módulo anterior mencionamos los diferentes frameworks y librerías basadas en Javascript para crear aplicaciones Web, así como características generales de Angular.

Es esta sección se instala y configura la herramientas git necesaria en nuestro ambiente de desarrollo para administrar el código de nuestros proyecto. También se instalará el gestor de versiones de nodejs, indispensable para instalar la versión 20.0 que usaremos en este curso, y que además nos permite instalar Angular como una biblioteca adicional de nodejs.

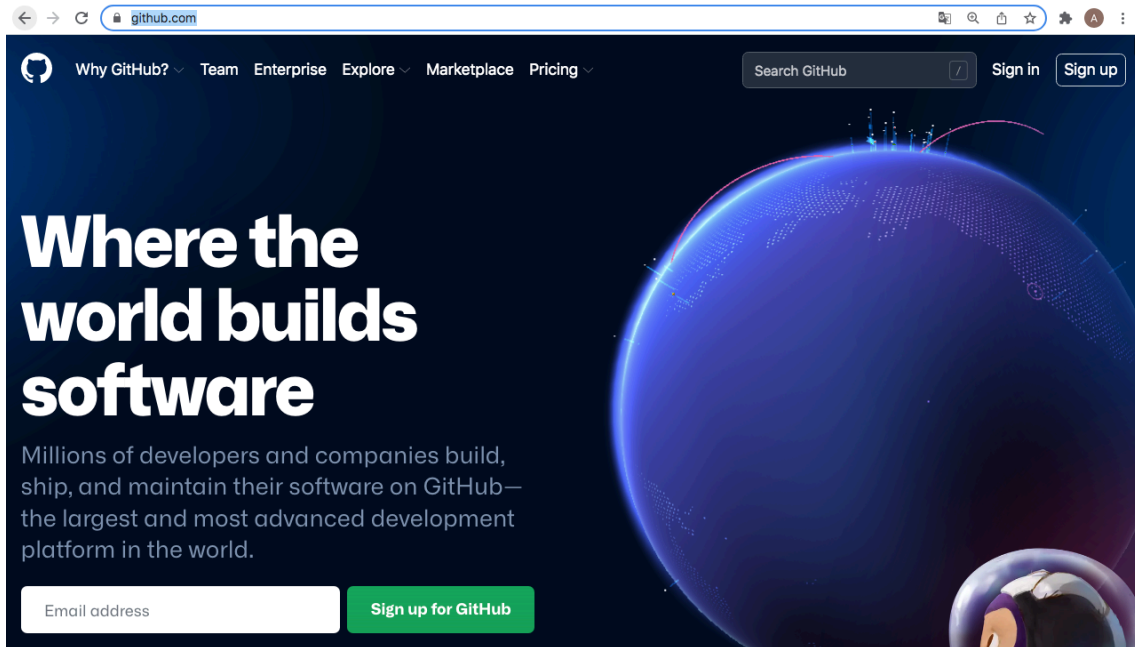
Es importante mencionar que cada proyecto en Angular está configurado para ser usado como un repositorio de git, listo para ser guardado en una herramienta de control de versiones, y listo para ser desplegado en producción en una nube de cómputo.

*Nota: Una herramienta de versionamiento de código usa el sistema de control de versiones **Git** diseñado por Linus Torvalds el creador del sistema operativo Linux. En un sistema de gestión de versiones los desarrolladores pueden administrar su proyecto, ordenar el código de cada una de las nuevas versiones y así evitar confusiones. Así, al tener copias de cada una de las versiones de su aplicación, no se perderán los estados anteriores cuando se actualiza el código.*

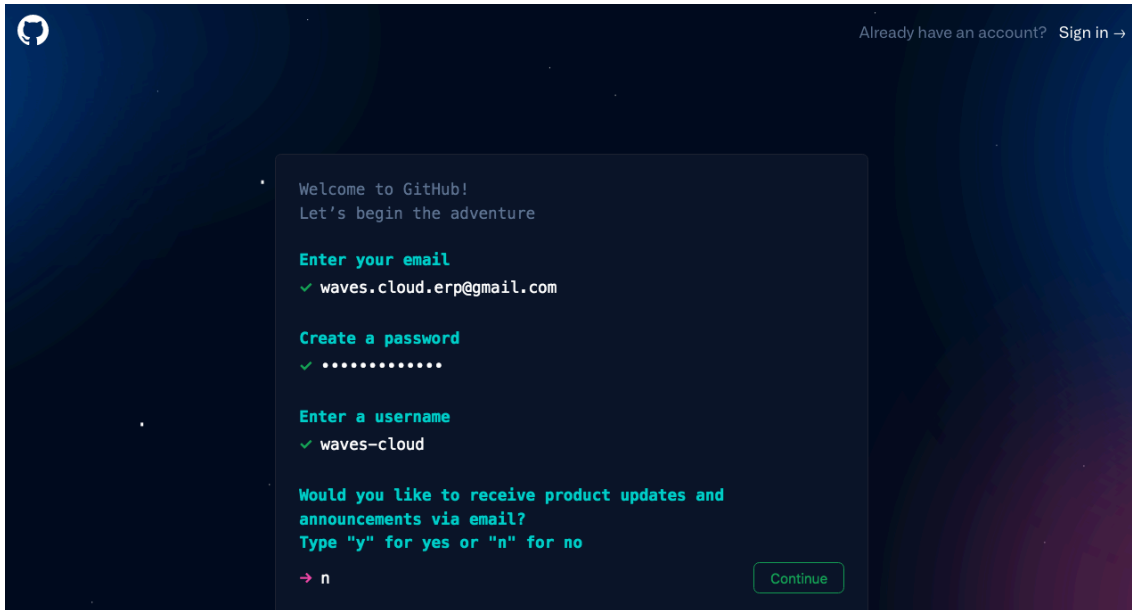
#### Creación de cuenta en github

Antes de instalar nuestro gestor de versiones de node llamado nvm, crearemos una cuenta en la plataforma <https://github.com>

Para usar Github, primero debemos crear una cuenta así que ingresa a: <https://github.com/> y presiona el botón **Sign up** que aparece en la esquina superior derecha.



- Posteriormente se te pedirá ingreses tu email (puedes usar cualquiera que ya tengas creado de gmail, outlook, institucional), password y nombre de usuario git. El nombre de usuario git es muy importante ya que todos los repositorios nuevos contienen tu nombre de usuario.



Welcome to GitHub!  
Let's begin the adventure

Enter your email  
✓ waves.cloud.erp@gmail.com

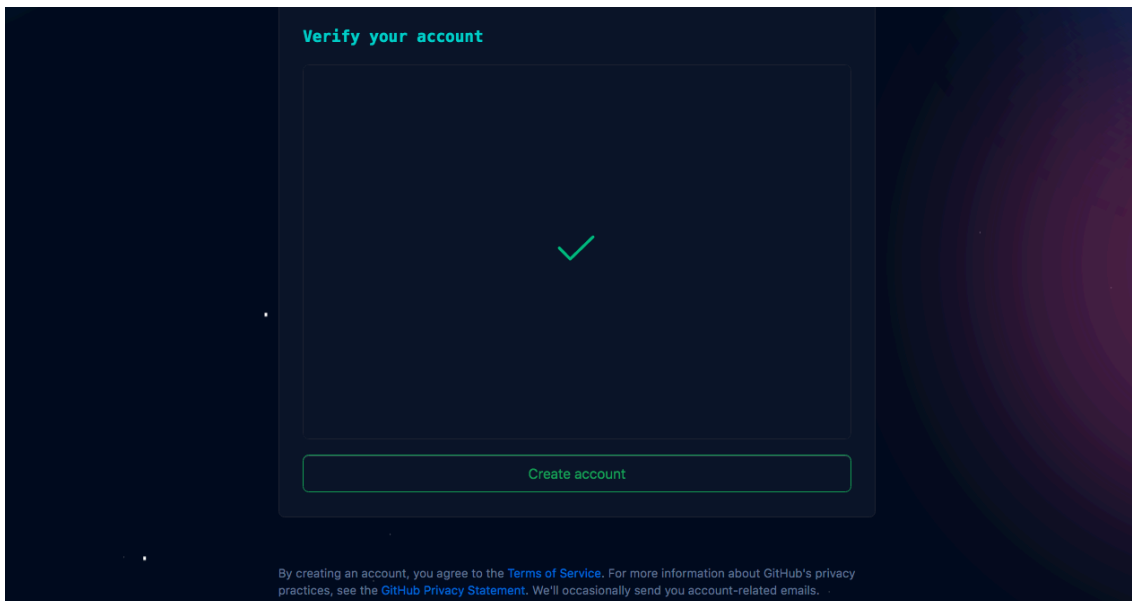
Create a password  
✓ .....

Enter a username  
✓ waves-cloud

Would you like to receive product updates and announcements via email?  
Type "y" for yes or "n" for no  
→ n

Continue

- Finalmente para asegurar que eres un humano y no un robot se te pedirá resuelvas un **Puzzle** que se genera de forma automática y aleatoria, una vez resuelto te confirmara si deseas crear tu cuenta.



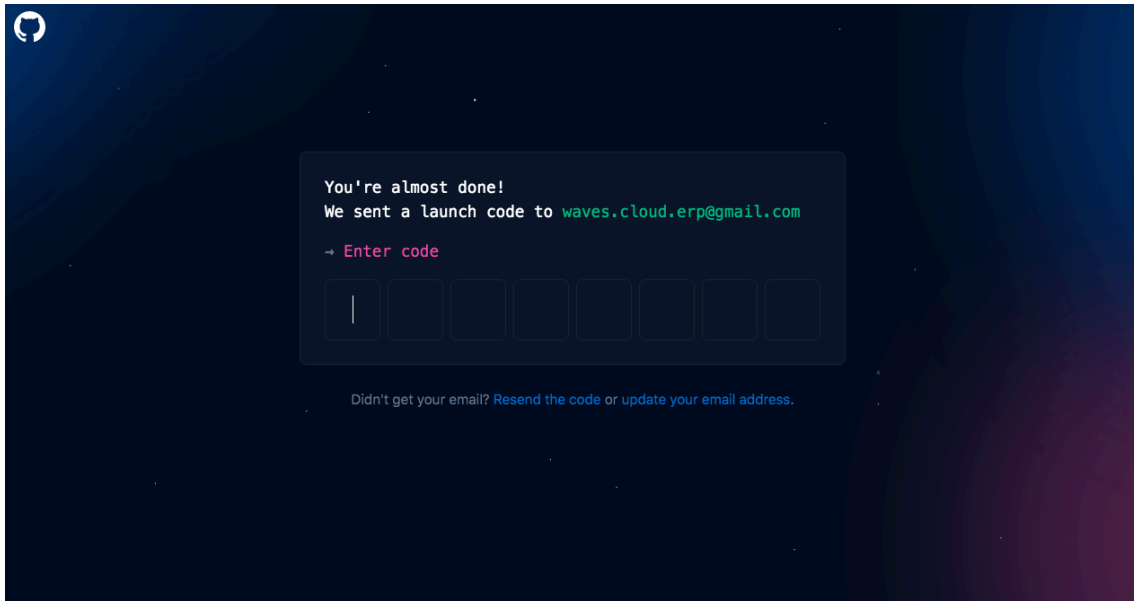
Verify your account

✓

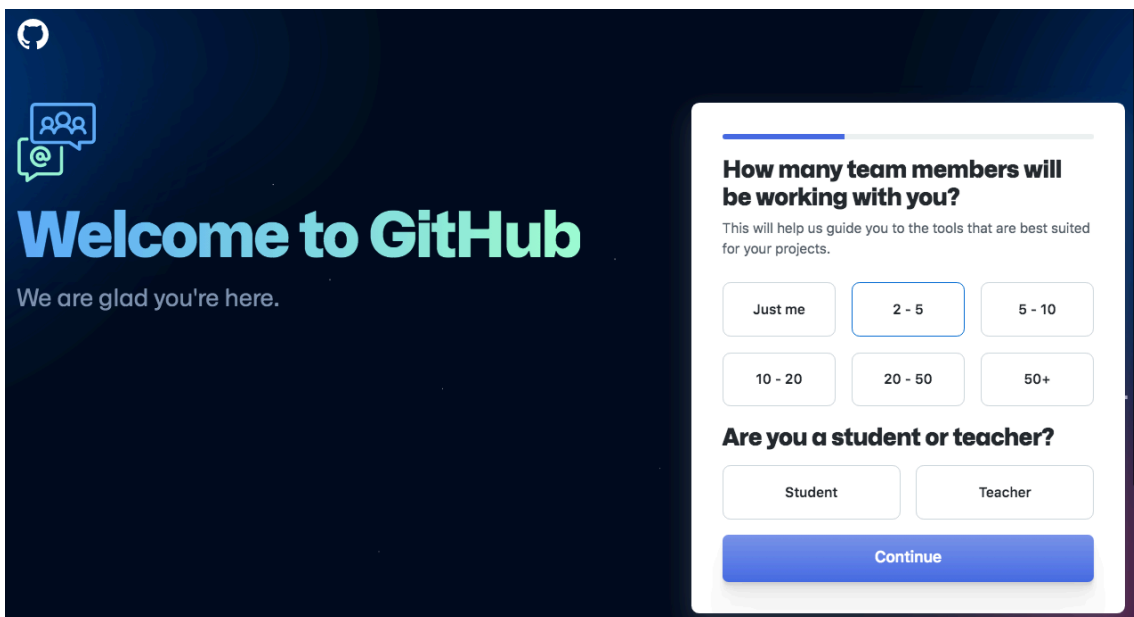
Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

- Posteriormente se te pedirá verificar con un código enviado a tu email proporcionado en los pasos anteriores

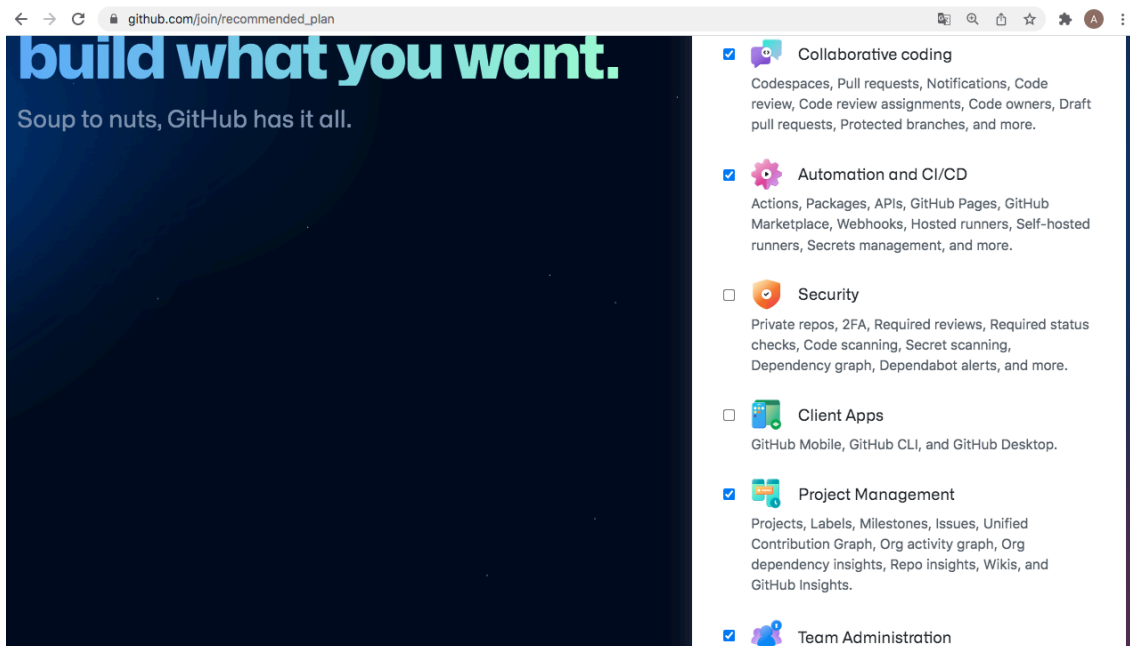


- Otro punto importante es escoger el tamaño de tu equipo de ingenieros, teniendo en cuenta que los equipos de 4 ingenieros pueden tener repositorios privados sin costo. Asimismo podemos tener repositorios públicos con cualquier cantidad de ingenieros. Elegimos equipos de 2-5 integrantes.

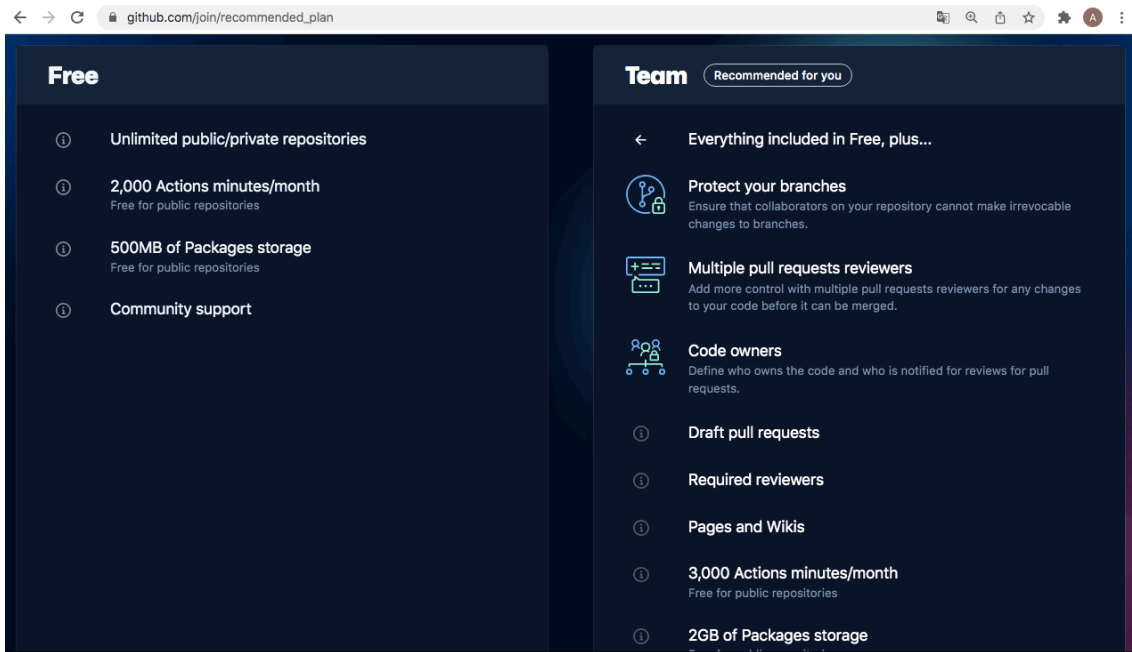


- Después debemos seleccionar las características que usaremos de Github, podemos seleccionar las 4 básicas relacionadas con la administración de proyectos como son: Trabajo de código colaborativo, automatización para

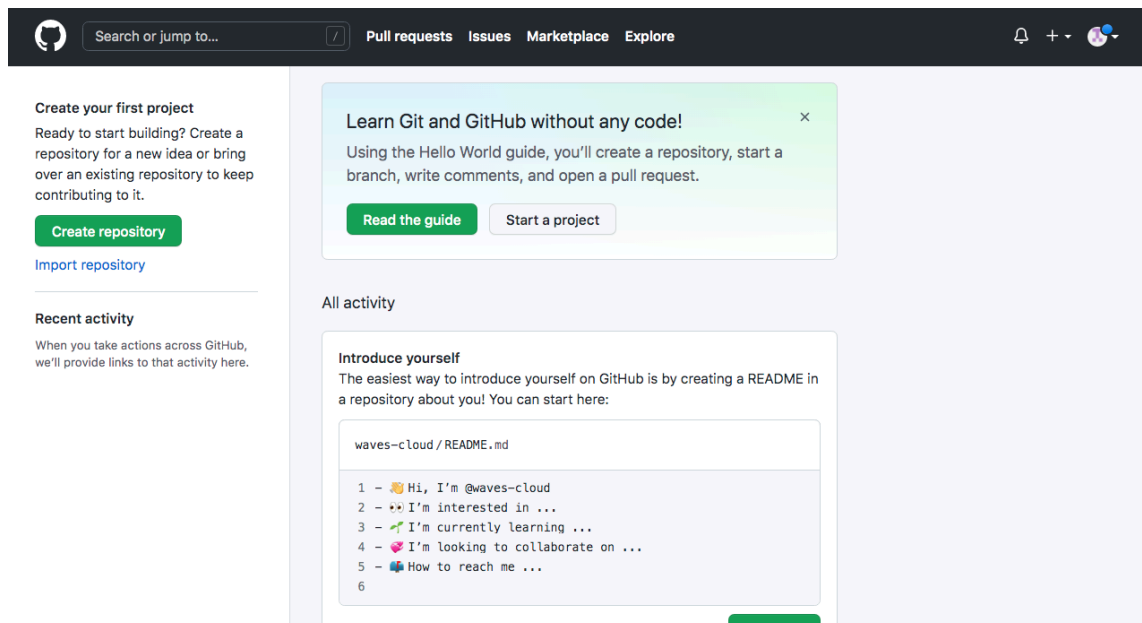
integración y despliegue continuo, administración de de proyectos y finalmente administración de equipos.



- El costo de usar Github es gratuito con las siguientes características: Repositorios públicos ilimitados sin restricción en tamaño del equipo, un repositorio público es visible para todos los usuarios de Github, cualquier persona puede clonarlo pero no pueden escribir en él, es la opción ideal para proyectos de open source o con fines didácticos y/o entrenamiento. Repositorios privados con máximo 4 integrantes esta es nuestra mejor opción para el proyecto, solo es visible para los ingenieros que sean invitados a colaborar y no es visible para nadie más. Por tanto escogemos la opción **Free**.



- Entonces, para empezar con el proyecto.. crearemos nuestro primer repositorio de código, dar click en Botón Create Repository



- o **Configurar repositorio:** En la pantalla de creación de repositorio nos preguntará el nombre de nuestro repositorio, ponemos el nombre **hello-angular**. Nota que el nombre está precedido por el nombre del usuario escogido en los pasos anteriores, por que el nombre de este repositorio seria: **<tu-usuariogit>/hello-angular** y el nombre completo es:

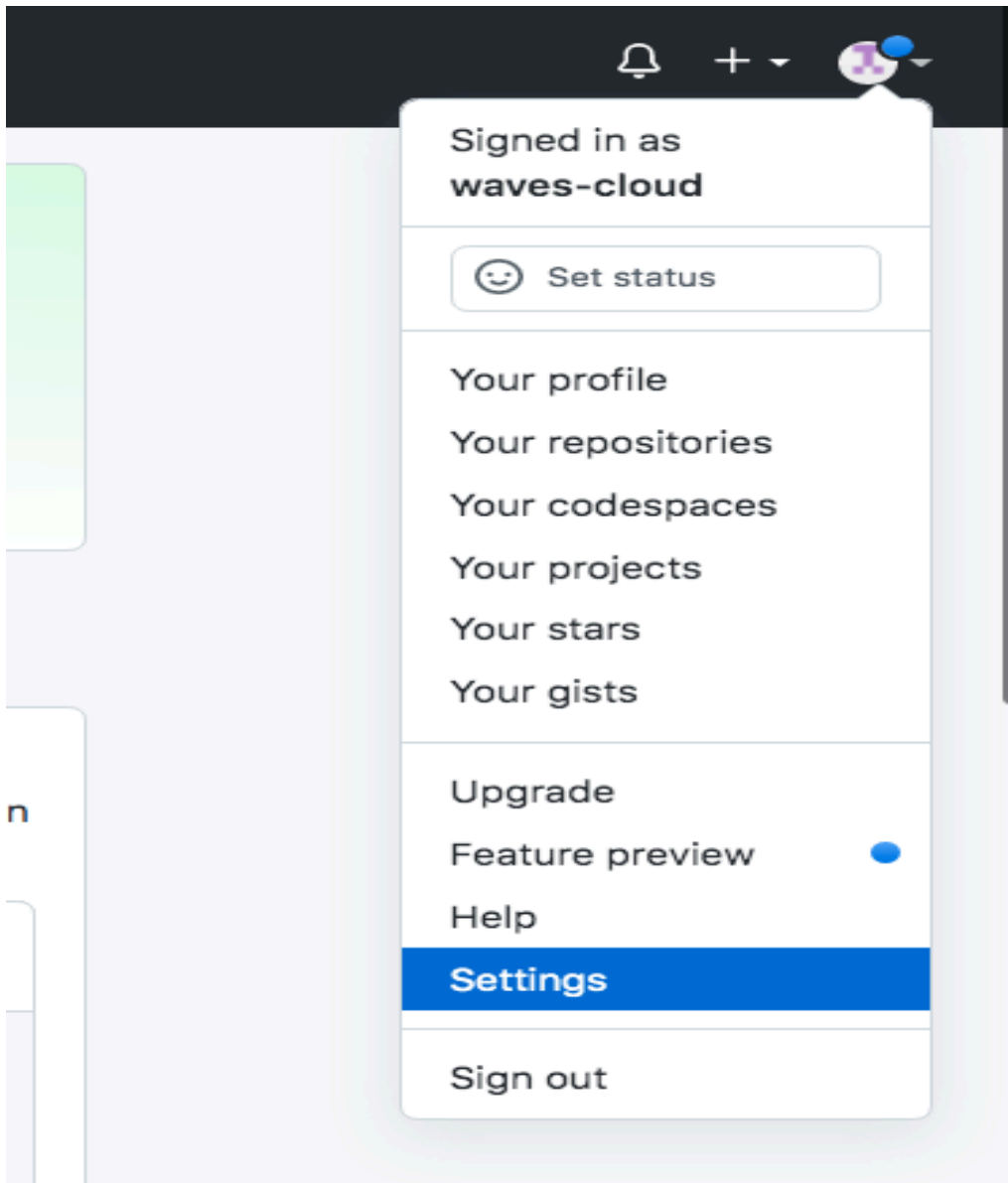
<https://github.com/<tu-usuario>/hello-angular/> ( formado por 3 elementos: <https://github.com/> , nombre usuario y nombre del repositorio.

- o Después nos pide seleccionar si el repositorio es **Público** (visible para todo mundo) o **Privado** (visible para los miembros del equipo)
- o También nos pregunta si deseamos inicializar nuestro repositorio con algunos archivos de configuración como:
  - **README.-** Usado para documentar el propósito de nuestro repositorio, para Angular es importante no crear un README.
  - **.gitignore.-** usado para configurar que archivos no deben ser subidos al repositorio es decir deben ser ignorados, no marcar esta opción, ya que el proyecto de Angular ya contiene un .gitignore
  - **Choose a license.-** Escoger el tipo de licencia que aplicará para nuestro proyecto (MIT license, Apache License, entre otros)
- o Finalmente el repositorio listo para usar se vera asi

### Crear token de github





Crearemos un Token para comprobar que somos los propietarios del repositorio. Ingresamos nuevamente a <https://github.com> con nuestro usuario y contraseña. Seleccionamos el menú de nuestro perfil en la esquina superior derecha y la opción **Settings**.





- Una vez seleccionado **Settings**, ingresar a la opción **Developer Settings**.



## Code, planning, and automation

-  Repositories
-  Packages
-  Pages
-  Saved replies


## Security

-  Code security and analysis

## Integrations

-  Applications
-  Scheduled reminders

## Archives

-  Security log
-  Sponsorship log

## <> Developer settings

- Dentro de la opción de **Developer settings**, ingresar a **Personal access tokens**

[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

### Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

personal token — repo

Last used within the last week Delete

Expires on Wed, Apr 27 2022.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

- En **Personal access tokens**, dar click en **Generate new token**, dentro de la pantalla **New personal access token**, ingresar un Nota o

descripción para nuestro token nuevo, poner el tiempo de expiración de nuestro token, para nuestro ejemplo podemos ponerlo a 90 días, finalmente establecemos los permisos otorgados a nuestro token, para este caso seleccionamos los permisos relacionados a **repo**, y presionamos el botón de **Create Token**.

GitHub Apps

OAuth Apps

Personal access tokens

### New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

personal token

What's this token for?

**Expiration \***

90 days

The token will expire on Fri, Apr 29 2022

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows

- Copiamos en un lugar seguro nuestro token, en un bloc de notas o en cualquier otro editor de texto plano (Ejemplo de token: **ghp\_awRYci6NpAqWVbi0IKfyB6iM8sliKJ3Z514L**)

[Settings](#) / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

## Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_awRYci6NpAqWVbi0IKfyB6iM8sliKJ3Z514L
Delete

personal token — repo
Last used within the last week
Delete

Expires on Wed, Apr 27 2022.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Ahora, debemos instalar el cliente de git en nuestra computadora, así podremos subir nuestros proyectos a la plataforma de github.com

## Para Windows

<https://git-scm.com/downloads/win>

Descargar instalador para 64 bits, instalar con todas las opciones **predeterminadas**.

Este paquete además de instalar el cliente de git, agregara un intérprete de comandos llamado **git bash** el cual usaremos para las prácticas de este curso.

Una vez completada la instalación, buscar la aplicación **git bash** y abrirlo, probar la instalación de git con: `$ git --version`

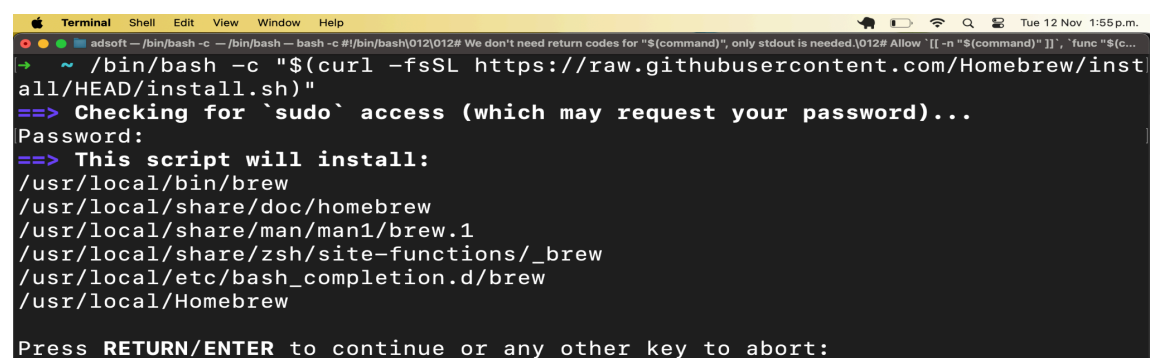
```
→ ~ git --version
git version 2.45.2
```

## Para Mac

Instalar el gestor de paquetes brew ( solo en caso de no tenerlo)

Abrir una terminal de mac y ejecutar el siguiente comando

```
$ /bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

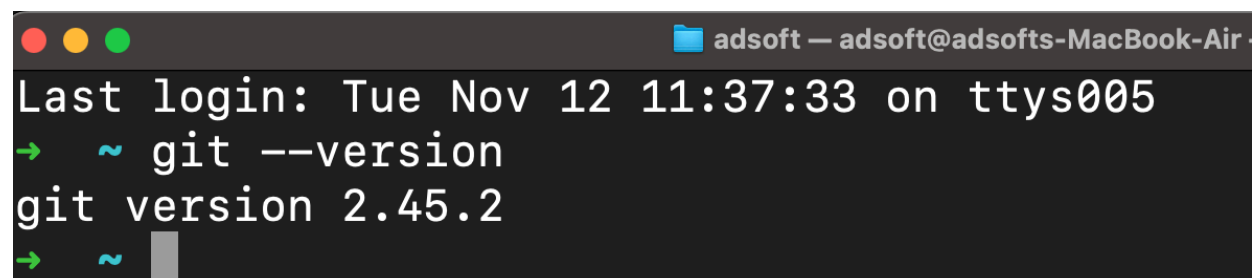


```
Terminal Shell Edit View Window Help
adsoft — /bin/bash — /bin/bash — bash -c #!/bin/bash[012]012# We don't need return codes for "${command}", only stdout is needed.[012]# Allow "[[ -n "${command}" ]]", 'func "${c...
→ ~ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password)...
Password:
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
Press RETURN/ENTER to continue or any other key to abort:
```

Una vez instalado brew, install git con el comando: `$ brew install git`

```
→ ~ brew install git
==> Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE_SECS or disable with
HOMEBREW_NO_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man
brew`).
```

Comprobar la instalación de git con : `$ git --version`



```
adsoft — adsoft@adsofts-MacBook-Air
Last login: Tue Nov 12 11:37:33 on ttys005
→ ~ git --version
git version 2.45.2
→ ~
```

## Para linux

Regularmente los sistemas basados en Linux, ya sea de la familia Debian (ubuntu, mint) o los basados en Redhat (Centos, Rocky, Fedora) ya tienen instalado git, verificar con:

```
$ git --version
```

en caso que no exista instalar con:

```
$ sudo apt-get install git    # debian, ubuntu
$ sudo yum install git       # redhat, centos, fedora
```

**Una vez instalado git, estamos listos para abordar el tema de nvm.**

NVM (Node Version Manager) es una herramienta que permite gestionar múltiples versiones de Node.js en un mismo sistema. Esta herramienta es especialmente útil cuando se trabajan en proyectos que requieren versiones específicas de Node.js o cuando se necesita probar la compatibilidad de una aplicación con diferentes versiones del entorno.

Con NVM, seras capaz de :

- Instalar versiones específicas de Node.js
- Cambiar entre ellas en cualquier momento
- Configurar una versión predeterminada que se utilizará en el sistema
- Mantener un entorno de desarrollo coherente y asegurarte de que todos los miembros del equipo estén utilizando la misma versión de Node.js en cada proyecto

NVM es una herramienta muy útil para desarrolladores y administradores de sistemas que trabajan con Node.js, ya que les permite:

- Ejecutar varios proyectos con versiones diferentes de Node.js
- Asegurarse de que cada proyecto tenga su propia versión de Node.js y sus propias dependencias
- Simplificar el proceso de actualización de Node.js y NPM
- Garantizar la compatibilidad entre proyectos existentes y nuevas versiones de Node.js

En resumen, NVM es una herramienta que te permite administrar y cambiar entre diferentes versiones de Node.js en tu sistema, lo que te brinda flexibilidad y control en tus proyectos de desarrollo.

### **Instalación de NVM**

## Windows

Descargue el instalador de nvm de la siguiente URL:

<https://github.com/coreybutler/nvm-windows/releases/download/1.1.12/nvm-setup.exe>

Instale con las opciones predeterminadas, al finalizar, abra una terminal de git bash, y pruebe nvm con el comando:

\$ nvm --version



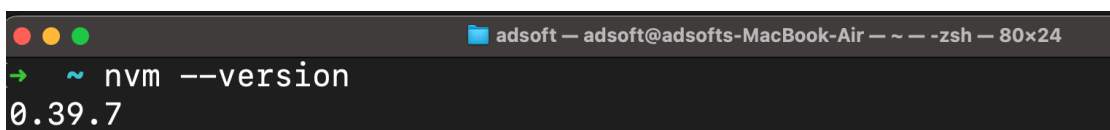
## Mac

Instale nvm con el gestor de paquetes brew, ejecutando el siguiente comando:

\$ brew install nvm

*Probar la instalación con:*

\$ nvm --version



## Linux

abrir una terminal, y ejecutar:

\$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash

verificar la instalación con:

\$ nvm --version

Hasta ahora hemos instalado git como motor de versiones de código y npm como gestor de versiones nodejs. Estas 2 herramientas nos permiten instalar y trabajar con cualquier framework basado en Javascript como Angular, React o Vue. Como dato adicional nodejs instalado y administrado por npm también podría ser usado para crear aplicaciones de Backend como APIS para trabajar con base de datos.

### Sala de inspiración

Piensa en tu empresa u organización y analiza las siguientes preguntas:

- o Cómo almacenas actualmente el código de tus proyectos en la actualidad, como manejas las versiones, cómo colaboras en equipos.
- o Crees que el uso de git podría ayudarte en una mejor administración de los proyectos
- o Si trabajas con distintos proyectos web algunos nuevos, otros solo para agregar módulos nuevos o bien realizar mantenimiento. Consideras que npm podría ayudarte a una mejor administración de los proyectos basados en nodejs.

### Sala de pruebas

Dadas las conclusiones a las que han llegado, y de cómo el uso efectivo de Github y npm puede potencializar la buena administración de un proyecto de software. Responde las siguientes preguntas:

**Pregunta 1:** *Herramienta que permite crear versiones de código fuente.*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b>  <b>git</b>	<b>Correcta:</b>  Git es el motor para administrar versiones de código de los proyectos más usado en el mundo.
<b>Opción b:</b>  <b>npm</b>	<b>Incorrecta:</b>



	Nvm es en gestor de versiones de nodejs
<b>Opción c:</b> nodejs	Incorrecta:  Nodejs es un entorno de de programación para Javascript
<b>Opción d:</b> npm	Incorrecta:  npm es un gestor de paquetes para proyectos basados en javascript

**Pregunta 2:** *Para instalar componentes adicionales de javascript usamos:*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b> nodejs	Incorrecta:  nodejs es un entorno de desarrollo para javascript, pero no este software no permite instalar componentes
<b>Opción b:</b> nvm	Incorrecta:  nvm es un gestor de versiones de nodejs
<b>Opción c:</b> npm	<b>Correcta:</b>  npm (node package manager) es el software que permite instalar paquetes de javascript en un proyecto.
<b>Opción d:</b> angular	Incorrecta:  Angular es un framework para crear aplicaciones web.

**Pregunta 3:** *Permite gestionar distintas versiones de nodejs*

Opciones de respuesta	Retroalimentación
-----------------------	-------------------

<p><b>Opción a:</b></p> <p>npm</p>	<p>Incorrecta:</p> <p>npm es el administrador de paquetes de javascript.</p>
<p><b>Opción b:</b></p> <p>nvm</p>	<p><b>Correcta:</b></p> <p>nvm (node version manager) es el software usado para administrar distintas versiones de nodejs para ser usados en diferentes proyectos.</p>
<p><b>Opción c:</b></p> <p>node</p>	<p>Incorrecta:</p> <p>node es un entorno de desarrollo para javascript.</p>
<p><b>Opción d:</b></p> <p>angular</p>	<p>Incorrecta:</p> <p>Angular es un framework para crear aplicaciones web.</p>

## SUBTEMA 3. Instalación de node 20.0 y Angular 18.0

Afortunadamente todo marcha muy bien hasta ahora, tenemos instalado git como herramienta para versionar código y nvm como nuestra herramienta para gestionar versiones de node js.

En esta sección instalaremos una versión de nodejs estable y compatible con el framework Angular 18, ver tabla de compatibilidad de las versiones de nodejs con Angular

Angular	Node.js	TypeScript	RxJS
18.1.x    18.2.x	^18.19.1    ^20.11.1    ^22.0.0	>=5.4.0 <5.6.0	^6.5.3    ^7.4.0
18.0.x	^18.19.1    ^20.11.1    ^22.0.0	>=5.4.0 <5.5.0	^6.5.3    ^7.4.0
17.3.x	^18.13.0    ^20.9.0	>=5.2.0 <5.5.0	^6.5.3    ^7.4.0
17.1.x    17.2.x	^18.13.0    ^20.9.0	>=5.2.0 <5.4.0	^6.5.3    ^7.4.0
17.0.x	^18.13.0    ^20.9.0	>=5.2.0 <5.3.0	^6.5.3    ^7.4.0
16.1.x    16.2.x	^16.14.0    ^18.10.0	>=4.9.3 <5.2.0	^6.5.3    ^7.4.0
16.0.x	^16.14.0    ^18.10.0	>=4.9.3 <5.1.0	^6.5.3    ^7.4.0

Para nuestras prácticas y proyecto elegiremos un stack tecnológico compuesto por git, nodejs 2.0 y angular 18.0

Para instalar version 20.11.1 de nodejs, usando el comando nvm install, como se muestra a continuación:

```
$ nvm install v20.11.1
```

```
➔ ~ nvm install v20.11.1
```

Now using node v20.11.1 (npm v10.2.4)



Para visualizar las versiones de nodejs instaladas usamos:

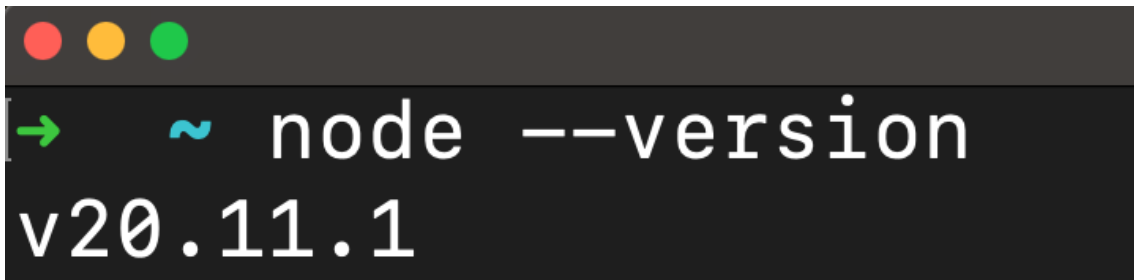
\$ nvm list

```
[→ ~ nvm list
      v10.9.0
      v16.20.2
      v18.0.0
->    v20.11.1
      system
default -> v20.11.1
iojs -> N/A (default)
unstable -> N/A (default)
node -> stable (-> v20.11.1) (default)
stable -> 20.11 (-> v20.11.1) (default)
lts/* -> lts/jod (-> N/A)
lts/argon -> v4.9.1 (-> N/A)
lts/boron -> v6.17.1 (-> N/A)
lts/carbon -> v8.17.0 (-> N/A)
lts/dubnium -> v10.24.1 (-> N/A)
lts/erbium -> v12.22.12 (-> N/A)
lts/fermium -> v14.21.3 (-> N/A)
lts/gallium -> v16.20.2
lts/hydrogen -> v18.20.5 (-> N/A)
lts/iron -> v20.18.0 (-> N/A)
lts/jod -> v22.11.0 (-> N/A)
[→ ~
```

En este caso, tenemos instaladas las versiones v10.9.0, v16.20.2, v18.0.0 y v20.11.1, la version activa es v20.11.1. También muestra la lista de nodejs disponibles para instalar.

Probamos la versión activa de nodejs con:

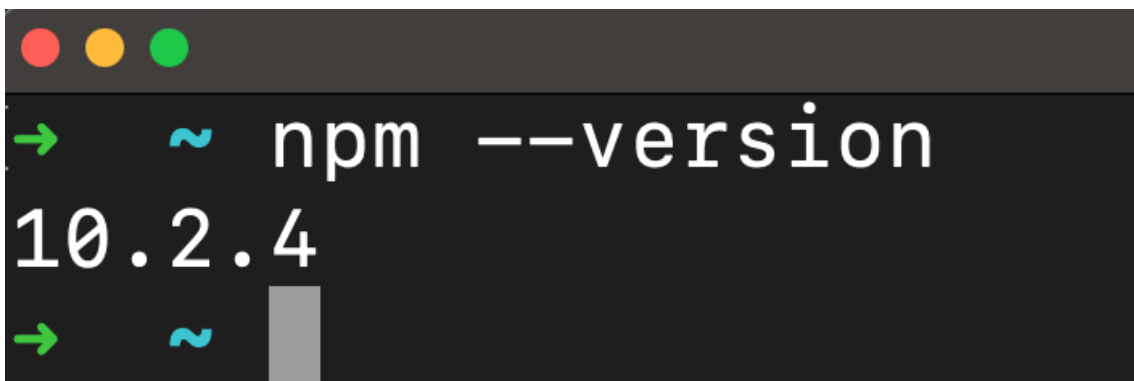
\$ node --version



```
→ ~ node --version
v20.11.1
```

Cada versión de nodejs tiene un instalador de paquetes de Javascript llamado npm (node package manager), para visualizar la versión de npm usamos:

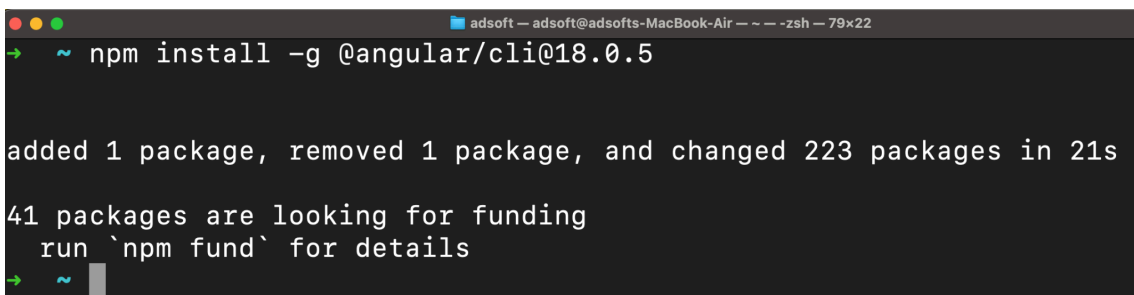
\$ npm --version



```
→ ~ npm --version
10.2.4
→ ~
```

Ahora instalaremos Angular 18, usando el gestor de paquetes npm

\$ npm install -g @angular/cli@18.0.5



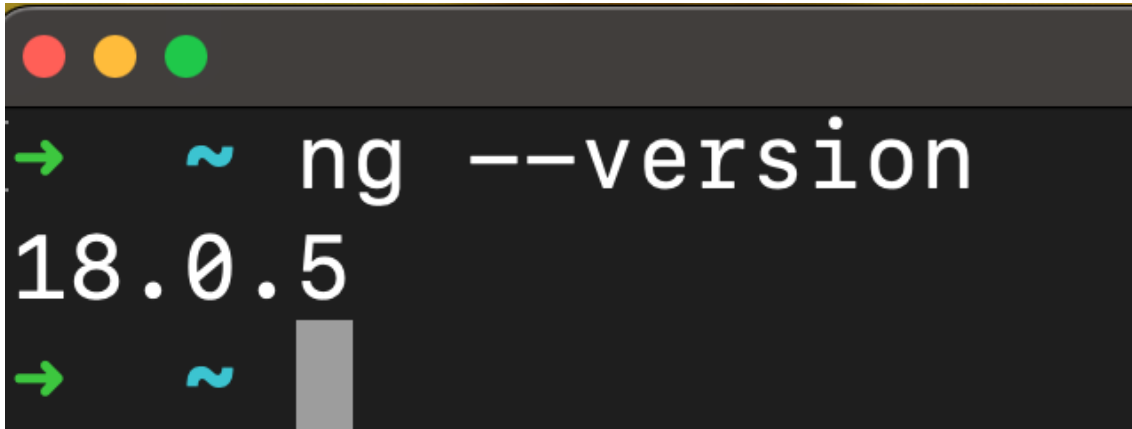
```
adsoft — adsoft@adsofts-MacBook-Air — — — zsh — 79x22
→ ~ npm install -g @angular/cli@18.0.5

added 1 package, removed 1 package, and changed 223 packages in 21s

41 packages are looking for funding
  run `npm fund` for details
→ ~
```

Podemos probar la instalación de angular

\$ ng --version



```
→ ~ ng --version
18.0.5
→ ~
```

Hasta ahora ya tenemos instalado nuestro stack tecnológico formado por git, nodejs 20.11.1 y Angular 18.0, tenemos todo listo para crear nuestro primer proyecto en Angular en el siguiente tema.

### Sala de inspiración

Piensa en diferentes proyectos de software que se puedan desarrollar en tu organización, qué versiones de nodejs y angular usarías para cada uno de ellos, ¿usarías la última versión de nodejs y angular ? en caso afirmativo por qué.

Tip, ten en cuenta que una plataforma desarrollada hace 6 años usando Angular, seguramente usa una versión muy antigua de nodejs 10.0 y Angular 4.0, por tanto si tienes que realizar modificaciones al proyecto deberás instalar las versiones compatibles con su stack tecnológico, por lo que usar nvm para gestionar diferentes versiones de nodejs es indispensable.

### Sala de pruebas

**Pregunta 1:** *Para instalar una nueva versión de nodejs usando nvm usamos:*

Opciones de respuesta	Retroalimentación
-----------------------	-------------------

<b>Opción a:</b> npm install	Incorrecta:  npm install permite instalar paquetes de javascript
<b>Opción b:</b> <b>nvm install</b>	<b>Correcta:</b>  nvm install seguido de la version a instalar, permite instalar una versión nueva de nodejs
<b>Opción c:</b> nvm use	Incorrecta:  nvm use permite poner activa una versión de nodejs ya instalada en el ambiente de desarrollo.
<b>Opción d:</b> nvm list	Incorrecta:  nvm list, muestra la lista de versiones de nodejs instaladas en nuestro equipo.

**Pregunta 2:** Para mostrar las versiones de nodejs instaladas usamos:

Opciones de respuesta	Retroalimentación
<b>Opción a:</b> nvm install	Incorrecta:  nvm install seguido de la version a instalar, permite instalar una versión nueva de nodejs
<b>Opción b:</b> node --version	Incorrecta:  node --version, nos permite consultar la versión de nodejs activa en el ambiente de desarrollo.
<b>Opción c:</b>	<b>Correcta:</b>

<b>nvm list</b>	<b>nvm list, muestra la lista de versiones de nodejs instaladas en nuestro equipo.</b>
<b>Opción d:</b> <b>npm install</b>	Incorrecta:  npm install permite instalar paquetes de javascript

**Pregunta 3:** *Para instalar angular usamos:*

Opciones de respuesta	Retroalimentación
<b>Opción a:</b>  nvm install	Incorrecta:  nvm install seguido de la version a instalar, permite instalar una versión nueva de nodejs
<b>Opción b:</b>  nvm use	Incorrecta:  nvm use permite poner activa una versión de nodejs ya instalada en el ambiente de desarrollo.
<b>Opción c:</b>  npm install	<b>Correcta:</b>  <b>npm install permite instalar paquetes adicionales de javascript, como el caso de Angular es un framework para procesar javascript.</b>
<b>Opción d:</b>  <b>nvm list</b>	Incorrecta:  nvm list, muestra la lista de versiones de nodejs instaladas en nuestro equipo.



### Contenido (NUTRE /SIGNIFICA)-

Hemos estudiado de forma general cómo crear una ambiente de desarrollo para proyectos web, con un stack tecnológico basado en git, nvm y angular. en este momento ya podemos crear un proyecto angular, lo cual se realizará en la siguiente sección.

### Pruébate ( de los 3 temas )

Este es el momento ideal para realizar algunos ejercicios adicionales con nvm y nodejs.

Investiga en <a href="https://nodejs.org/en/download/package-manager">https://nodejs.org/en/download/package-manager</a> cual es la ultima version de nodejs e instálala con nvm
--

Investiga cual es el angular compatible con la última versión de nodejs e instálala con npm install.
--

Usando nvm instala nodejs 18 y su angular compatible.
---

Repite el paso anterior para nodejs 10
--

### Ideas para llevar

Como parte de tu formación en desarrollo web ten en cuenta algunas ideas

- Git es el motor para administrar versiones de código, para guardar en la nube el código podemos usar alguna de las siguientes plataformas: <https://github.com>, <https://bitbucket.org> o <https://gitlab.com>
- Nvm permite instalar diversas versiones de node, cada versión de node permite crear un ambiente de desarrollo para cualquier tecnología como Angular, React, Vue, JQuery entre otros.
- Cada versión de node soporta una versión compatible de Angular

### Referencias.

<https://www.ijraset.com/research-paper/analysis-of-the-font-end-frameworks-and-libraries-in-web-development>

[https://reposit.haw-hamburg.de/bitstream/20.500.12738/8417/1/BA\\_Wohlgethan\\_2176410.pdf](https://reposit.haw-hamburg.de/bitstream/20.500.12738/8417/1/BA_Wohlgethan_2176410.pdf)

[https://en.wikipedia.org/wiki/Angular\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework))

<https://blog.angular.dev/meet-angular-v19-7b29dfd05b84>

<https://blog.angular.dev/angular-v18-is-now-available-e79d5ac0affe>

<https://github.com/angular/angular>

### Material de consulta

<https://git-scm.com/>

<https://github.com/nvm-sh/nvm>

<https://nodejs.org/en>

<https://angular.dev/>