

# Bootcamp Java Developer

Fase 1 - Java Analyst  
Módulo 1

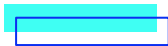


# Descarga e instalación de Git

## Versiones recientes

Si puedes, en general, **es útil instalar Git desde código fuente, porque obtendrás la versión más reciente.**

Cada versión de Git tiende a incluir **mejoras útiles en la interfaz de usuario**. Utilizar la última versión es a menudo el camino más adecuado si te sientes cómodo compilando software desde código fuente.



También ocurre que muchas distribuciones de Linux contienen paquetes muy antiguos; así que a menos que estés en una distribución muy actualizada o estés usando *backports*, instalar desde código fuente puede ser la mejor opción.



# Descarga e Instalación

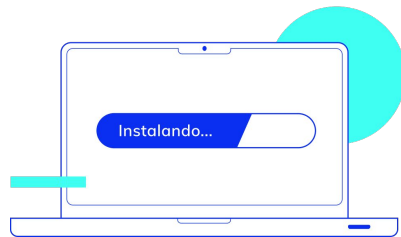
Para realizar la instalación, **necesitas tener las siguientes librerías de las que depende Git: *curl*, *zlib*, *openssl*, *expat* y *libiconv*.**

Para comenzar con la descarga en el sistema **operativo Windows**, de cualquier manera, simplemente basta con descargar el programa instalador ejecutable desde el siguiente link: [gitforwindows.org](https://gitforwindows.org).



**¡Importante!** Puedes encontrar una guía de instalación detallada, con pasos a seguir, y qué opciones elegir, en las descargas del Alumni.

Las instalaciones para otros Sistemas Operativos se encuentran en: [git-scm.com](https://git-scm.com).



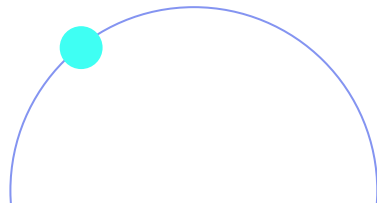
# Configuración inicial



Ahora que tenemos Git instalado en nuestro Sistema Operativo vamos a querer hacer unas cuantas cosas para **personalizar su entorno**.

Estas operaciones deberán realizarse una sola vez por computadora; van a quedar configuradas a través de actualizaciones. También podemos editarlas en cualquier momento con los mismos comandos.

Git viene con una herramienta llamada **git config** que **permite obtener y configurar variables que controlan aspectos de cómo Git se va a ver y cómo va a operar**.

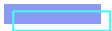
Estas variables pueden ser guardadas en tres lugares distintos que se detallan en la diapositiva a continuación.



1. **/etc/gitconfig:** contiene los valores aplicados a todos los usuarios del Sistema Operativo y todos los repositorios. Si se utiliza la opción **--system** con **git config**, entonces se estará leyendo y escribiendo desde este archivo específicamente. Dado que es un archivo de configuración del sistema, se necesitan privilegios de superusuario o administrador para realizar cambios en él.
  2. **~/.gitconfig ó ~/.config/git/config:** contiene valores específicos del usuario. Puedes hacer que Git lea y escriba sobre este archivo específico usando la opción **--global** y estaremos afectando a todos los repositorios en los trabajemos en el sistema.
  3. **./.git/config:** este archivo puede ser encontrado dentro del mismo directorio del repositorio donde estamos actualmente trabajando y sirve para configurar específicamente un solo repositorio. Podemos forzar Git para que lea y escriba sobre este archivo usando la opción **--local** que, de hecho, es la opción por defecto.
- 
- 

Cada nivel anula los valores del nivel anterior, es decir que los valores en **.git/config** pesan más que aquellos en **/etc/gitconfig**.

En sistemas Windows, Git busca por el archivo **.gitconfig** en el directorio **\$HOME** (**C:\Users\%USER** para la mayoría de las personas). También busca por **/etc/gitconfig**, que es relativo a la raíz de **MSys**, que está donde sea que hayan decidido instalar Git cuando se corre la instalación.



Si estás usando la versión 2.x o superior de Git para Windows, hay también un archivo de configuración de nivel sistema en **C:\Documents and Settings\All Users\Application Data\Git\config** en Windows XP, y **C:\ProgramData\Git\config** en Windows Vista y superior. Este archivo de configuración puede ser cambiado únicamente por **git config -f <archivo>** como administrador.

También podemos **ver todas las configuraciones y dónde se encuentra cada una** con:

```
> git config --list --show-origin
```

Lo primero que tenemos que hacer cuando instalamos Git es **configurar nuestro nombre de usuario y dirección de correo electrónico**. Esto es importante porque cada **commit** de Git usa esta información y es inmutablemente adherida a los **commits** creados.

#### Nombre de usuario:

```
> git config --global user.name "Mi Nombre de Usuario"
```

#### Correo electrónico:

```
> git config --global user.email "mi.direccion.de.email@ejemplo.com"
```





## Configuración adicional

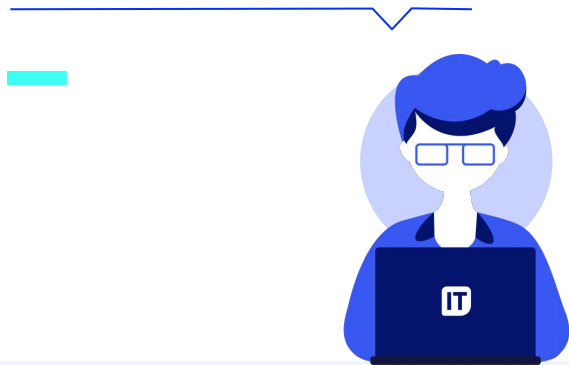
Ahora que ya tenemos la identidad, podemos **configurar el editor que se usará cuando Git necesite que escribas un mensaje**. Si no está configurado, Git usa el editor que venga por defecto en el sistema.

En Windows, si se desea utilizar un editor de texto diferente, tenemos que especificar la ruta completa del archivo ejecutable.

### Configuración del editor de texto:

```
> git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe'"
```

Para más información sobre cómo configurar distintos editores de texto, visita: [git-scm.com](https://git-scm.com).



**Para verificar las configuraciones** que se hayan hecho en el sistema se puede usar el siguiente comando:

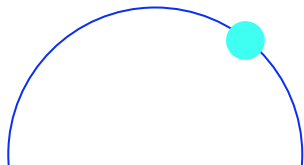
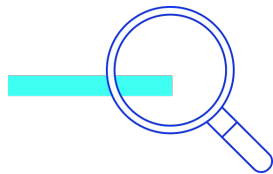
```
> git config --list
```

El mismo va a mostrarnos una lista de todas las variables que estén configuradas y que Git pueda encontrar hasta este punto.

Es probable que se vean variables más de una vez, porque Git lee la misma variable de diferentes archivos. En este caso, Git usa el último valor por cada variable única que vea.

También es posible preguntarle a Git cuál es el valor de una variable en particular con:

```
> git config user.name
```



**¡Sigamos  
trabajando!**