Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones [Git](http://es.wikipedia.org/wiki/Git).

Nota-El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

¿Para que sirve?

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.

¿Qué herramientas proporciona?

En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, caben destacar:

* Una wiki para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

¿Qué uso le daremos?

En nuestra especialidad *"Programación"*, fuimos aprendiendo cosas y creando programas de código abierto, fomentando el software libre; es por eso que presentamos esta gran herramienta enfocada al crecimiento de proyectos comunitariosy libres.

En esta página podremos crear una cuenta gratuita y comenzar a subir repositorios de código (o crearlos desde 0), para que con la ayuda de todos ese proyecto mejore; así como también fortalecer los proyectos de los demás para crecer como grupo.

4.8 Git en el Servidor - GitLab

GitLab

GitWeb es muy simple. Si buscas un servidor Git más moderno, con todas las funciones, tienes algunas soluciones de código abierto que puedes utilizar en su lugar. Puesto que GitLab es una de las más populares, vamos a ver aquí cómo se instala y se usa, a modo de ejemplo. Es algo más complejo que GitWeb y requiere algo más de mantenimiento, pero es una opción con muchas más funciones.

Instalación

GitLab es una aplicación web con base de datos, por lo que su instalación es algo más complicada. Por suerte, es un proceso muy bien documentado y soportado.

Hay algunos métodos que puedes seguir para instalar GitLab. Para tener algo rápidamente, puedes descargar una máquina virtual o un instalador one-click desde <https://bitnami.com/stack/gitlab>, y modificar la configuración para tu caso particular. La pantalla de inicio de Bitnami (a la que se accede con alt-→); te dirá la dirección IP y el usuario y contraseña utilizados para instalar GitLab.

Figure 50. Página de login de la máquina virtual Bitnami.

Para las demás cosas, utiliza como guía los archivos readme de la edición Community de GitLab, que se pueden encontrar en <https://gitlab.com/gitlab-org/gitlab-ce/tree/master>. Aquí encontrarás ayuda para instalar Gitlab usando recetas Chef, una máquina virtual para Digital Ocean, y paquetes RPM y DEB (los cuales, en el momento de escribir esto, aun estaban en beta). También hay guías “no oficiales” para configurar GitLab en sistemas operativos o con bases de datos no estándar, un script de instalación completamente manual y otros muchos temas.

Administración

La interfaz de administración de GitLab se accede mediante la web. Simplemente abre en tu navegador la IP o el nombre de máquina donde has instalado Gitlab, y entra con el usuario administrador. El usuario predeterminado es admin@local.host, con la contraseña 5iveL!fe (que te pedirá cambiar cuando entres por primera vez). Una vez dentro, pulsa en el icono “Admin area” del menú superior derecho.

Figure 51. El icono “Admin area” del menú de GitLab.

Usuarios

Los usuarios en Gitlab son las cuentas que abre la gente. Las cuentas de usuario no tienen ninguna complicación: viene a ser una colección de información personal unida a la información de login. Cada cuenta tiene un espacio de nombres (namespace) que es una agrupación lógica de los proyectos que pertenecen al usuario. De este modo, si el usuario jane tiene un proyecto llamado project, la URL de ese proyecto sería <http://server/jane/project>.

Figure 52. Pantalla de administración de usuarios en GitLab.

Tenemos dos formas de borrar usuarios. “Bloquear” un usuario evita que el usuario entre en Gitlab, pero los datos de su espacio de nombres se conservan, y los commits realizados por el usuario seguirán a su nombre y relacionados con su perfil.

“Destruir” un usuario, por su parte, borra completamente al usuario de la base de datos y el sistema de archivos. Todos los proyectos y datos de su espacio de nombres se perderán, así como cualquier grupo que le pertenezca. Esto es, por supuesto, la acción más permanente, destructiva y casi nunca se usa.

Grupos

Un grupo de GitLab es un conjunto de proyectos, junto con los datos acerca de los usuarios que tienen acceso. Cada grupo tiene también un espacio de nombres específico (al igual que los usuarios). Por ejemplo, si el grupo formacion tuviese un proyecto materiales su URL sería:<http://server/formacion/materiales>.

Figure 53. Pantalla de administración de grupos en GitLab.

Cada grupo se asocia con un conjunto de usuarios, donde cada usuario tiene un nivel de permisos sobre los proyectos así como el propio grupo. Estos permisos van desde el de “Invitado” (que solo permite manejar incidencias y chat) hasta el de “Propietario” (con control absoluto del grupo, sus miembros y sus proyectos). Los tipos de permisos son muy numerosos para detallarlos aquí, pero en la ayuda de la pantalla de administración de GitLab los encontraremos fácilmente.

Proyectos

Un proyecto en GitLab corresponde con un repositorio Git. Cada proyecto pertenece a un espacio de nombres, bien sea de usuario o de grupo. Si el proyecto pertenece a un usuario, el propietario del mismo tendrá control directo sobre quién tiene acceso al proyecto; si el proyecto pertenece a un grupo, los permisos de acceso por parte de los usuarios estarán también determinados por los niveles de acceso de los miembros del grupo.

Cada proyecto tiene también un nivel de visibilidad, que controla quién tiene acceso de lectura a las páginas del proyecto y al propio repositorio. Si un proyecto es Privado, el propietario debe conceder los accesos para que determinados usuarios tengan permisos. Un proyecto Interno es visible a cualquier usuario identificado, y un proyecto Público es visible a todos, incluso usuarios no identificados y visitantes. Observa que esto controla también el acceso de lectura git (“fetch”) así como el acceso a la página web del proyecto.

Enganches (hooks)

GitLab tiene soporte para los enganches (hooks), tanto a nivel de proyecto como del sistema. Para cualquiera de ellos, el servidor GitLab realizará una petición HTTP POST con determinados datos JSON cuando ocurran ciertos eventos. Es una manera interesante de conectar los repositorios y la instancia de GitLab con el resto de los mecanismos automáticos de desarrollo, como servidores de integración continua (CI), salas de charla y otras utilidades de despliegue.

Uso básico

Lo primero que tienes que hacer en GitLab es crear un nuevo proyecto. Esto lo consigues pulsando el icono “+” en la barra superior. Te preguntará por el nombre del proyecto, el espacio de nombres al que pertenece y qué nivel de visibilidad debe tener. Esta información, en su mayoría, no es fija y puedes cambiarla más tarde en la pantalla de ajustes. Pulsa en “Create Project” y habrás terminado.

Una vez que tengas el proyecto, querrás usarlo para un repositorio local de Git. Cada proyecto se puede acceder por HTTPS o SSH, protocolos que podemos configurar en nuestro repositorio como un Git remoto. La URL la encontrarás al principio de la página principal del proyecto. Para un repositorio local existente, puedes crear un remoto llamado gitlab del siguiente modo:

$ git remote add gitlab https://server/namespace/project.git

Si no tienes copia local del repositorio, puedes hacer esto:

$ git clone https://server/namespace/project.git

La interfaz web te permite acceder a diferentes vistas interesantes del repositorio. Además, la página principal del proyecto muestra la actividad reciente, así como enlaces que permiten acceder a los archivos del proyecto y a los diferentes commits.

Trabajando con GitLab

Para trabajar en un proyecto GitLab lo más simple es tener acceso de escritura (push) sobre el repositorio git. Puedes añadir usuarios al proyecto en la sección “Members” de los ajustes del mismo, y asociar el usuario con un nivel de acceso (los niveles los hemos visto en [Grupos](https://git-scm.com/book/es/v2/ch00/r_gitlab_groups_section)). Cualquier nivel de acceso tipo “Developer” o superior, permite al usuario enviar commits y ramas sin ninguna limitación.

Otra forma de colaboración, más desacoplada, es mediante las peticiones de fusión (merge requests). Esta característica permite a cualquier usuario con acceso de lectura, participar de manera controlada. Los usuarios con acceso directo pueden, simplemente, crear la rama, enviar commits y luego abrir una petición de fusión desde su rama hacia la rama master u otra cualquiera. Los usuarios sin permiso de push pueden hacer un “fork” (es decir, su propia copia del repositorio), enviar sus cambios a esa copia, y abrir una petición de fusión desde su fork hacia el proyecto del que partió. Este modelo permite al propietario tener un control total de lo que entra en el repositorio, permitiendo a su vez la cooperación de usuarios a los que no se confía el acceso total.

Las peticiones de fusión y las incidencias (issues) son las principales fuentes de discusión en los proyectos de GitLab. Cada petición de fusión permite una discusión sobre el cambio propuesto (similar a una revisión de código), así como un hilo de discusión general. Ambas pueden asignarse a usuarios, o ser organizadas en hitos (milestones).

Esta sección se ha enfocado principalmente hacia las características de GitLab relacionadas con Git, pero como proyecto ya maduro, tiene muchas otras características para ayudar en la coordinación de grupos de trabajo, como wikis de proyecto y utilidades de mantenimiento. Una ventaja de GitLab es que, una vez que el servidor está configurado y funcionando, rara vez tendrás que tocar un archivo de configuración o acceder al servidor mediante SSH; casi toda la administración y uso se realizará mediante el navegador web.

COMO SE DIVIDE LOGISTICAMENTE UN DISCO DURO

Siempre que se enciende el [computador](https://www.monografias.com/trabajos15/computadoras/computadoras.shtml), los discos sobre los que se almacenan los [datos](https://www.monografias.com/trabajos11/basda/basda.shtml) giran a una gran [velocidad](https://www.monografias.com/trabajos13/cinemat/cinemat2.shtml#TEORICO) (a menos que disminuyan su [potencia](https://www.monografias.com/trabajos14/trmnpot/trmnpot.shtml) para ahorrar [electricidad](https://www.monografias.com/trabajos10/nofu/nofu.shtml)).

Los [discos duros](https://www.monografias.com/trabajos37/discos-duros/discos-duros.shtml) de hoy, con capacidad de almacenar multigigabytes mantienen el mínimo principio de una cabeza de [Lectura](https://www.monografias.com/trabajos14/textos-escrit/textos-escrit.shtml)/[Escritura](https://www.monografias.com/trabajos16/metodo-lecto-escritura/metodo-lecto-escritura.shtml) suspendida sobre una superficie magnética que gira velozmente con precisión microscópica.

Pero hay un aspecto de los discos duros que probablemente permanecerá igual. A diferencia de otros componentes de la PC que obedecen a los [comandos](https://www.monografias.com/trabajos7/coman/coman.shtml) del [software](https://www.monografias.com/Computacion/Software/), el [disco duro](https://www.monografias.com/trabajos14/discosduros/discosduros.shtml) hace ruidos cuando emprende su [trabajo](https://www.monografias.com/trabajos34/el-trabajo/el-trabajo.shtml). Estos ruidos son recordatorio de que es uno de los pocos componentes de una PC que tiene carácter mecánico y electrónico al mismo tiempo

Los discos duros pertenecen a la llamada [memoria](https://www.monografias.com/trabajos13/memor/memor.shtml) secundaria o [almacenamiento](https://www.monografias.com/trabajos12/dispalm/dispalm.shtml) secundario. Al disco duro se le conoce con gran cantidad de denominaciones como disco duro, rígido (frente a los discos flexibles o por su fabricación a base de una capa rígida de [aluminio](https://www.monografias.com/trabajos13/tramat/tramat.shtml#ALUMIN)), fijo (por su situación en el ordenador de manera permanente). Estas denominaciones aunque son las habituales no son exactas ya que existen discos de iguales [prestaciones](https://www.monografias.com/trabajos15/cumplimiento-defectuoso/cumplimiento-defectuoso.shtml" \l "INCUMPL)pero son flexibles, o bien removibles o transportables, u otras [marcas](https://www.monografias.com/trabajos16/marca/marca.shtml) diferentes fabricantes de cabezas.

Las capacidades de los discos duros varían desde 10 Mb. hasta varios Gb. en minis y grandes ordenadores. Para conectar un disco duro a un ordenador es necesario disponer de una tarjeta controladora. La velocidad de acceso depende en gran parte de la tecnología del propio disco duro y de la tarjeta controladora asociada al discos duro.

Estos están compuestos por varios platos, es decir varios discos de material magnético montados sobre un eje central sobre el que se mueven. Para leer y escribir datos en estos platos se usan las cabezas de lectura/escritura que mediante un [proceso](https://www.monografias.com/trabajos14/administ-procesos/administ-procesos.shtml#PROCE) electromagnético codifican / decodifican la información que han de leer o escribir. La cabeza de lectura/escritura en un disco duro está muy cerca de la superficie, de forma que casi vuela sobre ella, sobre el colchón de [aire](https://www.monografias.com/trabajos/aire/aire.shtml) formado por su propio [movimiento](https://www.monografias.com/trabajos15/kinesiologia-biomecanica/kinesiologia-biomecanica.shtml). Debido a esto, están cerrados herméticamente, porque cualquier partícula de polvo puede dañarlos.

Unidad de disco duro:

Los discos duros se presentan recubiertos de una capa magnética delgada, habitualmente de óxido de [hierro](https://www.monografias.com/trabajos/metalprehis/metalprehis.shtml), y se dividen en unos círculos concéntricos cilindros (coincidentes con las pistas de los disquetes), que empiezan en la parte exterior del disco (primer cilindro) y terminan en la parte interior (último). Asimismo estos cilindros se dividen en sectores, cuyo número esta determinado por el tipo de disco y su formato, siendo todos ellos de un tamaño fijo en cualquier disco. Cilindros como sectores se identifican con una serie de números que se les asignan, empezando por el 1, pues el numero 0 de cada cilindro se reserva para propósitos de identificación mas que para almacenamiento de datos. Estos, escritos/leídos en el disco, deben ajustarse al tamaño fijado del almacenamiento de los sectores. Habitualmente, los [sistemas](https://www.monografias.com/trabajos11/teosis/teosis.shtml) de disco duro contienen más de una unidad en su interior, por lo que el número de caras puede ser más de 2. Estas se identifican con un número, siendo el 0 para la primera. En general su organización es igual a los disquetes. La capacidad del disco resulta de multiplicar el número de caras por el de pistas por cara y por el de sectores por pista, al total por el número de bytes por sector.

Para escribir, la cabeza se sitúa sobre la celda a grabar y se hace pasar por ella un pulso de corriente, lo cual crea un campo magnético en la superficie. Dependiendo del sentido de la corriente, así será la polaridad de la celda. ara leer, se mide la corriente inducida por el campo magnético de la celda. Es decir que al pasar sobre una zona detectará un campo magnético que según se encuentre magnetizada en un sentido u otro, indicará si en esa posición hay almacenado un 0 o un 1. En el caso de la escritura el proceso es el inverso, la cabeza recibe una corriente que provoca un campo magnético, el cual pone la posición sobre la que se encuentre la cabeza en 0 o en 1 dependiendo del [valor](https://www.monografias.com/trabajos14/nuevmicro/nuevmicro.shtml) del campo magnético provocado por dicha corriente.