



ENTORNO DE DESARROLLO

CONTROL DE VERSIONES



Enero 2019

Klever Uyana Barahona



CONTROL DE VERSIONES

Indice

1 ¿Que es el control de versiones?	3
2 Tipos de control de versiones.	4
Segun almacenamiento.	4
Segun la forma de realizar los cambios.	5
3 Ventajas del control de versiones	6
SCV mas utilizados.	6
Ejemplo de codigo libre en GIT	7
4 ¿QUE ES GIT?	8
GITHUB vs GITLAB	9
¿Que es GITHUB?	10
¿Que es GITLAD?	11
Contenido interesante.	12
Crear cuenta en Github	13
Crear repositorio en Github	16

CONTROL DE VERSIONES

¿QUE ES EL CONTROL DE VERSIONES?

- Herramienta de integración continua.
- Garantizar gestión de cambios a creadores.
- Garantizar gestión de cambios a creadores.
- Se llama control de versiones a la gestión de los cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.
- Facilita la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas.

Un CVS debe proporcionar.

- Mecanismos de almacenamiento
- Posibilidad de realizar cambios
- Registro historico de acciones
- Generación de informes
- Marcado de versiones

REPOSITORIOS

Conjunto de información gestionada por un CVS.
Contiene al historial de versiones.

CHECKOUT

Copia del repositorio que se hace un usuario.

CHECKIN/COMMIT

Subida de cambios al repositorio.

CONTROL DE VERSIONES

TIPOS DE SCV

Según almacenamiento.

Centralizados.

Repositorio de un único servidor.

Fácil de administrar ya que solo el perfil de administrador se realiza por un usuario o un grupo reducido.

Solo hay que estar pendiente de una base de datos.

Supone dificultad a la hora de trabajar y modificar en paralelo.(Muy rígidos)



Distribuidos.

Repositorio por usuario sin decisiones centralizadas.

Difícil de gestionar y administración al trabajar con diversos usuarios.(Conflicto entre versiones)

Para trabajar en equipo se hace compartiendo los repositorios.

Supone facilidad a la hora de trabajar y modificar en paralelo.



CONTROL DE VERSIONES

TIPOS DE SCV

Según la forma de realizar los
cambios.

ESCLUSIVOS.

- Se marca para su modificación una parte del proyecto para que un único usuario pueda trabajar sobre ella.
- Una vez modificado se sube al repositorio, y se marca que ya no esta bloqueada, así el resto de usuarios pueden trabajar sobre ellas.



COLABORATIVOS.

- Varios personas pueden modificar a la vez en una parte del proyecto.
- Conforme se suben las modificaciones el sistema se debe de encargar de mezclar las diferentes modificaciones, y hacer una nueva version.
- Puede genera una gran cantidad de conflictos si trabajar una gran cantidad de usuarios en una parte especifica del proyecto.



CONTROL DE VERSIONES

VENTAJAS DE SCV

Ventajas de usar control de versiones

Control exacto sobre:

Ultima versión.

Quién la subió.

Cuándo la subió.

Permite comparar versiones.

Crear ramas del proyecto, para otras proyectos parecidos.

Regresar a versiones anteriores.

Probar nuevas soluciones.

SCV MAS UTILIZADOS

¿Cuales son los SCV mas utilizados?



Concurrent Version System
fue el primer SCV en desarrollarse
, muy utilizado en los inicios.



Muy popular y utilizado en entornos
empresariales.

Centralizados

CONTROL DE VERSIONES

¿Cuales son los SCV mas utilizados?



Los primeros distribuidos fueron Mercurial y Monotone, gran auge en el desarrollo de software libre.

Evolucionó



Actualmente es el mas popular.
Con gran éxito en la comunidad de código libre.
Red social.
Muchas empresas se estan parando a git.

Distribuidos

Ejemplo de codigo libre en



Popcorn Time es una aplicación de código abierto desarrollada por la comunidad git cuya función es la reproducción de medios mediante P2P, es decir, de usuario a usuario.(Ver torrents en streaming)

Popcorn Time echó el cierre a mediados de 2015, concretamente en octubre, para evitar problemas legales con la justicia. Desde entonces han salido versiones no oficiales, algunas de ellas incluso con malware incluido. Gran parte de los desarrolladores de Popcorn Time abandonó el barco tras la presión legal. El resto, tal como vemos ahora, han continuado. La vuelta de Popcorn Time se debe a que estos desarrolladores que sobrevivieron están tratando de jugársela a la MPAA, quien les obligó a cerrar.

Y es que están tratando de burlarles, pues ahora se centrarán en el desarrollo de Butter, que viene a ser lo mismo que Popcorn Time pero de manera legal. De este modo, Popcorn Time se servirá del código fuente de Butter, y todo será mucho más legal.

CONTROL DE VERSIONES



¿QUE ES GIT?

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT.

Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena.

Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

En cuanto a derechos de autor Git es un software libre distribuible bajo los términos de la versión 2 de la Licencia Pública General de GNU.

CURSOS DE GIT



<https://www.youtube.com/watch?v=jSJ8xhKtfP4&list=PLTd5ehIj0goMCnj6V5NdzSIHBgrIXckGU>



<https://codigofacilito.com/cursos/git>

LIBRO



<https://git-scm.com/book/es/v2>

CONTROL DE VERSIONES



VS.



CONTROL DE VERSIONES

¿Que es GITHUB ?

GitHub es un sitio web y un servicio en la nube que ayuda a los desarrolladores a almacenar y administrar su código, al igual que llevar un registro y control de cualquier cambio sobre este código.

Compañía sin fines de lucro que ofrece un servicio de hosting de repositorios almacenados en la nube. Esencialmente, hace que sea más fácil para individuos y equipos usar Git como la versión de control y colaboración.

GitHub es tan fácil de usar, que incluso algunas personas usan GitHub para administrar otro tipo de proyectos – como escribir libros , diseño grafico entre otros.



Fundada:2008

GitHub hace dinero vendiendo alojamiento para repositorios de código privado, al igual que otros planes enfocados para negocios, que hace más fácil que las organizaciones administren a los miembros de su equipo y su seguridad.

Cualquier persona puede inscribirse y ser hospedar un repositorio de código público completamente gratuito, el cual hace que GitHub sea especialmente popular con proyectos de fuente abierta.

GitHub como red social para desarrolladores.

GitHub

CONTROL DE VERSIONES

¿Que es Gitlad?

GitLab es como GitHub pero en esteroides. Es un servicio que también ofrece alojamiento de repositorios con varias funciones de seguimientos de problemas, pero además tiene características extra.

Inició como un proyecto en GitHub. A diferencia de este último, en GitLab usan una aplicación única creada desde cero para dar soporte al ciclo entero de desarrollo, en lugar de integrar múltiples herramientas diferentes.



Fundada:2011

También tiene un plan básico gratuito que permite aprovechar la plataforma para construir y ejecutar tus aplicaciones. Tienes planes de pago Gold y Ultimate a los nuevos usuarios.

GitLab tampoco es ningún desconocido, múltiples empresas y marcas conocidas lo usan. Y recientemente Tras la compra de GITHUB por parte de Microsoft, en concreto el 4 de junio de 2018, por la cantidad de 7.500 millones de dólares, numerosos proyectos se mudaron a la plataforma ademas porque buscaban simplificar el proceso de colaboración entre sus múltiples contribuidores.

GitLab no es un simple clon de GitHub.



GitLab

CONTROL DE VERSIONES

La gente huye a Gitlad tras la compra de GitHub por
parte de Microsoft
¿Que significa para los programadores?



<https://www.youtube.com/watch?v=eR-A3Po9NqM>

PRESENTACION PREZI



<https://prezi.com/view/52DQFnzkiyQJNtE68C1w/>

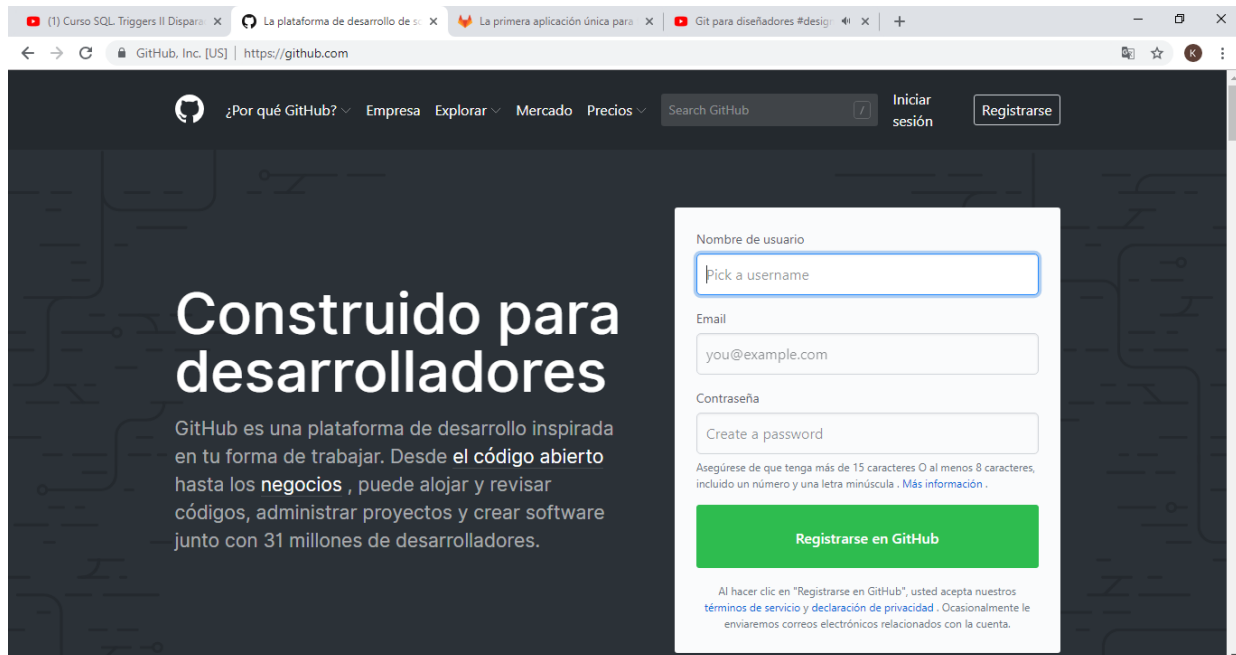
CONTROL DE VERSIONES

Crear cuenta en Github

Para la crear una cuenta en Github tenemos que tener ingresamos en su pagina.

<https://github.com/>

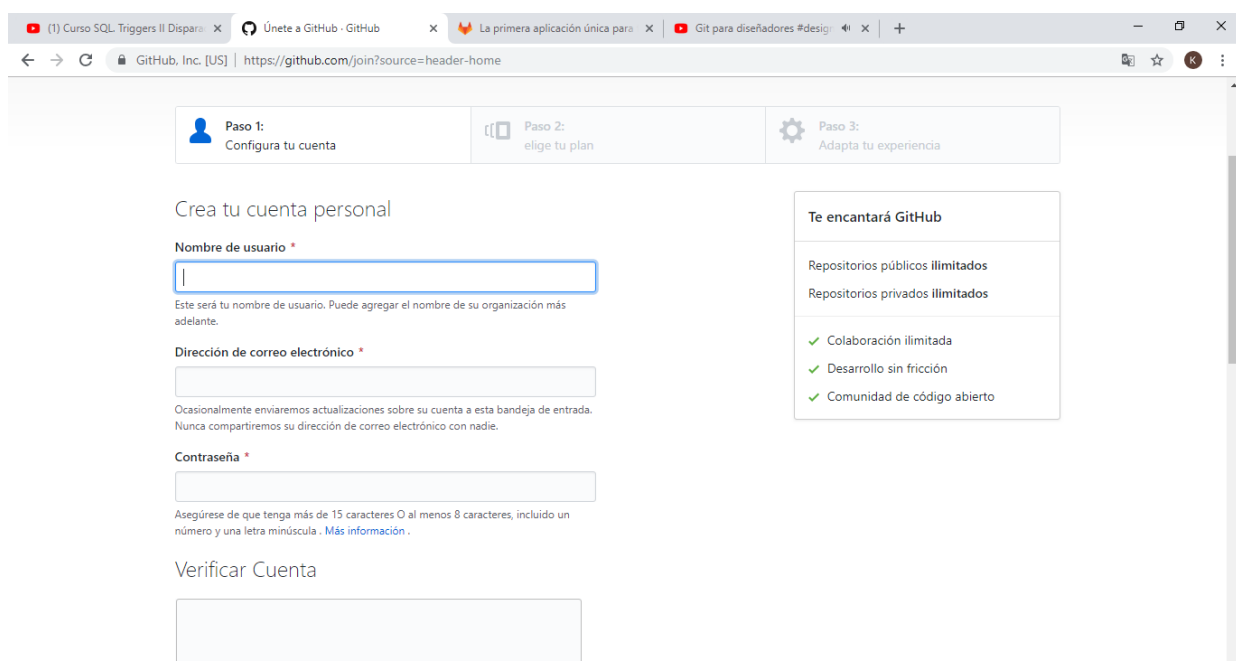
Le damos a registrarse.



The screenshot shows the GitHub homepage with a dark background and a white registration form on the right. The form has the following fields and elements:

- Nombre de usuario:** A text input field with the placeholder "Pick a username".
- Email:** A text input field with the placeholder "you@example.com".
- Contraseña:** A text input field with the placeholder "Create a password".
- Registration Instructions:** Below the password field, it says "Asegúrese de que tenga más de 15 caracteres O al menos 8 caracteres, incluido un número y una letra minúscula. [Más información](#)."
- Registrarse en GitHub:** A large green button.
- Footer:** At the bottom, it says "Al hacer clic en 'Registrarse en GitHub', usted acepta nuestros [términos de servicio](#) y [declaración de privacidad](#). Ocasionamente le enviaremos correos electrónicos relacionados con la cuenta."

Seguimos los pasos para la creación de la cuenta .



The screenshot shows the GitHub account creation process with three steps:

- Paso 1: Configura tu cuenta**
- Paso 2: elige tu plan**
- Paso 3: Adapta tu experiencia**

The first step, "Crea tu cuenta personal", includes the following fields and elements:

- Nombre de usuario ***: A text input field.
- Dirección de correo electrónico ***: A text input field.
- Contraseña ***: A text input field.
- Verificar Cuenta**: A text input field.

On the right side, there is a box titled "Te encantará GitHub" with the following content:

- Repositorios públicos ilimitados
- Repositorios privados ilimitados
- ✓ Colaboración ilimitada
- ✓ Desarrollo sin fricción
- ✓ Comunidad de código abierto

CONTROL DE VERSIONES

Le damos a verificar y resolvemos el captcha.

Triggers II Dispara x Únete a GitHub · GitHub x La primera aplicación única para

GitHub, Inc. [US] | https://github.com/join?source=header-home

Contraseña *

Asegúrese de que tenga más de 15 caracteres O al menos 8 caracteres, incluido un número y una letra minúscula. [Más información](#).

Verificar Cuenta

Por favor, resuelve este puzle para que sepamos que eres una persona real

Verificar

Al hacer clic en "Crear una cuenta" a continuación, acepta nuestros [términos de servicio](#) y [declaración de privacidad](#). Ocasionalmente le enviaremos correos electrónicos relacionados con la cuenta.

Triggers II Dispara x Únete a GitHub · GitHub x La primera aplicación única para

GitHub, Inc. [US] | https://github.com/join?source=header-home

Contraseña *

Asegúrese de que tenga más de 15 caracteres O al menos 8 caracteres, incluido un número y una letra minúscula. [Más información](#).

Verificar Cuenta

Cuando la imagen, **esté bien orientada**, pulsa ¡Hecho!

Hecho

Al hacer clic en "Crear una cuenta" a continuación, acepta nuestros [términos de servicio](#) y [declaración de privacidad](#). Ocasionalmente le enviaremos correos electrónicos relacionados con la cuenta.

Una ves resuelto creamos la cuenta .

Triggers II Dispara x Únete a GitHub · GitHub x La primera aplicación única para

GitHub, Inc. [US] | https://github.com/join?source=header-home

Verificar Cuenta

Al hacer clic en "Crear una cuenta" a continuación, acepta nuestros [términos de servicio](#) y [declaración de privacidad](#). Ocasionalmente le enviaremos correos electrónicos relacionados con la cuenta.

Crea una cuenta

Seleccionamos el plan gratuito

Triggers II Dispara x Únete a GitHub · Donde se construye el x La primera aplicación única para x Git para diseñador

GitHub, Inc. [US] | https://github.com/join/plan

<p>✓</p> <p>Gratis</p> <p>Los fundamentos de GitHub para cada desarrollador.</p> <p>\$ 0 por mes</p> <p>Incluye:</p> <ul style="list-style-type: none">Repos Repositorios públicos y privados ilimitados.✓ 3 colaboradores para repositorios privados.✓ Problemas y seguimiento de errores.✓ Gestión de proyectos. <p>Eres estudiante ? Obtén acceso a las mejores herramientas de desarrollo de forma gratuita con el paquete de desarrollo para estudiantes de GitHub.</p>	<p>○</p> <p>Pro</p> <p>Pro herramientas para desarrolladores con requisitos avanzados.</p> <p>\$ 7 por mes (ver en EUR)</p> <p>Incluye:</p> <ul style="list-style-type: none">Repos Repositorios públicos y privados ilimitados.Collabor colaboradores ilimitados✓ Problemas y seguimiento de errores.✓ Gestión de proyectos.✓ Herramientas avanzadas y conocimientos
--	--

CONTROL DE VERSIONES

Completado

Configurar una cuenta personal

Paso 2:

elige tu plan

Paso 3:

Adapta tu experiencia

¿Cómo describirías tu nivel de experiencia en programación?

☐ Muy experimentado

☐ Algo experimentado

☒ Totalmente nuevo para la programación.

¿Para qué planeas usar GitHub? (marque todo lo que corresponda)

☒ Proyectos escolares

☒ Diseño

☐ Investigación

☐ Desarrollo

☒ Gestión de proyectos

☐ Otros (especificar)

¿Cuál es el más cercano a cómo te describirías?

☐ Soy un profesional

☐ Soy un aficionado

☒ soy un estudiante

☐ Otros (especificar)

¿En que estas interesado?

aprendizaje

x

Por ejemplo, tutoriales, android, ruby, desarrollo web, aprendizaje automático, código abierto.

☐ Otros (especificar)

¿En que estas interesado?

aprendizaje

x

Por ejemplo, tutoriales, android, ruby, desarrollo web, aprendizaje automático, código abierto.

Enviar

omitir este paso

Por ultimo rellenamos el pequeño cuestionario y le damos a **enviar**

Repositorios

Tus repositorios más activos aparecerán aquí.

Crear un repositorio o explorar repositorios .

¡Aprende Git y GitHub sin ningún código!

Con la guía Hello World, creará un repositorio, iniciará una rama, escribirá comentarios y abrirá una solicitud de extracción.

Leer la guía

Iniciar un proyecto

Bienvenido al nuevo panel de control.

Acércate a las cosas que más te interesan.

Descubrir repositorios

Tránsito masivo

Distributed Application Framework para .NET

C # 1.9k ★

Fourdee / DietPi

Justicia ligera para tu computadora de una sola placa.

Shell 1.3k ★

DroidKaigi / conference-app-2019

La aplicación oficial de la conferencia para DroidKaigi 2019 Tokio

Kotlin 479 ★

Ir a Explorar →

CONTROL DE VERSIONES

Crear repositorio en Github


Para crear un repositorio la damos en **iniciar un proyecto** desde la pagina principal de nuestra cuenta.

Le damos un nombre (al ser una cuenta gratuita el repositorio solo puede ser publico)

Crear un nuevo repositorio

Un repositorio contiene todos los archivos para su proyecto, incluido el historial de revisiones.

Propietario

 klipsmf4 ▾

Nombre del repositorio *

Los grandes nombres de los repositorios son cortos y memorables. ¿Necesitas inspiración? ¿Qué hay de **probable-octo-risita** ?

Descripción (opcional)

☒  **Público**

Cualquier persona puede ver este repositorio. Tú eliges quién puede comprometerse.

☐  **Privado**

Tú eliges quién puede ver y comprometerse con este repositorio.

Existente.

Añadir .gitignore: Ninguno ▾

Añadir una licencia: Ninguna ▾ ⓘ

.gitignore

Ninguna

Actionscript

Ada

Agda

Androide

...


© 2019 GitHub, Inc. [Contacta con GitHub](#)



[Precios](#) [API](#) [Formación](#) [Blog](#) [Acerca de](#)


Seleccionamos el tipo de lenguaje con el que se va desarrollar el proyecto así como su tipo de licencia (no es necesario si no lo deseas)

Configuración rápida - si has hecho este tipo de cosas antes

 Configurar en escritorio

☐ HTTPS

☒ SSH



Comience [creando un nuevo archivo](#) o [cargando un archivo existente](#). Recomendamos que cada repositorio incluya [README](#), [LICENCIA](#) y [.gitignore](#).

... o crear un nuevo repositorio en la línea de comandos

```
echo "# Githubklever" >> README.md
```

Github creara automáticamente la direccion HTTPS asi como en protocolo SSH

Listo para trabajar de forma remota desde Egit con eclipse o directamente con Git desde tu PC