UNIVERSIDAD PRIVADA DE TACNA
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS
INGENIERÍA DE SISTEMAS

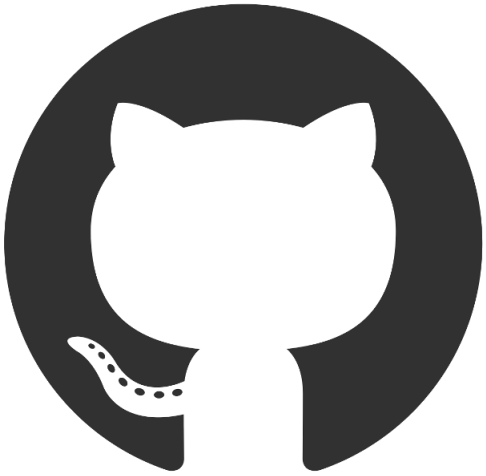
Sistemas de Control de Versiones Online

Integrantes : Brandon Dennis Mamani Ayala
2015052715
Profesor : Ing. Patrick Cuadros Quiroga
Curso : Base de Datos II
Fecha : 19 de agosto de 2018

Tabla de Contenido

Capítulo 1

GitHub



1.1. ¿Qué es Github?

Github es una plataforma creada para facilitar el desarrollo colaborativo de software, nos permite alojar proyectos en la web gratuitamente, por lo general de forma pública, aunque podemos alojar los proyectos de modo privado, si pagamos una pequeña suscripción mensual.

1.2. ¿Qué nos ofrece?

Github ofrece al desarrollador toda la potencia y agilidad del sistema de control de versiones Git, más un interesante set de herramientas añadidas:

- Wiki.
- Sistema de seguimiento de incidencias.
- Interfaz gráfica para revisión/comparación de código.
- Visor de ramas de desarrollo.

1.3. ¿Cómo funciona Github?

Lo primero que debemos hacer es crear una cuenta en https://github.com. La activamos por mail y ya podemos crear nuestros repositorios. Los repositorios de Github son el almacén que utilizamos para guardar nuestro código. Github nos ofrece la opción de crear un repositorio vacío, recomendable cuando vamos a iniciar un nuevo desarrollo, o la opción de importar un proyecto ya existente, elegimos la que más nos convenga y mediante unos pocos comandos de consola configuramos la rama principal de nuestro repositorio, que por defecto se llamará "master". Cada programador puede crear sus propias ramas de desarrollo, donde tiene que llevar a cabo sus modificaciones, sin interferir en el trabajo de sus compañeros. Cuando terminamos y validamos un desarrollo paralelo, lo unimos con la rama principal y todos los miembros del equipo pueden descargar las nuevas modificaciones, sin alterar los desarrollos que estén llevando cabo en ese momento. Después de alojar el repositorio público en Github.com, cualquier usuario de la comunidad podrá aportar ideas, hacer un seguimiento del proyecto, incluso copiarlo y modificarlo a su gusto bajo la misma licencia.

1.4. Crear un nuevo repositorio

Un repositorio generalmente se usa para organizar un solo proyecto. Los repositorios pueden contener carpetas y archivos, imágenes, videos, hojas de cálculo y conjuntos de datos, cualquier cosa que su proyecto necesite. Se recomienda incluir un archivo README o un archivo con información sobre su proyecto.

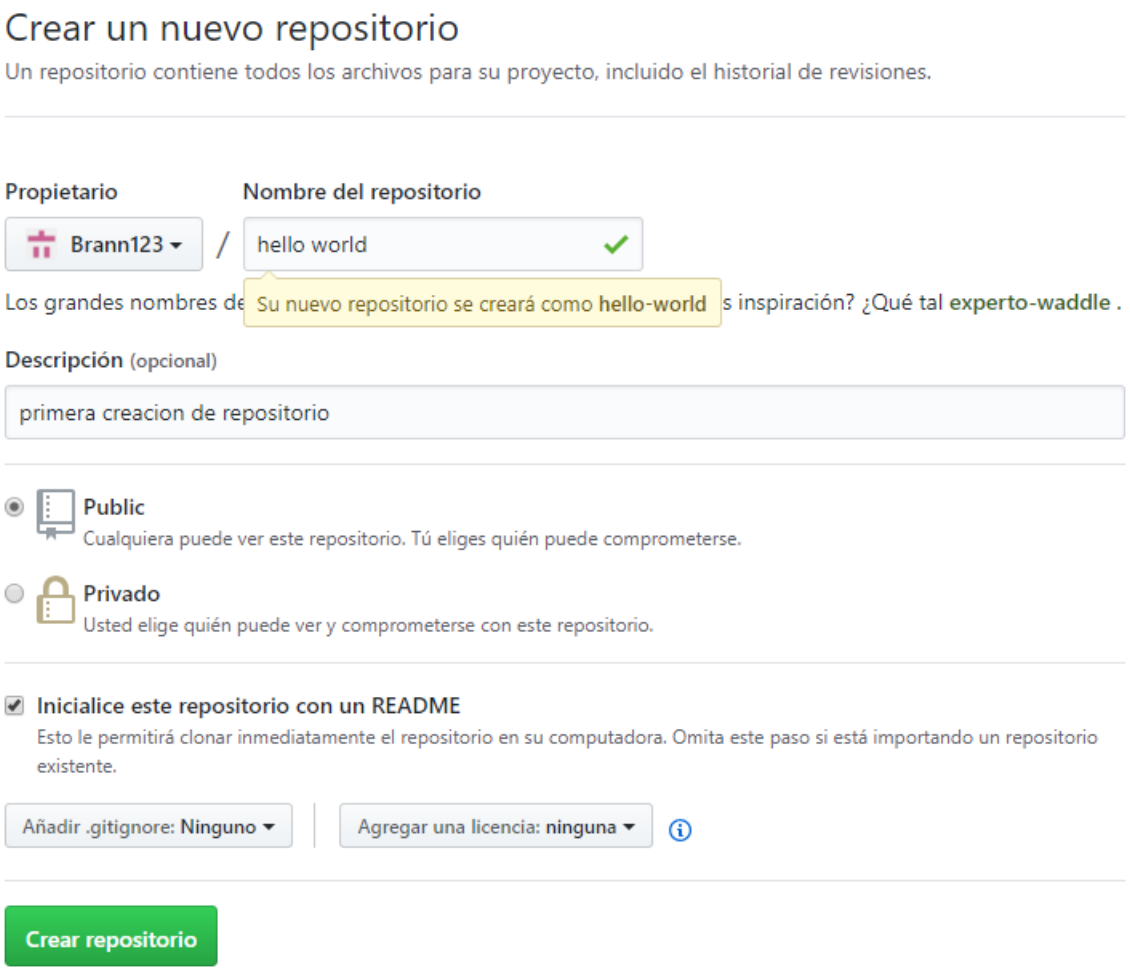
1. En la esquina superior derecha, al lado de tu avatar o identicon, haz clic en + (más) y luego seleccione **Nuevo repositorio**.

2. Nombra tu repositorio .

3. Escribe una breve descripción.

4. Seleccione Inicializar este repositorio con un README.

5. Finalmente haga click en **Crear repositorio**.



Capítulo 2

GitLab

2.1. Historia

Inicialmente el producto se publicó como un software completamente libre bajo la licencia MIT. Sin embargo, tras la división del proyecto en julio de 2013 en dos versiones distintas, GitLab CE (Community Edition) y GitLab EE (Enterprise Edition), finalmente en febrero de 2014 GitLab EE se comenzó a desarrollar bajo una licencia propietaria, con características que no están presentes en la versión libre.5En enero de 2017, la base de datos del servidor de producción fue eliminada accidentalmente, lo que supuso la pérdida de toda la actividad que se había desarrollado en las 6 horas anteriores.6En marzo de 2017, se anunció la compra del servicio de mensajería instantánea Gitter por parte de GitLab, sin que esto supusiera la fusión de ambos servicios, sino que continuarían como proyectos totalmente independientes. Además, la compañía anunció que publicaría el código de la nueva adquisición bajo la licencia MIT antes de junio de 2017.

2.2. ¿Qué es Gitlab?

GitLab nació como un sistema de alojamiento de repositorios Git, es decir, un hosting para proyectos gestionados por el sistema de versiones Git. Sin embargo, alrededor de esta herramienta han surgido muchas otras herramientas muy interesantes para programadores y equipos de desarrollo, que envuelven todo el flujo del desarrollo y el despliegue de aplicaciones, test, etc.

Sin duda, para dar una idea de lo que es GitLab, lo más rápido sería compararlo con uno de sus competidores, GitHub, pues éste último es especialmente conocido en el mundo del desarrollo de software. Todos más o menos estamos familiarizados con lo que ofrece, la gestión de repositorios, las issues, los pull request, GitHub Pages, etc. GitLab sería algo muy similar a lo que encontramos en GitHub, aunque a veces con otros nombres. Otras alternativas de programas o servicios para Git como Bitbucket están muy por detrás en posibilidades.

Aunque GitHub es un mounstruo, en cuanto a número de repositorios y funcionalidad, GitLab ha conseguido llegar aún más lejos, ofreciendo un conjunto más amplio de servicios que harán las delicias no solo de desarrolladores, sino también de devops.

2.3. ¿Cómo se usa?

GitLab es una herramienta basada en Git, que usas de la misma manera que cualquier otra herramienta similar. Generalmente usas Git a través de la línea de comandos, o a través de programas de interfaz gráfica, o del propio editor de código. Toda esa operativa que ya conoces y que hemos explicado en el Manual de Git, no cambia.

Además del hosting remoto para repositorios GitLab ofrece una interfaz web para controlar el repositorio y muchas otras herramientas. Ofrece la posibilidad de examinar el código en cualquiera de sus versiones, realizar acciones relacionadas con el sistema de repositorios como mergear el código de versiones de proyecto o gestionar las "pull request"(que en GitLab se llaman "merge request"), gestionar problemática de tu software diversa, automatizar procesos como el despliegue o la ejecución de pruebas del software, etc. Toda esta operativa la realizas, o configuras, en GitLab por medio de una web.

Por tanto, para usar GitLab simplemente necesitas las mismas herramientas que ya utilizas en tu día a día, el terminal o un programa de interfaz gráfica para gestionar tu repositorio,

así como el navegador web para acceder a el ecosistema de herramientas disponible en el sitio de GitLab. Por supuesto, todas estas herramientas las puedes usar desde cualquier ordenador conectado a Internet, independientemente de su sistema operativo.

2.4. Funcionalidades de Gitlab

En GitLab podemos gestionar principalmente proyectos, Grupos y Sinppets. Los proyectos son los protagonistas del sistema, básicamente repositorios de software gestionados por GitLab y todo el ecosistema GitLab. Los grupos son básicamente empresas y usuarios. Los snippets por su parte son como pedazos de código que puedes dejar para hacer cualquier cosa. Como decimos, dentro de los proyectos es donde se aglutinan la mayoría de las funcionalidades que vamos a resumir:

a) Overview:

Es un listado de todo el proyecto, los archivos, los README.md. Es parecido a lo que vemos cuando accedemos a un proyecto con GitHub. Te da el resumen del repositorio, archivos, commits, etc.Luego tiene dos subsecciones: En Activity del proyecto te ofrece toda la actividad, de una manera estadística. En Cycle Analytics además te ofrece algo muy novedoso, no disponible en otras herramientas. Básicamente informa el tiempo que se tarda en realizar una funcionalidad, desde que tienes la idea hasta que se incorpora al software, de modo que cualquier persona, incluso sin conocimientos de programación, puede saber el tiempo que ocupó el hacer las tareas. Una información muy valiosa que puede ayudar a futuro a estimar mejor el tiempo de trabajo necesario para nuevas funcionalidades. Obviamente, cuantas más issues tengas en el sistema, más datos tendrás para saber el tiempo que necesitas para las próximas tareas.

b) Repository:

Dentro de la sección Repository tenemos varias opciones diversas que afectan al repositorio del proyecto.Tenemos "Files", donde se puede navegar por los directorios y archivos, cuyo código podemos ver, e incluso editar los ficheros. Está disponible una

visualización por ramas y dispone de utilidades diversas para poder hacer cosas relacionadas con el repositorio remoto, ahorrando la necesidad de lanzar comandos. Tiene un buscador de archivos muy potente.En Çommits.ençontramos un listado de todos los commits realizados contra el repositorio, junto con datos específicos de cada uno de ellos.Las ramas "Branches"sirven para ver las ramas que tenemos en el repositorio.La siguiente sección, "Tags", es importante también, pues es el mecanismo disponible en Git para definir puntos del estado del código, correspondientes a cada release.Además esta sección tiene otras áreas también importantes, que os dejamos para vuestra propia investigación. Especialmente sería destacable la parte de "Locked files", disponible solo en GitLab como servicio, que es algo que no ofrece el propio sistema de control de versiones Git pero que han implementado dentro de GitLab, que permite bloquear un fichero para que solo ciertas personas lo puedan editar.

c) Issues:

Este es otra de las grandes utilidades de GitLab, que permite definir cualquier problema que se detecta en el software y darle seguimiento. Seguro que las conocemos porque es una de las partes fundamentales de GitHub y habremos navegado por ellas en decenas de ocasiones.Básicamente nos permite ver las issues generadas en un proyecto, mantener discusiones sobre ellas, y controlar los flujos de trabajo para su resolución, permitiendo definir las personas que deben resolverla, el tiempo estimado y el usado, la fecha límite, el peso de las tareas, etc.En GitLab han publicado otra interesante innovación que es un tablero de issues (Issue Boards), que permite visualizar las tareas, de una manera similar a los boards de Trello. Como gestores somos capaces de definir los tableros y las etiquetas. GitLab, por medio de la gestión de las Issues, es capaz actualizar el estado de las tareas, permitiendo visualizar su evolución por medio de los tableros.Otra cosa muy interesante es el "Service desk", que te ofrece un email que lo puedes proporcionar al cliente. Sin que el cliente se registre en GitLab, ni tenga acceso al proyecto, puede enviar mensajes a ese email, adjuntando texto, imágenes y archivos. GitLab, al recibir el correo, da de alta automáticamente una issue con ese contenido.

d) Merge Request:

Son como las Pull Request de GitHub. Te permiten controlar todas las solicitudes de combinación o unión de código de distintas ramas o forks. Es muy importante que los merges se resuelvan mediante la interfaz gráfica, ya que nos ofrece muchas posibilidades interesantes, como automatización de tests, la posibilidad de revisión de los cambios por parte de componentes del equipo, implementar diversas políticas de control sobre el código del proyecto, etc.

e) CI/CD:

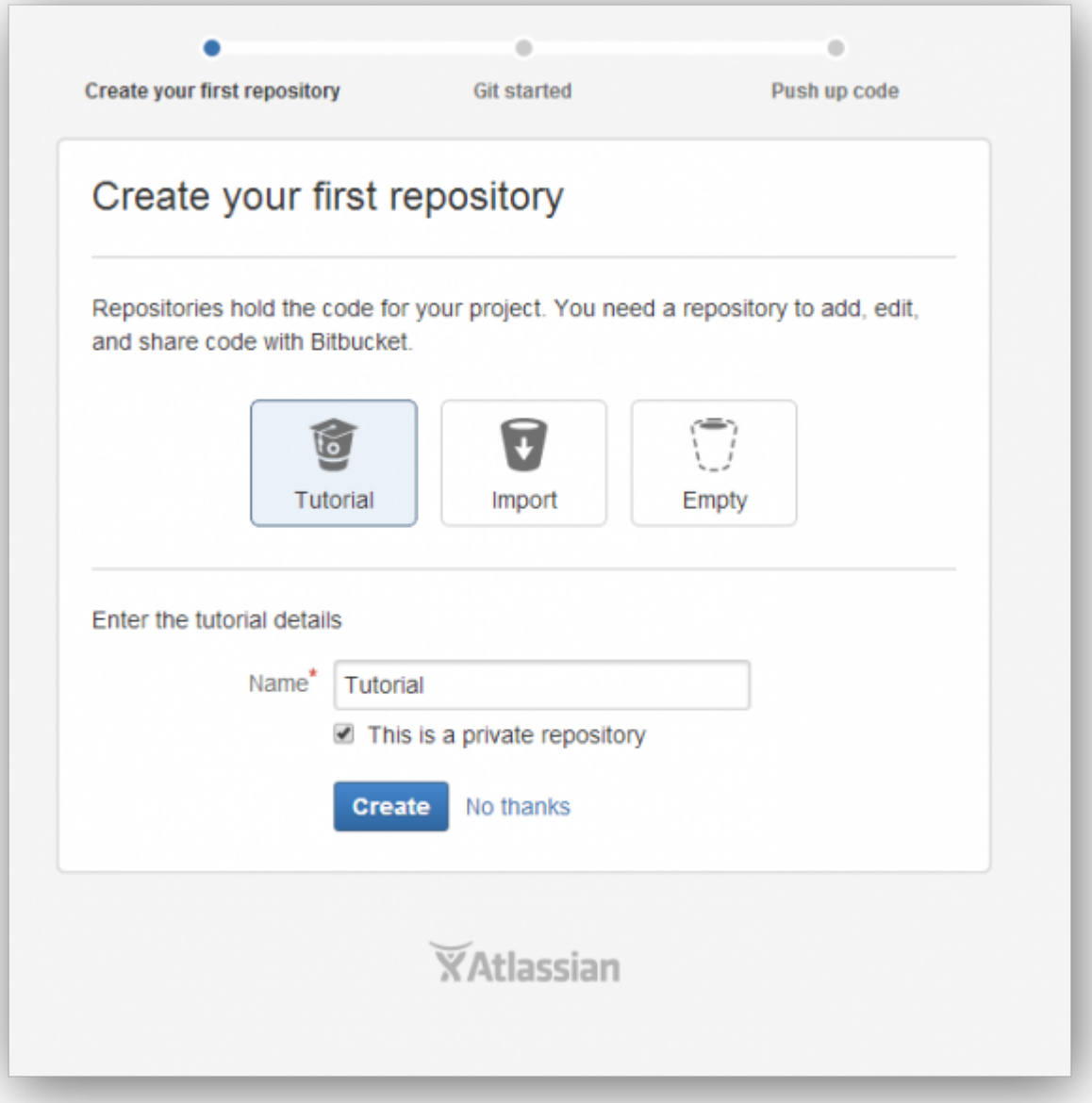
Es una de las maravillas que dispone GitLab, una herramienta sencilla y muy útil para los procesos de integración continua y despliegue continuo. Existen muchas herramientas que se pueden integrar para automatizar los procesos y llegar a crear flujos de trabajo completamente automatizados. De modo que se lancen los test y si todo va bien se puedan realizar una serie de tareas definida, que pueden llegar a producir el despliegue automático de las aplicaciones.Solo disponer de esta sección es suficiente motivo para pasarse a GitLab. No llega la complejidad de herramientas específicas como Jenkins, pero resuelve de manera muy potente problemas similares.

Capítulo 3

Bitbucket

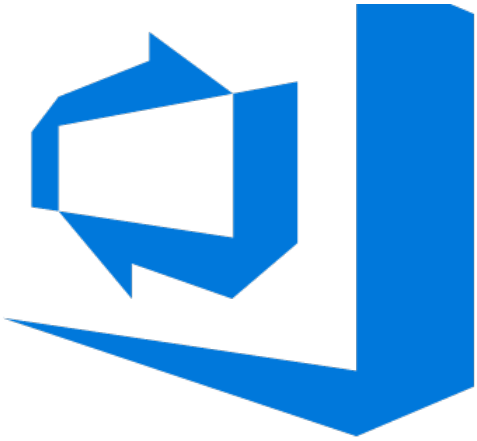
3.1. ¿Qué es Bitbucket?

Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados (que puede tener hasta cinco usuarios en el caso de cuentas gratuitas) desde septiembre de 2010, los repositorios privados no se muestran en las páginas de perfil - si un usuario sólo tiene depósitos privados, el sitio web dará el mensaje .Este usuario no tiene repositorios". El servicio está escrito en Python. Es similar a GitHub, que utiliza Git. En una entrada de blog del 2008, Bruce Eckel hace una comparación favorablemente de Bitbucket frente a Launchpad, que utiliza Bazaar.


Create your first repository
Git started
Push up code
Create your first repository
Repositories hold the code for your project. You need a repository to add, edit, and share code with Bitbucket.
Tutorial
Import
Empty
Enter the tutorial details
Name*
Tutorial
This is a private repository
Create
No thanks
Atlassian

Capítulo 4

Visual Studio Team Services



4.1. ¿Qué es VSTS?

Visual Studio Team Services (Team Services) es un conjunto de herramientas de colaboración basado en la nube que sirve para planear, desarrollar y administrar proyectos de software de cualquier tamaño, en cualquier lenguaje de programación de software. Usando las capacidades de Team Foundation Server, junto con servicios en la nube adicionales, Team Services le ayuda a administrar su código fuente, elementos de trabajo, compilaciones, pruebas y otros recursos.

4.2. Seguridad

Team Services usa la infraestructura de Plataforma como servicio (PaaS, por sus siglas en inglés) de Microsoft Azure y muchos servicios de Azure, incluidas las bases de datos SQL de Azure, para proporcionar un servicio fiable y disponible a nivel global para sus proyectos. Team Services también puede usar Azure Active Directory (Azure AD, por sus siglas en inglés) para autenticar de forma segura a los usuarios y controlar el acceso a los recursos críticos del equipo. Puede administrar los permisos de su cuenta de Team Services agregando grupos de Azure AD y definir los niveles de acceso para determinar las características que pueden usar los miembros del equipo.

4.3. Privacidad

Durante más de dos décadas, el enfoque de Microsoft de la privacidad y la protección de datos se ha basado en nuestro compromiso de proporcionar a las organizaciones la propiedad y el control sobre la recopilación, el uso y la distribución de los datos de sus clientes.

Cómo administra Microsoft los datos de Team Services: Usamos los datos del cliente con el único fin de proporcionar los servicios acordados y no para fines de marketing o publicitarios. Si pone fin a su suscripción, Microsoft se rige por estándares estrictos y sigue procesos específicos para eliminar los datos de nuestros sistemas.

Dónde se encuentra los datos de Team Services: Los clientes que deben mantener sus datos en una ubicación geográfica específica, por ejemplo, dentro de la Unión Europea (UE), pueden confiar en nuestra red de centros de datos global en expansión. Microsoft cumple además con las leyes internacionales de protección de datos relativas a las transferencias de los datos del cliente más allá de las fronteras.

Quién puede acceder a los datos de Team Services y en qué condiciones: Aplicamos fuertes medidas de seguridad para proteger sus datos frente al acceso indebido y el uso por parte de personas no autorizadas. Restringimos el acceso al personal y subcontratistas de Microsoft, y definimos cuidadosamente los requisitos para responder a las órdenes gubernamentales de los datos del cliente. Sin embargo, usted sí puede acceder a los datos de sus clientes en cualquier momento y por cualquier motivo.