

SERENITY BDD con CUCUMBER

7

Control de Versiones

Git

Bitbucket

(Tiempo ejecución 3 horas)

Objetivos:

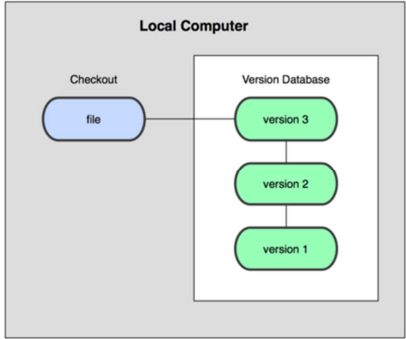
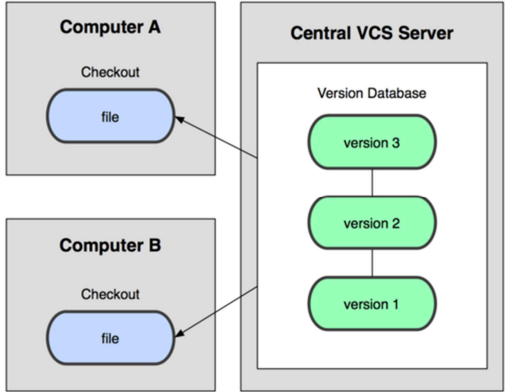
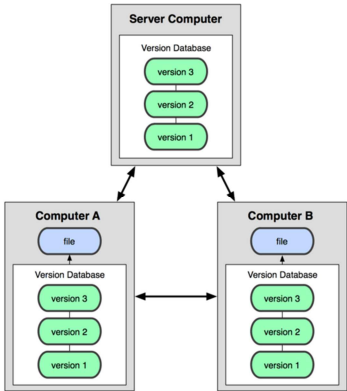
Comprender el concepto del versionamiento, controles e importancia.

Que es el control de versiones?

El control de versiones es un sistema (**Version Control System** o **VCS**) que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

Además, permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más.

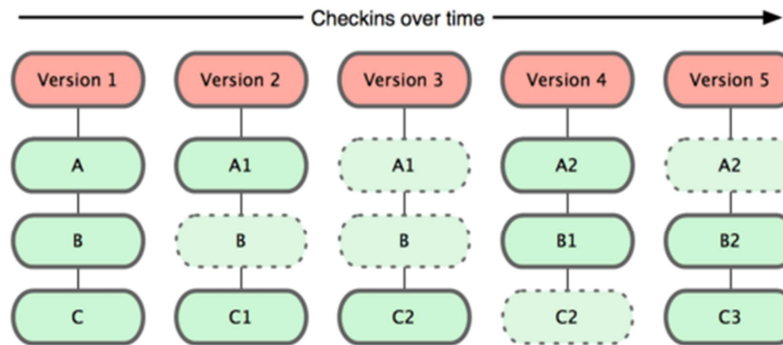
Tipos de VCS?

<p>Sistemas de control de versiones locales.</p> <p>Consiste en la creación de una simple base de datos en la que se lleva el registro de todos los cambios realizados sobre los archivos .</p>	
<p>Sistemas de control de versiones centralizados. (Centralized Version Control Systems o CVCSs). Estos sistemas, como CVS, Subversion, y Perforce, tienen un único servidor que contiene todos los archivos versionados, y varios clientes que descargan los archivos desde ese lugar central. Durante muchos años éste ha sido el estándar para el control de versiones.</p> <p>Ventajas:</p> <ul style="list-style-type: none">* Todo el mundo puede saber (hasta cierto punto) en qué están trabajando los otros colaboradores del proyecto.* Los administradores tienen control detallado de qué puede hacer cada uno; y es mucho más fácil administrar un CVCS que tener que lidiar con bases de datos locales en cada cliente. <p>Desventajas</p> <ul style="list-style-type: none">* El punto único de fallo que representa el servidor centralizado. Si ese servidor se cae durante una hora, entonces durante esa hora nadie puede colaborar o guardar cambios versionados de aquello en que están trabajando.* Si el disco duro en el que se encuentra la base de datos central se corrompe, y no se han llevado copias de seguridad adecuadamente, pierdes absolutamente todo	
<p>Sistemas de control de versiones distribuidos. (Distributed Version Control Systems o DVCSs en inglés). En un DVCS (como Git, Mercurial, Bazaar o Darcs), los clientes no sólo descargan la última instantánea de los archivos: replican completamente el repositorio. Así, si un servidor muere, y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios de los clientes puede copiarse en el servidor para restaurarlo. Cada vez que se descarga una instantánea, en realidad se hace una copia de seguridad completa de todos los datos</p>	

Que es GIT?

Es un software de control de versiones diseñado por Linus Torvalds. Permite la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración, proporcionando las herramientas para desarrollar un trabajo en equipo de manera inteligente y rápida y por trabajo nos referimos a algún software o página que implique código el cual necesitemos hacerlo con un grupo de personas.

Git almacena la información como instantáneas del proyecto a lo largo del tiempo, básicamente hace una foto del aspecto de todos tus archivos en ese momento, y guarda una referencia a esa instantánea.



Estados

Git tiene tres estados principales en los que se pueden encontrar tus archivos: confirmado (committed), modificado (modified), y preparado (staged).

confirmado (committed): significa que los datos están almacenados de manera segura en tu base de datos local.

modificado (modified): significa que has modificado el archivo pero todavía no lo has confirmado a tu base de datos.

preparado (staged): Preparado significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

Esto nos lleva a las tres secciones principales de un proyecto de Git: el directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area).

*El directorio de Git, es donde se almacena los metadatos y la base de datos de objetos para tu proyecto. Es la parte más importante de Git, y es lo que se copia cuando clonas un repositorio desde otro ordenador.

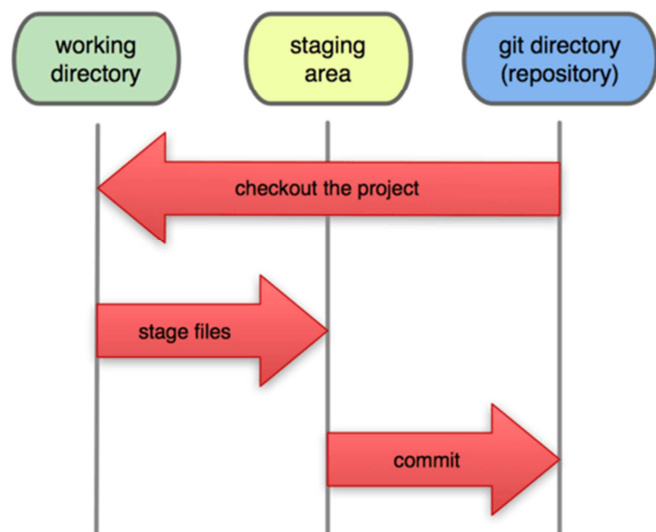
*El directorio de trabajo es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que los puedas usar o modificar.

*El área de preparación es un sencillo archivo, generalmente contenido en tu directorio de Git, que almacena información acerca de lo que va a ir en tu próxima confirmación.

El flujo de trabajo básico en Git es algo así:

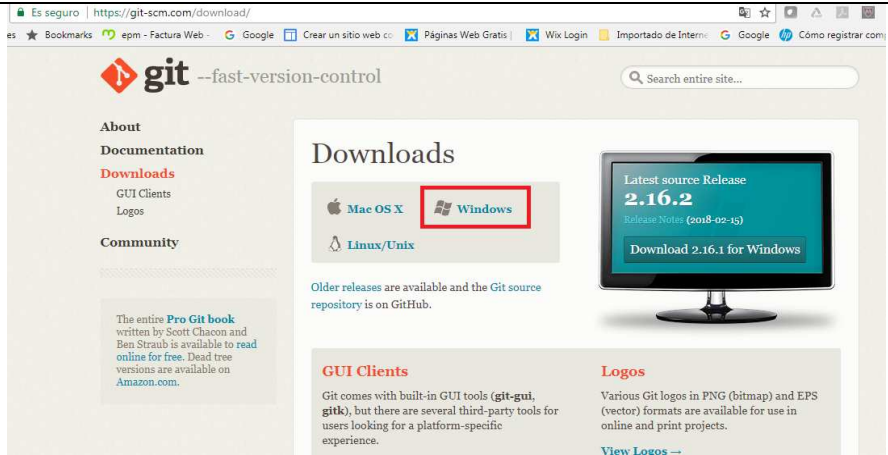
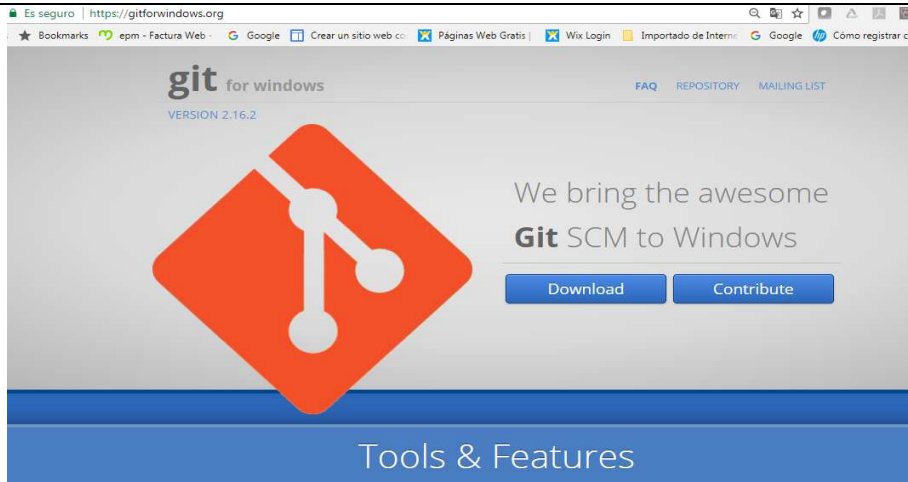
- Modificas una serie de archivos en tu directorio de trabajo.
- Preparas los archivos, añadiéndolos a tu área de preparación.
- Confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación, y almacena esas instantáneas de manera permanente en tu directorio de Git.

Local Operations



Descarga e Instalación:

Descargar la versión más reciente de git.

<p>https://git-scm.com/download/</p>	
<p>https://gitforwindows.org/</p>	

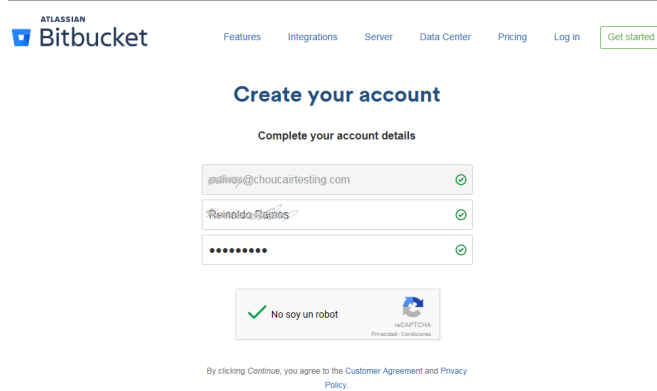
Repositorio:

Github y Bitbucket: son dos servicios que te permiten administrar tus proyectos usando el popular sistema de control de versiones Git, pero en la nube.

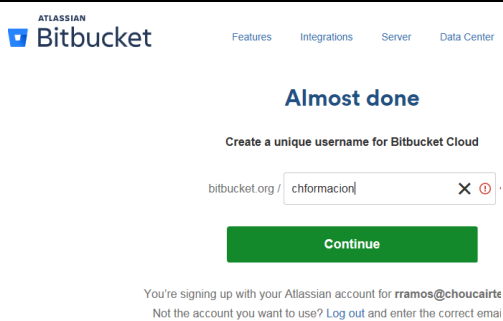
Crear Cuenta en Bitbucket:

Ingresar a bitbucket
<https://bitbucket.org/account/signup/>

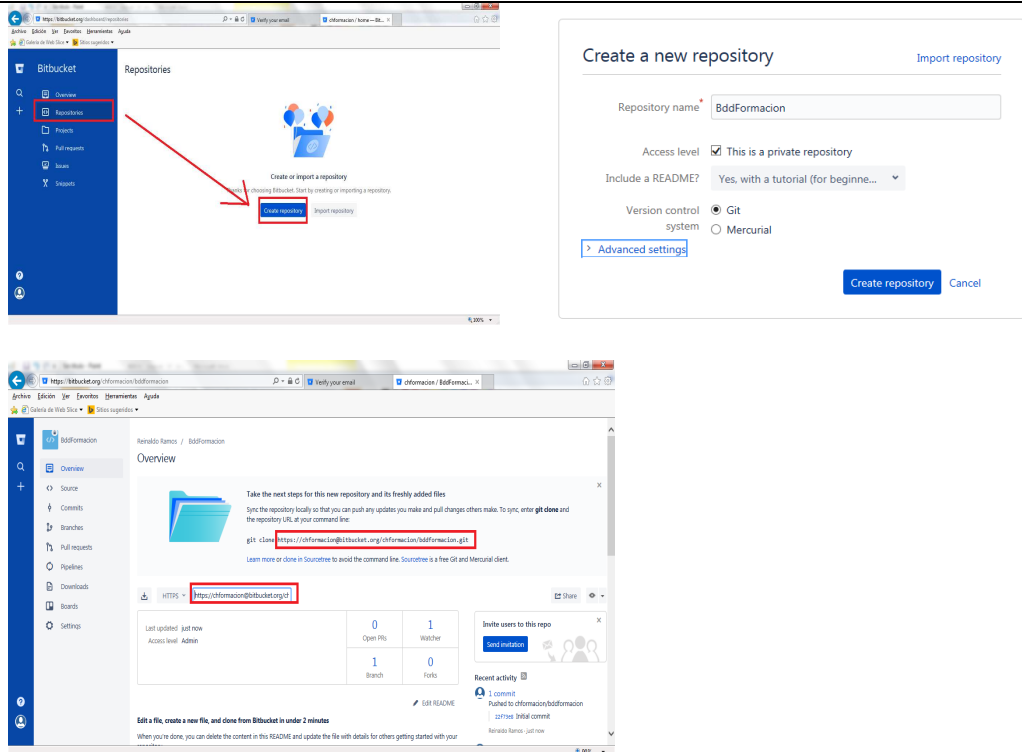
Crear una cuenta con los datos de mail usuario y clave.

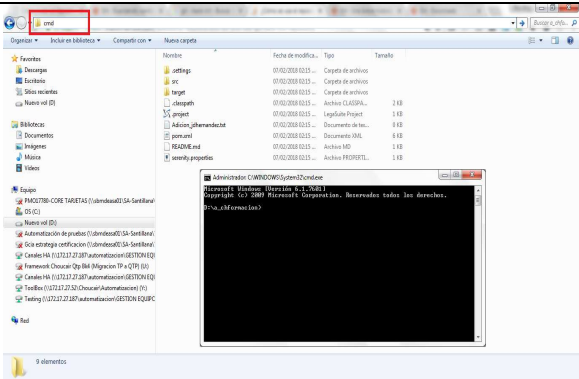


Una vez verifique la cuenta, asigne un nombre a su cuenta.



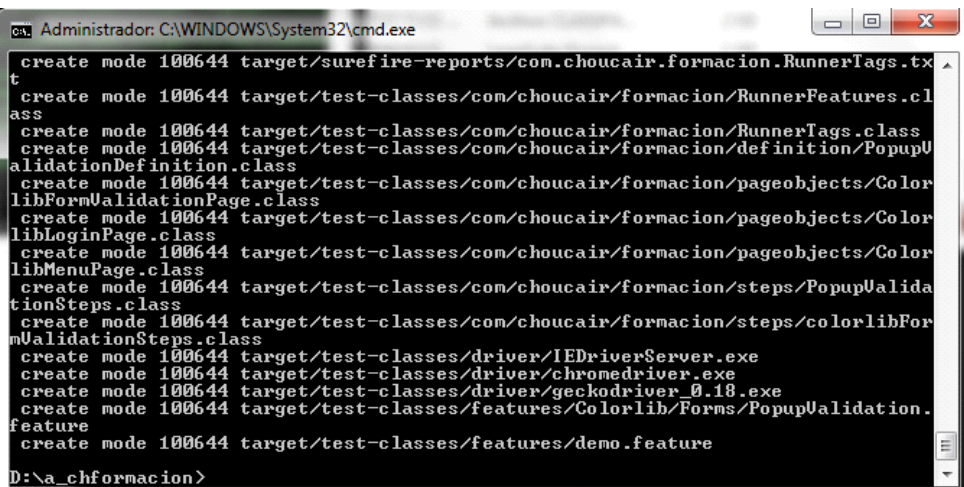
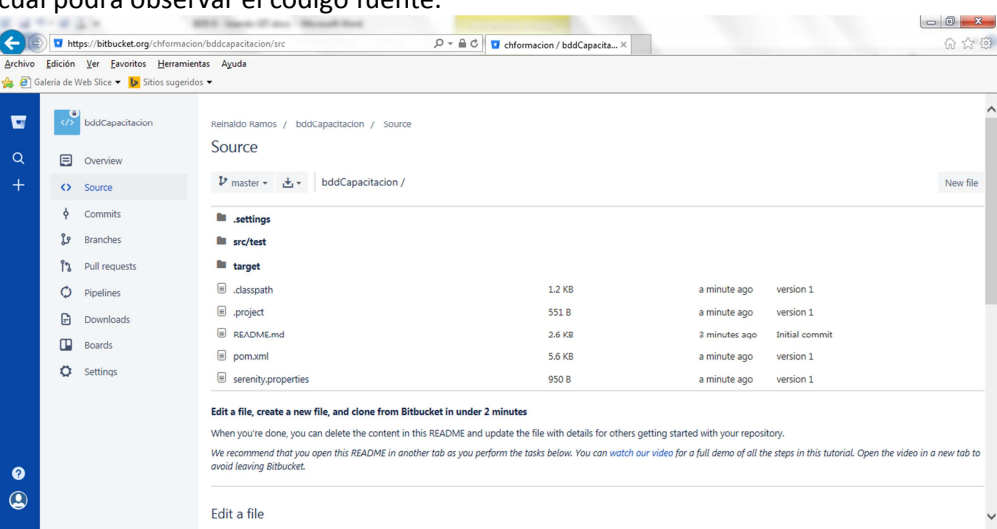
Agregar un repositorio en bitbucket.
*Asignar un nombre al repositorio
*Puede crear el repositorio privado o público.
*Si desea puede incluir un tutorial "readme".
*clic en Create repository.
*use la url de la pestaña overview para hacer referencia al repositorio con git.



<p>Ejecutar la consola desde la raíz del del proyecto.</p> <p>Otra forma es dar clic derecho y luego “git bash here”</p>	
<p>Crear proyecto</p>	<p>Git init</p> <p>Este comando crea un repositorio de Git vacío, básicamente un directorio .git con subdirectorios para objetos, refs / heads, refs / tags y archivos de plantilla.</p> <pre>D:\a_chformacion>git init Initialized empty Git repository in D:/a_chformacion/.git/ D:\a_chformacion></pre>
<p>Conectar a repositorio bitbucket</p>	<p>Git remote</p> <p>Con este comando se establece una conexión “origin” al repositorio a través de la url (tome la url dada en bitbucket para el repositorio)</p> <pre>D:\a_chformacion>git remote add origin https://chformacion@bitbucket.org/chformacion/bddformacion.git _</pre> <p>Con este comando, se consultan las conexiones establecidas</p> <pre>D:\a_chformacion>git remote -v origin https://chformacion@bitbucket.org/chformacion/bddformacion.git (fetch) origin https://chformacion@bitbucket.org/chformacion/bddformacion.git (push)</pre>
<p>Sincronizar bitbucket con repositorio local</p>	<pre>D:\a_chformacion>git pull origin master</pre>

Una vez sincronizado el repositorio web con el repositorio local, procedemos a subir el proyecto a bitbucket. Los siguientes pasos, son los que se deben seguir realizando cada vez que deseemos actualizar los cambios a nuestro proyecto en el repositorio web.

<p>Agregar archivos a seguimiento</p>	<p>Git Add, marca los archivos que van a ser utilizados en un próximo commit.</p> <p>-A :agrega todo el contenido del árbol.</p> <pre>D:\a_chformacion>git add -A _</pre>
<p>Consultar los archivos marcados</p>	<p>Git status, consulta los archivos agregados al index.</p> <pre>D:\a_chformacion>git status _</pre> <p>Se visualiza en color verde los archivos marcados</p> 

<p>Grabar el cambio</p>	<p>Git commit, Almacena el contenido actual del índice en una confirmación nueva junto con un mensaje de registro del usuario que describe los cambios.</p> <pre>D:\a_chformacion>git commit -m "version 1.0"</pre> 
<p>Actualizar el repositorio remoto</p>	<p>Git push, <origin> es el alias asignado previamente con el git remote. <master> es el branch principal del repositorio.</p> <pre>D:\a_chformacion>git push origin master</pre> <p>El sistema iniciará el proceso de cargue, al finalizar consulte en bitbucket.</p>
<p>Consultar fuentes en repositorio bitbucket</p>	<p>Una vez realizado el push respectivo, consulte el repositorio, opción “Source” en la cuál podrá observar el código fuente.</p> 

Compartir repositorio

Desde bitbucket es posible compartir el proyecto a otras cuentas de bitbucket, de esta forma otros usuarios podrán consultar nuestro proyecto.

Overview > share > buscar cuenta > agregar

Para compartir el proyecto al formador, agregue la cuenta chformacion.

