**CONTROL DE VERSIONES**

**(SOFTSES)**

**NATALIA GRANDA CANO.**

**LLESSICA GIRADLO BUSTAMANTE.**

**SHARON OSORNO.  
  
  
  
  
  
  
  
  
  
IDENTIFICACIÓN DE ESTÁNDARES PARA LA DOCUMENTACIÓN Y CONSTRUCCIÓN DE INFORMES**

**DIANA PATRICIA BEDOYA RUÍZ**

**JHON JAIRO**

**TÉCNICA PROFESIONAL EN PROGRAMACIÓN DE SISTEMAS DE INFORMACIÓN**

**(CICLO PRO. CON TSD)**

**POLITÉCNICO COLOMBIANO JAIME ISAZA CADAVID**

**MEDELLÍN**

**2015**

**CUADRO COMPARATIVO ENTRE HERRAMIENTAS PARA VERSIONAMIENTO**

|  |  |  |
| --- | --- | --- |
| **Herramienta para versionamiento** | **PRO´S** | **CONTRA´S** |
| **Git** | -Aloja el repositorio de código en el que trabaja todo el equipo  -Posee un visor de código en linea lo cual permite modificarlo desde allí mismo  -Tiene resaltado de sintaxis  -Se puede dividir en ramas para trabajo colectivo  -Es gratuito para alojar el código como público  -Sistema distribuido, podemos hacer commits en nuestra máquina local, por lo que podemos seguir trabajando sin conexión a Internet | **-**Muchos comandos y el significado de estos  -No indexa directorios |
| **Subversión** | **-** Subversión puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras.  -Fomenta la colaboración  -Sistema centralizado | **-**El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado. -No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita llevar la cuenta de qué cambios se han realizado |
| **Mercurial** | -Más simple que Git **-**No hay tantas funciones que aprender, son similares a los otros sistemas | **-**Solo para grandes proyectos |

**2. Seleccione la opción más apropiada y configure su sistema**

Estas tres herramientas fueron preseleccionadas debido a que el IDE de Desarrollo con el que trabajamos actualmente, NetBeans cuenta con estas para el trabajo de versionamiento, las tres son compatibles para este trabajo pero al final la escogida definitiva es Git.

Se escoge principalmente por lo fácil de su manejo, implementación con el IDE NetBeans es muy fluido, su configuración es corta y sencilla además de las funciones de add, commit, push, y pull que se hacen en muy pocos pasos

**3. Lleve a cabo las siguientes tareas, documentando los pasos necesarios para llevarlas a cabo en el sistema seleccionado**.

**A.** **Instalación y/o configuración del servidor de versionamiento**

El servidor seleccionado es GitHub; el servidor arranca cuando se crea una cuenta en GitHub donde en la parte derecha de la ventana saldrá un formulario de registro con tres campos básicos que son nombre de usuario, correo electrónico y contraseña una vez llenados todos los campos se da clic en crear cuenta y listo, ya tenemos una cuenta en GitHub, paso seguido crear un repositorio, se hace de la siguiente manera:   
estando dentro de nuestra cuenta de GitHub en la parte derecha hay un botón que dice “ +New repository” o nuevo repositorio en español al cual se le dará clic, se ingresará el nombre del repositorio y una descripción, se elige si será público o privado (repositorio privado cuesta dinero) y clic en crear repositorio, hasta el momento ya esta creado el repositorio, ahora para configurarlo nos vamos al IDE NetBeans donde daremos clic en la pestaña “Team” luego nos posicionamos en “Git” y clic en “Clone” nos pedirá un link el cual esta en la página donde creamos nuestro repositorio lo copiamos y pegamos, clic en Next nos pedira seleccionar la rama clic en Next luego pedirá la carpeta donde se guardará el proyecto se selecciona la carpeta y clic en Finish y con esto ya tendremos configurado GitHub con NetBeans.

**B. Base de datos de Contraseñas**

Los grupos y permisos se definen al momento de crear la cuenta en GitHub donde se registran uno o más usuarios que trabajaran en el proyecto, en una cuenta de GitHub se pueden añadir más correos de más personas que actuaran como colaboradores subiendo cada uno sus versiones

.

**C. Estructura del repositorio**

Para importar un proyecto al repositorio remoto de GitHub desde nuestro ordenador se puede hacer la opción Push con la cual el proyecto que tenemos en nuestro ordenador se subirá al repositorio de GitHub, esto se puede hacer mediante el comado git push   
Los contenidos del repositorio se pueden listar o mejor dicho ver al momento de dar clic sobre el repositorio, cuando se da clic sobre el repositorio se ve toda la información que este posee, se pueden ver los commit las ramas que se tienen en general todo tipo de contenido que se haya subido al repositorio remoto de GitHub   
Para crear un directorio de archivos se debe dar clic en el nombre del repositorio, este se abrirá, al lado del nombre del repositorio hay un símbolo “+” al cual si se le da clic abrirá un editor en el cual podremos hacer un nuevo archivo para el proyecto , como no se puede crear vacío el archivo, lo crea en un directorio al finalizar de digitar los cambios se da clic en guardar o salvar.

**D. Comandos básicos del sistema de control de versiones**

Las copias de trabajo se hacen para tener un respaldo en caso de que el original se pierda dañe etc, estas se pueden hacer con el comando git clone o en NetBeans dando clic derecho sobre el proyecto que se esta trabajando luego posicionarse en git, luego posicionarse en remote y luego clic en clone, nos pedirá el link del repositorio sonde se va a hacer la “copia de seguridad”.  
Con la ayuda del comando git fetch es posible observar todo el estado del repositorio en el momento actual.  
Para agregar documentos al proyecto basta con usar el comando git add o en su defecto , usando NetBeans, se sitúa en el proyecto que se esta trabajando clic derecho luego clic en git y luego clic en add con esto los archivos se añadirán al proyecto.  
Para guardar los cambios hechos durante cierto tiempo de trabajo sobre el proyecto basta con usar el comado git commit o en su defecto usando NetBeans nos situamos en el proyecto que se esta trabajando clic derecho , luego clic en git, y luego clic en commit (antes de hacer esto se debe añadir al proyecto) con esto los archivos del proyecto se guardaran (todavia no están subidos al repositorio remoto de GitHub).

**E. Ciclo de trabajo con un programa de control de versiones**

Para trabajar con la versión más reciente que se haya subido al repositorio, se puede hacer mediante el comando git pull o en su defecto si trabajamos en NetBeans nos situamos en el último proyecto que tengamos o nuestra última versión seguido a esto clic derecho encima del proyecto nos situamos en git, luego en remote y por ultimo clic en pull, se abrirá una ventana donde nos pedirá de que repositorio bajar la última versión se pone el link del repositorio, clic en Next, luego nos pide la rama y se selecciona la rama master y por ultimo clic en Finish y ya tendremos lista la última versión para trabajar sobre ella.  
Para ver la historia o cambios realizados de un documento o archivo ya subido al servidor remoto de GitHub se posiciona sobre el proyecto en NetBeans, se le da clic derecho luego, se posiciona en git, y por último Show History, sale una ventana donde salen todos los cambios realizados con mensajes de commit fechas de creación y autores.

Para revisar los cambios que se ha realizado en una sesión entramos a la página del repositorio remoto, allí están los datos de quien ha hecho un cambio últimamente al proyecto.

Para saber quién hizo un cambio especifico en alguna parte del proyecto debemos estar en la página del repositorio, cada vez que un usuario hace un cambio y lo sube en dicha página se ve el cambio que hizo si ingresamos al cambio se verá la fecha del cambio, que cambio, y quien hizo ese cambio (cuando se hace el commit hay un campo que dice quién es el autor del cambio, en la página del repositorio aparecerá este nombre)

**F. Etiquetas y Ramas**

Para la creación de una rama basta con posicionarnos sobre nuestro proyecto de NetBeans dar clic derecho, luego posicionarse en git, luego posicionarse en branch y luego clic en Create Branch, del mismo modo se procede al crear una etiqueta, nos posicionamos sobre nuestro proyecto de NetBeans dar clic derecho, luego posicionarse en git, luego posicionarse en branch y luego clic en Create Tag, en la página del repositorio remoto, el la pestaña branch aparecen todas las ramas creadas.  
Para integrar las ramas a un tronco, al momento de subir los cambios guardados de nuestro proyecto con “push” nos pide en que rama se quiere subir este cambio que se esta haciendo, al subirse a la rama master se esta subiendo a la rama principal.

**BIBLIOGRAFÍA**

*Git, La guía sencilla* (20 de Enero de 2012) Obtenido de RogerDudler  
 http://rogerdudler.github.io/git-guide/index.es.html

*Sistemas de control de versiones* (25 de Octubre de 2013) Obtenido de   
 https://portfoliogabrielfcr.wordpress.com/2013/10/25/62/

*Introducción a Git* (11 de Marzo de 2011) Obtenido de Slideshare   
 http://es.slideshare.net/srus/introduccin-a-git-7233932

Derechos de autor, Meliza Guarín

www.youtube.com