## MERCURIAL Y TORTOISE EN UBUNTU Y WINDOWS

El objetivo de la práctica es asentar los conocimientos teóricos del funcionamiento de un sistema de control de versiones y aprender el manejo básico (clone, push, pull y merge) de Mercurial. Para ello usaremos la aplicación Tortoise en Windows y el terminal de Ubuntu.

Usaremos una cuenta en bitbucket.org, en donde crearemos nuestro repositorio. A este repositorio es al que subiremos (push) y desde el que bajaremos (pull) nuestros archivos, de manera que así podremos tenerlos siempre actualizados en cualquier máquina.

**PREPARACIÓN**

|  |  |
| --- | --- |
| **Windows – Tortoise**   * Abrimos Tortoise Hg Wokbench 🡪 File 🡪 Settings. * Seleccionamos “**Consignar**”. * En “N**ombre de usuario”** ponemos nuestro nombre y a continuación nuestro mail entre “<” y “>” (sin las comillas). * En “**Tortoise Hg”**, en **“Three-way Merge Tool”** y en **“Visual Diff Tool”** seleccionaremos “**kdiff3”**. | **Ubuntu**   * Editamos el archivo **hgrc** (situado en /etc/mercurial/) y añadimos las siguientes líneas:   **[ui]**  **username = nombre\_de\_usuario <dirección\_de\_correo>**  donde usuario2 es el nombre de usuario y dirección\_de\_correo su mail.   * A continuación, para que no nos de aviso más adelante de problemas de certificados, añadiremos también lo siguiente:   **[web]**  **cacerts = /etc/ssl/certs/ca-certificates.crt**   * Para obtener ayuda de los comandos:   **hg –h nombre\_del\_comando** |

Ya tenemos creado el repositorio en bitbucket, ahora tendremos que clonarlo a nuestro ordenador para trabajar con él en local.

|  |  |
| --- | --- |
| **Windows – Tortoise**   * Tortoise workbench 🡪File 🡪Clone repository     Dirección web en donde está almacenado repositorio  Destino dónde lo queremos guardar guardar | **Ubuntu**   * **hg clone https://losenlaces@bitbucket.org/losenlaces/prueba2 /home/ubuntu/Escritorio/Repprueba2**   Origen de la clonación  Destino de la clonación |

En el destino que le hayamos dicho se creará una carpeta con el nombre de nuestro repositorio. Creamos dentro de la carpeta un archivo de texto al que llamaremos “operaciones”. En él escribiremos “suma = a + b” (sin las comillas).

Ahora en nuestro repositorio local existirá un archivo (“operaciones”), cuyo contenido es “suma = a + b”, mientras que en el repositorio web no habrá nada. Tendremos que subir nuestro archivo al repositorio web creado en bitbucket.

|  |  |
| --- | --- |
| **Windows – Tortoise**   * Nos ponemos sobre el archivo 🡪 botón derecho 🡪 Hg realizar (commit).   Seleccionaremos el archivo deseado (1) y pondremos un texto para identificar el archivo (2). Le damos a “Añadir” y luego a “Confirmar”.    2  1   * Ahora el archivo ya “está preparado” para subirlo (push) al repositorio web. | **Ubuntu**   * Nos situamos en el directorio de nuestro repositorio. Una vez ahí, tenemos que añadir el archivo al que vamos a hacer un “commit”.   **hg add “operaciones”** (si queremos añadir varios: hg add archivo1 archivo2)  **hg commit –m “añadida suma”**   * Ahora el archivo ya “está preparado” para subirlo (push) al repositorio web.   **hg push** |

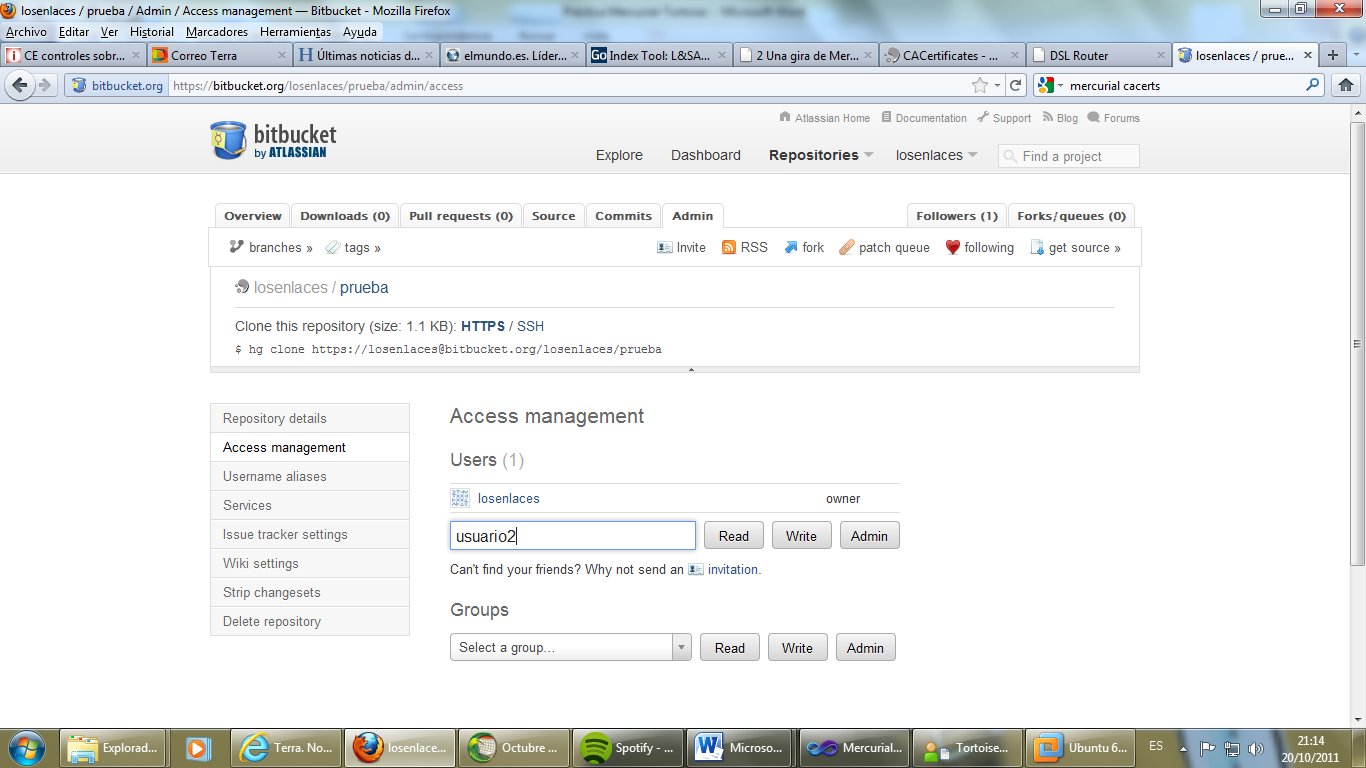
**TRABAJANDO MÁS DE UNA PERSONA EN UN MISMO REPOSITORIO**

La práctica se realizará en grupos de trabajo de 2 ordenadores.

Uno de ellos arrancará en Windows mientras que el otro lo hará en Ubuntu. Luego se intercambiarán las posiciones de manera que los que al principio estaban trabajando en Windows lo harán en Ubuntu y viceversa.

En bitbucket.org, uno de los participantes del grupo de trabajo (usuario1), modificará el repositorio anterior haciéndolo público.

Al otro usuario del grupo de trabajo (usuario2) se le dará derecho de escritura en el repositorio para que pueda hacer modificaciones y así poder trabajar los dos con el mismo repositorio cada uno en local y luego subirlo al repositorio de bitbucket.



Ahora, en la página de bitbucket del usuario2, en “dashboard”, aparecerá el repositorio del usuario1. Pulsamos sobre él y vemos la URL para clonar en local el repositorio del usuario1.

* Clonamos el repositorio del usuario1 en el ordenador del usuario2.
* El usuario2 modifica el archivo añadiendo “multiplicacion = a \* b” y lo sube al repositorio (commit, push).
* Comprobamos que efectivamente ha cambiado tanto en la página de bitbucket del usuario1 como en la del 2.
* Habrá que hacer un pull en el usuario1 para que este lo tenga actualizado en local.

|  |  |
| --- | --- |
| **Windows – Tortoise**    Pulsamos sobre el botón rodeado con un círculo rojo, y luego sobre la línea azul damos botón derecho del ratón y “Actualizar”. | **Ubuntu**  **hg pull**  **hg update** |

**CONFLICTOS**

¿Qué pasaría si el usuario1 modifica el archivo y a la vez lo está modificando el usuario2? Cuando el último de los 2 usuarios hiciera un push, mercurial detectaría que hay un conflicto y nos lo indicaría.

Vamos a ver como se resuelven los conflictos.

* El usuario1 añade al archivo “modulo = a%b” y hace un commit pero no hace push.
* El usuario2 añade al archivo “division = a/b”, hace un commit y push.
* Ahora el usuario1 hace push (se producirá un conflicto)

|  |  |
| --- | --- |
| **Windows – Tortoise**  conflicto    Para resolverlo, hacemos pull.    Botón derecho sobre la cabeza de la nueva rama y “Merge with local”.  Le damos “Siguiente”, luego “Resolved” y “Tool resolve”.  Se abrirá kDiff3, desde donde podremos resolver el conflicto.    Kdiff3 muestra simultáneamente los archivos implicados en el conflicto. Copiando y pegando o dando a los iconos con las letras “A”, “B”, “C” dejaremos el archivo como nos convenga.  Lo grabamos, decimos a todo que sí y, finalmente, hacemos push, actualizándose el repositorio web.  Para terminar, habrá que hacer pull tanto el usuario1 como el 2 para que se actualicen sus repositorios locales. | **Ubuntu** |