



Grado en Diseño y Desarrollo de Videojuegos

VJ1202 Informática básica

Laboratorio Sesión 02

Introducción

En esta sesión vamos a continuar aprendiendo a trabajar con *Git*. En la sesión anterior vimos los conceptos de *Commit* y *Push*. En esta sesión vamos a aprender los conceptos de *Clone* y *Pull*. También empezaremos a ver cómo se trabaja en equipo con *Git*.

Seguiremos usando **Github.com** como servicio web para almacenar nuestros repositorios. Para trabajar con los repositorios continuaremos con el uso del software **Github desktop**.

En este video tutorial se puede encontrar información de interés:

https://www.youtube.com/watch?v=j8CSUPIB8mA&list=PLvimn1Ins-43-1sXQmGZPWLjNjPyGNi0R&ab_channel=MitoCode

Objetivos

- Clonar (*Clone*) un repositorio remoto.
- Actualizar (*Pull*) nuestro repositorio con los cambios incluidos en un repositorio remoto.
- Trabajar en equipo.
- Modificar el fichero *Readme.md*.

Metodología

Cada estudiante debe entregar el trabajo realizado de forma **individual**. Ahora bien, no hay problema para que os juntéis para compartir dudas. Es decir, que podéis hacer equipos de dos o tres personas para compartir conocimientos, pero cada estudiante deberá realizar el trabajo en su ordenador.

La profesora evaluará el trabajo realizado de forma individual.

Antes de preguntar al profesor sigue los siguientes pasos:

1. **Lee** de nuevo el enunciado del ejercicio.
2. Dedicar unos minutos a **pensar** la solución por tí mismo/a.

3. **Pregunta** a algún compañero/a. Normalmente, el compañero/a habrá tenido la misma duda y la podrá resolver de forma inmediata.
4. Si ninguno de los pasos anteriores funciona, preguntar a la profesora. Ésta te dará una **pista** para que te sirva de ayuda, pero no te dará la solución.
5. Si no consigues resolver el problema, vuelve a preguntar a la profesora.

Trabajo a realizar

IMPORTANTE

Antes de empezar con la práctica es importante que tengas el repositorio que hiciste en la práctica anterior en el ordenador, para que Github Desktop te avise de cualquier cambio que se produzca. Si utilizaste un ordenador de aula en la práctica anterior y ahora no tienes la carpeta de la práctica anterior, invierte el orden de los dos primeros ejercicios: realiza primero el ejercicio 2 y después el ejercicio 1.

Ejercicio 1

Imaginemos que hoy es el último día para entregar el trabajo fin de grado en el que hemos trabajado 8 horas al día durante los últimos 4 meses. Justo hoy se estropea vuestro ordenador, borras por error la carpeta del proyecto o cualquier otra situación catastrófica que podáis imaginar. ¿Y ahora qué podemos hacer para poder entregar el trabajo de tantos meses a tiempo? Si habéis usado *Git* como sistema de control de versiones, podéis estar tranquilos/as.

Vamos a simular una situación catastrófica como la anterior:

- A. Ves a tu carpeta de repositorios y borra “sin miedo” la carpeta *ConversorDec2Bin*.

Afortunadamente, has usado *Git* y entonces puedes recuperar el repositorio de forma muy fácil.

- B. Abre *Github Desktop*. Fíjate que el programa (ver la *Figura 2*) recuerda que teníamos el repositorio que acabamos de borrar y nos informa que no es capaz de encontrarlo en nuestro equipo local. Para solucionar el problema simplemente tenemos que pulsar en *Clone Again*. Hazlo y comprueba que la carpeta ha vuelto a su sitio.

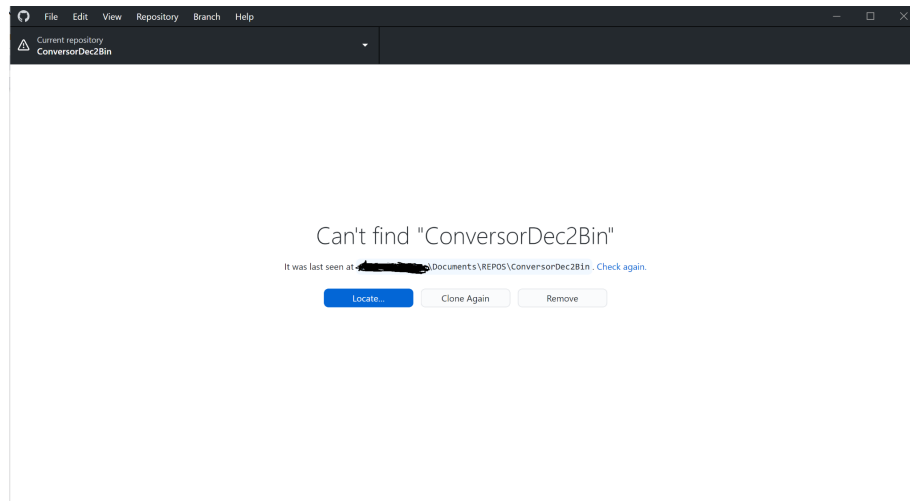


Figura 2: Pantall de *Github Desktop* tras borrar un repositorio local

Ejercicio 2

Vamos a provocar una situación aún más catastrófica.

- A. Vuelve a borrar el repositorio local y cuando te salga la pantalla que se ve en la *Figura 2*, pulsa sobre *Remove*.

Ahora no tenemos ninguna forma de recuperar el repositorio puesto que lo hemos borrado en nuestro equipo (estamos asumiendo que también hemos vaciado la papelera de reciclaje) y *Github desktop* ha perdido el “recuerdo” de haber existido ese repositorio en nuestro equipo.

Afortunadamente tenemos una copia en un servidor web remoto, es decir en *Github.com*.

- B. Ves a tu cuenta de *Github.com* y localiza el repositorio. Para recuperar el código tienes varias opciones. Todas ellas están incluidas en el botón *Code* (de color verde). Puedes descargarte el código en un zip, usar el comando *git clone* en línea de comandos o lo más fácil, usar *Github desktop*. Pulsa en “*Open with Github desktop*”.
- C. Se abrirá el programa y te preguntará dónde quieres clonar el repositorio. Lo más razonable es clonarlo en la carpeta de repositorios de nuestro equipo local. Clona el repositorio pulsando en *Clone*.

Ahora tendrás el repositorio de nuevo en tu equipo local.

Ejercicio 3

Los repositorios tienen un fichero *Readme.md* (md es para indicar que es un markdown) que sirven para realizar una presentación del contenido del repositorio. Es muy importante rellenar correctamente el fichero para que cuando alguien visite nuestro repositorio, pueda entender en qué consiste el repositorio, como instalar el software, como se usa, etc.

En los dos siguientes enlaces hay información sobre el formato de los markdown:

- [Sintaxis de escritura y formato básico](#)
- [Markdown Cheatsheet](#)

Realiza los siguientes ejercicios:

- A. Consulta cómo se crean los ficheros markdown en los enlaces anteriores.
- B. Entra en tu repositorio de la práctica desde github.com. Crea un archivo que se llame *Readme.md*
- C. Modifica el fichero *Readme.md* para que explique correctamente el código incluido en el repositorio. Utiliza tus propias palabras o incluso puedes utilizar texto falso (<https://es.lipsum.com/>). El fichero deberá incluir al menos:
 - a. Título, subtítulo
 - b. palabras en negrita, cursiva, subrayado,...
 - c. una lista (puede ser también una lista de tareas)
 - d. una imagen y un emoji
 - e. una alerta
 - f. una línea horizontal

Ejercicio 4

A partir de este momento vamos a aprender a trabajar en equipo. Los ficheros que debes utilizar en este ejercicio están en el Aula Virtual.

- A. Forma un equipo de 3 personas. También puede ser de dos personas.

Un miembro del equipo hará de jefe/a del equipo “J”, los otros dos de miembros del equipo de trabajo “M1” y “M2”.

- B. “J” tiene que crear un repositorio nuevo e incluir el fichero *dec2bin.py*.

Para que “M1” y “M2” puedan realizar cambios en el repositorio “J” tiene que darles de alta como colaboradores. Para ello, en *Github.com*, accede al menú

Settings> Manage access> Invite a collaborator

Hay que añadir el *username* de “M1” y “M2”.

- C. “M1” y “M2” tienen que clonar el repositorio creado por “J”.
- D. “M1” tiene que incluir en el repositorio el programa en python *bin2dec.py* que convierta de binario a decimal.
- E. “M2” tiene que incluir en el repositorio el programa en python *bin2hex.py* que convierta de binario a hexadecimal.
- F. “J” tiene que modificar el fichero *Readme.md* para que incluya los cambios.

Si cada miembro del equipo toca un fichero diferente no se producirá ningún conflicto. Pero si no es el caso, se produce un conflicto que hay que resolver.

- G. Conforme los tres miembros del equipo acaban sus partes, tienen que hacer *Commit* y *Push* para actualizar el repositorio remoto.
- H. Para actualizar el repositorio local hay que hacer **Pull**. Por lo tanto, los tres miembros del equipo tienen que hacer *Pull* para actualizar los repositorios.

Fíjate que el concepto de *Push* (empujar) es desde el repositorio local al remoto, mientras que *Pull* (atraer) es desde el repositorio remoto al local.

Si todo ha ido bien, tanto el repositorio remoto del equipo como los repositorios locales de cada miembro del equipo han de contener los tres ficheros python y el fichero *Readme.md* actualizado.

Ejercicio 5

- A. Rellena la **tarea del aula virtual** con el url del repositorio que has creado en el ejercicio 4.

Evaluación

La puntuación de esta sesión se obtendrá según el número de ejercicios correctamente realizados. En concreto:

- 6 puntos por realizar los ejercicios 1 y 2.
- 8 puntos por realizar también el ejercicio 3.
- 10 puntos por realizar también los ejercicios 4 y 5.