
 Universitat Autònoma de Barcelona	Fonaments d'Enginyeria 
Curs acadèmic: 2019-2020	Codi assignatura: 103805
Pràctica 2	Centre: Escola d'Enginyeria

Objetivos

- Aprender a usar Git en entorno local y distribuido
- Ser capaz de crear, juntar y gestionar ramas
- Adquirir las habilidades de trabajar en equipo con Git.

Introducción

Esta segunda práctica tiene por objetivo trabajar con el sistema de control de versiones **Git**, así como desarrollar los conocimientos de **Bash** y descubrir nuevos potenciales de estas herramientas.

La práctica tiene como punto de partida una aplicación desarrollada para FakeNetflix que ya incorpora los requisitos de la primera práctica, es decir, **vuestro proyecto inicial (rama master del proyecto) será vuestra versión final de la práctica 1.**

Esta práctica (práctica 2), consistirá en desarrollar nuevas funcionalidades a vuestra práctica 1 utilizando el sistema de control de versiones Git para controlar y gestionar las modificaciones hechas.

Cada equipo de 2 alumnos dispondrá de un repositorio privado en GitHub donde podrá subir el código inicial del proyecto y todas las modificaciones que se vayan realizando.

IMPORTANTE:

- Siempre que sea posible, se recomienda que cada equipo de prácticas traiga un portátil para poder practicar el trabajo remoto en equipo durante la sesión.

Modo de trabajo

El modo de trabajo será el siguiente:

- Añadir al menú principal de vuestra aplicación 4 opciones más, una por cada funcionalidad que hay que añadir.
- Cada nueva funcionalidad de nuestra aplicación se desarrollará en una rama diferente de vuestro proyecto, como veremos a continuación, esto implica que tendréis como mínimo 4 ramas de desarrollo diferentes a la rama `master`.
- Una vez finalizada una funcionalidad de forma correcta, hay que fusionarla (`merge`) con la rama `master` y hacer `push` para que el colaborador/a (compañero/a de prácticas) del proyecto pueda acceder a los cambios realizados.

Funcionalidades:

1.- Sincronización automática de contenidos

El director de FakeNetflix se ha dado cuenta que debe poder distribuir de forma automática entre todos sus usuarios el catálogo actualizado de las películas, por lo que nos pide que añadamos a nuestra aplicación la opción de Sincronización.

En la URL que viene a continuación, estará siempre publicada la última versión de la base de datos estructurada (archivo csv) para ser descargada en cualquier instante que sea necesario: <https://github.com/acocauab/practica2csv>

- Cada vez que arranque el programa, se deberá comprobar si el archivo publicado es diferente del que tenemos guardado en local en nuestro repositorio, si la base de datos obtenida difiere de la que tenemos en nuestro repositorio, habrá que actualizar el archivo local.

Pistas: Explorar los siguientes comandos wget, diff

2.- Separación anual automática

El director se ha dado cuenta que le gustaría poder explorar de forma más sencilla las películas que tenemos en la base de datos (netflix.csv). Los requerimientos de esta nueva funcionalidad son:

- Se deberá generar un directorio por cada año en el que haya alguna película en la base de datos. Es decir, por ejemplo, si en nuestra base de datos solo tuviésemos películas de los años 1980, 2001 y 2018, se deberían crear tres directorios, uno por cada año, con los nombres 1980, 2001 y 2018.
- En dichos directorios deberemos encontrar un archivo csv que tenga por nombre "año-nombre_original.csv", donde nombre_original es el nombre del archivo csv que usamos como base de datos. Este archivo incluirá todas las películas el año de las cuales, coincida con el año del directorio. Es decir, para el ejemplo anterior y, suponiendo que el nombre del fichero de la base de datos global es netflix.csv, el directorio 1980 tendría un único fichero csv con el nombre 1980-netflix.csv. En el directorio 2001, habría un fichero con el nombre 2001-netflix.csv y, finalmente, en el directorio 2018 existiría un fichero con el nombre 2018-netflix.csv.

Pistas: explorar los comandos mkdir, grep, awk.

3.- Alta de nuevas películas

Para esta funcionalidad, los requerimientos del director son:

- Cada usuario debe poder añadir películas al csv que alberga localmente su aplicación, para ello, deberemos añadir una opción que nos permita añadir nuevos registros a nuestra base de datos.

- Cuando se seleccione la opción, el sistema nos debe solicitar todos y cada uno de los campos del csv y, al final, añadir la información completa de la película (el nuevo registro) en la base de datos (ficheros csv).
- Estas altas, si se descarga un nuevo catálogo, van a ser desechadas.

Pistas: uso de archivos temporales.

4.- Eliminación de registros

Deseamos poder eliminar registros de la base de datos seleccionados por título, para ello:

- Se le solicitará al usuario la introducción de una cadena de caracteres y todos aquellos registros en los cuales el campo título tenga alguna coincidencia (aunque sea parcial) con el texto introducido, serán eliminados de la base de datos.
- Listar las coincidencias encontradas.
- Ante de eliminar definitivamente la información de la base de datos, se pedirá al usuario una doble confirmación para asegurarnos que está seguro de realizar una operación que implica eliminar datos.
- Si se confirma la operación, eliminaremos el/los registro/s correspondientes de la base de datos.
- Estas eliminaciones, si se descarga un nuevo catálogo, van a ser desechadas.

DESGLOSE POR SESIONES:

Sesión 1:

Para esta sesión:

- Todos los alumnos deben tener una cuenta de GitHub (github.com)

Durante la sesión:

- Se os proporcionará el enlace para conectaros a la classroom de GitHub correspondiente y obtener la url de vuestro repositorio.
- Además, se os darán las indicaciones necesarias para formar los equipos en la classroom de GitHub y, así, poder trabajar de forma colaborativa en vuestro repositorio.
- Iniciar la implementación de la Funcionalidad 1.

Sesión 2:

Para esta sesión hay que traer:

- la Funcionalidad 1 (Sincronización) funcionando.

Durante la sesión:

- se trabajará en el resto de las Funcionalidades. El orden de desarrollo de las Funcionalidades 2, 3 y 4 lo determinará el equipo de desarrollo.

Sesión 3:

Para esta sesión hay que traer:

- **dos** Funcionalidades más implementadas y funcionando correctamente. Estas dos funcionalidades las seleccionará el equipo de desarrollo.

Durante esta última sesión:

- se trabajará en la Funcionalidad pendiente de finalizar.

Entrega de la práctica

Código y actividad en el repositorio:

El repositorio privado de cada equipo de prácticas en GitHub se congelará el día de la entrega de la práctica, así pues, no hace falta entregar ningún código vía Campus Virtual ya que se evaluará el contenido del repositorio de GitHub al completo.

Informe de la práctica:

El informe de esta práctica deberá seguir los contenidos definidos en el documento `Directrius_Avaluació_Pràctiques_FE.pdf`. En este caso, el informe de la práctica será un **fichero de texto llamado README.md** que deberá estar presente en vuestro repositorio, en el directorio raíz del mismo.

Evaluación de la práctica

Como se comenta en el documento `Directrius_Avaluació_Pràctiques_FE.pdf` publicado en Campus Virtual, en este apartado se desglosa como se puntuará el 60% de la nota de la práctica correspondiente al Funcionamiento del Código:

- Sincronización : hasta 1 punto
- Separación anual : hasta 1 punto
- Alta : hasta 1 punto
- Eliminación : hasta 1 punto
- Por cada rama/funcionalidad fusionada correctamente: hasta 1 punto (máximo 4 puntos)
- Actividad continua en el repositorio por parte de **los dos miembros** del equipo: hasta 2 puntos.

Fecha de entrega de la práctica

El repositorio se cerrará de forma automática el día 20 de diciembre de 2019 a las 23:00. Para esta entrega **NO EXISTE PERIODO DE GRACIA**, es decir, todos los repositorios se congelarán simultáneamente en la misma fecha, siendo la fecha límite de entrega de la práctica un “**hard deadline**”.