CFGM SMX
MP08 – APLICACIONS WEB

GIT : Usat amb GitHub

Git és un software de control de versions, és a dir que manté un registre dels canvis realitzats als fitxers guardats en un repositori de codi, així que també en proporciona còpies de seguretat. Git facilita el treball col·laboratiu sobre el mateix codi font, ja que permet treballar amb una còpia local per unir-lo a la branca principal un cop finalitzat.

Github és una aplicació que utilitza Git. Proporciona hosting gratuït i accés online al teu codi. A més d'una interfície més vistosa que la que ofereix la terminal. GitHub facilita la compartició de codi entre programadors de tot el món.

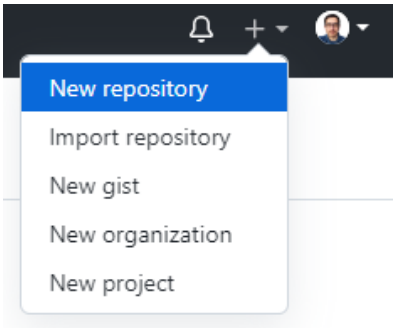
Git és el sistema de control de versions més usat al món i per altra banda, GitHub és el servei d'allotjament online de codi font més usat al món.

Per on comencem?

Primerament hem d'instal·lar Git al teu ordinador, podem descarregar-lo des de la seva pàgina oficial: https://git-scm.com/downloads

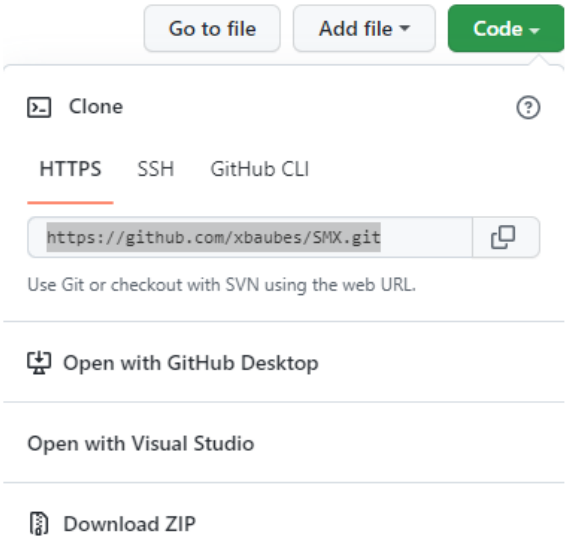
I com que l'usarem a través de GitHub ens hi crearem un compte, aquesta n'és la pàgina web: https://github.com/

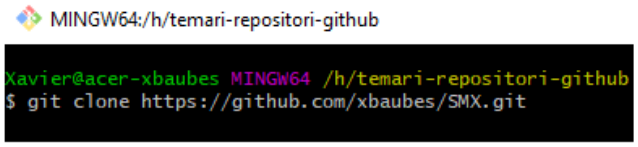
I com funciona?



Es crea un repositori per cada projecte o tema.
Nosaltres crearem el repositori pel nostre codi des de GitHub.
De nom li posarem: *HTML*, ja que hi guardarem el codi de la pràctica d'HTML que ja hem dissenyat.
El repositori el mantenim com a *public*, ens interessa que tothom hi pugui accedir, ja que es tracta de mostrar els projectes que som capaços d'elaborar.

Ara farem una còpia local del repositori al nostre escriptori.
Per fer-ho, copiem la URL del repositori creat al GitHub.
Obrim la terminal git al directori on volem guardar el repositori local.
I hi introduïm la següent instrucció: *git clone [URL repositori]*. Aquesta instrucció copiarà el repositori indicat al directori on ens trobem. El repositori remot l'hem creat al pas anterior, de manera que actualment està buit, a continuació hi afegirem el nostre projecte HTML.





Entrem al directori copiat on només hi ha un fitxer ocult: *.git*.
Hi creem una carpeta de nom *practicaBasica* i a dins hi copiem la pràctica HTML.
Si introduïm ara la instrucció *git status* veurem que detecta el canvi i el marca en vermell tot indicant que caldrà fer un *git add* per poder-lo pujar al repositori remot. Fem-ho!
git add [ruta fitxer] : Guardem l'estat del fitxer indicat.
git add . : Guardem l'estat de tots els fitxers modificats.

Confirmem els canvis realitzats amb la instrucció: *git commit -m «[detalls dels canvis realitzats]»*.
Els canvis queden guardats i preparats per ser pujats.

Per pujar-lo definitivament al repositori remot usem la instrucció: *git push*.
D'aquesta manera ja has guardat al teu codi en un repositori online i de públic accés.

I si treballem en equip?

És recomanable no treballar directament sobre la branca principal (o main) en el cas que vàries persones treballin en el projecte. En aquests casos és recomanable que cada programador creï la seva pròpia branca per cada tasca a desenvolupar.
Una branca és una bifurcació d'una branca partint del codi que hi havia en aquell moment a la branca. Permet desenvolupar codi en paral·lel a la principal durant el temps que faci falta ja que no efecte a aquesta, i a més en facilita la unió amb la principal, o amb d'altres, un cop finalitzat el seu cicle de vida.

<u>Seqüència de la creació de branques:</u>
git pull : Importem els canvis realitzats al repositori remot al local per tal de tenir la versió del codi més actualitzada possible.
git branch : Veiem les branques del projecte i en quina ens trobem.
git branch [nom branca] : Creem una branca amb el nom indicat.
git checkout [nom branca] : Ens situem a la branca amb el nom indicat.

Seqüència de la fusió de branques:
Per fusionar la nostra branca de treball amb la main primer ens ubiquem a main: *git checkout main*. I introduïm la següent comanda: *git merge [nom branca de treball]*.

El treball en grup pot produir conflictes a l'hora d'executar el merge si s'ha modificat la mateixa línia de codi. Si els conflictes produïts no es poden resoldre automàticament ens tocarà escollir la versió definitiva manualment.

Esquema de conflicte de branques:

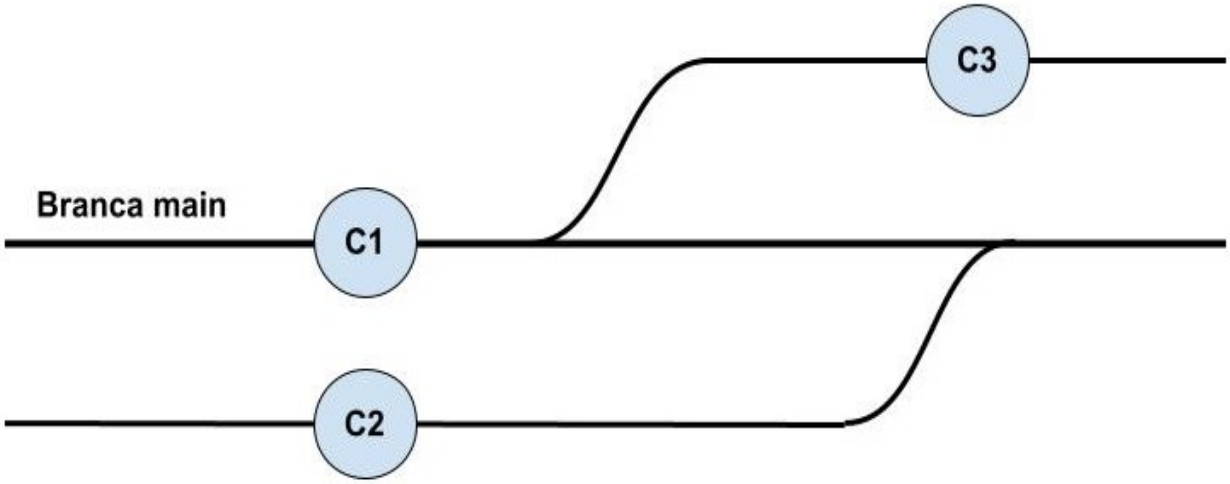


Figura 1. Les rodones representen commits i les línies branques.

Analitzem la *Figura 1*. En el cas que els commits C1 i C2 modifiquin la mateixa línia es pot haver produir un conflicte en el merge realitzat entre la branca inferior i la main. De la mateixa manera es podria produir amb el merge entre la branca superior i main si els commits C2 i C3 es solapen.

Metodologia de treball amb git

És una bona metodologia de treball desenvolupar el vostre codi des del repositori local, que podeu guardar en un pendrive si voleu treball amb varis dispositius. És important fer pujades periòdiques del codi al repositori remot, per tal de minimitzar les pèrdues en cas d'accident.

Us recomano que tots els treballs que realitzeu durant aquest curs els pugeu a GitHub, per tal de tenir-ne una còpia de seguretat, construir el vostre portfoli professional i familiaritzar-vos amb l'eina.

BIBLIOGRAFIA I WEBGRAFIA

«HubSpot». https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
«Atlassian». https://www.atlassian.com/es/git/tutorials/git-ssh

---