

Git: mooc_git-entrega4_branch

Objetivo

Usar dos repositorios remotos, uno como respaldo del repositorio de trabajo local y otro como repositorio central donde las contribuciones se envían con pull-request. Además se practicará con ramas remotas de diversos tipos y se seguirá realizando integraciones de desarrollos sencillos.

Resumen de la práctica

Duplique el repositorio https://github.com/jquemada/cal_2com en su cuenta de GitHub usando el botón de Fork. Este repositorio se creó en el Tema 3 con 2 commits en la rama **master**. El primer commit añade los ficheros README.md y LICENSE, y el segundo commit añade el fichero calculator.html con una calculadora que tiene solo el botón x^2 .

Cree una organización o una segunda cuenta con un nombre diferente (<su_cuenta_2>) en GitHub. Duplique el repositorio (https://github.com/<su_cuenta>/cal_2com), ya clonado en su primera cuenta, en esta última.

Clonar el repositorio cal_2com de <su_cuenta_2> en un repositorio local en su ordenador.

→ Cree un nuevo commit en la rama **master** del repositorio local que añada al fichero calculator.html una cabecera HTML (<h1>) al comienzo del body con el texto "Calculadora de <nombre y apellidos>".

Copiar la rama **inverse** del repositorio https://github.com/jquemada/cal_branches a su repositorio local. Esta rama tiene 3 commits, el primero y el segundo son los mismos que los dos primeros de la rama **master** de cal_2com y el tercer añade el botón $1/x$ a la calculadora.

Integrar primero la rama **master** en la rama **inverse**. La integración debe incorporar el título <h1> de la rama **master** y los dos botones x^2 y $1/x$ de la rama **inverse**.

Integre a continuación también la rama **inverse** en la rama **master** (con FF). Subir ambas ramas al repositorio origen de la clonación en la segunda cuenta.

Una vez subidas envíe la integración realizada en la rama **master** con pull-request desde este repositorio al repositorio en la primera cuenta, del que copió este Fork. Acepte el pull-request e intégrele en el repositorio en GitHub en su primera cuenta.

Prueba de la práctica

Para comprobar que la práctica ha sido realizada correctamente hay que utilizar el validador de este repositorio

https://github.com/practicas-ging/mooc_git-entrega4_branch

Recuerde que para utilizar el validador se debe tener node.js (y npm) (<https://nodejs.org/es/>) y Git instalados. El proyecto se descarga, instala y ejecuta en el ordenador local con estos comandos:

```

$                                     ## El proyecto debe clonarse en el ordenador local
$ git clone https://github.com/practicas-ging/mooc_git-entrega4_branch
$
$ cd mooc_git-entrega4_branch         ## Entrar en el directorio de trabajo
$
$ npm install                         ## Instala el programa de test
$
$ npm run checks [nombre_de_la_cuenta] [nombre_de_la_cuenta_2]
.....                               ## Pasa los tests sobre el repositorio en github
.....                               ## indicando que partes están correctamente
.....                               ## implementadas y cuales no.
... (resultado de los tests)
$

```

Debe cambiar [nombre_de_la_cuenta] y [nombre_de_la_cuenta_2] por el nombre de sus cuentas en GitHub.

Entrega y evaluación

La entrega consiste en subir a MiriadaX únicamente el **nombre de las dos cuentas** en GitHub (<su_cuenta> y <su_cuenta_2>). Recuerde que **antes de hacer** la **entrega** debe haber **actualizado ambos repositorios** con lo que **solicita la practica**.

¡Cuidado! Compruebe que el nombre de la cuenta subido es el correcto y que el repositorio esta actualizado con la última versión.

El evaluador debe comprobar que en ambas cuentas está el **repositorio cal_2com**. A ambos repositorios se se debe haber añadido todos los commits que se han especificado en la practica. En el último commit de master el repositorio cal_2com incorpora un título con el nombre de la persona que hace la entrega y todos los botones incorporados.

RÚBRICA: La resolución de cada uno de estos puntos dará un el % indicado de la nota total:

- 10%: Existe el repositorio cal_2com en cuenta_2
- 10%: Existe el repositorio cal_2com en cuenta_1 y es el origen del Fork del repo de cuenta _2
- 10%: La rama master del repo de cuenta_2 tiene como tercer commit "Título con autor"
- 10%: El repo de cuenta_2 tiene una rama **inverse** integrada con master
- 10%: La rama master del repo de cuenta_2 tiene como último commit "1/x button"
- 20%: Que el fichero calculator.html de este último commit funciona, tiene el título requerido y los botones x^2 y $1/x$
- 30%: Que el repositorio cal_2com en cuenta_1 tiene una rama master con los mismos commits de master del repo cal_2com de cuenta _2

El objetivo de este curso es sacar el máximo provecho al trabajo que están dedicando. Les recomendamos que utilicen la evaluación para ayudar a sus compañeros enviando comentarios sobre la corrección del código, su claridad, legibilidad, estructuración y documentación. Dado que es un curso para principiantes, ante la duda les pedimos que sean benevolentes con sus compañeros, porque hay participantes que empiezan y los primeros pasos siempre son difíciles.

¡Cuidado! Una vez enviada la evaluación, está no se puede cambiar. Piensen bien su evaluación antes de enviarla.

Pasos a seguir en el desarrollo de la práctica

Paso 1) Copiar el repositorio `https://github.com/jquemada/cal_2com` en su cuenta de GitHub usando el botón de Fork.

Es el nombre del usuario en Github ya utilizado, el que debe incluir el nombre y apellidos, o las iniciales de la persona que realiza la entrega.

Paso 1bis) Crear una segunda cuenta (<su_cuenta_2>) en GitHub y copiar el repositorio de su primera cuenta (`https://github.com/<su_cuenta>/cal_2com`) en esta segunda usando el botón de Fork. El nombre de esta cuenta puede ser cualquiera.

Paso 2) Clonar el repositorio `<su_cuenta_2>/cal_2com` (de su cuenta en GitHub) en su ordenador local.

Sugerencia de comandos a utilizar

```
$ git clone https://github.com/<su_cuenta_2>/cal_2com
```

Paso 3) Entrar en el directorio de trabajo del nuevo repositorio clonado y añadir al fichero `calculator.html`, al principio del bloque `<body>` de HTML, un título `<h1>` con su nombre y apellidos. Cerrar un nuevo commit "Título con autor" con esta modificación.

Sugerencia de comandos a utilizar (además del editor):

```
$ cd cal_2com
$ git status ...
$ git log ...
$ git diff ....
$ git add <ficheros> ...
$ git commit -m "Título con autor"
```

Paso 4) Copiar la rama remota `inverse` del repositorio `https://github.com/jquemada/cal_branches` como una rama local con git `fetch` y la `refspec` correspondiente utilizando el comando: `git fetch`

Sugerencia de comandos a utilizar:

```
$ git fetch https://github.com/jquemada/cal_branches Remota : Local inverse:inverse
$ git branch ...
```

Paso 5) Integrar en la rama `inverse`, copiada en el repositorio local, la rama `master` con (`git merge`) e identificar el nuevo commit con el mensaje "Integrar inverse y master". La integración debe incorporar el título y los dos botones x^2 y $1/x$.

DENTRO DE INVERSE

`git merge master`

Sugerencia de comandos a utilizar: Habrá conflictos así que se edita a mano

```
$ git merge -m "Integrar inverse y master" ...
$ git merge --continue
$ git commit -m "Integrar inverse y master" ...
$ git branch ...
$ git log --graph ...
```

Paso 6) Integrar en master la rama inverse del repositorio local (integrada en el paso anterior) con git merge también.

Sugerencia de comandos a utilizar:

\$ git merge ...

\$ git branch ...

\$ git log --graph ...

Paso 7) Subir al repositorio origin (origen de la clonación en la segunda cuenta) las dos ramas locales (**master** e **inverse**) con git push.

Sugerencia de comandos a utilizar:

\$ git branch ...

\$ git push --all ...

Paso 8) Enviar un pull-request de la rama **master** desde el repositorio cal_2com de su segunda cuenta al repositorio cal_2com de su primera cuenta desde el que creó este segundo con Fork.

Se deben utilizar las facilidades de GitHub para enviar un pull-request, porque es la única forma de hacerlo..

Paso 9) Aceptar el pull-request. Github permite aceptar el merge asociado a un pull-request directamente en Github.

También se puede clonar el repositorio de la primera cuenta en local. Descargar la rama master asociada al pull-request con "git fetch" e integrarla en la rama master del repositorio local. Volver a subir la rama master ya integrada al repositorio en la primera cuenta.

Utilizar los comandos ya vistos en los pasos anteriores y en las transparencias.