

TALLER 2B

Objetivo

Al finalizar el taller deberá estar en capacidad de entender el trabajo de ramas de un repositorio usando distintas formas de integrar trabajo en la rama principal

Consigna

En los siguientes ejercicios, clonará proyectos con distintas ramas, cada uno de los puntos tiene instrucciones específicas.

En todos los ejercicios haga el diagrama de versiones para entender cómo se secciona el proyecto

Ejercicio 1

En el repositorio

<https://github.com/Domiciano/PasaporteGITEjercicioG>

Haga merge de las 4 branches de trabajo **funcion<n>**. Únalas a master en cada uno de los 3 casos. Reflexione por qué razón el primer commit es *fast-forward* y los demás son *recursive strategy*.

Reflexione además por qué NO causa conflictos

Ejercicio 2

En el repositorio

<https://github.com/Domiciano/PasaporteGITEjercicioH>

Haga merge de las 3 branches de trabajo **funcion<n>**. Únalas a master en cada uno de los 3 casos.

Realice el ejercicio con los siguientes objetivos:

1. Index.html debe tener una lista de números en ingles únicamente en etiquetas **h1**. NO en español
2. Index.html debe tener una lista de letras griegas en etiquetas **h2**.
3. Index.html debe tener una lista de deportes en etiquetas **h3**.
4. Index.html debe tener una lista de géneros musicales en etiquetas **h4**.

Todo esto ya está escrito en las branches de función. Su trabajo únicamente es mezclar las funciones en master y editar el archivo de ser necesario si surgen conflictos.

Ejercicio 3

En el repositorio

<https://github.com/Domiciano/PasaporteGITEjercicioI>

Que es el mismo repositorio anterior, realice el mismo ejercicio, pero use una estrategia de modo que usted pueda resolver los conflictos en cada una de las ramas *funcion1*, *funcion2*, *funcion3* justo antes de intentar hacer merges con master.

La idea es que mezcle los branches sin poner en peligro a master

Al hacer esto, reflexione por qué cada vez que hace merge de las funciones en master son ahora fast-forward.

NOTAS: Seguro que tendrá conflictos al intentar unir, pero evite que estos conflictos se produzcan estando parado en master.

Ejercicio 4

En el repositorio

<https://github.com/Domiciano/PasaporteGITEjercicioI>

Existe la rama **master**. **Test** es una branch de **master** y **funcion<n>** son branches de **test**. Adicionalmente hay una rama **hotfix1**, que es branch de la rama **master**. Evidentemente es un árbol de trabajo, de modo que su tarea es mezclar todos los branches. Para eso divida el trabajo:

1. Inicialmente mezcle *funcion1* y *funcion2* en **test**. Luego, mezcle **test** en **master**. Luego de mezcladas, las funciones 1 y 2 deben ser eliminadas
2. Mezcle *funcion3* y *funcion4* en **test**. Luego, mezcle **test** en **master** nuevamente. **funcion3** y **funcion4** deben ser eliminadas.
3. Finalmente, aplique el fix mezclando **hotfix1** a **master**. Luego del merge, **test** habrá quedado desactualizada, entonces use **git pull** para actualizar **test**, respecto a **master**.

En caso de conflictos priorice el código recolectado en las ramas de función por encima de las que puedan estar en **test** y **master**.

Ejercicio 5

En el repositorio

<https://github.com/Domiciano/PasaporteGITEjercicioK>

Existe la rama master. alfa y psi son ramas de master. funcion1 es rama de alfa y funcion2 es rama de psi. Mezcle las funciones en sus respectivas ramas. Al finalizar, elimine las ramas de función.

Luego, tanto alfa como psi, que son ramas de master deben mezclarse con master, para que el trabajo de las funciones llegue a la rama principal.

Finalmente, garantice que tanto alfa como psi queden actualizadas, es decir, que al final el código de master sea igual a alfa y psi.