



UNIVERSIDAD DE COSTA RICA

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

Laboratorio GIT

Estudiante:

Rodolfo Rojas Ocampo

Profesor:

Julian Gairaud Benavides

III Periodo

IE0117 - Programación Bajo Plataformas Abiertas

4 de febrero de 2023

1. Haga un *fork* del repositorio en su repositorio personal, con configuración pública.
2. En una rama nueva, la cual llamará “*feature*”, realice un *commit* individual para cada una para las siguientes acciones:
 - a. Cambie el nombre del objeto gato1. aasdasd
 - b. En medio del objeto gato2 y funcionesTarea.printlista(), cree un gato3 con información diferente a los gatos anteriores. asdasd
 - c. Suba la rama “*feature*” a su repositorio remoto con el mismo nombre. asdasd
3. De vuelta en la rama *main*, cambie el nombre del objeto gato1 (diferente a los anteriores)
4. Haga un *merge* de la rama *feature* hacia la rama *main* (en otras palabras, incluya *feature* en *main*)
5. Tome un screenshot de su editor de texto mostrando los conflictos de ambas ramas

```
vboxuser@cokin:~/LABGIT/gatosGit$ git merge feature
Actualizando 1640128..12f17f1
error: Los cambios locales de los siguientes archivos serán sobrescritos al fusionar:
    gatosGit/gatos.py
Por favor, confirma tus cambios o aguárdalos antes de fusionar.
Abortando
```

Figura 1: Error al utilizar el comando `git merge feature` luego de cambiar `gatos.py`

6. Resuelva el conflicto causado por el nombre, de forma que mantenga el nombre de gato1 que acaba de cambiaren *main*, pero se incluyan los demás cambios hechos en *feature*

7. Incluyalos cambios en su repositorio remoto, posteriormente corra el comando `git log --oneline --graph` y tome un screenshot

```
vboxuser@cokin:~/LABGIT/gatosGit$ git log --oneline --graph
* 0f2e613 (HEAD -> main, origin/main, origin/HEAD) Se realiza la union de las
ramas main y feature
| \
| * 12f17f1 (origin/feature, feature) Se agrega un tercer gato
| * ae60854 Cambio de nombre del gato1 a Kira
* | dedc243 Commit antes de hacer merge de main y feature
|/
* 1640128 included gato2 and list printing
* c2b877a corrected mistakes
* d02a6d2 included gato1
* 7ab0adc included functions file
* a13fd08 first commit
```

Figura 2: Utilización del comando `git log --oneline --graph` donde se observa en una linea y de forma gráfica los diferentes commits utilizados

8. Del manual de `git log` que aparece al usar el comando `--help`, tome un screenshot de su terminal mostrando para qué sirve el comando `--graph`, otro mostrando qué significa el comando `--oneline` y otro más mostrando para qué fue "diseñado" `--oneline` (3 screenshots en total para el punto 8)

```
--graph
Draw a text-based graphical representation of the commit history on
the left hand side of the output. This may cause extra lines to be
printed in between commits, in order for the graph history to be
drawn properly. Cannot be combined with --no-walk.

This enables parent rewriting, see History Simplification above.

This implies the --topo-order option by default, but the
--date-order option may also be specified.
```

Figura 3: Revisión de la bandera `--graph`

```
--oneline
This is a shorthand for "--pretty=oneline --abbrev-commit" used
together.
```

Figura 4: Revisión de la bandera `--oneline`

```
• oneline

    <hash> <title line>

    This is designed to be as compact as possible.
```

Figura 5: Parte donde se muestra para que fue diseñado --oneline

9. Tome un screenshot del archivo `gatos.py` resultante corriendo en su computadora

```
vboxuser@cokin:~/LABGIT/gatosGit$ python3 gatos.py
Kira, 4 annos, color: negro
Pelusa, 3 annos, color: cafe
Minichu, 10 annos, color: blanco
```

Figura 6: Corriendo el script modificado de `gatos.py`

10. Haga un *commit* en su repositorio local, en la rama `main`, con un pdf que contenga los screenshots de los puntos 5, 7, 8 y 9

11. Haga un *push* de la rama `main` local a la rama `main` de su repositorio remoto

Referencias