Soluciones: Ejercicios nivel intermedio de Git

1. *Preparación del entorno*

```
$ echo "Pareja de prácticas formada por alumno1 y alumno2" >
README Apellido1(alumno1)_Apellido1(alumno2)
$ git init
$ git add .
$ git commit –m "Entorno de trabajo"
```

2. *Creando una rama*

```
$ git branch Apellido1(alumno1)_Apellido1(alumno2)
$ git branch
```

El * significa que actualmente estamos trabajando/situados en esa rama

```
$ git checkout Apellido1(alumno1)_Apellido1(alumno2)
```

El * se encuentra ahora en la rama Apellido1(alumno1)_Apellido1(alumno2), lo que quiere decir que estamos situados en ésta.

```
$ git checkout –b NombreRamaNueva
```

3. *Navegando y haciendo cambios en las ramas (solo alumno1)*

```
$ echo "Este es el entorno de trabajo de Alumno1 y Alumno2, en
el que se almacenarán los ficheros de la clase de prácticas de
la asignatura Evolución del Software" > README
$ git add README
$ git commit –m "README añadido"
$ git add .
$ git commit –m "Fichero pb.c añadido"
$ ls //Estando en la rama Apellido1(alumno1)_Apellido1(alumno2)
$ git checkout master //Nos cambiamos a la rama master
$ ls //Estando en la rama master
```

Se comprueba que los ficheros README y pb.c no están en la rama master y que el fichero README Apellido1(alumno1)_Apellido1(alumno2)está en ambas.

4. *¿Qué ha sido cambiado?*

```
git whatchanged --format=full
```

5. *Pull y Push, trabajando con un repositorio remoto (alumno1)*

```
$ git pull https://github.com/lorgut/EvolSoft.git master
$ ls //Comprobando el contenido de la rama master
$ git push https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
```

6. *Actualizando y modificando directorio de trabajo*

```
$ git checkout Apellido1(alumno1)_Apellido1(alumno2)
$ git pull https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
$ ls //Comprobando el contenido de la rama
$ emacs pb.c & //Realizamos los cambios
$ git add pb.c
$ git commit –m "Añadiendo argumento –favorite"
$ emacs Makefile & //Creamos el makefile
$ git add .
$ git commit –m "Makefile para compilar pb.c"
$ git push https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
```

7. *Más modificaciones con ramas*

```
$ git pull https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
$ git checkout –b RamaAlumno1
$ emacs pb.c //Realizamos los cambios
$ git add .
$ git commit –m "Función My_sort_func"
$ git push https://github.com/lorgut/EvolSoft.git RamaAlumno1
```

8. *Merge (sin conflictos)*

```
$ git branch RamaAlumno1
$ git pull https://github.com/lorgut/EvolSoft.git RamaAlumno1
$ emacs pb.c //Miramos que estén los cambios
$ git merge RamaAlumno1 //Situados en la rama
Apellido1(alumno1)_Apellido1(alumno2)
$ emacs pb.c //Miramos que se haya fusionado y realizamos los cambios
$ git add .
$ git commit –m "Comentarios"
$ git push https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
```

9. *Merge (sin conflictos) El alumno2 sigue con el siguiente ejercicio*

```
$ git pull https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
$ git checkout RamaAlumno1
$ git merge Apellido1(alumno1)_Apellido1(alumno2)
$ emacs pb.c & //Realiza los cambios
$ git add pb.c
$ git commit –m "Usando el tipo bool"
$ git push https://github.com/lorgut/EvolSoft.git RamaAlumno1
```

10. *Merge (conflictos)*

Alumno2

```
$ emacs pb.c & //Realiza los cambios
$ git add pb.c
$ git commit –m "Arreglando error ortográfico a inglés
británico"
$ git push https://github.com/lorgut/EvolSoft.git RamaAlumno1
```

Alumno1

```
$ git pull https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
$ git checkout RamaAlumno1
$ git merge Apellido1(alumno1)_Apellido1(alumno2)
$ emacs pb.c & //Realiza los cambios
$ git add .
$ git commit –m "Arreglados los conflictos tras la fusión
```

11. Merge y ¿cuántos commits has hecho tú más que yo?

```
$ git pull https://github.com/lorgut/EvolSoft.git RamaAlumno1
$ git checkout Apellido1(alumno1)_Apellido1(alumno2)
$ git merge RamaAlumno1
$ emacs pb.c & //Realiza los cambios
$ git add .
$ git commit –m "Comentario sobre la función de probabilidad"
$ git push https://github.com/lorgut/EvolSoft.git
Apellido1(alumno1)_Apellido1(alumno2)
$ git shortlog –s -n
```

12. *Eliminando ramas remotas*

```
$ git branch
$ git ls-remote --heads https://github.com/lorgut/EvolSoft.git
$ git push https://github.com/lorgut/EvolSoft.git :RamaAlumno1
$ git branch
$ git ls-remote --heads https://github.com/lorgut/EvolSoft.git
```

Para eliminar las ramas locales usaría el comando git branch –d NombreRama

13. ¿Qué commits no son de esta rama?

```
$ git help –w cherry
$ git cherry –v master
$ git cherry –v Apellido1(alumno1)_Apellido1(alumno2)
$ git cherry –v RamaAlumno1
```

14. Información sobre nuestro entorno de trabajo

```
$ git remote show https://github.com/lorgut/EvolSoft.git
$ git checkout master
$ git pull https://github.com/lorgut/EvolSoft.git master
$ ls //Comprobando el contenido de la rama
```

15. Exportar el repositorio

```
$ git archive HEAD --format=zip > Nombrearchivo.zip.
```