



Tema 3: Trabajo con Git

Condiciones de entrega:

La forma de entregar los ejercicios será en un fichero ZIP con el formato **ApellidosNombreTarea3.pdf**.

Puntuación:

El boletín está puntuado sobre 10 puntos. En caso de no entregarlo en fecha, hay una prórroga de 3 días en las que se puede entregar con penalización de 1 punto por día hasta un mínimo de 3.5. A partir del tercer día la tarea será puntuada con un cero.

El ejercicio debe funcionar cuando se ejecute.

La nota mínima para considerar aprobado el boletín es de 3.5 puntos.

Ejercicios:

1. Abriremos el proyecto adjunto Cartas y crearemos los comentarios necesarios para la documentación en **Javadoc**. Generaremos los ficheros HTML y los adjuntaremos.
(3 puntos)
2. Crea una documentación de cómo realizarías los siguientes pasos utilizando los sistemas de control de versiones conocidos (Git y GitHub).
(7 puntos)

NOTA: Debes utilizar pantallazos y explicación de cada paso del proceso.

Pasos a seguir:

1. Crear un repositorio Git con la forma **NombreApellidos**.
2. Crear la carpeta **Tarea3** y meter dentro un fichero **CarnavalCarnaval.txt**.
3. Subir la información al repositorio.
4. Crear un repositorio en GitHub **NombreApellidos_Remoto**.
5. Subir la información al repositorio de GitHub.
6. Crear una rama **Tarea3_RamaNueva**.
7. Crear el fichero **CarnavalSeAcabo.txt** en la rama nueva.
8. Hacer un merge de las dos ramas.
9. Borrar la rama **Tarea3_RamaNueva**.
10. Hacer un fork del siguiente repositorio:
<https://github.com/martingarciafigueira/Tarea3DAW>.
11. Añadir un fichero con el título **Nombre_Apellidos_Fork.txt**.
12. Realizar el pull request.



Antes de proceder a crear los repositorios de Git y realizar esta tarea, hemos de lanzar unos comandos de configuración una vez instalemos Git en nuestro equipo:

```
Microsoft Windows [Versión 10.0.18363.1379]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\hadri>git config --global user.email "hadrianvillarcuadrado@gmail.com"

C:\Users\hadri>git config --global user.name "Hadrián Villar"
```

1. Crear un repositorio Git con la forma NombreApellidos.

Para crear un repositorio de Git, entraremos desde la **consola de comandos** o desde **git bash** y una vez nos localicemos en el directorio nuevo lanzaremos el siguiente comando:

```
D:\Clase DAW\HadrianVillarCuadrado>git init
Initialized empty Git repository in D:/Clase DAW/HadrianVillarCuadrado/.git/
```

Nombre	Fecha de modificación	Tipo
.git	27/02/2021 17:20	Carpeta de archivos

2. Crear la carpeta Tarea3 y meter dentro un fichero CarnavalCarnaval.txt.

Crearemos una carpeta llamada **Tarea3** y añadiremos el archivo "**CarnavalCarnaval.txt**"

Este equipo > Secundario (D:) > Clase DAW > HadrianVillarCuadrado > Tarea3

Nombre	Fecha de modificación	Tipo
CarnavalCarnaval.txt	27/02/2021 20:31	Documento de te...

3. Subir la información al repositorio.

Antes de subir la información al repositorio hemos de añadir todos los ficheros nuevos con el comando: "**git add .**" y posteriormente hemos de hacer el "**git commit**" para subir la información al **repositorio local**

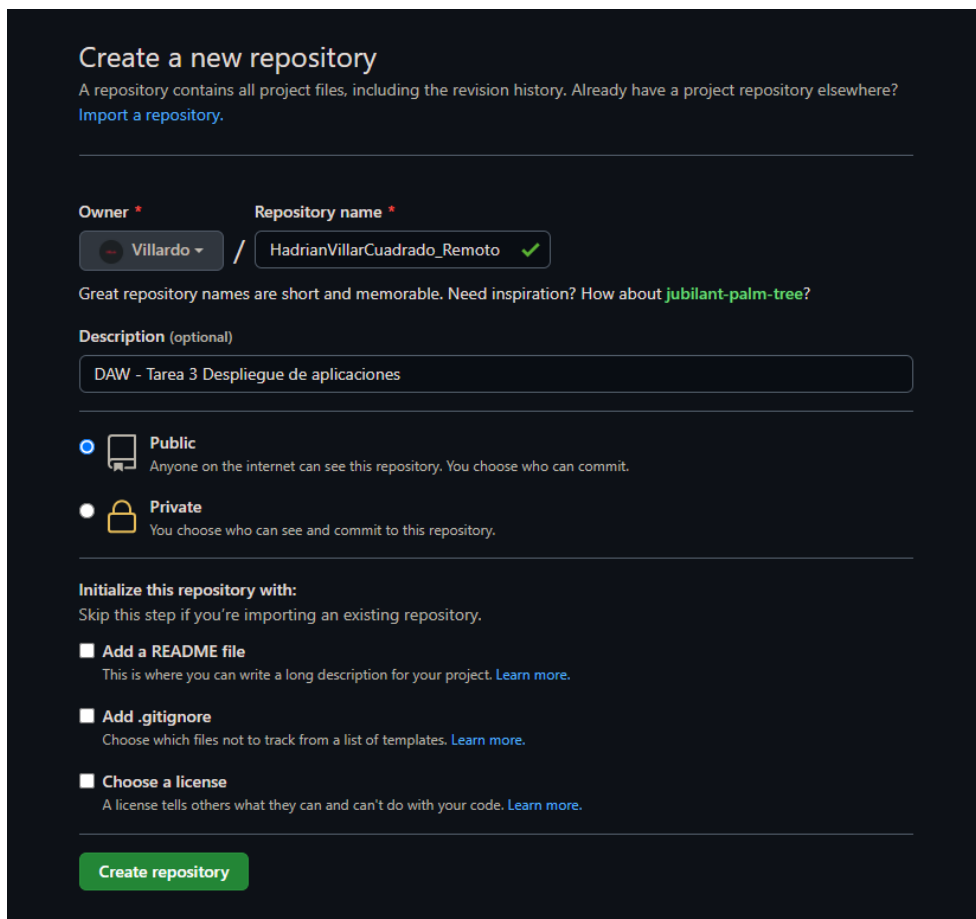
```
D:\Clase DAW\HadrianVillarCuadrado>git init
Initialized empty Git repository in D:/Clase DAW/HadrianVillarCuadrado/.git/

D:\Clase DAW\HadrianVillarCuadrado>git add .

D:\Clase DAW\HadrianVillarCuadrado>git commit -m "Se ha creado la carpeta y añadido el archivo txt"
[master (root-commit) 38373b5] Se ha creado la carpeta y añadido el archivo txt
1 file changed, 1 insertion(+)
create mode 100644 Tarea3/CarnavalCarnaval.txt
```

4. Crear un repositorio en GitHub NombreApellidos_Remoto.

Para crear un repositorio en **GitHub**, nos crearemos una cuenta y una vez iniciemos sesión, le daremos al “+” de la esquina superior derecha y a crear un **nuevo repositorio**. Una vez le demos, nos llevará a esta página donde podremos editar diferentes elementos como los que se muestran a continuación:



5. Subir la información al repositorio de GitHub.

Para subir los ficheros al repositorio de Github “enlazaremos” el repositorio local con el remoto con el comando: **“git remote add origin https://github.com/NuevoRepositorio.git”** y enviaremos todo el contenido con el comando: **“git push -u origin main”**.

```
D:\Clase DAW\HadrianVillarCuadrado>git remote add origin https://github.com/Villardo/HadrianVillarCuadrado_Remoto.git
fatal: remote origin already exists.

D:\Clase DAW\HadrianVillarCuadrado>git branch -M main

D:\Clase DAW\HadrianVillarCuadrado>git push -u origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': Villardo
Password for 'https://Villardo@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Writing objects: 100% (4/4), 332 bytes | 83.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Villardo/HadrianVillarCuadrado_Remoto
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```



6. Crear una rama Tarea3_RamaNueva.

Para crear una nueva rama del repositorio utilizaremos el comando de a continuación “**git branch nuevaRama**”. Una vez lo ejecutemos, creará la nueva rama, para poder comprobarlo escribiremos “**git branch**”. Nos indicará en **verde** la rama actual.

```
D:\Clase DAW\HadrianVillarCuadrado>git branch Tarea3_RamaNueva

D:\Clase DAW\HadrianVillarCuadrado>git branch
  Tarea3_RamaNueva
* main

D:\Clase DAW\HadrianVillarCuadrado>_
```

7. Crear el fichero CarnavalSeAcabo.txt en la rama nueva.

Ahora crearemos otro archivo de texto llamado **CarnavalSeAcabo.txt** y lo adjuntaremos a la nueva rama. En mi caso, me da un error con el inicio de sesión y he de loguearme desde cmd otra vez. Ya creado y adjuntado a la rama mediante un “**push**”, volveremos a hacer un **commit** y un **push** para añadir los ficheros al repositorio remoto de **GitHub**.

```
D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git push -u origin Tarea3_RamaNueva
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': Villardo
Password for 'https://Villardo@github.com':
Everything up-to-date
Branch 'Tarea3_RamaNueva' set up to track remote branch 'Tarea3_RamaNueva' from 'origin'.

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git commit -m "Añadido el segundo archivo de texto"
[Tarea3_RamaNueva 48a50a4] Añadido el segundo archivo de texto
 1 file changed, 1 insertion(+)
 create mode 100644 Tarea3/CarnavalSeAcabo.txt

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git push -u origin Tarea3_RamaNueva
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': Villardo
Password for 'https://Villardo@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 328 bytes | 164.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Villardo/HadrianVillarCuadrado_Remoto
 38373b5..48a50a4 Tarea3_RamaNueva -> Tarea3_RamaNueva
Branch 'Tarea3_RamaNueva' set up to track remote branch 'Tarea3_RamaNueva' from 'origin'.

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>_
```

8. Hacer un merge de las dos ramas.

Para hacer el **merge** y así fusionar las 2 ramas, nos situaremos en la rama **principal** y decidiremos con que rama lo fusionaremos.

```
D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git merge Tarea3_RamaNueva -m "Hacemos merge de Tarea3_RamaNueva a main"
Already up to date.

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git commit -m "El merge se ha llevado a cabo desde la rama Tarea3_RamaNueva"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>_
```



9. Borrar la rama Tarea3_RamaNueva.

Borraremos la rama **Tarea3_RamaNueva** tanto del repositorio **local**, como **remoto** con los siguientes comandos:

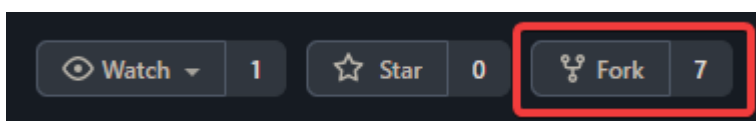
```
D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git branch -d Tarea3_RamaNueva
Deleted branch Tarea3_RamaNueva (was 48a50a4).

D:\Clase DAW\HadrianVillarCuadrado\Tarea3>git push origin --delete Tarea3_RamaNueva
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': Villardo
Password for 'https://Villardo@github.com':
To https://github.com/Villardo/HadrianVillarCuadrado_Remoto
- [deleted]          Tarea3_RamaNueva

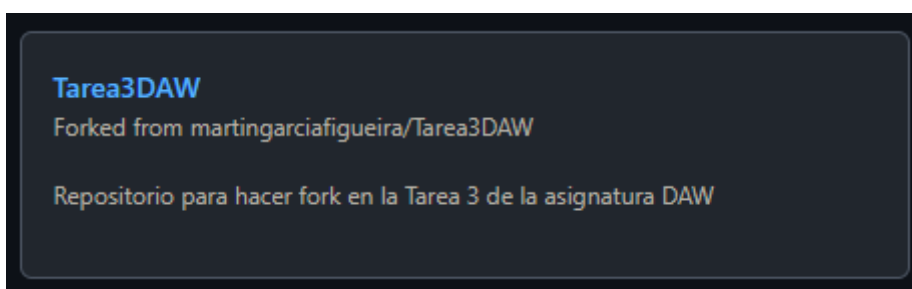
D:\Clase DAW\HadrianVillarCuadrado\Tarea3>
```

10. Hacer fork de un repositorio:

Lo primero que hay que hacer, es entender qué es un **fork**. Un fork, es una copia exacta del repositorio publicado en **GitHub**. Para hacerlo, hemos de localizar el siguiente botón en el repositorio del cual queremos hacer la copia en la esquina superior derecha.



Una vez pulsemos el botón del **fork**, en nuestros repositorios nos aparecerá uno nuevo como se muestra a continuación:





11. Añadir un fichero con el título Nombre_Apellidos_Fork.txt.

Para añadir el fichero deseado, clonaremos el repositorio **remoto** a **local** mediante el comando **"git clone rutaRepositorioFork"**. Una vez descargados todos los archivos modificaremos a gusto y enviaremos mediante un **commit** y un **push** otra vez al repositorio remoto de Github:

```
D:\Clase DAW>git clone https://github.com/Villardo/Tarea3DAW.git
Cloning into 'Tarea3DAW'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 2), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (15/15), 4.08 KiB | 2.00 KiB/s, done.
D:\Clase DAW>
```

```
Directorio de D:\Clase DAW\Tarea3DAW
27/02/2021 18:56 <DIR>      .
27/02/2021 18:56 <DIR>      ..
27/02/2021 18:56           0 Alberto_GonzalezRodriguez_Fork.txt
27/02/2021 18:56           9 Alejandro_AcuñaOtero_Fork.txt
27/02/2021 18:51           0 Hadrian_VillarCuadrado_Fork.txt
27/02/2021 18:56           2 Maya_Reñones_Francisco_Fork.txt
27/02/2021 18:56          77 README.md
                    5 archivos      88 bytes
                    2 dirs 607.365.160.960 bytes libres

D:\Clase DAW\Tarea3DAW>git commit -m "Añadido el archivo txt"
[main b1a38f8] Añadido el archivo txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Hadrian_VillarCuadrado_Fork.txt

D:\Clase DAW\Tarea3DAW>git push origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': Villardo
Password for 'https://Villardo@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 277 bytes | 277.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Villardo/Tarea3DAW.git
   661a7df..b1a38f8  main -> main

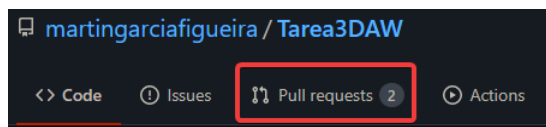
D:\Clase DAW\Tarea3DAW>
```



12. Realizar el pull request.

¿Qué es un **pull request**? Se trata de una serie de cambios realizados en un repositorio al que hemos hecho fork y queremos proponérselos al dueño para que los aplique.

Para ello hemos de ir al repositorio al cual hemos hecho fork y debemos presionar el botón que se muestra a continuación:



Nos llevará a la siguiente página donde podremos comparar los cambios entre las diferentes versiones y generar dicha pull request que le llegará al propietario el cual ha de aceptar las versiones que proponen los demás.

