

## **PRÁCTICA 2: CONTROL DE VERSIONES CON GIT**

### **PARTE 1: GIT Y GITLAB**

#### **ENLACES WEB A LOS REPOSITORIOS:**

URL web repositorio ejercicio 1: <https://gitlab.etsit.urjc.es/aalberto/texmex-lsis>

URL web repositorio ejercicio 2: <https://gitlab.etsit.urjc.es/aalberto/upstream>

URL web repositorio ejercicio 3 (repositorio): <https://github.com/aleon2020/texmex-github>

URL web repositorio ejercicio 3 (fork): <https://github.com/aleon2020/texmex-github-1/tree/characters>

#### **EJERCICIO 1:**

Crea un nuevo repositorio personal en tu cuenta de usuario GitLab de la EIF. El nombre del repositorio debe ser texmex-lsis. Este repositorio se utilizará para todas las tareas que se realicen en el ejercicio.

Para resolverlo, tendrás que enviar en el espacio de entrega de Aula virtual un documento de texto plano nombrado nombre-apellidos-1.txt, donde [nombre] es tu nombre y [apellidos] tus apellidos. El fichero debe contener una sola línea con una URL clonable del repositorio que has creado en el GitLab EIF.

**NOMBRE DEL FICHERO:** Alberto-LeónLuengo-1.txt

**URL clonable repositorio:** [git@gitlab.etsit.urjc.es:aalberto/texmex-lsis.git](https://gitlab.etsit.urjc.es/aalberto/texmex-lsis.git)

Comienza la práctica clonando en tu máquina o cuenta de laboratorio local el repositorio que acabas de crear. Al final del proceso, el repositorio (local y remoto) debe contener exactamente tres commits.

#### **ABRIR TERMINAL (POR DEFECTO EN EL HOME/CARPETA PERSONAL):**

**\$ cd Desktop/**

**\$ git clone [git@gitlab.etsit.urjc.es:aalberto/texmex-lsis.git](https://gitlab.etsit.urjc.es/aalberto/texmex-lsis.git)**

1. El primer commit debe estar en la rama main y debe agregar exactamente un archivo llamado estudiante.txt. Este archivo de texto plano debe contener una sola línea con tu nombre de usuario de dominio único URJC (quitando el @alumnos.urjc.es).

**\$ cd texmex-lsis/**

**\$ echo "a.leon.2020" > estudiante.txt**

**\$ git add estudiante.txt**

**\$ git commit -m "Archivo estudiante.txt"**

2. El segundo commit debe renombrar/mover el archivo estudiante.txt a username.txt. Además, debe añadir un segundo archivo llamado README.md que puede albergar cualquier contenido.

**\$ mv estudiante.txt username.txt**

**\$ echo "# README ejercicio 1" > README.md**

**\$ git add username.txt README.md**

**\$ git commit -m "Archivos username.txt y README.md"**

3. El tercer commit debe añadir un archivo nuevo de nombre nachos.recipe, que contenga una sola línea con la URL de una página de Internet que explique una receta para hacer nachos con guacamole. Además, este commit también debe añadir un archivo de imagen (en el formato que prefieras), con el resultado del plato preparado. La ruta a este archivo de imagen debe ser img/nachos.jpg (la extensión debe ser la que

corresponda en tu caso). Es decir, esta imagen debe almacenarse en un subdirectorio img dentro del directorio raíz de tu repositorio.

```
$ echo
```

```
"https://www.lavanguardia.com/comer/recetas/20220224/8077696/receta-nachos-guacamole-mas-rapida-sabrosa-celebrar-dia-mundial.html" > nachos.recipe
```

```
$ mkdir img
```

```
$ cd img/
```

```
"Agregar la imagen nachos.jpg descargada de internet al directorio img"
```

```
$ git add nachos.recipe img/nachos.jpg
```

```
$ git commit -m "Archivo nachos.recipe y directorio img/ con imagen nachos.jpg"
```

4. Debes etiquetar (con una etiqueta annotated) el tercer commit como la versión v0.5.

```
$ git tag -a v0.5 -m "v0.5"
```

5. Estos tres commits y la etiqueta deben ser enviados al repositorio remoto en el GitLab EIF.

```
$ git push
```

```
$ git push --tags
```

## EJERCICIO 2:

Continúa trabajando con el repositorio clonado localmente del proyecto, desde el último commit en que lo dejaste al final del Ejercicio 1.

1. Crea una nueva rama llamada new-feature en tu repositorio. Cambia a esa rama para seguir desarrollando en ella. Añade un nuevo archivo de nombre kitchen.txt que contenga lo que quieras. Añádalo con un commit a esa rama.

```
$ git checkout -b new-feature
```

```
$ echo "Welcome to the Gitlab's Kitchen!" > kitchen.txt
```

```
$ git add kitchen.txt
```

```
$ git commit -m "Archivo kitchen.txt"
```

2. De vuelta a la rama master, realiza un merge de la rama new-feature sobre master. Tras esta operación, la rama master debería incorporar el commit realizado en la rama new-feature.

```
$ git checkout master
```

```
$ git merge new-feature
```

3. Todavía en la rama master, crea un nuevo archivo llamado recetas.md con el contenido que prefieras y añádalo con un commit. Crea otra rama nueva de nombre falso-chef y cambia a esa nueva rama. Modifica el archivo recetas.md dentro de la nueva rama y confirma los cambios en un nuevo commit. No mezcles aún los cambios de vuelta a la rama master.

```
$ echo "Nachos with guacamole recipe" > recetas.md
```

```
$ git add recetas.md
```

```
$ git commit -m "Archivo recetas.md"
```

```
$ git checkout -b falso-chef
```

```
"Añadir DELICIOUS!!!" en la segunda línea del fichero recetas.md
```

```
$ git add recetas.md
```

```
$ git commit -m "Archivo recetas.md modificado"
```

4. Cambia de nuevo a la rama master y modifica el archivo recetas.md asegurándote de que realizas cambios incompatibles con los que efectuaste en la versión del archivo en la rama falso-chef. Realiza un commit con esos cambios. Ahora, intenta realizar un merge de la rama falso-chef sobre master. Esto provocará un conflicto entre los cambios en ambas ramas.

**\$ git checkout master**

**“Añadir AMAZING!!!” en la segunda línea del fichero recetas.md**

**\$ git add recetas.md**

**\$ git commit -m “Archivo recetas.md modificado nuevamente”**

**\$ git merge falso-chef**

5. Resuelve el conflicto y concluye la operación de merging de la rama falso-chef sobre la rama master.

**“CONFLICTO! Cambiar en la rama master AMAZING!!! por DELICIOUS!!! para que coincida”**

**\$ git add recetas.md**

**\$ git commit -m “Archivo recetas.md (solución del conflicto)”**

**\$ git merge falso-chef**

6. Por último, etiqueta el último commit (con una etiqueta annotated) como la v1.0. Sube todos los cambios y la etiqueta al repositorio remoto en GitLab EIF.

**\$ git tag -a v1.0 -m “v1.0”**

**\$ git push**

**\$ git push --tags**

7. Para completar el ejercicio, debes configurar un segundo repositorio remoto para tu repositorio local, situado en otro proyecto diferente de tu cuenta GitLab EIF. Este segundo repositorio remoto se debe identificar en tu copia local por el alias upstream. Debes subir todos los cambios realizados y las etiquetas a ambos repositorios remotos para obtener la máxima calificación de este ejercicio.

**“Crea un nuevo repositorio vacío en Gitlab llamado upstream”**

**ABRIR TERMINAL (POR DEFECTO EN EL HOME/CARPETA PERSONAL):**

**\$ cd Desktop/texmex-lsis/**

**\$ git remote add upstream https://gitlab.etsit.urjc.es/aalberto/upstream**

**\$ git remote -v**

**\$ git push origin master**

**\$ git push upstream master**

**\$ git push --tags upstream**

Debes enviar en el espacio de entrega de Aula virtual un documento de texto plano nombrado nombre-apellidos-2.txt, donde [nombre] es tu nombre y [apellidos] tus apellidos. El fichero debe contener dos líneas con las URL clonables de los dos repositorios remotos que has creado en el GitLab EIF en este ejercicio.

**NOMBRE DEL FICHERO: Alberto-LeónLuengo-2.txt**

**URL clonable primer repositorio: [git@gitlab.etsit.urjc.es:aalberto/texmex-lsis.git](https://gitlab.etsit.urjc.es/aalberto/texmex-lsis.git)**

**URL clonable segundo repositorio: [git@gitlab.etsit.urjc.es:aalberto/upstream.git](https://gitlab.etsit.urjc.es/aalberto/upstream.git)**

## **PARTE 2: GITHUB**

### **EJERCICIO 3:**

Para este ejercicio, vas a crear un nuevo repositorio Git en tu cuenta de GitHub. El repositorio se denomina texmex-github.

Atención: En este caso, debes crear primero el repositorio en tu máquina local o en tu cuenta de laboratorio. Después, crea un repositorio vacío en GitHub con el nombre propuesto y configura dicho repositorio como copia remota del proyecto local.

**ABRIR TERMINAL (POR DEFECTO EN EL HOME/CARPETA PERSONAL):**

```
$ cd Desktop/  
$ mkdir texmex-github  
$ cd texmex-github/  
$ git init  
"Crea un nuevo repositorio vacío en Github llamado texmex-github"  
$ git remote add origin https://github.com/aleon2020/texmex-github.git
```

1. El primer commit debe contener un archivo README.md, que explique brevemente el objetivo de este repositorio (elige libremente el texto que añades).

```
$ echo "We are working with Github in this exercise!" > README.md  
$ git add README.md  
$ git commit -m "Archivo README.md"
```

2. Crea un script en Python llamado imprime\_receta.py que imprime por la terminal el contenido del archivo recetas.md del Ejercicio 2. Añade mediante un segundo commit tanto el script de Python como el archivo recetas.md al repositorio.

**"Añadir al repositorio el archivo recetas.md del ejercicio 2"**

```
import sys  
with open('recetas.md', 'r') as archivo:  
    contenido = archivo.read()  
print(contenido)  
$ echo * > imprime_receta.py  
* : Contenido del archivo imprime_receta.py  
$ git add imprime_receta.py recetas.md  
$ git commit -m "Archivos imprime_receta.py y recetas.md"  
$ git push
```

3. Pide a otra persona de clase la URL de su repositorio para este ejercicio en GitHub. A continuación, efectúa un fork de su repositorio en tu propia cuenta de usuario.

**"Buscar en Github el repo de un compañero y crear un fork (fork -> create a new fork)"**

4. Clona este fork que acabas de crear localmente. En la copia local, crea una nueva rama de nombre snoopy. Cambia a esa nueva rama y añade con un commit un archivo llamado characters.md, que contiene exactamente 4 líneas de texto. Cada línea debe contener el nombre de uno de los Personajes de la tira cómica Peanuts.

**ABRIR TERMINAL (POR DEFECTO EN EL HOME/CARPETA PERSONAL):**

```
$ cd Desktop/  
$ git clone https://github.com/aleon2020/texmex-github-1.git  
$ cd texmex-github-1/  
$ git checkout -b characters  
$ echo "Snoopy\nCharlie Brown\nLinus van Pelt\nPeppermint Patty" > characters.md  
$ git add characters.md
```

```
$ git commit -m "Archivo characters.md"
$ git push --set-upstream origin characters
```

**NOTA:** En mi caso, a la rama snoopy la he llamado characters.

5. Por último, prepara un pull request de los cambios en esa nueva rama y realiza una captura de la pantalla final que obtienes justo después de haber enviado la solicitud al repositorio original. Para resolver este ejercicio, debes entregar un archivo de texto plano nombrado nombre-apellidos-1.txt (**renombrado como nombre-apellidos-3.txt para evitar confusiones**), donde [nombre] es tu nombre y [apellidos] tus apellidos. El fichero debe contener dos líneas:

**NOMBRE DEL FICHERO:** Alberto-LeónLuengo-3.txt

- La URL clonable de tu repositorio en GitHub.

**URL clonable repositorio:** <https://github.com/aleon2020/texmex-github.git>

- La URL del fork del repositorio de la otra persona de clase, que has creado bajo tu usuario.

**URL clonable fork:** <https://github.com/aleon2020/texmex-github-1.git>

Acompañando al fichero de texto debes incluir la imagen con la captura de pantalla solicitada en el enunciado de este ejercicio.

