

Karen Ortellado

Actividad de Git y GitHub

Repositorio: <https://github.com/KarenOrtellado/UTN-TUPaD-P1>

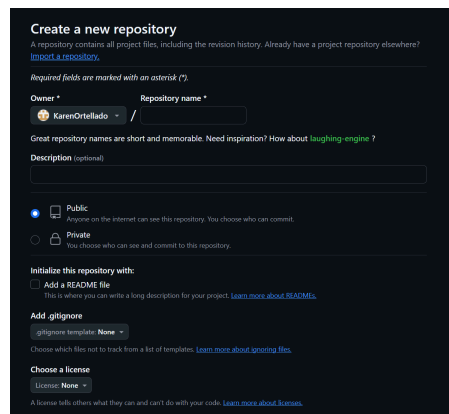
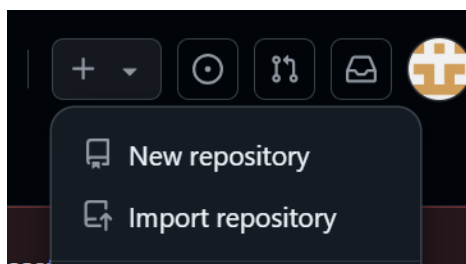
Preguntas punto 1:

a) ¿Qué es GitHub?

Es una plataforma donde podemos guardar y compartir códigos que utiliza y controla GIT. Con Github podemos almacenar, gestionar y colaborar con otros proyectos de software de forma remota

b) ¿Cómo crear un repositorio en GitHub?

Primero iniciamos sesión en Github, y buscamos en la parte superior derecha el signo + donde encontraremos la opción de nuevo repositorio, y luego completamos con un nuevo nombre, si será de dominio público o privado y lo creamos haciendo click en “Crear repositorio”



c) ¿Cómo crear una rama en Git?

Para crear una nueva rama en Git, tenemos que escribir el siguiente código de comando:

git branch nombre_de_rama y hacer click en la tecla **Enter**

d) ¿Cómo cambiar a una rama en Git?

Para cambiar de rama, tenemos que escribir el siguiente código comando:

git checkout nombre_de_rama

e) ¿Cómo fusionar ramas en Git?

Para fusionar ramas, tenemos que escribir el siguiente código comando:

git merge nombre_de_rama y podremos fusionar por ejemplo la rama nombre_de_rama y la rama central main

f) ¿Cómo crear un commit en Git?

Para crear un commit luego de haber editado un archivo, tenemos que escribir el siguiente código comando: **git commit -m “comentario”**, luego haciendo click en enter se guarda el commit.

g) ¿Cómo enviar un commit a GitHub?

Tenemos que escribir: **git push origin nombre_de_rama**

h) ¿Qué es un repositorio remoto?

Un repositorio remoto es en varios casos, una copia de tu repositorio local Git, en un servidor externo, puede ser en éste caso Github

i) ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a git, tenemos que seguir el siguiente código comando: **git remote add origin URL_del_repositorio** (URL_del_repositorio es la dirección dónde se encuentra el repositorio)

j) ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios locales al repositorio remoto, debemos seguir el siguiente comando:

git push origin nombre_de_rama

k) ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar cambios de un repositorio remoto tenemos que seguir el siguiente comando:

git pull origin nombre_de_rama

l) ¿Qué es un fork de repositorio?

Un **Fork** es una copia de un repositorio que nos va a permitir realizar cambios sin afectar el repositorio original.

m) ¿Cómo crear un fork de un repositorio?

Para crear un fork, tenemos que ingresar a Github y buscar el repositorio que necesitamos, hacemos click en FORK y eso nos va a permitir crear la copia en nuestra cuenta de github, podemos modificar el nombre y agregarle alguna descripción.

n) ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

GitHub te sugerirá abrir un **Pull Request**. Haz clic en "**New Pull Request**" y elegimos la rama y los cambios que deseas enviar.

o) ¿Cómo aceptar una solicitud de extracción?

Podemos aceptar una solicitud de extracción haciendo click en "**Merge Pull Request**" y "**Confirm Request**"

p) ¿Qué es una etiqueta en Git?

Es una marca (Tag) con la que podemos señalar puntos específicos en el historial de un proyecto.

q) ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en GIT tenemos que realizar el comando: **git tag -a nombre_etiqueta -m "Mensaje de la etiqueta"**

r) ¿Cómo enviar una etiqueta a GitHub?

Tenemos que seguir el siguiente comando: **git push origin nombre_de_etiqueta**

s) ¿Qué es un historial de Git?

Un historial en Git es un registro de todos los cambios que se hicieron en un repositorio, como los commits, y aparece el registro de quién y cuándo se realizó.

t) ¿Cómo ver el historial de Git?

Para ver el historial tenemos que escribir **git log** y hacer click en enter.

u) ¿Cómo buscar en el historial de Git?

Para buscar algo en específico, tenemos que escribir: **git log --grep="texto_a_buscar"**

v) ¿Cómo borrar el historial de Git?

Para borrar el historial: **git reset --hard HEAD~n** (donde n es el número de commits que vamos a retroceder).

w) ¿Qué es un repositorio privado en GitHub?

Un **repositorio privado** es aquel que solo puede ser accedido por los usuarios que han sido invitados a él. No es visible para el público en general.

x) ¿Cómo crear un repositorio privado en GitHub?

creamos un nuevo repositorio, pero seleccionamos la opción de "Privado" antes de hacer click en Crear

y) ¿Cómo invitar a alguien a un repositorio privado en GitHub?

en la configuración de ese repositorio privado hacemos click en "Manejar accesos" y agregamos a la persona a través de su usuario de github o su cuenta de correo.

z) ¿Qué es un repositorio público en GitHub?

Un **repositorio público** es aquel que cualquiera puede ver y clonar, es accesible para todo el mundo.

aa) ¿Cómo crear un repositorio público en GitHub?

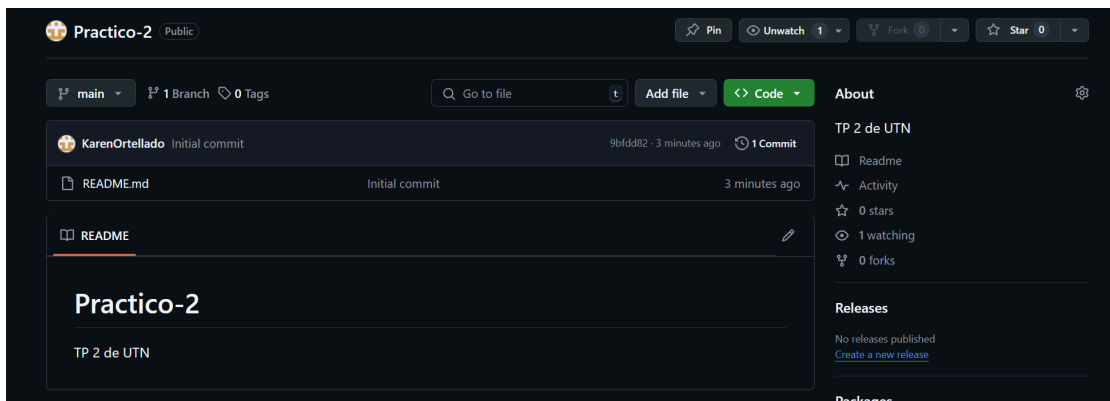
Hacemos click en crear repositorio, y luego de nombrarlo y agregar alguna descripción, hacemos click en la opción "Público", luego click en Crear

bb) ¿Cómo compartir un repositorio público en GitHub?

Solo es necesario compartir la URL del repositorio, ya que no es necesario otorgar permisos.

Actividades siguientes:

Realizar la siguiente actividad: Paso 1: Crear un repositorio en GitHub • Ve a GitHub e inicia sesión en tu cuenta. • Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio. • Asigna un nombre al repositorio, por ejemplo, conflict-exercise. • Opcionalmente, añade una descripción. • Marca la opción "Initialize this repository with a README". • Haz clic en "Create repository".



Paso 2: Clonar el repositorio a tu máquina local • Copia la URL del repositorio (usualmente algo como <https://github.com/tuusuario/conflict-exercise.git>). • Abre la terminal o línea de comandos en tu máquina. • Clona el repositorio usando el comando: `git clone https://github.com/tuusuario/conflict-exercise.git`

```
Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT (master)
$ git clone https://github.com/KarenOrtellado/Practico-2
Cloning into 'Practico-2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

• Entra en el directorio del repositorio: `cd conflict-exercise` Paso 3: Crear una nueva rama y editar un archivo • Crea una nueva rama llamada feature-branch: `git checkout -b feature-branch`

```
Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT (master)
$ cd Practico-2
Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (main)
$ git branch feature-branch
Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (main)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'
```

Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo: Este es un cambio en la feature branch. • Guarda los cambios y haz un commit: `git add README.md` `git commit -m "Added a line in feature-branch"`

```
Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (feature-brach)
$ git add README.md

Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (feature-brach)
$ git commit "Added a line in feature-branch"
error: pathspec 'Added a line in feature-branch' did not match any file(s) known to git

Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (feature-brach)
$ git status
On branch feature-brach
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

Activity@ACTVT_LAPTOP MINGW64 /c/users/Activity/OneDrive/Escritorio/UTN Primer Cuatrimestre/Programación I/Repositorio GIT/Practico-2 (feature-brach)
$ git commit -m "Added a line in feature-branch"
[feature-brach 7f70f4b] Added a line in feature-branch
1 file changed, 2 insertions(+)
```