PROGRAMACIÓN I
Práctico 2: Git y GitHub

1)

• ¿Qué es GitHub?

GitHub es una plataforma para alojar repositorios en la nube, que facilita la colaboración de distintas personas en un mismo proyecto y provee acceso remoto. También es útil para gestionar proyectos ya que ofrece herramientas como issues, para informar errores o mejoras, y pull requests, para solicitar revisión de código.

• ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub primero debemos iniciar sesión en la web o bien registrarnos, si es que no tenemos una cuenta. Luego hacemos click en el botón "+" de la esquina superior derecha de la página, y seleccionamos "Nuevo repositorio". Posteriormente, debemos completar obligatoriamente el nombre del repositorio, e indicar si será público o privado. El resto de los campos, al ser opcionales, pueden no completarse si así lo deseamos.

• ¿Cómo crear una rama en Git?

Para crear una rama en Git, debemos usar el comando git branch seguido del nombre que le pondremos a la rama. Por ejemplo: `git branch production`.

• ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama en Git, debemos usar el comando git checkout seguido del nombre de la rama a la que queremos cambiar. Por ejemplo: `git checkout develop`.

• ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, debemos posicionarnos sobre la rama donde queremos que se guarden los cambios, y ejecutar el comando git merge seguido de la rama que queremos fusionar. Por ejemplo: si queremos fusionar los cambios de la rama belen-dev con la rama dev, nos pasamos sobre dev y ejecutamos `git merge belen-dev`.

• ¿Cómo crear un commit en Git?

Para crear un commit en Git, escribimos el comando git commit -m seguido de un mensaje explicativo, entre comillas, de los cambios que hemos realizado. Por ejemplo: `git commit -m "Funcion suma de productos agregada al carrito de compras"`.

• ¿Cómo enviar un commit a GitHub?

Para enviar un commit a GitHub, debemos ejecutar el comando git push origin seguido del nombre de la rama que estamos enviando. Usamos "origin" para referirnos al repositorio remoto principal. Por ejemplo: `git push origin main`.

• ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de nuestro proyecto que está alojada en un servidor en línea, que puede ser GitHub u otro, como GitLab o Bitbucket. A diferencia de un repositorio local, que se encuentra en nuestra computadora, un repositorio remoto está accesible a través de Internet. Esto permite principalmente que múltiples desarrolladores trabajen en el mismo proyecto simultáneamente, facilitando la colaboración.

• ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git debemos ejecutar el comando git remote add seguido del nombre remoto y la url del repositorio remoto. Por ejemplo: `git remote add origin https://github.com/belenyb/mi-repositorio.git`.

• ¿Cómo empujar cambios a un repositorio remoto?
Para empujar cambios hacia un repositorio remoto debemos ejecutar el comando git push seguido del nombre remoto y el nombre de la rama. Por ejemplo: `git push origin main`.

• ¿Cómo tirar de cambios de un repositorio remoto?
Para traer cambios de un repositorio remoto debemos ejecutar el comando git pull seguido del nombre remoto y el nombre de la rama que contiene los cambios que queremos incorporar. Por ejemplo: `git pull origin main`.

• ¿Qué es un fork de repositorio?
Un fork de repositorio es una copia de un repositorio remoto en GitHub. Permite crear una versión personal de un repositorio existente y realizar cambios sin afectar el repositorio original.

• ¿Cómo crear un fork de un repositorio?
Para crear un fork de un repositorio, debemos dirigirnos a la página del repositorio en GitHub y hacer click en el botón “Fork” de la esquina superior derecha.

• ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?
Para enviar un pull request a un repositorio, primero debemos realizar un git push a la rama remota que deseamos modificar. Luego, dentro de GitHub, nos dirigimos a la rama indicada, dentro del repositorio correspondiente. Allí veremos la opción para crear una nueva Pull Request, donde tendremos que seleccionar la rama que contiene nuestros cambios y la rama a la que deseamos fusionarlos.

• ¿Cómo aceptar una solicitud de extracción?
Para aceptar una solicitud de extracción debemos apretar el botón de “Merge” dentro del pull request. Si este merge automático no es posible, primero deben resolverse los conflictos para luego aceptar el pull request.

• ¿Qué es una etiqueta en Git?
Una etiqueta en Git es una forma de marcar un punto específico en el historial del repositorio. Se utilizan comúnmente para marcar versiones de software. Por ejemplo, "v1.0".

• ¿Cómo crear una etiqueta en Git?
En Git, las etiquetas se crean con el comando git tag, seguido del nombre de la etiqueta, o también especificando una descripción de esa etiqueta. Para el primer caso utilizaremos `git tag nombre-etiqueta` y, para el segundo, `git tag -a nombre-etiqueta -m "Descripcion de la etiqueta"`. Por ejemplo: `git tag -a v1.0 -m "Versión 1.0 del proyecto"`.

• ¿Cómo enviar una etiqueta a GitHub?
Para enviar una etiqueta a GitHub, usamos el comando git push origin seguido del nombre de la etiqueta. Por ejemplo: `git push origin v1.0`. Si deseamos enviar todas las etiquetas a GitHub, podemos usar el comando `git push origin --tags`.

• ¿Qué es un historial de Git?

Un historial de Git es un registro de todos los cambios realizados en un repositorio a lo largo del tiempo, que permite rastrear y gestionar las modificaciones del código fuente.

• ¿Cómo ver el historial de Git?

Para ver el historial de Git, ejecutamos el comando `git log`. Este comando muestra el historial de commits completo, incluyendo el autor, la fecha, el mensaje del commit y el hash del commit.

• ¿Cómo buscar en el historial de Git?

Hay diversas formas de buscar dentro del historial de Git:

- Por commit, usando `git log --grep="texto de commit"`
- Por archivo, ejecutando `git log index.html` para ver los cambios que han sido aplicados a un archivo en particular.
- Por contenido, por ejemplo buscando cambios en un fragmento de código específico, como una función importante. Usamos `git log -S "getUserData"`.
- Por autor, ejecutando `git log --author=<belenyb>`.
- Por fecha, usando el comando `git log --since="2025-01-01" --until="2025-06-30"` con fechas desde y hasta.
- Por cambios específicos en el contenido, usando `git log -p` para obtener el historial de commits junto con las diferencias (diffs) de los cambios realizados en cada commit.

También se pueden combinar métodos para que la búsqueda sea más efectiva.
Por ejemplo, buscamos todos los commits del autor belenyb realizados desde el corriente mes: `git log --author="belenyb" --since="2025-03-01"`.

• ¿Cómo borrar el historial de Git?

Para borrar el historial local de Git, es decir, cambios que no hemos subido aún al repositorio remoto, realizamos el comando `git reset --hard` seguido del id del commit al que queremos regresar: `git reset --hard <commit_id>`. Si lo que deseamos es quitar algún archivo del historial, por ejemplo sin subimos sin querer un archivo .env, debemos ejecutar `git filter-branch`. Si queremos limpiar el historial de commits, utilizamos el comando `git rebase -i`, por ejemplo usando `git rebase -i HEAD~3` si queremos borrar los últimos 3 commits de la rama actual.

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio de código al que solo pueden acceder las personas que tienen permisos explícitos. Son ideales para almacenar código sensible o información confidencial (por ejemplo repositorios de empresas) y permiten que la colaboración más controlada y segura en proyectos privados.

• ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub, debemos hacer clic en el botón "+" (Nuevo) localizado en la esquina superior derecha de la página, y seleccionar "Nuevo repositorio". Dentro de los campos a completar, debemos seleccionar la opción "Privado".

• ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Para invitar a alguien a un repositorio privado en GitHub, debemos dirigirnos a la página del repositorio y, dentro de la pestaña “Configuración”, seleccionar la opción "Colaboradores". Allí escribiremos el nombre de usuario de GitHub o la dirección de correo electrónico de la persona que deseamos invitar. El usuario recibirá una invitación por correo electrónico. Una vez que acepte, tendrá acceso al repositorio según los permisos que le designamos.

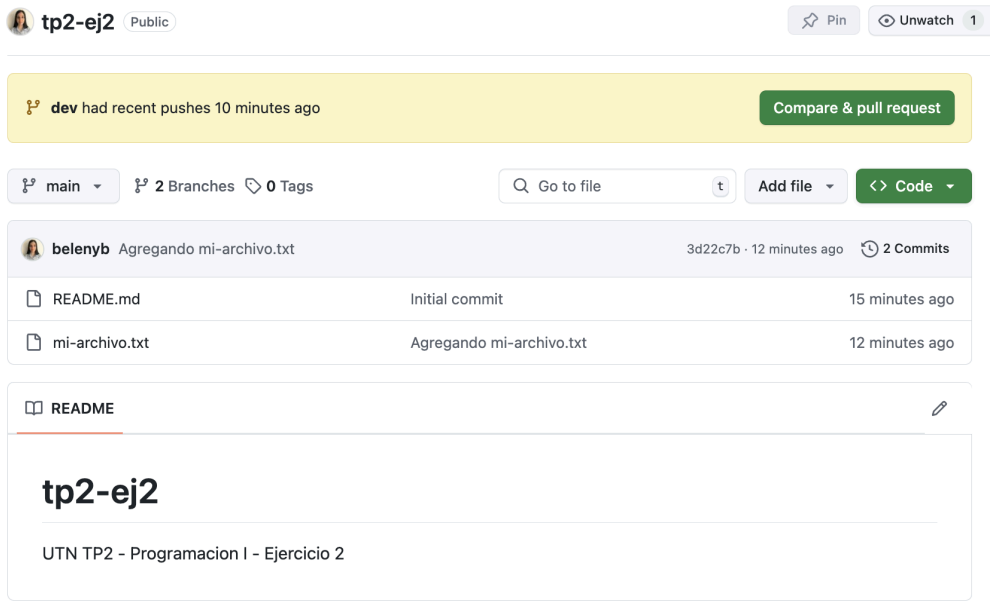
• ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio de código que es visible para cualquier persona en Internet. Esto significa que cualquier persona puede ver, clonar y, en algunos casos, contribuir al código del repositorio.

• ¿Cómo crear un repositorio público en GitHub?
Para crear un repositorio público en GitHub, seguimos los pasos convencionales ya mencionados pero, antes de hacer clic en “Crear repositorio”, marcamos la opción “Público”.

• ¿Cómo compartir un repositorio público en GitHub?
La forma más sencilla de compartir un repositorio público en GitHub es a través de la URL ya que, de esta manera, cualquier persona podrá acceder al repositorio y ver el código que lo compone.

2) https://github.com/belenyb/tp2-ej2



3) https://github.com/belenyb/conflict-exercise