1) Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas) :

• ¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo que funciona como una red social para programadores. Permite almacenar proyectos de código, trabajar en equipo, gestionar versiones de un proyecto y colaborar en el mismo repositorio junto a otros desarrolladores.

• ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub, primero debes iniciar sesión en tu cuenta. Luego, haces clic en el botón 'New Repository', asignas un nombre al repositorio, elegís si será público o privado, podés agregar una descripción y, si querés, inicializarlo con un README. Finalmente, hacés clic en 'Create repository' para crearlo.

• ¿Cómo crear una rama en Git?

Para crear una rama en Git, primero se debe estar ubicado en el repositorio local a través de la terminal. Luego, se utiliza el comando **git branch nombre-de-la-rama** para crear una nueva rama.
En caso de querer crear y cambiarse automáticamente a esa nueva rama, se puede utilizar el comando **git checkout -b nombre-de-la-rama.**
Trabajar en ramas permite desarrollar nuevas funcionalidades sin afectar el código principal.

• ¿Cómo cambiar a una rama en Git?

Para cambiar a otra rama en Git, primero se puede verificar qué ramas existen en el repositorio utilizando el comando **git branch**. Luego, para moverse a la rama deseada, se usa el comando **git checkout nombre-de-la-rama.**
Este procedimiento permite trabajar en diferentes versiones del proyecto de manera organizada.

• ¿Cómo fusionar ramas en Git?

Para fusionar ramas en Git, primero se debe posicionar en la rama donde se desea **traer** los cambios, utilizando el comando **git checkout nombre-de-la-rama-destino** (por ejemplo, **main**).
Luego, se ejecuta el comando **git merge nombre-de-la-rama-origen**, lo que permite unir los cambios realizados en la rama de origen hacia la rama destino.
En caso de existir conflictos durante la fusión, Git solicitará resolverlos antes de completar el proceso.

• ¿Cómo crear un commit en Git?

Para crear un commit en Git, primero es necesario guardar los cambios en el área de preparación utilizando el comando **git add nombre-del-archivo**, o **git add .** si se quieren agregar todos los archivos modificados.
Una vez que los cambios están preparados, se crea el commit con el comando **git commit -m "mensaje descriptivo**", donde el mensaje debe explicar brevemente qué cambios se realizaron.

• ¿Cómo enviar un commit a GitHub?

Una vez creado el commit de forma local, para enviarlo a GitHub es necesario usar el comando **git push**.
Primero se debe asegurar que se está trabajando en la rama correcta (por ejemplo, **main** o **master**) y luego ejecutar **git push origin nombre-de-la-rama.**
Este comando envía los cambios locales al repositorio remoto en GitHub, actualizando el proyecto y permitiendo que otros colaboradores también vean las últimas actualizaciones.

• ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión del proyecto que está alojada en un servidor externo, como GitHub.
A diferencia del repositorio local, que se encuentra en la computadora personal del desarrollador, el repositorio remoto permite almacenar el código en la nube y facilita el trabajo colaborativo entre varias personas.
Gracias a los repositorios remotos, se pueden compartir los cambios, sincronizar el proyecto entre distintos equipos y mantener una copia de respaldo segura del trabajo realizado.

• ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a un proyecto en Git, se utiliza el comando **git remote add origin URL-del-repositorio**.
Primero, es necesario tener creado el repositorio en una plataforma como GitHub. Luego, desde la terminal y dentro del proyecto local, se ejecuta ese comando, reemplazando URL-del-repositorio por la dirección que proporciona la plataforma.
Esto establece un vínculo entre el proyecto local y el repositorio remoto, permitiendo enviar y sincronizar los cambios entre ambos.

• ¿Cómo empujar cambios a un repositorio remoto?

Para empujar (enviar) cambios a un repositorio remoto, primero se debe guardar todo localmente mediante un commit **(git commit**).
Luego, se utiliza el comando **git push origin nombre-de-la-rama** para enviar esos

cambios al repositorio remoto, como GitHub.
Este proceso actualiza el repositorio en línea con las últimas modificaciones realizadas en el repositorio local.

• ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar (traer) cambios de un repositorio remoto y actualizarlos en el repositorio local, se utiliza el comando **git pull origin nombre-de-la-rama.**
Este comando descarga todos los cambios nuevos que hayan sido subidos al repositorio remoto por otros colaboradores y los fusiona automáticamente con el trabajo local.
Es una buena práctica hacer un **git pull** antes de comenzar a trabajar para asegurarse de tener la última versión del proyecto.

• ¿Qué es un fork de repositorio?

Un fork de un repositorio es una copia completa de un proyecto que se crea dentro de tu propia cuenta de GitHub.
Hacer un fork permite trabajar de manera independiente sobre un proyecto sin afectar el original. Es muy utilizado cuando se quiere proponer cambios en proyectos de otros o simplemente practicar y desarrollar sobre una base ya existente.

• ¿Cómo crear un fork de un repositorio?

Para crear un fork, primero hay que ingresar al repositorio original en GitHub.
Luego, se hace clic en el botón "Fork", ubicado en la parte superior derecha de la página del repositorio.
Automáticamente, GitHub creará una copia del proyecto en tu propia cuenta, y desde ahí podrás trabajar libremente en tus modificaciones, sin alterar el proyecto original.

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

Para enviar una solicitud de extracción (o pull request), primero hay que realizar los cambios necesarios en una rama dentro del fork del proyecto.
Luego, desde GitHub, se debe ir al repositorio original y hacer clic en el botón **"New Pull Request".**
Se elige la rama donde hiciste los cambios y se compara con la rama principal del repositorio original.
Finalmente, se completa un pequeño formulario donde se explica qué cambios se proponen y se envía la solicitud para que los administradores del proyecto la revisen.

• ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción, se debe ser administrador o tener permisos de colaboración en el repositorio original.
Primero, se entra a la sección de "**Pull requests**", se elige la solicitud pendiente, se revisan los cambios propuestos y si todo está correcto, se hace clic en el botón "**Merge pull request".**
Luego se confirma la acción. Esto fusiona los cambios sugeridos por otra persona en la rama principal del proyecto.

• ¿Qué es una etiqueta en Git?

Una etiqueta en Git (tag) es un marcador especial que se utiliza para señalar puntos importantes en la historia de un proyecto, como una versión estable o un lanzamiento. Las etiquetas permiten identificar fácilmente versiones específicas del proyecto, por ejemplo, "v1.0" o "v2.1", y son muy útiles para el control de versiones y la distribución de software.

• ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git, primero hay que asegurarse de estar en el commit correcto.
Luego se usa el comando **git tag nombre-de-la-etiqueta**.
Si se quiere agregar una etiqueta con un mensaje, se puede usar **git tag -a nombre-de-la-etiqueta -m "mensaje descriptivo"**, donde el parámetro **-a** indica que es una etiqueta anotada y **-m** permite escribir un mensaje explicando el propósito de la etiqueta.

• ¿Cómo enviar una etiqueta a GitHub?

Una vez creada la etiqueta en el repositorio local, se debe enviar al repositorio remoto utilizando el comando **git push origin nombre-de-la-etiqueta.**
También se puede enviar todas las etiquetas de una sola vez con **git push origin --tags.**
Esto asegura que las etiquetas locales también queden almacenadas en GitHub, junto con el resto del historial del proyecto.

• ¿Qué es un historial de Git?

El historial de Git es un registro de todas las modificaciones realizadas en un repositorio. Incluye información sobre cada "commit", como el autor, la fecha, el mensaje del commit y los cambios realizados. Permite rastrear el progreso del proyecto, revisar cambios específicos y revertir modificaciones si es necesario.

• ¿Cómo ver el historial de Git?

Puedes ver el historial de commits en Git usando el siguiente comando:
git log

• ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes usar el siguiente comando:

git log --grep="palabra_clave"

• ¿Cómo borrar el historial de Git?

Eliminar todos los commits pero mantener los archivos:

```
git checkout --orphan latest_branch
git add -A
git commit -m "Primer commit"
git push --force origin latest_branch
```

Esto crea un nuevo historial sin tocar los archivos del proyecto.

Borrar un commit específico: Para eliminar un commit específico y actualizar el historial, puedes usar **git rebase:**

git rebase -i <commit_anterior_al_que_quieres_eliminar>

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio al que solo los usuarios a los que se les ha dado acceso pueden ver y colaborar. A diferencia de los repositorios públicos, un repositorio privado no es accesible para el público en general.

• ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub:

1. Inicia sesión en tu cuenta de GitHub.
2. En la página principal, haz clic en New (Nuevo).
3. Llena los detalles del repositorio: nombre, descripción, etc.
4. Bajo Visibility, selecciona la opción Private (Privado).
5. Haz clic en Create repository.

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

Para invitar a alguien a un repositorio privado:

1. Accede al repositorio en GitHub.
2. Ve a **Settings** (Configuraciones).
3. En la barra lateral, selecciona **Manage access** (Gestionar acceso).
4. Haz clic en **Invite a collaborator** (Invitar a un colaborador).
5. Ingresa el nombre de usuario de la persona que quieres invitar y selecciona el rol adecuado (por ejemplo, colaborador).
6. Haz clic en **Add collaborator**.

• ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub es un repositorio accesible por cualquier persona en internet. Cualquier usuario puede ver, descargar, y contribuir al proyecto si tienen los permisos adecuados.

• ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público:

1. Inicia sesión en tu cuenta de GitHub.
2. Haz clic en New (Nuevo).
3. Llena los detalles del repositorio: nombre, descripción, etc.
4. Bajo Visibility, selecciona la opción Public (Público).
5. Haz clic en Create repository.

• ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub:

1. Abre el repositorio en GitHub.

2. Copia la URL del repositorio en la barra de direcciones de tu navegador. La URL será algo como **https://github.com/usuario/repositorio.**

3. Comparte el enlace directamente con otras personas

Repositorio GitHub: https://github.com/yanimica/Tp2-Git-GitHub.git