Link git hub: https://github.com/juliangonend/lab1\_JulianGon\_B.git

1- Contestar las siguientes preguntas utilizando las guías y documentación proporcionada

(Desarrollar las respuestas) :

• ¿Qué es GitHub?

Es un servicio de alojamiento de repositorios de control de versiones en el que se puede almacenar y gestionar sus proyectos de software

• ¿Cuáles son los beneficios de usar GitHub?

Las ventajas para usar GitHub para control de versiones es :

* Facilidad de uso : fácil de usar incluso para los desarrolladores que no estén familiarizados con el control de versiones
* Escalabilidad: satisface las necesidades de proyecto de cualquier tamaño
* Seguridad: cuenta con características de seguridad para proteger a los usuarios
* Comunidad : cuenta con una gran comunidad de desarrolladores que están dispuestos a ayudar y colaborar

• ¿Cómo crear un repositorio en GitHub?

1. Debes crearte una cuenta en GitHub
2. Crear un repositorio desde la página:
   1. Ir al ícono de la esquina superior derecha y seleccionar “Your repositories”
   2. En la página de repositorios has click en el repositorio de New
3. Agregar en nombre al repositorio y una breve descripción y elegir si lo queres con el archivo README , .gitignore , etc
4. Deslizarte hasta abajo y darle a crear repositorio

• ¿Cuáles son los tipos de ramas en Git?

Se utilizan básicamente 5 ramas : Master, Desarrollo, QA, Features, y Hotfix.

1. Master: es la rama principal. Contiene el repositorio de lo que se encuentra en producción.

2. Desarrollo: Es una rama sacada del master. Es una rama de integración. Luego que se realice la integración y se corrijan los errores (en caso de que haya alguno), luego de que se encuentre estable, se puede hacer un merge de desarrollo sobre la rama QA.

3. Features: cada nueva funcionalidad se debe realizar en una rama nueva específica para esa funcionalidad. Sale de la rama Desarrollo. Una vez que la funcionalidad esté desarrollada, se hace un merge sobre la rama de Desarrollo, donde se integrará con el resto de las funcionalidades.

4. QA: es una rama sacada del desarrollo que fue probado y que pasa a QA para que el usuario final realice la prueba de aceptación.

5. Hotfix: son bugs que surgen en producción, por lo que se deben arreglar e implementar en producción de forma urgente. Por eso son ramas sacadas de Master. Una vez corregido el error, se debe hacer un merge (une las ramas), sobre el Master. Luego, para que no quede desactualizada la versión, se debe hacer un merge sobre QA y luego sobre Desarrollo.

• ¿Cómo crear una rama en Git?

Para crear una rama en git lo haremos con el comando git checkout seguido del nombre de la rama que queremos crear

• ¿Cómo cambiar a una rama en Git?

Para pasar de una rama lo moveremos con el comando git checkout seguido del nombre de la rama que queremos cambiarnos

• ¿Cómo fusionar ramas en Git?

Para fusionar ramas en git lo haremos con el git merge por ejemplo si queremos fusionar la rama master con la “ramaGit” lo haremos con git merge ramaGit

• ¿Cómo crear un commit en Git?

Para crear un commit en Git lo haremos con el comando git commit -m”acá insertamos el comentario”

• ¿Cómo enviar un commit a GitHub?

Lo enviaremos con un git push <nombre-remoto> <nombre de la rama>

• ¿Qué es un repositorio remoto?

Son versiones de proyecto que están hospedadas en Internet o en cualquier otra red. Puedes tener varios de ellos, y cada uno tendrás generalmente permisos de solo lectura o de lectura y escritura

• ¿Cómo agregar un repositorio remoto a Git?}

Se agrega con el comando git remote add [nombre]

• ¿Cómo empujar cambios a un repositorio remoto?

Los cambios a un repositorio remoto es con git push[nombre-remoto][nombre-rama]

• ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar los cambios de un repositorio se hace con git remote rm <nombre\_del\_repositorio\_remoto>

• ¿Qué es un fork de repositorio?

Un fork en un repositorio es hacer una copia de un repositorio externo de nuestro agrado y hacerle modificaciones , luego los cambios de este fork se guardaran en una copia local

• ¿Cómo crear un fork de un repositorio?

En github has click en la esquina superior derecha en el botón de fork del repositorio que quieres bajar , luego de eso se guardara el repositorio en la cuenta de GitHub

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

Nos dirigiremos a la zona de Pull Request y le daremos click a Create pull Request donde colocaremos algún asusto y más abajo mencionaremos los cambios que hemos realizado y por que ese cambio

• ¿Cómo aceptar una solicitud de extracción?

Para solicitar una solicitud de extracción lo hacemos desde la pagina de GitHub en la parte de pull request hacemos click en merge Pull request y de esta manera sumara los cambios de la solicitud de extracción al repositorio

• ¿Qué es un etiqueta en Git?

En Git, una etiqueta o tag es una forma de marcar un punto específico en la historia del repositorio. Las etiquetas se utilizan típicamente para marcar versiones o hitos importantes en el desarrollo del proyecto.

Es etiquetar puntos específicos del historial como importantes

Existen dos formas

Ligeras: es un puntero a un commit específico

Anotadas: se guardan en la base de datos de Git como objetos enteros

• ¿Cómo crear una etiqueta en Git?

Se puede crear de dos formas:

Las anotadas se crean especificando el -a cuando se utiliza el git tag -a -m”acá se colocara un mensaje específico , si no se agrega git habrira tu editor para lograrlo”

Ligeras: se crean simplemente con el comando git tag son agregarle -a -s ni -m como en el anterior

Anotadas: se guardan en la base de datos de Git como objetos enteros

• ¿Cómo enviar una etiqueta a GitHub?

Para enviar la etiqueta a GitHub es con el comando git push origin nombre-del-la-etiqueta

• ¿Qué es un historial de Git?

Se refiere al registro de cambios y versiones de un repositorio de código a lo largo del tiempo.

• ¿Cómo ver el historial de Git?

Para ver el historial de git con el comando de git log

• ¿Cómo buscar en el historial de Git?

Para bucar en el historial de Git lo hacemos con el comando de git log :

Con git log –-grep=”para buscar la palabra clave”

Git log -S “función específica”

Git log archivo-específico.txt

• ¿Cómo borrar el historial de Git?

Se borra con el comando rm -rf .git

Tecnicatura Superior en Programación

UTN-FRM

Laboratorio de Computación I – Git & GitHub

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es : un repositorio que tiene un acceso restringido donde las personas que son invitadas como colaboradores mas el creador pueden verlo y modificarlo. Donde los colaboradores pueden entrar en diferentes niveles de acceso (lectura, escritura y administración)

• ¿Cómo crear un repositorio privado en GitHub?

Creamos un repositorio , luego debajo de la descripción nos da la opción de elegir entre público o privado y elegimos privado.

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

En la pagina del repositorio vamos a settings , luego vamos a Manage Access y en esa pagina encontraremos una sección que se llame “Invite a collaborator” y ahí ingresaremos el nombre de usuario de la persona que queremos utilizar

• ¿Qué es un repositorio público en GitHub?

Un repositorio público en git es un repositorio que esta visible para todo el internet

• ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público en GitHub cuando creamos un nuevo repositorio debajo de la descripción tenemos dos opciones (público y privado) , le damos click en publico y luego en crear repositorio

• ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub lo hacemos copiando el link https de la pagina del repositorio y luego enviamos el link a donde lo queramos compartir

• ¿Qué es un repositorio de código en GitHub?

Es un espacio de almacenamiento en línea que se utiliza para gestionar y colaborar en proyectos de desarrollo de software

• ¿Cómo crear un repositorio de código en GitHub?

Para crear un repositorio de código en GitHub, se debe iniciar sesión en la cuenta, hacer click en el botón "+" en la esquina superior derecha del perfil y seleccionar "New repository". Luego, establecer un nombre para el repositorio, agregar una descripción opcional, marcar la opción "Public" o "Private", y hacer click en "Create repository".• ¿Cómo compartir un repositorio de código en GitHub?

Atraves del enlace del repositorio se lo mandamos a otras personas. También se pueden agregar colaboradores al repositorio para que contribuyan al proyecto. Para hacerlo, hay que ir a la pestaña "Settings" del repositorio, seleccionar "Collaborators" en el menú de la izquierda y agregar los nombres de usuario de los colaboradores.

• ¿Qué es un repositorio de documentación en GitHub?

Es un repositorio creado específicamente para almacenar y gestionar la documentación de un proyecto o sistema de software

• ¿Cómo crear un repositorio de documentación en GitHub?

Hay que iniciar sesión en la cuenta, hacer click en el botón "+" en la esquina superior derecha del perfil y seleccionar "New repository". Luego, establecer un nombre para el repositorio de documentación, agregar una descripción opcional y marcar la opción "Public" o "Private". Además, se puede seleccionar la opción "Initialize this repository with a README" para crear un archivo README.md inicial que se puede editar con la documentación. Finalmente, hacer click en "Create repository".

• ¿Cómo compartir un repositorio de documentación en GitHub?

Solo hay que compartir el enlace del repositorio con otras personas. También se pueden agregar colaboradores al repositorio para que puedan contribuir y editar la documentación. También se pueden utilizar las funciones de control de versiones y solicitudes de extracción para llevar un seguimiento de los cambios y revisar las actualizaciones realizadas por los colaboradores.

• ¿Qué es un repositorio de proyecto en GitHub?

Es donde se almacena y gestiona el código fuente, la documentación y cualquier otro recurso relacionado con un proyecto. Permite a los desarrolladores trabajar en equipo, llevar un historial de versiones, realizar seguimiento de problemas y solicitudes de extracción, y colaborar en la construcción del proyecto.

• ¿Cómo usar Git para trabajar con otros

Para usar git con otros se debe crear un entorno de trabajo que luego se conectara a un repositorio remoto y luego las otras personas clonaran el repositorio con git clone y, con los comandos git pull recibimos los cambios y con git push enviamos los cambios.

**2- Realizar la siguiente actividad:**

• Crear un repositorio.

o Dale un nombre al repositorio.

o Elije el repositorio sea público.

o Inicializa el repositorio con un archivo.

• Agregando un Archivo

o Crea un archivo simple, por ejemplo, "mi-archivo.txt".

o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.

o Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

• Creando Branch

o Crear una Branch

o Realizar cambios o agregar un archivo

o Subir la Branch

**Entrega**:

* ~~Responder las preguntas en el documento de Word.~~
* ~~Colocar Apellido y Nombre.~~
* Copiar el enlace al proyecto de GitHub.

**Importante**: La entrega de este trabajo se realizará de forma individual.

|