**Trabajo práctico Git-Github**

|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Programación I** | |
|  | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: 4 |
| **Carrera**: *Tecnicatura Universitaria en Programación* | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**

|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
| **Cinthia Rigoni** |  | 6 horas |

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

• **¿Qué es GitHub?**

**Respuesta:**

GitHub es una plataforma de desarrollo colaborativo basada en Git, un sistema de control de versiones distribuido. Utilizados para el alojamiento de proyectos de software y el control de versiones.

**• ¿Cuáles son los beneficios de usar GitHub?**

**Respuesta:**

1. Control de Versiones eficiente con Git.

2. Facilita la colaboración en equipo.

3. Ofrece visibilidad a los proyectos (especialmente útil para proyectos de código abierto).

4. Proporciona funciones de seguridad como autenticación de dos factores y análisis de dependencias.

5. Permite la automatización de tareas a través de GitHub Actions.

6. Gestión efectiva de problemas y tareas.

7. Integración Continua para pruebas automáticas y despliegues regulares.

8. Facilita la creación y mantenimiento de documentación y wikis.

9. Admite una amplia variedad de lenguajes de programación y tecnologías.

10. Fomenta la transparencia y la colaboración abierta en el desarrollo de software.

• **¿Cómo crear un repositorio en GitHub?**

**Respuesta:**

Para crear un repositorio en GitHub, primero, inicia sesión en tu cuenta de GitHub. Luego, haz clic en el signo más en la esquina superior derecha de la página y selecciona "Nuevo repositorio". Ingresa un nombre para tu repositorio, una descripción opcional y elige si quieres que sea público o privado. Después, haz clic en el botón "Crear repositorio".

**• ¿Cuáles son los tipos de ramas en Git?**

**Respuesta:**

1. Rama Principal (Master/Branch)

2. Ramas de Funcionalidades (Feature Branches)

3. Ramas de Publicación (Release Branches)

4. Ramas de Corrección de Errores (Hotfix Branches)

5. Ramas de Soporte (Support Branches)

6. Ramas de Tareas (Task Branches)

7. Ramas Remotas (Remote Branches)

**• ¿Cómo crear una rama en Git?**

**Respuesta:**

Para crear una nueva rama en Git, primero asegúrate de estar en el directorio del repositorio en tu terminal o línea de comandos. Luego, utiliza el comando `git branch` seguido del nombre de la nueva rama que deseas crear.

• **¿Cómo cambiar a una rama en Git?**

**Respuesta:**

Puedes cambiarte a una rama usando el comando `git checkout` seguido del nombre de la rama. O, en Git versión 2.23 y superior, puedes utilizar el comando `git switch` para cambiar a la nueva rama.

**• ¿Cómo fusionar ramas en Git?**

**Respuesta:**

Pimero, asegúrate de estar en la rama en la que deseas fusionar los cambios.Luego, utiliza el comando `git merge` seguido del nombre de la rama que deseas fusionar. Git intentará fusionar automáticamente los cambios.

**• ¿Cómo crear un commit en Git?**

**Respuesta:**

Primero, asegúrate de haber realizado cambios en los archivos de tu repositorio. Luego, utiliza el comando `git add` seguido de los nombres de los archivos que deseas incluir en el commit para preparar los cambios. Después, utiliza el comando `git commit -m "Mensaje del commit"` para confirmar los cambios preparados.

**• ¿Cómo enviar un commit a GitHub?**

**Respuesta:**

Se puede hacer utilizando el comando `git push origin nombre-de-la-rama`. Despues de hacer git add y git commit.

**• ¿Qué es un repositorio remoto?**

**Respuesta:**

Un repositorio remoto en Git es una versión de un proyecto que está alojada en un servidor en línea o en otro lugar accesible por internet. Estos repositorios actúan como puntos de conexión centralizados donde los colaboradores pueden contribuir al proyecto. Además de servir como una copia de seguridad segura, los repositorios remotos permiten la colaboración en equipo, ya que múltiples personas pueden trabajar en el mismo proyecto y compartir sus contribuciones a través de estas versiones en línea.

**• ¿Cómo agregar un repositorio remoto a Git?**

**Respuesta:**

Para agregar un repositorio remoto a Git, utiliza el comando git remote add nombre-remoto URL-del-repositorio.

**• ¿Cómo empujar cambios a un repositorio remoto?**

**Respuesta:**

Para empujar (push) cambios a un repositorio remoto en Git, primero, asegúrate de haber hecho un commit local de tus cambios. Luego, utiliza el comando git push nombre-remoto nombre-rama para enviar los cambios al repositorio remoto.

**• ¿Cómo tirar de cambios de un repositorio remoto?**

**Respuesta:**

Para tirar (pull) cambios de un repositorio remoto a tu repositorio local, utiliza el comando git pull nombre-remoto nombre-rama.

**• ¿Qué es un fork de repositorio?**

**Respuesta:**

Un fork en Git es una copia independiente de un repositorio ajeno en tu cuenta de GitHub (u otra plataforma de alojamiento de repositorios). Esto permite que puedas trabajar en el proyecto sin afectar el repositorio original.

**• ¿Cómo crear un fork de un repositorio?**

**Respuesta:**

Para crear un fork de un repositorio en GitHub, visita el repositorio que deseas forkear y haz clic en el botón "Fork" en la esquina superior derecha de la página. Esto creará una copia del repositorio en tu cuenta de GitHub. Luego, puedes clonar tu fork a tu máquina local usando git clone URL-del-fork.

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

**Respuesta:**

Visita la página del repositorio original en GitHub y haz clic en la pestaña "Pull Requests". Haz clic en el botón "New pull request" y selecciona las ramas que quieres comparar: la rama del repositorio original y tu rama con los cambios. Proporciona un título descriptivo y una descripción detallada de los cambios que has realizado. Después de revisar la solicitud, haz clic en "Create pull request".

**• ¿Cómo aceptar una solicitud de extracción?**

**Respuesta:**

Haz clic en el botón verde "Merge pull request". Esto integrará los cambios en la rama objetivo.

**• ¿Qué es un etiqueta en Git?**

**Respuesta:**

En Git, una etiqueta es una referencia a un punto específico en la historia del repositorio, generalmente utilizado para marcar versiones significativas de un proyecto, como lanzamientos (releases) o hitos importantes.

**• ¿Cómo crear una etiqueta en Git?**

**Respuesta:**

Para crear una etiqueta en Git, primero utiliza el comando `git tag -a nombre-de-la-etiqueta -m "Mensaje descriptivo"` para crear una etiqueta anotada.

**• ¿Cómo enviar una etiqueta a GitHub?**

**Respuesta:**

Una vez que hayas creado la etiqueta local, puedes enviarla al repositorio remoto en GitHub utilizando el comando `git push origin nombre-de-la-etiqueta`.

**• ¿Qué es un historial de Git?**

**Respuesta:**

El historial de Git se refiere al registro detallado de todos los cambios realizados en un repositorio de Git a lo largo del tiempo. Git almacena el historial de manera eficiente, guardando instantáneas (commits) de los archivos en lugar de duplicar todo el contenido en cada cambio.

**• ¿Cómo ver el historial de Git?**

**Respuesta:**

Para ver el historial de Git, puedes utilizar el comando `git log` en tu terminal o línea de comandos dentro del directorio del repositorio.

**• ¿Cómo buscar en el historial de Git?**

**Respuesta:**

Para buscar en el historial de Git, puedes utilizar el comando `git log` junto con diversas opciones y filtros para refinar tus búsquedas. Puedes buscar commits que contienen un mensaje específico con

--grep=”termino de busqueda”.

**• ¿Cómo borrar el historial de Git?**

**Respuesta:**

Eliminar el historial solo para ciertos archivos o carpetas sin eliminar el proyecto en sí, puedes usar el comando `filter-branch` .

• **¿Qué es un repositorio privado en GitHub?**

**Respuesta:**

Un repositorio privado en GitHub es un espacio de almacenamiento en línea donde puedes gestionar y versionar tu código fuente de forma segura, limitando el acceso solo a las personas que invites.

**• ¿Cómo crear un repositorio privado en GitHub?**

**Respuesta:**

Para crear un repositorio privado en GitHub, inicia sesión en tu cuenta, haz clic en el botón "New" en la página principal de GitHub, luego proporciona un nombre para tu repositorio, elige la opción "Private" para establecerlo como privado, y haz clic en "Create repository".

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

**Respuesta:**

Para invitar a alguien a tu repositorio privado en GitHub, navega al repositorio, haz clic en la pestaña "Settings", luego en "Manage access". Desde allí, haz clic en el botón "Invite a collaborator" y escribe el nombre de usuario o la dirección de correo electrónico del colaborador que deseas invitar. Después de seleccionar el usuario, haz clic en "Add <usuario>" para enviar la invitación. La persona invitada recibirá una notificación y podrá acceder al repositorio privado una vez que acepte la invitación.

**• ¿Qué es un repositorio público en GitHub?**

**Respuesta:**

Un repositorio público en GitHub es un espacio de almacenamiento en línea donde puedes alojar y compartir tu código fuente con cualquier persona en internet.

**• ¿Cómo crear un repositorio público en GitHub?**

**Respuesta:**

Crear un repositorio público en GitHub es similar a crear uno privado. Inicia sesión en tu cuenta, haz clic en el botón "New", proporciona un nombre para tu repositorio y asegúrate de que la opción "Public" esté marcada. Luego, haz clic en "Create repository" para crear el repositorio público.

**• ¿Cómo compartir un repositorio público en GitHub?**

**Respuesta:**

Para compartir un repositorio público en GitHub, simplemente comparte la URL del repositorio con cualquier persona que desees. Puedes copiar la URL desde la barra de direcciones del navegador o utilizar el botón "Code" en la página del repositorio para copiar la URL del clon.

**• ¿Qué es un repositorio de código en GitHub?**

**Respuesta:**

Un repositorio de código en GitHub es un espacio donde se almacena, gestiona y versiona el código fuente de un proyecto.

**• ¿Cómo crear un repositorio de código en GitHub?**

**Respuesta:**

Para crear un repositorio de código en GitHub, primero, inicia sesión en tu cuenta de GitHub y haz clic en el botón "New" en la esquina superior derecha de la página principal. Luego, proporciona un nombre para tu repositorio, una descripción opcional y selecciona si quieres que sea público o privado. Después de configurar las opciones según tus preferencias, haz clic en "Create repository".

**• ¿Cómo compartir un repositorio de código en GitHub?**

**Respuesta:**

Para compartir un repositorio de código en GitHub, primero, asegúrate de que sea público (o que hayas concedido acceso a las personas adecuadas en caso de ser privado). Luego, simplemente comparte la URL del repositorio con las personas que deseas invitar.

**• ¿Qué es un repositorio de documentación en GitHub?**

**Respuesta:**

Un repositorio de documentación en GitHub es un espacio donde se almacenan y gestionan documentos, manuales, guías de usuario u otros tipos de documentación relacionada con un proyecto.

**• ¿Cómo crear un repositorio de documentación en GitHub?**

**Respuesta:**

Crear un repositorio de documentación en GitHub sigue el mismo proceso que crear un repositorio de código. Inicia sesión en tu cuenta de GitHub, haz clic en "New", proporciona un nombre para tu repositorio y selecciona si quieres que sea público o privado.

**• ¿Cómo compartir un repositorio de documentación en GitHub?**

**Respuesta:**

Para compartir un repositorio de documentación en GitHub, asegúrate de que sea público y, al igual que con un repositorio de código, simplemente comparte la URL del repositorio con las personas que desees invitar.

**• ¿Qué es un repositorio de proyecto en GitHub?**

**Respuesta:**

Un repositorio de proyecto en GitHub es una herramienta para organizar y administrar el trabajo en torno a un proyecto específico.

**• ¿Cómo usar Git para trabajar con otros?**

**Respuesta:**

Para trabajar con otros en Git, primero, asegúrate de que todos estén trabajando en el mismo repositorio. Cada persona debe clonar el repositorio a su máquina local. Luego, cada colaborador puede crear su propia rama para trabajar en su parte del proyecto sin interferir con el trabajo de los demás. Una vez que se han hecho los cambios y se han confirmado (commit), estos pueden ser fusionados (merge) en la rama principal o en otras ramas pertinentes para integrar el trabajo de todos. El uso adecuado de ramas y la comprensión de los comandos básicos de Git, como git clone, git branch, git commit, y git merge, son esenciales para trabajar de manera colaborativa y eficiente en proyectos compartidos.