Instituto Tecnológico de Las Américas

Materia
Programación III

Docente
Kelyn Tejada Belliard

Tema
Tarea 3

Estudiantes
Albert De Los Santos

Matrículas
2023-0553

Fecha de entrega
Viernes, 4 de abril de 2025

Contenido

Parte 1 – Cuestionario (30%):

¿Qué es Git?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

- **Presentar o compartir** el trabajo.
- **Seguir y administrar** los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
- **Colaborar** en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

2. ¿Para qué sirve el comando git init?

git init es un comando que se utiliza una sola vez durante la configuración inicial de un repositorio nuevo. Al ejecutar este comando, se creará un nuevo subdirectorio .git en tu directorio de trabajo actual. También se creará una nueva rama principal.

3. ¿Qué es una rama en Git?

Las ramas son una de las principales utilidades que disponemos en Git para llevar un mejor control del código. Se trata de una bifurcación del estado del código que crea un nuevo camino de cara a la evolución del código, en paralelo a otras ramas que se puedan generar.

Una rama Git es simplemente un apuntador móvil apuntando a una de esas confirmaciones. La rama por defecto de Git es la rama master. Con la primera confirmación de cambios que realicemos, se creará esta rama principal master apuntando a dicha confirmación.

4. ¿Cómo saber en cuál rama estoy trabajando?

Para saber qué ramas están disponibles y cuál es el nombre de la rama actual, ejecutamos git branch.

5. ¿Quién creó Git?

Linus Torvalds Junio Hamano Software Freedom Conservancy el 7 de abril de 2005.

En 2005, la relación entre la comunidad que desarrolló el kernel de Linux y la empresa que desarrolló BitKeeper se rompió, y se revocó la gratuidad de la herramienta. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basándose en algunas de las lecciones aprendidas con BitKeeper. Algunos de los objetivos del nuevo sistema eran los siguientes:

- Velocidad
- Diseño simple
- Fuerte soporte para el desarrollo no lineal (miles de ramas paralelas)
- Totalmente distribuido
- Capaz de manejar proyectos grandes como el kernel de Linux de manera eficiente (velocidad y tamaño de datos)

Desde su creación en 2005, Git ha evolucionado y madurado para ser fácil de usar y, a la vez, conservar sus cualidades iniciales.

6. ¿Cuáles son los comandos esenciales de Git?

git init: Crea un nuevo repositorio GIT en el directorio actual o en un nuevo proyecto.

git clone: Copia un repositorio existente, ya sea remoto o local, en tu máquina local.

git add: Agrega archivos al área de preparación (staging).

git commit: Guarda los cambios preparados en el repositorio con un mensaje

git status: Muestra el estado del repositorio, indicando qué archivos han sido modificados, añadidos o están listos para el commit.

git push: Envía tus commits locales al repositorio remoto.

git pull: Descarga y fusiona los cambios del repositorio remoto al local.

git branch: Muestra las ramas actuales, crea una nueva rama o elimina una existente.

git checkout: Cambia entre ramas o crea una nueva rama y cambia a ella.

git merge: Fusiona una rama con otra. Esto suele utilizarse después de terminar el desarrollo en una rama de características.

git log: Muestra el historial de commits del repositorio.

git reset: Deshace cambios locales y reinicia el área de preparación al último commit.

git rm: Elimina archivos del área de preparación y del directorio de trabajo.

git stash: Guarda temporalmente los cambios no confirmados y limpia el área de trabajo para poder cambiar de ramas o realizar otras tareas sin perder tu trabajo actual.

git fetch: Descarga los cambios del repositorio remoto sin fusionarlos inmediatamente.

7. ¿Qué es Git Flow?

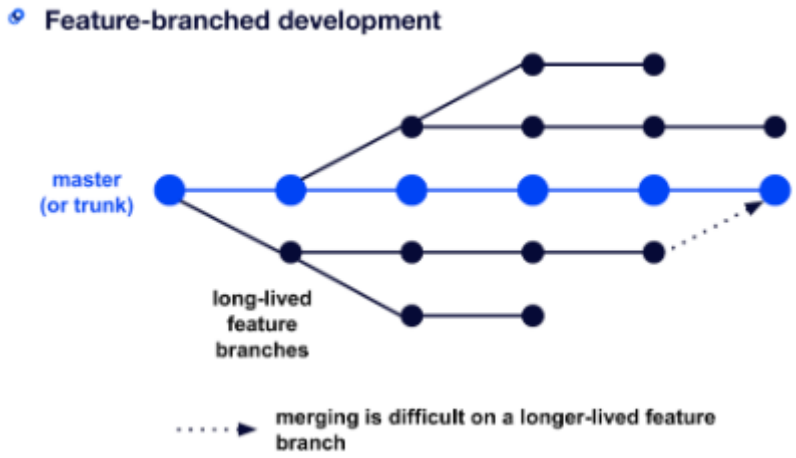
Gitflow es un modelo de ramificación para Git que ofrece un enfoque estructurado para el desarrollo de software. Define ramas específicas para diferentes propósitos y describe cómo deben interactuar. El objetivo es agilizar el proceso de desarrollo, gestionar los lanzamientos (*releases*) de manera eficaz y facilitar la colaboración entre los miembros del equipo.

8. ¿Qué es el desarrollo basado en trunk (Trunk Based Development)?

El desarrollo basado en troncos (TBD) es una estrategia de desarrollo de software en la que los ingenieros integran cambios menores con mayor frecuencia en el código base principal y trabajan a partir de la copia principal en lugar de trabajar en ramas de características de larga duración. Este modelo de desarrollo se suele utilizar como parte de un flujo de trabajo de desarrollo de integración continua.

Con muchos ingenieros trabajando en la misma base de código, es importante contar con una estrategia para el control del código fuente y la colaboración entre los ingenieros.

En el desarrollo de ramas de características, los desarrolladores de software individuales o los equipos de ingenieros no fusionan su nueva rama hasta que una característica está completa, a veces trabajando durante semanas o meses seguidos en una copia separada.



Parte 2 – Proyecto Práctico (70%):

Link GitHub: https://github.com/AyB05/2025-C3-P3-Tarea3.git

Bibliografía

https://docs.github.com/es/get-started/start-your-journey/about-github-and-git

https://www.atlassian.com/es/git/tutorials/setting-up-a-repository#:~:text=git%20init%20es%20un%20comando,crear%C3%A1%20una%20nueva%20rama%20principal.

https://www.arsys.es/blog/ramas-git

https://www.atlassian.com/es/git/tutorials/using-branches/git-checkout#:~:text=Para%20saber%20qu%C3%A9%20ramas%20est%C3%A1n,rama%20actual%2C%20ejecuta%20git%20branch%20.&text=En%20el%20ejemplo%20anterior%2C%20se,este%20caso%2C%20la%20rama%20feature_inprogress_branch%20.

https://es.wikipedia.org/wiki/Git

https://www.swhosting.com/es/blog/comandos-basicos-de-git-guia-para-principiantes

https://davidregalado255.medium.com/qu%C3%A9-es-gitflow-36dc19b5b8ec#:~:text=Gitflow%20es%20un%20modelo%20de%20ramificaci%C3%B3n%20potente%20que%20puede%20ayudar,gesti%C3%B3n%20eficaz%20de%20las%20versiones.

https://www.optimizely.com/optimization-glossary/trunk-based-development/