Git y GitHub

Creación de una historia colaborativa

# Objetivo

Con esta actividad se busca que los estudiantes sean capaces de reconocer y aplicar los conceptos básicos de control de versiones utilizando Git y GitHub en un entorno de desarrollo colaborativo a través de la **creación de una historia colaborativa**. Los estudiantes adquirirán habilidades básicas en el manejo de ramas, commits, pull, push, y comprenderán el flujo de trabajo colaborativo. Además, podrán verificar la autoría de las contribuciones mediante herramientas como "blame", fomentando así la responsabilidad individual en el contexto del trabajo colaborativo.

# Requisitos

* Esta actividad se realiza en equipos de trabajo entre 2 y 3 personas.
* La actividad se realizará utilizando cuentas de **GitHub** creadas con la cuenta de **correo** institucional **de La Universidad**. Si aún no tiene una cuenta con estas características, puede crearla en <https://github.com>.
* Instalar **GitHub Desktop** en su equipo local. Si aún no lo tiene instalado, siga las instrucciones en [https://desktop.github.com](https://desktop.github.com/)

# Preparación Inicial

En esta actividad crearán una breve historia, pero antes de iniciar en la escritura, es esencial que el equipo establezca un argumento general para la historia. Definan los personajes principales, los sucesos clave y la forma en que se resolverán los conflictos de la historia. Este paso ayudará a darle coherencia y dirección a la narrativa.

Una vez que tengan el esbozo general de la historia, deben asignar varias partes de la historia a cada miembro del equipo. Distribuir las responsabilidades permitirá que cada uno contribuya en múltiples ocasiones sin tener que trabajar de manera consecutiva. Aprovechen la diversidad de habilidades y perspectivas en su equipo para crear una historia única y cautivadora.

Definan también quien será el propietario del proyecto, su papel será crucial en la gestión del repositorio y la coordinación del equipo. Esta persona deberá crear el repositorio en GitHub, invitar a todos los colaboradores a unirse al repositorio para que puedan comenzar a contribuir, revisar y fusionar las contribuciones realizadas por los miembros del equipo, así como de comunicar las directrices sobre el flujo de trabajo, el estilo de escritura y cualquier requisito específico para la colaboración.

# Configuración de repositorio y colaboradores

1. **Creación del repositorio:** El propietario del proyecto se encargará de [Crear el primer repositorio mediante GitHub Desktop](https://docs.github.com/es/desktop/overview/creating-your-first-repository-using-github-desktop) (**saltarse** la sección “Crear y clonar un repositorio **de tutorial**”).
   1. **IMPORTANTE:** Este paso se realiza con la cuenta del propietario del proyecto, sin embargo, es importante que todos los miembros del equipo hagan el seguimiento de la actividad para identificar cómo se realiza la creación del proyecto y el envío de cambios.
   2. **En la parte 5, en el numeral 2**, al modificar el archivo **README.md**, en vez de poner la descripción del proyecto, pondremos **solamente** el título de su historia colaborativa.
2. Agregar colaboradores: El propietario del proyecto, en GitHub Desktop, deberá ir al menú **Repository > View on GitHub**. Una vez allí, sigan las indicaciones para [Invitar colaboradores a un repositorio personal - Documentación de GitHub](https://docs.github.com/es/account-and-profile/setting-up-and-managing-your-personal-account-on-github/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository)
3. Luego de agregar a los colaboradores, los demás miembros del equipo deberán [Clonar ~~y bifurcar~~ repositorios desde GitHub Desktop](https://docs.github.com/es/desktop/adding-and-cloning-repositories/cloning-and-forking-repositories-from-github-desktop). Importante: Lean atentamente la sección **Acerca de los repositorios locales** antes de **clonar un repositorio**. No es necesario revisar la parte de bifurcaciones por ahora.
4. Una vez finalizada esta configuración, cada uno de los integrantes del equipo tendrá en su máquina local una versión propia del repositorio de la historia.

# Creación del contenido

Para mantener un flujo de trabajo organizado y prevenir conflictos en la rama principal, es esencial que cada miembro del equipo cree su propia rama para realizar contribuciones individuales.

1. De manera individual, usa las indicaciones de [Administración de ramas en GitHub Desktop](https://docs.github.com/es/desktop/making-changes-in-a-branch/managing-branches-in-github-desktop) para que cada integrante cree tenga su propia rama de trabajo.

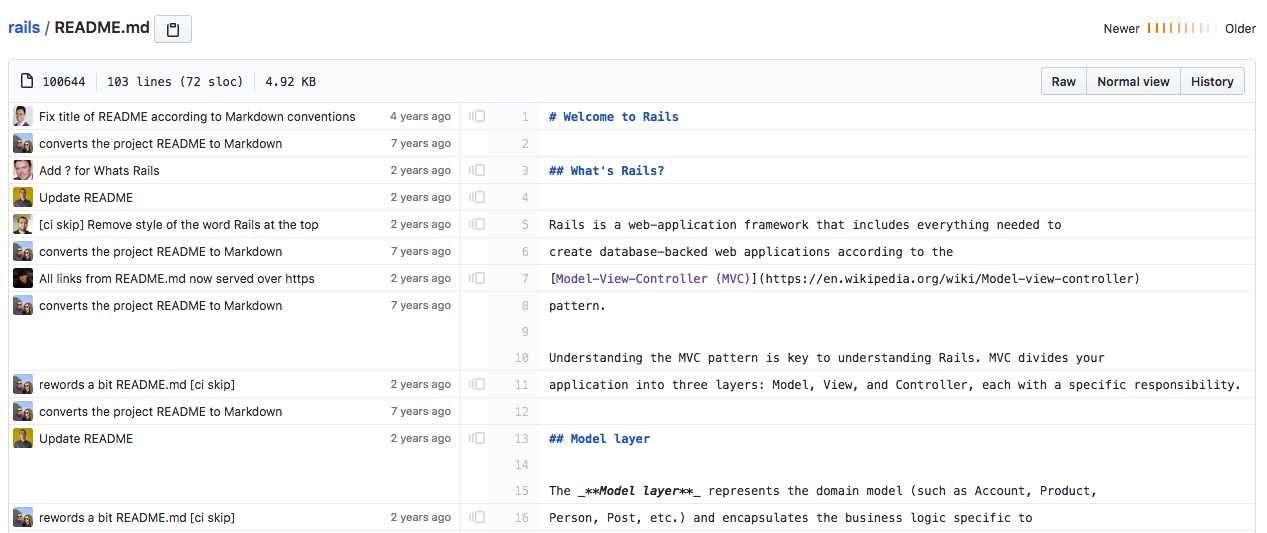
**NOTA IMPORTANTE:** La creación de ramas puede parecer demasiado para esta actividad que es pequeña, pero en la vida real, este enfoque no solo facilita la colaboración simultánea de múltiples personas, sino que también minimiza el riesgo de impactar el trabajo en progreso de otros al evitar commits inestables a la rama principal.

1. ¡Llegó el momento de la creatividad! De forma similar a como se hizo en la [Parte 5: Hacer, confirmar y subir cambios](https://docs.github.com/es/desktop/overview/creating-your-first-repository-using-github-desktop#part-5-making-committing-and-pushing-changes) de las instrucciones para crear el repositorio, agrega tu parte de la historia el archivo README.md, no olvides guardar, confirmar y subir los cambios al repositorio principal en tu rama propia.
2. Conforme se vayan completando las partes de la historia, el propietario del proyecto debe encargarse de unir los cambios de las diferentes ramas a la rama principal. Para lograrlo podrás seguir las instrucciones en [Syncing your branch in GitHub Desktop – Merging another branch into your project branch](https://docs.github.com/en/desktop/working-with-your-remote-repository-on-github-or-github-enterprise/syncing-your-branch-in-github-desktop#merging-another-branch-into-your-project-branch). Esta actividad se debe realizar en conjunto con el creador de cada rama para validar la persistencia de los cambios, en especial al presentarse conflictos.

# Validación del resultado final

Una vez finalizados los cambios, validen el contenido definitivo de su historia colaborativa. De ser necesario, pueden hacer ajustes finales para mejorar la coherencia de la redacción, no olviden guardarlos, confirmarlos y subirlos.

Para finalizar, abran el contenido del archivo README en GitHub y muestren el resultado final al profesor. Él revisará el “blame” del archivo y validará que **cada uno de los integrantes** tenga al menos una contribución en el archivo final.



# Recursos:

A continuación, algunos recursos que podrían ser de interés para esta actividad o trabajo posterior:

* <https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/GitHub>
* <https://dangitgit.com/>
* <https://github.com/k88hudson/git-flight-rules/blob/master/README_es.md>