



GitHub

INTRODUCCION A REPOSITORIOS





Anuncios

- ◆ Recordar a estudiantes entregar actividades pendientes.
- ◆ Recordar a estudiantes realizar el cuestionario del sábado.
- ◆ Retroalimentación de actividad del sábado.



Que es GitHub.

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

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

- 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.



GIT

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas y tú están haciendo cambios en los mismos archivos al mismo tiempo.





GIT

Normalmente, para hacerlo en un flujo de trabajo basado en Git, harías lo siguiente:

- **Crear** una rama a partir de la copia principal de archivos en los que tú (y tus colaboradores) estáis trabajando.
- **Realizar** modificaciones en los archivos de forma independiente y segura en tu propia rama personal.
- **Dejar** que Git fuse en mediante combinación y de forma inteligente los cambios específicos en la copia principal de archivos, de modo que los cambios no afecten a las actualizaciones de otras personas.
- **Dejar que Git** realice un seguimiento de tus cambios y los de otras personas, por lo que todos siguen trabajando en la versión más actualizada del proyecto.



Como funcionan GIT y GitHub juntos

Al cargar archivos en GitHub, los almacena en un "repositorio de Git". Esto significa que al realizar cambios (o "compromisos") en los archivos de GitHub, Git se iniciará automáticamente para realizar un seguimiento y administrar los cambios.

Hay muchas acciones relacionadas con Git que puedes completar en GitHub directamente en el navegador, como crear un repositorio de Git, crear ramas y cargar y editar archivos.



Como funcionan GIT y GitHub juntos

Sin embargo, la mayoría de las personas trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando se empiece a colaborar con otros y todos necesiten trabajar en el mismo repositorio al mismo tiempo, haran lo siguiente continuamente:

- Extraer todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub.
- Transferir de vuelta los cambios al mismo repositorio remoto en GitHub.



Acerca del control de versiones

Sin embargo, la mayoría de las personas trabajan en sus archivos localmente (en su propio ordenador), luego sincronizan continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando empieces a colaborar con otros y todos necesitéis trabajar en el mismo repositorio al mismo tiempo, haréis lo siguiente continuamente: Extraer todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub. Transferir de vuelta los cambios al mismo repositorio remoto en GitHub.



Que se puede hacer con git

Los desarrolladores pueden revisar el historial de proyectos para averiguar:

- ¿Qué cambios se hicieron?
- ¿Quién los hizo?
- ¿Cuándo se hicieron?
- ¿Por qué se necesitaban?



git

En un sistema de control de versiones distribuido, cada desarrollador tiene una copia integral del proyecto y del historial del mismo. A diferencia de los sistemas de control de versiones centralizados populares, los DVCS necesitan una conexión constante a un repositorio central. Git es el sistema de control de versiones distribuido más popular. Git se utiliza habitualmente tanto para el desarrollo de proyectos de código abierto como comerciales y tiene beneficios significativos para los individuos, equipos y negocios.



Comandos basicos de git

Para utilizar Git, los desarrolladores utilizan comandos específicos para copiar, crear, cambiar y combinar el código. Estos comandos pueden ejecutarse directamente desde la línea de comandos o utilizando una aplicación como GitHub Desktop. Aquí tienes algunos comandos comunes para utilizar Git:

- **git init** inicializa un repositorio nuevo de Git y comienza a supervisar el directorio existente. Este agrega una subcarpeta oculta dentro del directorio existente que hospeda la estructura de datos interna que se requiere para el control de versiones.
- **git clone** crea una copia local de un proyecto que ya existe remotamente. El clon incluye todos los archivos, historial y ramas del proyecto.



Comandos basicos de git

- **git add** almacena provisionalmente un cambio. Git rastrea los cambios que se hacen a la base de código de un desarrollador, pero es necesario probarlos y tomar una captura de pantalla de ellos para incluirla en el historial del proyecto. Este comando realiza pruebas, la primera parte de este proceso de dos pasos. Cualquier cambio que se pruebe, se convertirá en parte de la siguiente captura de pantalla y también del historial del proyecto. Las pruebas y confirmaciones por separado otorgan a los desarrolladores el control completo sobre el historial y sobre el proyecto sin cambiar la forma en la que codifican y trabajan.



Comandos basicos de git

- **git commit** guarda la instantánea del historial del proyecto y completa el proceso de seguimiento de los cambios. En resumen, una confirmación funciona tal como el tomar una fotografía. Todo lo que se haya almacenado provisionalmente con git add pasará a formar parte de la instantánea con git commit.
- **git status** muestra el estado de los cambios como sin seguimiento, modificados o almacenados provisionalmente.
- **git branch** muestra las ramas en las que se trabaja localmente.



Comandos basicos de git

- **git merge** combina las líneas de desarrollo. Este comando habitualmente se utiliza para combinar los cambios que se realizan en dos ramas distintas. Por ejemplo, un desarrollador podría hacer una fusión cuando necesite combinar los cambios de una rama de característica en la rama de desarrollo principal.
- **git pull** actualiza la línea de desarrollo local con actualizaciones de sus contrapartes remotas. Los desarrolladores utilizan este comando si un compañero de equipo hizo confirmaciones en una rama en un repositorio remoto y quieren reflejarlos en su ambiente local.
- **git push** actualiza el repositorio remoto con las confirmaciones realizadas localmente en una rama.



Diferencia entre Git y GitHub.

Git forma es definido como un Sistema de Control de Versiones Distribuido (DVCS), es decir administra y distribuye versiones de un archivo. En cambio, Github es una plataforma que permite alojar distribuciones de archivos de Git y está basada en la web.





GitHub Desktop

GitHub Desktop es una aplicación gratuita de código abierto que te ayuda a trabajar con código hospedado en GitHub u otros servicios de hospedaje de Git. Con GitHub Desktop, puedes realizar comandos de Git, como confirmar e insertar cambios, en una interfaz gráfica de usuario, en lugar de mediante la línea de comandos.

Puedes instalar GitHub Desktop en cualquier sistema operativo compatible. Para instalar GitHub Desktop, visite la página de descarga de [GitHub Desktop](#).



Instalacion de GitHub Desktop

Welcome to
GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

[Sign in to GitHub.com](#)

[Sign in to GitHub Enterprise](#)

New to GitHub? [Create your free account](#).

[Skip this step](#)

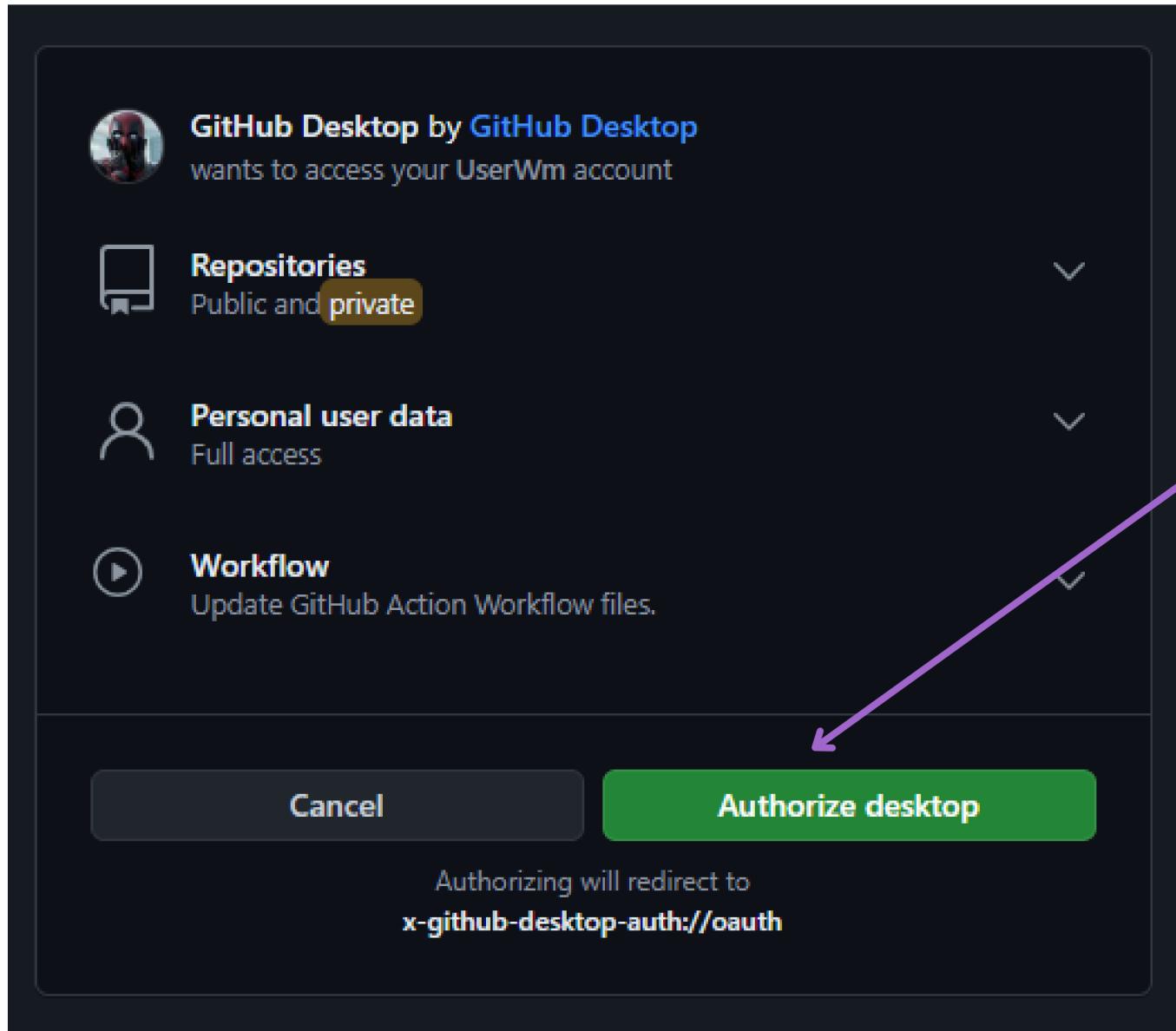
By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. [Learn more about user metrics](#).

Seleccionemos la primera
opcion e iniciamos sesión



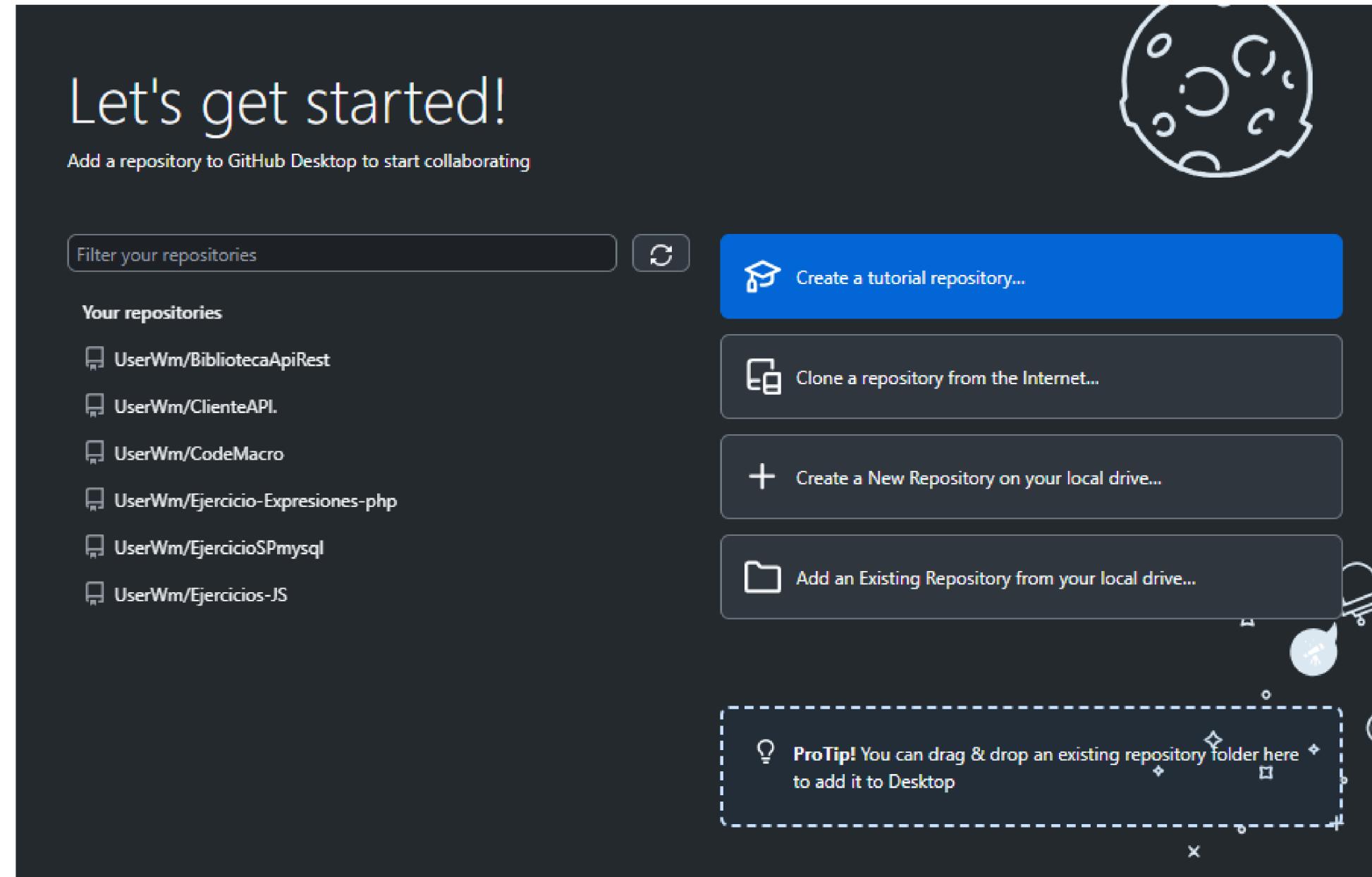
Autorizacion de repositorios



Autorizamos los repositorios y luego determinamos nuestro usuario y correo.



Listo tenemos configurado GitHub Desktop





Instalar Git



The screenshot shows the official Git website. The header features the Git logo and the tagline "distributed-even-if-your-workflow-isn't". A search bar and a "Google Translate" button are also present. The left sidebar includes links for "About", "Documentation", "Downloads" (which is highlighted in red), "GUI Clients", and "Logos". The main content area is titled "Downloads" and shows download links for "macOS", "Windows", and "Linux/Unix". A note below the links states: "Older releases are available and the Git source repository is on GitHub." To the right, a large image of a computer monitor displays the "Latest source Release" as "2.45.1" with a "Release Notes (2024-04-29)" link and a "Download for Windows" button.

git --distributed-even-if-your-workflow-isn't

About

Documentation

Downloads

GUI Clients

Logos

Community

macOS

Windows

Linux/Unix

Latest source Release
2.45.1
[Release Notes \(2024-04-29\)](#)

Download for Windows

Older releases are available and the Git source repository is on GitHub.

Descargue la version para su sistema operativo.



Abril la consola de git

Hasta este punto ya tenemos instalado git



Realicemos la practica de GitHub Desktop.

