



TIC



# Sesión 7

## Programación Nivel Básico



UNIVERSIDAD  
LIBRE®

eTraining®



Sesión 7:

# Trabajando con Git

Clonación de un repositorio remoto,  
colaboración mediante GitHub,  
creación de git page

# Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:



Brindar una introducción clara y concisa sobre la clonación de repositorios remotos y la colaboración en GitHub



Comprender la importancia de estas herramientas en el desarrollo de software moderno.



Conocer la utilidad de la clonación para acceder a código remoto, realizar cambios y colaborar de forma eficiente.



Explorar la plataforma de GitHub como un espacio centralizado para la colaboración, la gestión de versiones y el desarrollo de proyectos de código abierto.

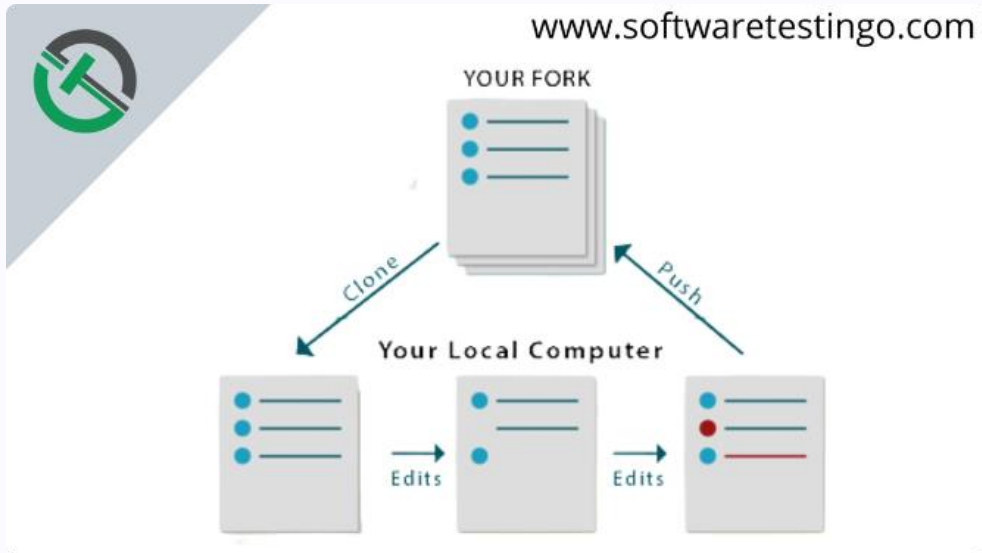


# Introducción a la clonación de repositorios remotos

En el mundo del desarrollo de software, la gestión de código es esencial para la colaboración y la eficiencia. Los repositorios de código, como los que se alojan en plataformas como GitHub, actúan como almacenes centrales donde se guarda el código fuente de un proyecto. La clonación de un repositorio remoto es un proceso fundamental que permite a los desarrolladores obtener una copia local del repositorio, lo que les brinda la posibilidad de trabajar en el código, realizar cambios y contribuir al proyecto.



# Definición de clonación de repositorios



La clonación de un repositorio remoto es el proceso de crear una copia local de un repositorio de código fuente que se encuentra alojado en un servidor remoto. Este proceso implica descargar todos los archivos, la historia de commits y las ramas del repositorio remoto a tu computadora, creando una réplica completa del repositorio.

1

## Crear una copia local

La clonación permite a los desarrolladores trabajar en el código de forma independiente, sin afectar el repositorio original. Esto es especialmente útil cuando se trabaja en colaboración, ya que cada desarrollador puede tener su propia copia local para realizar modificaciones.

2

## Acceso a la historia del código

La copia local no solo contiene el código actual, sino también toda la historia de commits, lo que permite rastrear los cambios realizados, identificar errores y volver a versiones anteriores del código.

3

## Prepararse para la colaboración

La clonación es el primer paso para colaborar en proyectos de código. Permite a los desarrolladores obtener una copia del código y comenzar a contribuir al proyecto de forma conjunta.



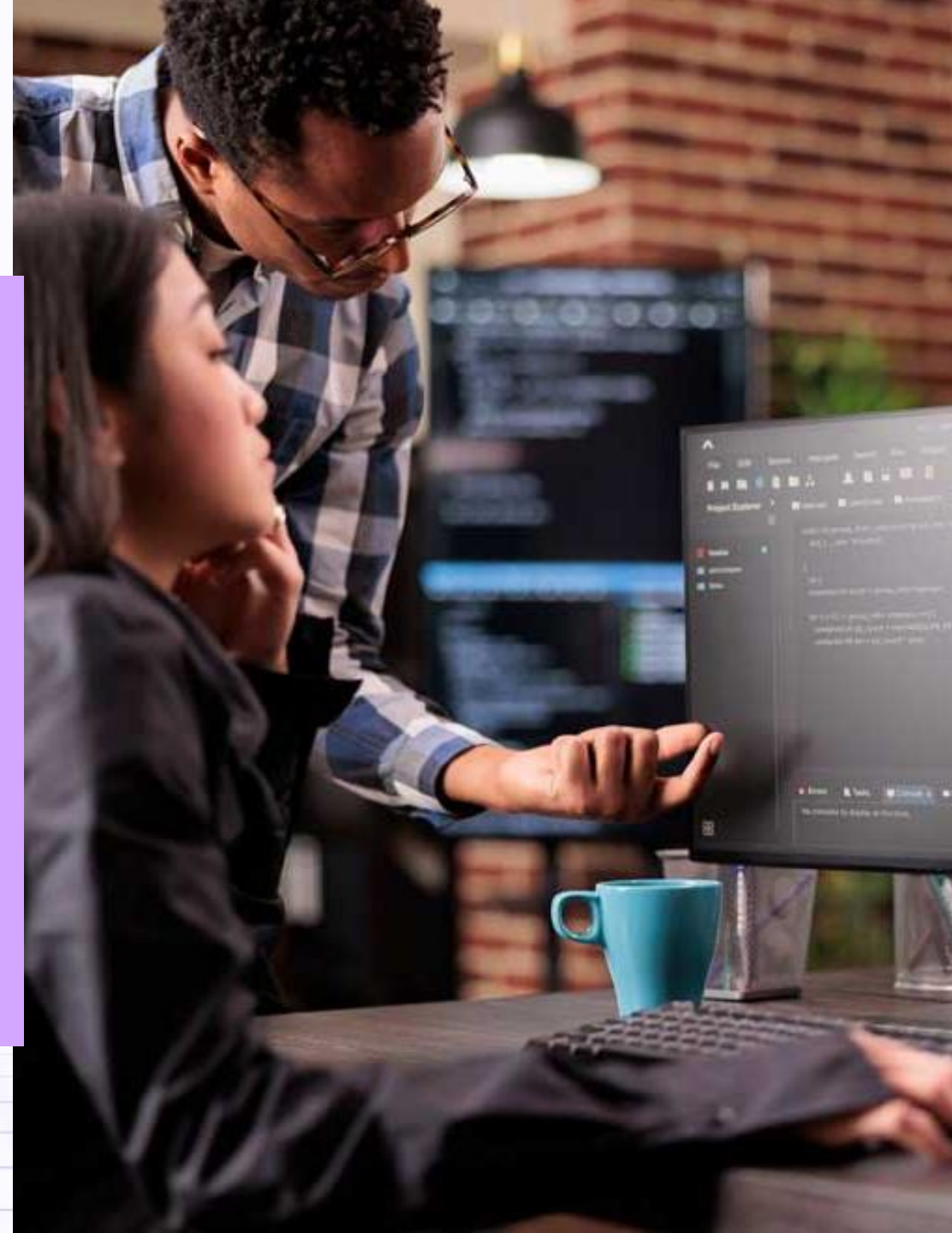
# Beneficios de clonar un repositorio remoto

Clonar un repositorio remoto ofrece numerosas ventajas para los desarrolladores, mejorando la colaboración, la eficiencia y la seguridad del desarrollo de software.

## Colaboración

La clonación facilita la colaboración entre equipos de desarrollo, ya que cada miembro puede obtener una copia local del repositorio y trabajar de forma independiente, sin afectar el código principal.

Los cambios realizados en la copia local se pueden sincronizar con el repositorio principal, permitiendo a otros miembros acceder a las actualizaciones.

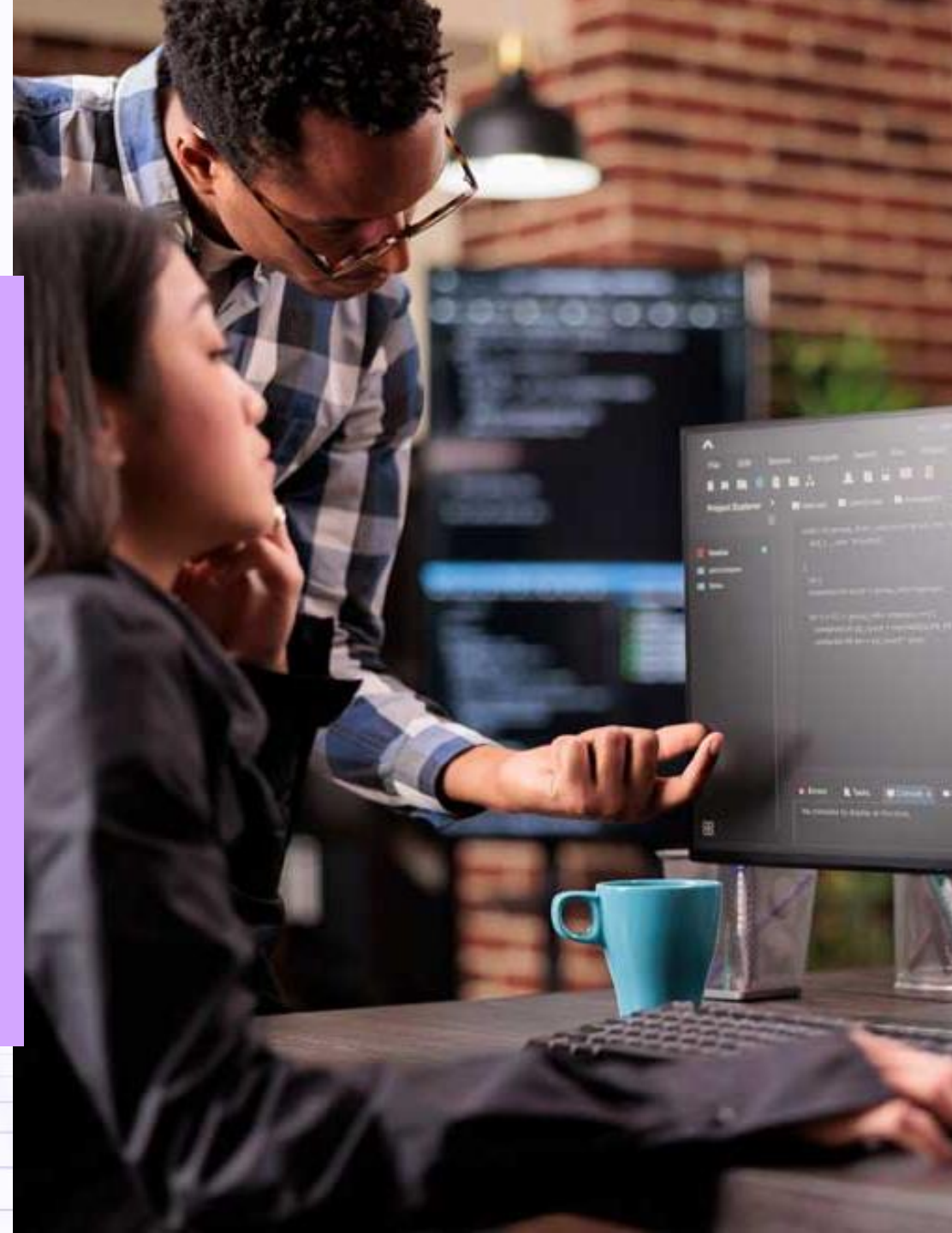


## Control de versiones

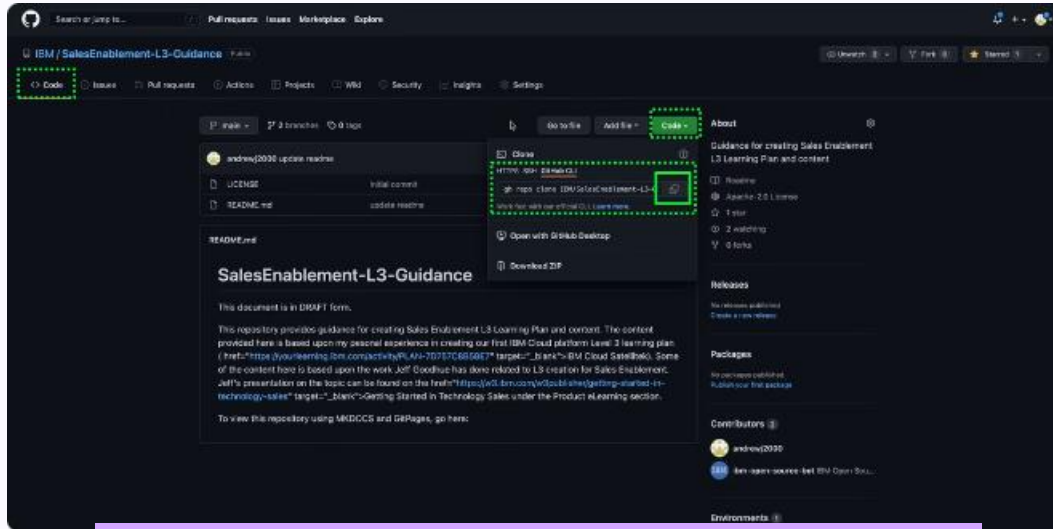
La clonación permite un control completo de la historia del código, al guardar todos los commits realizados, lo que facilita la detección de errores, la reversión a versiones anteriores y el seguimiento de los cambios realizados.

## Seguridad

Al clonar un repositorio, se crea una copia local del código, lo que proporciona una copia de seguridad en caso de pérdida o corrupción del repositorio remoto. Además, al trabajar en la copia local, los desarrolladores pueden realizar cambios sin afectar el código principal, asegurando la estabilidad del proyecto.



# Pasos para clonar un repositorio remoto



El proceso de clonación de un repositorio remoto es sencillo y se puede realizar utilizando herramientas de línea de comandos como Git o interfaces gráficas como GitHub Desktop.

1

## Obtener la URL del repositorio

El primer paso es obtener la URL del repositorio remoto que deseas clonar. Esta URL se puede encontrar en la página del repositorio en la plataforma donde está alojado, como GitHub.

2

## Abrir la terminal

Abre la terminal o la línea de comandos en tu computadora.

3

## Ejecutar el comando Git clone

Ejecuta el comando Git clone seguido de la URL del repositorio. Este comando descargará el repositorio remoto a tu computadora, creando una copia local en la carpeta actual.





# Colaboración mediante GitHub

GitHub es una plataforma popular para alojar repositorios de código fuente y facilitar la colaboración entre desarrolladores. Permite a los equipos de desarrollo trabajar juntos en proyectos de código, realizar seguimiento de los cambios, resolver problemas y construir software de forma eficiente.



## Creación de repositorios

GitHub permite crear repositorios públicos y privados, donde se puede guardar el código fuente de un proyecto. Los repositorios públicos son accesibles a todos, mientras que los privados solo se pueden acceder por los miembros del equipo.

## Control de versiones

GitHub utiliza Git para gestionar el control de versiones del código, lo que permite a los desarrolladores rastrear los cambios realizados, revertir a versiones anteriores y colaborar de forma eficiente en proyectos de código.

## Colaboración en tiempo real

GitHub ofrece herramientas para la colaboración en tiempo real, permitiendo a los equipos trabajar juntos en el código, realizar comentarios, resolver conflictos y contribuir al proyecto de forma eficiente.

# Definición de colaboración en GitHub

La colaboración en GitHub implica la interacción entre varios desarrolladores para trabajar juntos en un proyecto de código. Esta colaboración puede abarcar diferentes áreas como la escritura de código, la revisión de código, la resolución de problemas y la gestión de tareas.

## Creación de pull requests

Los desarrolladores crean pull requests para solicitar que sus cambios se fusionen con el repositorio principal. Las pull requests permiten a otros miembros del equipo revisar el código y proporcionar comentarios antes de que se fusionen con el proyecto principal.

1

2

## Resolución de problemas

GitHub ofrece herramientas para la gestión de problemas, lo que permite a los equipos de desarrollo identificar, rastrear y resolver los errores encontrados en el código.

3

4

## Revisión de código

La revisión de código es un proceso esencial en la colaboración de GitHub, donde los desarrolladores examinan el código escrito por otros para identificar errores, mejorar la calidad del código y garantizar que cumple con las mejores prácticas de desarrollo.

## Gestión de tareas

GitHub ofrece funciones para gestionar las tareas asociadas a un proyecto de desarrollo, lo que permite a los equipos asignar tareas, realizar seguimiento del progreso y asegurar que todos los miembros del equipo estén informados sobre el estado del proyecto.



# Ventajas de la colaboración en GitHub

La colaboración en GitHub ofrece múltiples ventajas para los equipos de desarrollo, mejorando la eficiencia, la calidad del código y la comunicación.



## Trabajo en equipo

GitHub facilita la colaboración entre equipos de desarrollo, permitiendo que varios miembros trabajen juntos en el mismo proyecto de forma eficiente y organizada.



## Calidad del código

La revisión de código y la colaboración en GitHub ayudan a mejorar la calidad del código, al detectar errores, mejorar la legibilidad y garantizar que se cumplan las mejores prácticas de desarrollo.



## Comunicación

GitHub ofrece herramientas para la comunicación entre los miembros del equipo, como los comentarios en pull requests, la gestión de problemas y la mensajería instantánea, lo que facilita la colaboración y la resolución de problemas.



## Velocidad

La colaboración en GitHub agiliza el proceso de desarrollo, al permitir que los equipos trabajen juntos en tiempo real, resuelvan problemas de forma rápida y compartan el código de forma eficiente.



# Slate theme

Slate is a theme for GitHub Pages.

## Ejemplos de colaboración en proyectos

La colaboración en GitHub es un proceso fundamental en el desarrollo de software, y se puede aplicar a una amplia variedad de proyectos, desde aplicaciones web y móviles hasta proyectos de código abierto y de investigación.

<b>**Proyecto**</b>	<b>**Colaboración**</b>
Desarrollo de una aplicación web	Un equipo de desarrolladores trabaja juntos en el diseño, la implementación y las pruebas de la aplicación.
Desarrollo de una biblioteca de software	Un grupo de desarrolladores contribuye a la creación, documentación y mantenimiento de una biblioteca de código reutilizable.
Proyecto de investigación	Los investigadores colaboran en la creación, análisis y publicación de los resultados de su investigación.



# Creación de una Git Page

Una Git Page es una página web estática creada a partir de un repositorio de Git. Es una forma sencilla y eficiente de mostrar y compartir contenido como código fuente, documentación, proyectos personales o incluso una simple página de perfil.





# Introducción

El uso de Git Pages se ha popularizado debido a su facilidad de implementación y su capacidad para albergar contenido web de manera rápida y sin necesidad de configurar un servidor web tradicional. Con solo unos pocos comandos, puedes publicar tu página web y compartirla con el mundo.

1

## Funcionalidad de Git Pages

Git Pages permite crear páginas web estáticas utilizando el repositorio de Git como fuente de contenido. Esto significa que tu página web estará alojada en un servidor de GitHub o GitLab, proporcionando un hosting gratuito y fácil de usar.





## 2 **Simpleza y Flexibilidad**

La creación y publicación de una Git Page se realiza con comandos sencillos, lo que la hace accesible para desarrolladores y personas sin experiencia previa en servidores web. La flexibilidad de Git te permite gestionar el contenido de tu página web de manera eficiente, actualizando y versionando cada cambio.

## 3 **Integración con Git**

Git Pages se integra perfectamente con la plataforma de Git, lo que facilita la gestión del contenido y el control de versiones. Puedes utilizar las herramientas y flujos de trabajo que ya conoces de Git para gestionar tu página web.

## 4 **Hosting Gratuito**

GitHub y GitLab ofrecen hosting gratuito para Git Pages, lo que la convierte en una opción atractiva para proyectos personales o sitios web de pequeño tamaño.



# Objetivos

Al crear una Git Page, te enfocas en crear un sitio web que muestre tu trabajo o tu portfolio. Esta página web puede ser utilizada para una variedad de propósitos, incluyendo:

## Compartir Código Fuente

Puedes crear una Git Page para mostrar el código fuente de tus proyectos y facilitar la colaboración. Puedes incluir ejemplos de código, documentación y herramientas para facilitar el uso y la comprensión de tu trabajo.

## Documentación

La creación de una Git Page es ideal para documentar proyectos o compartir información relevante con otros. Puedes incluir tutoriales, guías, artículos y recursos para facilitar la comprensión de un tema específico.

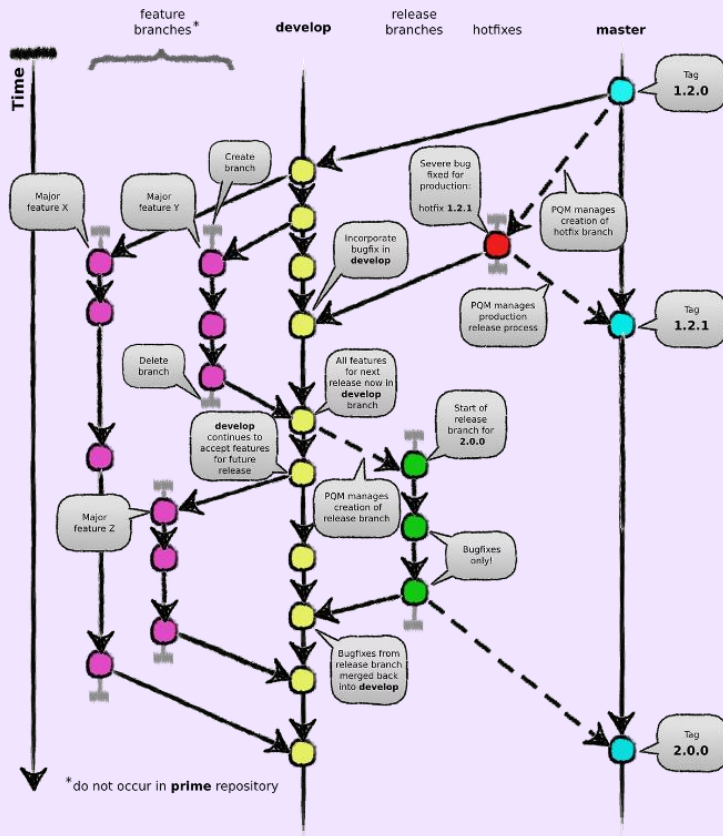
## Portfolio Personal

Puedes crear una Git Page para mostrar tu trabajo como desarrollador o diseñador. Puedes incluir información sobre tu experiencia, habilidades, proyectos realizados y ejemplos de tu trabajo.



# Definiciones

Para entender cómo funcionan las Git Pages, es importante comprender algunos conceptos básicos relacionados con Git:



## Repositorio de Git

Es un conjunto de archivos y su historial de cambios, almacenado en un servidor como GitHub o GitLab.

## Rama (Branch)

Una rama es una copia independiente del repositorio principal, que permite realizar cambios sin afectar el trabajo de otros.

## Página web estática

Un sitio web que no requiere la interacción de un servidor web dinámico para generar su contenido.

Owner



Repository name

GitHub-Pages-Example



Great repository names are short and memorable. Need inspiration? How about **studious-octo-sniffle**.

Description (optional)

# Ejemplos

Veamos algunos ejemplos de cómo se pueden utilizar las Git Pages:



## Portfolio

Puedes crear una Git Page para mostrar tus proyectos de desarrollo web o diseño. Esta página web puede contener información sobre ti, tus habilidades, y ejemplos de tu trabajo.



## Documentación

Puedes crear una Git Page para documentar un proyecto de código abierto, incluyendo información sobre su funcionamiento, instalación, ejemplos de uso, y detalles de la API.



## Blog

Puedes utilizar Git Pages para crear un blog personal o profesional, donde puedes compartir tus ideas, experiencias, y artículos técnicos.



## Código Fuente

Puedes crear una Git Page para mostrar el código fuente de un proyecto, incluyendo ejemplos de código, documentación, y herramientas para facilitar la colaboración.

# Casos de uso

Las Git Pages se utilizan en una gran variedad de escenarios, ofreciendo una solución sencilla y eficiente para diferentes necesidades:

1

## Proyectos de Código Abierto

Mostrar el código fuente, documentación y ejemplos de uso para facilitar la colaboración.

2

## Documentación de Productos

Proporcionar una plataforma para compartir información sobre productos, servicios o proyectos.

3

## Sitios web de Marketing

Crear landing pages, páginas de destino o páginas web de promoción para campañas de marketing.

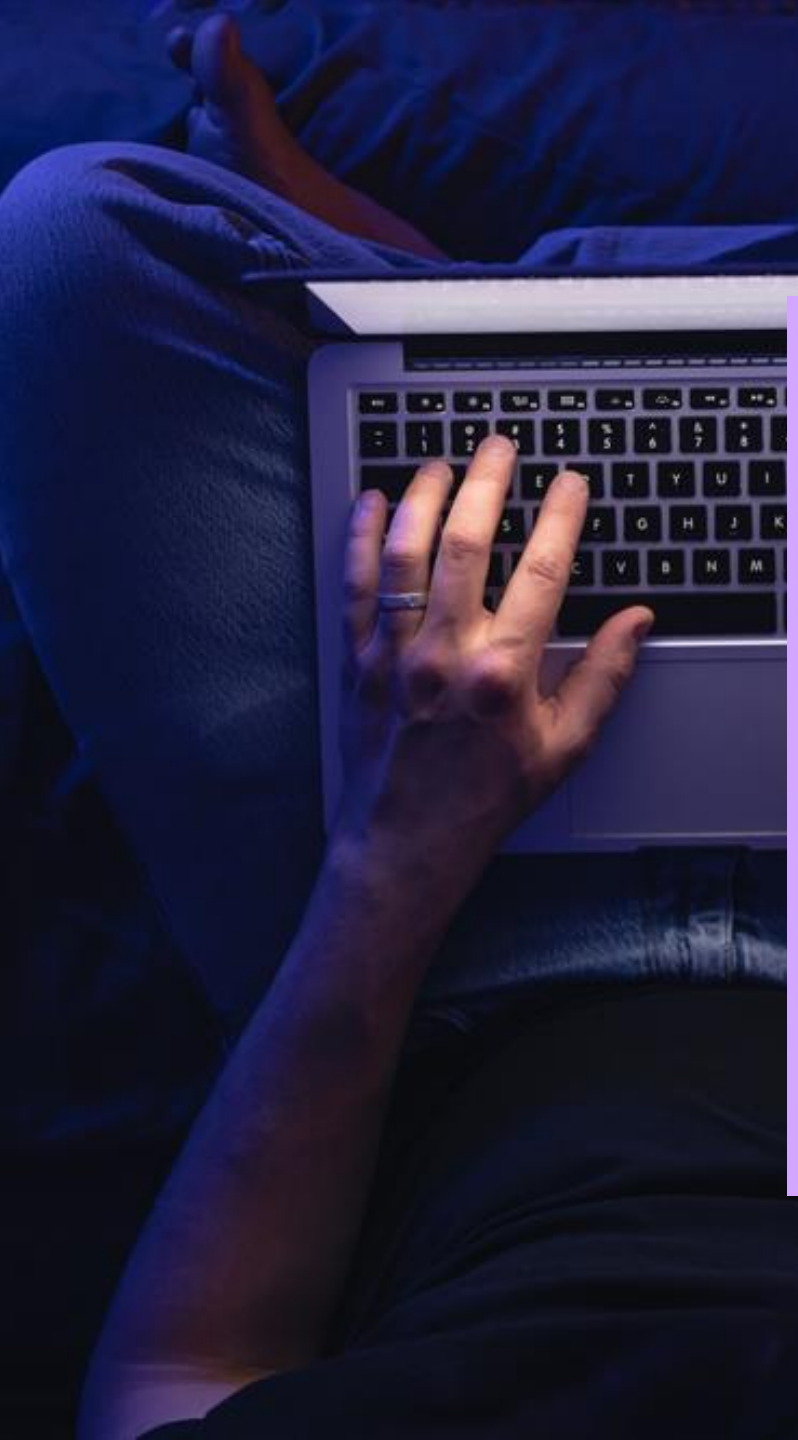
4

## Presentaciones y Portfolios

Mostrar trabajos realizados, habilidades y proyectos personales.







# Mejores prácticas

Para aprovechar al máximo las Git Pages, es importante seguir algunas mejores prácticas:

1

## Organización del Repositorio

Mantener una estructura de archivos clara y organizada para facilitar la gestión del contenido de tu página web.

2

## Uso de Markdown

Utilizar Markdown para formatear el contenido de tu página web, lo que facilita la escritura y la edición.

3

## Optimización del Rendimiento

Minimizar el tamaño de los archivos, optimizar las imágenes y utilizar técnicas de caché para mejorar el tiempo de carga de la página web.

4

## Seguridad

Implementar medidas de seguridad para proteger tu página web de ataques y amenazas cibernéticas.





## Conclusión

Crear una Git Page es una forma sencilla y efectiva de mostrar y compartir contenido online. La facilidad de uso, la integración con Git, el hosting gratuito y la flexibilidad para personalizar el contenido la convierten en una herramienta poderosa para desarrolladores, diseñadores y profesionales que desean crear sitios web estáticos de alta calidad.

# Recursos y material de apoyo



Existen numerosos recursos y materiales de apoyo disponibles para aprender sobre la clonación de repositorios remotos, la colaboración en GitHub y las mejores prácticas de desarrollo de software.

- **Documentación de Git:** <https://git-scm.com/doc>
- **Documentación de GitHub:** <https://docs.github.com/en>
- **Tutoriales de GitHub:** <https://guides.github.com/>
- **Foros de GitHub:** <https://github.com/topics>
- **Documentación de GitHub Pages:** <https://docs.github.com/en/pages>
- **Documentación de GitLab Pages:**  
<https://docs.gitlab.com/ee/user/project/pages/>
- **Tutoriales de Git Pages en YouTube:**  
[https://www.youtube.com/results?search\\_query=git+pages+tutorial](https://www.youtube.com/results?search_query=git+pages+tutorial)
- **Foros de GitHub y GitLab:** <https://github.com/topics/github-pages> y <https://gitlab.com/help>



# Ejercicios de práctica



**iGracias**  
por ser parte de  
esta experiencia  
de aprendizaje!

