Evidencias de desempeño: instalación y configuración de herramienta de versionamiento (Local / Web).

GA7-220501096-AA1-EV04

Introducción

En el ámbito del desarrollo de software, la gestión eficiente de versiones es fundamental para mantener la integridad, la colaboración y el seguimiento de los cambios en el código fuente y otros activos del proyecto. Las herramientas de control de versiones permiten a los equipos de desarrollo registrar y administrar las modificaciones realizadas en el código, facilitando la colaboración entre desarrolladores y asegurando un flujo de trabajo organizado y controlado.

Este documento tiene como objetivo proporcionar una guía práctica para la instalación y configuración de herramientas de versionamiento, tanto locales como basadas en web. Se explorarán dos herramientas ampliamente utilizadas en la industria del desarrollo de software: Git para versionamiento local y GitHub para hospedaje de repositorios en la nube.

En primer lugar, se abordará la instalación y configuración de Git, un sistema de control de versiones distribuido que permite a los desarrolladores mantener un historial completo de cambios y trabajar de manera eficiente en diferentes ramas de desarrollo. Luego, se explorará la configuración básica y el uso de GitHub, una plataforma de desarrollo colaborativo que utiliza Git como sistema de control de versiones central.

Al comprender y aplicar adecuadamente estas herramientas, los equipos de desarrollo podrán optimizar la gestión de código fuente, mejorar la colaboración entre miembros del equipo y asegurar la trazabilidad y la reproducibilidad de los proyectos de software.

Instalación de Git:

Guía paso a paso para la instalación de Git en diferentes sistemas operativos (Windows, macOS, Linux).

Configuración inicial de Git, incluyendo la configuración global de nombre de usuario y dirección de correo electrónico.

Configuración básica de Git:

Configuración de opciones adicionales en Git, como la configuración de alias y la configuración de colores.

Uso de comandos básicos de Git para inicializar un repositorio local, realizar confirmaciones (commits) y trabajar con ramas (branches).

Integración con GitHub:

Creación de una cuenta en GitHub y configuración inicial de la autenticación mediante claves SSH.

Creación de un repositorio remoto en GitHub y vinculación con un repositorio local existente.

Gestión básica de repositorios en GitHub:

Uso de GitHub Desktop para gestionar repositorios de manera gráfica.

Uso de comandos básicos de Git para sincronizar cambios entre repositorios locales y remotos.

Exploración de características avanzadas:

Uso de ramificaciones (branches) y fusiones (merges) para gestionar el desarrollo paralelo de características.

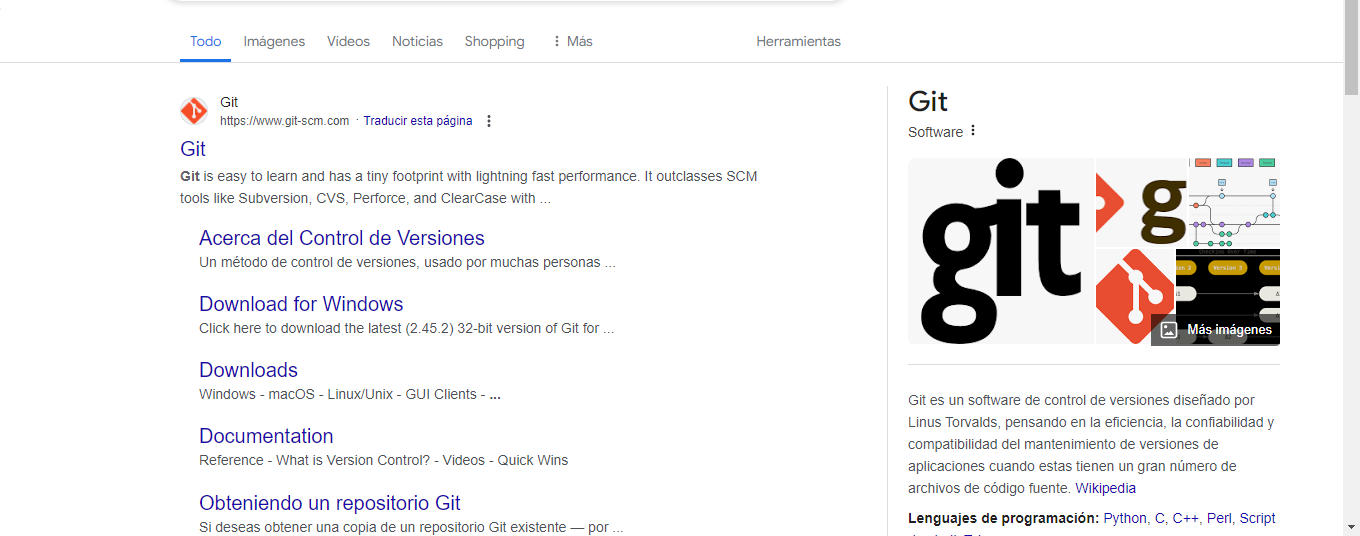
Uso de etiquetas (tags) para marcar versiones específicas del código fuente.

Instalación y Configuración de Git (Ejemplo para herramienta local)

Paso 1: Descarga de Git

Visita el sitio web oficial de Git: git-scm.com.

Descarga la versión adecuada para tu sistema operativo (Windows, macOS, Linux).

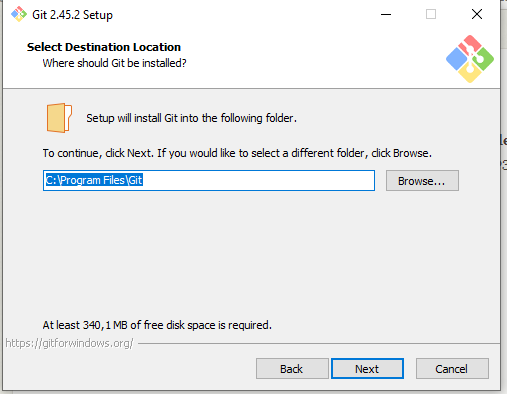


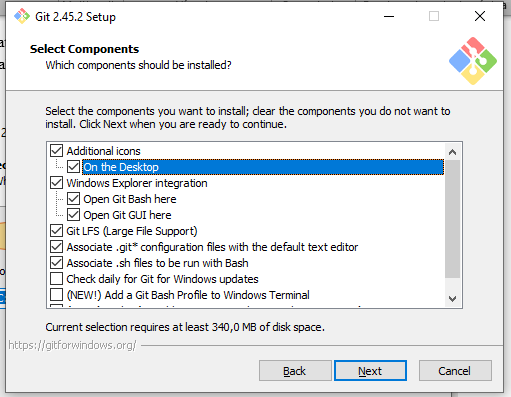


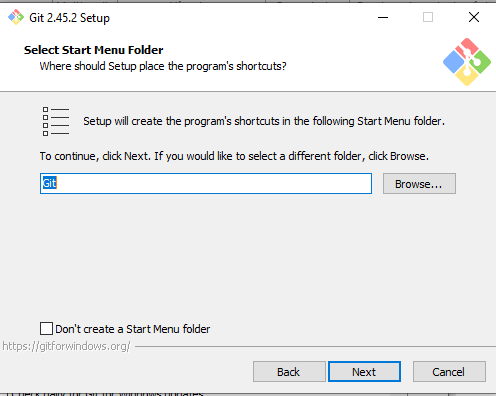
Paso 2: Instalación de Git

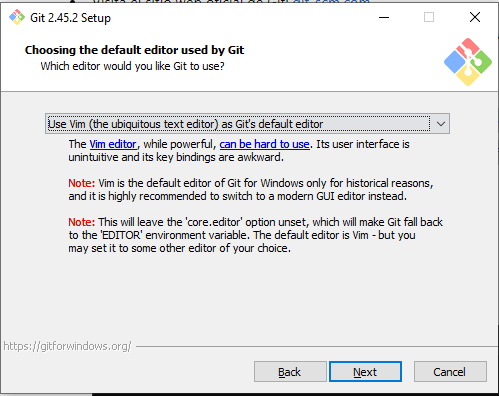
Ejecuta el archivo descargado y sigue las instrucciones del asistente de instalación.

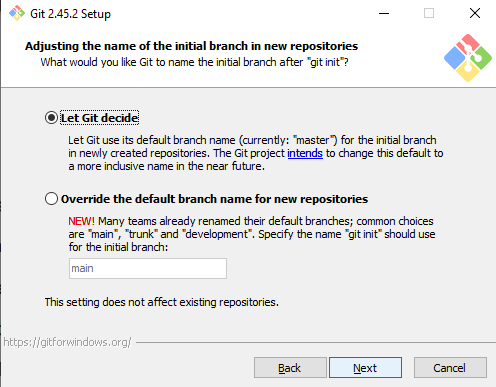
Asegúrate de seleccionar las opciones adecuadas según tus preferencias (componentes adicionales, editor de texto, ajustes de configuración).

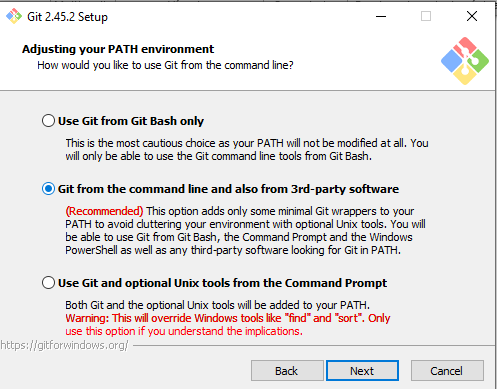


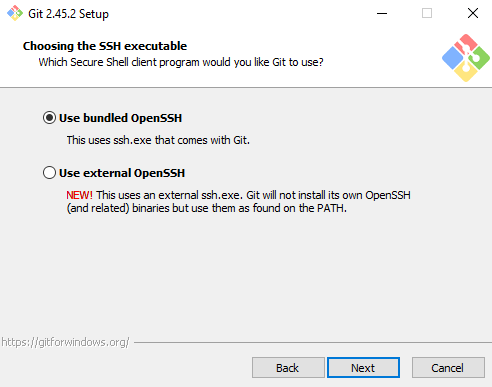


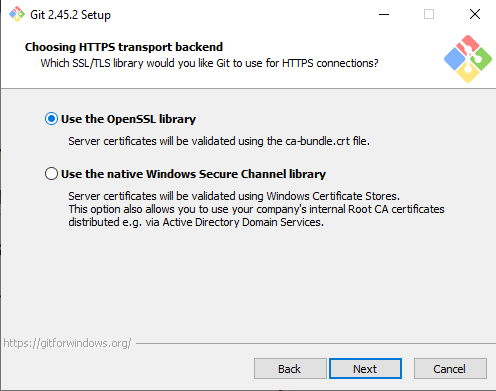


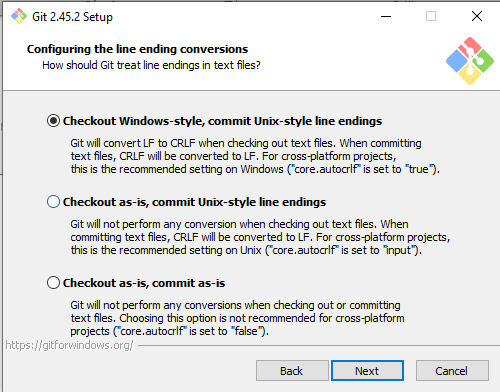


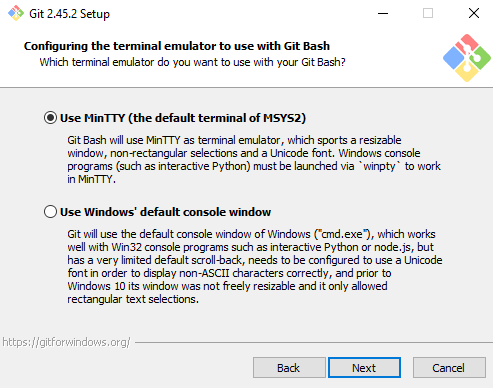


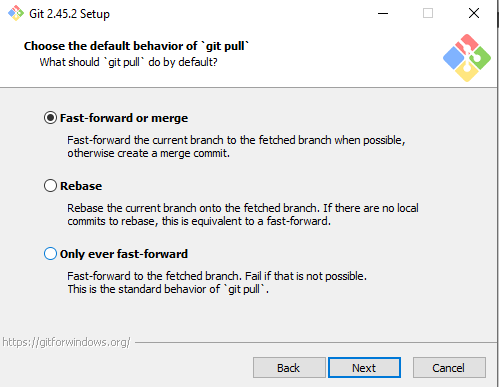


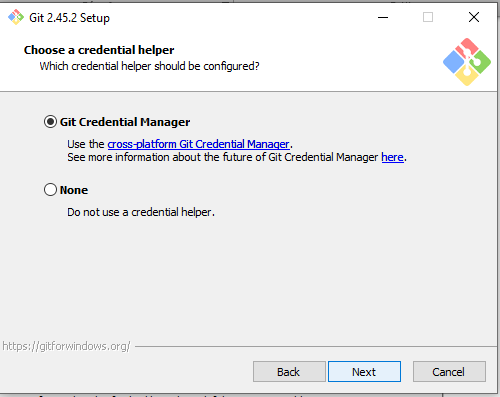


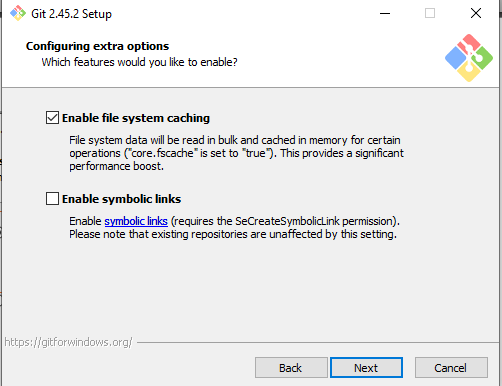


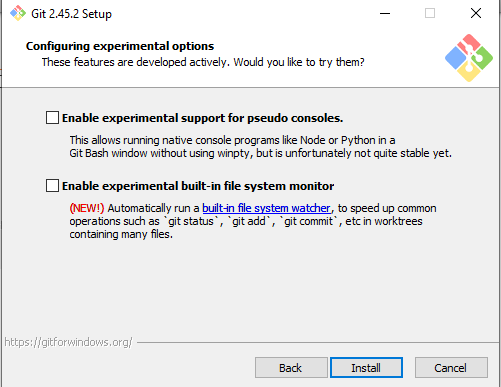


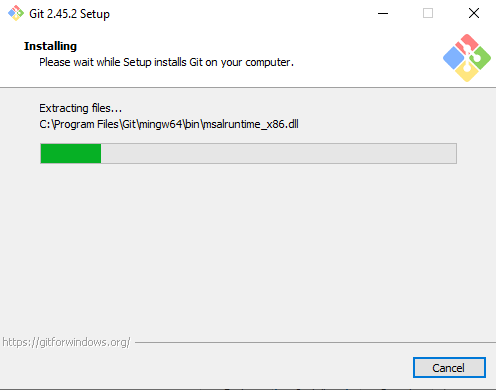












Conclusión

En este documento hemos explorado la instalación, configuración y uso básico de Git y su integración con GitHub, herramientas fundamentales para el control de versiones en el desarrollo de software. A través de este proceso, hemos alcanzado los siguientes puntos clave:

Instalación y Configuración Inicial: Hemos aprendido cómo instalar Git en diferentes sistemas operativos y configurar correctamente las opciones globales como el nombre de usuario y la dirección de correo electrónico. Esta configuración inicial es crucial para empezar a utilizar Git de manera efectiva.

Uso Básico de Git: Hemos comprendido cómo inicializar un repositorio local, realizar confirmaciones de cambios (commits) y gestionar ramas (branches) para el desarrollo paralelo de funcionalidades. Estas habilidades son fundamentales para mantener un historial claro y organizado del código fuente.

Integración con GitHub: Exploramos cómo crear un repositorio remoto en GitHub y vincularlo con nuestro repositorio local. Además, configuramos la autenticación mediante claves SSH para asegurar la comunicación segura entre nuestro entorno local y GitHub.

Gestión Avanzada de Versiones: Aunque nos hemos centrado en conceptos y prácticas básicas, hemos identificado características avanzadas como el uso de ramas y etiquetas para la gestión flexible y estructurada del desarrollo de software a lo largo del tiempo.

Beneficios y Consideraciones: Reconocemos los beneficios de utilizar Git y GitHub para la colaboración eficiente en equipos de desarrollo distribuidos, así como las consideraciones como la curva de aprendizaje inicial y la necesidad de práctica constante para dominar estas herramientas.

En resumen, Git y GitHub ofrecen una base robusta y escalable para la gestión de versiones en proyectos de desarrollo de software, facilitando la colaboración, el seguimiento de cambios y la integración continua. Continuar explorando y practicando con estas herramientas es esencial para mejorar nuestras habilidades y eficiencia en el desarrollo de software moderno.