GA7-220501096-AA1-EV04 INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTA DE VERSIONAMIENTO (LOCAL / WEB)

PRESENTADO POR

UBEIMAR BEDOYA ARANGO

PROFESOR

CRISTIAN ARIAS

SENA

REGIONAL CALDAS

ANÁLISIS Y DESARROLLO DE SOFTWARE 2721447

MEDELLÍN

2024

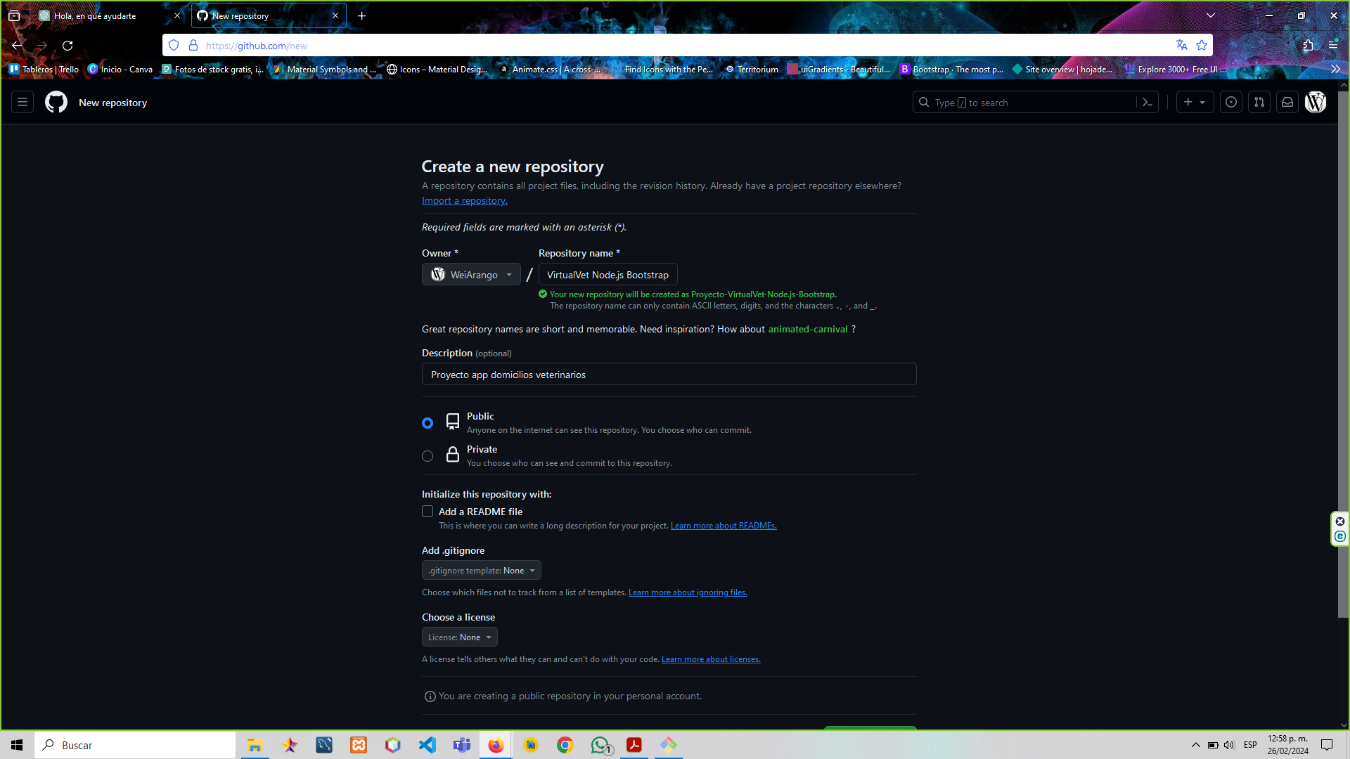
INTRODUCCIÓN

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Fue creado por Linus Torvalds en 2005 y ha ganado popularidad debido a su eficiencia y flexibilidad en el manejo de proyectos de software. El propósito principal de Git es rastrear los cambios en el código fuente durante el desarrollo de un proyecto, permitiendo a los desarrolladores trabajar de manera colaborativa y mantener un historial completo de todas las modificaciones realizadas.

**Algunos conceptos clave en Git incluyen:**

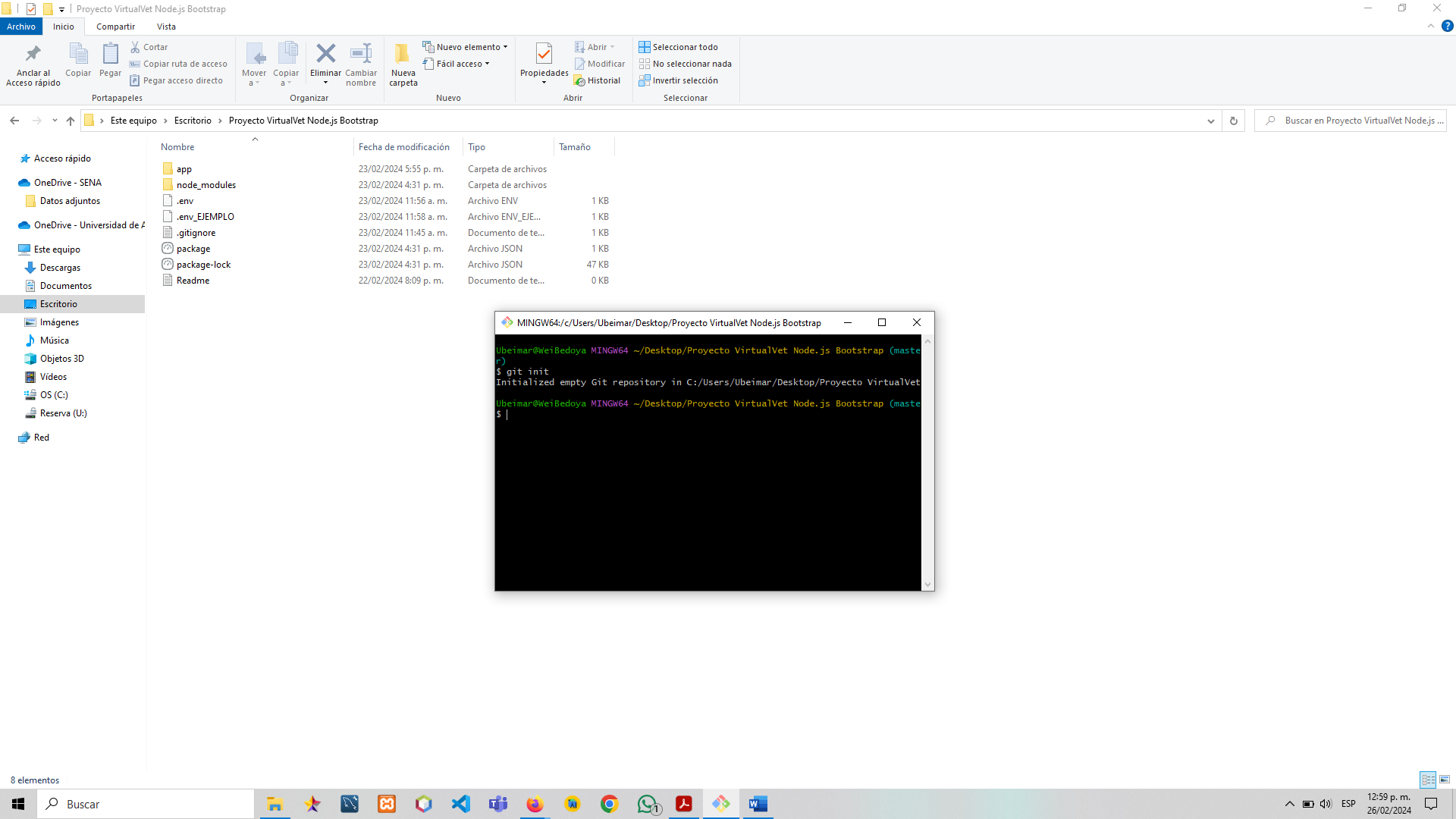
* **Repositorio**: Es un lugar donde se almacenan los archivos y carpetas de un proyecto, junto con la información de seguimiento de cambios.
* **Commit**: Un commit (confirmación) representa un conjunto de cambios realizados en los archivos del proyecto en un momento específico. Cada commit tiene un mensaje descriptivo que proporciona información sobre los cambios realizados.
* **Branch (Rama)**: Git permite la creación de ramas independientes que permiten a los desarrolladores trabajar en características o correcciones de errores sin afectar directamente la rama principal del proyecto. Posteriormente, estas ramas pueden fusionarse de nuevo en la rama principal.
* **Clone (Clonar)**: Clonar un repositorio Git significa copiar todo el historial de versiones y archivos de un repositorio remoto a una máquina local.
* **Push (Empujar) y Pull (Tirar)**: Push se refiere a enviar los cambios locales a un repositorio remoto, mientras que pull implica obtener los cambios del repositorio remoto y fusionarlos con la versión local.
* **Tag (Etiqueta)**: Git permite etiquetar puntos específicos en la historia del proyecto, como versiones estables o hitos importantes.

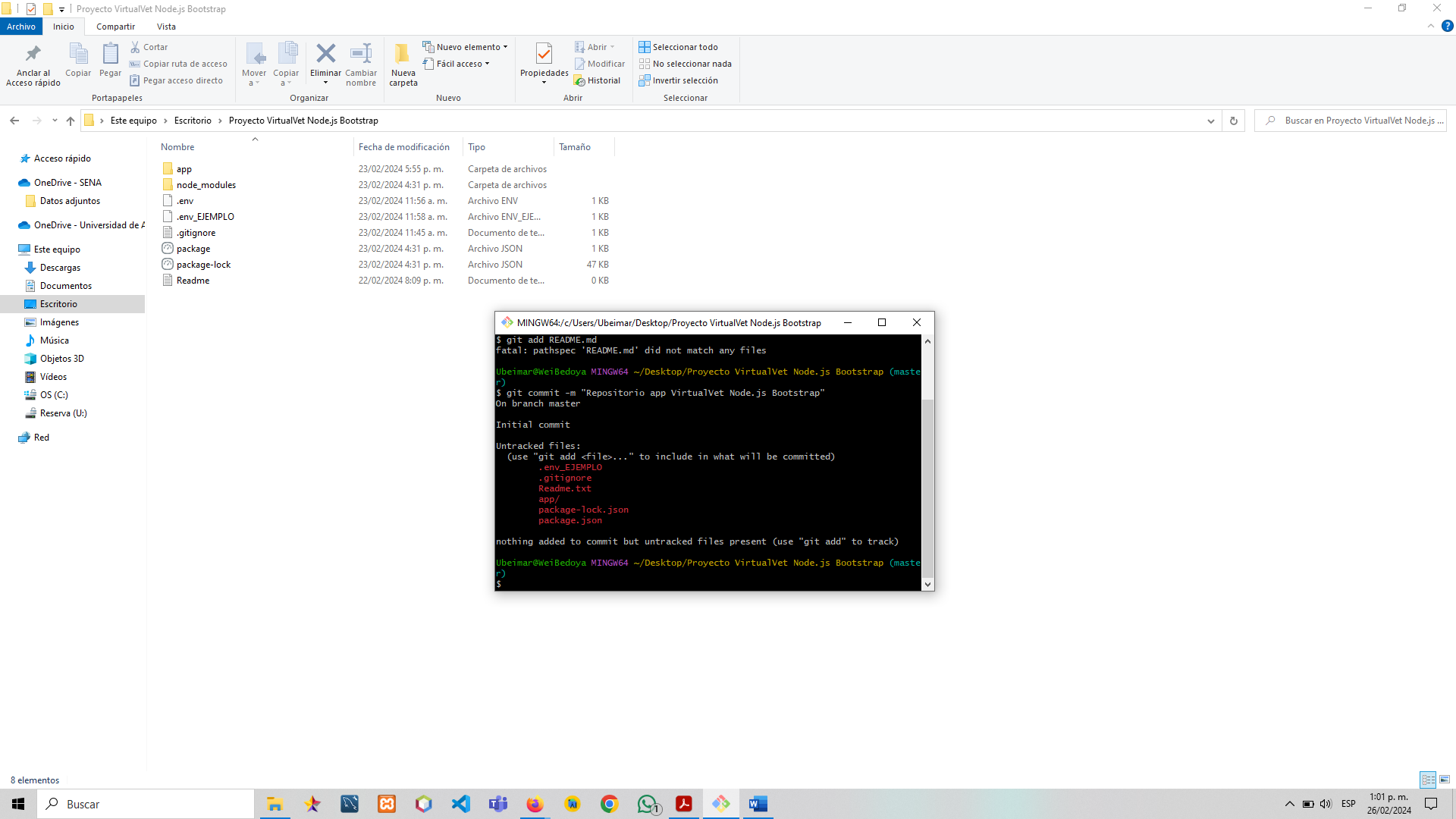
**Objetivos de git**

1. **Rastreo de Cambios Eficiente:** Git tiene como objetivo proporcionar un sistema eficiente para rastrear los cambios en el código fuente a lo largo del tiempo. Utiliza un modelo de datos basado en instantáneas (snapshots) en lugar de un modelo basado en diferencias, lo que permite una gestión de versiones rápida y eficiente.
2. **Descentralización:** Git es un sistema de control de versiones distribuido, lo que significa que cada desarrollador tiene una copia completa del repositorio, incluido el historial completo. Esto facilita el trabajo en entornos descentralizados y permite a los desarrolladores trabajar sin conexión a internet.
3. **Colaboración Efectiva:** Git facilita la colaboración entre desarrolladores al permitir ramificaciones (branches) independientes y fusiones (merges) sencillas. Cada desarrollador puede trabajar en su propia rama sin interferir con el trabajo de otros, y luego fusionar los cambios de manera controlada.
4. **Integridad de Datos:** Git utiliza hash criptográficos para identificar de manera única cada cambio en el repositorio. Esto asegura la integridad de los datos y permite detectar cualquier alteración no autorizada.
5. **Velocidad y Rendimiento:** Git se diseñó para ser rápido y eficiente en operaciones clave como la creación de ramas, confirmaciones (commits) y fusiones (merges). La estructura interna de Git y su enfoque en instantáneas contribuyen a su rendimiento.
6. **Flexibilidad:** Git es muy flexible y se puede adaptar a una variedad de flujos de trabajo de desarrollo. Permite a los equipos adoptar prácticas que se ajusten a sus necesidades específicas.
7. **Compatibilidad y Portabilidad:** Git es compatible con múltiples plataformas y sistemas operativos. Además, es posible utilizar Git en conjunto con servicios de alojamiento en la nube como GitHub, GitLab o Bitbucket.
8. **Historial Completo:** Git mantiene un historial completo de cada cambio realizado en el repositorio, lo que permite a los desarrolladores retroceder en el tiempo y revisar cualquier versión anterior del código.

Captura de pantalla de computadora

Descripción generada automáticamente





Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

