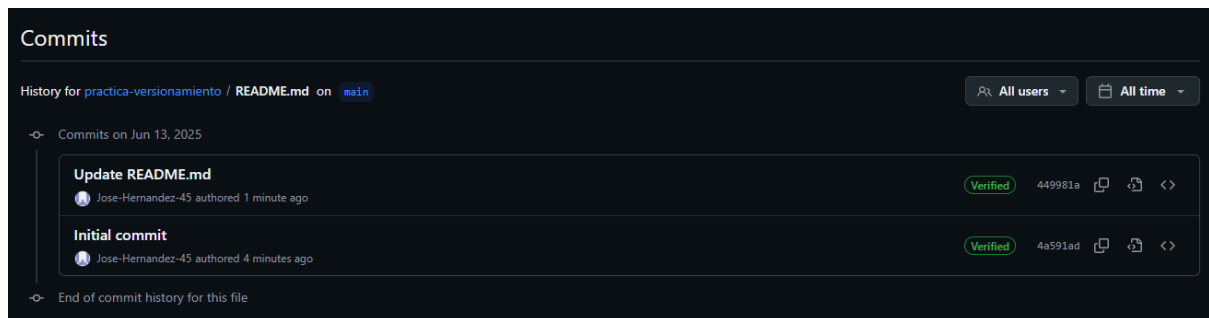
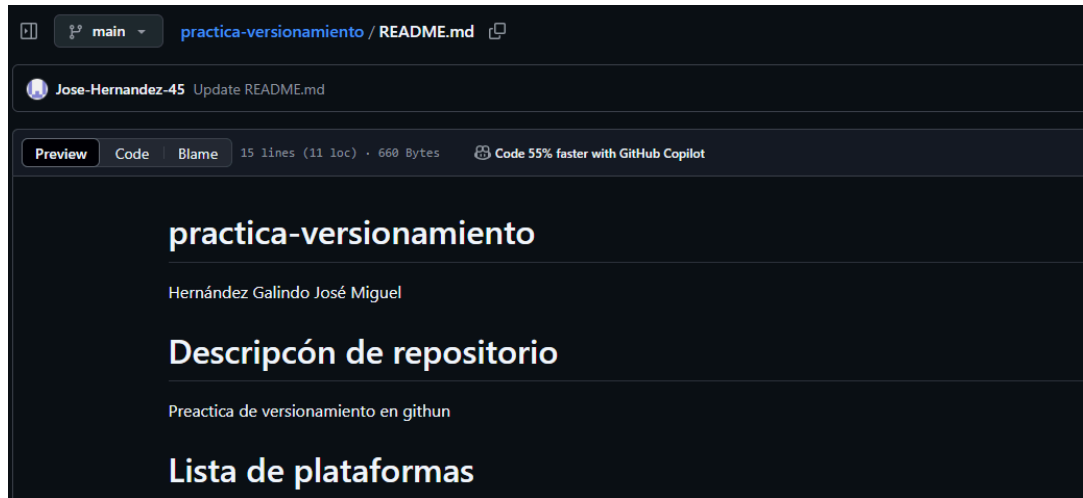


Identificación y uso de plataformas de versionamiento de software

Plataforma	¿Qué es?	¿Es pública, privada o ambas?	Características clave	Uso común en proyectos
GitHub	Plataforma de alojamiento de código basada en Git.	Ambas. Repositorios públicos y privados.	<ul style="list-style-type: none">- Interfaz muy popular y amigable- Gran comunidad- GitHub Actions- Integración con VS Code y Copilot	abierto <ul style="list-style-type: none">- Desarrollo colaborativo- Integraciones con IA y automatizaciones
GitLab	Plataforma completa que permite desde la planificación hasta la entrega.	Ambas. Repositorios públicos y privados.	<ul style="list-style-type: none">- CI/CD integrado por defecto- Flujo DevOps completo- Auto-hosting disponible- Buen control de permisos	<ul style="list-style-type: none">- Gestión completa de proyectos empresariales- CI/CD desde el inicio del repositorio
Bitbucket	Plataforma de gestión de código fuente con integración de Jira y CI/CD.	Ambas. Repositorios públicos y privados.	<ul style="list-style-type: none">- Integración nativa con herramientas Atlassian- Soporte para pipelines- Permite Git y Mercurial	<ul style="list-style-type: none">- Desarrollo empresarial con Jira- Equipos que ya usan el ecosistema Atlassian

Repositorio creado e historial de commits

Qué tipo de cuenta tienes y porque en GitHub, GitLab o Bitbucket (elige al menos una). Uso GitHub por la facilidad de crear repositorios y por el control que se tiene con Visual Studio, también porque es fácil de manejar en los versionamientos.



url: <https://github.com/Jose-Hernandez-45/practica-versionamiento>

Preguntas y reflexión

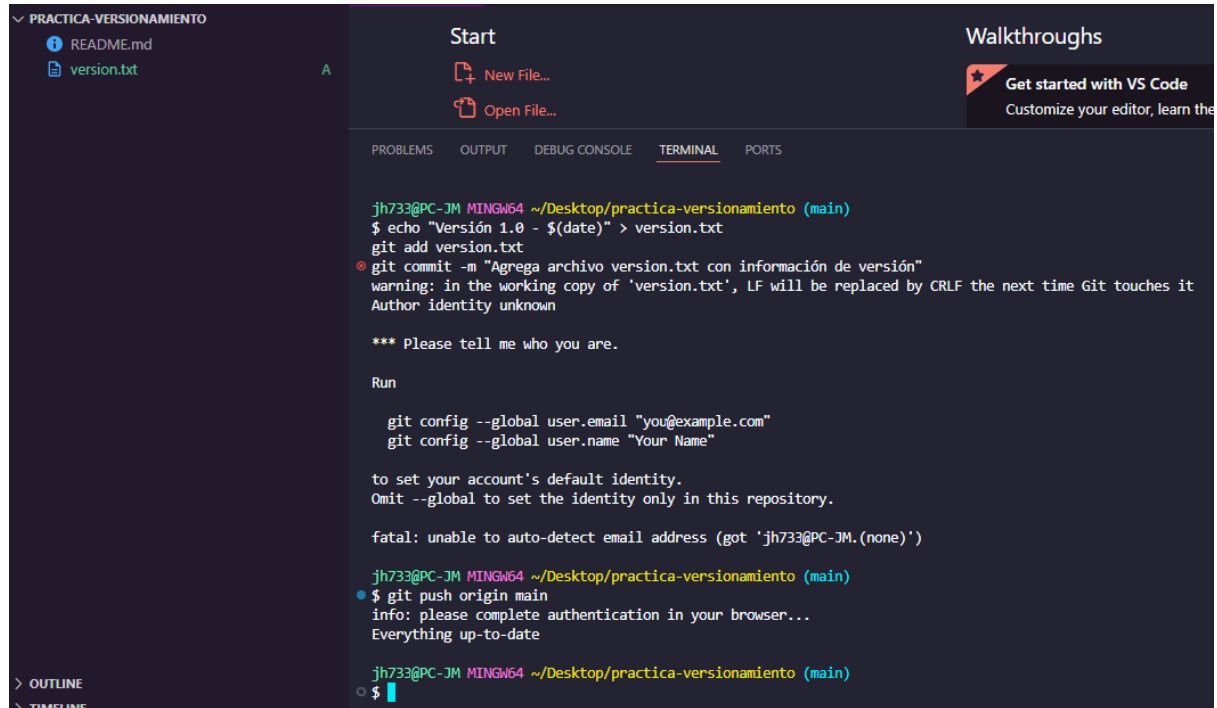
Comparte tu experiencia con la plataforma usada. Use GitHub y si la experiencia es agradable, es fácil de usar y sirve para proyectos pequeños o medianos.

Comenta ventajas o dificultades encontradas. Una de las ventajas es que nos sirve para todo tipo de proyecto, lenguaje o estructura y una de las desventajas es su falta de intuitividad, ya que se puede confundir con otras opciones dentro de git.

¿Cuál plataforma te pareció más intuitiva y por qué? ¿Qué recomendarías para un proyecto colaborativo escolar o profesional? Me pareció parcialmente intuitiva github ya que las demás son más para un ámbito más técnico o más profesional, y github se presta para un proyecto o producto que se realiza.

Integración con editor de código

1. Abre uno de los repositorios en Visual Studio Code.
2. Usa la terminal integrada para hacer un nuevo commit (agrega un archivo version.txt).



The screenshot shows the Visual Studio Code interface with a project named 'PRACTICA-VERSIONAMIENTO'. The Explorer sidebar on the left shows 'README.md' and 'version.txt'. The main editor area is empty. The integrated terminal at the bottom shows the following commands and output:

```
jh733@PC-JM MINGW64 ~/Desktop/practica-versionamiento (main)
$ echo "Versión 1.0 - $(date)" > version.txt
git add version.txt
⊗ git commit -m "Agrega archivo version.txt con información de versión"
warning: in the working copy of 'version.txt', LF will be replaced by CRLF the next time Git touches it
Author identity unknown

*** Please tell me who you are.

Run

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'jh733@PC-JM.(none)')

jh733@PC-JM MINGW64 ~/Desktop/practica-versionamiento (main)
● $ git push origin main
info: please complete authentication in your browser...
Everything up-to-date

jh733@PC-JM MINGW64 ~/Desktop/practica-versionamiento (main)
○ $
```