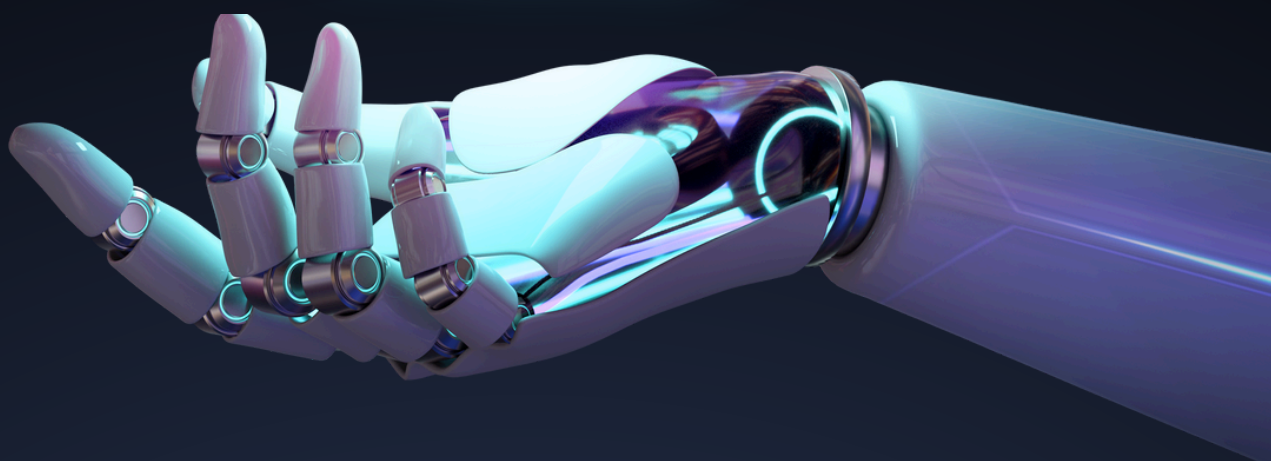


GIT Y GIT HUB



GIT

- Ventajas:
- Seguimiento de historial de cambios.
- Trabajo en paralelo sin interferencias.
- Fusión y resolución de conflictos de manera eficiente.
- Componentes clave:
- Repositorio (Repo): Donde se almacenan los archivos del proyecto y su historial.
- Commit: Registro de cambios realizados en el repositorio.
- Branch (Rama): Línea paralela de desarrollo que permite trabajar en nuevas funcionalidades sin afectar la rama principal.

GIT HUB

- Qué es: Plataforma de alojamiento de repositorios Git basada en la web.
- Función principal: Facilita la colaboración en proyectos a través de herramientas adicionales como issues, pull requests y wikis.
- Ventajas:
- Interfaz gráfica para gestionar repositorios.
- Herramientas de colaboración y revisión de código.
- Integración con servicios CI/CD.
- Almacenamiento en la nube de repositorios.

COMANDOS GIT

- Configuración Inicial:
 - `git config --global user.name "Tu Nombre"`
 - `git config --global user.email "tuemail@ejemplo.com"`
- Iniciar un Repositorio:
 - `git init`: Inicializa un nuevo repositorio Git.
- Clonar un Repositorio:
 - `git clone <URL del repositorio>`: Clona un repositorio existente.
- Estados y Seguimiento:
 - `git status`: Muestra el estado de los cambios en el repositorio.
 - `git add <archivo>`: Agrega un archivo al área de preparación (staging area).

COMANDOS GIT HUB

1. Autenticación:
 - `gh auth login`: Inicia sesión en GitHub desde la línea de comandos.
2. Gestión de Repositorios:
 - `gh repo create`: Crea un nuevo repositorio en GitHub.
 - `gh repo clone <nombre-del-repositorio>`: Clona un repositorio de GitHub.
3. Issues y Pull Requests:
 - `gh issue list`: Lista todas las issues abiertas en un repositorio.
 - `gh issue create`: Crea una nueva issue.
 - `gh pr create`: Crea un nuevo pull request.
 - `gh pr list`: Lista todos los pull requests abiertos.