


NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jan Carlos Peguero	4 - 1	Micro	25-01-24

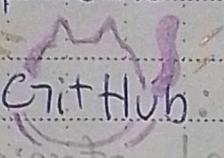
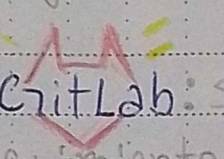
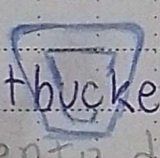
Title: **Git**

Keyword	Topic: Conceptos introductorios
Gestión Administra- ción Control	<p>¿Que es? es un sistema de control de versiones distribuido, diseñado para la eficiencia y la confiabilidad.</p> <p> Permite a los usuarios realizar un seguimiento de los cambios en su código fuente, archivos de configuración y otros documentos a lo largo del tiempo.</p>
Questions	<p>Uso: Git se usa principalmente para el desarrollo de software, pero también se puede utilizar para cualquier tipo de proyecto que implique el seguimiento de cambios en archivos. Ejemplos de uso:</p> <p>¿Por qué su uso masivo? ¿Por qué es tan asequible?</p> <ul style="list-style-type: none"> → Control de versiones de código fuente. → Gestión de configuraciones → Documentación

Summary: Git es un gestor de proyectos. Es más usado para desarrollo de software, pero, por su versatilidad, se puede extrapolar a otros ambientes.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jan Carlos Peguero	21 - 2	Micro	25-01-24

Title: Git

Keyword	Topic: Apps y Comandos
- cooperación - seguimiento	Aplicaciones asociadas a Git:
	 <p>→ GitHub: es una plataforma de alojamiento de código que utiliza Git para la gestión de Versiones.</p>
	 <p>→ GitLab: similar a Hub y sirve para el seguimiento de problemas, wikis e integración continua.</p>
Questions	 <p>→ Bitbucket: es una Plataforma de alojamiento de código propiedad de Atlassian. Ofrece integración con otras herramientas de la misma.</p>
¿Por que no se une todo en 1?	
	Comandos importantes de Git
	<p>→ git init: inicializa un nuevo repositorio de Git en el directorio actual.</p>

Summary: Git usa aplicaciones para el seguimiento del proyecto en curso. Estas aplicaciones nos ayudan a monitorear más detalladamente nuestros archivos.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jan Carlos Peguero	4-3	Micro	25-01-24

Title: Git

Keyword	Topic: <u>Comandos</u>
<ul style="list-style-type: none"> - Comandos - cmd 	<ul style="list-style-type: none"> +git clone: clona un repositorio Git existente en el directorio actual. +git branch: crea una rama en el repositorio. +git add: Agrega archivos al área de preparación para su confirmación. +git commit: confirma los cambios en el área de preparación en el repositorio local. +git push: envía los cambios confirmados al repositorio remoto. +git pull: Descarga los cambios del repositorio remoto y los fusiona con el repositorio local. +git checkout: cambia a una rama diferente en el repositorio local. +git merge: Combina dos ramas en el repositorio local. +git log: muestra un historial de los cambios en el repositorio. +git status: muestra el estado actual del repositorio.
<p>Questions</p> <p>¿Por qué no hay un ide para el manejo de los comandos?</p>	

Summary: Git se puede controlar desde caja de comandos y con los comandos mostrados anteriormente.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Jan Carlos Peguero	4-4	Micro	25-01-24

Title: Git

Keyword	Topic: Ejemplos de uso de comandos
- Crear proyectos.	Crear un repositorio: (terminal de comandos)
- Directorios	mkdir mi_proyecto (directorio)
- Repositorios	cd mi_proyecto
	git init
	Crear una ramificación (en rama base)
	git branch nueva-rama
	Unir una rama al repositorio base
	git checkout nueva-rama (Cambiar a la rama base)
	git merge main
	Eliminar un repositorio (local)
	rm -rf mi_proyecto
	Eliminar una rama (no estar en local)
	git checkout main
	git branch -d nueva-rama
Questions	Nota:
¿Por que es importante hacer commit antes de eliminar o hacer modificaciones?	antes de hacer algún cambio, se recomienda hacer un commit
- Para confirmar los arreglos antes de eliminar cualquier cosa.	

Summary: Estos son algunos ejemplos de comandos de uso de los comandos para modificar aspectos del proyecto que estemos trabajando.