

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	1/7	Electiva-C1-25	

Title: Comandos de Git "Clone"

Keyword	Topic: <u>CLONE</u>
Notes:	<p>El comando "clone" es realmente una herramienta de utilidad para fijar como objetivo un Repositorio Existente con el fin de crear una copia local.</p> <p>Llevando un analisis a esta configuración y los casos donde la tendremos de uso común serian los siguientes: clonar un Repositorio localmente o Remoto, clonar un Repositorio vacío, utilizar operaciones super-fuertes para clonar Repositorios Parcialmente.</p> <p>- Si tu disco de tu servidor se corrompe puedes usar cualquier de los "clone" en cualquier de los clientes para devolver el servidor al estado en el que estaba cuando fue clonado. Puedes clonar un Repositorio con "[git clone URL]". Por Ejemplo: de Entre muchas otras comandos para utilizar la función</p>
Questions	

Summary: Ya vimos que es y para que sirve, en una breve oración podemos decir que el comando "git clone" copia el Repositorio y descarga todo el historial del Proyecto y crea una copia local del mismo donde se desea clonar: Esto fue todo"

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	2/7	Electiva-cy-25	

Title: Commands Git "Push"

Keyword	Topic: Push
Notes:	<p>Este es uno de los comandos más utilizados en git, este se usa para comunicar con otro repositorio, calcular lo que tu base de datos local tiene que lo remoto no tiene y luego subir (push git) la diferencia al otro repositorio, se requiere Acceso de Escritura al otro repositorio y por tanto normalmente se autoriza de alguna otra manera. Si podemos ver el comando, Envía tus Remotos Aquí podemos ver los funciones de subir una Rama a un Repositorio, y Publicar un push más en la subida de Ramas Específicas y en hacer seguimientos a estos. con el Push también podemos empujar una subida con un Refspec completo en lugar de los Atajos Generales que se utilizan normalmente ya vimos que son buena a la hora de hacer Push en git.</p>
Questions	

Notes: Este es uno de los comandos más utilizados en git, este se usa para conectar con otro repositorio, calcular lo que tu base de datos local tiene que la remota no tiene y luego subir (push git) la diferencia al otro repositorio. Se requiere Acceso de Escritura al otro repositorio y por tanto normalmente se autentica de alguna otra manera. Si podemos ver el comando `Envia tus Remotos` Aquí podemos ver los `funcionales` por subir una `Prueba` a un `Repositorio` y `Publicar` un `pro` más en la `subida` de `Branch` Específicos y en `hacer` `seguimientos` a estos. con el `Push` también podemos `eliminar` `violencia` a la `subida` con un `Refspec` completo en lugar de los `Atributos` Generales que se utilizan normalmente ya vimos que `Envia` a la `Herencia` `Push` en `git`.

Summary: Como podemos comprender el comando "Push" permite subir los commits desde tu rama (local) en tu repositorio remoto.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Ariel Medina	3/7	Electiva-c1-25	

Title:

Comandos de Git "Pull"

Keyword

Topic:

"Pull"

Notes:

Este es básicamente una combinación de los comandos [Git-fetch y git Merge], donde el Git descarga desde el Repositorio Remoto Especificado a continuación, de forma inmediata internamente combinarlo en la Rama en la que te encuentres. Este rápidamente trae y combina Remotos, mostrando la forma de ver lo que se fusionará si se Ejecuta en Inspeccionar Remoto también vemos como usarlos para Ayudar con dificultades de Rebase en Reorganizar una Reorganización.

Questions

Trabajar con Git Pull y Git Push es muy importante cuando varias personas Están colaborando en un proyecto al mismo tiempo y necesitan tener Acceso constante al último Estado "Git Pull" Actualiza Rápidamente!!

Summary:

Bien como vimos este comando permite que tu y los miembros de tu Equipo trabajen en un mismo Proyecto simultáneamente!! Eso fue todo!!

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	4/7	Electiva-C1-25	

Title: Comandos de Git - 'Commit'

Keyword	Topic: <u>"Commit"</u>
	Notes: Este comando guarda todos los cambios hechos en la zona de trabajo o área de preparación (staging area) junto con una breve descripción del uso, en un commit a Repositorio local.
Questions	<ul style="list-style-type: none"> - Los commit son una parte del corazón del uso, pues puedo pensar en un commit como una captura a tu proyecto, donde se crea una nueva versión de ese proyecto en el Repositorio actual: - Puedes recordar hacer cambios a los que les hizo un "commit" en una Fecha Posterior, o revertir el Proyecto a esa versión guardada. - si varios "commit" editan diferentes Parts del Proyecto, no se sobreEscriben Entre si aunque los autores de los commit NO se conozcan Entre si.

Summary: Bien!! El commit, Es una captura de pantalla, o del Proceso hablando de códigos instantanea de los cambios preparados en ese momento del proyecto.

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	5/7	Electiva - 12-25	

Title: Comandos de Git "Branch"

Keyword	Topic: "Branch"
Notes:	<p>Primero debemos de saber esto, Git almacena una Rama como Referencia a un commit, En este sentido, una Rama Representa la punta de una serie de commit, es es un contenedor de los commit. El comando Branch Git, Permite Enumerar, Crear, Cambiar el nombre y eliminar Branch, NO permite cambiar el Branch ni fusionar los historiales de diferentes Branch, y por esto razón, Git Branch Este estrechamente Integrado con los comandos Git checkout y Git Merge.</p> <p>Sin argumentos, el comando Git Branch Enumera los Branch Locales del Repositorio. La Rama actualmente esta en uso este marcado como estable.</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>* Git Branch * Main</p> </div> <p>* Especificando el nombre de un Branch como Ejemplo: una opción.</p>
Questions	

Summary: Bien!! Entendimos que el comando SE usa para Gestionar ramas de un Repositorio y Este permite : Crear, Listar, Eliminar Ramas Esto es todo!!

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	6/7	Electiva - CI - 25	

Title:

Comandos de Git "Fetch"

Keyword

Topic:

"Fetch"

Notes:

Git Fetch es descargar el commit Remos, obtenidos desde el Repositorio Remoto al Repositorio Local. Los nuevos contenidos recuperados a través de git Fetch deberían aplicarse explícitamente a la propia working copy, lo que hace que la ejecución de este comando sea una operación segura para recuperar nuevos commit, sin tener que aplicarlos necesariamente a su trabajo en curso.

Questions

Bien entonces el comando fetch no sobrescribe el código existente en la Rama Actual. El comando obtiene un registro de los cambios en el Repositorio Remoto, lo que permite conocer el historial de ramas antes de realizar el cambio.

Summary:

Bien. Masla Ahora comprendimos que el comando descarga el proceso o los cambios del Repositorio sin fusionarlos automáticamente en la Rama Actual!!
Esto es todo!!

NAME	PAGES	SPEAKER/CLASS	DATE - TIME
Axel Medina	7/7	Electiva - CI-25	

Title: Comando de Git "tag"

Keyword	Topic: "LAG"
	Notes: "Etiquetado" Las Etiquetas son Referencias que apuntan a punto concretos en el historial de git. Generalmente, el Etiquetado se usa para capturar un punto en el historial que se utiliza para publicacion de una version marcada.
Questions	una etiqueta es como una rama que no cambia. a diferencia de las ramas, las etiquetas, tras crearse, no tienen más historial de confirmación, siquieras más. Información sobre las ramas ya es diferente.
	Las etiquetas anotadas se almacenan como objetos en la Base de datos de git. Para que quede claro, almacenan meta datos como (nombre, correo electrónico y Fecha).

Summary: Pues bien, el tag crea, Administra los Etiquetas de git, que marcan las Versiones Especificas en el Historial del Repositorio.