Introducción **a GIT**

Presentado por Alexander Díaz

www.consultec-ti.com

ACADEMIA CONSULTEC

Sistema de **control de versiones**

Es una herramienta fundamental para el desarrollo de cualquier proyecto.

Permite mantener un historial de cambios en los archivos, evitar la pérdida de datos, facilitar la colaboración en equipo y mejorar la gestión de cambios en el proyecto.

¿Qué es GIT?

Git es un sistema de control de versiones distribuido que se utiliza principalmente para el desarrollo de software.

Permite controlar las diferentes versiones de un proyecto, trabajar en equipo de manera más eficiente y es muy flexible y personalizable.

ACADEMIA CONSULTEC

Archivos de texto y binarios

Ambos son manejados de la misma manera.

Creando una copia completa de cada archivo en cada commit, pero los archivos binarios son más grandes y consumen más espacio en disco que los archivos de texto.

ACADEMIA CONSULTEC

Día 1

Comandos básicos
en GIT

Crear un repositorio y un commit

Escribir el comando **git init** y presionar Enter. Esto inicializará un nuevo repositorio Git en el directorio actual.

Se pueden agregar los archivos al repositorio con el comando **git add <archivo>** para agregar todos los archivos del directorio actual.

Una vez que se han agregado los archivos al repositorio, se puede crear un nuevo commit con el comando **git commit -m "mensaje del commit".**

ACADEMIA CONSULTEC

GIT DIFF

Escribir el comando **git diff <archivo>** y presionar Enter. Esto mostrará las diferencias entre el archivo actual y su última versión guardada.

Para comparar una versión específica del archivo con la versión actual, se debe escribir **git diff <hash> <archivo>,** donde <hash> es el número de identificación del commit que contiene la versión del archivo que se desea comparar.

ACADEMIA CONSULTEC

¿Qué es el **Staging** y los **Branch**?

Escribir el comando **git add <archivo>** para agregar los cambios de un archivo específico al Staging.

Escribir el comando **git add .** para agregar todos los cambios en el directorio actual al Staging.

Escribir el comando **git status** para ver los cambios que se han agregado al Staging.

ACADEMIA CONSULTEC

¿Qué es el **Staging** y los **Branch**?

Escribir el comando **git branch <nombre de la rama>** para crear una nueva Branch.

Escribir el comando **git checkout <nombre de la rama>** para cambiar a la nueva Branch.

Realizar los cambios necesarios en la nueva Branch utilizando el Staging y los commits.

Escribir el comando **git merge <nombre de la rama>** para fusionar la nueva Branch con la rama principal del repositorio.

ACADEMIA CONSULTEC

¿Qué es un **Merge**?

Cambiar a la rama de destino: primero, debe cambiarse a la rama de destino en la que se desean fusionar los cambios utilizando el comando git checkout <rama de destino>.

Ejecutar el comando de Merge: después, debe ejecutarse el comando git merge <rama a fusionar> para combinar los cambios de la rama que se desea fusionar con la rama de destino.

ACADEMIA CONSULTEC

¿Cómo volver en el tiempo?

Checkout a una versión anterior: puedes cambiar al estado de un commit anterior utilizando el comando **git checkout <ID del commit>** en la consola de Git. Esto te permitirá trabajar en esa versión anterior de tu proyecto.

Crear una nueva rama desde un commit anterior: puedes crear una nueva rama de tu proyecto a partir de un commit anterior utilizando el comando **git branch <nombre de la nueva rama> <ID del commit>** en la consola de Git. Esto creará una nueva rama que comienza desde el commit especificado.

ACADEMIA CONSULTEC

¿Cómo revertir cambios?

Si deseas revertir los cambios realizados en un commit específico, puedes utilizar el comando **git revert <ID del commit>** en la consola de Git. Esto creará un nuevo commit que deshace los cambios realizados en el commit anterior.

El comando **git reset** te permite deshacer cambios sin crear un nuevo commit. Esto es útil si deseas deshacer varios cambios en un solo comando.

ACADEMIA CONSULTEC

Receso

15 min

Laboratorio

1. Crear un un ambiente local un repositorio git con los comandos aprendidos en clase.

2. Crear un Branch con en nombre de lab/01

3. Crear un archivo README.md y escribir un resumen de los comandos aprendidos.

ACADEMIA CONSULTEC

Flujos de trabajo en control de versiones

Repositorios remotos

En Git, un repositorio remoto es una versión de tu proyecto que se encuentra alojada en un servidor remoto, en lugar de en tu computadora local.

Para trabajar con repositorios remotos en Git, debes vincular tu repositorio local con el repositorio remote, utilizando el comando **git remote add ‹nombre del repositorio› ‹URL del repositorio›.**

ACADEMIA CONSULTEC

Petición de cambios

Es una forma de solicitar que un colaborador revise y apruebe tus cambios, antes de fusionarlos con la rama principal del proyecto.

Es una práctica común en proyectos colaborativos, puede ayudar a mantener la calidad del código y evitar errores.

Merge entre **ramas y resolución de conflictos**

El merge se puede hacer de forma automática o manual.

Si Git detecta que los cambios en las dos ramas no interfieren entre sí, el merge se realizará automáticamente.

Si hay conflictos, Git pedirá al usuario que resuelva los conflictos manualmente antes de completar el merge.

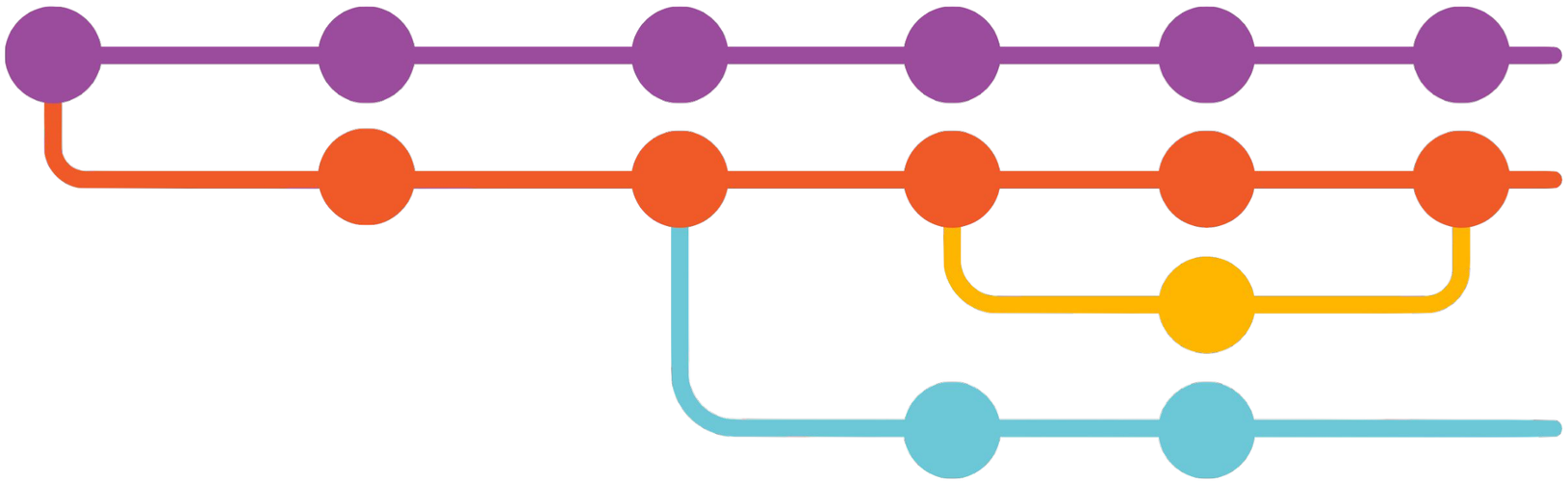
ACADEMIA CONSULTEC

Almuerzo

45 min

Gestión de ramas

Es la posibilidad de gestionar en una rama una línea independiente de desarrollo que se deriva de otra rama, normalmente la rama principal (también llamada "master" o "main").



Laboratorio

1. Crear un un ambiente remoto un repositorio git.
2. Agregar al repositorio del laboratorio anterior el enlace del repositorio remoto que creamos.
3. Sincronizar nuestros repositorios.
4. Crear un pull request de nuestro Branch lab/01 a nuestro main Branch.
5. Crear un Branch lab/02 y colocar un resumen de los nuevos comandos aprendidos.
6. Repetir el paso nro 4.

ACADEMIA CONSULTEC

Receso

15 min

Forks

Forks en **control de versiones**

Los forks son especialmente útiles para proyectos de código abierto, ya que permiten a los desarrolladores contribuir al proyecto sin necesidad de tener permisos de escritura en el repositorio original.

Además, los forks permiten a los desarrolladores experimentar con el código sin temor a dañar el repositorio original.

ACADEMIA CONSULTEC

Evaluación **del día**

Día **2**

Versionar Commits
en GIT

¿Qué son los **tags**?

Los "tags" o "etiquetas" en Git son identificadores asociados a versiones específicas de un repositorio. Se utilizan para marcar versiones importantes, como lanzamientos de software, y para marcar hitos importantes en el desarrollo de un proyecto.

```
git tag v1.0
git tag -s v1.0
git push –tags
git show v1.0
```

ACADEMIA CONSULTEC

GIT **fetch, pull y push**

git fetch

Se utiliza para descargar los cambios de un repositorio remoto a tu repositorio local.

git pull

Combina automáticamente los cambios del repositorio remoto con la rama actual en tu repositorio local.

git push

Se utiliza para enviar los cambios en la rama actual de tu repositorio local al repositorio remoto.

ACADEMIA CONSULTEC

Laboratorio

1

Unificar todos nuestros cambios en el main Branch.

2

Crear un TAG para cada commit que represente la versión que resuelve los ejercicios de cada laboratorio.

ACADEMIA CONSULTEC

Flujos de **trabajo avanzados**

Archivo **.gitignore**

Esto es especialmente útil cuando tienes archivos que son generados automáticamente por tu aplicación o sistema de construcción, como archivos de registro o archivos de caché.

Agregar estos archivos al control de versiones no es útil y puede llevar a conflictos innecesarios al fusionar ramas.

ACADEMIA CONSULTEC

Archivo **README.md**

README.md es un archivo que se utiliza para proporcionar información básica y contexto sobre un proyecto en un repositorio Git.

El archivo se muestra en la página principal del repositorio y es la primera cosa que alguien verá al visitar el repositorio.

ACADEMIA CONSULTEC

Receso

15 min

Convenciones para **comentar commit**

Comenzar con una oración imperativa que resuma el cambio realizado.	Limitar la longitud del mensaje del commit para que sea fácil.	Agregar detalles adicionales si es necesario.	Incluir un número de tarea en el mensaje del commit.	Usar verbos precisos, como ”fix", ”add", ”delete" o ”feat".

ACADEMIA CONSULTEC

Convenciones para el **nombramiento de branchs**

Utilizar nombres cortos y descriptivos.	Utilizar nombres en minúsculas separados por guiones (-) o barras diagonales (/).	Utilizar prefijos para indicar el tipo de branch.	Evitar nombres genéricos como dev, feature1, nuevo, etc.

ACADEMIA CONSULTEC

Cómo documentar una **solicitud de cambio**

Título

Debe ser breve y descriptivo del cambio que se propone.

Descripción

Proporcione una descripción detallada de lo que se espera que resuelva y los motivos detrás del cambio.

Capturas de pantalla

Proporcione capturas para ilustrar los cambios propuestos.

Problemas relacionados

Proporcione un enlace al problema.

Etiquetas

Asegúrese de etiquetar adecuadamente la solicitud de cambio.

ACADEMIA CONSULTEC

¿Qué es el **CODE REVIEW**?

Es una parte importante del proceso de desarrollo de software, ya que ayuda a mejorar la calidad del código, reduce el número de errores y facilita la colaboración entre los miembros del equipo.

Además, permite compartir conocimientos y experiencias entre los programadores, lo que a su vez mejora la productividad y la eficiencia del equipo de desarrollo.

ACADEMIA CONSULTEC

Laboratorio

1. El instructor creará un repositorio compartido.

2. Cada participante deberá crear un fork de este repositorio.

3. El instructor deberá carga una plantilla de un archivo README.md con el titulo de todo el contenido impartido en clase actualmente.

4. Cada participante deberá tomar uno o más temas para colocar su resumen del contenido y crear un pullrequest al repositorio principal.

ACADEMIA CONSULTEC

GIT en múltiples entornos de trabajo

GIT **clean y rebase**

GIT clean

Es un comando que se utiliza para eliminar archivos no rastreados en un directorio de trabajo local.

GIT rebase

Se utiliza para integrar cambios de una rama a otra de una manera más ordenada y lineal.

ACADEMIA CONSULTEC

GIT **stash**

Guarda temporalmente los cambios en un estado intermedio para que puedas trabajar en otra cosa sin tener que hacer commit de los cambios actuales.

Luego, puedes volver al estado guardado y continuar trabajando desde donde lo dejaste. También es posible aplicar el estado guardado a otra rama o a otra parte del proyecto.

ACADEMIA CONSULTEC

GIT **cherry-pick**

Permite copiar un commit específico de una rama y aplicarlo en otra, lo que puede ser útil para corregir errores en una rama de producción utilizando los cambios ya realizados en una rama de desarrollo.

ACADEMIA CONSULTEC

Almuerzo

45 min

Comandos de **emergencia en GIT**

GIT **reset y reflog**

reset

Es un comando que permite mover la rama actual y la cabeza del repositorio a un commit anterior.

reflog

Muestra un historial detallado de todos los movimientos de la cabeza del repositorio, incluidos los movimientos realizados mediante reset o rebase.

ACADEMIA CONSULTEC

Uso avanzado del **commit (amend)**

El comando **git commit --amend** se utiliza para modificar el commit anterior en caso de que se haya olvidado algún archivo o mensaje de confirmación, o para fusionar múltiples confirmaciones en una sola.

Esto permite mantener la historia del proyecto más limpia y fácil de seguir.

ACADEMIA CONSULTEC

Uso avanzado del **checkout**

Al usar git checkout -- <file>, podemos descartar los cambios en un archivo y volver a la última versión guardada.

También podemos usar git checkout <branch> -- <file> para restaurar un archivo de una rama específica en nuestra rama actual.

GIT **blame**

Muestra quién y cuándo realizó cambios en cada línea de un archivo determinado.

Se puede rastrear la autoría de cambios específicos en un archivo.

También puede ser útil para dar crédito a los colaboradores y mantener un registro de la historia de un archivo.

GIT **grep** y **log**

git grep

Es un comando que se utiliza para buscar cadenas de texto dentro de los archivos de un repositorio.

git log

Es un comando que se utiliza para mostrar el historial de confirmaciones en un repositorio.

ACADEMIA CONSULTEC

Repositorios dentro **de otros repositorios**

Submódulos

Los submódulos en Git permiten incluir un repositorio de Git dentro de otro repositorio de Git, como una subcarpeta.

Es una forma de mantener y utilizar proyectos relacionados o dependientes.

Receso

15 min

Laboratorios

Cada participante deberá tomar completar todos los temas que le fueron asignados.

Todos los participantes deberán hacer code review del archivo compartido.

Una vez se aprueben los pull request los participantes deberán ir resolviendo conflictos hasta que todos los pull request sean cerrados.

ACADEMIA CONSULTEC

Evaluación **del día**

Contacto

adiaz@consultec-ti.com
+ 507 6408 1387
Panamá

info@consultec-ti.com

@consulteclatam

@consultec-ti

consultec-ti.com

GRACIAS

¡Nos vemos pronto!