



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

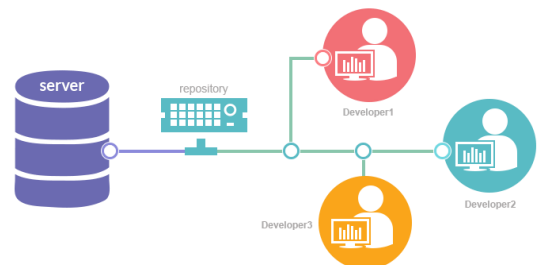
- **Denominación del Programa de Formación:** Análisis y Desarrollo de Sistemas de Información.
- **Código del Programa de Formación:** 228106
- **Nombre del Proyecto:** Sistematización De Procesos Administrativos Para Organizaciones ~
- **Fase del Proyecto:** Desarrollo
- **Actividad de Proyecto:** Codificar Los Módulos Del Sistema De Información.
- **Competencias:**
 - 220501007 - CONSTRUIR EL SISTEMA QUE CUMPLA CON LOS REQUISITOS DE LA SOLUCIÓN INFORMÁTICA.
- **Resultados de Aprendizaje Alcanzar:**
 - Construir la interfaz de usuario, apoyado en la evaluación del prototipo, determinando las entradas y salidas requeridas en el diseño y definiéndolos lineamientos para la navegación, de acuerdo con las necesidades del usuario.
 - Construir la base de datos, a partir del modelo de datos determinado en el diseño del sistema, utilizando sistemas de gestión de base de datos, según los protocolos establecidos en la organización.

Duración de la Guía: 40 Horas (Trabajo Directo 30 Horas – Trabajo Independiente 10 Horas)

2. PRESENTACIÓN

El sistema de control de versiones (SCV) es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que se pueda recuperar versiones específicas más adelante. Cualquier tipo de archivo que se encuentre en un ordenador puede ponerse bajo control de versiones.

Si es desarrollador, y quiere mantener cada versión de los archivos de un proyecto, un sistema de control de versiones es una elección obligatoria, Ya que permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo, ver quién modificó por última vez algo que puede estar causando un problema, quién introdujo un error y cuándo, y mucho más. Usar un VCS también significa generalmente que, si se dañan o se pierden archivos, se recuperar fácilmente.



En este caso utilizaremos GIT. La principal diferencia entre Git y cualquier otro VCS es cómo Git modela sus datos. Conceptualmente, la mayoría de los demás sistemas almacenan la información como una lista de cambios en los archivos. Estos sistemas (CVS, Subversion, Perforce, Bazaar, etc.) modelan la información que almacenan como un conjunto de archivos y las modificaciones hechas sobre cada uno de ellos a lo largo del tiempo.

GFPI-F-135 V01

En cambio, Git modela sus datos más como un conjunto de instantáneas de un mini sistema de archivos. Cada vez que



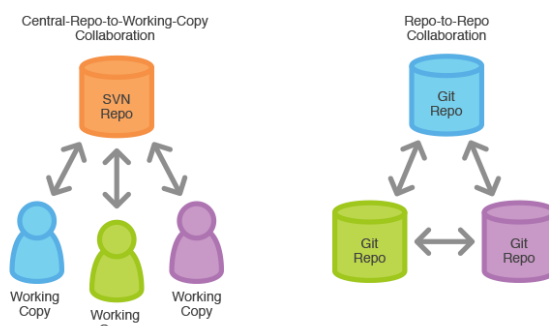
confirmas un cambio, o guardas el estado de tu proyecto en Git, él básicamente hace una foto del aspecto de todos tus archivos en ese momento, y guarda una referencia a esa instantánea. Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo, sólo un enlace al archivo anterior idéntico que ya tiene almacenado.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

Señor aprendiz esta actividad ha sido planteada con el propósito de ayudarlo a identificar los presaberes, de esta actividad de aprendizaje, la cual aplicará mediante el siguiente cuestionario que contestará en el foro correspondiente en la plataforma Blackboard denominado “**SCV y GIT**”, en este debe expresar sus ideas y debatirse con sus compañeros de acuerdo a sus conocimientos previos (Las participaciones en los foros **no** se deben de enviar como **archivos adjuntos**) y deben debatirse con al menos dos compañeros.

- Le ha ocurrido en algún momento de su vida que le toca modificar un documento en Excel, ¿Word, PowerPoint, Etc. con dos o más compañeros y al momento de unir los cambios se ha perdido información?
- ¿Para usted cuál cree que es la mejor manera de que dos o más personas trabajen en un mismo archivo?
- ¿Ha usado Dropbox o Google Drive, como funciona la opción de compartir archivos?
- ¿Cuándo le ha tocado trabajar un proyecto de programación en grupo como ha hecho para unir los cambios de todos los integrantes?
- ¿Para usted que se puede deducir de la imagen del lado derecho?



Modo Desarrollo: Individual de Forma Presencial y Desescolarizada.

Ambiente: Aula Taller – Blackboard.

Materiales: Plataforma Blackboard, Conexión a Internet.

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.

“Los individuos marcan goles, pero los equipos ganan partidos, Toda empresa de desarrollo trabaja en equipo, ya que esto permite llevar a cabalidad desde pequeños proyectos hasta grandes desarrollos de forma eficaz y satisfactoria, pero no todo es color de rosa en un equipo de desarrollo, en el camino nos encontraremos con Proyectos difíciles de gestionar y liderar, Riesgos a sobreescribir con mi código el avance formal del equipo y la centralización y poca probabilidad de trabajar remotamente. Para solucionar estos problemas es que nacen los sistemas de control de versiones, un orden en medio del caos”

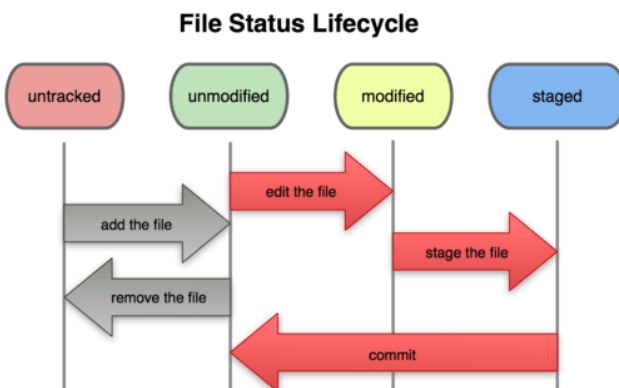
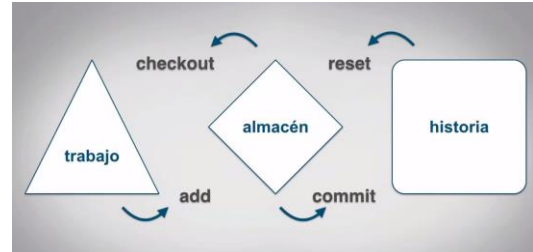
- 3.2.1** Realice una investigación y responda a las siguientes preguntas, generando una conclusión sobre cada uno de ellos con sus propias palabras:

- ✓ ¿Qué es un Sistema de Control de Versiones?
- ✓ ¿En sus palabras porque es necesario el control de versiones?
- ✓ ¿Sistema de Control de Versiones Locales, Centralizadas y Distribuidas?
- ✓ ¿Qué es GIT y su relación con Linus Torvals, Por qué y cómo nació?
- ✓ ¿Por qué usar Git, Características del GIT Frente a otros SCV?.



3.2.2 Ver los video tutoriales del curso (“Git”) dados por el instructor y realizar los ejemplos y ejercicios realizados en los videos.

3.2.3 Según las siguientes imágenes explique de forma de detallada y completa cuál es el esquema de trabajo de GIT, explicando cada uno de los espacios de trabajo, los comandos para pasar de un espacio a otro tanto de adelante como de hacia atrás (revocando cambios).



3.2.4 Leer y comprender el material dispuesto en el material del curso en la carpeta denominada fase de análisis - “Git_manual.pdf” donde encontrará un resumen general del trabajo con GIT.

Modo Desarrollo: Individual de Forma Presencial y Desescolarizada.

Ambiente: Aula Taller – Blackboard.

Materiales: Plataforma Blackboard, Conexión a Internet.

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

3.3.1 Descargue o solicite al instructor los siguientes programas y realice la instalación de estos:

- [Git For Windows](https://goo.gl/fQox63) (<https://goo.gl/fQox63> => Tutorial).
- [GitHub for Windows](#).
- [GitKraken](#)
- [Sublime Text 3](#).
- Beyond Compare => (Solicitar al instructor).



3.3.2 Abra una cuenta en los siguientes servicios de hosting de repositorios GIT y realice una tabla de comparación entre ellos destacando sus ventajas y desventajas.

- <https://gitlab.com/>
- <https://github.com/>
- <https://education.github.com/>

Modo Desarrollo: Individual de Forma Presencial y Desescolarizada.

Ambiente: Aula Taller – Blackboard.

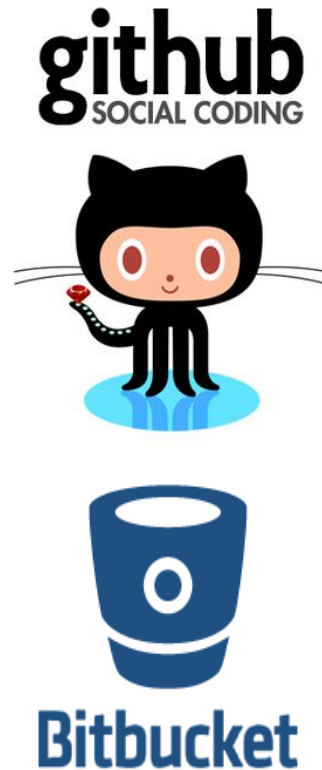
Materiales: Plataforma Blackboard, Conexión a Internet.



3.4 Actividades de transferencia del conocimiento.

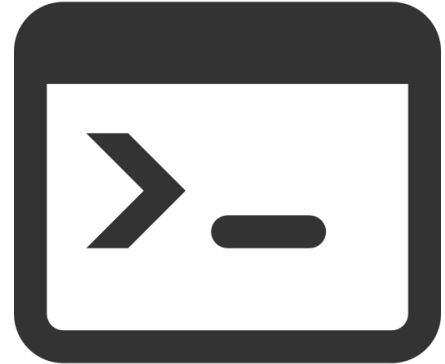
3.4.1 Por grupos de trabajo de 3 personas según distribución del instructor exponer 3 comandos de GIT por consola, Describiendo para que sirve, sus opciones (parámetros) y ejemplos del parámetro con las principales opciones:

- git help
- git config
- git init
- git clone
- git add
- git commit
- git rm
- git mv
- git status
- git diff
- git log
- git show
- git tag
- git reset
- git clean
- git blame
- git checkout
- git branch
- git stash
- git merge
- git pull
- git push
- git fetch
- git remote
- git remote add
- git gui
- git clone --bare / git init --bare



3.4.2 Según los grupos anteriores realizar un video o manual realizando los siguientes pasos en la consola de git:

- ✓ Configure Git con su nombre completo y correo misena, como editor de git establecer sublime text, como herramienta de merge Beyon Compare.
- ✓ Ejecute el comando git config --list.
- ✓ Crear un repositorio vacío.
- ✓ Añadir 2 Archivos de Texto uno con sus nombres y otro con su identificación.
- ✓ Configure el git ignore para ocultar los archivos .pdf y coloque uno dentro del repositorio
- ✓ Confirmar sus cambios (commit)
- ✓ Mostrar el estado del repositorio
- ✓ Mostrar un listado de las confirmaciones realizadas en el repositorio.
- ✓ Modificar los archivos y mostrar que cambios se realizaron.
- ✓ Agregue un archivo añádalo, elimínalo.
- ✓ Agregue un archivo añádalo, confírmelo y elimínalo.



- 3.4.3** Prepárese para presentar una prueba por la plataforma blackboard de la temática vista:

Ambiente: Aula Taller – Blackboard.

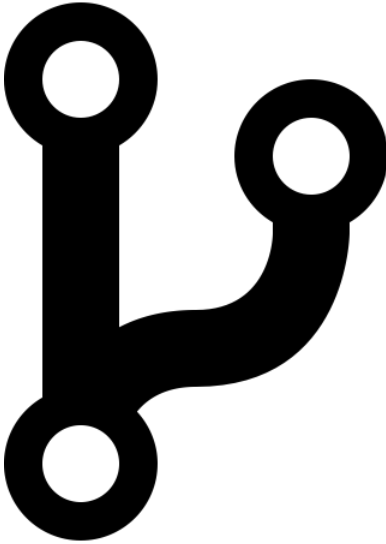
4. ACTIVIDADES DE EVALUACIÓN

GFPI-F-135 V01



Evaluación de Conocimientos (3.4.3)		
--	--	--

5. GLOSARIO DE TÉRMINOS



Stage (Git): Before making a Commit a series of amendments, I have to put the modified files Stage. It is as an intermediate stage.

Stash (Git): Pincipio mistook the Stage ... sometimes, some of the actions that allows me to git require that there are no changes in the working directory. If I do not want to lose these changes can make a stash modified files. Once the action that required the working directory unchanged completed, I can remove files Stash and return them to the working directory.

Fork (Github): This is one of the words most often heard when talking to Github. There are many pages of developers in which we can see that ask us to let us forking in one of his projects ("Fork me in Github") ... but what do really Fork. When I press the button Fork in a draft Github what I am doing is to create a repository from my personal repository that clones the source repository on which I made the Fork.

Rebase (Git): Relining is very similar to Merge command. In fact serves the same. The difference is that Exceeding tries to keep the story of a Branch linearly, while Merge generates ramifications in this story

Patch (Git): Sometimes we can not do directly push on a remote repository because we can not have sufficient privileges. To solve this you can create a Parch. A Patch is a set of changes, ie, a COMMIT (or several), which puedenadjuntar easily and send for example in an email. Subsequently, the receiver of this Patch, you can apply the same about the repository and thus make a Merge on it.



Pull Request (Github): At some point we may want to contribute to a particular project that we are interested in Github. To do this we have to create a Pull Request to inform the user of the project owner our intention. It is but a request to make a Pull.

Follow (Github): This Github has the same meaning as in any other social network. I can become a follower of a user Github and so I see and send notifications work being done.



6. REFERENTES BIBLIOGRÁFICOS

- ✓ Standards Icontec y Normas APA: <http://goo.gl/GozfAs>, <https://goo.gl/fvPFkr>
- ✓ SublimeText 3: <http://goo.gl/HBKLfi>
- ✓ Documentación Oficial de GIT: <https://goo.gl/h6mJ53> y <http://git-scm.com/documentation/>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Diego Alonso Ojeda	Instructor	Sistemas	Noviembre 15 del 2019.

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)	Diego Alonso Ojeda, Lili Ana Sandoval.	Instructor	Sistemas	Febrero 06 del 2021	Revisión de contenidos
	Diego Alonso Ojeda	Instructor	Sistemas	Noviembre 09 del 2021	Actualización version Formato