



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENIZAJE: GUÍA **VERSIONAMIENTO**

- Denominación del Programa de Formación: **TECNÓLOGO EN ANÁLISIS DESARROLLO DE SOFTWARE.**
- Código del Programa de Formación: **228106**
- Nombre del Proyecto: **SISTEMA INTEGRAL WEB PARA GESTION DE PROCESOS EDUCATIVOS DEL CEET** · Fase del Proyecto: **Análisis.**
- Actividad de Proyecto: **Sistema Integral Web Para Gestión De Procesos CSF.** · Competencia: **220501006 - Especificar los requisitos necesarios para desarrollar el sistema de información de acuerdo con las necesidades del cliente.**
- Resultados de Aprendizaje Alcanzar: **Diseñar la estructura tecnológica del sistema integral** · Duración de la Guía: **14 HORAS**

A. PRESENTACIÓN

La construcción de aplicaciones empresariales requiere con frecuencia del trabajo de equipos de desarrolladores que trabajan simultáneamente. Lo anterior requiere una coordinación precisa, de lo contrario puede suceder que cada desarrollador tenga una versión distinta del proyecto y se presenten problemas al momento de realizar el empalme.

B. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

2.1 - Material de apoyo y conceptos: Para iniciar con las actividades de

la presente guía realizaremos un recorrido por el material de apoyo:

OA_versionamiento el cual proporcionara los conocimientos necesarios

para conocer e iniciar el versionamiento de nuestros proyectos o futuros desarrollos de software. **TIEMPO – 2 HORAS (Grupal).**



2.2 – Investigación: Para poder continuar con la actividad propuesta para esta guía es necesario realizar la siguiente investigación. **TIEMPO – 2 HORAS (Individual).**

1. ¿Qué es un repositorio y mencion 2 ejemplos de los mismos (proveedores online)?

Es un espacio de almacenamiento donde se guarda y gestiona el código fuente de un proyecto, junto con su historial de cambios. Es utilizado para organizar y colaborar en proyectos de desarrollo. Ejemplos de proveedores online:

- GitHub

- GitLab

2. ¿Cuáles son los principales componentes de un versionamiento en la herramienta **GIT**?

- **Commit:** Es un registro de cambios en el código, con un mensaje descriptivo.
- **Branch (rama):** Una línea independiente de desarrollo dentro del proyecto.
- **Staging Area:** Una zona intermedia donde se preparan los archivos antes de confirmar los cambios.
- **Repositorio (local y remoto):** El local se puede encontrar en la pc del usuario, y el remoto es el que está en la nube o en un servidor.
- **Merge:** Proceso de combinar cambios de diferentes ramas.

3. ¿Mencione con sus propias palabras las ventajas que tiene **GIT** frente a otros proveedores de repositorios?

- **Distribuido:** Git permite tener una copia completa del repositorio en cada máquina, sin depender del servidor central.
- **Rendimiento:** Es rápido al realizar operaciones como commits, diffs y merges.
- **Control de ramas:** Permite trabajar con ramas de manera eficiente, lo que facilita el trabajo colaborativo y el manejo de diferentes versiones del proyecto.
- **Historial detallado:** Mantiene un historial completo de cambios que permite revertir errores o analizar el progreso del proyecto.
- **Flexibilidad:** Se integra fácilmente con diversos proveedores de repositorios como GitHub o GitLab.

4. Mencione por lo menos 5 ejemplos de los comandos básicos que se usan en GIT. -Algunos comandos básicos de GIT son:

- **git init:** Inicializa un nuevo repositorio en el directorio actual.
- **git clone <URL>:** Clona un repositorio remoto en tu máquina local.
- **git add <archivo>:** Añade archivos al área de preparación (staging area).
- **git commit -m "mensaje":** Guarda los cambios en el repositorio con un mensaje descriptivo.
- **git push:** Envía los cambios locales al repositorio remoto.

5. ¿Que son y cuáles son las funciones de los **branch**?

Un branch (rama) es una línea de desarrollo separada dentro de un proyecto; Permite trabajar en una característica, corrección de errores o experimentación sin afectar el código principal. Las funciones que permite :

- Facilitar el trabajo colaborativo al permitir que varios desarrolladores trabajen en paralelo.
- Ayudar a mantener el código principal estable mientras se desarrollan nuevas funcionalidades.
- Facilitar el control de versiones al trabajar con diferentes entornos (por ejemplo, desarrollo, pruebas y producción).

6. ¿Cuál es el Nombre del **branch** principal?

El nombre del branch principal en Git es main. Anteriormente, se utilizaba master, pero muchos proyectos han adoptado main como el nombre predeterminado para el branch principal.



2.3 – Gestor GIT: Como se puede observar en el material de apoyo y con

ayuda de lo mencionado en clase realice la investigación de **1 de gestor**

de GIT de forma local y realice el paso a paso de para su instalación y funcionamiento como lo muestra el ejemplo de **Netbeans**.

TIEMPO – 2 HORAS (Individual).



2.4 – Ejemplo Gestor Instructor: revisar el video tutorial suministrado por el instructor del gestor del repositorio de **GIT** en el cual usted deberá realizar el siguiente ejercicio:

- Crear cuenta en **GitHub**
- Clonar el repositorio de forma local en el equipo de trabajo
- Realizar el **Readme** definiendo el componente metodológico de su proyecto Individual.
- Crear un Branch llamado **(Project)**.
- Crear por lo menos **3 Commits** subiendo los trabajos realizados en su proyecto.



Link video: <https://www.youtube.com/watch?v=jVHEkgiawJo>

Nota: Revisar documentación y ejemplos de:

- ✓ Markdown: <https://guides.github.com/features/mastering-markdown/>
- ✓ Commits: <https://codigofacilito.com/articulos/buenas-practicas-en-commits-de-git>

TIEMPO – 4 HORAS (Individual).

2.5 – Implementación Proyecto: Desarrollar **Repositorio** correspondientes a su **Proyecto Grupal** como se estipula en el ejercicio anterior tomando referencia las indicaciones del instructor. **TIEMPO – 3 HORAS (Grupal).**



2.6 – Cuestionario Kahoot: Concluyendo el desarrollo de la guía, se implementará la actividad en Kahoot que permitirá recopilar conceptos y terminología de la presente guía – **Versionamiento**. **TIEMPO – 1 HORA (Grupal).**

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Evidencias de Desempeño: Evidencias de Producto:	Elaborar Informe donde se identifiquen los procesos, entradas y salida del sistema de información para identificar las necesidades del usuario.	1. Repositorios de proyectos 2. Manual de Gestor GIT 3. Investigación

GFPI-F-135 V02



5. GLOSARIO DE TERMINOS

- **Hash:** firma o sello digital que permite identificar si los contenidos de uno o varios archivos han sido modificados.
- **Linux:** sistema operativo de fuente abierta y gratuito de gran uso a nivel mundial.
- **SHA-1:** algoritmo para la generación de identificadores de tipo Hash.
- **Ssh:** acrónimo de Secure SHell. Software que permite encriptar las sesiones de trabajo.
- **Vcs:** acrónimo de Version Control System. Sistema de control de versiones de software.

6. REFERENTES BIBLIOGRÁFICOS

- Chacon, S., Straub, B. (2014). Pro Git Second Edition. Apress Open

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Javier Leonardo Pineda Uribe	Instructor	Contratista	27 de Febrero de 2023

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

GFPI-F-135 V02