



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

- .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 – **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 mencione 2 ejemplos de los mismos (proveedores online)?

-Un repositorio es un espacio de almacenamiento donde se guarda, organiza y gestiona el código fuente de un proyecto de software, permitiendo el control de versiones y la colaboración entre desarrolladores.



Ejemplos: GitHub y GitLab.

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

-Los principales componentes de versionamiento en Git son:

- **Repositorio:** almacén central donde se guarda el historial del proyecto.
- **Commit:** registro de cambios con un mensaje descriptivo.
- **Branch (rama):** permite trabajar en diferentes líneas de desarrollo.
- **Merge:** combina cambios de una rama a otra.
- **Staging area:** espacio temporal donde se preparan cambios antes de hacer un commit.

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

-Git tiene varias ventajas frente a otros sistemas de control de versiones:

- **Distribución:** Al ser un sistema distribuido, cada desarrollador tiene una copia completa del proyecto, permitiendo trabajar offline y mejorando la seguridad del código.
- **Velocidad y eficiencia:** Git maneja cambios y commits de forma rápida, ideal para proyectos grandes.
- **Ramas y fusiones:** Permite crear ramas fácilmente, trabajar en paralelo y fusionar cambios con menos conflictos, promoviendo una colaboración ágil.
- **Historial detallado:** Git registra cada cambio, permitiendo revertir a versiones anteriores o rastrear modificaciones específicas.
- **Amplia integración:** Funciona bien con plataformas como GitHub, GitLab y Bitbucket, facilitando la colaboración y el despliegue.

1. Mencione por lo menos 5 ejemplos de los comandos básicos que se usan en GIT.

-**git init:** inicia un nuevo repositorio en el directorio actual.

git clone [URL]: clona un repositorio remoto al equipo local.

git add [archivo]: agrega un archivo al área de staging (preparación para commit).

git commit -m "[mensaje]": guarda los cambios en el repositorio con un mensaje descriptivo.

git push: envía los commits locales al repositorio remoto.

2. ¿Qué son y cuáles son las funciones de los **branch**?

-Los *branches* o ramas en Git son versiones paralelas del proyecto donde los desarrolladores pueden trabajar en características o correcciones sin afectar la rama principal.

Funciones principales:

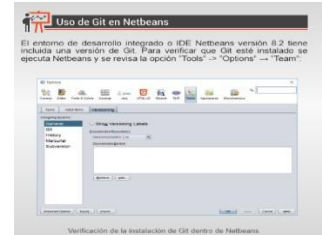
- **Aislamiento:** permite trabajar en cambios específicos sin interferir con el código estable.
- **Colaboración:** facilita el trabajo en equipo al permitir que cada desarrollador tenga su propia rama.
- **Experimentación:** se pueden probar nuevas ideas y descartar ramas sin afectar el proyecto principal.
- **Organización:** ayuda a estructurar el desarrollo (por ejemplo, con ramas de desarrollo, pruebas y producción).



1. ¿Cuál es el nombre del **branch** principal?
- Main

- .2 – **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).



- .3 – **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 metodologico 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).

- .4 – **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).**



- .5 – **Cuestionario Kahoot:** Concluyendo el desarrollo de la guía, se implementara 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.	<ol style="list-style-type: none">1. Repositorios de proyectos2. Manual de Gestor GIT3. Investigación
--	---	---



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)					