# 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 CSF**
* 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**

# 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.

# 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 mención 2 ejemplos de los mismos (proveedores online)?

Archivo online que almacena recursos digitales, como documentos, imágenes, vídeos, etc., para que sean accesibles a través de internet.

* **RePEc (Research Papers in Economics):** Un repositorio temático que concentra la investigación económica.
* **Repositorio Institucional de la Universidad de Cambridge:** Un ejemplo de repositorio institucional que alberga la producción académica de la universidad.
  + 1. ¿Cuáles son los principales componentes de un versionamiento en la herramienta **GIT**?

Los principales componentes de un versionamiento en Git son el directorio de Git (o repositorio), el directorio de trabajo y el área de preparación (staging area). El directorio de Git almacena el historial de versiones del proyecto. El directorio de trabajo es donde se realizan los cambios en los archivos. Y el área de preparación es donde se marcan los cambios que se quieren incluir en la próxima confirmación (commit).

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

Una de las principales ventajas es su naturaleza distribuida, lo que permite a cada desarrollador tener una copia completa del repositorio en su máquina, lo que facilita la colaboración y la autonomía. También destaca por su eficiencia en la gestión de ramas y fusiones, haciendo que los cambios sean fáciles de realizar y revertir.

* + 1. Mencione por lo menos 5 ejemplos de los comandos básicos que se usan en GIT.
* **git init:** Crea un nuevo repositorio local en el directorio actual. Este comando se utiliza para empezar a usar Git en un nuevo proyecto o directorio.
* **git add:** Agrega los cambios de los archivos al área de preparación (staging area). Este comando prepara los archivos para ser incluidos en el próximo commit.
* **git commit:** Guarda los cambios en el historial de versiones del repositorio. Este comando incluye los archivos que se han agregado al área de preparación y crea un punto de control en el repositorio.
* **git push:** Envía los commits locales al repositorio remoto. Este comando permite sincronizar los cambios locales con el repositorio compartido.
* **git pull:** Obtiene los cambios del repositorio remoto y los integra con el repositorio local. Este comando permite sincronizar los cambios remotos con el repositorio local.
  + 1. ¿Que son y cuáles son las funciones de los **branch**?

Son versiones separadas del código de un proyecto. Permiten desarrollar nuevas características, arreglar errores o experimentar con cambios sin afectar la rama principal

* + 1. ¿Cuál es el Nombre del **branch** principal?

El nombre del branch principal en Git puede ser "main" o "master". [Platzi dice](https://platzi.com/clases/1557-git-github/19947-que-es-un-branch-rama-y-como-funciona-un-merge-en/) que, por defecto, los proyectos se crean en una rama llamada "main" (anteriormente "master"). También menciona que GitHub, por defecto, asigna la rama "main" a repositorios nuevos, [GitHub Docs dice](https://docs.github.com.en2es.search.translate.goog/articles/about-branches) y recomienda a los usuarios cambiar el nombre de la rama principal de "master" a "main" para asegurar la compatibilidad futura, [DreamHost dice](https://help.dreamhost.com/hc/es/articles/4466702078740-Configurar-git-para-usar-main-como-rama-principal)

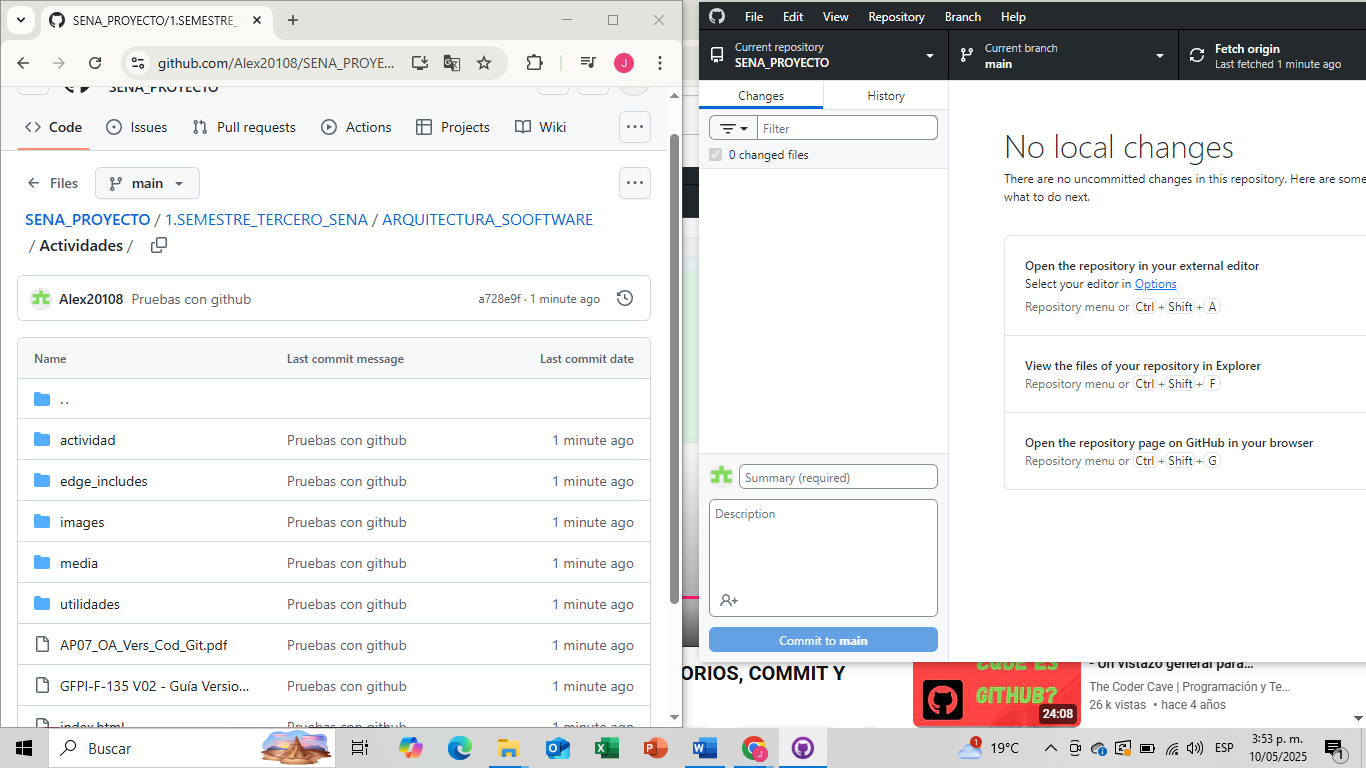
En resumen, el branch principal en Git puede ser "main" o "master", siendo "main" el nombre preferido por GitHub y otros sistemas de control de versiones.

* Blog Biblioteca CID. (2018, 1 abril). *¿Qué es un repositorio? Un ejemplo práctico: consulta y descarga la tesis de Stephen Hawking desde el repositorio de la Universidad de Cambridge*. WordPress. <https://blogbibliotecacid.wordpress.com/2018/04/01/que-es-un-repositorio-un-ejemplo-practico-consulta-y-descarga-la-tesis-de-stephen-hawking-desde-el-repositorio-de-la-universidad-de-cambridge/>
* Chacon, S., & Straub, B. (s.f.). *Fundamentos de Git*. Git. <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Fundamentos-de-Git>
* Kranio. (s.f.). *Descubriendo Git: características y ventajas*. <https://www.kranio.io/blog/descubriendo-git-caracteristicas-y-ventajas>
* Atlassian. (s.f.). *Glosario de Git*. <https://www.atlassian.com/es/git/glossary>
* Platzi. (s.f.). *Introducción a las ramas o branches de Git*. <https://platzi.com/clases/1557-git-github/19940-introduccion-a-las-ramas-o-branches-de-git/>
* Platzi. (s.f.). *¿Qué es un branch (rama) y cómo funciona un merge?*. <https://platzi.com/clases/1557-git-github/19947-que-es-un-branch-rama-y-como-funciona-un-merge-en/>
  1. **– 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).

## https://drive.google.com/file/d/1pc1rcuihJ4ct3aW82gDWSMvjzPysEjVQ/view?usp=sharing

* 1. **– 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**](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).

* 1. **– 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. **– 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. | 1. Repositorios de proyectos 2. Manual de Gestor GIT 3. Investigación |

# 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.

# REFERENTES BIBLIOGRÁFICOS

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

# CONTROL DEL DOCUMENTO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | **Javier Leonardo Pineda Uribe** | **Instructor** |  | **27 de Febrero de 2024** |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| **Autor (es)** | Jefferson Alexander Castañeda Rodríguez | Aprendiz |  | 10/05/2025 | Inicio de la guía |