**AUTOMATIZACIÓN DE REDES**

**TALLER No.2 –** **DISEÑO Y DESARROLLO DE SOFTWARE**

**ACTIVIDADES PROPUESTAS:**

1. Conforme equipos de trabajo de 3 integrantes y realice una presentación ágil y dinámica sobre la metodología de desarrollo de software asignada, en la que mencione su enfoque, características, etapas y aplicabilidad. Para la socialización de la actividad tendrá 12 minutos distribuidos de la siguiente manera:

* 7 minutos para la socialización de la metodología asignada – Todos los integrantes del equipo deben participar.
* 5 minutos para integración de la ficha a través de aplicación de actividad didáctica sobre la metodología (Crucigrama, Concéntrese, relacionamiento, mapa interactivo, etc.) Libre elección.

1. Evidencia Individual: Con el objetivo de implementar un GitHub como herramienta de control de versiones realice el siguiente componente practico.

***Ejercicio 1 – Registro e instalación de aplicaciones***

* Realice el registro en GitHub con su correo institucional @misena.edu.co
* Instale en su PC chocolatey y GitHub para establecer un directorio de trabajo local.
* En su PC establezca un directorio de trabajo local, siendo nombrado de acuerdo con la siguiente estructura.

***Ejercicio 2 – Creación y actualización de repositorios***

* En GitHub.com realice la creación del repositorio **AR-Apellido Aprendiz**
* Establezca la inicialización del repositorio, cargue y sincronice la información del directorio de trabajo con el repositorio.
* Compruebe el estado del repositorio.
* Empleando el **Git Bash Here**, cree una carpeta nombrada **GRD\_NombreAprendiz** y dentro de ella los ficheros Teleinformatica.txt y CGMLTI.txt
* Compruebe el estado del repositorio.
* Realice la actualización del repositorio y su respectivo **commit** de los cambios realizados.
* Verifique de nuevo el estado del repositorio.
* A través de la **Git Bash Here**, edite el fichero Teleinformatica.txt para que contenga lo siguiente:
* Visualice los cambios con respecto a la ultima versión que se guardó en el repositorio.

***Ejercicio 3 – Manejo del Historial de cambios***

* Visualice el historial de los cambios en el repositorio.
* Empleando el **Git Bash Here**, edite el fichero CGMLTI.txt con la siguiente información:
* Adicione los cambios al repositorio y realice el commit de los cambios efectuados.
* Visualice las diferencias entre la última versión y la versión anterior del repositorio.
* Muestre quien realizo los cambios sobre el fichero CGMLTI.txt

***Ejercicio 4 – Deshacer Cambios***

* Empleando el **Git Bash Here**, Elimine las dos últimas líneas del fichero CGMLTI.txt y guarde los cambios.
* Compruebe el estado del repositorio.
* Adicione los cambios de estado en el repositorio.
* Compruebe el estado del repositorio.
* Elimine los cambios en el repositorio, pero manténgalos en el directorio de trabajo.
* Verifique de nuevo el estado del repositorio.
* Regrese a la versión anterior del fichero CGMLTI.txt, deshaciendo los cambios realizados en el primer ítem del ejercicio 4.
* Compruebe el estado del repositorio.

***Ejercicio 5 – Branch o ramificaciones***

* Cree un nuevo Branch llamado “Automatizacion” y visualícelo en el repositorio.
* Cambie al Branch “Automatizacion”.
* Verifique el estado del repositorio.
* Desde el Git Bash Here, cree un fichero llamado Script.txt y adicione dos líneas de texto.
* Adicione los cambios al repositorio y su respectivo commit.
* Visualice el historial del repositorio incluyendo todas las ramas.
* Fusione el Branch de “Automatizacion” con el Branch Master.
* Visualice el historial del repositorio incluyendo todas las ramas.
* Elimine el Branch de “Automatizacion”.
* Observe de nuevo el historial del repositorio incluyendo todas las ramas.

**EVIDENCIA(S) A ENTREGAR:**

1. **Evidencia de Conocimiento:** Socialización de las metodologías de desarrollo de software.
2. **Evidencia de desempeño**: Demostración al instructor del funcionamiento del componente practico descrito en la actividad.
3. **Evidencia de producto:** Elabore un video tipo time lapse donde explique los principales comandos de configuración de GitHub para la creación, actualización y gestión de la aplicación de control de versiones. Duración máxima del video: 5 minutos.

**CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | Carmen Elizabeth Fajardo Diaz | Instructor | Teleinformática | 18/02/2022 |

**CONTROL DE CAMBIOS** (diligenciar únicamente si realizan ajustes al taller)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| **Autor (es)** |  |  |  |  |  |