


	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programacion web 2				
TÍTULO DE LA PRÁCTICA:	Git y GitHub				
NÚMERO DE PRÁCTICA:	2	AÑO LECTIVO:	2024	NRO. SEMESTRE:	I (Primero)
FECHA DE PRESENTACIÓN	11/05/24	HORA DE PRESENTACIÓN	20:30		
INTEGRANTE (s) Tapia Huamantuma Antony Juan				NOTA (0-20)	
DOCENTE(s): LINO JOSE PINTO OPPE					

Introduccion
<p>En este informe, se detalla el proceso de estudio y aplicación de Git y GitHub, herramientas fundamentales en el desarrollo de software colaborativo. Se abordará su uso básico, así como algunas recomendaciones y prácticas para optimizar su funcionamiento.</p>
Antecedentes
<p>Antes de profundizar en el estudio de Git y GitHub, es importante comprender su contexto y relevancia en el desarrollo de software. Git es un sistema de control de versiones distribuido ampliamente utilizado, mientras que GitHub es una plataforma que permite alojar proyectos utilizando Git. Ambos son esenciales para la colaboración eficiente en proyectos de software.</p>
Uso de git y github
<p>Conceptos Fundamentales</p> <ul style="list-style-type: none"> - Repositorio: Un repositorio es un espacio donde se almacenan los archivos de un proyecto junto con el historial de cambios. - Commit: Un commit es un registro de cambios en un repositorio. Cada commit tiene un mensaje que describe los cambios realizados. - Branch: Un branch es una línea independiente de desarrollo en Git. Permite trabajar en nuevas características sin afectar la rama principal (master/main). <p>Comandos Básicos de Git</p> <ul style="list-style-type: none"> - git init: Inicializa un repositorio Git en un directorio. - git add: Agrega cambios al área de preparación (staging).

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 2</p>

- **git commit:** Guarda los cambios en el repositorio.
- **git push:** Envía cambios locales a un repositorio remoto en GitHub.
- **git pull:** Obtiene cambios del repositorio remoto y los fusiona con el repositorio local.
- **git branch:** Lista, crea o borra ramas.

Prácticas Recomendadas

- Escribir mensajes de commit claros y descriptivos.
- Crear branches para trabajar en nuevas características.
- Realizar pull requests para revisar y fusionar cambios.
- Mantener un flujo de trabajo ordenado y consistente.

Ejemplo de Uso: Creación de un Repositorio en GitHub

A continuación, se detalla el proceso para crear un repositorio en GitHub y trabajar en él utilizando Git.

- Crear una cuenta en GitHub si aún no se tiene una.
- Iniciar sesión en GitHub y hacer clic en el botón "New" para crear un nuevo repositorio.
- Asignar un nombre al repositorio, agregar una descripción opcional y elegir la visibilidad del repositorio.
- Clonar el repositorio en la máquina local utilizando el comando git clone.
- Realizar cambios en los archivos del repositorio, agregar los cambios con git add y hacer un commit con git commit.
- Enviar los cambios al repositorio remoto en GitHub con git push.

CONCLUSIONES

El estudio de Git y GitHub es fundamental para cualquier desarrollador de software, ya que facilita la colaboración, el seguimiento de cambios y la gestión de versiones en proyectos. Con una comprensión sólida de sus conceptos y prácticas recomendadas, se puede mejorar significativamente la eficiencia y calidad del trabajo en equipo.

METODOLOGÍA DE TRABAJO

Programar todos los ejercicios, con los conocimientos dados en clases de Fundamentos de Programación 2.

REFERENCIAS Y BIBLIOGRAFÍA

<https://docs.oracle.com/javase/8/docs/api/>

Contenido y demostración		Puntos		Checklist Estudiante	
				Profesor	
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error u omisión)	4			
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	2			
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	2			
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	1			
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2			
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4			
Total		16			

