

TRABAJO PRÁCTICO #1

INTRODUCCIÓN AL IoT

TECNICATURA SUPERIOR EN
TELECOMUNICACIONES

ALUMNO: SANTIAGO PÉREZ SEGOVIA

PROFESOR: GONZALO VERA

MATERIA: PROYECTO INTEGRADOR

FECHA DE ENTREGA: 14/04/2024

¿Qué es GIT?

Git es un sistema de control de versiones distribuido diseñado para rastrear cambios en archivos y coordinar el trabajo entre múltiples personas en proyectos de desarrollo de software.

¿Que permite la utilización de GIT?

Permite a los desarrolladores trabajar de manera colaborativa en un proyecto, registrar cambios en el código fuente, mantener un historial de versiones y fusionar fácilmente el trabajo de diferentes personas.

Qué pueden realizar los desarrolladores con GIT?

Con Git, los desarrolladores pueden realizar un seguimiento de los cambios realizados en el código y revertir a versiones anteriores si es necesario. También facilita la colaboración al permitir que múltiples personas trabajen en diferentes partes del código al mismo tiempo, sin temor a sobrescribir los cambios de los demás.

¿Que es GITHUB?

GitHub es una plataforma de alojamiento de código basada en la web que utiliza Git como su sistema de control de versiones subyacente.

¿Que permite Github a los desarrolladores?

Permite a los desarrolladores almacenar, gestionar y colaborar en proyectos de software de forma remota. Además de alojar repositorios Git, GitHub ofrece una variedad de características y herramientas adicionales, como seguimiento de problemas, solicitudes de extracción (pull requests), wikis y proyectos, que facilitan la colaboración y la gestión de proyectos de desarrollo de software.

Conclusión.

Git es el sistema de control de versiones que permite a los desarrolladores rastrear cambios en el código y colaborar en proyectos, mientras que GitHub es una plataforma que utiliza Git para alojar repositorios de código y proporcionar herramientas adicionales para la colaboración y gestión de proyectos.

Uso de Git:

1. **Instalación de Git:** Primero, asegúrate de tener Git instalado en tu computadora. Puedes descargarlo e instalarlo desde git-scm.com.
2. **Inicializar un Repositorio Git:** Para comenzar a rastrear los cambios en tu proyecto, inicializa un repositorio Git en el directorio de tu proyecto utilizando el comando `git init`.
3. **Agregar Archivos al Área de Preparación:** Utiliza el comando `git add` para agregar archivos modificados al área de preparación (staging) para el próximo commit.
4. **Confirmar Cambios:** Una vez que hayas preparado los archivos, confirma los cambios con el comando `git commit`. Asegúrate de incluir un mensaje descriptivo utilizando la opción `-m`.
5. **Gestionar Ramas (Branches):** Puedes crear, cambiar y fusionar ramas utilizando los comandos `'git branch'`, `'git checkout'` y `'git merge'`.
6. **Colaborar con Repositorios Remotos:** Utiliza comandos como `git remote` y `git push` para colaborar con repositorios remotos alojados en servicios como GitHub.

Uso de GitHub:

1. **Crear una Cuenta:** Si aún no tienes una, crea una cuenta en GitHub.
2. **Crear un Repositorio:** En tu página principal de GitHub, haz clic en el botón "New" para crear un nuevo repositorio. Proporciona un nombre, una descripción opcional y configura la visibilidad del repositorio.
3. **Clonar un Repositorio:** Para trabajar en un repositorio existente, clona el repositorio remoto a tu computadora utilizando el comando **git clone**.
4. **Enviar Cambios:** Después de realizar cambios en tu proyecto local, agrega los cambios al área de preparación con **git add**, confirma los cambios con **git commit** y envía los cambios al repositorio remoto con **git push**.
5. **Crear Pull Requests:** Si deseas contribuir a un proyecto de código abierto o colaborar con otros desarrolladores, crea un pull request para proponer los cambios al repositorio principal.
6. **Explorar Repositorios:** Utiliza la interfaz de GitHub para explorar y descubrir nuevos proyectos, colaborar en proyectos existentes y participar en la comunidad de desarrollo de software.