

Temario curso git

INTRODUCCIÓN A GIT Y CONTROL DE VERSIONES

FECHAS DE REALIZACIÓN: del 9 al 20 de junio

Nº DE EDICIONES: 1

HORAS LECTIVAS POR EDICIÓN: 20

HORARIO: 9:00 a 11:00h

PLAZAS: 20

LUGAR DE IMPARTICIÓN: Online síncrono

PRESENTACIÓN Y OBJETIVOS:

Usar un control de versiones como Git es fundamental en el uso de diversos lenguajes de

programación (R, Stata, Python, etc), ya que permite a los programadores llevar un seguimiento detallado de cada cambio en el código a lo largo del tiempo. El objetivo de este curso es entender los conceptos fundamentales de Git y utilizarlo en proyectos Colaborativos.

PERFIL DE LOS PARTICIPANTES:

El curso está dirigido a todo el personal relacionado con el uso de la programación en R, Stata, Python, etc. Se recomienda tener nociones básicas de programación.

CONTENIDOS/PROGRAMA:

Introducción a Control de Versiones y Git:

- ¿Qué es el control de versiones?.
- Problemas comunes que soluciona el control de versiones.
- Introducción a Git: historia y conceptos clave.
- Instalación de Git en diferentes sistemas operativos.
- Configuración inicial de Git (git config, user.name, user.email).
- Primeros pasos en la terminal para usuarios que no la conozcan.

Conceptos Básicos de Git:

- Creación y clonación de repositorios.
- Ramas (branches): qué son y para qué se utilizan.
- Concepto de commit: qué es y su importancia en el registro de cambios.
- Comandos básicos: git init, git clone, git add, git commit, git status, git log.
- Ciclo de vida de los archivos en Git: Untracked, Staged, Modified, Committed.
- Visualización de cambios: git diff.

Trabajo con Ramas:

- Creación y gestión de ramas (git branch, git checkout, git switch).
- Fusión de ramas (Merge): tipos de merge y cómo resolver conflictos.
- Estrategias de branching en proyectos individuales y colaborativos.
- Comandos adicionales para trabajar con ramas: git merge, git log --online --
graph.

Git Remoto y Trabajo en Equipo:

- Repositorios remotos y su configuración (git remote, git push, git pull, git fetch).
- Introducción a plataformas como GitHub, GitLab o Bitbucket.
- Clonación de repositorios remotos.
- Manejo de ramas en repositorios remotos.
- Flujo de trabajo colaborativo: Pull Request (PR) y revisión de código.
- Resolución de conflictos en repositorios remotos.

Buenas Prácticas en el Uso de Git:

- Importancia de los mensajes de commit claros y concisos.
- Convenciones de nombres para ramas y organización de commits.
- Estrategias para mantener un historial de commits limpio (squash, rebase).
- Creación de .gitignore para excluir archivos innecesarios del repositorio.

- Restaurar cambios con git revert, git reset, y git checkout.

Git Avanzado y Herramientas Útiles:

- Comandos avanzados: git stash, git tag, git cherry-pick.
- Trabajo con submódulos.
- Introducción a Git hooks: automatización de tareas.
- Integración de herramientas externas (extensiones, plugins de editor).
- Análisis de historial con git blame y git bisect para depuración.

Proyecto Final:

- Planteamiento del proyecto: desarrollo de una aplicación simple o documento
- colaborativo.
- Organización del equipo y asignación de tareas.
- Integración de todo el flujo de trabajo Git en el proyecto.
- Revisión de código entre pares y Pull Requests en GitHub.