**GIT**

**INTRODUCCION**

**SISTEMA DE CONTROL DE VERSIONES**

Estos sistemas son herramientas que permiten realizar un seguimiento de los cambios y tambien permiten proteger el codigo de errores accidentales. Tambien este sistema permite el trabajo en equipo.

**CONCEPTO DE VERSION**

(Tambien llamado revision o edicion)

Hace referencia al estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificacion.

Los sistema de control de versiones utilizan repositorios para almacenar el proyecto actualizado junto a sus cambios historicos. Los sistemas de control de versiones centralizados almacenan todo el codigo en un unico repositorio, es decir que un unico servidor contiene todos los archivos versionados.

**VENTAJAS**

**HISTORIAL DE CAMBIOS:** Se puede saber el autor, la fecha y notas escritas sobre los cambios realizados. Tambien permite volver a versiones anteriores.

**CREACION Y FUSION DE RAMAS:** Al tener varios integrantes del equipo trabajando al mismo tiempo, cada uno en una tarea diferente, pueden beneficiarse de tener flujos de trabajo independientes. Posteriormente se pueden fusionar estos flujos de trabajos o ramas a una principal.

**TRAZABILIDAD DE CAMBIOS EN SOFTWARE:** Poder conectar el sis. De control de versiones con un software de gestion de proyectos y seguimiento de errores ayuda con el analisis de la causa raiz de los problemas y con la recopilacion de informacion.

**GIT**

Este sistema presenta una arquitectura distribuida, cada desarrollador posee una copia del trabajo en un repositorio local donde puede albergar el historial completo de todos los cambios y, mediante determinados comandos, realiza sincronizaciones al repositorio remoto

**AREAS Y ESTADOS**

**SECCIONES FUNDAMENTALES**

**DIRECTORIO DE TRABAJO (WORKING DIRECTORY):** Es una copia de una version del proyecto. Son archivos sacados de la base de datos compriida y se colocan en el disco para poder ser usados o modificados

**AREA DE PREPARACION (STAGING AREA):** Es un archivo que se encuentra dentro del directorio de Git y que contiene informacion acerca de lo que va a ir en la proxima confirmacion

**DIRECTORIO DE GIT (LOCAL REPOSITORY):** Es el lugar donde se almacenan los metadatos y la base de datos de objetos del proyecto. Es lo que se coia cuando se clona un repositorio desde otra fuente

**REPOSITORIO REMOTO (REMOTE REPOSITORY):** Es el repositorio que se encuentra en un servidor remoto y con el que eventualmente se sincronizan los trabajos entre los diferentes integrantes del equipo

**COMANDOS BASICOS**

**git init:** Inicializa un directorio como repositorio Git, se ejecuta dentro del directorio del proyecto y, como resultado, crea un subdirectorio .git que contiene todos los archivos para poder realizar el seguimiento de los cambios, etiquetas, etc.

**git add <file>:** Luego de cambios en un archivo, estos quedan unicamente en el area de trabajo, por lo tanto es necesario pasarlos al area de preparacion mediante el uso de comando git add, para que sea incluido dentro de la siguiente confirmacion (commit)

**git status:** Permite conocer en que estado se encuentran los archivos

**git commit:** Confirma todos los cambios registrados en el area de preparacion, o lo que es lo mismo, se pasan los cambios al repositorio local.

**git push:** Envia todas las confirmaciones registradas en el repositorio local a un repositorio remoto

**git pull:** Funciona al inversio de git push, trayendo todos los cambios al repositorio local, pero tambien dejandolos disponibles directamente para su modificacion o revision en el area de trabajo. Es importante mencionar que se utiliza cuando ya se tiene un repositorio local vinculado a uno remoto.

**git clone:** En el caso de necesitar “bajar” un repositorio remoto de algun proyecto ya existente se puede ejecutar este comando. Genera un directorio (con el nombre del repositorio o uno especificado) que contiene todo lo propio al proyecto, ademas del subdirectorio .git necesario para poder gestionar los cambios y todo lo pertinente al repositorio Git.