

## Anaconda

Es considerada un plataforma o suite de código abierto que funciona como un gestor de entorno para el desarrollo de ciencia de datos y aprendizaje automático (machine learning) en Python, así mismo está integrada por una serie de aplicaciones, librerías y paquetes. Uno de los objetivos principales es simplificar el despliegue y administración de los paquetes y librerías para ciencia de datos.

Anaconda está integrada por cuatro grandes áreas: anaconda navigator, anaconda project, librerías de ciencia de datos y Conda (sistema de gestión de paquetes).

Algunas de las características principales son: es de código abierto, multiplataforma, administrar de forma fácil paquetes, dependencias y entornos para ciencia de datos, múltiples aplicaciones, herramientas e IDEs y simplifica la implementación de proyectos de ciencia de datos.

Como conclusión para crear entornos de desarrollo de ciencia de datos en donde estos deban ser fáciles de implementar, gestionar de manera adecuada las dependencias y librerías del proyecto, en donde no dependamos de la plataforma o sistema operativo, anaconda es una buena opción.

## Git

Es un software de código abierto para el control de versiones con enfoque en el desarrollo de software y trabajo en equipo (sistema de control de versiones). El control de versiones es la gestión de los cambios que se van haciendo a lo largo del tiempo de un producto (para este caso software) por un equipo de trabajo o de forma individual.

Git a diferencia de otros sistemas de control de versiones destaca por su arquitectura distribuida, en la que cada usuario tiene una copia completa del proyecto incluyendo todo su historial, cada copia tiene el mismo nivel y no existe repositorio central.

Git nos puede apoyar en la organización del código de software, manejar el historial de cambios y por ende su evolución, navegando entre las diferentes versiones y creando nuevos ambientes de trabajo a partir del origen por medio de ramas (branches). Así mismo con Git podemos tener un trabajo colaborativo, en donde varias personas están trabajando sobre el mismo software y todos necesitan estar sincronizados.

A manera de conclusión Git puede ser aplicado a cualquier producto donde exista la necesidad de controlar las versiones y trabajo en equipo.