Juan David Quintero 201910905

Nikolas Santacruz 201821946

Catalina Holguin 201728211

1. ¿Cuál es el ciclo regular para actualizar código en un repositorio GIT?

Primero es necesario realizar un Git: pull que actualice los archivos locales con lo que se encuentra subido en el repositorio. Si mis compañeros hacen commit de sus cambios en la misma rama, debo traer sus cambios a mi computador antes de poder subir los míos. A continuación, al hacer uso de Git: Stage all changes, añado todos los cambios que he hecho al commit que voy a realizar. Después, puedo hacer el commit utilizando Git: commit. Finalmente, Git: push sube el commit realizado al repositorio en GitHub.

1. ¿Qué ventajas y limitantes tiene el uso de Ramas/Branches?

El uso de Ramas tiene muchas ventajas al momento de trabajar en un proyecto. En primer lugar, mantiene todo el trabajo y la información en un mismo lugar y, además, todos los colaboradores tienen la posibilidad ingresar a la misma rama para trabajar en ella. Por otro lado, puede resultar fácil abandonar Ramas y que estas resulten amontonadas, adicionalmente, al usar Ramas siempre es necesario agregar a los miembros del equipo como contribuidores antes de que puedan crear una rama propia.

1. ¿Cuáles serían las buenas prácticas para solucionar conflictos?

* Tener un estándar para nombrar los componentes de las funciones, variables etc.
* Coordinar las responsabilidades de cada miembro del grupo.
* Que no se sobrescriban los trabajos realizados por otros integrantes del grupo.
* Utilizar herramientas para la integración del trabajo colaborativo como el merge.

1. ¿Qué orden de complejidad tendría las funciones (consulta y lectura de archivo)?

Es posible que estas funciones tengan un orden de crecimiento lineal (n). Esto se debe a que el archivo de csv se guarda en memoria utilizando una forma matricial en donde las columnas son representadas con listas. Al analizar las características de esta estructura de python se puede intuir que la lista se guarda en memoria como una lista encadenada, debido a que si se quiere encontrar un elemento sin tener su ubicación se tiene que pasar por todas las posiciones anteriores al dato objetivo. Al guardar en memoria los datos del archivo de csv también se utiliza los diccionarios para representar las filas. Gracias a que se utilizan diccionarios en vez de listas, el orden de crecimiento no se incrementaría, ya que al buscar un dato en esta estructura se puede encontrar sin tener que recorrer los elementos anteriores, por esta razón los diccionarios se pueden relacionar con los arreglos.

1. ¿Cómo podría reducir o aumentar la complejidad de la consulta?

Si se puede aumentar la complejidad de la consulta dependiendo de cuales estructuras de datos se escoger para guardar en memoria los archivos a utilizar. Es posible que no se pueda reducir la complejidad si es que se utilizan funciones como loadCSVFile() para cargar los datos, porque este tipo de herramientas ya han sido optimizadas para cumplir la tarea en el menor tiempo posible.

1. ¿Cómo afecta un TAD en la complejidad?, ¿Qué alternativas existen

Si las estructuras de los datos ya tienen un tamaño fijo, arreglos, la lectura de estos será mucho más rápida, casi instantánea ya que se sabe en dónde están los datos, pero el insertar o borrar datos que no esté en la última posición es más tardío y complicado. Por otro lado, las listas se vuelven mucho más fácil agregar o borrar datos ya que estos no están unidos en una secuencia o atadas a un tamaño fijo, los datos pueden estar en cualquier posición de la memoria, esto los hacen más complejos ya que cada dato tiene que tener su información y además la posición del siguiente dato.