# **Retrospectiva**

## **Instrucciones para modificar la función para cargar archivos y consultar datos:**

* ¿Cómo implementaría la función?
  + Para esta función, primero se va a crear una nueva lista para almacenar los diccionarios con los datos de cada línea, seguido de esto se va a abrir el archivo correspondiente solo para lectura, y se va a leer línea por línea. Después se va a realizar un recorrido por cada una de estas, y se van a separar todos los datos con “;”, ya con los datos separados, se empezarán a crear los diccionarios con los datos de cada línea, y al mismo tiempo estos diccionarios serán añadidos a la lista principal. Al hacer el recorrido total, la función tiene como valor de retorno la lista con los diccionarios.
* ¿Qué orden de complejidad tendría la función implementada?
  + Esta función tendría un orden de complejidad o(n), ya que el tiempo que se demora en ejecutar la función, es proporcional a la cantidad de datos que tenga el archivo a trabajar.

## **Discusión en clase sobre GitHub, VS Code y Código implementado.**

* ¿Cuál es el ciclo regular para actualizar código en un repositorio GIT?
  + Primero hay que traer todos los cambios de la rama actual al repositorio local utilizando “git pull”, de esta manera se evitan algunos conflictos de versiones.
  + Después de trabajar en el código, se revisan cuales cambios se quieren mantener y se corrigen los conflictos entre las versiones utilizando “git stage changes”.
  + Luego de revisar todos los cambios que puedan causar conflictos, utilizando “git commit”, se crea un commit que contiene los cambios y un comentario del desarrollador que permite rastrear el conjunto de cambios en un árbol de tiempo.
  + Por último, usando el comando “git push” se actualiza la rama actual con los cambios almacenados en el commit
* ¿Qué ventajas y limitantes tiene el uso de Ramas/Branches?
  + Ventajas: gracias al uso de ramas, es posible experimentar con el código sin temor a dañar el trabajo de otros o afectar la estabilidad actual del programa.
  + Desventajas: integrar todos los cambios a la rama principal puede ser un dolor de cabeza si no se tiene una buena comunicación con el equipo de trabajo
* ¿Cuáles serían las buenas prácticas para solucionar conflictos?
  + Una buena práctica para solucionar los conflictos es traer los cambios al repositorio local utilizando “git pull” antes de realizar un commit
  + Además, es indispensable discutir los conflictos de código con el equipo antes de corregirlos, de esta manera será menos difícil fusionar ramas
* ¿Qué orden de complejidad tendría las funciones (consulta y lectura de archivo)?
  + Las funciones de consulta y lectura tendrían una complejidad de orden o(n), ya que el tiempo que se demora en desarrollarse la función aumenta proporcionalmente respecto a la cantidad de datos suministrados, y asimismo se reduce la complejidad de ambas funciones
* ¿Cómo podría reducir o aumentar la complejidad de la consulta?
  + Todo depende de cómo se realicen la función de la consulta, ya que, si se ejecutan recorridos totales dentro de otro recorrido, nos va a aumentar la complejidad a una cuadrática, en cambio si simplemente ejecutamos un recorrido para toda la función, o lo dividimos en recorridos parciales, nos puede bajar la complejidad a una lineal.
* ¿Cómo afecta un TAD en la complejidad?, ¿Qué alternativas existen?
  + El TAD no afecta la complejidad, pues el TAD es un intermediario entre la estructura de datos y la solución, por lo tanto, no causaría ningún efecto en la complejidad.