Minutas Grupo

Integrantes:

* Castorina, Matias
* Frangolini, Luciano
* Noseda, Demian

# 18/10/2021

* Comenzamos el desarrollo del nodo tracker.
* Se plantearon distintas estructuras y los datos necesarios.
* Se implemento una función store primitiva que guardaba la información de un archivo y el nodo par que debería tenerlo, de forma local.

# 20/10/2021

* Se implementa envió de mensajes en el tracker vía sockets udp.
* Se envía un store rudimentario de un tracker a otro, forzando a que se guarde localmente en la llegada del mensaje, pero no funciona.

# 23/10/2021

* El error anteriormente mencionado se debía a que no estaba bien definido la recepción de un mensaje, y al utilizar el mismo código, los trackers se respondían en un bucle infinito.
* Se envía información de un tracker a otro y se guarda de forma exitosa.
* Se reestructura el tracker.
* Se plantea la duda del uso exacto de messageId en la interfaz del store entre trackers, queda para consultar la próxima clase.

# 29/10/2021

* Se consulta sobre messageId en clase y se definen las interfaces para la interacción entre pares y pares, y pares y trackers.
* Se define una forma de dividir los dominios vía configuración, la cual consta en dividir el mismo sobre el total de nodos trackers asignando un espacio de direcciones iguales a cada uno, exceptuando al último que poseerá más si es que la división no diera un cociente entero.

# 01/11/2021

* Se desarrolla un cliente web para las funciones preestablecidas utilizando React.
* Se prueba la función store definiendo en la recepción del mensaje si estoy en el dominio correcto, con dos trackers corriendo simultáneamente.

# 02/11/2021

* Se desarrolla el servidor con las correspondientes interfaces cliente-servidor y queda a la espera de la finalización del desarrollo de los nodos trackers y pares para conectarse y testearse.
* Se completa la función store y se prueba con 3 trackers corriendo simultáneamente, pero se olvidó integrar correctamente el caso donde colisiona sobre el dominio, pero no es el mismo hash (o sea el mismo archivo en cuestión).

# 03/11/2021

* Se corrigió el error en el guardado de las colisiones por dominio, ahora la función store funciona de la forma esperada. Recibiendo como input la interfaz establecida, revisará si la el hash corresponde a ser guardado en su dominio, de no ser así lo pasará a otro tracker vía mensaje udp, y revisará la ruta para así saber que debe invocar nuevamente a la función store y repetir el proceso.
* Se ponderan formas de implementar cambios en los dominios de forma dinámica para la implementación de la función join.

# 04/11/2021

* Se implementó la función scan en el tracker y se prueba con 3 trackers corriendo simultáneamente.
* Intentamos conectar el servidor con uno de los trackers, pero encontramos el problema de la conexión TCP del cliente/servidor con los mensajes UDP del servidor/tracker. Buscamos una forma de esperar la llegada de un mensaje de un tracker, posiblemente con un timeout de conexión por si se cayera un nodo tracker en el proceso de scaneo.

# 05/11/2021

* Se implementó la función count en el tracker y se prueba con 3 trackers corriendo simultáneamente.

# 09/11/2021

* Se agrego un Map en los pares para que estos puedan conocer que archivos tienen y responder a las solicitudes de descarga con la información que se obtiene de los .torrente.
* Se modificaron las funciones que solicitan/emiten la descarga de un archivo para buscar en el Map el mismo y enviarlo si se encuentra almacenado.

# 12/11/2021

* Implementado la búsqueda de que pares tiene un archivo para la comunicación par/tracker aunque falta modificar la función para que tome todos los datos por parámetro ya que se encuentra hardcodeado para realizar pruebas. Las pruebas se realizaron solicitando la búsqueda de un archivo desde un par y tres trackers corriendo simultáneamente. Gracias a esta prueba se encontró un error en la función found de los tracker, y se completo de manera correcta el uso de la interfaz de los mismos.