Proyecto BITCORRENT ler Cuatrimestre 2022

Los 4 Rustásticos

Dante Reinaudo - Sofia Feijoo - Facundo Milhas - Tizziana Mazza

TEMARIO

Entidades	
Llegada archivo .torrent	
Conexión con el tracker	
Conexión con peers	
Descarga de piezas	
Servidor	
Concurrencia	
Interfaz	





Bit Client
Bit field
Downloader
Logger
Peers

PeerProtocol

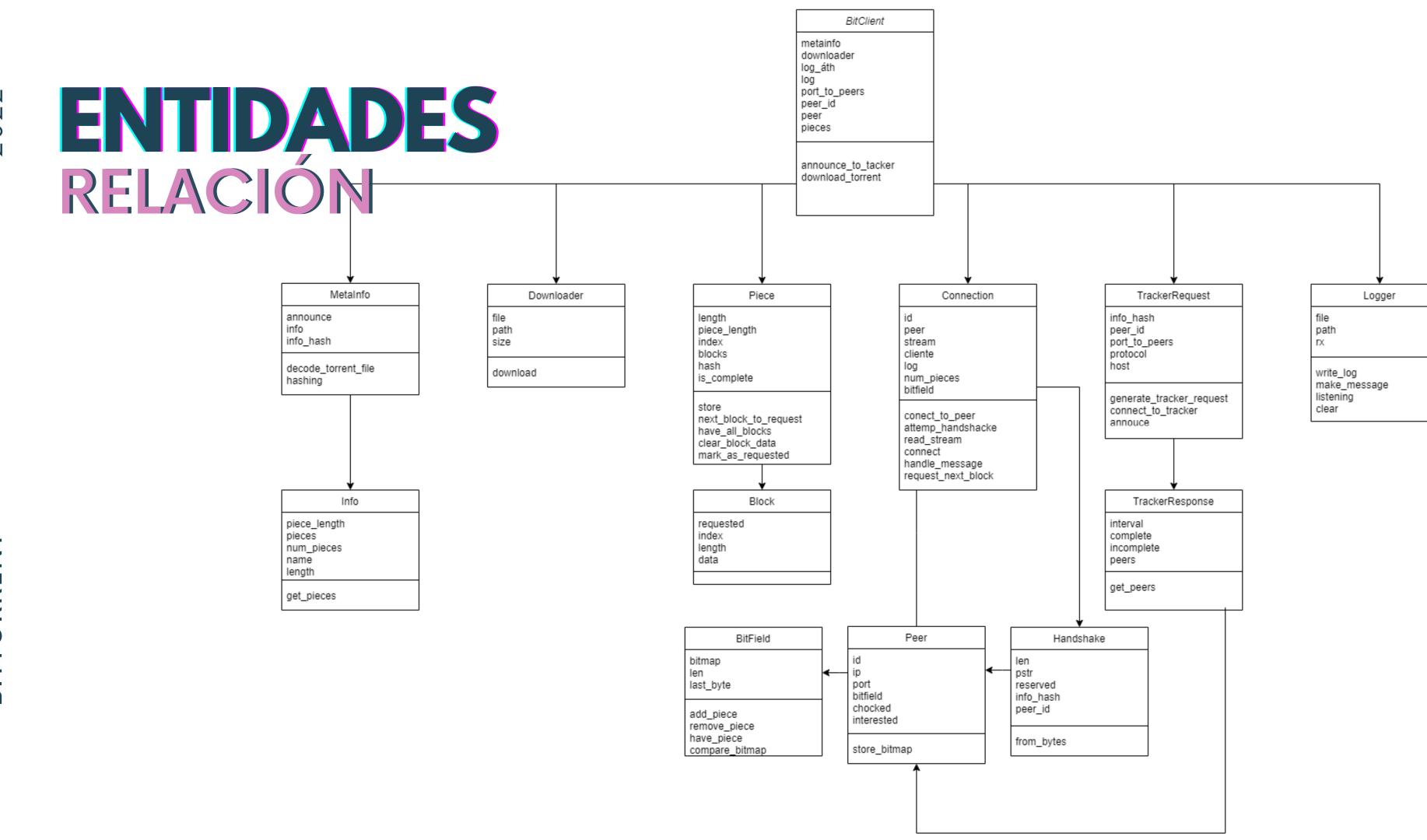
Piece

Block

Connection

Metainfo

Tracker



ENTIDADES BITCLIENT

BitClient metainfo downloader log áth log port_to_peers peer_id peer pieces announce_to_tacker download torrent Connection TrackerRequest Logger id info_hash file peer_id path peer stream port_to_peers ΓX cliente protocol host write_log num_pieces make_message bitfield listening generate_tracker_request clear connect_to_tracker conect_to_peer annouce

MetaInfo announce info_hash

decode_torrent_file hashing

Downloader

path size

download

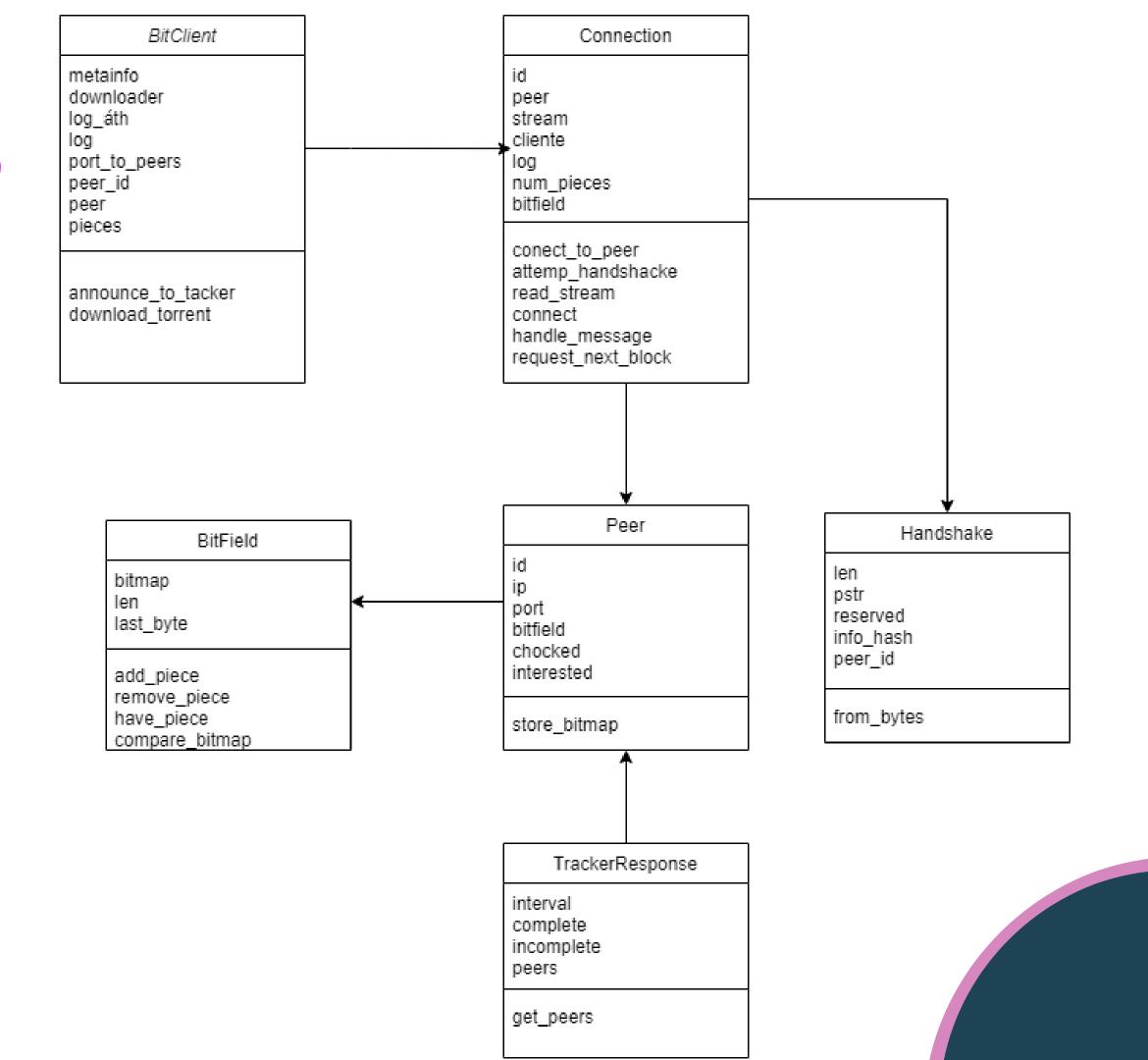
Piece

length piece_length index blocks hash is_complete

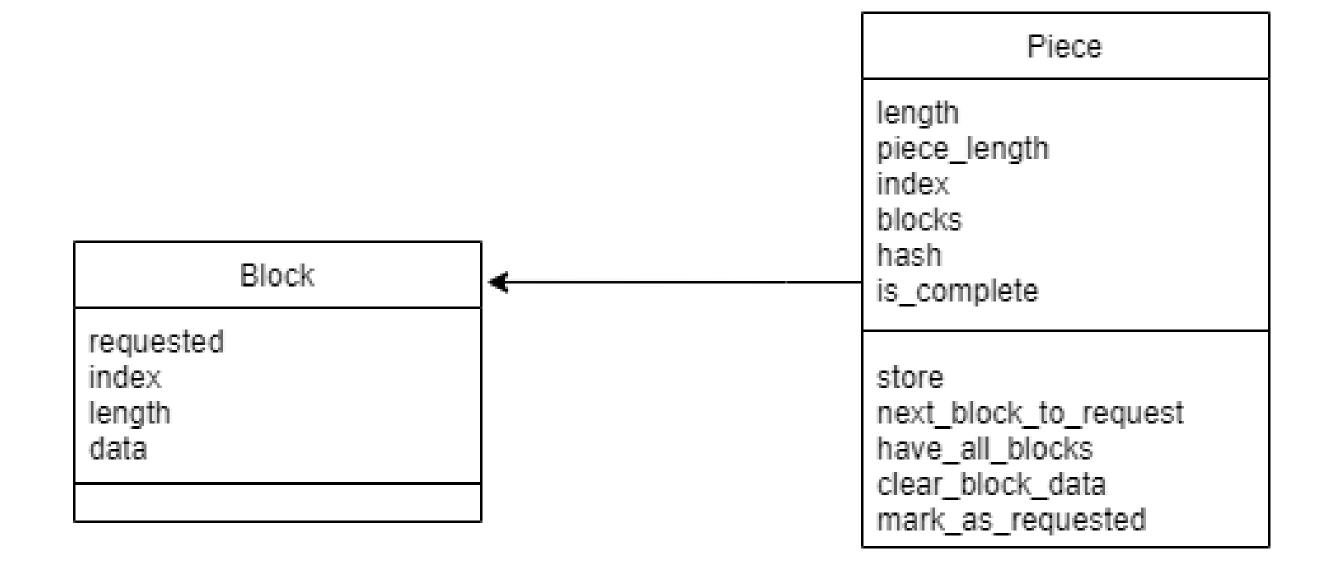
store next_block_to_request have_all_blocks clear_block_data mark_as_requested

attemp handshacke read_stream connect handle_message request_next_block

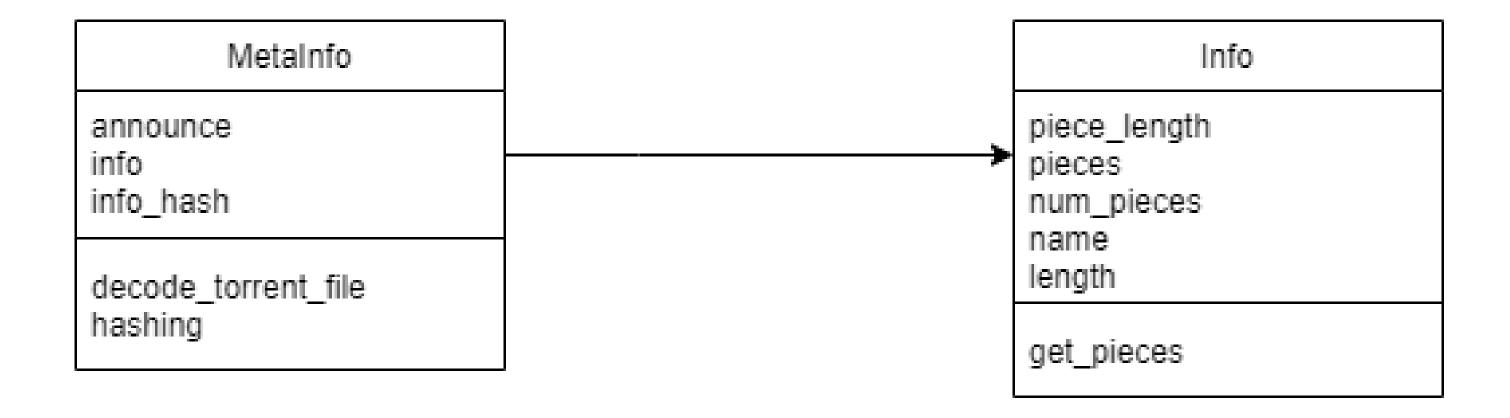
ENTIDADES PER



ENTIDADES PIECE - BLOCK



ENTIDADES METAINFO



ENTIDADES TRACKER



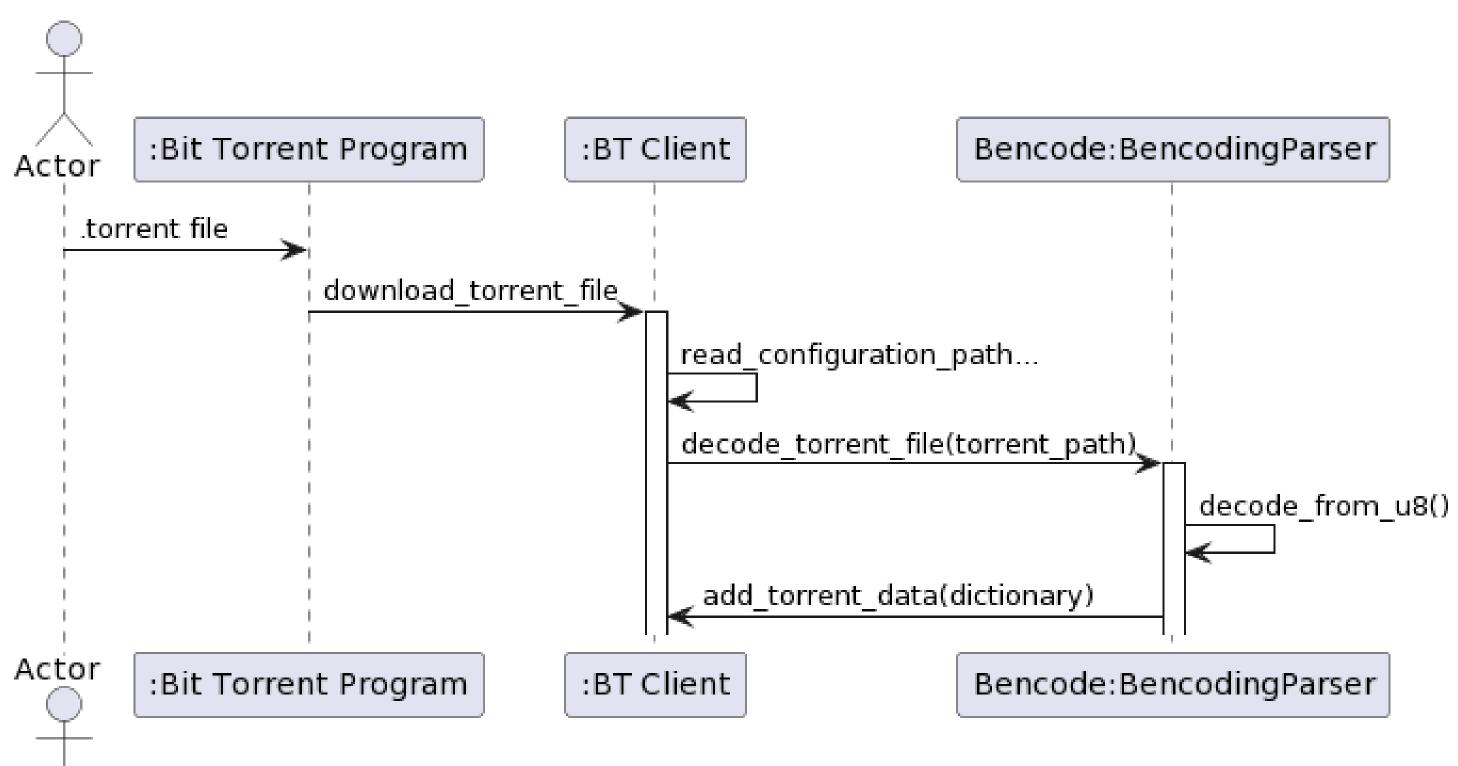


LLEGADA DEL ARCHIVO TORRENT

Nos llega un archivo .torrent, lo leemos, encodeamos su informacion y nos guardamos la data importante para luego poder utilizarla a la hora de conectarnos con un peer y descargar lo deseado.

LLEGADA DEL ARCHIVO .TORRENT

DIAGRAMA SECUENCIAL Llegada de un archivo .torrent



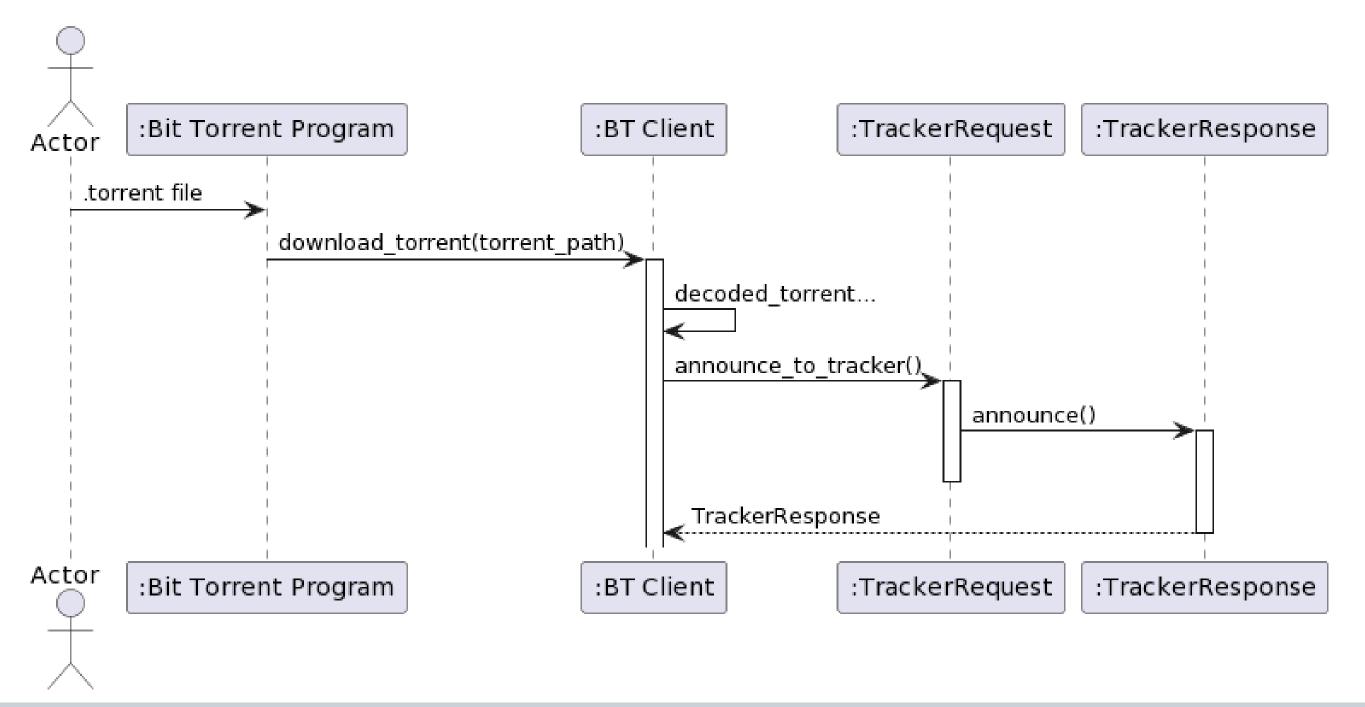


COMUNICACIÓN CON EL TRACKER

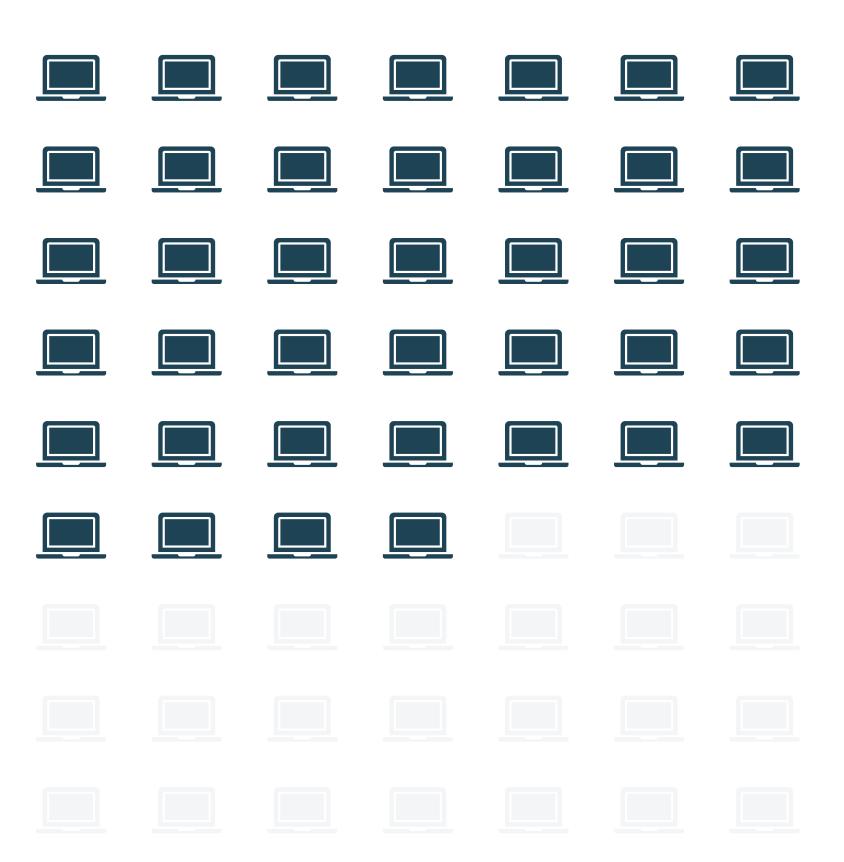
Como clientes, lo primero que tenemos que hacer es comunicarnos con el tracker de forma tal que nos de la información de los peers que tienen lo que estamos buscando.

COMUNICACIÓN CON EL TRACKER

DIAGRAMA SECUENCIAL Conexión con el tracker



El bitClient hace uso de la estructura TrackerRequest para ordenar y modelar los campos a enviar al Tracker. Con la misma hacemos el announce, recibiendo la respuesta y almacenandola en la estructura TrackerResponce.

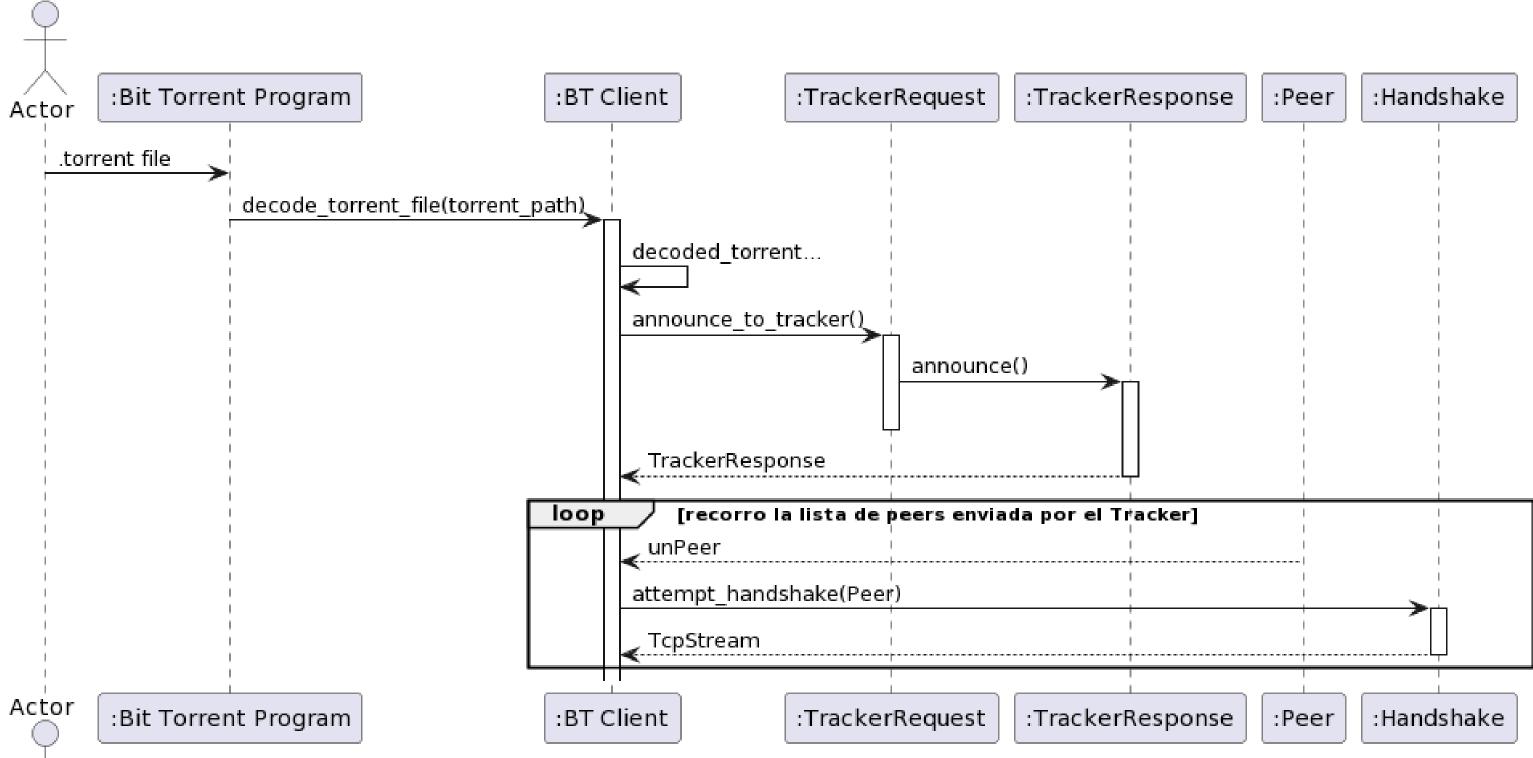


COMUNICACIÓN CON EL PEER

Llega el momento de hacer el handshake con el peer que tiene la pieza que queremos descargar. Una vez realizado el handshake se continua con una serie de mensajes hasta que nos mande la pieza deseada.

COMUNICACIÓN CON EL PEER

HANDSHAKE E intercambio de mensajes con un peer



COMUNICACIÓN CON EL PEER

HANDSHAKE E intercambio de mensajes con un peer

KEEP ALIVE

BITFIELD

INTERESTED

NOT INTERESTED

HAVE

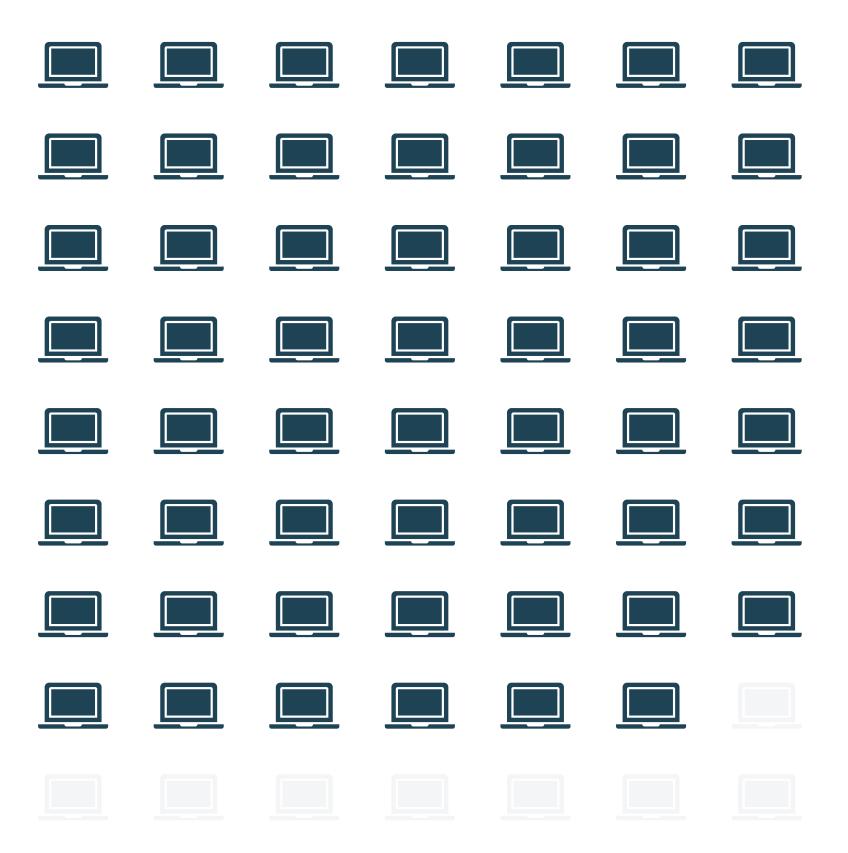
CHOCKE

UNCHOCKE

REQUEST

PIECE

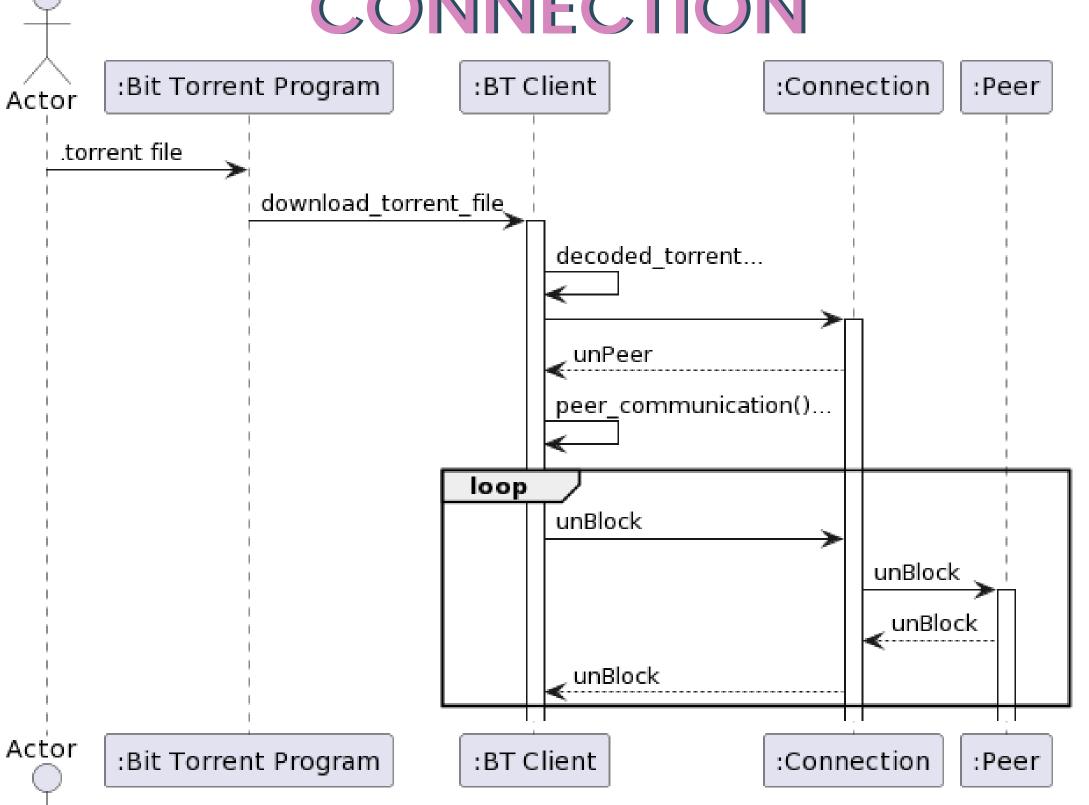
CANCEL



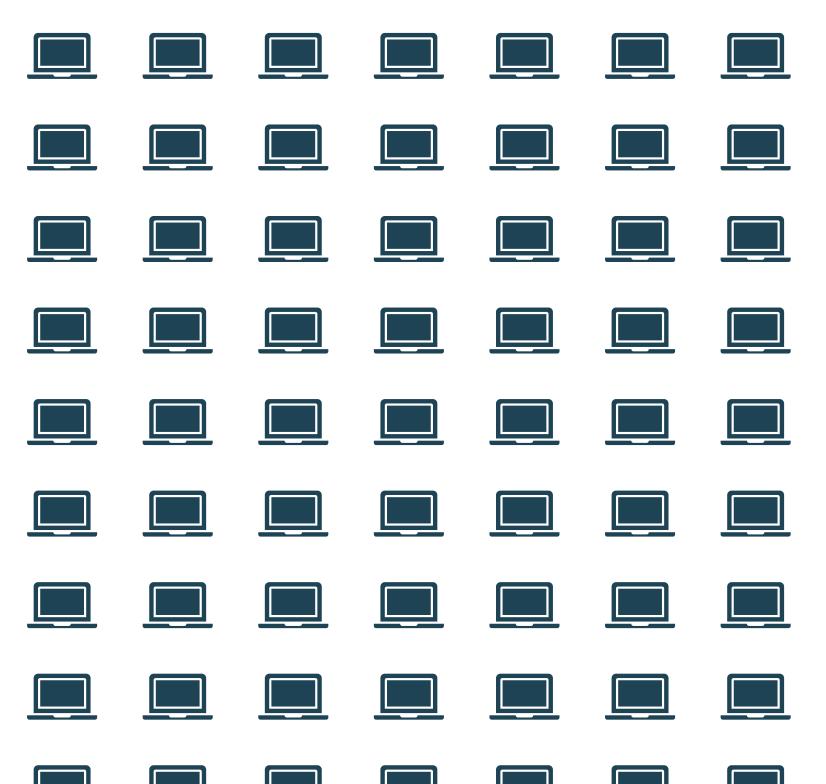
DESCARGA DE PIEZAS

Obtenemos los bloques de las piezas deseadas, y de esta forma completamos las piezas.

DESCARGA DE PIEZAS CONNECTION



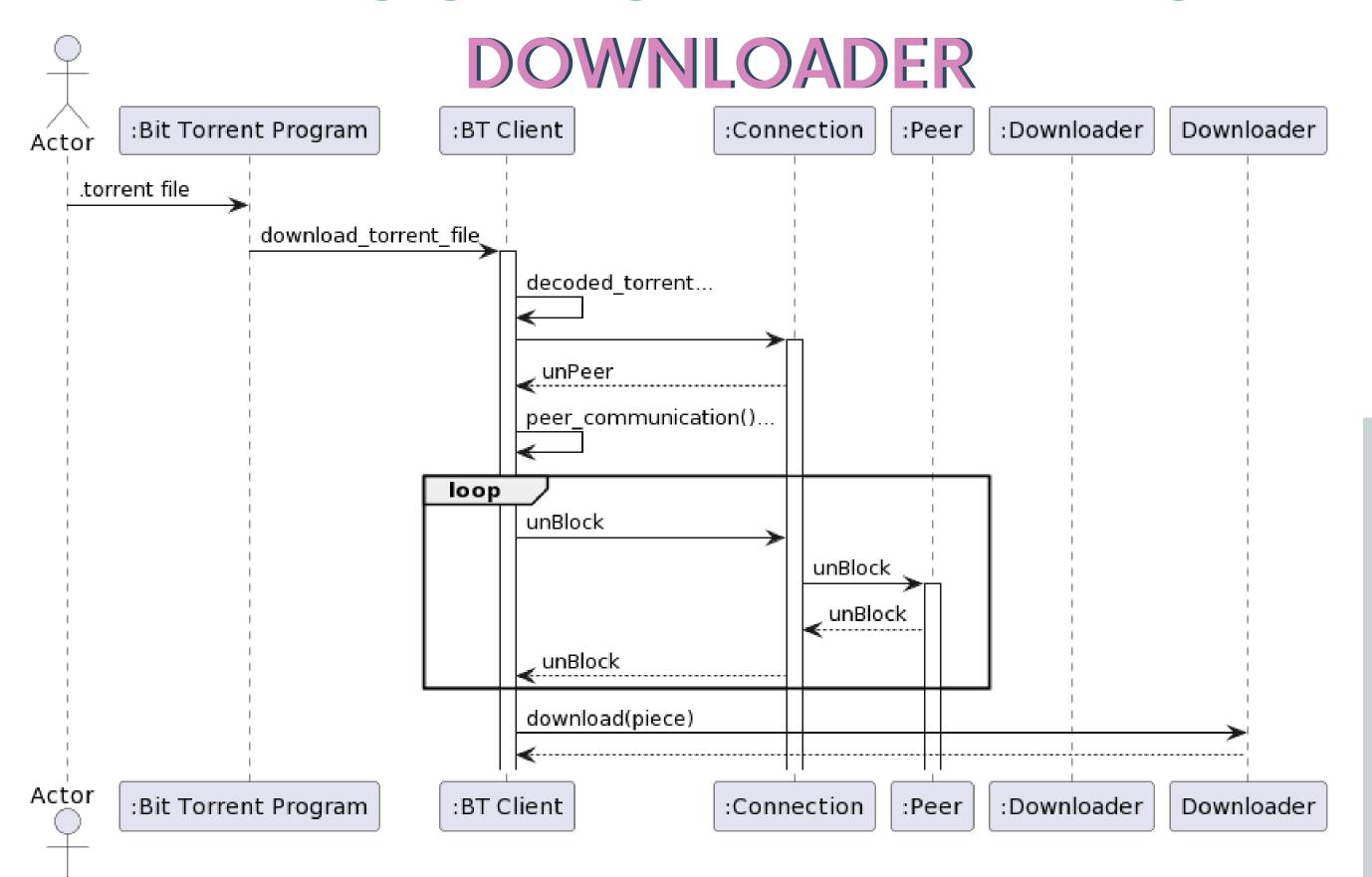
Una vez
enviados los
mensajes
correspondient
es, realizamos
el Request de
la pieza que
nos interesa.



TORRENT DESCARGADO

Una vez que tenemos las piezas las unimos para construir el torrent que se está descargando.

DESCARGA DE PIEZAS



Una vez
recibida la
pieza, es
pasada a la
entidad
Downloader, que
crea un archivo y
lo almacena en
disco.



BITTORRENI













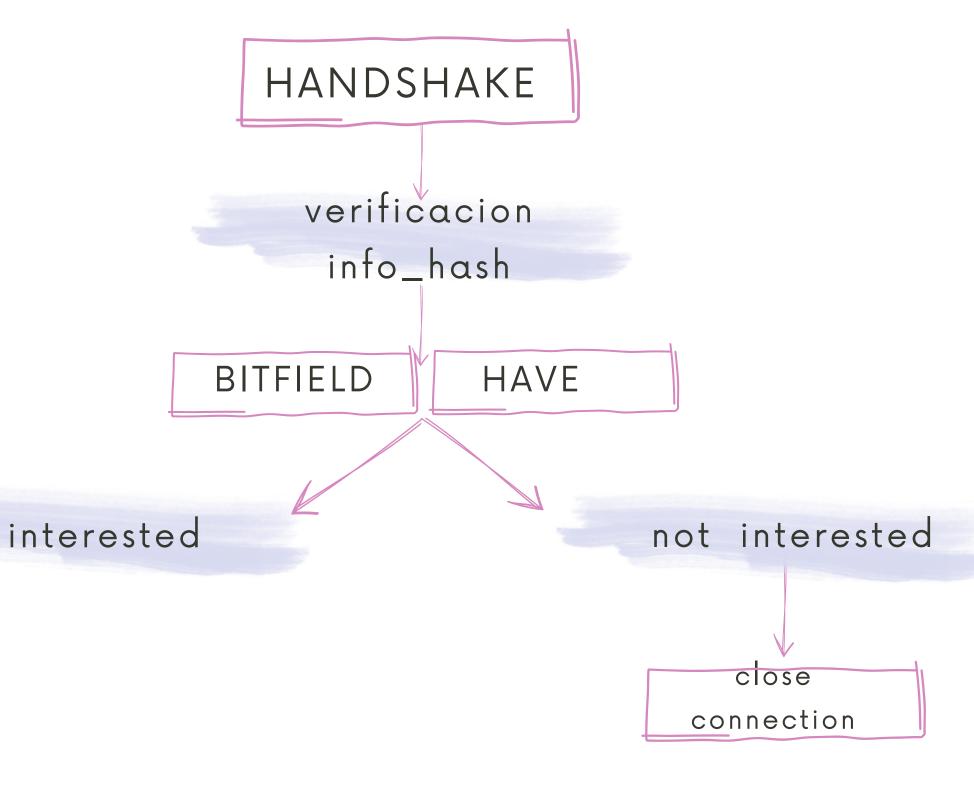


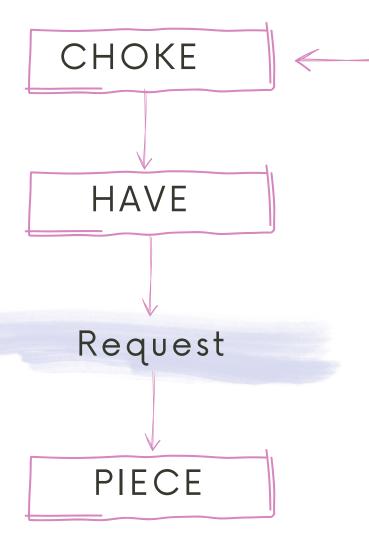




Ahora vamos a hablar sobre la implementación del modo servidor.

SERVIDOR FLUJO











Rebobinamos un poquito, y vemos la concurrencia utilizada en el proyecto.











BITTORRENT

CONCURRENCIA THREADS

O1 Logger

O2 Server

O3 Cliente





Último paso, interfaz

descarga.

gráfica. Es hora de ver la







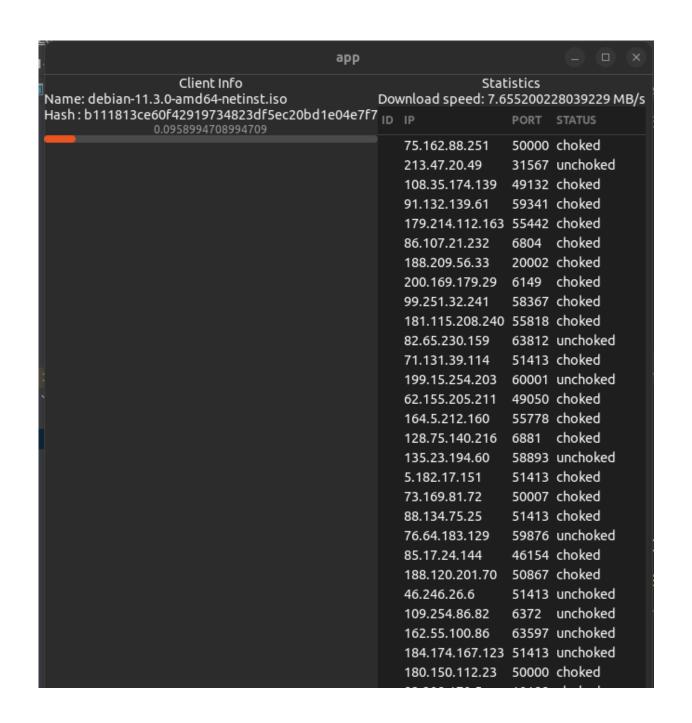








INTERFAZ DESCARGA DE UN TORRENT



GRACIAS POR LA ATENCION!

¿Preguntas?