**PRÁCTICA OPCIONAL:**

**DISEÑO DE UNA DHT. IMPLEMENTACIÓN BÁSICA DE UNA APLICACIÓN DE BITTORRENT**

Francisco Pinto Santos 70918455W

Héctor Sánchez San Blas 70901148Z

Índice

# Introducción

Para la realización de esta práctica se solicita realizar el diseño de un sistema para compartir datos mediante una dht. Puesto que se otorga liberta en el diseño y la realización de este, se decidió llevar a cabo una implementación más funcional mediante el desarrollo de una aplicación, es por ello que se ha realizado la implementación de una pequeña aplicación basada en bittorrent cuyo objetivo es el de poder compartir archivos mediante su subida a través de la creación de un .torrent del mismo y su indexación con dht. De esta forma, cualquier persona que posea dicho archivo .torrent puede descargarlo. Teniendo el objetivo claro, se estableció que un lenguaje de programación adecuado para realizar dicho trabajo era typescript junto al entorno en tiempo de ejecución nodejs. Para dar una mayor visualización a la aplicación se ha utilizado como lenguaje de frontend vue.js junto a la librería vuetify. Se utiliza el módulo express para la comunicación entre frontend y backend.

A continuación, se van a explicar que módulos se utilizan, el funcionamiento de la aplicación y la explicación del código.

# Proceso de búsqueda de módulos

Puesto que se usa nodejs, como entorno en tiempo de ejecución, y se busca desarrollar una aplicación más funcional, se llevó a cabo una búsqueda de módulos de node con los cuales desarrollar la propuesta, para ello, se buscaban aquellos módulos que supusiesen un apoyo en la realización de las funciones de dht junto a la creación y tratamiento de los .torrent. Se va a explicar el proceso de búsqueda y descarte realizado hasta llegar a los módulos que se creen idóneos y, por tanto, que se han utilizado:

* <https://github.com/jeanlauliac/kademlia-dht> : implementación en javascript para el protocolo Kademlia (entendido como el conjunto de algoritmos y estructuras de datos) en la implementación de DHT. Este módulo proporciona las funciones principales de Kademlia, como set y get, para almacenar clave/valor y recuperar clave/valor respectivamente, además, se trata de una implementación adaptable al navegador. Al considerar utilizar este módulo, se descartó debido a dos motivos:
  + No existe el manejo de comportamiento tales como expiración de los pares clave/valor, replicación y republicación de los pares.
  + Es una implementación menos práctica de lo buscado para la solución que se quiere desarrollar, puesto que deja en el aire algunas soluciones como la creación y tratamiento de los .torrent.
* <https://github.com/khaosdoctor/node-dht> : se trata de una implementación llevada a cabo en la UFABC de una simple DHT con unas reglas definidas establecidas con respecto a la teoría del funcionamiento de las DHT. A pesar de ser una implementación correcta de DHT fue descartada por motivos similares al de la implementación de Kademlia.
* <https://github.com/libp2p/js-libp2p> : módulo cuya finalidad es la de construcción de aplicaciones P2P que define interfaces que, una vez expuestas, permiten que otros protocolos y aplicaciones que las usen intercambien información, permitiendo su actualización y adaptabilidad en tiempo de ejecución, todo ello sin tocar nada relacionado con la API. En sí, es un módulo bastante completo para estas comunicaciones e implementaciones de P2P, sin embargo, encontramos un problema principal y es que, como se encuentra en una versión alfa, existen errores sin resolver que a penas están documentados y, de los cuales, no se otorga una solución clara, cosa que es un inconveniente para el desarrollo de nuestra aplicación, puesto que se busca algo funcional en un corto periodo de tiempo. Por todo esto se descartó.
* <https://github.com/webtorrent/webtorrent> : cliente de streaming Torrent para node.js y navegador, es decir, se trata de un cliente web. Usa TCP y UDP para la comunicación, en la web, se utiliza WebRTC para el transporte P2P. Otorga una ventaja respecto a que no es necesario la inclusión de plugins, extensiones o instalaciones en el navegador. Este módulo incluye todas aquellas funciones necesarias para la implementación del protocolo BitTorrent, sin embargo, se encontró que era demasiado completo y perdía sentido incluirlo en el proyecto puesto que se iba a reducir demasiado el trabajo que se debía llevar a cabo.

A pesar de que se habían descartado los módulos anteriores, respecto al último, se confirmó que utilizaba otro módulos para su implementación que, al consultarlos, se vieron oportunos algunos de ellos para el desarrollo del trabajo:

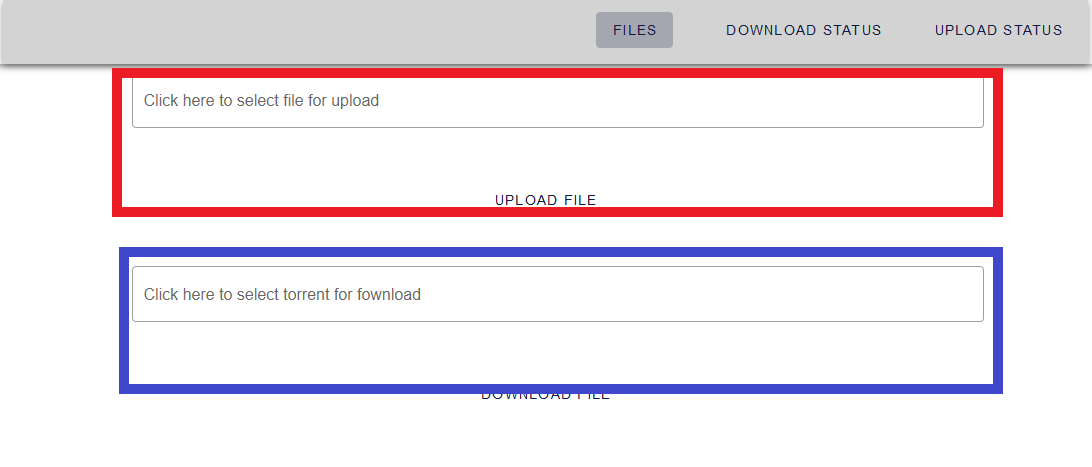
* <https://github.com/webtorrent/bittorrent-dht> : implementación del protocolo BitTorrent siguiendo las especificaciones establecidas para ello. Se trata de un módulo robusto, probado y documentado que otorga las funciones necesarias para llevar a cabo una implementación funcional siguiendo el protocolo BitTorrent. Se trata de un DHT público a la vez que se puede realizar la conexión entre peers de una misma red.
* <https://github.com/webtorrent/create-torrent> : módulo que otorga las funciones necesarias para la creación de un archivo .torrent a partir del archivo que se desea compartir. Se escoge debido a que facilita el proceso de desarrollo de la aplicación. El parseo de estos archivos se hace de forma manual puesto que en esta implementación no se utilizan trackers.

Con los módulos escogidos, se pudo proceder a la implementación funcional de la aplicación.

# Funcionamiento de la aplicación

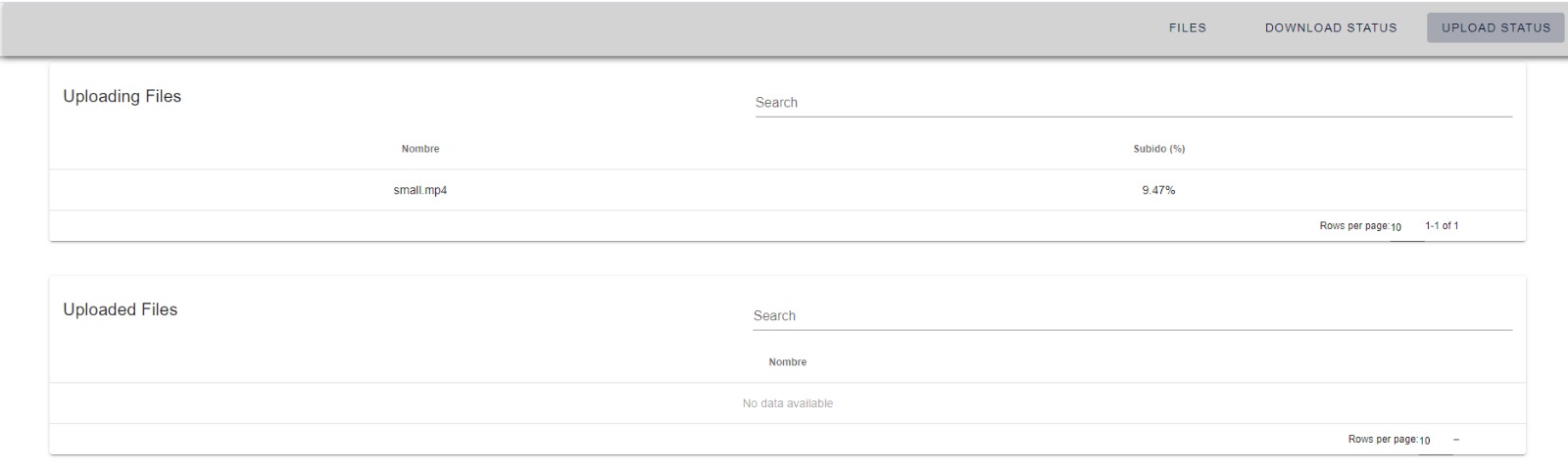
En este apartado, se explicará de forma visual, cual es el funcionamiento de la aplicación, para ello se explicarán las distintas ventanas y su contenido:

* **Ventana “Files”:** En esta ventana nos encontramos dos opciones, subir archivo descargar a partir de un archivo .torrent :

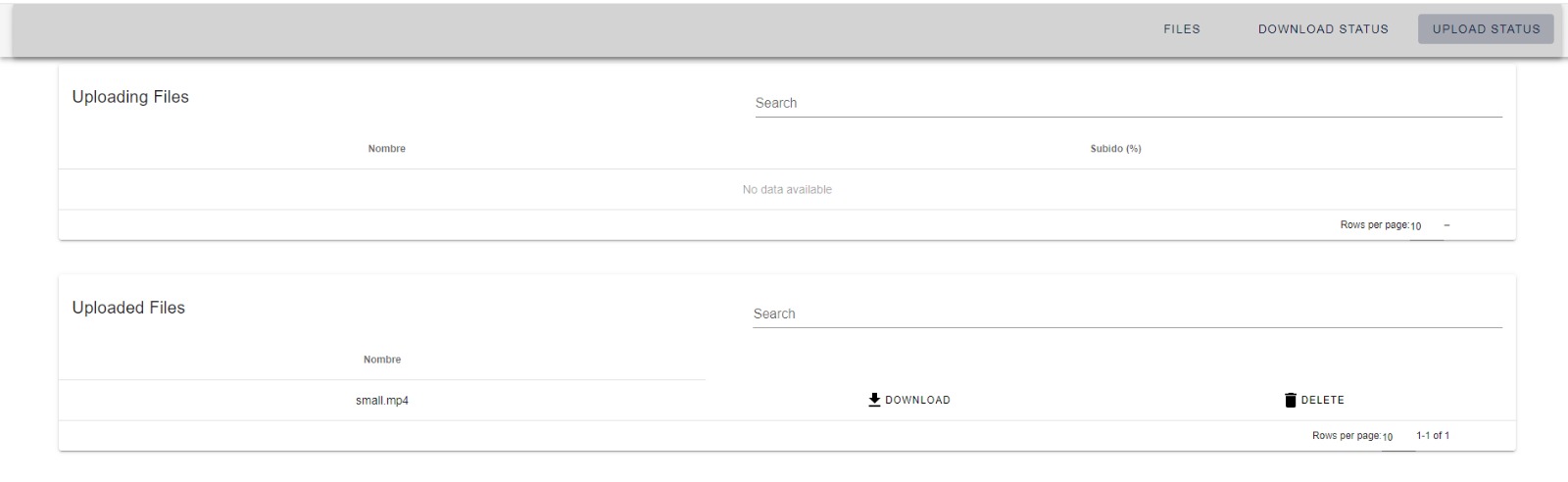


En rojo se encuentra la zona para seleccionar un archivo y subirlo. En azul la zona para seleccionar un archivo .torrent y comenzar la descarga de su respectivo archivo.

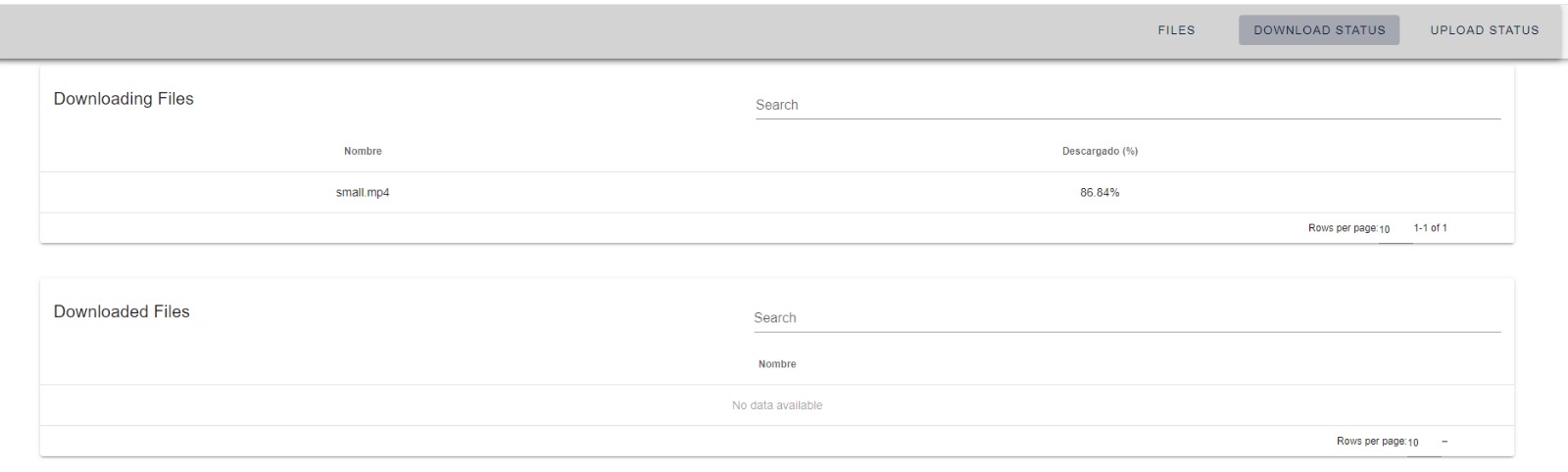
* **Ventana “Upload Status”:** en esta ventana se pueden observar dos tablas, la primera nos mostrará el estado de la subida de los archivos (porcentaje subido), es decir, el porcentaje de chunks que se han almacenado en el dht. Esto se puede observar en la siguiente imagen:



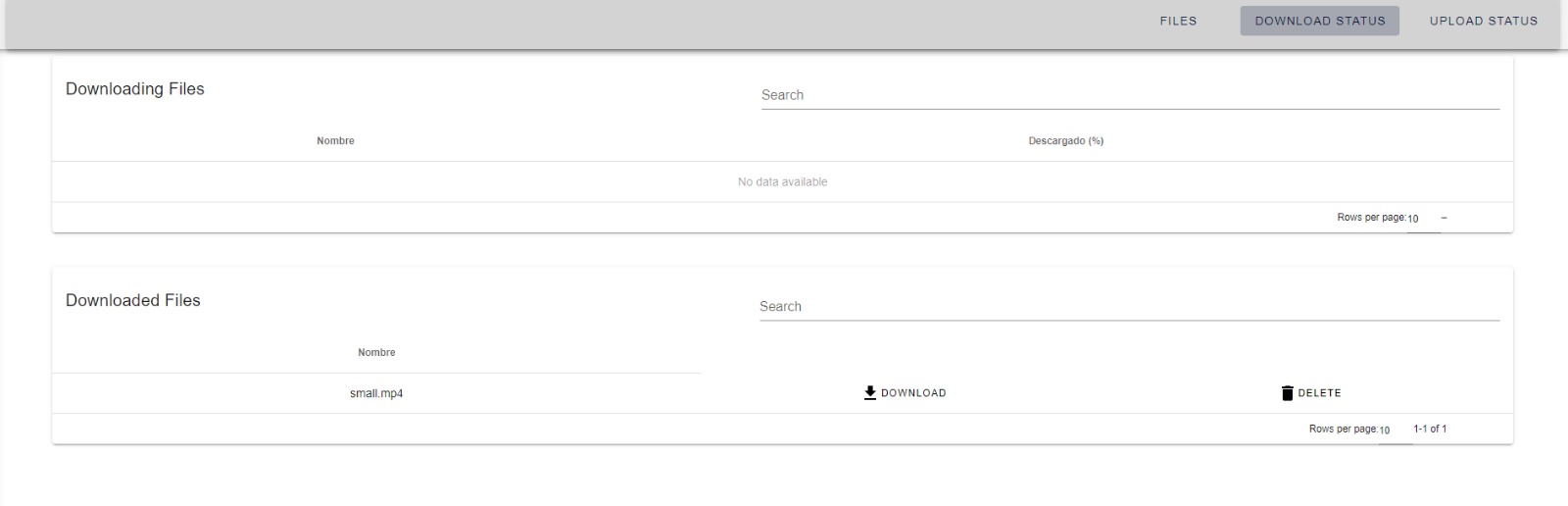
Cuando todos los chunks han sido almacenados, el archivo se mostrará en la tabla de archivos que se han subido completamente, de forma que nos dará la opción de obtener el archivo .torrent mediante el símbolo de descarga que aparecer al lado del nombre del archivo subido tal y como se puede observar a continuación:



* **Ventana “Download Status”:** ventana similar a la de Upload pero destinada a la obtención de archivos a partir del archivo .torrent. En esta ventana se pueden observar dos tablas, la primera nos mostrará el estado de la descarga de los archivos (porcentaje descargado), es decir, el porcentaje de chunks que se han obtenido del dht. Esto se puede observar en la siguiente imagen:



Cuando todos los chunks han sidoobtenidos, el archivo se mostrará en la tabla de archivos que se ha descargado completamente, de forma que nos dará la opción de obtener el archivo mediante el símbolo de descarga que aparecer al lado del nombre del archivo subido tal y como se puede observar a continuación:



Una vez mostradas las ventanas, se puede proceder a explicar cual es el funcionamiento de la aplicación. Cuando se desea compartir un archivo, se debe subir en mediante la opción mostrada en la pestaña “Files”, seleccionando el archivo a compartir y dándole al botón Upload. Una vez hecho esto, si nos trasladamos a la pestaña “Upload Status” se puede observar la evolución del porcentaje almacenado en la DHT. Cuando el porcentaje es 100, es decir, está completamente subido, el nombre del archivo aparecerá en la segunda tabla dándonos la opción del obtener el archivo .torrent generado. Se debe obtener de esta manera puesto que en los requisitos del trabajo se especifica que no debe de ser un proceso centralizado, por tanto, si alguien quiere obtener un archivo subido por otra persona, debe de haber recibido de otro participante el archivo .torrent generado al subirlo para compartir.

En lo referente al proceso de descarga u obtención del archivo, en la pestaña “Files”, en el apartado de “Download” debemos seleccionar el archivo .torrent del cual se desea obtener el archivo original. Una vez seleccionado y habiendo hecho click en el botón de “Download”, si accedemos a la pestaña “Download Status” nos encontraremos con algo similar a lo sucedido en “Upload Status” pero adaptado de la descarga del archivo, es decir, la obtención de la información almacenada en la DHT. Cuando el proceso termina, si se utiliza el botón de descarga obtendremos el archivo completo.

Cabe destacar que, en ambos casos (Upload y Download), cuando un archivo ha terminado de subirse o descargarse, respectivamente, después de descargar el archivo generado, se puede eliminar cuando el proceso ha terminado mediante el botón que así lo indica. Esto tiene como ventaja la eliminación dentro de la lista para evitar la acumulación de archivos dentro de la tabla, sin embargo, una vez eliminado, no se puede obtener el archivo relacionado con esa opción.

# Explicación del código