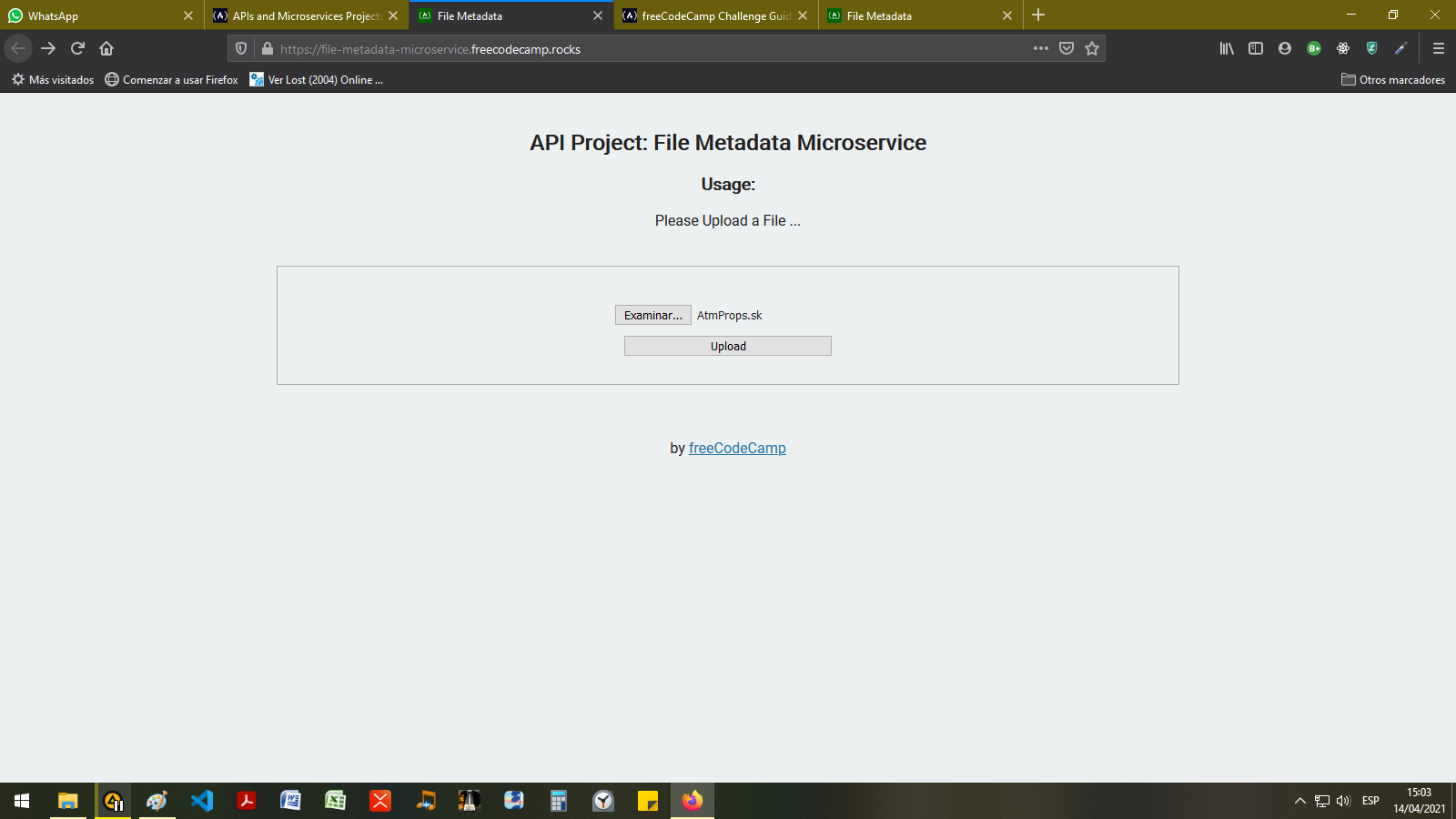
**Proyecto N°5 - File Metadata Microservice**

**Quinto proyecto del modulo APIs y Microservices de freeCodeCamp.**

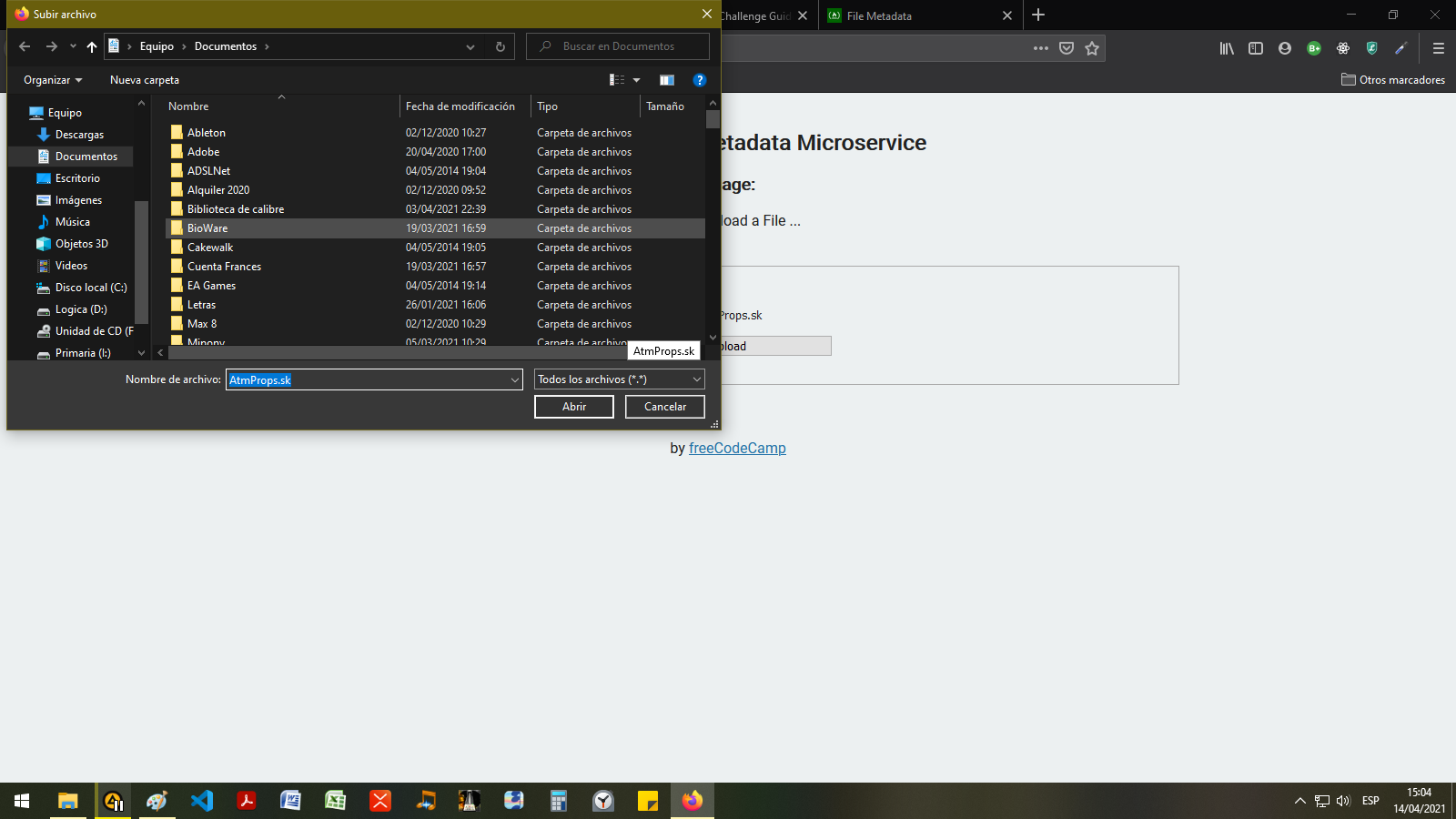
**Introducción:**

Tras el éxito de esta metodología, le daremos un segundo intento en este proyecto. Es el último de la tanda y a simple vista, parece simple.

**Descripción del Proyecto:**

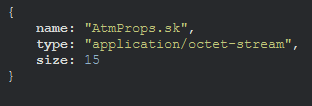
La página parece ser muy simple: un formulario donde tengo un botón que abre un cuadro de diálogo donde puedo elegir un archivo que se encuentra en mi computadora. 

Al hacer click en el botón, puedo elegir mi archivo



Tras elegir uno vuelvo a la pantalla anterior, con el nombre del archivo al lado del botón que abrió el cuadro de diálogo.

Luego, al hacer click en el botón UPLOAD, el servidor me devuelve un objeto json con 3 key-value pairs



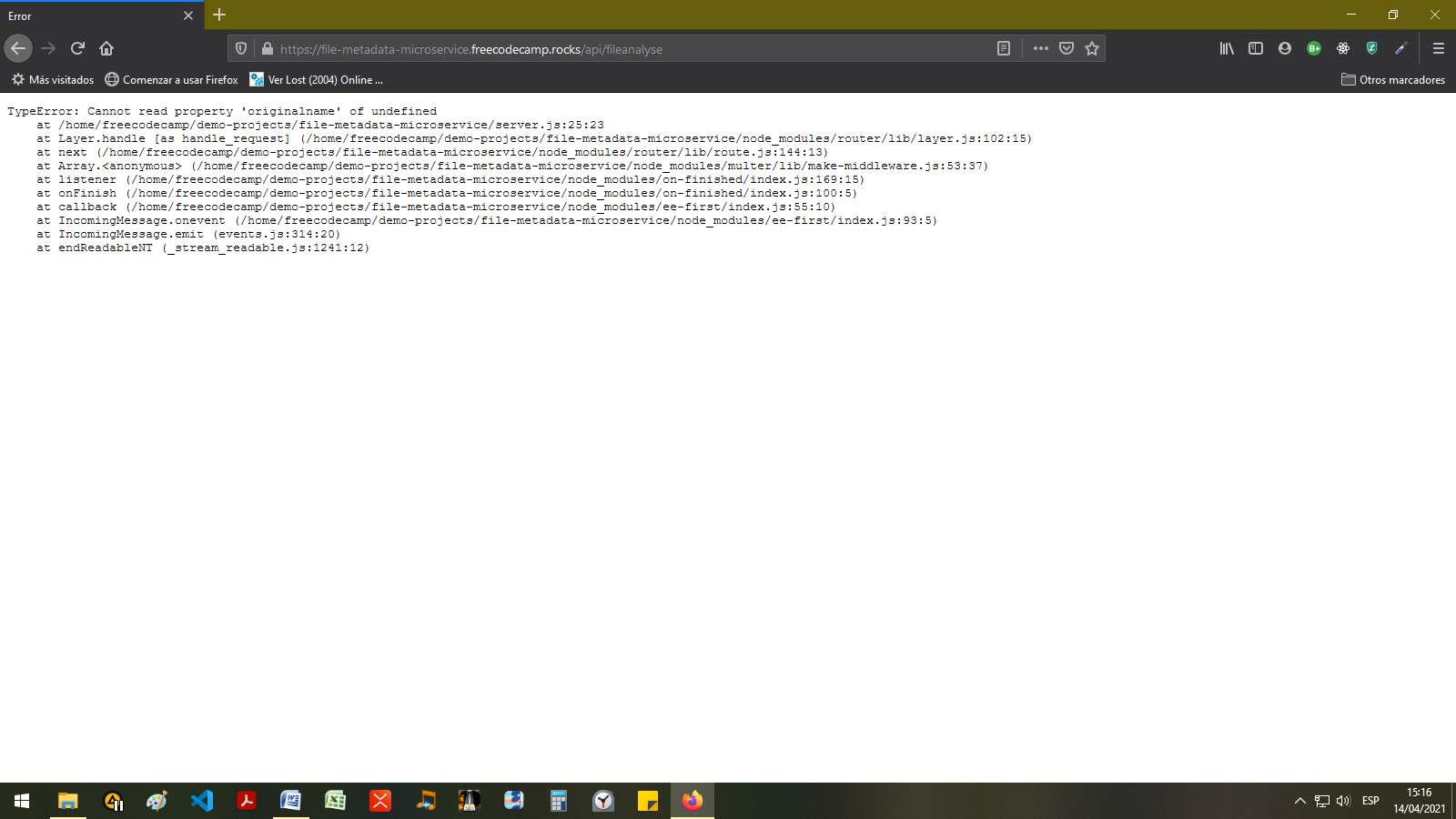
Eso es todo.

**USER STORIES**

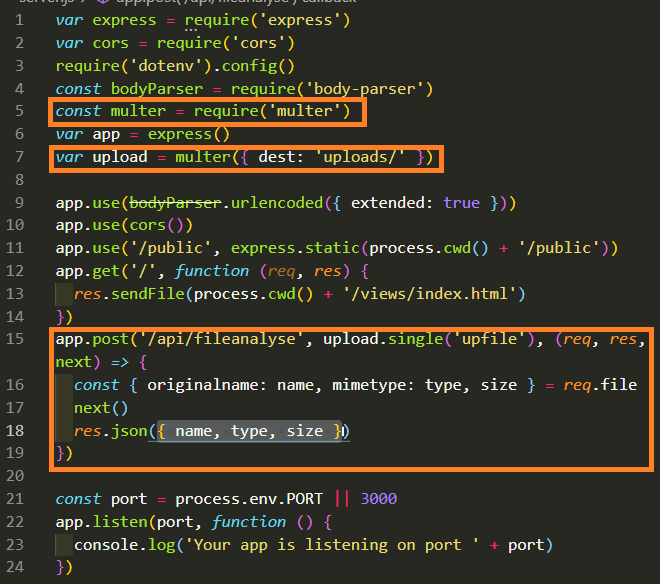
Los ‘user stories’ que explican que es lo que se pretende de dicha aplicación son:

1. You should provide your own project, not the example URL.
2. You can submit a form that includes a file upload.
3. The form file input field has the name attribute set to upfile.
4. When you submit a file, you receive the file name, type, and size in bytes within the JSON response.

**Observaciones:**

* Lo desafiante de este proyecto es que no sé cómo es que se puede subir un archivo al servidor para ser analizado. La página sugiere ‘*You can use the multer npm package to handle file uploading’.*
* Algo que me parece interesante destacar es que en este proyecto no parece ser necesario usar la base de datos.
* Si uno no sube un archivo y hace click en el SUBMIT, la página se rompe en mil pedazos: Así que no parece necesario lidiar con este error(una de las pocas variantes que se me ocurren) de manera elegante.

**Pasos a seguir:**

1. **Setup y boilerplate.**
2. **Crear la lógica del formulario:**
   1. Escribir el HTML.
   2. **Determinar cómo lidiar con la subida del archivo.**
      1. **Averiguar que es multer npm**
         1. Por lo visto multer es una forma de acceder a los archivos que sube un usuario en un servidor. Luego de la configuración inicial, se puede acceder a dicho archivo a través del objeto file en REQ.
            1. Configuración Inicial:

Hay que requerir multer, luego elegir donde se almacenarán los archivos subidos al servidor.

Además, al escribir el endpoint, notesé el uso del middleware a través de upload.single(‘upfile’) upload es el destino que elegí anteriormente para las subidas, single nos dice que es un único archivo o un único input. Upfile es el nombre de dicho input.

* + - * 1. La información del archivo se graba en un objeto file dentro del objeto req.