Departamento de Electrónica y Sistemas de Información

Forma, Círculo

Descripción generada automáticamente

Programación de Aplicaciones de Escritorio

Autores:

Luis Eduardo De la Mora Fernández

Ricardo Cuevas Rosas

Documentación de proyecto

Ingeniería en Sistemas Computacionales

4 de noviembre de 2022. Tlaquepaque, Jalisco.

Proyecto Infinilist

El proyecto consistirá en una aplicación web que permitirá a un usuario registrado crear listas donde se podrán cargar diversos tipos de archivos, notas, sitios web, ubicaciones, etc. Estas listas presentarán o desplegarán su contenido de manera correspondiente de acuerdo con su tipo de archivo. Estas listas podrán ser únicamente visibles a para el usuario o podrán ser públicas para compartirlas en cualquier red social de preferencia.

El concepto de la aplicación es un servicio con características de Google Drive y Google Notes

Para la realización de este proyecto se utilizarán las siguientes tecnologías:

* HTML
* CSS con SASS
* Angular
* Node Express
* TypeScript
* Google Maps API para despliegue de ubicaciones
* Google API para autenticación de usuarios/cuentas
* Servicios de AWS para almacenamiento de archivos
* Conexión a base de datos MongoDB

La aplicación contara con las siguientes operaciones CRUD:

* Creación y borrado de cuentas
* Creación y borrado de listas
* Añadir y borrar elementos de lista

Como módulos principales de la aplicación se contará con un área de administrador para tener acceso a todos los métodos CRUD completos antes mencionados  
  
Como área de usuario se contará con los siguientes módulos:

* Perfil
* Foto de perfil
* Listas Publicas
* El registro e inicio de sesión como secciones extra; será posible realizarlos con la cuenta de Google o en su defecto utilizando un correo electrónico válido.
* Cerrar sesión

Flujo para crear una lista, ya sea pública o privada  
   
Flujo para subir/crear un elemento a una lista

API Endpoints:

* Creación de cuenta: se le otorga el correo, nombre y contraseña y retorna un status 202 de que se pudo crear la cuenta correctamente, 400 de tener información incorrecta o 409 en caso de que la cuenta ya exista.
* Log in: se otorga el correo y la contraseña y retorna un status 200 en caso de que el log in sea exitoso o 400 si falló.
* Perfil: despliega la información del usuario y da la opción de editarlo o cerrar sesión en caso de ser el perfil propio.
* Colección de listas: una vez que se inició sesión galería y regresa las listas creadas o guardadas por el usuario.
* Página de la lista: despliega los elementos y archivos de la lista, de ser el usuario que la creó, ofrece opciones de edición.
* Búsqueda de listas: regresa las listas que coincidan con los criterios de la búsqueda.

Diagrama UML:

Diagrama

Descripción generada automáticamente

Conexión a la base de datos:

Para el almacenamiento de los usuarios y las listas se creó una base de datos en MongoDB, esta contiene una colección para ambos elementos del diagrama UML, las colecciones pueden ser editadas libremente desde nuestro servicio.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Documentación API:

La documentación de la API se hizo vía Swagger, si se accede a la ruta api-docs se pueden observar los endpoints que generamos y los schemas utilizados:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Patrón de fondo, Rectángulo

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

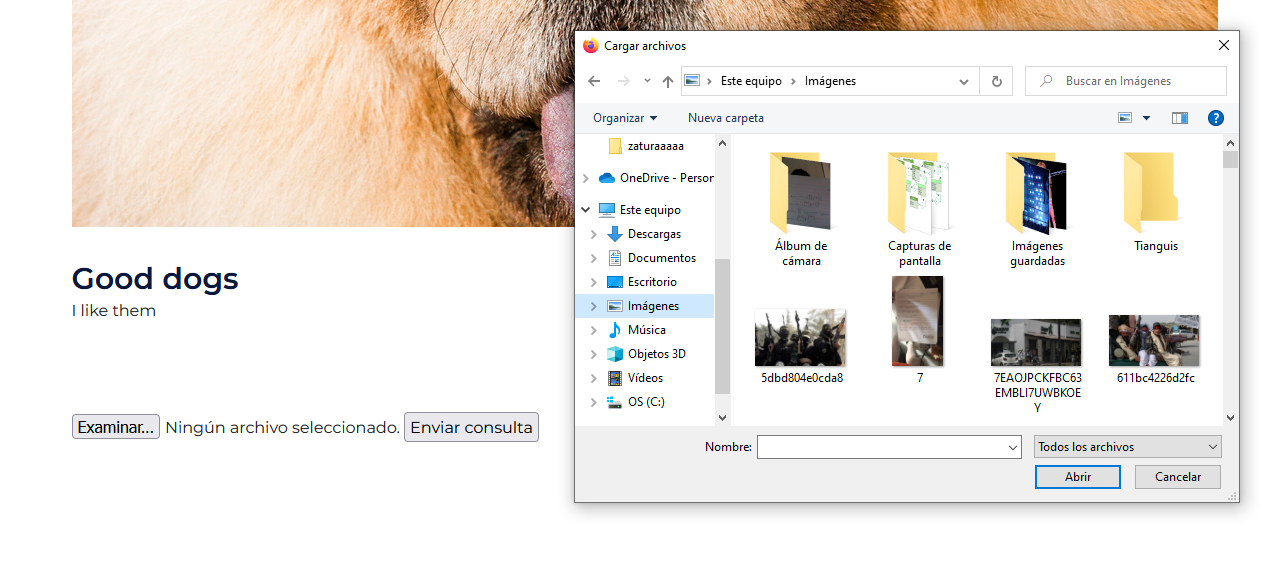
Descripción generada automáticamente

Multer y Socket.io:

Para el manejo de archivos y notificaciones de las sesiones de los usuarios se utilizaron Multer y Socket.io respectivamente. En el caso de multer, su implementación se encuentra endpoint de la una lista individual, donde el usuario puede cargar el archivo que quiera.

Texto

Descripción generada automáticamente



Repositorios de GitHub:

* <https://github.com/ReshCuevas/Infinilist-API>
* <https://github.com/ReshCuevas/Infinilist-FrontEnd>