 05 MARZO 2025

# MeshUp

Documentacion Técnica

PRESENTADO POR

Roberto Carlos Dominguez Espinosa #1994006

Ana Sofía Hernández Salazar #2008229

## PROPUESTA

MeshUp es una página web tipo red social donde nuestros usuarios podrán explorar el mundo del modelado 3d mediante las herramientas de nuestra plataforma ya sea compartiendo sus modelos propios o explorando los modelos de otros usuarios.

Nuestra página principal contara con un feed personalizado donde podrá encontrar sugerencias de modelos así como, filtrar este feed ya sea por categorías o a través de nuestra barra de búsqueda, donde podrá buscar modelos usando palabras clave.

El usuario podra crear publicaciones que deberán de llevar un nombre una corta descripción y el modelo con su textura, así mismo podrá editar algunos elementos como el nombre o la descripcion de esta publicacion si asi lo desea. Todas las publicaciones que el usuario generen podran ser accesadas dentro de la seccion de publicaciones en el perfil del usuario, donde ademas de poder visualizarlas para editar la publicacion estas podran ser eliminadas de manera permanente del sitio.

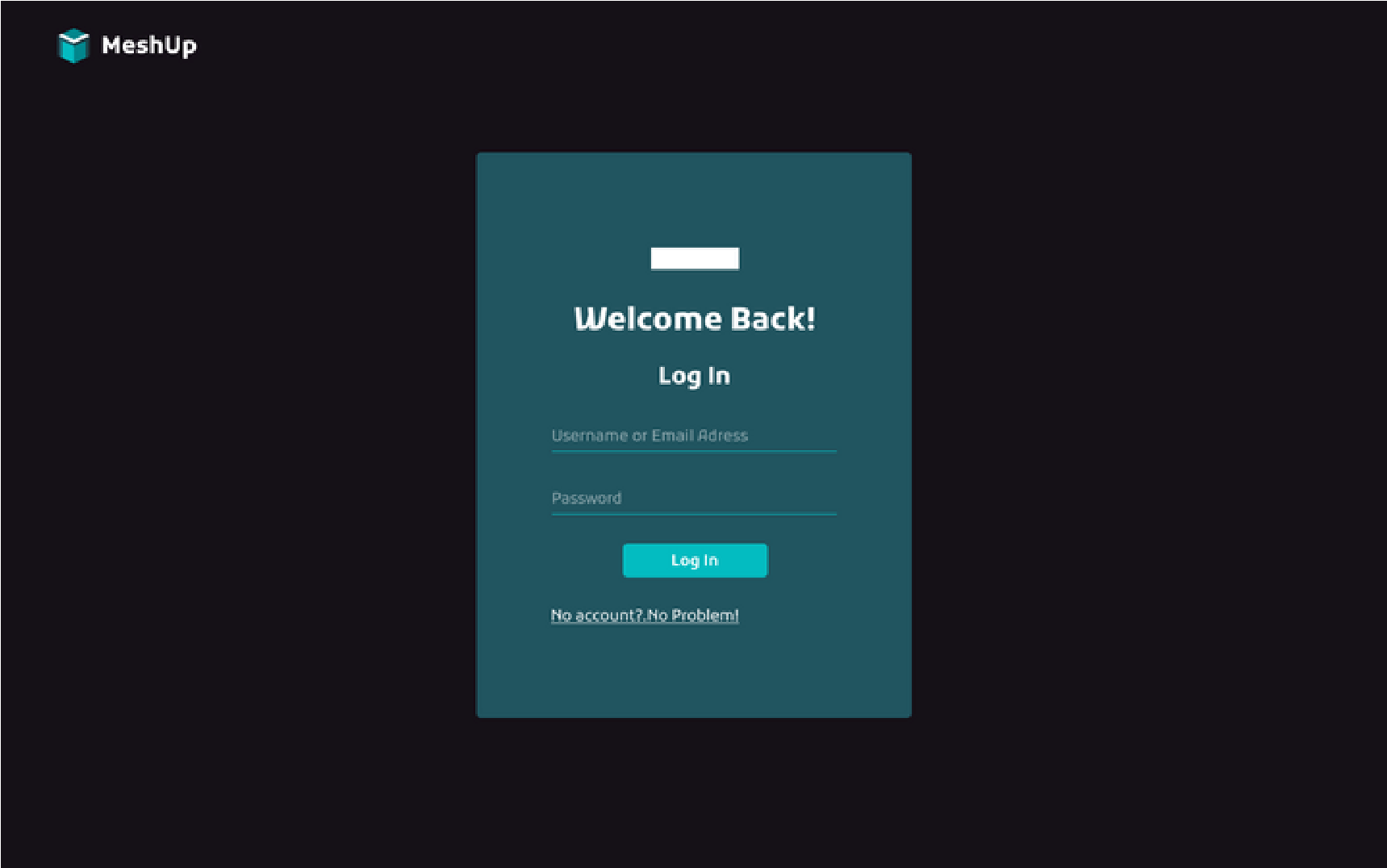
El usuario podra dejar likes y/o comentarios de tipo reseña dentro de la publicacion, asi mismo dentro de esta pagina principal el usuario podra guardar los modelos que mas le interesen en colecciones de distintos modelos ya sea propios o de otros usuarios. Estas colecciones deberan llevar un nombre y de ser deseado este podra ser cambiado dentro de las colecciones de los usuarios. Asi mismo si se desea se podran eliminar publicaciones de esta coleccion o eliminar la coleccion completa.

Los usuarios podran recibir noticias de cada una de las interacciones que otros usuarios tengan con sus publicaciones ya sea likes, comentarios o que otro usuario agrego su publicacion a una coleccion propia, estas interacciones se podran observar dentro de la seccion de notificaciones.

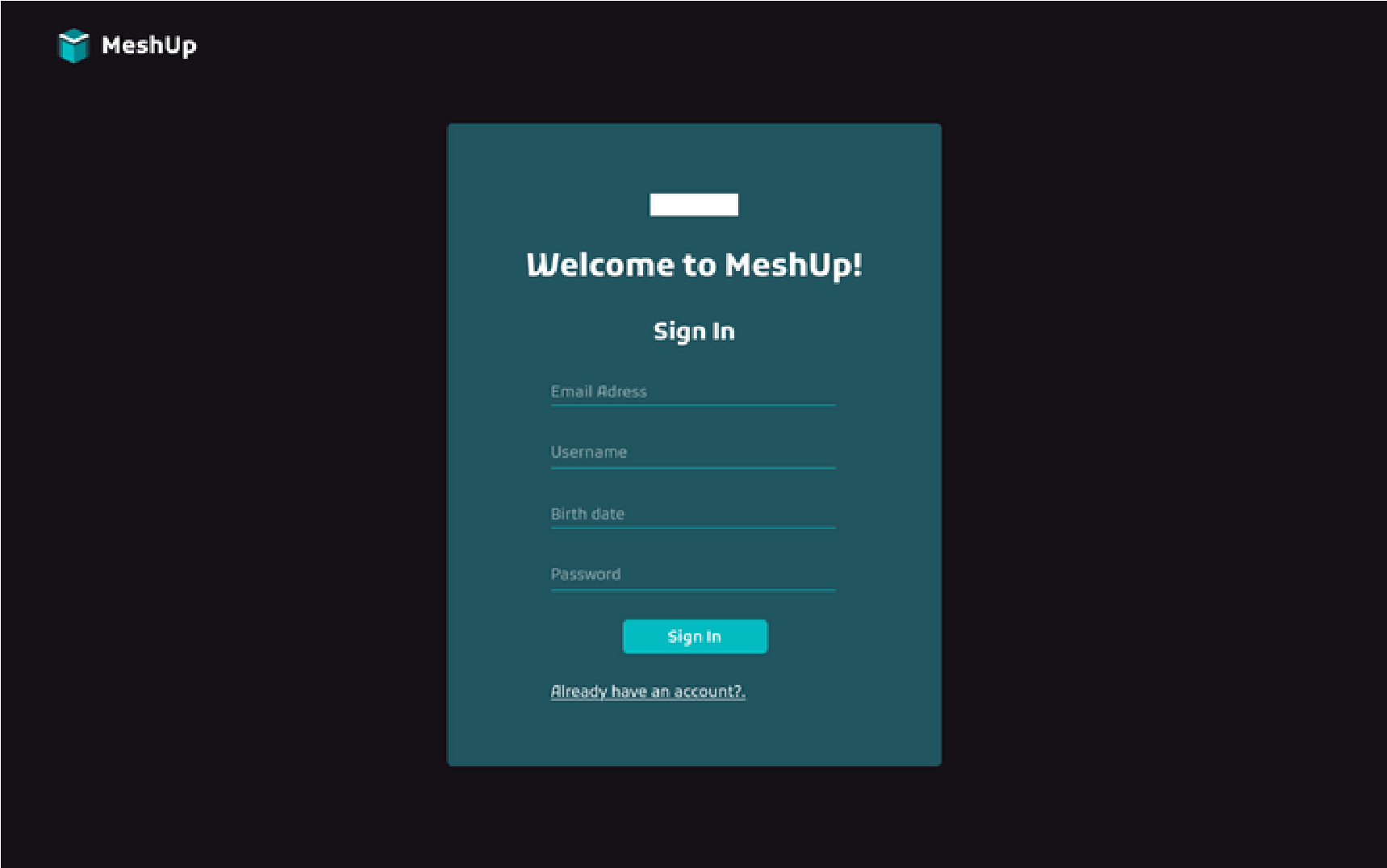
Ofreceremos una seccion de reportes donde el usuario podra disponer de todas las estadisticas de sus publicaciones ya sea por likes, comentarios o guardados, y asi mismo filtrar estos reportes por fechas, categorias o status de cada publicacion.

Por ultimo tendremos el historial de busqueda del usuario donde se mostraran todos los elementos buscados por dia y hora. Cada uno de estos elementos podra ser eliminado ya sea de manera individual como por grupos de varios o en su defecto eliminar todos los elementos de historial de manera permanente.

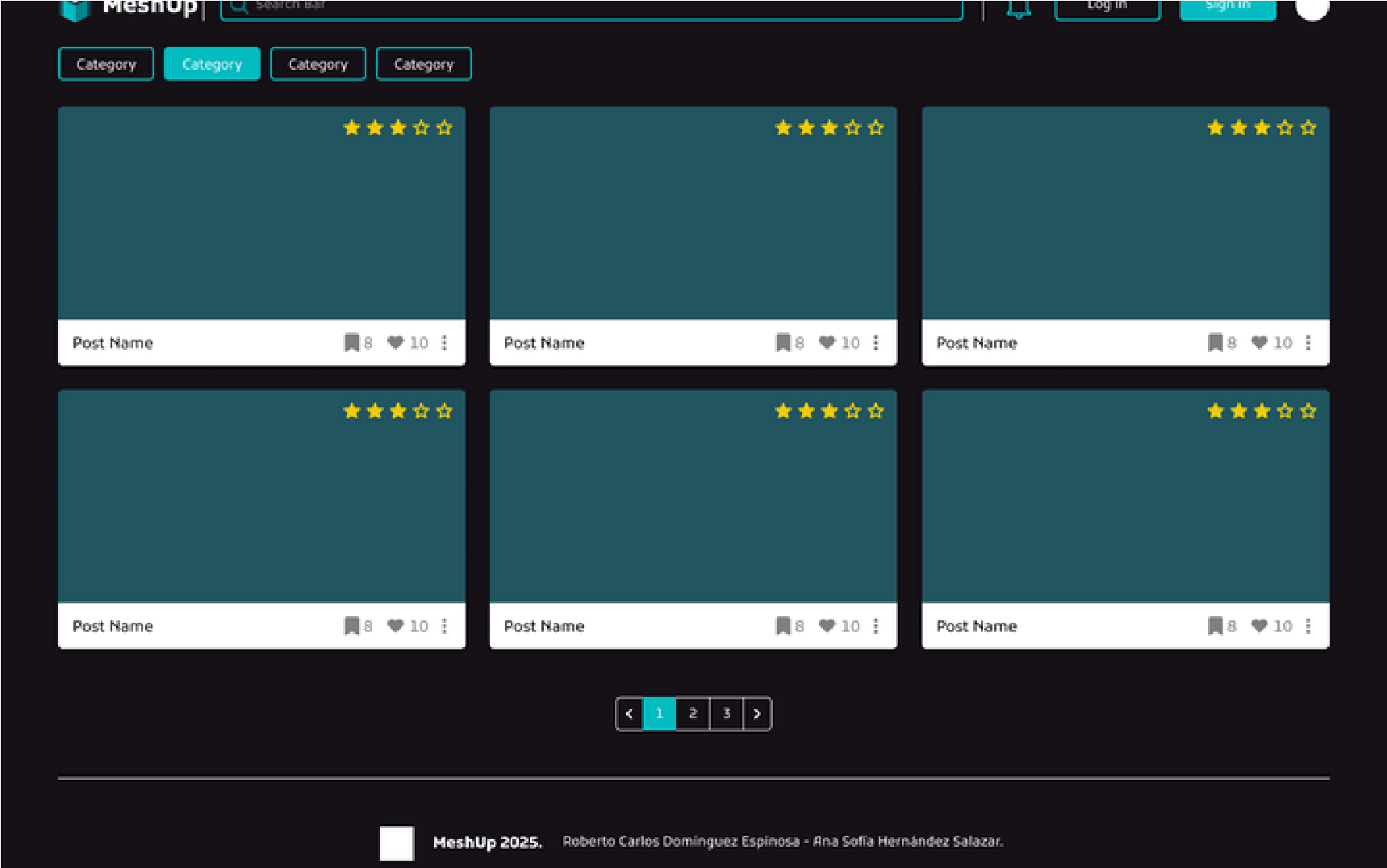
**Log In**



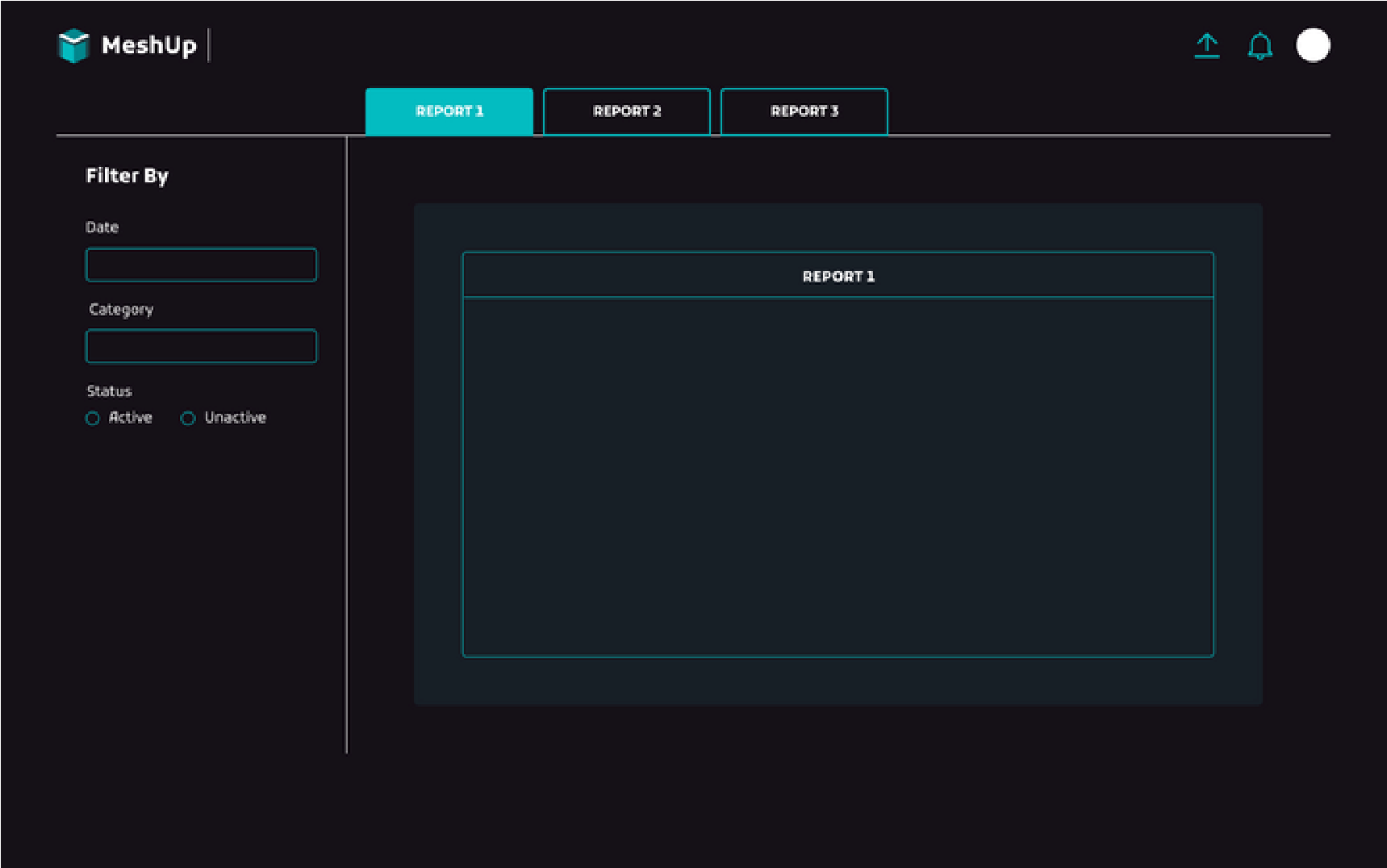
**Sign In**



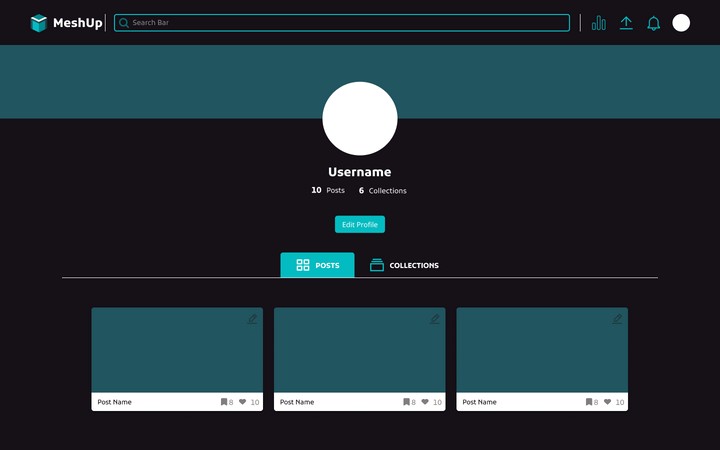
**Home**



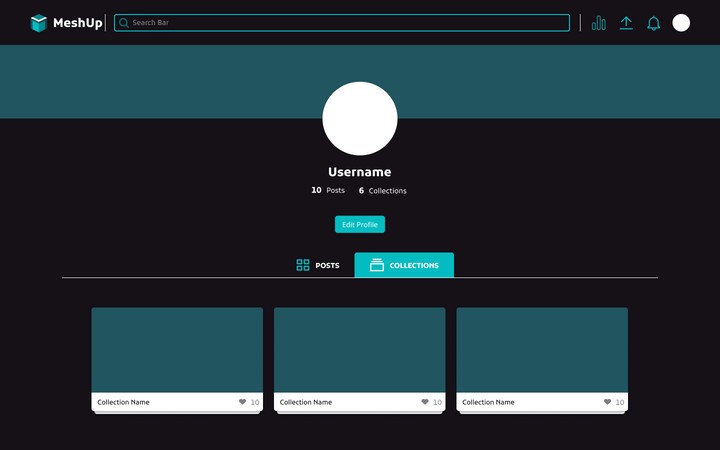
**Reportes**



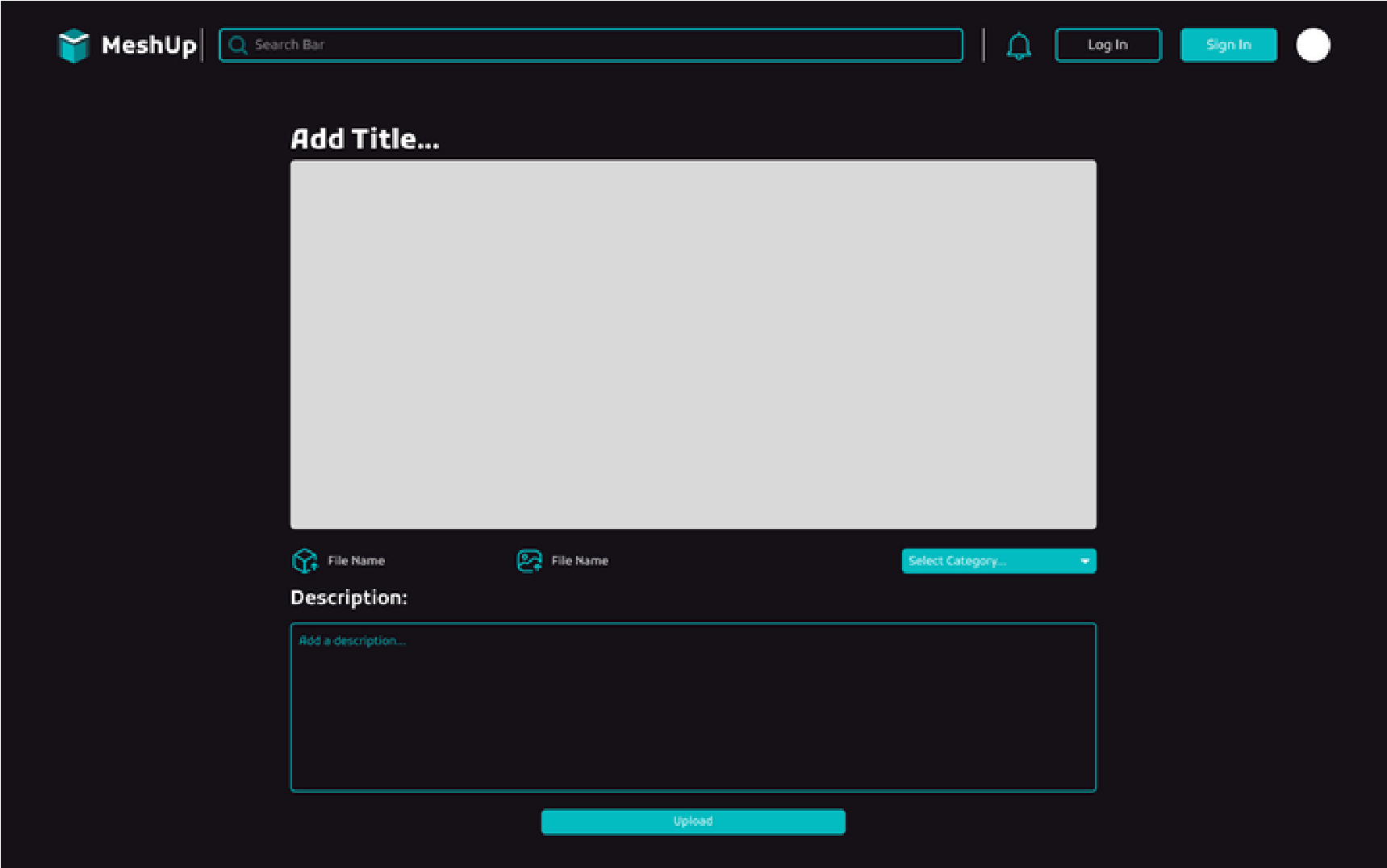
**Perfil (Posts)**



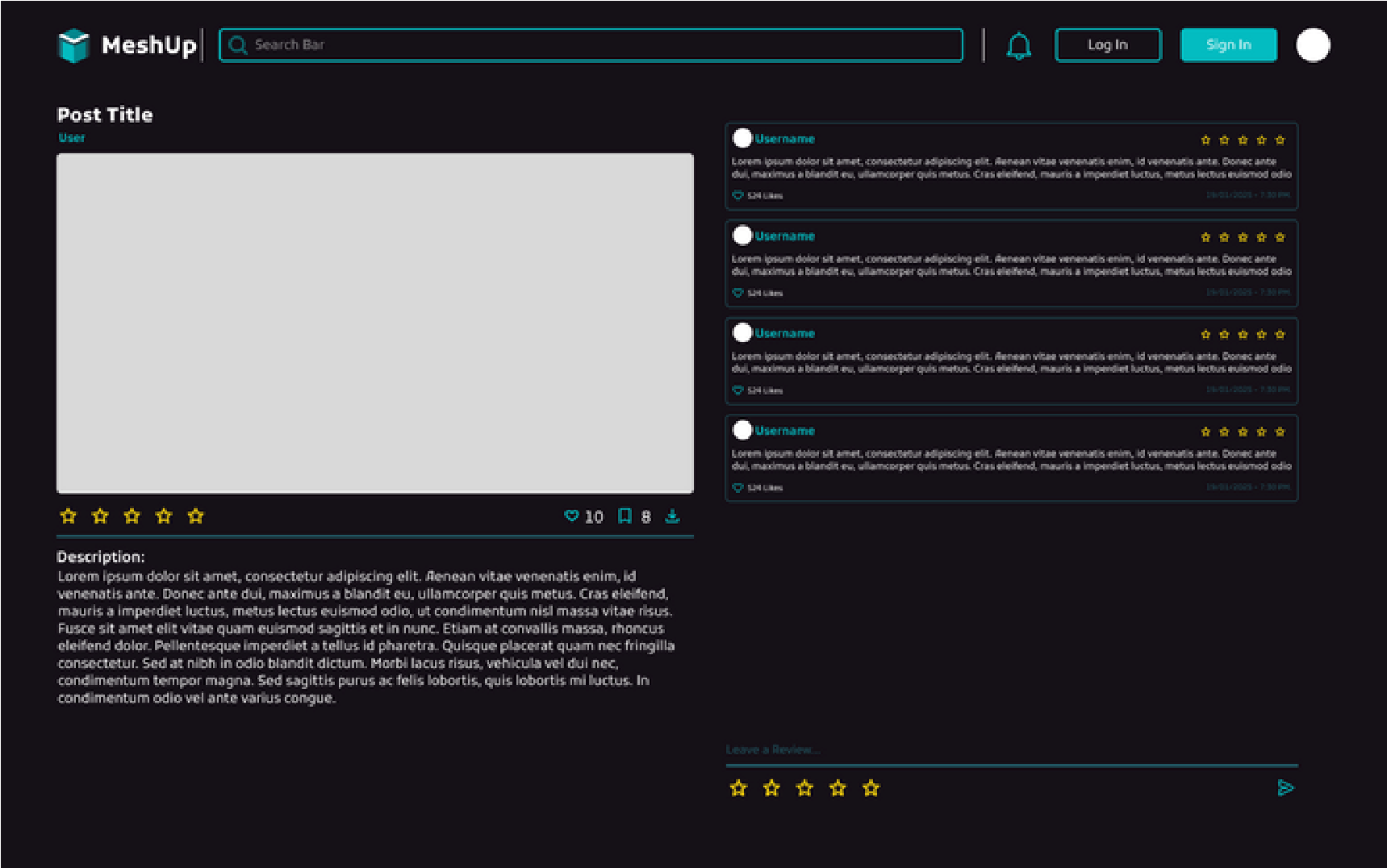
**Perfil (Colecciones)**



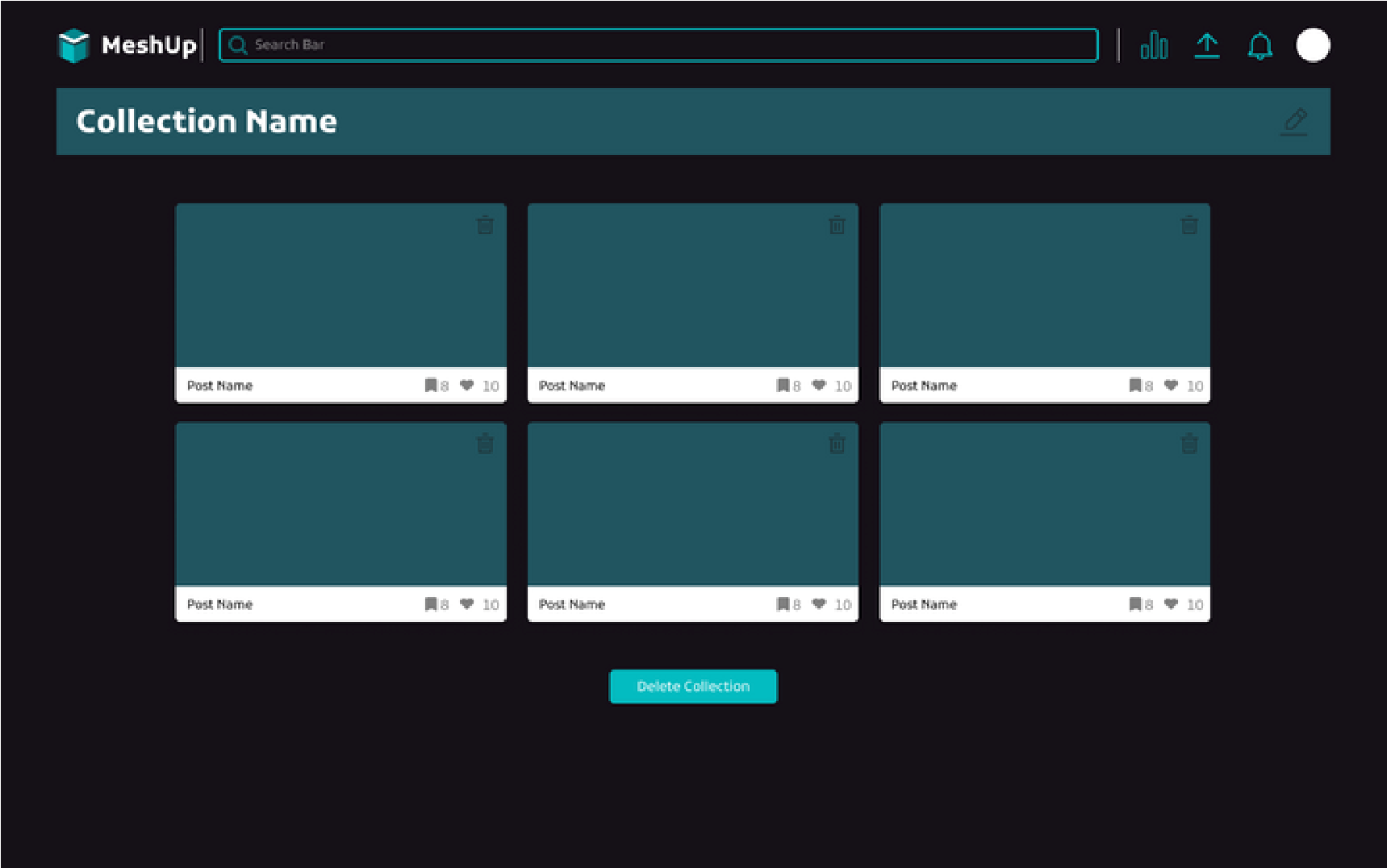
**Publicar**



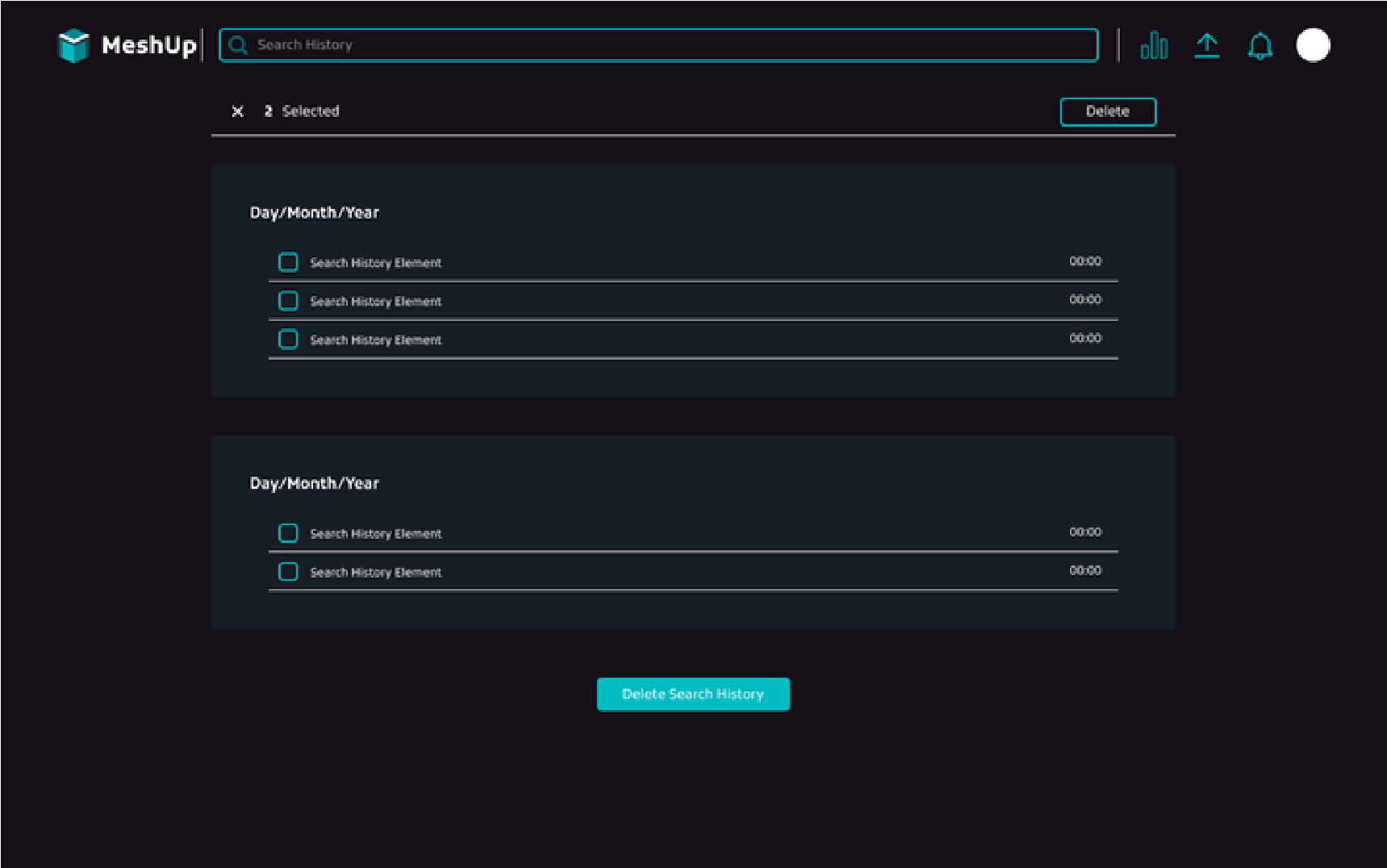
**Ver Publicacion**



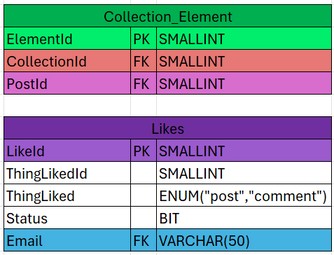
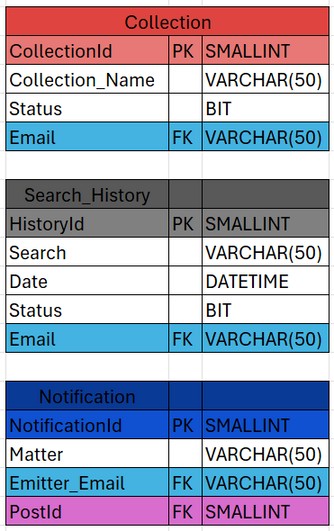
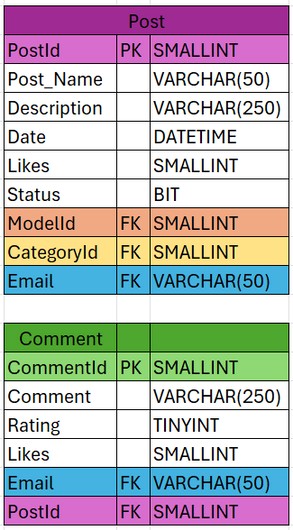
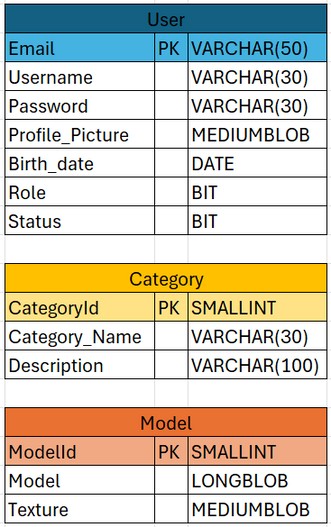
**Colleccion**



**Historial**



## DISEÑO DE BASE DE DATOS



## MODELOS DE TABLAS:

### User:



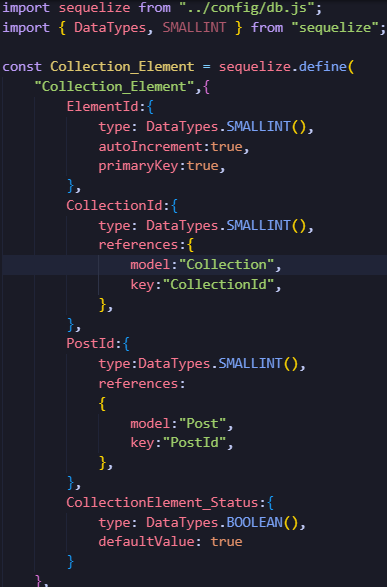
### Model:



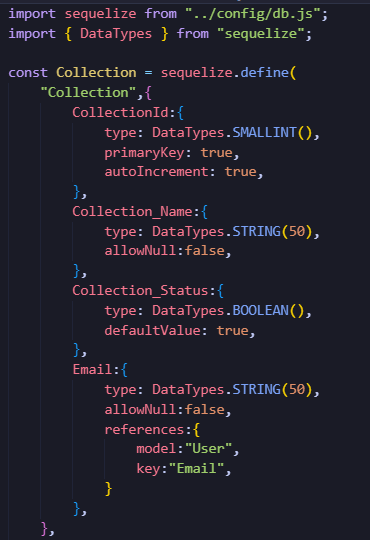
### Category:



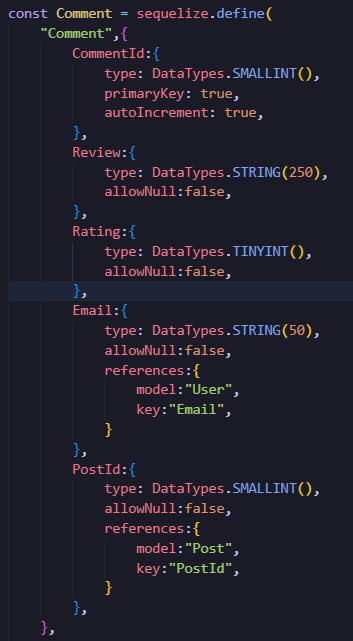
### CollectionElement:



### Collection:



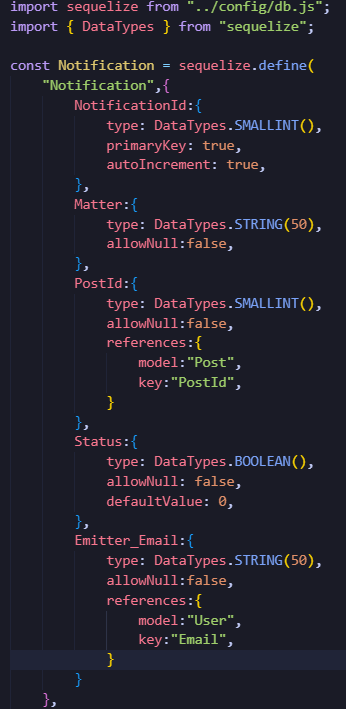
### Comment:



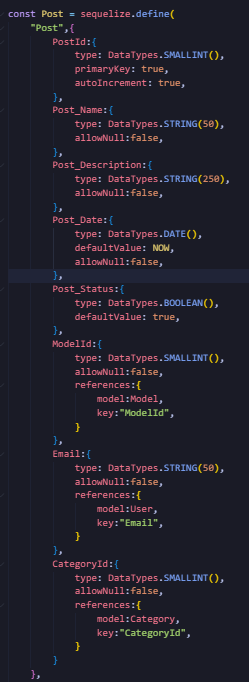
### Like:



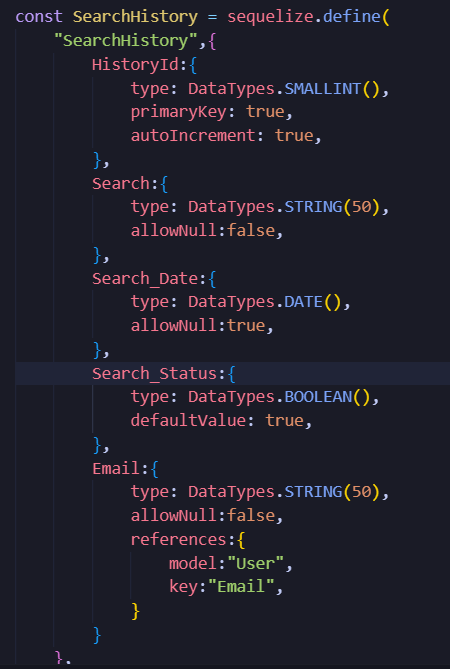
### Notification:



### Post:



### SearchHistory:



## Endpoints generados:

## Usuario:

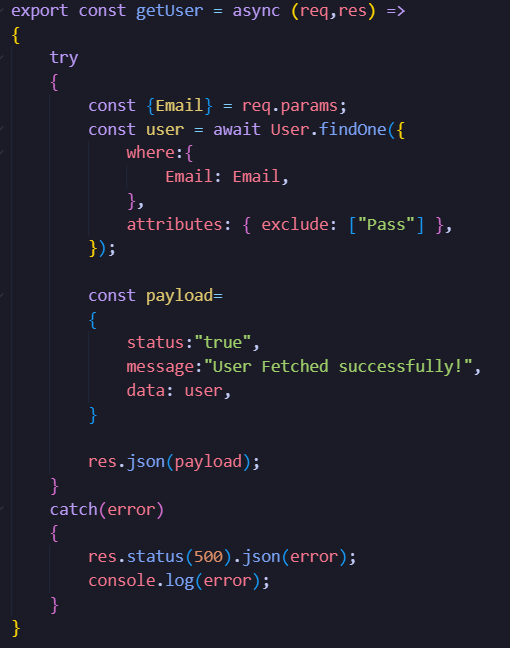
### Obtener todos los usuarios:

Función generada para obtener todos los usuarios registrados en la base de datos por medio de User.findAll(), retornando un Objeto donde están todos los usuarios encontrados.



### Obtener usuario en especifico:

Función generada para obtener la información en especifico de un usuario, necesaria para la renderización de un perfil en especifico, excluyendo la contraseña por cuestiones de seguridad.



### Login de Usuario:

Función generada para el LogIn de los usuarios donde específicamente obtenemos el único usuario encontrado en la base de datos por medio de User.findOne(), en el payload mandamos la información de manera dinámica por medio de terniarios.



### Registrar Usuario:

Función generada para registrar un usuario en la base de datos, antes de registrarlo de manera exitosa se realizan unas validaciones como si el usuario ya existe, y si es que ya existe ver si el Email o el Nombre del usuario es el que coincide con los parámetros recibidos, si es así se le retorna al cliente que ese dato ya existe, si no, genera el registro de manera exitosa.



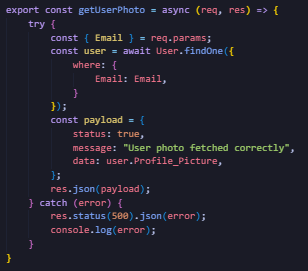
### Actualizar Usuario:

Función generada para actualizar usuario, las mismas validaciones que existen en registrar usuario también existen en actualizar usuario, también se registra la foto de perfil que se recibe como parámetro.



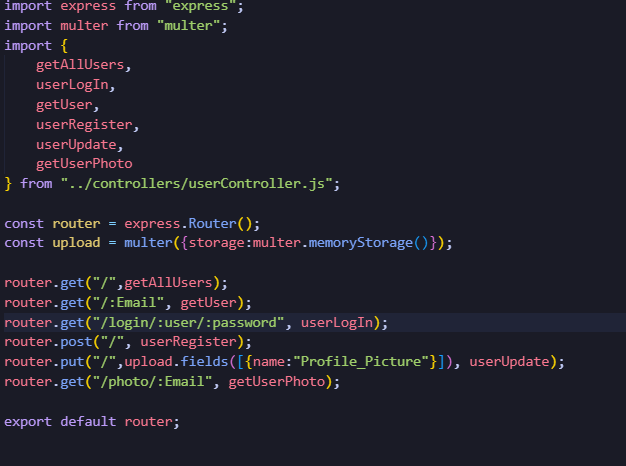
### Obtener Foto de Usuario:

Función generada para obtener la foto de algún usuario en especifico por medio de los parámetros de la solicitud.



### Rutas generadas:

Todas las rutas generadas en el controlador de Usuario con sus métodos correspondientes.



## REPORTES:

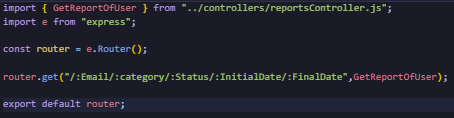
### Obtener Reporte de Usuario:

Función generada para obtener el reporte de todos los posts de un usuario en específico, el reporte se genera a base de los parámetros que le llegan por la solicitud.



### Ruta Generada:

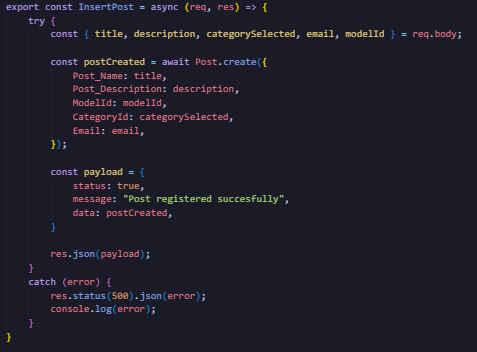
Ruta generada para el modelo Reportes.



## POST:

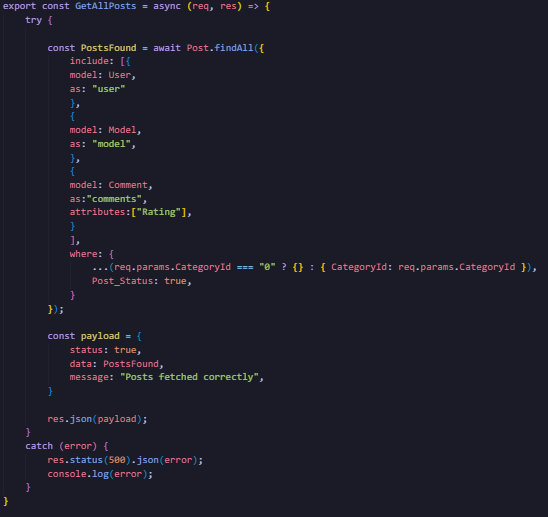
### Crear Post:

Función generada para la creación de un Post por medio de los parámetros del cuerpo.



### Obtener todos los Posts:

Función generada para obtener todos los posts en general de la página, filtrando únicamente si no son privados.



### Obtener posts de usuario:

Función generada para obtener todos los posts de un usuario en especifico, al momento de recibirlo en el frontEnd se filtran los privados o públicos



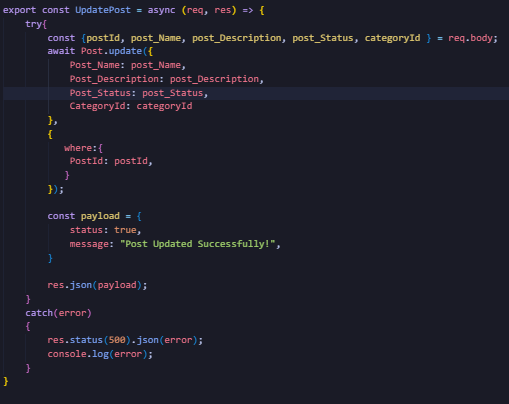
### Obtener post en especifico:

Función generada para obtener un post en especifico, incluyendo los comentarios del Post al igual que el modelo para la renderización.



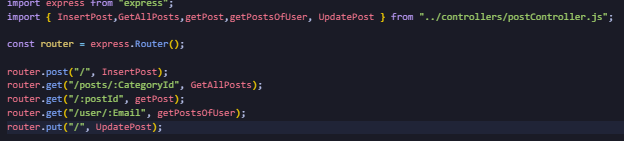
### Actualizar post:

Función generada para actualizar la información textual y la categoría de un post en especifico.



### Rutas Generadas:

Todas las rutas generadas en el modelo Post.



## Notificación:

### Obtener notificaciones de usuario:

Función generada para obtener todas las notificaciones no leídas por un usuario en especifico.



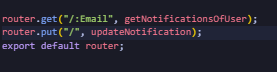
### Actualizar Notificación:

Actualizar el estado de una notificación si es que la llego a visualizar el usuario.



### Rutas generadas:

Rutas generadas para el modelo Notification.



## Like:

### Obtener likes:

Función generada para obtener todos los likes de un post o comentario en especifico, esto viene definido en la parte de los parámetros que le llega a la API, también obtenemos si el usuario en especifico le dio like a ese componente, esto para renderizar un status correcto en el lado del FrontEnd.



### Insertar like:

Función generada para insertar un like en un post o comentario, se registra el like y el usuario que fue quien le dio like.



### Actualizar Like:

Función generada para actualizar el estatus de un like en un post o comentario de un usuario en especifico.



### Rutas generadas:

Rutas generadas para el modelo Like.



## Historial:

### Insertar Historial:

Función generada para crear un registro en el historial al momento de hacer una búsqueda en la pagina, se obtienen los parámetros por medio del cuerpo de la petición y se retorna si la creación del registro fue exitoso o no.



### Obtener Historial:

Función generada para obtener el historial de un usuario en especifico y se retorna si es que existe algún registro o no.



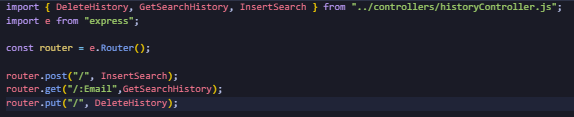
### Eliminar Historial:

Función generada para borrar elementos específicos de un historial, los elementos son los que le llegan por medio del cuerpo de la petición, desde el frontEnd se puede mandar a borrar todo el historial o solo algunos elementos en especifico.



### Rutas generadas:

Rutas generadas para el modelo SearchHistory



## Comentarios:

### Insertar Comentario:

Función generada para crear un comentario en las publicaciones, los parámetros son los recibidos en la petición.



### Obtener Comentarios:

Función generada para obtener todos los comentarios de un post en especifico



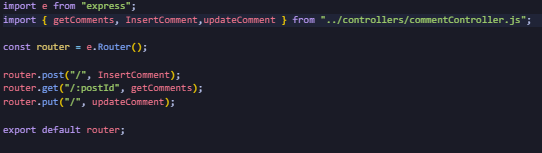
### Actualizar Comentario:

Función generada para actualizar el rating o review de un comentario en especifico, se toma en cuenta cual comentario es por el ID del Comentario mandado desde el frontEnd.



### Rutas generadas:

Rutas generadas para el modelo Comments



## Collecciones:

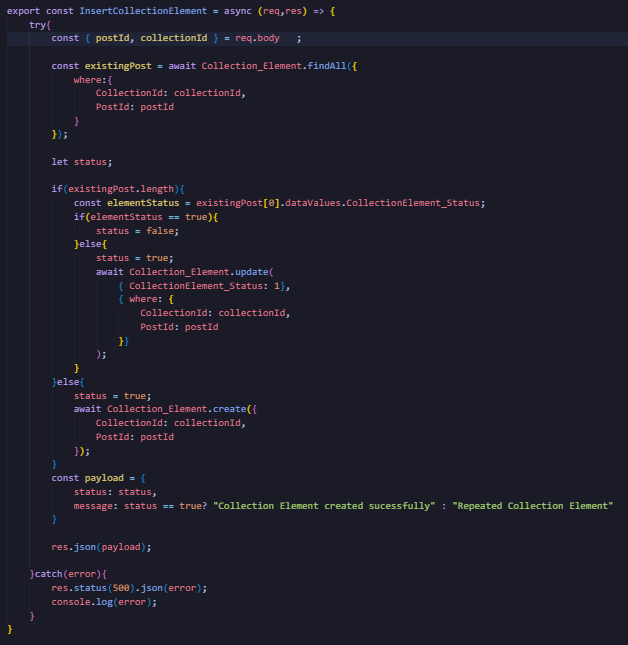
### Crear Colección:

Función generada para crear una colección a base de los parámetros que vienen de la petición.



### Insertar Elemento a Colección:

Función generada para insertar un nuevo elemento a una colección, antes de insertarlo se verifica si es que el elemento a insertar ya existe en la tabla o no, si es que ya existe se regresa como falso ya que el elemento ya existe en la colección, si no, se genera la creación del registro correctamente.



### Obtener Colecciones de Usuario

Función generada para obtener todas las colecciones de un usuario en especifico, su principal uso para el despliegue de todas las colecciones en el perfil del usuario.



### Obtener Elementos de Colección:

Función para obtener todos los elementos en especifico de una colección, al momento de hacer la consulta se toma en cuenta si están activos o no en la colección.



### Obtener Numero de guardados de post:

Función generada para obtener el numero de veces aparecido en la tabla de colección elemento para saber cuántos usuarios han guardado esa publicación en específico.



### Actualizar Colección:

Función para actualizar el nombre de una colección en especifico por medio de la ID de la colección mandada desde la petición.



### Eliminar Colección:

Función generada para eliminar una colección en general, cambiando su estatus de verdadero a falso.



### Eliminar elemento de colección:

Función generada para eliminar elementos de las colecciones en si, se identifican por el id de la colección y la publicación a borrar de esa colección.



### Rutas Generadas:

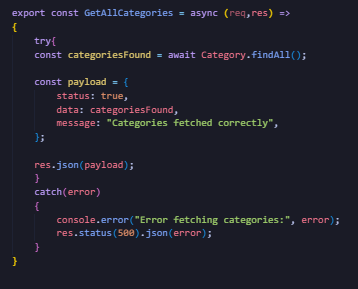
Todas las rutas generadas para el modelo Collection

### 

## Categorías:

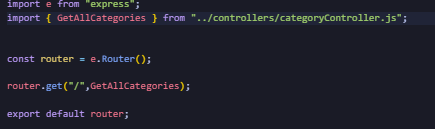
### Obtener todas las categorías:

Función generada para obtener todas las categorías que existan en nuestra base de datos para el despliegue en el dropdown al crear una publicación o al momento de estar usandola en los filtros.



### Rutas Generadas:

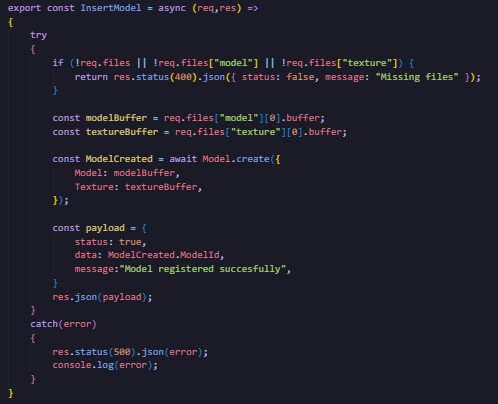
Todas las rutas generadas para el modelo Category



## Modelo 3D:

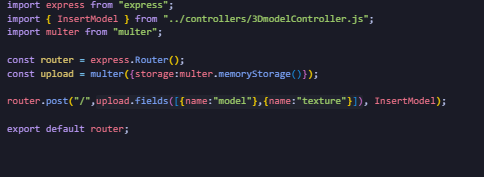
### Insertar Modelo 3D:

Función generada para insertar el archivo del modelo 3D al igual que la textura antes de crear el post, esta función regresa el ID del registro que acaba de crearse para el uso en la creación de un post.



### Rutas Generadas:

Rutas generadas para el modelo 3DModel



## Rutas de Todos los modelos:

Todas las rutas de todos los modelos tienen la ruta inicial que seria /api, esto para especificar diferenciar las rutas del FrontEnd con el BackEnd, para separarlas por modelos concatenamos a la ruta el modelo en especifico de cada controlador.

