# Recursos

* Colección de fotos: */fotos.*
* Foto individual: */fotos/ID*, siendo ID el identificador de la foto.
* Foto individual por ruta: */fotos/img/:ruta*

# Operaciones

Aspectos generales:

* En cualquier petición, en caso de error en el lado del servidor, se devolverá código HTTP 500 “Internal Server Error”.
* En cualquier acceso a un recurso individual, si éste no existe se devolverá código HTTP 404 “Not Found”.
* Las creaciones con éxito responderán HTTP 201 “Created” con la cabecera *Location* indicando la URI del recurso recién creado.
* Siempre que los datos proporcionados por el cliente estén incompletos o sean incorrectos, se devolverá HTTP 400 “Bad Request”.

## Búsqueda de Foto por ruta

Devuelve la foto que cumplan con la ruta indicada.

* Método: *GET.*
* Recurso: foto individual por ruta

Contenido de las respuestas posibles si todo va bien:

* + Cabeceras:
    - *Content-type: application/json.*
  + Cuerpo: array JSON con las fotos encontradas en el cuerpo de la respuesta. Si no hay ninguna a devolver, el array estará vacío.

## Búsqueda de Fotos

Devuelve las que cumplan con los criterios de búsqueda indicados. Método: *GET.*

* Recurso: colección de fotos.
* Parámetros en URL (todos opcionales):
  + *categories*: categorías de una foto, separadas por comas.

Contenido de las respuestas posibles si todo va bien:

* + Cabeceras:
    - *Content-type: application/json.*
  + Cuerpo: array JSON con las fotos encontradas en el cuerpo de la respuesta. Si no hay ninguna a devolver, el array estará vacío.

## Alta de una Foto

Se sube una foto nueva junto a su información en la base de datos, usando los datos que vengan en el cuerpo de la petición, que deben ser un objeto JSON.

* Método: *POST.*
* Recurso: colección de fotos.
* Cabeceras*:*
  + *Content-type: application/json.*
* Cuerpo: objeto JSON con los siguientes campos:
  + *Img*: Ruta del archivo. El cual debe estar en formato (.jpg y .png) con codificación “multipart/form-data” y tendrá un límite de tamaño.
  + *title:* Título de la foto.
  + *dateAndTime:* Fecha y hora en que se sube la foto (autogenerado).
  + *categories*: Array con las categorías a las que pertenece una foto.

Separadas por coma, mínimo una y pudiendo ser las siguientes:

* *Retrato*
* *Paisaje*
* *Urbano*
* *Macro*
* *Abstracto*
* *Ninguna*
  + *location*: Ubicación en la que se realizó la foto (No obligatorio).
  + *lens:* Lente utilizado. (“Canon 50mm 1.8","Canon EF 100mm F2/8","Canon EF135mm F/2","Sony DT 18-55mm f/3.5).
  + *camera:* Cámara utilizada ("Canon EOS 4000D","Canon EOS 5D Mark IV","Canon EOS 2000D","Sony Alpha A3000","Sony Evil Alpha 7 M3").
  + *description:* Texto que describe la foto (No obligatorio).
  + *photographer: Nombre del f*otógrafo que realizo la foto.

## Modificación Parcial de una Foto

Puntúa una foto existente en la base de datos. La puntuación debe venir en el cuerpo como objeto JSON.

* Método: *PATCH.*
* Recurso: foto individual.
* Cabeceras*:*
  + *Content-type: application/json*
* Cuerpo: objeto JSON con los siguientes campos:
* *score*: puntuación que se desea dar a la foto, debiendo ser un valor entero de 1 a 5.

## Borrado de una Foto

Elimina una foto junto a su información existente de la base de datos.

* Método: *DELETE.*
* Recurso: foto individual.