* **Recursos**
* Colección de perros : /dogs.
* Perro individual : /dogs/id
* **Operaciones**
* Peticiones correctas se devolverá un código HTTP(con su respectivo código) un mensaje descriptivo en la línea del estado de la respuesta
* Error en parte del servidor HTTP 500 mensaje descriptivo en la línea del estado de la respuesta
* **Búsqueda de perros**

Devuelve los perros según los criterios de búsqueda indicados (filtros de búsqueda) los filtros de búsqueda son opcionales en caso de no especificarle filtros devolverá por defecto todos los perros existentes;

* Método: *GET.*
* Recurso: colección de dogs.
* Parámetros en URL (todos opcionales):
* *raza*: tipo de raza del perro.
* *Pedigrí : verdadero o falso*
* *Popularidad : numero de votos de un perro (podrían ser Mas populares , menos populares o un gte de un determinador numero de votos)*

Respuestas posibles:

* 200 (“OK”) si la petición se procesa con éxito:
* Cabeceras:
* *Content-type: application/json.*
* Cuerpo: array JSON con los vinos encontrados en el cuerpo de la respuesta.
* 404 (“NOT FOUND”) si no hay ningun perrro en la BD o si no se encuentran datos según los filtros
* **Búsqueda de perro**

Devuelve un perro de un recurso individual ya sea por solamente por id

o por id + clave secreta de un perro que es usada para poder editar eliminar

* Método: *GET.*
* Recurso: recurso individual.
* Parámetros en URL (id obligatorio (para mostrar /read),(para eliminar y editar es obligatorio el id mas la clave secreta) ):
* *id*: id del perro.
* *clave : clave secreta*

Respuestas posibles:

* 200 (“OK”) si la petición se procesa con éxito:
* Cabeceras:
* *Content-type: application/json.*
* Cuerpo: array JSON con los vinos encontrados en el cuerpo de la respuesta.
* 404 (“NOT FOUND”) si no hay ningun perrro en la BD o si no se encuentran datos según los filtros
* **Alta de un perro**

Crea un perro nuevo en la base de datos, usando los datos que vengan en el cuerpo de la petición (el post se hace mediante un formulacio html method="post" enctype="multipart/form-data" action="/upload

)

* Método: *POST.*
* Recurso: colección de vinos.
* Cabeceras*:*
* *Content-type: application/json.*
* Cuerpo: objeto JSON con los siguientes campos (todos obligatorios )
* *name*: nombre del perro.
* *Raza* : raza del perro
* *Pedigrí* : verdadero o falso
* *Imágen : recibido sobre un modulo (multer)*

*1: direccion de almacenamiento :* public/images-clientes

2: como es almacenado : fechaDeSubida+nombrefichero+extencion

* *Descripción* : descripción del perro.
* Fecha\_nacimiento : tipo date
* *Edad* : campo calculado apartir de la fecha de nacimiento.

Respuestas posibles:

* 201 (“Created”) si la petición se procesa con éxito:
* Cabeceras:
* *Location: /dogs/ID,* siendo *ID* el identificador del perro creado.
* Redirigidas al index
* 400 (“Bad Request”) si el cuerpo de la petición no lleva los datos requeridos (faltan campos, o algún campo no tiene un valor aceptable, etc.)
* **Modificación de popularidad de un perro**

Da like o dislike a un perro existente en la base de datos . El valor de like o dislike deberá venir en el cuerpo de un objeto JSON como un entero (1 : sumara un punto a su popularidad 2 : restara un punto a la popularidad del perro).

* Método: *PATCH.*
* Recurso: perro individual.
* Cabeceras*:*
* *Content-type: application/json*
* Cuerpo: objeto JSON con los siguientes campos:
* popularidad : el valor será un campo entero de dos opciones (1:Like/2:Dislike)

Respuestas posibles:

* 204 (“No Content”) si la petición se procesa con éxito.
* 404 (“Not Found”) si no existe el vino indicado.
* 400 (“Bad Request”) si el cuerpo de la petición no lleva el campo requerido con un valor aceptable.
* **Modificación de general de un perro**

Los datos de un perro en concreto podría modificarse uno o varios campos.

* Método: *PATCH.*
* Recurso: perro individual.
* Cabeceras*:*
* *Content-type: application/json*
* Cuerpo: objeto JSON con los siguientes campos opcionales, pero como mínimo un campo:
* *name*: nombre del perro.
* *Descripción* : descripción del perro.
* *pedrigi : verdadero o false*
* *Raza : tipo de raza del perro (string)*

Respuestas posibles:

* 204 (“No Content”) si la petición se procesa con éxito.
* 404 (“Not Found”) si no existe el vino indicado.
* 400 (“Bad Request”) si el cuerpo de la petición no lleva el campo requerido con un valor aceptable.
* **Borrado de un vino**
* Método: *DELETE.*
* Recurso: perro individual.

Respuestas posibles:

* 204 (“No Content”) si la petición se procesa con éxito.
* 404 (“Not Found”) si no existe el perro indicado.