# Documento Problemas y Mejoras Api Tunubi

Lista de problemas encontrados

1. Función “addPoll”: Esta función es la encargada de agregar preguntas, y recibe un formato JSON, que debe ser enviado por método POST, pero la función solo acepta por defecto método GET que no es el apropiado para manejar esta petición en particular.

Se propone en la función “addPoll” en el decorador permitir que el método aceptado sea de tipo POST

@app.route('/addPoll',methods=['POST'])

1. Función “addAnswer”: Como en el caso anterior la respuesta que se envía también recibe un formato JSON, que debe ser enviado por método POST, pero la función que acepta es por defecto GET.

Se propone en la función “addAnswer” en el decorador permitir que el método aceptado sea de tipo POST

@app.route('/addAnswer',methods=["POST"])

1. Tanto en la función “addPoll” como “addAnswer” al esperar una petición JSON, debería validarse que el request recibido tenga un formato JSON, para en el caso que no sea correcto el formato, devolver algún mensaje de error indicando el JSON mal formado, y de esta forma evitar errores posteriores por asumir que la petición tiene el formato JSON correcto.

Se propone usar la función “request.is\_json” para validar el correcto formato de la petición

1. En el código se definió guardar en la base mongo datos de tipo [“polls”] y [“answers”], por lo que en las funciones respectivas “addPoll” y “addAnswer”, vengan los datos “question” , “answer” y poll\_id incluidos dentro del JSON, para definir una regla de negocio clara, y validar la coherencia y relación entre preguntas y respuestas , ya que esto facilita el trabajo con los datos, y por supuesto son parte de la lógica de la API de encuestas.

Se propone validar que en el JSON exista la clave “question” para las peticiones a /addPoll, y que exista la clave “answers”, y “poll\_id” para las peticiones a /addAnswer

1. Continuando con la lógica de encuestas, preguntas y respuestas. Como menciono en el punto anterior las peticiones /addAnswer deben tener un dato respuesta “answer” y un “poll\_id” pero este id no puede ser cualquiera, sino que tiene que ser relacionado con alguna pregunta previamente cargada en la base de datos.

Se propone validar que el “poll\_id” que viene en las peticiones /addAnswer exista en la base de mongo, para esto se puede hacer uso de la función “find\_byid” definida en la clase MongoAPI