Protocolo interno

# Cliente – API

Llamadas tipo GET:

* GET("api/user/:username") => devuelve OK o ERR
* GET("api/event/:eventid") => devuelve JSON del evento o ERR
* GET("api/list/:username") => devuelve JSON con la lista (puede estar vacía) de eventos filtrada por (organizer === :username) o ERR (si no existe usuario

Llamadas tipo POST:

* POST("api/add/event", JSON con todos los datos de event menos eventid) => devuelve eventid o ERR
* POST("api/remove/event", JSON con :eventid) => devuelve OK o ERR
* POST("api/add/assistant", JSON con :username y :eventid) => devuelve OK o ERR
* POST("api/remove/assistant", JSON con :username y :eventid) => devuelve OK o ERR
* POST("api/add/user", JSON con :username) => devuelve OK o ERR

# API – BD

* Conexión mediante sockets TCP con la librería 'zmq' de Nodejs.
* Se envían *strings* en los mensajes que parten de formato JSON.
* Si se trata de mensaje de API a BD, se indican los siguientes parámetros importantes:
  + - 'component': indica si se trata de una petición de ‘user’ o ‘event’.
    - 'id': identificador del elemento que quiera introducirse o consultarse.
    - 'body': cuerpo de la petición. Se forma de parámetros que detallan el elemento concreto y de ‘op’ (pudiendo ser únicamente *get* o *put*).
* Si se trata de mensaje de BD a API, se establece respuesta por cada tipo de petición:
  + Para 'put' de 'user': *Done* si la operación se ha realizado correctamente y *Failed* en caso contrario.
  + Para 'put' de 'event': 'id' aleatorio generado para el nuevo evento registrado (de ser correcto) y 'Failed' en caso contrario.
  + Para 'get' de 'user': *OK* si existe el usuario y *Failed* en caso contrario.
  + Para 'get' de 'event': los detalles que lo definen y 'Failed' en caso contrario.

# Puertos utilizados

* Se sirve el cliente web en el puerto 4000.
* Se establece la API en el puerto 3000, de modo que el cliente se comunique con ella a través de dicho puerto.
* Se establece el servicio de Base de Datos en el puerto 1234, de tal forma que la API envíe las peticiones a través de este puerto.

# Entidades almacenadas en BD

* User
  + Name (string)
* Event
  + ID (string)
  + Name (string)
  + Datetime (string)
  + Description (string)
  + Organizer (string)
  + Assistants (list of string)