ADR Base de Datos

# Contexto

Es necesario definir qué tipo de Base de Datos se va a utilizar para almacenar y gestionar los datos dispuestos en la aplicación.

# Procesos

1. Interacción continua entre las tres componentes de la aplicación para que la Base de Datos realice su cometido (utilizando JSON para dar estructura a los datos en los envíos):
   1. El usuario realiza una operación desde el cliente web (registrarse, identificarse, crear un nuevo evento, indicar que se asiste a un evento, etc.).
   2. El cliente web manda la acción correspondiente a la API con los datos proporcionados por el usuario.
   3. La API se comunica con la Base de Datos para indicar la petición a realizar en ella (get o put, es decir, guardar o devolver datos) con los datos disponibles.
   4. La Base de Datos almacena datos nuevos o devuelve los almacenados según haya sido realizada la petición.
      1. Los datos críticos deben estar cifrados (en este caso, únicamente la contraseña del usuario se considera relevante).
      2. No se deben perder datos debido a errores de gestión.
   5. La Base de Datos devuelve información a la API, ya sea como confirmación de éxito de una operación como de los datos requeridos.
   6. La API envía la información obtenida al cliente web.
   7. El cliente web se actualiza para mostrarse del modo que corresponda (cambio de página, confirmación de operación realizada, etc.).

Cabe resaltar las características que se requieren para el sistema, de modo que, aunque no se trata de un proceso como tal, se tengan en cuenta para la decisión: alta disponibilidad, escalabilidad y almacenamiento persistente (lo cual conlleva a que no se produzcan pérdidas por fallos internos o de la red o por cualquier otra razón).

# Opciones consideradas

Aunque hay un amplio abanico de posibilidades para la implementación de una Base de Datos, incluyendo tanto locales como en la nube, se decide acotar la comparación a dos opciones: una muy completa pero complicada de dominar y otra muy básica pero sencilla de utilizar.

1. ElasticSearch:
   1. Pros:
      1. Permite escalabilidad horizontal.
      2. Está orientada a ficheros JSON.
      3. Incluye un sistema de *master-workers* distribuido en un clúster que facilita una alta disponibilidad con alta tolerancia a fallos.
      4. Proporciona un sistema de replicación automático.
   2. Cons: el equipo no está familiarizado con su uso.
2. LevelDB:
   1. Pros:
      1. Librería de Nodejs.
      2. Uso sencillo.
      3. Más de la mitad del equipo está familiarizado con su uso.
   2. Cons:
      1. Sólo permite almacenamiento de pares clave-valor.
      2. No facilita los requerimientos del sistema.

# Decisión

Se ha escogido **LevelDB** como Base de Datos para la aplicación.

La decisión ha sido tomada en base a que la aplicación ya se está implementando en JavaScript y Nodejs. Además, algunos de los desarrolladores ya tienen experiencia tratando con la librería, por lo que no va a ser necesario dedicar horas a formación del personal antes de comenzar la programación de la Base de Datos.

Los datos que se van a utilizar en la aplicación son muy básicos y no necesitan de estructuras relacionales muy complejas, por lo que la simpleza de esta librería es suficiente.

Por último, el alcance de este proyecto es reducido debido a los recursos tecnológicos y temporales de los que se dispone, por lo que conseguir un sistema con todas las características mencionadas anteriormente no es crucial en este momento; sino que, aun siendo básico, sea funcional.

# Estado

(Tachar las que no apliquen)

1. Aceptado
2. ~~Rechazado~~
3. ~~Reemplazado~~

# Implicaciones

1. Instalación de la librería LevelDB de Nodejs en la máquina virtual utilizada para el proyecto.
2. El sistema no cumplirá, en principio, las características requeridas.

Hay que destacar que, si el contexto fuese otro, ElasticSearch sería la mejor opción para la Base de Datos, dado que permitiría alcanzar las características principales para el sistema de forma satisfactoria y medianamente automática.

### Fecha

21 de mayo. [La decisión no había sido plasmada como ADR]