

Se desea realizar una API GraphQL para informar del resultado de los partidos de una liga de baloncesto.

Los partidos tienen los siguientes datos:

- Nombre equipo 1.
- Nombre equipo 2.
- Resultado, ejemplo "27-32"
- Minuto de juego.
- Finalizado (boolean)

Se deberán crear las colecciones necesarias en una base de datos de Mongo alojada en Mongo Atlas.

No se hará gestión de usuarios, por lo que aunque no sea realista todos los usuarios pueden hacer cualquier de las siguientes operaciones.

Las operaciones GraphQL disponibles serán:

listMatches (1.5 puntos)

- Devolverá un array con todos los partidos en juego (no finalizados)
- Si no hay partidos en juego devolverá un array vacío.

getMatch (id) (1.5 puntos)

- Devolverá un partido, con todos sus datos, independientemente de que esté finalizado o no.
- Si no existe un partido con dicho *id* devolverá un error 404.

subscribeMatch (id) - (3 puntos)

- Permitirá suscribirse a un partido que esté en juego, de modo que **cada vez que cambie el resultado o cuando finalice el partido** me llegará a través de la suscripción los datos del partido.
- Si no existe un partido con dicho *id* devolverá un **error 404**.

setMatchData(id: ID!, result: String, minute: Int, ended: boolean) (2 puntos)

- Actualiza los datos de un partido (el nombre de los equipos no se puede actualizar)
- Debe comprobar que los datos son coherentes (*en caso contrario devolverá un **error 442***)
 - La puntuación de ninguno de los 2 equipos es menor que la puntuación anterior.
 - Los minutos transcurridos son mayores que los minutos anteriores.
 - Si el partido ya había finalizado no permite modificar ningún dato.
- Si no existe un partido con dicho *id* devolverá un **error 404**.

startMatch (team1: String, team2: String) (2 puntos)

- Crea un nuevo partido, poniendo el resultado a "0-0" y el minuto a 0
- Debe comprobar que no existe ningún partido **en curso** con el mismo nombre de equipos (aunque estén en distinto orden). *En caso contrario devolverá un **error 442***

(leer notas de la página siguiente)

NOTAS IMPORTANTES

- Las credenciales de acceso a la DDBB de mongo deben ir en un archivo `.env` que no se añadirá a github
- El repositorio de github debe contener un archivo `.env.sample` que sirva como ejemplo de cómo debe ser el archivo `.env`
- No se deben subir a github archivos no necesarios
- En el README.md se debe explicar cómo lanzar el servidor.

(en caso de no hacerlo siguiendo estas notas se restarán 2 puntos sobre el resultado final)