

Matches – Partidos

Nuevos partidos

El objeto de Match contiene información correspondiente al resultado del enfrentamiento de 2 equipos. La información contenida deberá distinguir entre el equipo de casa y el equipo visitante; también deberá incluir el puntaje de cada equipo y el ganador del partido.

La creación de partidos ocurrirán en 2 eventos:

1. Durante el registro de equipos dentro de un grupo
 - a. Dependiendo de las reglas del torneo, un partido puede ser creado o actualizado por cada evento
 - b. Una vez determinado que un torneo este "lleno" todos los grupos tengan equipos asignados se crean todos los partidos de la eliminatoria a playoffs (fase de playoffs no aplica para torneo de eliminación doble)
2. Al registrar el resultado de un partido de eliminación sencilla
 - a. Ejemplo: Al registrar el marcador de un partido de cuartos de final se genera el siguiente partido de semifinal con el ganador; al registrar el ganador del siguiente partido solo se actualiza el equipo ganador en el partido ya creado de semifinal

Caso especial para eliminación doble: Siendo que eliminación doble es una versión de eliminación sencilla, los partidos pueden ser generados al inicio e ir llenando los equipos mientras evolucione el torneo. De igual manera se puede ir creando los partidos conforme evoluciona el torneo

Detalles de implementación

Eventos

- Generar evento al registrar un grupo a un torneo
 - Decidir en generar un partido mientras cada equipo este registrado a un torneo o generar todos los partidos una vez terminado de registrar todos los equipos
- Generar un evento al registrar el marcador de un partido
 - Una vez terminada la primera fase, generar los partidos para eliminación sencilla (cuartos de final), y asignar los equipos segun sean las reglas de cada torneo.
No aplica para eliminación doble
 - Identificar si el partido es de eliminación sencilla, asignar al ganador al siguiente juego. **No aplica para eliminación doble**
 - Para eliminación doble, al registrar el resultado de un partido asignar al ganador y perdedor a su siguiente partido correspondiente. En el partido final hay una posibilidad de jugar 2 partidos

API

URI: /tournaments/<TOURNAMENT_ID>/matches

Método: GET

Respuesta

- HTTP 404
 - Si el torneo no existe
- HTTP 500
 - En caso de error de sistema
 - Base de datos no accesible
- HTTP 200
 - Arreglo vacío si no existen partidos
- HTTP 200, agregar filtros con parametros de query HTTP
 - /matches?showMatches=played
 - /matches?showMatches=pending

```
[  
 {  
   "home": {  
     "id": "<team-id>",  
     "name": "<team-name>"  
   },  
   "visitor": {  
     "id": "<team-id>",  
     "name": "<team-name>"  
   },  
   "round": "regular"  
 },  
 {  
   "home": {  
     "id": "<team-id>",  
     "name": "<team-name>"  
   },  
   "visitor": {  
     "id": "<team-id>",  
     "name": "<team-name>"  
   },  
   "round": "regular"  
 }]
```

```

    "visitor": {
        "id": "<team-id>",
        "name": "<team-name>"
    },
    "round": "regular",
    "score": {
        "home": 1,
        "visitor": 2
    }
}
]

```

URI: /tournaments/<TOURNAMENT_ID>/matches/<MATCH_ID>

Método: GET

Respuesta

- HTTP 404
 - Si el partido no existe
- HTTP 500
 - En caso de error de sistema
 - Base de datos no accesible
- HTTP 200

```
{
    "home": {
        "id": "1223445",
        "name": "Equipo1"
    },
    "visitor": {
        "id": "09887766",
        "name": "Equipo2"
    },
    "round": "regular",
}
```

```
        "score": {  
            "home": 1,  
            "visitor": 2  
        }  
    }  
  
URI: /tournaments/<TOURNAMENT_ID>/matches/<MATCH_ID>
```

Método: PATCH

Cuerpo

```
{  
    "score": {  
        "home": 1,  
        "visitor": 2  
    }  
}
```

Respuesta

- HTTP 404
 - Si el partido no existe
- HTTP 500
 - En caso de error de sistema
 - Base de datos no accesible
- HTTP 422
 - En caso de validación incorrecta del marcador (depende de los detalles de cada torneo)
- HTTP 204

Unit tests

Match Listener

- Hacer un prueba para cada operación en donde valide las respuestas documentadas

MatchDelegate

- Hacer las pruebas analogas al controller
- Hacer las pruebas que corroboren las reglas de negocio al insertar el marcador, e. g., validar empates para juegos en la fase de eliminación sencilla.

Consumidores de Eventos

Equipo Agregado

El consumidor de este evento puede tener 2 estrategias para cumplir su función; 1) mientras los equipos se estén agregando, crear los partidos sobre la marcha; 2) Una vez agregados todos los equipos generar todos los partidos en el último evento. Agregar validaciones para NO generar los partidos si por alguna razón siguen llegando mensajes.

Consideraciones técnicas

En un consumidor, no existe un API REST, la "actividad" del código es iniciada por la presencia de un mensaje en la cola de ActiveMQ.

Unit test esperados

- Creación de todos los partidos de primera fase (excluyendo los de eliminación sencilla, "playoffs")
- Dependiendo la estrategia, pruebas en donde "no se hace nada" cuando el torneo esté incompleto

Registro de puntuación (Score)

El consumidor de este evento es encargado de generar partidos durante o al finalizar la primera fase, e. g. una vez capturado el último puntaje de la primera fase, aplicar las reglas de tabulación de acuerdo a cada torneo y generar los partidos de eliminación sencilla, para los partidos de eliminación sencilla, pasar a los equipos correspondientes al siguiente partido.

En el caso de eliminación doble, no existe "primera" fase, por lo tanto el marcador de cada partido siempre define los participantes del siguiente partido

Agregar un campo en donde se pueda identificar que el torneo ha concluido

Unit test esperados

- Para los torneos que aplique, validar la creación de la fase de eliminación sencilla
- En caso de partidos de eliminación sencilla, validar que el equipo apropiado sea asignado en el siguiente partido, de octavos de final a cuartos de final; de cuartos de final a semifinal; y semifinal a final
- En caso de partidos de eliminación doble, validar que el equipo apropiados sea asignado al siguiente partido