

# Tasca S5.02 Joc de Daus

Por hacer: Recibir una calificación

## Descripció

Aquest és el teu projecte final, una API 100% dissenyada per tu on aplicaràs tot el que saps fins ara per a crear una aplicació al complet, des de la base de dades a la seguretat. Aplica tot el que saps fins i tot el que no es demana.

## Nivell 1

El joc de daus s'hi juga amb dos daus. En cas que el resultat de la suma dels dos daus sigui 7, la partida és guanyada, sinó és perduda. Un jugador pot veure un llistat de totes les tirades que ha fet i el percentatge d'èxit.

Per poder jugar al joc i realitzar una tirada, un usuari s'ha de registrar amb un nom no repetit. Al crear-se, se l'hi assigna un identificador numèric únic i una data de registre. Si l'usuari així ho desitja, pots no afegir cap nom i es dirà "ANÒNIM". Pot haver-hi més d'un jugador "ANÒNIM".

Cada jugador pot veure un llistat de totes les tirades que ha fet, amb el valor de cada dau i si s'ha guanyat o no la partida. A més, pot saber el seu percentatge d'èxit per totes les tirades que ha realitzat.

No es pot eliminar una partida en concret, però si que es pot eliminar tot el llistat de tirades per un jugador.

El software ha de permetre llistar tots els jugadors que hi ha al sistema, el percentatge d'èxit de cada jugador i el percentatge d'èxit mig de tots els jugadors en el sistema.

El software ha de respectar els principals patrons de disseny.

### NOTES

Has de tindre en compte els següents detalls de construcció:

- URL's:
- POST: /players : crea un jugador
- PUT /players : modifica el nom del jugador
- POST /players/{id}/games/ : un jugador específic realitza una tirada dels daus.
- DELETE /players/{id}/games: elimina les tirades del jugador
- GET /players/: retorna el llistat de tots els jugadors del sistema amb el seu percentatge mig d'èxits
- GET /players/{id}/games: retorna el llistat de jugades per un jugador.
- GET /players/ranking: retorna el ranking mig de tots els jugadors del sistema . És a dir, el percentatge mig d'èxits.
- GET /players/ranking/loser: retorna el jugador amb pitjor percentatge d'èxit
- GET /players/ranking/winner: retorna el jugador amb pitjor percentatge d'èxit

## - Fase 1

- Persistència: utilitza com a base de dades mysql

## - Fase 2

- Canvia la configuració i utilitza MongoDB per persistir les dades

## - Fase 3

- Afegix seguretat : inclou autenticació per JWT en tots els accessos a les URL de l'microservei.

## Nivell 2

- Afegeix **tests** unitaris, de component i d'integració al projecte amb llibreries **jUnit**, **AssertJ** o **Hamcrest**.
- Afegeix **Mocks** al testing del projecte (Mockito) i **Contract Tests** (<https://docs.pact.io/>)



## Nivell 3

- Dissenya i modifica el projecte diversificant la persistència perquè utilitzi **dos esquemes de base de dades alhora, MySQL i Mongo.**

## Recursos

- **JWT**
- **JWT + Spring**
- **Spring Boot Initial Demo Project**

## Objectius

- JPA
- MySQL
- MongoDB
- Spring Security
- JWT
- Testing

**Durada: 11 dies**

**Lliurament:**

- Treballa sobre el teu propi repositori
- El lliurament es farà mitjançant pull-request (el mentor ha de tenir permisos sobre el teu repositori).
- **La qualificació inclou la defensa del teu codi en una entrevista amb el mentor.**

## Estado de la entrega

Estado de la entrega	No entregado
Estado de la calificación	Sin calificar
Última modificación	-