

Black Jack

Descripció:

Ens demanen desenvolupar una API Rest basada en el joc del Black Jack.

L'usuari jugarà la partida contra la màquina i guanyarà qui estigui més aprop de 21 sense passar-se.

En primer lloc, el jugador es registra al sistema i, a partir d'aquí, podrà crear la partida per començar a jugar.

La partida sempre comença amb 0 punts i el jugador pot anar demanant cartes. Es van sumant tants punts com el valor de la carta (un valor a l'entre 1 i 12). Si arriba a 21 exacte, es marca la partida com a guanyada. En cas de passar-se de 21, quedarà perduda.

En qualsevol moment, el jugador pot decidir plantar-se, marcant la partida com "stand". En aquell moment, es mira la puntuació de la màquina per decidir qui serà el guanyador.

El jugador a de poder veure l'historial de partides, on es mostrarà el seu estat i el seu resultat.

A més, tindrà emmagatzemat en el seu perfil, quantes partides ha jugat, quantes ha guanyat i quantes ha perdut.

Estructura en detall:

Usuari:

Un usuari es registra a la plataforma sense cap partida activa.

L'usuari té un email, nom d'usuari, una contrasenya i una data de creació.

La contrasenya ha de tenir mínim 8 caràcters i almenys un ha de ser numèric.

Es pot actualitzar la contrasenya i el nom d'usuari del jugador.

Partida:

Una partida ha de contenir l'estat actual: playing, stand, won, draw o lost.

Es important tenir en compte que un jugador pot tenir màxim 3 partides actives, és a dir, que no estiguin guanyades, perdudes o empatades.

Quan es crea la partida, s'inicialitza amb playing. El jugador anirà demanant cartes fins que:

- a) S'arribi a 21, marcant la partida com WON
- b) Es superi 21, marcant la partida com LOST
- c) El jugador decideixi plantar-se, deixant la partida com STAND. No es pot plantar si ja s'ha guanyat o perdut.

Si la partida queda STAND, automàticament s'agafarà un número a l'atzar entre 13 i 25 que marcarà el resultat de la màquina:

- a) Si el resultat de la màquina és superior a 21, la partida serà WON.
- b) Si el resultat de la màquina és superior al jugador e inferior a 21, la partida serà LOST.
- c) Si hi ha empat de punts, es marcarà com DRAW.

Sempre que es demani una partida, s'ha de poder veure la puntuació actual i el seu estat.

Al acabar una partida, s'ha d'actualitzar les estadístiques del jugador.

Disseny de la API:

Amb aquest enunciat, hem dissenyat la següent API:

	/users	/id	/games	/id
POST	CREAR USUARI	X	CREAR PARTIDA	DEMANAR CARTA
PUT	X	ACTUALITZAR USUARI	X	DEIXAR LA PARTIDA A "STAND"
GET	RETORNAR TOTS ELS USUARIS	x	RETORNA TOTES LES PARTIDES DEL JUGADOR	RETORNA LA PARTIDA DEL JUGADOR
DELETE	X	ESBORRA L'USUARI AMB LES SEVES PARTIDES	X	

Entrega:

Es demana desenvolupar aquest projecte amb Spring (JAVA) i integrar-ho amb la Bases de Dades MySQL.

Es pot fer el desenvolupament amb JPA (Spring data) o JDBC, a la vostra elecció.

Es imprescindible aplicar correctament el patró capes i que el codi sigui en anglès.

El projecte ha de contenir el vostre nom:

EXAMEN_BLACK_JACK_JORDIBOIX

L'entrega s'ha de realitzar per mitjà de GIT, pujant el codi al assignament que hi ha disponible al e-campus.

IMPORTANT: El codi final ha d'estar a la branca "MASTER" o "MAIN" des d'on es realitzarà la correcció.

PUNTUACIÓ:

	/users	/id	/games	/id
POST	CREAR USUARI (0.5 punt)	X	CREAR PARTIDA (1.5 punt)	DEMANAR CARTA (2 punt)
PUT	X	ACTUALITZAR USUARI (0.5 punt)	X	DEIXAR LA PARTIDA A "STAND" (2 punt)
GET	RETORNAR TOTS ELS USUARIS (0.5 punt)	x	RETORNA TOTES LES PARTIDES DEL JUGADOR (1 punt)	x
DELETE	X	ESBORRA L'USUARI AMB LES SEVES PARTIDES (1 punt)	ELIMINA TOTES LES PARTIDES DEL JUGADOR (1 punt)	x

Per aconseguir la puntuació màxima de cada apartat, ha d'estar correctament desenvolupat a totes les capes, la API Rest i la Base de dades.