KATA TÉCNICA

APP DE BASKET

Tramada por Juan G. Rodríguez Carrión

1 Enunciado

Se pretende realizar una aplicación por consola con el objetivo de gestionar un pequeño equipo de baloncesto local. El entrenador, mediante un pequeño shell de linux, la usará para poder gestionar el acta del partido con la alineación y la táctica elegida, para entregarla al árbitro antes de que comience el partido.

La aplicación deberá de ser capaz de gestionar un listado de jugadores y de tácticas, y también nos sugerirá de cuál es la mejor alineación a utilizar.

2 Casos de uso

2.1 Alta de jugadores (Prioridad alta)

Los datos que me interesa gestionar de un jugador son:

- El número de su dorsal, que lo identifica.
- El nombre de la camiseta
- Su rol preferido (BASE, ESCOLTA, ALERO, ALA-PIVOT, PIVOT)
- La valoración media del entrenador (un entero de 0 a 100)

2.2 Baja de jugadores (Prioridad baja)

Se necesita borrar físicamente jugadores por el número del dorsal.

2.3 Listado de jugadores (Prioridad alta)

Listado con todos los jugadores del equipo. Como mejora, se debe permitir ordenar el listado por:

- Dorsal
- Valoración media.

2.4 Tácticas (Prioridad media)

El sistema debe de tener preconfiguradas las siguientes alineaciones básicas:

- Defensa 1-3-1: BASE + ESCOLTA + ESCOLTA + ALA-PIVOT + PIVOT
- Defensa Zonal 2-3 : BASE + BASE + ALERO + PIVOT + ALA-PIVOT
- Ataque 2-2-1: BASE + ALERO + ESCOLTA + PIVOT + ALA-PIVOT

2.5 Calculadora de alineaciones (prioridad media).

El entrenador debe poder elegir una táctica, y la aplicación debe de sugerirle la alineación ideal, atendiendo a la valoración y al rol de los jugadores que tiene, o indicarle de que no puede aplicar dicha táctica.

2.6 Histórico de operaciones (Prioridad alta)

La aplicación deberá guardar todas las operaciones de altas y bajas de jugadores que se realicen, y la hora a la que se realizaron. Ojo, no es un log.

3 Consejos

- Los datos de salida de todos los casos de uso, en JSON. Basta con un simple json_encode, pero puedes usar otras formas más avanzadas.
- En el código adjunto, hay un pequeño ejemplo de cómo escribir una aplicación que ejecute comandos por consola, cómo persistir en disco, y cómo testear. Pero ojo, no está bien organizado, eso es cosa tuya cuando copies ideas.
- No hay que complicarse con el sistema de persistencia. Usa la idea de la demo, que consiste en un simple serialize - unserialize de php, o usa un repositorio en memoria con alguna pequeña cantidad de información inicial.
- En el listado de jugadores hay un orden. No la ordenes dentro del repositorio. Haz que éste sólo devuelva el listado, y que la ordenación la haga en PHP la clase que llame al repositorio.
- La prioridad de los ejercicios sirve para seguir un orden a la hora de entregar la funcionalidad. Al seguir el orden, tocarás los puntos básicos de la arquitectura hexagonal y DDD.
- No es necesario realizar todos los ejercicios para tener una buena valoración; aunque siempre es mejor que sobren y no que falten. La calidad es más importante que la cantidad.
- Añadir funcionalidad extra, como un sistema de seguridad, o un sistema de log, se valorará; pero céntrate en la funcionalidad básica que piden los casos de uso. Si te sobra tiempo, entonces aprovéchalo para lucirte.