

Ejercicio de modelado de ejércitos

El ejercicio consiste en modelar ejércitos.

Ejércitos

Los ejércitos poseen una cantidad inicial de piqueros, arqueros y caballeros dependiendo de la civilización. Pueden coexistir muchos ejércitos de la misma civilización. También poseen 1000 monedas de oro al momento de la creación. Cada ejército posee un historial de todas las batallas en las que participó.

Unidades

A cada unidad le puedo pedir los años de vida de la misma.
Hay tres tipos de unidades:

- Piquero
- Arquero
- Caballero

Cada unidad posee puntos de fuerza, según la siguiente tabla:

Unidad	Puntos aportados
Piquero	5 puntos
Arquero	10 puntos
Caballero	20 puntos

La cantidad inicial de unidades es determinada por la siguiente tabla:

Civilización	Piqueros	Arqueros	Caballeros
Chinos	2	25	2
Ingleses	10	10	10
Bizantinos	5	8	15

Entrenamiento

Cada unidad se puede entrenar, esto tiene un costo en monedas de oro y un beneficio en puntos de fuerza que se le suman a la unidad.

Unidad	Puntos obtenidos	Costo del entrenamiento
Piquero	3	puntos 10
Arquero	7	puntos 20
Caballero	10	puntos 30

Transformación

Cada unidad puede entrenar, a un costo, para convertirse en otra, según la siguiente tabla:

Unidad original	Unidad a la que se convierte	Costo (monedas de oro)
Piquero	Arquero	30
Arquero	Caballero	40
Caballero	No se puede entrenar	-

Batallas

Un ejército puede atacar a otro en cualquier momento, incluso si son de la misma civilización. Al hacerlo el ganador de la batalla es simplemente el ejército que tiene más puntos. Las consecuencias de la batalla son las siguientes:

- Ejército perdedor: Pierde las dos unidades con mayor puntaje.
- Ejército ganador: Obtiene 100 unidades de oro.
- En caso de empate: Ambos jugadores pierden alguna unidad (queda a criterio del programador).

Notas

El ejercicio es de modelado, lo más importante es representar correctamente el dominio del problema.

Debe ser desarrollado en Ruby on Rails, implementando el endpoint de transformar una unidad y el de obtener las unidades de un ejército o por lo menos dejar este último planteado.

Focalizarse en desarrollar lo que se pide, no es necesario ni recomendable realizar funcionalidad extra.

No luchar con detalles no esenciales, ante dudas o ambigüedades del enunciado tomar una decisión y, si se cree necesario, justificarla.

No se debe realizar una interfaz de usuario.