



20 de septiembre de 2018

No Evaluada

Actividad 06

EDD 201

Introducción

Luego de oír que en la Tierra existe un *supersaiyayin* capaz de derrotar al poderoso Billis, *Thanenzo el Titini Loco* ha decidido comenzar la conquista de la galaxia en nuestro querido planeta. Para esto necesitará en su campaña al mejor ejército del universo, pero aún no sabe cómo escogerlo. Es por esto que ha mandado a *Benjony Maw*, uno de los miembros de la *Orden Negra*, a abducirte para que le ayudes.



Tu deber será crear un programa que permita evaluar la afinidad que hay en cada ejército, para así poder compararlos. Para esto, *Thanenzo* te ha pedido que modelos y evalúes algunos ejércitos que él tiene pensado.

Ejércitos

Cada ejército del universo está formado por distintas **entidades** que se rigen según una estricta jerarquía en forma de **árbol** a base de rangos, los cuales ordenados en forma decreciente son: **General**, **Teniente**, **Mayor**, **Capitán** y **Soldado**.

A grandes rasgos, la estructura de la jerarquía es la siguiente: cada entidad puede tener a cargo solamente a entidades del grado inmediatamente inferior, y están bajo el mandato directo de una entidad de grado inmediatamente superior¹. Por último, existe exactamente un general por ejército.

¹Excepto el general, que no le manda nadie.

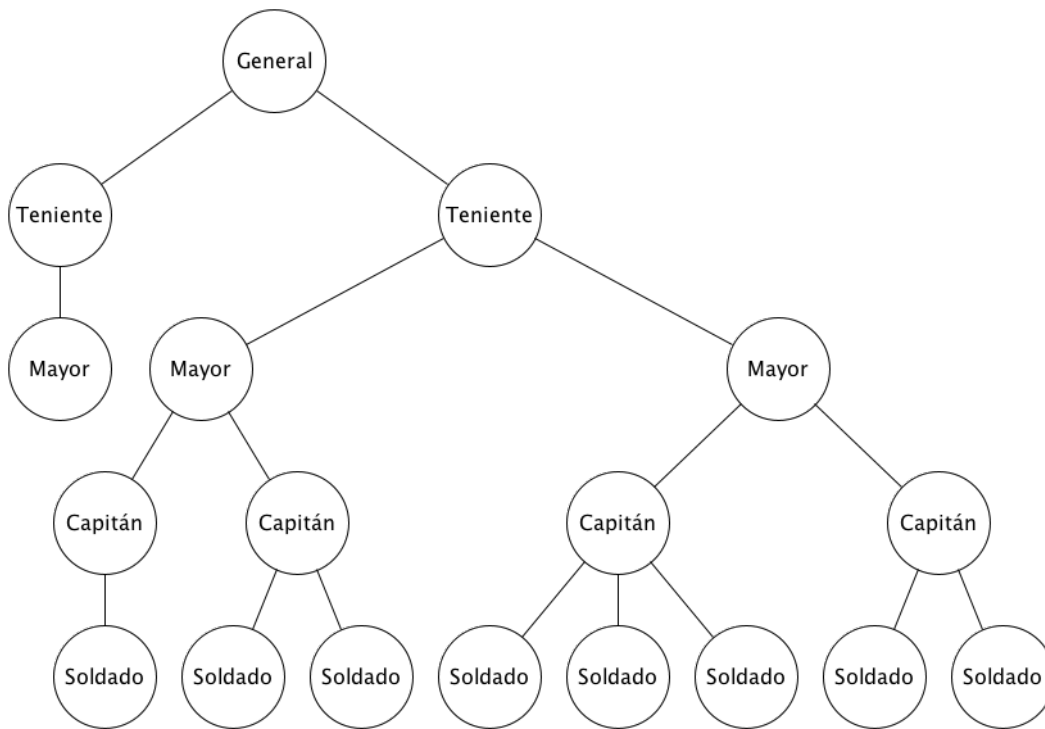


Figura 1: Árbol de ejemplo.

Atributos de las entidades

Cada entidad que compone la jerarquía tiene al menos los siguientes atributos o *properties*:

- `self.rango`: es un *string* que indica el rango de la entidad.
- `self.items`: es un *set* de *strings* que representan los ítems que carga cada entidad, como por ejemplo, armas, armadura, talismanes, u otros.
- `self.superior`: es la entidad que está a cargo en forma directa de la actual.
- `self.superiores`: es una lista de todas las entidades que están a cargo de la actual, ya sea en forma directa o indirecta. Por ejemplo, si la entidad es un **mayor**, entonces esta lista debería incluir un **teniente** y un **general**.
- `self.afinidad_superiores`: es un entero que indica la suma de la afinidad con cada uno de sus superiores, directos o indirectos. La afinidad de una entidad con algún superior es la cantidad de ítems que ellos tienen en común.
- `self.subordinados`: es una lista de los subordinados directos de la entidad.
- `self.rangos_subordinables`: es un *set* de rangos que puede subordinar esta entidad, en forma directa o indirecta. Por ejemplo, un teniente siempre puede subordinar mayores, puesto que los maneja directamente. Sin embargo, si el teniente no cuenta con ningún mayor hasta el momento, él no podrá tener bajo su jerarquía capitanes, ni menos soldados.

Armar el ejército

El primer integrante que se agrega al ejército es su general. Cada nueva entidad que se **agrega** al ejército es recibida por el general, quien decide cuál de sus tenientes es el que tiene **mejor afinidad con el nuevo integrante**, y se lo entrega². Luego este teniente hace lo mismo, hasta posicionar la nueva entidad en su lugar correspondiente del árbol.

Además, cada ejército tiene un determinado **poder** que se mide sumando el valor `afinidad_superiores` para cada entidad del árbol.

Dado lo anterior, las entidades deben tener el siguiente método:

- `def agregar_subordinado(self, entidad):` recibe una entidad y la agrega a sus subordinados o escoge de ellos aquel con el que tenga mejor afinidad, según corresponda.

Archivos

Thanenzo te dio cuatro documentos en formato **CSV**, donde cada uno describe un ejército. Cada línea del archivo describe un integrante del ejército, primero colocando su rango, y después una cantidad arbitraria de ítems con los que carga. En otras palabras, cada línea del archivo es de la forma:

`rango,item 1,item 2,...,item n`

El general será **siempre** el primero de la lista, y las entidades estarán en un orden en que siempre es posible agregarlas todas al árbol. Un ejemplo de archivo es el siguiente:

```
general,dagas,cota de malla,espada,martillo de guerra
teniente,dagas,cota de malla
mayor,armadura de placa,veneno,frascos,martillo de guerra
```

La idea es que en el `main` de tu programa puedas cargar estos archivos y mostrar el poder para cada uno de estos ejércitos.

Notas

- No olvide que esta actividad es de estructuras de datos con nodos, en particular, árboles. No debería ser difícil modelar el problema de esta manera.

Entrega

- **Lugar:** En su repositorio privado de GitHub, en la **carpeta** `Actividades/AC06/`
- **Hora del último *push*:** 16:00

²Hay que cuidar entregarlo solamente a entidades que van a poder manejar al nuevo integrante (ver `self.rangos_subordinables`)