

Ejercicios de Programación orientada a objetos

Programación — DAW

Ricardo Pérez López
IES Doñana

Curso 2020/2021

1. Traducir a clases y objetos todo el código que tenemos actualmente en <https://github.com/iesdonana/vampiro.git>, teniendo en cuenta que:
 - a) Las localidades deben ser objetos de la clase `Localidad`.
 - b) Es posible que no sea necesario hacer una clase `Localidades`.
 - c) El jugador debe ser un único objeto de la clase `Jugador` (a esto se le llama *Singleton*).
 - d) Las conexiones de una localidad se deben almacenar dentro de la localidad.
 - e) Cada conexión puede ser un objeto de una clase `Conexion`, o puede que no merezca la pena crear una clase para eso. En tal caso, las conexiones serían parejas de elementos (que se podrían representar con cualquier estructura tipo lista, tupla, diccionario...) que contenga una dirección y una localidad de destino.
 - f) Los ítems (objetos que aparecen en el juego, como el crucifijo o la ristra de ajos) deben ser objetos de la clase `Item`.
 - g) Las colecciones deben ser objetos de la clase `Coleccion`.
 - h) Los ítems pueden estar en una localidad o en el inventario del jugador. Para ello, hay que usar objetos de la clase `Coleccion`.
 - i) Un *token* representa una palabra con significado propio y distinto de otras palabras. Cada *token* debe ser un objeto de la clase `Token`.
 - j) Cada grupo de palabras del mismo tipo («verbo», «nombre», etc.) debe ser un objeto de la clase `Vocabulario`. Así, debe haber un vocabulario que contenga todos los verbos, otro que contenga todos los nombres, etc.

Cada objeto de la clase `Vocabulario` debe inicializarse con un diccionario que asocie cada *token* con una lista de lexemas que sean sinónimos.

Por ejemplo:

```
ABRIR = Token()
ARRIBA = Token()
verbos = Vocabulario({
    ABRIR: ['ABRIR', 'ABRE'],
    ARRIBA: ['ARRIBA', 'SUBIR', 'SUBE']
})
CRUCIFIJO = Token()
AJOS = Token()
nombres = Vocabulario({
    CRUCIFIJO: ['CRUCIFIJO', 'CRUZ'],
    AJOS: ['AJOS', 'AJO', 'RISTRA']
})
```

- k) La función `interpretar` debe implementarse como un método estático de una clase (que puede ser la misma clase `Vocabulario`) que se encargue de analizar sintácticamente la entrada del jugador a partir de los vocabulario de verbos y de nombres. Ese método debe devolver los *tokens* del verbo y el nombre encontrados, o un *token* especial (llamado *token nulo*) que represente que no se ha encontrado el verbo o el nombre.