

## DSA. Ejercicio de colecciones en Java

Primero vamos a ver el siguiente tutorial, insistiendo en los objetos *List* y *Map*.

<http://tutorials.jenkov.com/java-collections/index.html>

Vamos a implementar un inventario para nuestro juego de acción. Constará inicialmente de 2 clases, una para modelar un usuario del juego y otro para modelar un objeto presente en un inventario. Cada usuario tendrá un conjunto de objetos en su inventario, diferente de los presentes en los inventarios de otros usuarios.

### Usuario

- Nombre (string, el nombre ha de ser único en el sistema)
- Password (string)
- Nivel (int)
- Ataque (int)
- Defensa (int)
- Resistencia (int)

### Objeto

- Nombre (string, este nombre no ha de ser único en el sistema, i.e un usuario puede tener dos "Espada")
- Tipo: Ataque, Defensa, Llave, Poción (string)
- Descripción (string)
- Valor (int)
- Coste (int)

Las operaciones que nuestro sistema debe implementar serán:

- Crear un nuevo usuario en el sistema con sus atributos. Retorna un booleano indicando si la operación ha sido correcta (i.e el nombre no estaba ya siendo utilizado)

```
bool crearUsuario(Usuario u)
```

- Eliminar un usuario existente en el sistema. Retorna un booleano indicando si la operación ha sido correcta (i.e existe un usuario con ese nombre)

```
bool eliminarUsuario(String nombre)
```

- Consultar la información básica de un usuario dado su nombre. Retorna el objeto Usuario con la información o *null* si no existe el nombre.

```
Usuario consultarUsuario(String nombre)
```

- Añade un objeto a un usuario.

```
void añadirObjetoAUsuario(Usuario u, Objeto o)
```

- Consultar los objetos que tiene un usuario. Notar que hay que decidir que retorna esta función según la tarea 2 detallada más adelante.

```
<?> consultarObjetosDeUsuario (Usuario u)
```

- Obtener un objeto concreto del inventario de un objeto de un usuario. Retorna el primer objeto en caso de que haya más de uno con el mismo nombre y *null* si no existe objeto con ese nombre.

```
Objeto consultarObjetoDeUsuario (Usuario u, String  
nombreObjeto)
```

- Eliminar un objeto del inventario de un usuario. Retorna un booleano indicando si el objeto con ese nombre existe y se ha podido borrar.

```
bool eliminarObjetosDeUsuario (Usuario u)
```

- Transferir objeto de un usuario a otro.

```
void transferirObjetoEntreUsuarios (Usuario  
origen, Usuario destino, Objeto o)
```

Se pide:

1. Implementar las 2 clases Usuario y Objeto con, como mínimo, los atributos indicados más arriba.
2. Seleccionar las colecciones más adecuadas para almacenar:
  1. Los usuarios del juego
  2. Los objetos del inventario de cada usuario
3. Implementar una clase Mundo que contenga las colecciones de nuestro sistema y que implemente las operaciones indicadas más arriba (i.e el código de las operaciones NO puede estar en el programa principal)
4. Implementar un programa principal que permita ejecutar y probar las operaciones previas. El programa principal únicamente ha de incluir el código que interacciona con el usuario (i.e Console.WriteLine / Console.ReadLine) no la implementación de las operaciones del sistema).

Documentación adicional:

[https://www.tutorialspoint.com/java/java\\_collections.htm](https://www.tutorialspoint.com/java/java_collections.htm)