

Classes and objects

# Comunidad de vecinos



## Objetivos/Conceptos

Desarrollar capacidades respecto a la programación de clases y objetos

Tiempo estimado	Dificultad	Tipo de actividad
3 horas	Media	Evaluación

## Requirements

Este ejercicio consiste en hacer una simulación de un torneo de artes marciales. Las clases a implementar son las siguientes:

### **Cosas a tener en cuenta:**

- Se trata de hacer una simulación de un torneo de artes marciales.
- Cuando le toque a un combatiente, puede querer atacar.
- Para saber si tiene éxito un ataque, se genera un random entre 0 y 1, si es menor que la capacidad ataque del atacante, el golpe alcanza al objetivo. Faltaría ver si el objetivo quiere esquivar o parar el golpe, y si lo consigue.

### **Utils**

- Es una clase que contiene únicamente atributos y métodos de clase.
- Métodos:
  - **GetRandom**(min: double, max: double) : double → Devuelve un número aleatorio entre el mínimo y el máximo que se le pasan.

### **ITorneo**

- Es una interfaz
- Métodos
  - **Init()** → Es un método que inicia un torneo con 16 participantes. Cada participante será de una de las clases que se especifica a continuación de manera random.
  - **Execute()** : List<string> → Realiza la serie de combates pertinentes, y devuelve una lista con cada combate realizado (y el ganador). Los combates se organizan de esta manera:



Por ejemplo, una posible lista devuelta por esta función puede ser:

Persona1 vs Persona2 -> Ganador Persona1

Persona34 vs Persona2 -> Ganador Persona34

...

Persona12 vs Persona3 -> Ganador Persona3, ganador del torneo

- **Visit** La idea es que esta función recorra todos los combatientes y los imprima por pantalla.
  - En vez de hacer que se impriman por pantalla, para tener más nota, puedes hacer que llame a una lambda que se le pase como parámetro.

## **Torneo**

- Es una clase que implementa la interfaz ITorneo

## **Persona**

- Es una clase
- Tiene los siguientes atributos privados
  - nombre
  - raza (es un enum)
  - energía, es un double random entre 1000 y 2000
  - deseo de esquivar. Es un real entre 0.1 y 0.9.
- Cada atributo tiene un getter
- Tiene los siguientes métodos:
  - **QuitarEnergía**(double) → Quita energía del atributo
  - **Atacar**(Persona) → Es una función abstracta. Cuando se llama a esta función hace que la persona con la que se usa, ataque a otra persona
  - **ObtenerCapacidadDeEsquiva**() : real → Es una función abstracta.
  - **ObtenerCapacidadDeParada**() : real → Es una función abstracta.
  - **QuiereEsquivar**() : booleano → Cada vez que sea atacado, se le tiene que preguntar al contrincante si quiere esquivar. Se usa un random entre 0 y 1, si el random es menor que el "Deseo de Esquivar", devuelve true.

## **Humano**

- Hereda de Persona
- Tiene los siguientes atributos privados
  - ataque con golpes → Es un real entre 0.1 y 0.8
  - capacidad de esquivar → Es un real entre 0.4 y 0.6
  - capacidad de parar → Es un real entre 0.7 y 0.9
- Cuando ataca, gasta 1 punto de energía. Si el contrincante:
  - lo esquiva, no le hará nada
  - lo para, le quitará 0.5 puntos de energía
  - si le da, le quitará 3 puntos de energía

## **Guerrero del espacio**

- Hereda de Persona
- Tiene los siguientes atributos privados

- Ataque con rayo → Es un real entre 0.3 y 0.6
- Ataque con golpes → Es un real entre 0.1 y 0.8
- Capacidad de esquivar → Es un real entre 0.2 y 0.4
- Capacidad de parar → Es un real entre 0.4 y 0.9
- Cuando ataca, puede usar los rayos o los golpes (usar un random como IA).
- Si ataca con rayo, el guerrero del espacio gasta 100 puntos de energía. Si el contrincante:
  - lo esquivo, no le hará nada
  - lo para, le quitará 25 puntos de energía
  - si le da, le quitará 300 puntos de energía
- Si ataca con golpe, el guerrero del espacio gasta 5 puntos de energía. Si el contrincante:
  - lo esquivo, no le hará nada
  - lo para, le quitará 2 puntos de energía
  - si le da, le quitará 7 puntos de energía

### **Supersaiyajin**

- Hereda de Guerrero del espacio
- En su turno de ataque, puede realizar entre 1 y 3 acciones como las del guerrero del espacio, y además, cuando impacta, quita el doble de energía que un guerrero del espacio normal

### **Namekiano**

- Hereda de Persona
- Tiene los siguientes atributos privados
  - Ataque con rayo → Es un real entre 0.1 y 0.4
  - Ataque con golpes → Es un real entre 0.3 y 0.8
  - Capacidad de esquivar → Es un real entre 0.5 y 0.7
  - Capacidad de parar → Es un real entre 0.3 y 0.6
- Cuando ataca, puede usar los rayos o los golpes (usar un random como IA).
- Si ataca con rayo, el guerrero del espacio gasta 50 puntos de energía. Si el contrincante:
  - lo esquivo, no le hará nada
  - lo para, le quitará 20 puntos de energía
  - si le da, le quitará 100 puntos de energía
- Si ataca con golpe, el namekiano gasta 10 puntos de energía. Si el contrincante:
  - lo esquivo, no le hará nada
  - lo para, le quitará 7 puntos de energía
  - si le da, le quitará 19 puntos de energía

### **Restrictions**

- Todas las clases tienen que estar programadas en inglés