

---

```

using UnityEngine;
using System.Collections;
using System;
using System.Text;
using System.Linq;
using System.Collections.Generic;

public enum Est{veneno,dormir,normal};

public struct Estado {
    public Stats statActual;
    public Est est;
    public Estado(Stats s, Est e){
        statActual = s;
        est = e;
    }
    override public string ToString(){
        return statActual.ToString();
    }
    public Estado(string s){
        statActual = new Stats(s);
        est = Est.normal;
    }
}

public struct MovLv{
    public string mov;
    public int lv;
    public MovLv(string m, int l){
        mov = m;
        lv = l;
    }
}

public abstract class Monstruo {
    public string nombre;
    public string especie;
    public Tipo tipo;
    public int lv;
    public int exp;
    public Stats baseStats;
    public Stats modStats = new Stats(0,0,0,0,0,0,0);
    public Estado estado = new Estado(new Stats(0,0,0,0,0,0,0),Est.normal);
    public MovLv[] movPosibles;
    public string imgDir;
    protected virtual void Evolve(){

    }

    public static Monstruo CreateMonster(string monster, string name, int lv){
        Type types = Type.GetType(monster);
        if (types == null)
            throw new InvalidOperationException("The given monster does not have a Type associated with it.");

        return Activator.CreateInstance(types,name,lv) as Monstruo;
    }
    public static Monstruo CreateMonster(string monster, string name, int exp, Stats modS,Estado estado){

        Type types = Type.GetType(monster);

        if (types == null)
            throw new InvalidOperationException("The given monster does not have a Type associated with it.");

        return Activator.CreateInstance(types,name,exp,modS,estado) as Monstruo;
    }
}

```

---

```

private System.Random Rnd = new System.Random();

public int GetDamage(int dam, tipos t, int punteria){
    float pexito = Rnd.Next(0,100)+estado.statActual.velocidad;
    dam = Mathf.RoundToInt(dam*tipo.Modificador(t));
    if((pexito > punteria)|| (pexito>98)){
        Log.AddLine(nombre+" logro esquivar el golpe!");
        return estado.statActual.vida;
    }
    if(pexito < 5){
        Log.AddLine("Fue un golpe bien propinado!");
        dam *= 2;
    }
    if(dam <= 0){
        Log.AddLine("No hizo mucho daño");
        dam = 1;
    }
    estado.statActual.vida -= dam;
    if(estado.statActual.vida <= 0){
        Log.AddLine(nombre+" fue derrotado!");
        estado.statActual.vida = 0;
    }
    return estado.statActual.vida;
}

public int Restaurar(int hp){
    estado.statActual.vida += hp;
    if(estado.statActual.vida >= GetStats().vida){
        estado.statActual.vida = GetStats().vida;
    }
    return estado.statActual.vida;
}

public void Restaurar(){
    estado.statActual = GetStats();
    estado.est = Est.normal;
}

public string[] GetMov(int nivel){
    List<String> temp = new List<string>();
    int i;
    for(i=0;i<movPosibles.Length;i++){
        if(movPosibles[i].lv <= nivel){
            temp.Add(movPosibles[i].mov);
        }
    }
    return temp.ToArray();
}

public string[] GetMov(){
    return GetMov(lv);
}

private int FormulaStat(int b, int m){
    return Mathf.RoundToInt(((m+(2*b)+100)*lv)/100 +10);
}

public Stats GetStats(int nivel){
    Stats temp = new Stats(baseStats.ToString());
    Stats temp2 = new Stats(modStats.ToString());

    temp.vida = Mathf.RoundToInt(((temp2.vida + (2*temp.vida)+100)*lv)/100 +10);
    temp.fuerza = FormulaStat(temp.fuerza,temp2.fuerza);
    temp.defensa = FormulaStat(temp.defensa,temp2.defensa);
    temp.fespecial = FormulaStat(temp.fespecial,temp2.fespecial);
    temp.despecial = FormulaStat(temp.despecial,temp2.despecial);
}

```

```
        temp.velocidad = FormulaStat(temp.velocidad,temp2.velocidad);
        return temp;
    }
    public Stats GetStats(){
        return GetStats(lv);
    }

    public void AddExp(int e){
        exp += (int) (e/lv)*3;
        if(lv < GetLv(exp)){
            Log.AddLine(nombre+" subio de nivel!");
            int hp = estado.statActual.vida;
            lv = GetLv(exp);
            estado.statActual = GetStats();
            estado.statActual.vida = hp;
            Evolve();
        }
    }

    public static int GetExp(int nivel){
        return (int) Mathf.Pow(nivel,5f);
    }
    public static int GetLv(int exp){
        return (int) Mathf.Pow(exp,0.2f);
    }

    public void GetCaught(int prob){
        float pexito = Rnd.Next(0,100)+estado.statActual.vida;
        if(pexito < prob){
            SaveMonster.AddMonster(this,true);
            estado.statActual.vida = 0;
            Log.AddLine(nombre+" fue atrapado!");
        }
    }
}
```