

Model

Jugador

- nombre
- atk
- def
- hp
- hpmax
- inventario[]
- armaActiva
- armas[]

Enemigo

- nombre
- atk
- def
- hp
- hpmax

Arma

- nombre
- atk
- def
- daño

Objeto

- nombre
- atk
- def
- hp

Getters / setters / toString

DATOS

- armas [] (todas las armas una vez instanciadas, se guardan aqui y cuando el jugador las obtiene se copian en su array de armas (método obtener arma))
- objetos [] (lo mismo que con armas)
- jugadores [] (se guardan aquí todos los jugadores que se vayan creando (metodo añadir jugador))
- elecciones [] (se guardan las elecciones que el jugador vaya haciendo en la historia)
- nombrebase [] (el nombre base de todas las armas en el mismo orden)
- atkbase [] (el atk base de todas las armas en el mismo orden)
- defbase [] (la def base de todas las armas en el mismo orden)
- dañobase [] (el dañobase de todas las armas en el mismo orden)

CRUD

Arma

- generarArmas () Con un for, genera todas las armas dentro del array armas de base de datos usando los datos base de los demás arrays.

Objetos

- generarObjetos () Lo mismo que armas.

Jugador

- addJugador (String nombre) Se le da el nombre pedido por teclado, se llama al constructor de jugador y se crea un jugador dentro de la base de datos de jugadores, se aumenta un contador para guardar el siguiente jugador en otro hueco.
- obtenerArma (Arma) Se le un arma de la base de datos y la copia o iguala en el array de armas del jugador, además la activa si es la única arma que tiene.
- cambiarArma (Arma) Se le da un arma del array del jugador, y la pone en la activa
- checkVidaMax () Comprueba si la vida actual del jugador es superior a la máxima, si es así la iguala a la máxima
- obtenerObjeto (Objeto) Igual que obtenerArma, se le da un objeto de la base de datos y lo copia en la slot libre del inventario del jugador.
- aplicarObjeto (Objeto) Se le da un objeto del inventario del jugador, lo activa, suma al jugador todas las características de este.. También llama al método checkVidaMax por si hay sobrecura. Elimina el objeto del inventario del jugador.
- atacar (tirada) Se le pasa la tirada de dados que el jugador haya hecho, la suma a su ataque, y al ataque de su arma activa. Después, debe llamar al método de defensa del enemigo, y compararlos, si el atk supera a la def del enemigo devuelve true, sino, false.
- defender() Se llama al método de tirar dados, se suma a su def y a la def de su arma activa. Devuelve el valor total. Usado para comparar en el ataque del enemigo.