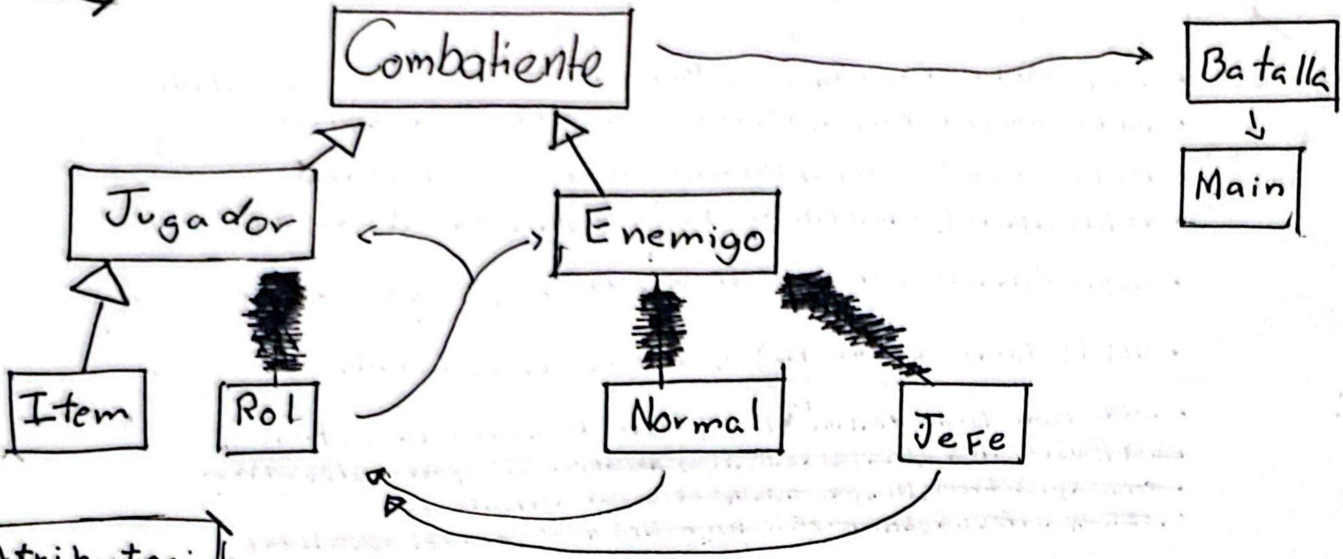
UML:



Atributos:

- Combatiente:
  - String(): Nombre -> Nombre del combatiente
  - int Vida -> vida del combatirnte, si es 0 muere
  - int Ataque -> Ataque del combatiente, daño
- Jugador:
  - RolJugador: rol -> rol del jugador
  - Array List(Item): Inventario -> Lista de objetos disponibles
  - ~~Nombre, String~~
  - ~~Vida~~
- Enemigo:
  - String(): tipo -> tipo de enemigo [ejemplo orco ó troll]
  - String: nombre
  - ~~B~~int: vida
  - int: ataque
- Normol:
  - [illegible]
  - ~~Defensa int -> Defensa del enemigo~~
  - ~~ta~~
  - No tiene
- Jefe:
  - ~~int DefensaExtra -> [illegible]~~
  - String: nombre
- Item:
  - Nombre: String -> Nombre del item
  - tipo: String -> tipo de ejecto
  - Efecto: int -> cantidad de vida o daño que da el efecto
  - Cantidad: int -> Cantidad de itens
- Rol:
  - String: nombre
  - Int: vida
  - Int: ataque
- Batalla:
  - Jugador jugador -> Jugador que participa
  - Array List (enemigos) -> Lista de enemigos
  - Array List() -> Accion -> ultima accion
  - Array List(Combatiente) -> Turno actual

Pablo Orantes 25563

Kevin Moreno 25730

Metodos

- void: mensaje Final -> mostrar mensaje final
- void: tomarturno -> toma el turno teniendo en cuenta el Combatiente.
- void recibir Daño -> (int cantidad).

Combatiente:

- ~~String~~ this nombre = nombre.
- this.vida = vida
- this ataque = ataque

- Void: Mostrar mensaje Inicio -> Mensaje al entrar a la batalla
- Void: Mensaje Victoria -> Mensaje al ganar la batalla
- Void: Mensaje Derrota -> Mensaje al perder el combate
- Void: Atacar (Combatiente objetivo) -> Resta vida al objetivo
- bool: EstaVivo () -> true si la vida > 0, = 0 muerte
- Void: Turno (Batalla tur) -> su turno en la batalla

Jugador:

- Void: tomar Turno (Batalla tur b) -> Menu de accion en su turno
- void: mensajeinicio
- void: mensajefinal
- List<Item> getInventario (return items).
- rol. get vida
- rol. getAtaque.

Enemigo:

- Void: habilidadEspecial (Combatiente Objetivo) -> Cada enemigo implementa su Poder especial

Normal:

- Nombre, vida, ataque.
- void: mensaje inicio (Override)
- void: mensaje final
- void: tomar turno
- void: habilidadespecial

Jefe:

- Void. habilidadEspecial (combatiente objetivo) -> habilidad especial mejoradas
- void mensajeinicio
- void habilidadEspecial (Combatiente objetivo)

Item:

- ~~void~~
- void: usar (Combatiente objetivo).
- String: getNombre (return nombre
- Int getcantidad (return cantidad).

Batalla:

- Void Iniciar () -> empieza batalla
- void Mostrar Estado () -> Muestra vida y el estado
- void siguiente turno () -> Pasa el turno al siguiente
- Void registrar Accion (String accion) -> true si han muerto
- bool batalla Fin () -> Guarda la ultima accion

Rol:

- Nombre String -> get nombre
- ~~Tipo String~~ ->
- Es un jefe void. ->
- Es uno normal void ->
- Vida int -> get vida
- Ataque int -> get Ataque
- Guerrero void() ->
- Aventurero void() ->