

## PERSONAJES (herencia):

- Identidad:
  - nombre: identificador mostrado en combate.
  - tipoClase: enum o texto (GUERRERO, MAGO, SACERDOTE).
- Vida:
  - vidaMax
  - vidaActual
- Recurso (según diseño):
  - recursoMax (mana/energía)
  - recursoActual
  - Si una clase no usa recurso, debe existir igualmente (por uniformidad) con valores 0 o una implementación que lo ignore.
- Estadísticas base:
  - ataqueBase: contribuye a daño físico o general.
  - poderMagico (opcional, pero recomendable si hay daño mágico).
  - defensaBase: reduce daño recibido.
  - precision / evasión (opcional).
- Equipamiento:
  - arma: referencia a Arma (obligatoria).
- Colecciones asociadas:
  - estadosActivos: lista de Estado (ArrayList).
  - hechizos: lista de Hechizo (ArrayList) o catálogo por clase.
  - cooldowns: diccionario/mapa si se implementa reutilización.
- Estado de combate:
  - vivo: no hace falta si se deriva de vida, pero se permite.
  - defendiendo (opcional).

### Guerrero:

- Geralt (Personaje BUENO)
- Imlerith (Personaje MALO)

**Rol:** daño físico sostenido + resistencia.

**Rasgos esperados:**

- vidaMax alta, defensaBase alta.
- recurso opcional (puede ser 0 y no usar hechizos, o energía simple).
- Acciones:
  - ataque con arma como acción principal.
  - opcional: habilidad de “Golpe poderoso” (daño más alto con coste o cooldown).
- Polimorfismo:
  - puede sobrescribir recibirDanio para aplicar reducción adicional si está defendiendo.
  - puede sobrescribir métodos de cálculo si se diferencian daño físico/mágico.

**Mago:**

- Yennefer (Personaje BUENO)
- Caranthal (Personaje Malo)

**Rol:** daño mágico y aplicación de estados (quemadura).

**Rasgos esperados:**

- vidaMax baja/media, defensaBase baja.
- recursoMax alto (mana).
- Acciones:
  - ataque básico (si se permite) más débil.
  - hechizos ofensivos como núcleo: daño directo y daño en el tiempo.

**Polimorfismo:**

- puede sobrescribir un método tipo calcularPoderHechizo() o similar.
- sus hechizos aplican Quemadura con duración y potencia.

**Sacerdote:**

Ciri (Personaje BUENO)

Eredin (Personaje MALO)

**Rol:** soporte, curación directa y curación en el tiempo (Renovar).

**Rasgos esperados:**

- vidaMax media, defensaBase media/baja.
- recursoMax medio/alto (mana).
- Acciones:
  - curación directa a un aliado.
  - curación por turnos (Renovar).
  - opcional: hechizo ofensivo ligero o disipación (extra).
- Polimorfismo:
  - puede sobrescribir reglas de curación (por ejemplo, bonificación al curar).
  - puede tener prioridad de IA distinta (curar al aliado más débil).

## Clase base Arma

### Atributos mínimos

- nombre: texto descriptivo.
- tipo: enum o texto (CUERPO\_A\_CUERPO / A\_DISTANCIA).
- danioBase: entero.
- modificador: opcional (por ejemplo, multiplicador o bonificación fija). • probCritico y multiplicadorCritico: opcional, para introducir variabilidad. • precision: opcional si se implementa fallo/esquiva.

### Métodos mínimos

- int calcularDanio(Personaje atacante, Personaje defensor): método central. Debe usar stats del atacante (por ejemplo, ataqueBase) y considerar, si se desea, stats del defensor (defensa, evasión). Este método debe ser **polimórfico**: subclases lo pueden sobrescribir.
- String descripcion(): texto corto para mostrar en selección o resumen. • Opcional: boolean golpea(...) si se separa precisión de daño.

### Reglas de diseño

- El arma no debe modificar directamente vida. Solo calcula daño.
- La reducción por defensa puede implementarse en:
  - recibirDanio del defensor (recomendado, centraliza defensa), o
  - calcularDanio del arma (válido si se quiere que algunas armas ignoren defensa). Debe elegirse una estrategia y mantenerla consistente.

## 4.2 Subclases obligatorias

Se requieren, como mínimo, dos subclases:

### a) ArmaCuerpoACuerpo

**Concepto:** armas de contacto (espada, hacha, maza).  
**Diferencias esperadas (ejemplos implementables)**

- Daño más estable o más alto en base.
- Puede beneficiarse más del ataqueBase del personaje.
- Opcional: puede tener crítico mayor o multiplicador mayor.
- Opcional: penalización contra ciertos estados (no obligatorio).

### Polimorfismo

- Sobrescribe calcularDanio(...) para aplicar su fórmula particular.

**b) ArmaADistancia**

**Concepto:** arco, ballesta, varita, rifle (según ambientación).