**Pruebas**

**01\_Unidad**

* Probar que se puede instanciar bien un objeto de una clase que herede Unidad
* Que no haya problemas con las asignaciones polimorficas

**02\_Capa**

* Probar que se puede decorar una clase con la clase decoradora Capa
* Probar que el poder de la unidad decorada sea el 90% del poder original
* Probar que la energía de la unidad decorada sea el doble de la original

**03\_Escudo**

* Probar que se puede decorar una clase con la clase decoradora Escudo
* Probar que la unidad decorada recibe solo el 40% del daño si tiene un escudo

**04\_Puñal**

* Probar que se puede decorar una clase con la clase decoradora Puñal
* Probar que la unidad decorada tiene 3 puntos mas de poder si usa un puñal
* Probar que la unidad decorada tiene 3 puntos menos de defensa (recibe 3 puntos mas de daño por ataque) si usa un puñal

**05\_Energia**

* Probar si un soldado deja de atacar si su energía llega a cero
* Probar si funciona la poción de agua
* Probar si decorar la unidad con una capa realmente le duplica la energía o si solo la decora. Para probar esto debería dejar de atacar al décimo golpe.

**06\_Posicion**

* Probar si funciona el método moverHacia()
* Probar si el método funciona tanto para desplazamientos solo en X como también para desplazamientos solo en Y.
* Probar si el método funciona cuando las unidades están en diagonal
* Probar si el método funciona cuando se le insertan números negativos como parámetros

**07\_PuedeAtacar**

* Probar si funciona el método puedeAtacar()
* Para arquero: probar si funciona cuando el arquero se queda sin flechas.
* Para soldado: probar si funciona cuando el soldado se queda sin energía.
* Para el caballero: probar si al atacar tres veces el caballo se vuelve rebelde y no le permite realizar un ataque.
* Probar los métodos correspondientes (cargarFlechas para arquero y tomarPocionAgua para soldado y caballero) para que las unidades puedan volver a atacar.

**08\_RecibirAtaque**

* Probar si se disminuye la salud de la unidad según lo que se le pase por parametro
* Probar si se permite dar un golpe con daño mayor a la salud de la unidad, y en tal caso, que la salud quede en cero y no en negativo.
* Probar si se permite dar un golpe con daño de tipo Double.