



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 Programación Avanzada (2020-1)

Tarea 01

Entrega

- **Avance de tarea**
 - **Fecha y hora:** miércoles 8 de abril de 2020, 20:00
 - **Lugar:** GitHub — Carpeta: Tareas/T01/
- **Tarea**
 - **Fecha y hora:** sábado 18 de abril de 2020, 20:00
 - **Lugar:** Repositorio personal de GitHub — Carpeta: Tareas/T01/
- **README.md**
 - **Fecha y hora:** lunes 20 de abril de 2020, 20:00
 - **Lugar:** Repositorio personal de GitHub — Carpeta: Tareas/T01/

Objetivos

- Aplicar conceptos de programación orientada a objetos (POO) para modelar y resolver un problema.
- Utilizar *properties*, clases abstractas y polimorfismo como herramientas de modelación.
- Comunicar diseños orientados a objetos a través de documentación externa.
- Procesar *input* del usuario de forma robusta, manejando potenciales errores de formato.

Índice

1. DCCriaturas Fantásticas	3
2. Flujo del programa	3
3. Menús	4
3.1. Menú de inicio	4
3.2. Menú de acciones	4
3.3. Menú cuidar DCCriaturas	4
3.4. Menú DCC	5
3.5. Pasar al día siguiente	6
4. Entidades	7
4.1. Magizoólogos	7
4.2. DCCriaturas	9
4.3. Alimentos	12
4.4. DCC	12
5. Archivos	14
5.1. magizoologos.csv	14
5.2. criaturas.csv	15
5.3. parametros.py	15
6. Bonus	16
6.1. Super Magizoólogo DocencioTareoHíbrido (3 décimas)	16
6.2. Peleas entre DCCriaturas (5 décimas)	17
7. Diagrama de clases y avance de tarea	18
8. .gitignore	19
9. Importante: Corrección de la tarea	19
10. Restricciones y alcances	19

1. DCCriaturas Fantásticas

Bienvenidos alumnos y alumnas de Programación Avanzada al mundo de la ~~programación~~ magia. Luego de pasar semanas en cuarentena, Lily416Potter decidió ver las películas de Harry Potter que no había visto nunca antes. Luego de ver las ocho películas sin interrupción y ver Animales Fantásticos, se enamoró de las pequeñas criaturas mágicas, por lo que se le ocurrió crear un mundo virtual con ellas.

~~Como no tiene tiempo por tantas cosas que tiene que hacer~~ Como tiene que crear la Tarea 1, se le ocurrió hacer que los estudiantes de Programación Avanzada creen este mundo mágico. Es por esto que se te ha dado la importante tarea de crear el mundo de DCCriaturas Fantásticas, y utilizando todos tus conocimientos sobre **programación orientada a objetos** serás capaz de hacer magia y dar vida a las DCCriaturas.



Figura 1: Logo de DCCriaturas Fantásticas.

2. Flujo del programa

DCCriaturas Fantásticas es un programa que simula el cuidado de criaturas día a día mediante interacción por consola con el usuario. Este tiene la opción de ser cierto tipo de Magizoólogo, una persona con habilidades especiales que le permite criar múltiples DCCriaturas, los animales mágicos de este mundo.

Adoptar una DCCriatura conlleva una gran responsabilidad, ya que son seres con poderes mágicos. Es necesario alimentarlas y cuidarlas, con la finalidad de mantener su salud, evitar que sufran enfermedades o se escapen. Para apoyar y regularizar el cuidado que el usuario brinda a sus DCCriaturas, existe el DCC. Esta es la entidad que provee tanto las nuevas DCCriaturas que pueden ser adoptadas por los Magizoólogos, como los alimentos que son necesarios para su cuidado. Adicionalmente, el DCC evaluará diariamente al Magizoólogo según su desempeño de cuidado, le pagará por sus resultados, pero también lo fiscalizará.

Varias acciones realizables por el usuario requieren el uso de su magia y en algunos casos también dinero (usando la moneda mágica, Sickles). Todos los días se comienza con cierta cantidad de energía, la cual se va agotando a medida que se realizan acciones. En cualquier momento el usuario tiene la opción de pasar al siguiente día, lo cual renovará su energía y aumentará su dinero según su desempeño, pero también permite la posibilidad de que distintos eventos le ocurran a sus DCCriaturas y de que sea fiscalizado por el DCC.

La sección [Menús](#) profundiza sobre la interacción por consola que se espera que DCCriaturas Fantásticas efectúe, mientras que la sección [Entidades](#) explica las entidades mágicas que existen en este mundo mágico y debes modelar. Puedes leerlas en el orden que estimes conveniente.

3. Menús

En esta tarea, al igual que en la anterior, la interacción se realizará a través de una serie de menús que tienes que programar considerando que sean **a prueba de errores de usuario**, y adicionalmente, que cada uno tenga la opción de **volver atrás** y **salir** (cada uno como opciones independientes). A continuación se explicará cuales son los menús mínimos a incluir, y se muestran en la [Figura 2](#) ejemplos de como se podrían visualizar. Queda a tu elección si quieres añadir más menús.

3.1. Menú de inicio

Primero es necesario ingresar al programa como un Magizoólogo. Al ejecutar DCCriaturas Fantásticas se deberá desplegar un menú de bienvenida en donde permita al usuario directamente seleccionar entre la opción **crear Magizoólogo**, la opción **cargar Magizoólogo** y salir¹.

En el caso de seleccionar la opción **Crear Magizoólogo**, el usuario deberá entregar un nombre de usuario no utilizado entre todos los Magizoólogos existentes, debe ser único en el sistema. Solo debe usar caracteres alfanuméricos y debe considerar indistintamente las mayúsculas de las minúsculas. Es decir, `lily416potter` y `Lily416Potter` son nombres de usuario válidos que corresponden al mismo usuario. Si se ingresa un nombre de usuario repetido o no válido, se le debe indicar su error y dar la opción de volver a intentarlo, volver atrás o salir. Luego, se le entrega la opción de seleccionar el tipo de Magizoólogo que será el usuario: **Docencio**, **Tareo** o **Híbrido**. Las características de cada tipo se detallan en la sección [Magizoólogos](#).

Luego de haber escogido el tipo de Magizoólogo, es momento para el usuario de adoptar su primera DCCriatura. Se deberá desplegar todas las especies de DCCriaturas como opciones (que se listan en la sección [DCCriaturas](#)). Luego de seleccionar la especie, el usuario debe ingresar un nombre único entre todas las DCCriaturas existentes. Similar al nombre de usuario, solo debe usar caracteres alfanuméricos, debe considerar indistintamente las mayúsculas de las minúsculas, se debe notificar al usuario en caso de un error y dar opciones para continuar. Al cumplir todo lo anterior se dirige al usuario al [Menú de acciones](#).

En cambio, en caso de seleccionar la opción de **Cargar Magizoólogo**, deberá pedirle al usuario el nombre del Magizoólogo preexistente. En el caso de que no exista, deberás mostrarle un mensaje que lo indique y darle la opción de volver a ingresar un nuevo nombre de usuario, volver atrás o salir. Luego de seleccionar al Magizoólogo correctamente, se dirige al usuario al [Menú de acciones](#).

3.2. Menú de acciones

Este es el menú principal que mostrará las acciones disponibles que el Magizoólogo puede realizar. Estas opciones pueden dirigir a otros menús o ser una acción en sí. Ninguna de las acciones en este menú tiene costo energético ni monetario, por lo que el usuario puede escogerlas cuanta veces quiera. Una vez resuelta la acción o menú, se vuelve a mostrar este menú (de acciones).

En base a lo anterior, las acciones a mostrar serán **Menú cuidar DCCriaturas**, **Menú DCC**, **Pasar al día siguiente**, además de la acción volver atrás y salir. Luego de seleccionar la acción se debe enviar al usuario al menú correspondiente o realizar la acción que corresponda.

3.3. Menú cuidar DCCriaturas

En este menú deberán aparecer todas las opciones que pueden realizar los Magizoólogos con sus DCCriaturas. Estas opciones son **Alimentar DCCriatura**, **Recuperar DCCriatura**, **Sanar DCCriatura** y **Usar habilidad especial**, además de las opciones volver atrás y salir.

¹Es la única excepción donde no hay opción de volver atrás.

En el caso de seleccionar la opción **Alimentar DCCriatura**, se debe dar la opción de seleccionar cuál de las DCCriaturas del usuario se desea alimentar y qué alimento entre los que posee se dará, para luego alimentarse. Por otro lado, al seleccionar la opción **Recuperar DCCriatura** se deberá mostrar las DCCriaturas que se han escapado, seleccionar una y realizar un intento de recuperarla. En el caso de seleccionar la opción **Sanar DCCriatura**, se deberá mostrar las DCCriaturas que se han enfermado, seleccionar una y realizar un intento de sanarla de ese estado. Finalmente, la opción **Usar habilidad especial** sólo se podrá utilizar una vez por usuario y su efecto cambia según el tipo de Magizoólogo.

Todas estas posibles acciones (menos volver atrás y salir) tienen un requisito en costo de energía mágica constante. En caso de seleccionarse una de estas opciones y no tener suficiente energía, no se efectúa la acción y se notifica al usuario. En caso de tener suficiente energía para la acción, una vez seleccionada se debe efectuar y descontar el costo energético, en este punto no es necesario un menú para volver atrás o salir. A su vez, cada acción tiene ligeras variaciones que dependen de la especialización del usuario. Los detalles de acciones y especializaciones se especifican en la sección [Magizoólogos](#).

3.4. Menú DCC

Este menú representa a la entidad del DCC en donde podrás **Adoptar DCCriaturas**, **Comprar alimentos** y **Ver estado de Magizoólogo y DCCriaturas**.

En el caso de seleccionar la opción **Adoptar DCCriaturas al DCC** se deja la posibilidad al Magizoólogo de adoptar una nueva DCCriatura a cambio de Sickles. Similarmente, al seleccionar la opción **Comprar Alimentos al DCC**, se deja la posibilidad al Magizoólogo de poder comprar cualquiera de los alimentos. Para ambas opciones, queda a tu criterio como generar las interacciones, pero se recomienda listar todas(os) las DCCriaturas (o alimentos) junto con su precio. Las DCCriaturas existentes se detallan en la sección [DCCriaturas](#) y alimentos en la sección [Alimentos](#)

Tanto adoptar una DCCriatura como comprar un alimento tienen un requisito monetario, y en caso de seleccionar adoptar o comprar sin los fondos suficientes, no se efectúa la acción y debe mostrarse un error explicando al usuario lo sucedido. En caso contrario, se efectúa la compra y se descuentan los fondos correspondientes. Los específicos de estas opciones se explicitan en la sección [DCC](#).

Ver estado de Magizoólogo y DCCriaturas es una acción sin coste de realizar. Consiste en imprimir en pantalla los datos actuales del Magizoólogo y de las DCCriaturas que adoptó, y así informarle al usuario su estado actual. Todos los atributos internos que deben mostrarse se listan en la sección [DCC](#).

***** Menú de inicio *****

Seleccione una opción:

- [1] Crear Magizoólogo
- [2] Cargar Magizoólogo
- [3] Salir

Indique su opción (1, 2 o 3): (input de usuario)

(a) Ejemplo Menú de inicio

***** Menú de acciones *****

Seleccione una opción:

- [1] Menú cuidar DCCriaturas
- [2] Menú DCC
- [3] Pasar al día siguiente
- [4] Volver atrás
- [5] Salir

Indique su opción (1, 2, 3, 4 o 5): (input de usuario)

(b) Ejemplo Menú acciones

***** Menú de cuidar DCCriaturas *****

Seleccione una opción:

- [1] Alimentar DCCriatura
- [2] Recuperar DCCriatura
- [3] Sanar DCCriatura
- [4] Usar habilidad especial
- [5] Volver atrás
- [6] Salir

Indique su opción (1, 2, 3, 4, 5 o 6): (input de usuario)

(c) Ejemplo Menú cuidar DCCriaturas

***** Menú DCC *****

Seleccione una opción:

- [1] Adoptar una DCCriatura
- [2] Comprar Alimentos
- [3] Ver estado de Magizoólogo y DCCriaturas
- [4] Volver atrás
- [5] Salir

Indique su opción (1, 2, 3, 4 o 5): (input de usuario)

(d) Ejemplo Menú DCC

Figura 2: Ejemplos de distintos menús en DCCriaturas Fantásticas.

3.5. Pasar al día siguiente

Al escoger esta opción, el programa debe simular el paso de un día. En un nuevo día ocurren diversos eventos importantes para el Magizoólogo. Cada día, cada criatura tiene una cierta probabilidad de enfermarse y cierta probabilidad de escaparse. Si lleva cierta cantidad de días sin comer, entonces cambia su hambre. Además, si una criatura está hambrienta o enferma, pierde cierta cantidad de salud. Una DCCriatura puede sufrir simultáneamente enfermedad, hambre y además escaparse. Luego de estos efectos, se simula la fiscalización del DCC sobre el usuario Magizoólogo. Este proceso calcula un nivel de aprobación y multas para el Magizoólogo que depende de sus atributos y criaturas. Los Magizoólogos que mantienen un buen nivel de aprobación mantienen o recuperan una **licencia** que lo certifica, pero en caso de obtener un valor bajo de aprobación, se quita dicha licencia, lo cual tiene consecuencias negativas para el usuario. Por otro lado, el DCC le paga cierta cantidad de dinero al día, pero luego también puede fijarle multas por la ocurrencia de ciertos eventos.

El programa debe imprimir el resultado de estos eventos, es decir, mostrar las **criaturas que se enfermaron**, las **criaturas que se escaparon** y las criaturas que pasaron a estar **hambrientas**. Además, se debe mencionar qué criaturas perdieron puntos de salud por estar enfermas o hambrientas.

También se debe mostrar los resultados de la fiscalización del DCC sobre el Magizoólogo, imprimiendo en pantalla el **nivel de aprobación** obtenido ese día e indicar el estado de la **licencia**, es decir, si la mantiene, si la recuperó, si la perdió ahora, o si la perdió anteriormente. También se debe mostrar en pantalla el **pago del DCC**, y luego las **multas recibidas**. Luego debes mostrar el nuevo saldo total en Sickles con los descuentos asociados a las multas. Puedes encontrar más detalle sobre la fiscalización en la subsección **DCC**. Finalmente se debe volver a desplegar el **Menú de acciones**. Un ejemplo de lo que debe mostrar tu programa al pasar un nuevo día es el siguiente:

```
¡¡Has pasado al día siguiente!!
*****
Resumen de los eventos de hoy:

Criaturas que enfermaron: Nifflenzini
Criaturas que escaparon:
Criaturas hambrientas: Augurinto
Nifflenzini perdió salud por enfermedad :(
Augurinto perdió salud por hambre :(
*****
Nivel de aprobación: 70
¡Felicidades! Continúas con tu licencia :D
Recibiste una multa porque Nifflenzini enfermó
```

```
No recibiste multas por criaturas que escaparon :D
El DCC te ha pagado 130 Sickles
Se te descontaron 70 Sickles en multas
Tu saldo actual es: 340 Sickles
```

4. Entidades

En esta sección se detallan las entidades que existen en este mundo mágico y que **debes modelar** utilizando los conceptos de **programación orientada a objetos**, *properties*, **polimorfismo** y **clases abstractas** donde corresponda.

Es importante notar, que tanto los Magizoólogos como las DCCriaturas mantienen su estado en distintas ejecuciones de DCCriaturas Fantásticas. Específicamente, una vez creado un nuevo Magizoólogo o DCCriatura, los datos internos de cada uno se almacenarán, por medio de archivos, para volver a utilizarse en futuras ejecuciones. Si los datos internos se actualizan, entonces también deben reflejarse estos cambios en futuras ejecuciones. Los valores iniciales de nuevas entidades se especifican en cada subsección.

4.1. Magizoólogos

Son quienes se encargan de cuidar las DCCriaturas con el objetivo de preservar su existencia. Todo Magizoólogo puede y debe alimentar a sus DCCriaturas, capturarlas en caso de que escapen y sanarlas en caso de que enfermen. Poseen diversos atributos que los caracterizan:

- **Nombre:** Es el identificador único del Magizoólogo que corresponde a un `str`. Ningún otro Magizoólogo puede tener el nombre de uno ya existente y este debe estar formado exclusivamente por caracteres alfanuméricos, y sin distinción entre mayúsculas y minúsculas.
- **Criaturas:** Son las distintas DCCriaturas que están a su cuidado. Todo magizoólogo tiene al menos una criatura bajo su cuidado. Estas se detallan en la sección [DCCriaturas](#).
- **Alimentos:** Son los productos esenciales para el cuidado de tus DCCriaturas. Cada nuevo Magizoólogo parte con `1` alimento aleatorio. Los alimentos existentes se detallan en la sección [Alimentos](#).
- **Sickles:** Los Sickles son la moneda principal en la economía del mundo mágico. La cantidad de Sickles corresponde a un `int`. Todo Magizoólogo **no puede** tener saldo negativo y todos comienzan con una cantidad inicial de `500` Sickles.
- **Energía actual:** Es el recurso energético que se regenera día a día para ejecutar acciones, puede estar entre `0` y su valor máximo y disminuye por cada acción que se realiza. Su valor inicial diario y máximo dependerá de cada Magizoólogo.
- **Licencia:** La licencia es un valor de tipo `bool` que indica si el Magizoólogo está certificado como cuidador. Cada nuevo Magizoólogo comienza con su licencia, pero si la pierde el DCC no le permitirá adoptar nuevas criaturas hasta que la recupere.
- **Nivel de aprobación:** El nivel de aprobación es un indicador de tipo `int` que corresponde a la calificación diaria otorgada por el DCC según el estado de las DCCriaturas que se posee. Varía entre `0` y `100`, y si es menor a `60` el Magizoólogo pierde su licencia. Esto se detalla más en la sección [DCC](#).

Los Magizoólogos también poseen otros atributos que influyen directamente en su desempeño como Magizoólogo, cada uno de estos atributo corresponde a un `int`. Cuando un Magizoólogo nuevo es creado, se determina aleatoriamente el valor de estos atributos de forma uniforme² dentro de dos valores específicos para cada tipo de Magizoólogo. Estos atributos son los siguientes:

²La función `randint` de la librería `random` te podría ser útil.

- **Nivel mágico:** Este valor indicará que tan apto es el Magizoólogo para alimentar, recuperar y sanar a sus DCCriaturas.
- **Destreza:** Cuando una de tus DCCriaturas huye, el nivel de destreza influirá en las posibilidades de recuperarla.
- **Energía total:** El nivel de energía inicial y máxima que determina la cantidad de acciones que un Magizoólogo puede hacer por día. Cada vez que comienza un día nuevo, la energía actual se recupera a este nivel de energía.
- **Responsabilidad:** Este atributo influirá en las posibilidades de enfermar y escapar que tienen las DCCriaturas que se poseen.

Acciones de Magizoólogos

Los detalles de cada acción que pueden realizar los Magizoólogos se presentan a continuación:

- **Adoptar DCCriaturas al DCC:** El Magizoólogo puede adoptar nuevas DCCriaturas al DCC, solo si posee actualmente su licencia. Además, esta acción está limitada por la capacidad monetaria del usuario que queda sujeta al precio de cada criatura.
- **Comprar Alimentos al DCC:** El Magizoólogo puede comprar alimentos al DCC. Esta acción solo está limitada por la capacidad monetaria del usuario que la fija el precio del alimento.
- **Alimentar DCCriatura:** El Magizoólogo puede decidir alimentar a una de sus DCCriaturas con alguno de sus alimentos, siempre y cuando posea alguno. En respuesta a esto, la DCCriatura puede atacar a su dueño. El costo energético de alimentar es de 5 puntos.
- **Recuperar DCCriatura:** Cuando un Magizoólogo intenta recuperar una de sus DCCriaturas, la probabilidad de éxito está determinada por la siguiente fórmula:

$$\min \left(1, \max \left(0, \frac{\text{destreza} + \text{nivel_magico_magizoologo} - \text{nivel_magico_criatura}}{\text{destreza} + \text{nivel_magico_magizoologo} + \text{nivel_magico_criatura}} \right) \right)$$

El costo energético de intentar capturar a una criatura es de 10 puntos.

- **Sanar DCCriatura:** Cuando un Magizoólogo intenta sanar a alguna de sus DCCriaturas que se encuentre enferma, la probabilidad de éxito en su sanación está determinada por la siguiente fórmula

$$\min \left(1, \max \left(0, \frac{\text{nivel_magico_magizoologo} - \text{salud_actual_criatura}}{\text{nivel_magico_magizoologo} + \text{salud_total_criatura}} \right) \right)$$

El costo energético de intentar curar a una criatura es de 8 puntos.

- **Usar habilidad especial:** Cada Magizoólogo tiene una habilidad especial que depende de su especialización. El costo energético de cualquiera de estas habilidades es de 15 puntos.

Cada Magizoólogo puede pertenecer a alguna de las siguientes clasificaciones, afectando tanto su comportamiento como sus atributos:

Magizoólogo Docencio

Los Magizoólogos Docencios tienen la peculiaridad de que al momento de alimentar a sus DCCriaturas logran aumentar en 5 puntos los puntos de salud totales de la criatura en cuestión. Por otra parte, siempre han tenido problemas al intentar recuperar alguna de sus DCCriaturas que han escapado, por lo que al

capturarlas merman la salud actual de la criatura en 7 puntos. Para más detalles sobre estos atributos, puedes ver la sección [DCCriaturas](#).

Los Magizoólogos de esta especialidad tienen la habilidad especial de saciar el hambre de todas sus criaturas sin la necesidad de darles alimentos y disminuyendo a cero la cantidad de días que llevan sin comer. Esto lo pueden hacer una sola vez como usuario.

El nivel mágico de los Docencios varia entre 40 y 60, su destreza oscila entre 30 y 40, mientras que su energía total estará entre 40 y 50. Por último, su responsabilidad variará entre 15 y 20.

Magizoólogo Tareo

Los Magizoólogos Tareos poseen la ventaja de que al alimentar a sus DCCriaturas poseen un 70 % de probabilidad³ de recuperar toda la salud actual de la criatura que recibió el alimento. Por otra parte, al recuperar a una de sus DCCriaturas que haya escapado esta no se verá afectada de forma negativa, a diferencia de los Docencios.

Estos Magizoólogos tienen la habilidad especial de poder recuperar a todas las criaturas que hayan escapado que aún no hayan sido recuperadas con un 100 % de efectividad. Esta habilidad solo se puede realizar una sola vez como usuario.

El nivel mágico de los Tareos varia entre 40 y 55, su destreza oscila entre 40 y 50, mientras que su energía total estará entre 35 y 45. Por último, su responsabilidad variará entre 10 y 25.

Magizoólogo Híbrido

Al alimentar una de sus DCCriaturas, los Magizoólogos Híbridos logran que la criatura en cuestión logre recuperar 10 puntos de salud. Además, al igual que los Tareos, cuando capturan una DCCriatura que haya escapado no provocan ningún tipo de efecto secundario en ella.

La habilidad especial de los Magizoólogos Híbridos es la de poder sanar a todas sus criaturas que se encuentren enfermas con un 100 % de efectividad. Esta acción solo puede realizarse una sola vez como usuario.

Por último, el nivel mágico de los Híbridos varia entre 35 y 45, su destreza oscila entre 30 y 50, mientras que su energía total estará entre 50 y 55. Por último, su responsabilidad variará entre 15 y 25.

4.2. DCCriaturas

Son las criaturas que cada Magizoólogo se dedica a cuidar. Todas las DCCriaturas pueden alimentarse, escapar e interactuar de forma negativa o positiva con su Magizoólogo. Poseen diversos atributos cuyos valores variarán según el tipo de criatura. Se caracterizan por tener las siguientes características:

- **Nombre:** Es el identificador único de cada criatura y corresponde a un `str`. No debe repetirse con el de ninguna otra criatura existente y este debe estar formado exclusivamente por caracteres alfanuméricos, y sin distinción entre mayúsculas y minúsculas.
- **Nivel mágico:** Valor de tipo `int` que indica cuán poderosa es la criatura. Este valor afecta en la cantidad de daño que puede hacer una criatura, en qué tan difícil es de capturar y en la cantidad de dinero que recibe el Magizoólogo por cuidarla. Varía de forma aleatoria y uniforme dentro de rangos específicos para cada **nueva** DCCriatura.

³Para determinar si el evento ocurre, es recomendable que usen la función `random.random` para obtener un número aleatorio entre 0 y 1, el cual deberás comprobar si es mayor o menor que la probabilidad del evento.

- **Puntos de salud total:** Es un valor del tipo `int` que indica la cantidad de vida total que posee la DCCriatura. Varía de forma aleatoria y uniforme dentro de rangos específicos para cada **nueva** DCCriatura.
- **Puntos de salud actual:** Es un valor del tipo `int` que indica la cantidad de vida que posee la DCCriatura actualmente. Comienza como el valor de puntos de salud total. Su valor no puede superar los puntos de salud total, y tiene como valor mínimo `1`.
- **Probabilidad de escape:** Valor de tipo `float` que indica la probabilidad base que tiene la criatura para escapar. Es distinto según tipo, algunas son más propensas a permanecer junto a su amo, mientras que otras son más rebeldes. Se mantiene entre `0` y `1`.
- **Probabilidad de enfermarse:** Valor de tipo `float` que indica la probabilidad base que tiene la criatura de enfermarse. Esta probabilidad varía por tipo de criatura y se mantiene entre `0` y `1`.
- **Estado de salud:** El estado de salud es un valor de tipo `bool` que indica si la criatura está enferma o no. En caso de que una criatura se enferme debe ser sanada por su amo, de lo contrario, su salud disminuirá en `7` puntos por cada día que tenga esta condición.
- **Nivel de hambre:** Es un `str` que indica el nivel de hambre de una criatura e influye en los puntos de salud de la misma criatura. Su valor puede ser `"satisfecha"` o `"hambrienta"`. Se puede pasar de `"satisfecha"` a `"hambrienta"` si no se come en una cierta cantidad de días determinados por el tipo de cada criatura. Por otra parte se puede pasar de `"hambrienta"` a `"satisfecha"` si la criatura consume alguno de los alimentos que se detallan en [Alimentos](#). Por cada día que una criatura esté `"hambrienta"` perderá `3` puntos de salud.
- **Días sin comer:** Valor de tipo `int` que representa la cantidad de días que lleva sin comer la criatura.
- **Nivel de agresividad:** Es un `str` que indica el nivel de agresividad de una criatura e influye en la probabilidad de que la criatura ataque al Magizoólogo al momento de que este le da alimentos. Puede ser `"inofensiva"`, `"arisca"` o `"peligrosa"`.

Acciones de DCCriaturas

Los detalles de cada acción que pueden realizar las DCCriaturas se señalan a continuación:

- **Alimentarse:** Todas las DCCriaturas comen al momento de que su Magizoólogo les da alimentos. Las DCCriaturas que estén `"hambrientas"` y sean alimentadas pasarán a estar `"satisfechas"`. Por otro lado, a la hora de comer existe una posibilidad de que la criatura ataque al Magizoólogo según la siguiente fórmula:

$$\text{mín} \left(1, \frac{\text{efecto_hambre} + \text{efecto_agresividad}}{100} \right)$$

donde `efecto_hambre` es igual a `0` si la criatura está `"satisfecha"` e igual a `15` si la criatura está `"hambrienta"`. Por otra parte, `efecto_agresividad` puede tomar el valor `0` si la criatura es `"inofensiva"`, `20` si es `"arisca"` y `40` si es `"peligrosa"`.

En el caso de ocurrir un ataque, el Magizoólogo perderá puntos de su **energía actual** determinados de la siguiente manera:

$$\text{máx} (10, \text{nivel_magico_magizoologo} - \text{nivel_magico_criatura})$$

- **Escaparse:** Las DCCriaturas tienden a ser un poco inquietas y suelen intentar escaparse, sobre todo si están hambrientas pues intentarían conseguir alimento por sus propias manos. La probabilidad total de que una criatura se escape al comienzo de un día es la siguiente:

$$\min \left(1, \text{prob_escaparse} + \max \left(0, \frac{\text{efecto_hambre} - \text{resp_magizoologo}}{100} \right) \right)$$

donde `prob_escaparse` corresponde al valor de **probabilidad de escape** de la DCCriatura, la **responsabilidad** del Magizoólogo corresponde a `resp_magizoologo` y `efecto_hambre` toma el valor de 20 si la criatura se encuentra "hambrienta" y 0 en otro caso.

- **Enfermarse:** Las DCCriaturas corren el constante riesgo de enfermarse. La probabilidad total de que una DCCriatura se enferme al comienzo de un día está determinada por la siguiente fórmula:

$$\min \left(1, \text{prob_enfermarse} + \max \left(0, \frac{\text{salud_total} - \text{salud_actual}}{\text{salud_total}} - \frac{\text{resp_magizoologo}}{100} \right) \right)$$

donde `prob_enfermarse`, `salud_actual` y `salud_total` corresponden a los atributos **probabilidad de enfermarse**, **puntos de salud actual** y **puntos de salud total** de la DCCriatura respectivamente. Por otra lado, `resp_magizoologo` corresponde a la **responsabilidad** del Magizoólogo.

Debes tener en cuenta que las probabilidades de que una DCCriatura enferme o escape son independientes entre sí y que una criatura **puede estar escapar y enfermar a la vez**.

Existen tres criaturas que tienen distinto comportamiento y características que las distinguen entre sí:

Augurey

El Augurey es un pájaro muy delgado, pequeño, de aspecto triste y con un plumaje extremadamente brillante. Es capaz de predecir la lluvia con muchos días de anticipación y le gusta volar solo cuando está muy bien cuidado. Es poco probable de que ataque a su dueño Magizoólogo.

Su nivel mágico varía entre 20 y 25, su probabilidad de escape es del 20 % y su probabilidad de enfermarse es igual a 30 %. Posee puntos de salud que oscilan entre 35 y 45, y pasa de estar "satisfecha" a "hambrienta" si no come en 3 días seguidos. Finalmente, su nivel de agresividad es "inofensiva".

Al comienzo de cada día, si el Augurey no tiene hambre, no se encuentra enfermo y tiene todos sus puntos de salud, volará para buscar un alimento y lo entregará como ofrenda a su dueño Magizoólogo.

Niffler

Un Niffler es una criatura pequeña, de pelaje sedoso y hocico largo, similar a un ornitorrinco. Es increíblemente inquieto y le encantan las cosas brillantes, por lo que suele robar Sickles para luego guardarlos en su pelaje. Tienden a atacar a su Magizoólogo.

Su nivel mágico varía entre 10 y 20, su probabilidad de escape es del 30 % y su probabilidad de enfermarse es igual a 20 %. Posee puntos de salud que oscilan entre 20 y 30, y pasa de estar "satisfecha" a "hambrienta" si no come en 2 días seguidos. Finalmente, su nivel de agresividad es "arisca".

Un Niffler, además de todas las características de una DCCriatura, tiene un **nivel de cleptomanía**, siendo este un valor del tipo `int` que toma un valor aleatorio uniforme entre 5 y 10 para cada nuevo Niffler. Si esta criatura está "satisfecha" al comienzo de un día, este le entregará Sickles al Magizoólogo, en

el caso de que se encuentre **"hambrienta"** le robará. La cantidad a de Sickles a entregar o robar esta determinada por la siguiente fórmula:

$$\text{nivel_cleptomania} \times 2$$

Erkling

Los Erklings son las DCCriaturas más peligrosas de todas. Con un aspecto parecido a los elfos, suelen pasar desapercibidos, pero son violentos.

Su nivel mágico varía entre 30 y 45, su probabilidad de escape es del 50 % y su probabilidad de enfermarse es igual a 30 %. Posee puntos de salud que oscilan entre 50 y 60, y pasa de estar **"satisfecha"** a **"hambrienta"** si no come 2 días. Finalmente, su nivel de agresividad es **"peligrosa"**.

Si esta criatura está **"hambrienta"**, al comienzo del día robará cualquiera de los alimentos que tenga el Magizoólogo y pasará a estar **"satisfecha"**. En el caso de que el Magizoólogo no tenga alimentos, el Erkling simplemente no hará nada.

4.3. Alimentos

Los alimentos son otra parte muy importante en el trabajo de los Magizoólogos. Estos son utilizados para mantener a las criaturas en buen estado y producen distintos efectos al ser consumidos. Tiene un único atributo: **efecto salud**. Corresponde a un valor de tipo **int** que indica en cuánto aumentarán los **puntos de salud actuales** de la criatura que lo consuma, y su valor depende del tipo de alimento. El efecto adicional que produce cada alimento no lo provoca ningún otro.

Tarta de Melaza

Este alimento es muy apetecido por los Magizoólogos para dárselo a las DCCriaturas, pues posee muy buenas propiedades respecto al resto de alimentos. Su efecto de salud es de 15, y si este alimento es consumido por un Niffler, existe un 15 % de probabilidades de disminuir su agresividad de **"arisca"** a **"inofensiva"** de forma **permanente**.

Hígado de Dragón

El Hígado de Dragón es conocido por ser un alimento con propiedades medicinales. Su efecto de salud es de 10, y si una criatura enferma consume un **Hígado de Dragón**, la criatura se sanará.

Buñuelo de Gusarajo

Los Buñuelos de Gusarajo es el alimento menos costoso de todos, pero su efecto de salud es de solo 5. Este alimento tiene la peculiaridad de tener un sabor amargo que no es del gusto de muchas criaturas, que provoca que la criatura rechace el alimento y el Magizoólogo lo perderá si lo intenta dar como alimento. Las posibilidades de que una criatura **coma** un Buñuelo de Gusarajo y no lo rechace es del 65 %.

4.4. DCC

Es la institución más seria y respetable del mundo mágico. Permite fiscalizar a los Magizoólogos y aplicarles multas en caso de que incurran en una falta. Además controlan el nivel de aprobación de los Magizoólogos, pudiendo quitarles su licencia. Debes preocuparte de que implemente las siguientes funcionalidades:

- **Calcular nivel de aprobación:** El DCC revisa todos los días el nivel de desempeño del Magizoólogo. Con el fin de mantener un alto estándar de calidad, si el nivel de aprobación obtenido es bajo 60, se debe quitar su licencia. Este se calcula con la siguiente fórmula:

$$\text{mín} \left(100, \text{máx} \left(0, \left[\frac{\text{n_criaturas_sanas} + \text{n_criaturas_retenidas}}{2 * \text{n_criaturas_totales}} \right] \times 100 \right) \right)$$

donde `n_criaturas_sanas` corresponde a la cantidad de **criaturas sanas** del Magizoólogo, `n_criaturas_retenidas` la cantidad de criaturas del Magizoólogo que **no han escapado** y `n_criaturas_totales` es la cantidad **total de criaturas** que pertenecen al Magizoólogo.

- **Pagar a los Magizoólogos:** Para que los Magizoólogos puedan desempeñar sus funciones el DCC paga una suma de Sickles todos días. Este paso siempre se realiza antes que la aplicación de multas, al comienzo de cada día. La cantidad a pagar se calcula con la siguiente fórmula:

$$\text{n_aprobacion} \times 4 + \text{cantidad_alimento} \times 15 + \text{nivel_magico_magizoologo} \times 3$$

donde `n_aprobación` corresponde al nivel de aprobación del Magizoólogo, `cantidad_alimento` la cantidad de alimento que tiene el Magizoólogo y `nivel_magico_magizoologo` el nivel nivel magico del Magizoólogo.

- **Fiscalizar a los Magizoólogos:** El DCC es muy riguroso en el control del desempeño de sus Magizoólogos y aplica multas, pero a veces algunos eventos se les escapan. El proceso de determinación de multas a aplicar siempre se realiza después de otorgar el pago diario. Puede aplicar multas en cada uno de los siguientes tipos de eventos **que ocurran en el día correspondiente**, con cierta probabilidad de efectivamente aplicar la multa:
 - **El escape de una criatura:** Se le pueden cobrar 50 Sickles con una probabilidad de 50 %.
 - **La enfermedad de una criatura:** Se pueden cobrar 70 Sickles con una probabilidad del 70 %.
 - **La salud baja de una criatura:** si una de las DCCriaturas llega al mínimo de salud, el Magizoólogo **debe** pagar la suma de 150 Sickles sin excepción.

Múltiples eventos del mismo tipo se **revisan de forma separada**, y en caso de no tener dinero suficiente para pagar una de estas multas, el Magizoólogo perderá su licencia y no se le cobrará el dinero correspondiente a esa multa.

- **Vender criatura a Magizoólogo:** Un Magizoólogo puede adquirir nuevas DCCriaturas a través del DCC siempre y cuando este **posea su licencia**. El costo de cada criatura es de 75 Sickles para un **Augurey**, 100 Sickles para un **Niffler** y 125 Sickles para un **Erkling**.
- **Vender alimentos a Magizoólogo:** El DCC se encargá de suministrar alimentos al Magizoólogo a cambio de los preciados Sickles. Una **Tarta de Melaza** tiene un costo de 10 Sickles, el **Hígado de Dragón** cuesta 15 Sickles y los **Buñuelos de Gusarajo** valen 3 Sickles.
- **Mostrar estado de Magizoólogo y DCCriaturas:** El DCC le muestra en pantalla todos los valores de atributos relevantes del usuario Magizoólogo. Específicamente, debe mostrar: su nombre, su cantidad de Sickles, su energía actual, si tiene o no su licencia, su nivel de aprobación actual, su nivel mágico, su destreza, responsabilidad, listar los alimentos comprados que le quedan y listar sus criaturas adoptadas. Para cada alimento, debe mostrar su tipo y efecto de salud. Para cada DCCriatura se debe mostrar su nombre, nivel mágico, puntos de salud actual, estado de salud, nivel de hambre y nivel de agresividad. El formato en que se muestre esta información queda a tu criterio.

5. Archivos

Para poder crear el mundo mágico de DCCriaturas Fantásticas, se te otorgarán los siguientes archivos en formato CSV, los cuales estarán separados por comas (,): **magizoologos.csv** y **criaturas.csv**. El primero almacena los Magizoólogos, y el segundo las DCCriaturas. Los alimentos comprados no tienen un archivo separado, estos se incluyen en **magizoologos.csv**.

Estos archivos deberán **modificarse** con cada ejecución nueva del programa. Ten en cuenta que estos archivos estarán en *encoding* UTF-8, por lo que se espera que estos archivos se puedan leer y/o modificar bajo el mismo formato a lo largo de la ejecución del programa.

Además deberás crear un archivo llamado **parametros.py**, el cual contendrá una serie de parámetros que serán de utilidad para la elaboración de tu tarea. Cabe mencionar que es **obligatorio importar** este archivo como un **módulo** y utilizar los parámetros contenidos en él.

5.1. **magizoologos.csv**

Contiene información sobre los Magizoólogos.

Nombre	Tipo de Dato	Descripción
Nombre	str	Nombre del Magizoólogo.
Tipo	str	Tipo de Magizoólogo. Determina algunos atributos de éste. Puede ser ' Docencio ', ' Tareo ', o ' Híbrido '.
Sickles	int	Dinero que posee el Magizoólogo.
Criaturas	list	Lista de nombres de criaturas que adoptó el Magizoólogo separados por punto y coma (;).
Alimentos	list	Lista de los alimentos que posee el Magizoólogo separados por punto y coma (;).
Licencia	bool	Boolean que indica si el Magizoólogo posee una licencia válida. True si es que está al día. False en caso contrario.
NivelMag	int	Nivel mágico que posee el Magizoólogo.
Destreza	int	Nivel de destreza que posee el Magizoólogo.
Energía	int	Nivel de energía total que posee el Magizoólogo.
Responsabilidad	int	Nivel de responsabilidad que posee el Magizoólogo.
HabEspecial	bool	Indica si está disponible el uso de la habilidad especial. True si es que se puede utilizar, False en caso contrario.

Un ejemplo de una fila sería la siguiente:

```
Lily416Potter,Tareo,500,Nifflenzini;Augurinto,Tarta de Melaza;Bueñuelo de Gusarajo,
True,50,48,41,32,True
```

La anterior corresponde al Magizoólogo Tareo Lily416Potter, que tiene 500 Sickles, dos criaturas a su cargo, dos alimentos comprados, y aún tiene su licencia y capacidad de usar su habilidad especial. Nota que la **energía actual** no es necesario almacenarla ya que se regenera diariamente.

5.2. criaturas.csv

Contiene información de las DCCriaturas.

Nombre	Tipo de Dato	Descripción
Nombre	<code>str</code>	Nombre de la criatura.
Tipo	<code>str</code>	Tipo de Criatura. Puede ser ' <code>Augurey</code> ', ' <code>Niffler</code> ' o ' <code>Erkling</code> '.
NivelMágico	<code>int</code>	Indica cuán poderosa es la criatura.
ProbEsc	<code>float</code>	Probabilidad de que la criatura se escape.
ProbEnf	<code>float</code>	Probabilidad de que la criatura se enferme.
EstadoSal	<code>bool</code>	Indica si la criatura se encuentra enferma o no. <code>True</code> si es que está enferma, <code>False</code> en caso contrario.
EstadoEsc	<code>bool</code>	Indica si la criatura se escapó o no. <code>True</code> si escapó, <code>False</code> en caso contrario.
PtsSaludTot	<code>int</code>	Cantidad de puntos de salud máximos que posee la criatura.
PtsSaludAct	<code>int</code>	Cantidad de puntos de salud actual que posee la criatura.
NivelHam	<code>str</code>	Indica el nivel de hambre de la criatura. Puede ser ' <code>satisfecha</code> ' o ' <code>hambrienta</code> '.
NivelAgr	<code>str</code>	Nivel de agresividad que tiene la criatura. Puede ser ' <code>inofensiva</code> ', ' <code>arisca</code> ' o ' <code>peligrosa</code> '.
DiasSinCom	<code>int</code>	Indica la cantidad de días que la criatura lleva sin comer.
NivelClept	<code>int</code>	Nivel de cleptomanía que tiene la criatura. Equivale a <code>0</code> en el caso de los ' <code>Augurey</code> ' y ' <code>Erkling</code> '.

Un ejemplo de una fila sería de la siguiente manera:

```
Nifflenzini,Niffler,15,0.3,0.2,False,True,30,22,satisfecha,arisca,1,9
```

En este ejemplo podemos ver que la criatura no se encuentra enferma, si se escapó, y aún le quedan puntos de salud restantes. Notar también que en los parámetros `float`, se utiliza un `'.'` en lugar de una `','` para separar la parte entera de la parte decimal.

5.3. parametros.py

A lo largo del enunciado se han ido presentando distintos números que permanecerán constante a lo largo de toda la ejecución de tu código, estos son conocidos como “parámetros” del programa y presentarán [ESTE FORMATO](#).

Para esta tarea deberás **crear y subir un archivo `parametros.py`** el cual contenga todos los parámetros nombrados anteriormente, además de todos los *paths* y cualquier otro valor constante que vayas a utilizar en tu código. Cada línea de este archivo representará una constante junto con su respectivo valor.

Para poder reconocer si un parámetro está bien creado o no, debes tener en cuenta una serie de consideraciones antes de escribirla. Un ejemplo de esto es asegurarse de que sus nombres sean descriptivos y fáciles de reconocer:

```
1 CIEN = 100 # mal parámetro
2 SICKLES_INICIAL = 100 # buen parámetro
3 MAX_DESTREZA_TAREO = 50 # buen parámetro
```

En el caso de que algún parámetro varíe de acuerdo a otros parámetros, una buena modelación sería la siguiente:


```

1  # En el caso de cambiar el valor de LARGO, ANCHO, o ALTO
2  # se cambiará el valor de VOLUMEN_CAJA
3  LARGO = 5
4  ANCHO = 6
5  ALTO = 10
6  VOLUMEN_CAJA = LARGO * ANCHO * ALTO

```

Es importante mencionar que en el archivo **parametros.py** deberás hacer uso de **todos** los parámetros almacenados y deberás **importarlos** correctamente. Cualquier uso del archivo **parametros.py** que no sea vinculado al almacenamiento de parámetros implicará un descuento en tu nota. Finalmente, cualquier parámetro que se encuentre *hardcodeado* dentro de tu código, también conllevaría a un descuento.

6. Bonus

En esta tarea habrá una serie de *bonus* que podrás obtener. Cabe recalcar que necesitas cumplir los siguientes requerimientos para poder obtener *bonus*:

1. La nota en tu tarea (sin *bonus*) debe ser **igual o superior a 4.0**⁴.
2. El *bonus* debe estar implementado **en su totalidad**, es decir, **no se dará puntaje intermedio**.

Finalmente, la cantidad máxima de décimas de bonus que se podrá obtener serán 8 décimas.

6.1. Super Magizoólogo DocencioTareoHíbrido (3 décimas)

Para obtener este *bonus* deberás implementar una nueva clase Magizoólogo DocencioTareoHíbrido, la cual deberá heredar de Magizoólogo Docencio, Magizoólogo Tareo y Magizoólogo Híbrido. Al pasar al día siguiente, si un Magizoólogo alcanza un **nivel de aprobación** igual a **100** entonces se convertirá en un Magizoólogo DocencioTareoHíbrido, notificando al usuario. El nuevo Magizoólogo DocencioTareoHíbrido deberá **conservar** los mismos atributos que tenía antes de la transformación. Además, podrá tener los mismos beneficios que tienen los Magizoólogo Docencios, Tareos e Híbridos por separado, siendo estos los que se ven a continuación:

- Al alimentar una criatura, existe una probabilidad del **70 %** de que recupere toda su salud.
- Al alimentar una criatura, ésta recupera **10** puntos de salud en caso de que no haya recuperado su total en el criterio anterior.
- Al capturar una DCCriatura, ésta no pierde salud.

En caso de que el Magizoólogo DocencioTareoHíbrido vuelva a tener un **nivel de aprobación** inferior a **100**, deberá **mantener** su estado de Magizoólogo DocencioTareoHíbrido, impidiendo volver a su **tipo** anterior. Para indicar este cambio, deberás actualizar el archivo **magizoologo.csv** y cambiar su tipo por **'Super'**.

Por último, el nuevo Magizoólogo DocencioTareoHíbrido tendrá la posibilidad de utilizar cualquiera de las tres **habilidades especiales** de cada tipo de Magizoólogo. Sin embargo, **solo** podrá ejecutar **una** de las tres habilidades especiales una **única** vez. Esto quiere decir de que se deberá dar la opción de elegir cuál habilidad especial utilizar, una vez seleccionada la opción **Usar habilidad especial** en el **Menú cuidar DCCriaturas**. Si el Magizoólogo ya utilizó su **habilidad especial** antes de la transformación, se deberá restaurar la posibilidad de utilizarla.

⁴Esta nota es sin considerar posibles descuentos.

6.2. Peleas entre DCCriaturas (5 décimas)

Debido a el aburrimiento de Lily416Potter durante la cuarentena la curiosidad de Lily416Potter por saber cuál de sus criaturas es la más poderosa, decidió darte la tarea de hacer una **simulación** de pelea entre dos DCCriaturas.

Para poner las cosas mas interesantes, el DCC te ha propuesto un reto en donde deberás apostar por cuál de tus criaturas es la mas fuerte. Para eso, en el [Menú cuidar DCCriaturas](#), deberás añadir la opción **Peleas entre DCCriaturas** en donde se mostrará un listado de todas las criaturas que están al cuidado de tu Magizoólogo⁵. Entre ellas, deberás elegir dos competidores, seleccionando a tu representante, mientras que la otra criatura será el representante del DCC⁶. Antes de comenzar la pelea, se deberá verificar si cumples con el requisito de 30 Sickles para realizar la apuesta. Luego de esto, se dará inicio a la batalla.

La Pelea entre ambas DCCriaturas se realizará mediante turnos. En cada turno se deberá notificar:

- La DCCriatura que está atacando a la otra DCCriatura.
- El daño que recibe la DCCriatura. En caso de que la DCCriatura haya esquivado el ataque, se deberá notificar en lugar del daño recibido.
- Los **puntos de salud** restantes que le queda a la DCCriatura atacada.
- Si la DCCriatura atacada queda con **1 punto de salud**, se finalizará el combate.

Como se señala, en cada turno una criatura atacará a la otra, realizando un determinado daño. Sin embargo, la otra criatura podrá **esquivar** el ataque en base a una función de probabilidad.

El **daño** de una DCCriatura a otra estará dada por la siguiente función:

- Si la DCCriatura atacante es **'inofensiva'**:

$$[criatura.nivel_magico \times 0.25]$$

- Si la DCCriatura atacante es **'arisca'**:

$$[criatura.nivel_magico \times 0.3]$$

- Si la DCCriatura atacante es **'peligrosa'**:

$$[criatura.nivel_magico \times 0.35]$$

Por otro lado, la DCCriatura atacada tendrá la probabilidad de **esquivar** el ataque del enemigo de acuerdo a la siguiente función:

$$(1 - criatura.prob_escapar) \times 0.5$$

La batalla finalizará cuando alguna DCCriatura queda con el mínimo de **puntos de salud** (**1**). En caso de que tu DCCriatura representante gane la pelea, deberás recibir un monto del **doblo** de Sickles apostados inicialmente. Por otra parte, si pierdes la batalla, el DCC se quedará con tu dinero. Para evitar cualquier conflicto al finalizar la batalla, el DCC **restaurará** los **puntos de salud** que tenían tus DCCriaturas antes de la batalla, para luego volver al menú anterior.

Un ejemplo de simulación de batalla sería el siguiente caso:

⁵En caso de poseer menos de dos DCCriaturas, se inhabilitará el acceso a esta modalidad.

⁶Recuerda que ambas DCCriaturas deben estar al cuidado de tu Magizoólogo.

```

¡¡El combate entre Nifflenzini y Augurinto ha comenzado!!
*****

Nifflenzini ha atacado a Augurinto
Augurinto ha recibido un daño de 6
Puntos de Salud restantes de Augurinto: 30

Augurinto ha atacado a Nifflenzini
Nifflenzini ha recibido un daño de 11
Puntos de Salud restantes de Nifflenzini: 26

Nifflenzini ha atacado a Augurinto
¡Augurinto ha esquivado el ataque!
Puntos de Salud restantes de Augurinto: 30

...

(El desenlace de esta épica batalla aún no ha sido escrita)
*****
¡Felicidades! tu DCCriatura ha ganado el combate! :D
El DCC te ha pagado 60 Sickles
Tus criaturas vuelven al estado de salud previo al combate
*****

```

7. Diagrama de clases y avance de tarea

En conjunto con el programa, se tendrá que realizar un diagrama de clases modelando las entidades necesarias para realizar el programa. Este diagrama se entregará en dos ocasiones:

Una versión preliminar, que corresponderá al avance de esta tarea. A partir de los diagramas entregados, se les brindará un *feedback* de cómo va la modelación de sus programas y además, les permitirá optar por **hasta 2 décimas** adicionales en la nota final de su tarea.

Luego de esto, junto a la entrega final, deberán entregar una versión final de su diagrama que **represente fielmente** la modelación del problema de su programa.

En ambos casos, el diagrama deberá:

- Entregarse en **formato PDF o de imagen**.
- Contener todas las clases junto con sus atributos y métodos.
- Contener todas las relaciones existentes entre las clases (agregación, composición y herencia).
- No es necesario indicar la cardinalidad ni la visibilidad (público o privado) de los métodos o atributos.

Para realizar el diagrama de clases te recomendamos utilizar draw.io, [lucidchart](https://lucidchart.com) o aplicaciones similares.

Sería conveniente que adjunten a su diagrama un documento con una explicación general de su modelación. Esto con el fin de ayudar la corrección del ayudante a reconocer su razonamiento.

Tanto el diagrama (en formato PDF o de imagen) como la explicación de su modelación (en formato [Markdown](#)) deben ubicarse en la misma carpeta de entrega de la tarea.

8. .gitignore

Para esta tarea **deberás utilizar un .gitignore** para ignorar los archivos entregados, este deberá estar dentro de tu carpeta T01/. Puedes encontrar un ejemplo de .gitignore en el siguiente [link](#).

Deberás ignorar el enunciado y todos los archivos de datos: `magizoologos.csv` y `criaturas.csv`. El archivo `parametros.py` **no debe ser ignorado debe ser entregado**.

Se espera que no se suban archivos autogenerados por entornos virtuales de Python, como por ejemplo: la carpeta `__pycache__`.

Para este punto es importante que hagan un correcto uso del archivo .gitignore, es decir, los archivos **no deben** subirse al repositorio **debido** al archivo .gitignore y no debido a otros medios.

9. Importante: Corrección de la tarea

Para esta tarea, el carácter funcional del juego será el pilar de la corrección, es decir, **sólo se corrigen tareas que se puedan ejecutar**. Por lo tanto, se recomienda hacer periódicamente pruebas de ejecución de su tarea y subir avances a sus repositorios remotos.

Cuando se publique la distribución de puntajes, se señalará con color amarillo cada ítem que será evaluado a nivel de código, todo aquel que no esté pintado de amarillo significa que será evaluado si y sólo si se puede probar con la ejecución de su tarea. En tu README deberás señalar el archivo y la línea donde se encuentran definidas las funciones o clases relacionados a esos ítems.

10. Restricciones y alcances

- Esta tarea es **estrictamente individual**, y está regida por el [Código de honor de Ingeniería](#).
- Tu programa debe ser desarrollado en Python 3.7.
- Tu programa debe estar compuesto por uno o más archivos de extensión `.py`.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier librería Python está prohibido. Pregunta en la *issue* especial del [foro](#) si es que es posible utilizar alguna librería en particular.
- Debes adjuntar un archivo `README.md` **conciso y claro**, donde describas los alcances de tu programa, cómo correrlo, las librerías usadas, los supuestos hechos, y las referencias a código externo. **Tendrás hasta 48 horas después del plazo de entrega** de la tarea para subir el README a tu repositorio.
- Tu tarea podría sufrir los descuentos descritos en la [guía de descuentos](#).
- Entregas con atraso de más de 24 horas tendrán calificación mínima (1,0).
- Cualquier aspecto no especificado queda a tu criterio, siempre que no pase por sobre otro.

Las tareas que no cumplan con las restricciones del enunciado obtendrán la calificación mínima (1,0).