



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

IIC2233 Programación Avanzada (I/2018)

Tarea 1

1. Entrega

- Diagrama de Clases
 - **Preliminar:** 25 de marzo del 2018, 23:59 horas.
 - **Final:** 08 de abril del 2018, 23:59 horas.
 - **Lugar:** un Google *form* (ver <https://goo.gl/forms/EhZVZYyDQRcMDDkn1>)
- Tarea
 - **Fecha/hora:** 08 de abril del 2018, 23:59 horas.
 - **Lugar:** GitHub – Carpeta: **Tareas/T01/**
- README.md
 - **Fecha/hora:** 09 de abril del 2018, 23:59 horas.
 - **Lugar:** GitHub – Carpeta: **Tareas/T01/**

2. Objetivos

- Desarrollar algoritmos para la resolución de problemas complejos.
- Aplicar conceptos de programación orientada a objetos (OOP) para modelar y resolver un problema.
- Utilizar correctamente las *properties* (*getter* y *setter*) cuando corresponde.
- Comunicar diseños orientados a objetos a través de documentación externa.
- Procesar *input* del usuario de forma robusta, manejando potenciales errores de formato.
- Trabajar con archivos de texto para leer, escribir y procesar datos.

Índice

1. Entrega	1
2. Objetivos	1
3. Introducción	3
4. <i>ChauCraft</i>	3
5. Entidades [28 %]	3
5.1. Galaxia	3
5.2. Planetas	3
5.3. Razas	4
5.3.1. Maestro	4
5.3.2. Aprendiz	5
5.3.3. Asesino	5
5.4. Edificios	6
6. Funcionalidades del programa [33 %]	6
6.1. Crear galaxia	6
6.2. Modificar una galaxia	6
6.3. Consultas sobre las galaxias	7
6.4. Jugar en la galaxia	7
6.4.1. Visitar Planeta	7
6.4.2. Guardar cambios	8
6.4.3. Volver atrás	8
7. Batalla [5 %]	8
8. Eventos [9 %]	9
9. Validación de <i>input</i> [8 %]	9
10. Base de datos [5 %]	9
10.1. Galaxia	10
10.2. Planetas	10
11. Consola [4 %]	10
12. Últimas consideraciones	10
13. Diagrama de clases [8 %]	11
14. Restricciones y alcances	11

3. Introducción

Hace algunos meses atrás, Chau, la hechicera, se despidió de este mundo y decidió retirarse de su vida como hechicera, para tomar el puesto de una Diosa capaz de jugar con las galaxias a su antojo. Durante estos meses se dedicó a crear nuevos planetas, razas y obligarlas a enfrentarse entre ellas, pero ha decidido tomarse unas vacaciones y dejarte a ti a cargo de seguir con su juego.

Como reemplazante, tu misión en esta tarea será crear un programa que permita simular todo lo anterior y mediante el uso de consola, permita al usuario (es decir, a ti) ponerse en el lugar de la Diosa, es decir, crear nuevas galaxias, modificar las ya existentes o jugar dentro de ellas mediante la construcción de edificios, generar soldados y comandar batallas entre diferentes planetas.

4. *ChauCraft*

ChauCraft es un simulador del trabajo de la Diosa Chau. Con esto, el usuario se pondrá en el puesto de la Diosa y será capaz de crear los planetas, poblarlo con diferentes razas y comandar batallas entre ellos con el fin de conquistar toda la galaxia creada. Para esto, deberás utilizar OOP donde corresponda para modelar la siguiente tarea y mediante el uso de la consola, permita crear planetas, modificarlos, poblarlos con alguna raza, etcétera.

5. Entidades [28 %]

5.1. Galaxia

En las galaxias se encuentran todos los planetas que pueden albergar alguna raza en su interior. Dentro de una galaxia es posible crear nuevos planetas, modificar los ya creados, conquistarlos y destruirlos. Cada galaxia es independiente a las otras; es decir, las acciones realizadas en una galaxia no tienen ninguna implicancia en las otras. Cada galaxia tiene sus propios planetas y reservas de recursos. En otras palabras, cada galaxia es una simulación distinta.

El programa puede manejar diversas galaxias y, por lo tanto, es necesario que cada galaxia tenga un identificador único. Este debe ser una palabra con mínimo 6 letras o números.

5.2. Planetas

Dentro de un planeta existen dos recursos: minerales y deuterio y con ellos es posible crear soldados, mejorarlos y crear edificios que ayudan en la defensa del planeta. Cada planeta permite albergar a una sola raza. Las características de un planeta son:

- **Nombre:** es el identificador único del planeta, debe ser una palabra con mínimo 6 letras o números. El nombre no se puede repetir con el nombre de un planeta de otra galaxia.
- **Raza:** indica la raza de su población, esta puede ser Maestro, Aprendiz o Asesinos.
- **Soldados actuales:** indica la cantidad de soldados actualmente en el planeta. Por defecto, un planeta nace sin ningún soldado.
- **Magos actuales:** indica la cantidad de Magos actualmente en el planeta. Por defecto, un planeta nace sin ningún mago.
- **Población máxima:** indica la cantidad máxima que se puede tener entre soldados y magos en un planeta. Esta cantidad depende de la raza de su población.

- **Tasa de minerales por segundo:** indica cuántos minerales se recolecta por segundo, este número se genera de forma aleatoria uniforme y puede ser entre 1 y 10.
- **Tasa de deuterio por segundo:** Indica cuánto deuterio se recolecta por segundo, este número se genera de forma aleatoria uniforme y puede ser entre 5 y 15.
- **Última recolección:** Indica la fecha de la última recolección de minerales y deuterio, en caso de no existir dicha fecha, se utiliza la fecha de creación. El formato de la fecha es: “año-mes-día hora:minuto:segundo”, por ejemplo: “2018-02-03 16:20:08”.
- **Nivel de ataque:** es un ponderador para aumentar el ataque básico del soldado. En el nivel 0, multiplica el ataque por 1. En el nivel 1, multiplica el ataque original por 1,2. En el nivel 2, multiplica el ataque original por 1,5 y en el nivel 3, multiplica el ataque original por 2. Cada mejora de nivel tiene un costo de 1.000 minerales, 2.000 unidades de deuterio y solo puede llegar hasta el nivel 3. Todo planeta parte con nivel 0 de ataque.
- **Nivel de economía:** es un ponderador que aumenta la tasa de minerales y tasa de deuterio por segundo. En el nivel 0, multiplica la tasa original por 1. En el nivel 1, multiplica la tasa original por 1,2. En el nivel 2, multiplica la tasa original por 1,5 y en el nivel 3, multiplica la tasa original por 2. Cada mejora de nivel tiene un costo de 2.000 minerales, 4.000 unidades de deuterio y solo puede llegar hasta el nivel 3. Todo planeta parte con nivel 0 de economía.
- **Conquistado:** indica si el planeta está conquistado o no.
- **Edificios:** dentro de un planeta pueden existir 2 edificios distintos: el cuartel y la torre de defensa. Sólo puede haber uno de cada tipo.
- **Evolución:** indica el nivel de evolución obtenido por un planeta (redondeado a la décima), este se obtiene como:

$$\text{Evolución} = \text{nivel de economía} + \text{nivel de ataque} + \frac{\text{cantidad de magos y soldados}}{\text{población total}} + \text{numero de edificios}$$

5.3. Razas

Las razas existentes son: *Maestro*, *Aprendiz* y *Asesino*. Cada raza tiene una habilidad única y la historia de cada raza repercute en el costo, ataque, vida y población máxima de los soldados y magos que se pueden generar en cada planeta.

Tanto el ataque como la vida de los soldados y magos está definida como un número aleatorio, que distribuye de forma uniforme dentro de un rango. Este rango depende exclusivamente de cada raza.

Cuando una raza logra conquistar un planeta, ésta realiza el siguiente grito de guerra: “Wubba Lubba Dub Dub, hemos logrado conquistar un nuevo planeta.” + **mensaje personalizado por raza**. Además, aumenta el ataque de cada soldado y mago participante de la batalla en 5 puntos y la vida en 10 puntos.

5.3.1. Maestro

Esta raza es la más antigua y con mayor conocimiento de todas. Dicho conocimiento los llevó a desarrollar grandes tecnologías y a entender que no es necesario tener una gran cantidad de población por planeta. Con su gran tecnología se volvió la raza con mayor poder, pero todo ese poder les hizo dejar de lado su vida. Además fueron capaces de crear un rayo con el potencial de matar a muchos invasores como forma de protección. Es la única raza con la capacidad de magos.

Los datos de los Maestros son:

- Población máxima entre soldados y magos: 100 unidades.
- Costo del soldado: 200 minerales y 300 unidades de deuterio.
- Costo del mago: 300 minerales y 400 unidades de deuterio.
- Ataque del soldado: Entre 60 y 80 puntos de daño.
- Ataque del mago: 80 y 120 puntos de daño.
- Vida del soldado: 200 y 250 puntos de vida.
- Vida del mago: 150 y 200 puntos de vida.
- Habilidad: Cuando su planeta es invadido, antes de empezar la batalla, tienen un 30 % de probabilidad de aniquilar a la mitad del ejército enemigo. Esta habilidad sólo se activa una vez.
- Grito de guerra cuando invaden un planeta: “¡Nuestro conocimiento nos ha otorgado una victoria más!”

5.3.2. Aprendiz

Esta raza busca el conocimiento que los Maestros tienen, pero aun así se deja llevar en varias ocasiones por sus instintos humanos. A raíz de esto, rechaza la doctrina de tener tan poca población por planeta. Sus armas siguen siendo primitivas, pero lo compensan con sus excelentes armaduras y medicinas que le otorgan una gran vida al soldado. Es una raza conocida por su capacidad de infiltrarse en las tropas enemigas y obtener recursos de ellos.

Los datos de los Aprendices son:

- Población máxima de soldados: 150 unidades.
- Costo del soldado: 300 minerales y 400 unidades de deuterio.
- Ataque del soldado: Entre 30 y 60 puntos de daño.
- Vida del soldado: Entre 600 y 700 puntos de vida.
- Habilidad: Sólo se puede utilizar cuando esta raza invade un planeta y se activa en cada ataque. Tiene la probabilidad de robar 200 minerales con una probabilidad del 70 %. Por ejemplo, si durante una invasión de esta raza a otro planeta, los soldados tienen 4 intentos de atacar. En cada intento, hay un 70 % de probabilidad de robar 200 minerales. Luego, en el caso de acertar las 4 veces, estos soldados se lograrán llevar 800 minerales. Si esta habilidad es utilizada por soldados del jugador, no hay límite de minerales para robar. En caso de que sea la máquina quien aplica esta habilidad, el límite de minerales a robar es la cantidad de minerales que dispone el jugador en ese momento.
- Grito de guerra cuando invaden un planeta: “¡Con una gran defensa y medicinas, nuestros soldados son invencibles!”.

5.3.3. Asesino

La última raza que nació en el universo, conocida por todos como la raza de las sombras. Es famosa por ser la raza con mayor población en el universo. Su forma de vivir los hace vulnerables a cualquier daño, lo cual hace que tengan menos vida que los Aprendices. A falta de tecnología, esta raza dominó el atacar los puntos vitales del enemigo, por lo que tienen la habilidad de realizar el doble de daño cuando invaden un planeta.

Los datos de los Asesinos son:

- Población máxima de soldados: 400 unidades.

- Costo del soldado: 100 minerales y 200 unidades de deuterio.
- Ataque del soldado: Entre 40 y 45 puntos de daño.
- Vida del soldado: Entre 250 y 270 puntos de vida.
- Habilidad: Sólo la pueden ocupar cuando invaden un planeta y en cada ataque hay un 40 % de probabilidad de realizar el doble de daño. Por ejemplo, si durante la invasión de los Asesinos a otro planeta, los soldados tienen 3 intentos y estos hacen en total un daño de 40, en cada intento hay un 30 % de hacer 80 de daño.
- Grito de guerra cuando invaden un planeta: “¡El poder de las sombras es lo único necesario para ganar estas batallas!”

5.4. Edificios

Existen dos tipos de edificios que todo planeta puede construir. Estos son el cuartel y la torre de defensa.

- **Cuartel:** Sólo cuando este edificio es construido, es posible crear soldados. Tiene un costo de 200 minerales y de 500 unidades de deuterio. Posee 5.000 puntos de vida.
- **Torre de defensa:** Este edificio ayuda a proteger al planeta durante una invasión. Tiene un costo de 150 minerales, 300 unidades de deuterio y un ataque de 1.000. Posee 2.000 puntos de vida.

6. Funcionalidades del programa [33 %]

El programa que deberán crear debe permitir la creación de galaxias y planetas, modificar los planetas ya existentes, realizar consultas y finalmente interactuar con los planetas. Para esto, se entregará un archivo *planetas.csv* con la información necesaria de cada planeta y un archivo *galaxias.csv* con la información de cada galaxia. En la sección 10 se detallará el formato e información que contiene cada archivo.

6.1. Crear galaxia

El usuario debe ser capaz de crear la cantidad de planetas que el quiera, donde elige el nombre del planeta y la raza de su población. Todos estos planetas deben ser guardados bajo el nombre de una galaxia que también es decidida por el usuario. Todos los planetas y galaxias creadas deben ser registradas en el archivo *planetas.csv* y *galaxias.csv* respectivamente.

Finalmente, antes de guardar, el usuario tiene que elegir 1 planeta para que quede conquistado desde el inicio. Todos los demás deberán ser poblados con el 75 % de su población máxima.

6.2. Modificar una galaxia

En esta opción, el usuario puede elegir cualquier galaxia que exista en los archivos *galaxias.csv* y dentro de ella, modificar cualquier planeta. Las capacidades de modificación incluyen:

- Agregar un nuevo planeta sin conquistar.
- Eliminar un planeta conquistado.
- Aumentar la tasa de minerales por segundo de los planetas no conquistados.
- Aumentar la tasa de deuterio por segundo de los planetas no conquistados.
- Aumentar la cantidad de soldados en los planetas no conquistados.

- Aumentar la cantidad de magos en los planetas no conquistados si es que la raza lo permite.

Todos los cambios a los planetas deben modificar al archivo `planetas.csv`.

6.3. Consultas sobre las galaxias

En esta opción, el usuario debe ser capaz de realizar las siguientes consultas:

- **Información general del usuario:** Se debe poder mostrar la cantidad de minerales, deuterio de cada galaxia junto con los planetas que tiene conquistado.
- **Información general del planeta:** Dado el nombre de un planeta, se debe poder mostrar la raza que posee, la población actual, la ultima recolección, el nivel de ataque, el nivel de economía, el estado de conquista, los edificios que posee, su nivel de evolución, a que galaxia pertenece y sus tasas de recolección.
- **Mejor galaxia:** Entre todas las galaxias, identificar la que tenga el mayor nivel de evolución. El nivel de evolución de una galaxia se calcula como el promedio de nivel de evolución de todos los planetas eliminando el planeta con mayor y menor nivel. Para esta consulta, solo se consideran las galaxias que tengan al menos 3 planetas.
- **Ranking de planetas:** De mayor a menor, el programa debe entregar los 5 planetas con mayor nivel de evolución y mostrar el nombre del planeta, galaxia a la que pertenece, nivel de evolución y raza de su población.

6.4. Jugar en la galaxia

Al momento de jugar con una galaxia, el usuario parte con 1.000 de minerales, 1.000 unidades de deuterio y con un planeta conquistado. El usuario puede ingresar a cualquier planeta que exista en la galaxia y sus acciones posteriores serán definidas en base a si el planeta está conquistado o no. Finalmente, el usuario debe tener la opción de guardar todo los cambios que han sucedido en su galaxia junto con los minerales y deuterio actual que tenga para poder jugar en otra ocasión.

6.4.1. Visitar Planeta

Si el planeta está conquistado el usuario puede recolectar los recursos, generar más unidades, construir alguno de los 3 edificios o realizar las mejoras de ataque o economía.

- **Construir edificio:** Si el usuario dispone con minerales y deuterio necesario, este puede solicitar construir el cuartel, torre de defensa y/o escudo.
- **Generar unidades:** Sólo si el planeta cuenta con el cuartel y el usuario tiene minerales y deuterio suficiente, puede generar más soldados hasta llegar al máximo de soldados por planetas.
- **Recolectar recursos:** Al momento de escoger esta opción, el programa debe calcular los segundos transcurridos entre la última recolección de recursos y el tiempo actual del computador y entregar la cantidad de minerales y deuterio correspondiente a la cantidad de segundos transcurridos y la tasa de recolección. En ese momento, la última recolección debe ser actualizada al tiempo actual del computador. Se puede utilizar el módulo `datetime` para trabajar los tiempos.
- **Realizar mejoras:** Si el usuario cuenta con minerales y deuterio necesario, puede realizar mejoras para aumentar el nivel de ataque o nivel de economía.

Si el planeta no está conquistado:

- **Invadir el planeta:** Debes escoger un planeta de los ya conquistados para que manden todo su ejército a destruir el ejercito del planeta no conquistado. En caso de reducir su ejército a 0, el planeta pasa a estar de tu control y es posible recolectar los recursos, construir soldados o edificios y realizar mejoras. En caso de que tu ejercito se reduzca a 0, el planeta queda sin conquistar. El sistema de batalla se especifica en la sección 7.
- **Comprar el planeta:** Si el usuario dispone de 1.000.000 minerales y 500.000 unidades de deuterio, puede conquistar de forma pacífica el planeta.

6.4.2. Guardar cambios

En esta opción, el sistema debe guardar el nuevo estado de la galaxia, esto incluye los nuevos edificios, unidades, últimas recolecciones de cada planeta, planetas conquistados, etcétera.

Dentro de los archivos CSV, no es necesario guardar el ataque y vida de cada soldado; sólo basta con el número de soldados y magos de cada planeta. Al momento de cargar los archivos, debe generarse nuevamente la vida y ataque de forma aleatoria en el rango correspondiente.

6.4.3. Volver atrás

El usuario debe tener la opción de salir de la galaxia sin guardar ningún cambio.

7. Batalla [5 %]

Las batallas entre dos planetas consiste en un pelea de turno entre ambos ejércitos. Cada ejército se comporta como una entidad cuyo ataque es la suma de los ataques de todos sus soldados y su vida es la suma de la vida de todos sus soldados. Por ejemplo, si dispones de 10 soldados cuyo ataque es 20 y vida es 40, este ejercito se comportará como una entidad con 200 de ataque y 400 de vida.

Si el planeta invadido dispone de una torre, el ataque de su entidad aumenta en el daño de la torre. Usando el ejemplo anterior, si el planeta de los 10 soldados dispone de una torre y el ataque de este edificio es 1.000, la entidad tendrá al final 1.200 de ataque.

En cada turno, un ejército ataca al otro y es en este momento en donde se aplican las habilidades de los asesinos o aprendices, es decir, hacer doble daño o robar minerales. Cuando una entidad ataca, la vida de la otra entidad (ejército enemigo) se reduce según la cantidad de daño que tiene el atacante. En el momento que una entidad llegue a tener vida menor o igual a 0, el ejército que representa es derrotado y al ganador se le calcula la cantidad de soldados sobrevivientes, para esto se realiza el siguiente cálculo:

$$\text{magos vivos} = \text{mínimo} \left(\left\lfloor \frac{\text{vida del ejército}}{\text{vida de un mago}} \right\rfloor, \text{número de magos inicial} \right)$$

$$\text{soldados vivos} = \left\lfloor \frac{\text{vida del ejército} - (\text{magos vivos} \cdot \text{vida de un mago})}{\text{vida de un soldado}} \right\rfloor$$

Cada vez que se ataca, es necesario imprimir en pantalla la cantidad de vida de cada ejército, cuánto daño se aplicó y si es que se utilizó una habilidad. Cuando se termine la batalla, se debe indicar en pantalla al ejército ganador (el invasor o el invadido) y la cantidad de magos y soldados sobrevivientes.

Si los Asesinos están atacando, tienen el primer turno. En otro caso, el invadido parte siendo el primero en este sistema de turnos.

8. Eventos [9 %]

Dentro del juego, cada vez decides visitar un nuevo planeta, puede ocurrir alguno de estos eventos:

- **Invasión del archimago:** este desolado soldado fue producto de la descendencia entre un Aprendiz y un Asesino. Un soldado así no era aceptado y fue exiliado. Cada cierto tiempo invade, al azar, uno de tus planetas conquistados y se enfrenta a tu ejército. Este tiene la habilidad única de atacar 3 veces seguido en cada turno y utilizar sus poderes de Aprendiz y Asesino para intentar atacar el doble y robarte minerales en cada ataque. Si el Archimago logró destruir todo tu ejército, el planeta deja de estar conquistado y antes de irse, deja el planeta con todas sus mejoras al máximo, todos sus edificios y con la población al máximo, para que te sea difícil recuperarlo. Este Archimago tiene 10.000 puntos de vida y hace un daño de 400 puntos de ataque.
- **Asteroide:** una gran masa de tierra solidificada cae en alguno de tus planetas conquistados y le quita entre 1.500 y 2.500 puntos de vida a todo edificio existente de dicho planeta y mata a la mitad del ejército. Si un edificio llega a tener 0 puntos de vida, este será destruido. El número se genera de forma aleatoria, siguiendo una distribución uniforme.

Existe un 20 % de probabilidad de que suceda algún evento cada vez que se decide visitar un planeta, y dentro de los dos eventos ambos tienen las mismas probabilidades de ocurrir.

9. Validación de *input* [8 %]

Al momento de modificar una galaxia (sección 6.2) existen ciertas restricciones para que la edición sea válida.

La tasa de minerales por segundo, deuterio por segundo o cantidad de soldados de un planeta no conquistado sólo son posibles de *setear* o modificar si es que se cumplen las siguientes condiciones:

- Son modificaciones que no superan la capacidad máxima permitida por el planeta.
- Son cantidades enteras mayores o iguales a 0.

El programa no debe avisar al usuario que la edición no es posible; solamente no debe *setear* los valores si no cumplen estas restricciones.

10. Base de datos [5 %]

Para el correcto funcionamiento del programa, *ChauCraft* cuenta con bases de datos previamente pobladas con la información actual de galaxias y planetas. Éstas bases de datos se encuentran en formato CSV (del inglés, *comma-separated values*). **Para leer y manipular estos archivos su programa debe leer las bases de datos sin usar el módulo `csv`.** Cabe destacar que, en el formato CSV, **el orden de las columnas puede variar**. Este orden es dado por la primera fila del archivo, el cual indica el nombre de cada columna y el tipo de dato que corresponde.

IMPORTANTE: cualquier modificación que se haga en alguno de estos archivos debe verse reflejado en todo momento, excepto cuando se está jugando en la galaxia, donde los cambios se deben guardar al elegir la opción de guardar cambios.

Junto con el enunciado, se entrega un *script* llamado `generar_archivos.py`. Cuando se ejecuta, se crean los archivos necesarios. Es recomendable probar su programa con varias ejecuciones del *script* para comprobar el correcto funcionamiento de la lectura y escritura de archivos `csv`. Las bases de datos se describen en las siguientes subsecciones.

10.1. Galaxia

Esta base de datos se encuentra en el archivo `galaxia.csv` y cada fila representa una galaxia del sistema. Cada columna del archivo representa lo indicado en la siguiente tabla.

Cuadro 1: `galaxia.csv`

Nombre	Tipo de dato	Comentarios
nombre	string	Nombre de la galaxia.
minerales	int	Cantidad de minerales en posesión.
deuterio	int	Cantidad de deuterio en posesión.

10.2. Planetas

Esta base de datos se encuentra en el archivo `planetas.csv` y cada fila representa un planeta junto a la información de este. Cada columna del archivo representa lo indicado en la siguiente tabla.

Cuadro 2: `planetas.csv`

Nombre	Tipo de dato	Comentarios
nombre	string	Nombre del planeta.
galaxia	string	Nombre de la galaxia a la que pertenece.
raza	string	Tipo de raza que habita el planeta.
magos	int	Cantidad de magos en el planeta.
soldados	int	Cantidad de soldados en el planeta.
tasa_minerales	int	Tasa de minerales por segundo del planeta.
tasa_deuterio	int	Tasa de deuterio por segundo del planeta.
ultima_recoleccion	datetime	Fecha de la última recolección de recursos.
nivel_ataque	int	Nivel de ataque del planeta.
nivel_economia	int	Nivel de economía del planeta.
conquistado	bool	Determina si el planeta está conquistado.
torre	bool	Indica si el planeta posee una torre.
cuartel	bool	Indica si el planeta posee un cuartel.

11. Consola [4 %]

Toda interacción del usuario con *ChauCraft* debe ser mediante la consola de Python y debe ser robusta a *inputs*; es decir, el programa **no puede generar error** si el usuario pone un *input* no esperado.¹

12. Últimas consideraciones

- No puedes editar el *script* para generar los archivos CSV.

¹En otras palabras, el programa no se puede caer *nunca*.

- Para una mejor corrección, se recomienda indicar en el `README.md` en qué archivo y línea se encuentra cada constante mencionada en el enunciado. Por ejemplo,
 - “Ataque raza Asesino: módulo `asesino.py` línea 4.”
 - “Vida raza Asesino: módulo `asesino.py` línea 5.”
 - “Ataque raza Aprendiz: módulo `aprendiz.py` línea 7.”
- Si considera que alguna constante como la vida de los soldados, ataque del archimago, probabilidad de efectuar una habilidad, etc. son muy altas o bajas, puede modificarlas.

13. Diagrama de clases [8 %]

Junto con el programa pedido, se debe entregar, mediante un **Google form**, un diagrama de clases con todo el modelamiento del problema. Esto incluye las clases junto con sus atributos, métodos y *properties*, y todas las relaciones existentes entre estas (asociación, composición y herencia). No es necesario indicar la cardinalidad.

14. Restricciones y alcances

- Esta tarea es **estrictamente individual**, y está regida por el *Código de honor de la Escuela*: haz clic aquí para leer.
- Tu programa debe ser desarrollado en Python v3.6.
- Tu código debe seguir la guía de estilos descrita en el PEP8.
- Si no se encuentra especificado en el enunciado, asume que el uso de cualquier librería de Python está prohibida. Pregunta en el foro si es que es posible utilizar alguna librería en particular.
- El ayudante puede castigar el puntaje² de tu tarea, si le parece adecuado. Es recomendable ordenar el código, siendo lo más claro y eficiente posible en la creación de algoritmos.
- Debes adjuntar un archivo `README.md` donde comentes sus alcances y el funcionamiento del sistema (*i.e.* manual de usuario) de forma concisa y clara. **Tendrás hasta 24 horas después de la fecha de entrega** de la tarea para subir el `README.md` a tu repositorio.
- Crea un módulo para cada conjunto de clases. Divídelas por las relaciones y los tipos que poseen en común. **Se descontará hasta 5 décimas si se entrega la tarea en un solo módulo**³.
- Cualquier aspecto no especificado queda a tu criterio, siempre que no pase por sobre otro.

Las tareas que no cumplan con las restricciones señaladas en este enunciado tendrán la calificación mínima (1.0).

²Hasta -5 décimas.

³No tomes tu código de un solo módulo para dividirlo en dos; separa el código de forma lógica