



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

IIC2233 Programación Avanzada (2023-1)

Tarea 1

Entrega

- **Avance de tarea**
 - **Fecha y hora:** viernes 14 de abril de 2023, 20:00
 - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T1/`
- **Tarea y README.md**
 - **Fecha y hora:** miércoles 26 de abril de 2023, 20:00
 - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T1/`

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. <i>DCCavaCava</i>	3
2. Flujo del programa	3
3. Menús	4
3.1. Menú de Inicio	4
3.2. Menú Principal	5
3.2.1. Simular día Torneo	5
3.2.2. Mostrar Estado Torneo	6
3.2.3. Menú ítems	7
3.2.4. Guardar partida	7
4. Entidades	8
4.1. Torneo	8
4.2. Arena	9
4.2.1. Tipos de arenas	10
4.3. Excavador	10
4.3.1. Tipos de excavadores	11
4.4. Ítem	12
4.4.1. Consumibles	12
4.4.2. Tesoros	12
5. Archivos	13
5.1. excavadores.csv	13
5.2. arenas.csv	14
5.3. consumibles.csv	14
5.4. tesoros.csv	15
5.5. DCCavaCava.txt	15
5.6. parametros.py	15
6. <i>Bonus</i>	16
6.1. Guardar Partida (3 décimas)	16
7. Diagrama de clases	17
8. Avance de tarea	17
9. .gitignore	17
10. Entregas atrasadas	18
11. Importante: Corrección de la tarea	18
12. Restricciones y alcances	18

1. *DCCavaCava*

Tras haber logrado defender exitosamente el castillo de **Lily416** y su ejército de tortugas, el malvado **Dr. Pinto** se ha rendido ante la idea de conquistar el reino. Avergonzado de su fracaso, decide embarcar un nuevo viaje en búsqueda de nuevos horizontes.

Luego de unas semanas de viaje, nuestro antagonista acaba llegando a un reino sucumbido al caos tras la ausencia de un líder. Ante esta oportunidad, el malvado **Dr. Pinto** aprovecha para autoproclamarse como nuevo Rey, pero no antes de que nuestro héroe, **Sir Hernan4444**, llegara para impedirlo. Para resolver este conflicto, deciden emprender un viaje a las costas del castillo.

Sin embargo, al momento de llegar a la playa, sienten la **suma e inexplicable urgencia** de tener que cavar un ~~hoyo~~ pozo en la arena. De esta forma, nuestros queridos ~~profesores~~ personajes deciden realizar el gran torneo *DCCavaCava*, donde aquel que cave el pozo más profundo será el ganador. **Sir Hernan4444**, desconfiado de los trucos del malvado **Dr. Pinto**, te contacta a ti como experto programador, para que le ayudes a conformar el mejor equipo de **excavadores** y así ganar el campeonato.

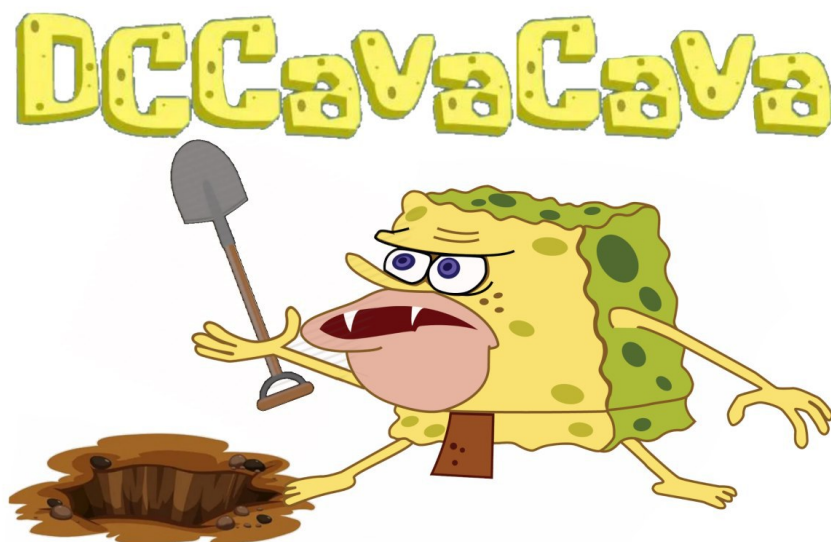


Figura 1: Logo del torneo *DCCavaCava*

2. Flujo del programa

Tu misión consiste en simular *DCCavaCava*, donde deberás comandar las decisiones de Sir Hernan4444 y cavar un pozo más profundo que el malvado Dr. Pinto. Para ello, contarás con un equipo de **Excavadores** y un tiempo de [DIAS_TOTALES_TORNEO](#)¹ días para cavar la mayor cantidad de metros posible. En cada día, deberás tomar distintas decisiones, conseguir recursos y superar diversos obstáculos que se te presentarán en el torneo.

Los **Excavadores** deberán cavar en la **Arena** del torneo. Al momento de cavar, existirá la probabilidad de que ocurra algún **Evento especial** que cambie el estado y las propiedades de la **Arena**, por lo que deberás diseñar una correcta estrategia para adaptarte y superar estas adversidades. Además, mientras los **Excavadores** cavan, podrás encontrar diferentes **Ítems** que podrás utilizar al finalizar el día. Estos **Ítems** podrán ser **Consumibles**, los cuales proporcionarán distintos beneficios a tus **Excavadores**, o

¹Las palabras escritas en [ESTE_FORMATO](#) son parámetros que tendrás que definir e importar desde el archivo [parametros.py](#)

bien pueden ser **Tesoros**, cuyos efectos serán especiales y permitirán cambiar el estado de la **Arena** o añadir nuevos **Excavadores** a tu equipo.

La interacción con el programa será exclusivamente mediante los **Menús** impresos en la consola. Al iniciar el programa se abrirá el **Menú de Inicio**, donde el jugador podrá empezar una nueva partida, cargar partida o salir. En caso de empezar una partida, se cargará un tipo de **Arena** determinada por `ARENA_INICIAL`², junto a un equipo inicial de `CANTIDAD_EXCAVADORES_INICIALES`³ **Excavadores**. Posterior a ello, se desplegará automáticamente el **Menú Principal**. En el caso de seleccionar cargar partida, se tomará la última partida guardada en base al archivo `DCCavaCava.txt` y se desplegará el **Menú Principal**. Si el archivo no existe, se le notificará al usuario por medio de la consola y se volverá a mostrar el **Menú de Inicio**. Dentro del **Menú Principal** podrás acceder a distintas opciones. Entre ellas la de simular un día del torneo, mostrar el estado del torneo, ver los ítems que tienes en tu mochila y la opción de guardar partida.

3. Menús

Para esta tarea, la simulación del torneo *DCCavaCava* será realizada a través de una serie de Menús en la consola. Estos menús deben ser a prueba de **todo tipo de errores** de usuario, es decir, en caso de que se ingrese un *input* no válido, se deberá advertir correctamente al usuario y volver a desplegar las opciones del menú. Adicionalmente, cada uno debe tener la opción de **volver atrás y salir**, a menos que se diga lo contrario. A continuación se explicarán los menús mínimos a incluir.

3.1. Menú de Inicio

Al iniciar el programa se deberá desplegar un Menú de inicio, en el cual se den las opciones de **Nueva partida**, **Cargar partida** y **Salir del programa**. Dado que es el primer menú que se muestra en pantalla, no es necesario implementar la opción de **volver atrás**.

En el caso de que se seleccione la opción de **Nueva partida**, se deberá seleccionar **automáticamente** la **Arena** y los **Excavadores** iniciales. En el caso de la **Arena**, deberás seleccionar de manera **aleatoria**⁴ una arena dentro del archivo `arenas.csv` tal que el tipo de esta sea igual al del parámetro `ARENA_INICIAL`. Para el caso de los **Excavadores**, deberás asignar al equipo inicial un número de `CANTIDAD_EXCAVADORES_INICIALES` excavadores de manera aleatoria a partir del archivo `excavadores.csv`. Luego de esto, deberás desplegar el **Menú Principal**

Para el caso de seleccionar **Cargar partida**, se deberá extraer la información dada en `DCCavaCava.txt` y mostrar el **Menú Principal**. En caso de que el archivo **no exista**, deberás informar al usuario en consola y volver a mostrar el **Menú de Inicio**. Finalmente, al ingresar la opción de **Salir**, se cerrará el programa.

```
*** Menú de Inicio ***
-----
[1] Nueva partida
[2] Cargar partida
[X] Salir
```

Indique su opción (1, 2 o X):

Figura 2: Ejemplo de Menú de Inicio

²Este parámetro deberá ser un `str` entre los siguientes: `'normal'`, `'mojada'`, `'rocosa'` o `'magnetica'`, los cuales están dados en `Arena`.

³Pueden asumir que este valor será mayor a 0 y menor que la cantidad total de excavadores dados en `excavadores.csv`

⁴Para esto, puedes revisar la librería `random`.

3.2. Menú Principal

En este menú se presentarán las acciones que se puede realizar en el *DCCavaCava*. Se tendrán las siguientes opciones principales: [Simular día Torneo](#), [Mostrar Estado Torneo](#), [Menú ítems](#) y [Guardar partida](#). Además, deberás mostrar el **día actual del torneo** y los **días totales**, junto al **tipo de arena** que está presente en el momento.

```
*** Menú Principal ***
-----
Día torneo DCCavaCava: 1/7
Tipo de arena: Normal

[1] Simular día torneo
[2] Ver estado torneo
[3] Ver ítems
[4] Guardar partida
[5] Volver
[X] Salir del programa

Indique su opción (1, 2, 3, 4, 5 o X):
```

Figura 3: Ejemplo de Menú de Acciones

3.2.1. Simular día Torneo

El objetivo de *DCCavaCava* es superar la cantidad de metros cavados por el malvado Dr. Pinto, el cual está dado por [METROS_META](#). Para lograr esto, deberás simular y cavar durante [DIAS_TORNEO](#) días para alcanzar la meta deseada. Al seleccionar esta opción, se simulará **un día** del torneo *DCCavaCava*, en el cual se efectuarán los siguientes eventos en el respectivo orden.

1. **Cavar:** Cada excavador que **no** esté descansando tiene que **cavar** una cantidad de metros en el día. Se deberá mostrar la cantidad de **metros cavados por cada uno**, que será conseguido mediante la fórmula **cavar** de la sección [Excavador](#), junto al nombre correspondiente. Si el excavador se encuentra descansando, se debe mostrar el nombre e indicar su estado. Finalmente se debe mostrar la cantidad de **metros totales** cavados en el día.
2. **Encontrar ítems:** En el día, cada excavador tiene la posibilidad de encontrar un ítem, ya sea tesoro o un consumible, dependiendo de su suerte y de los parámetros [PROB_ENCONTRAR_TESORO](#) y [PROB_ENCONTRAR_CONSUMIBLE](#), respectivamente. Se debe mostrar el **nombre de cada excavador**, si es que **consiguió o no** algún ítem y, si es que consigue un ítem, el **nombre** y el **tipo** del ítem encontrado. Finalmente, se debe mostrar la cantidad de ítems totales encontrados en el día, especificando la cantidad de consumibles y tesoros.
3. **Ocurrencia evento:** Durante el día existirá una probabilidad [PROB_INICIAR_EVENTO](#) de que ocurra un evento, en donde la arena y los excavadores pueden sufrir cambios. Deberás mostrar el **tipo de evento ocurrido**, el **tipo de arena** resultante y el **efecto que tenga tu equipo** frente a este evento. Los cambios que cada evento genera en el Torneo están explicados en [Torneo](#).

Además de toda esta información, se deberá imprimir en consola el **número de día simulado** y se deberá mostrar el nombre de los excavadores que estén **descansando**. Puedes incluir más información si la consideras relevante.

Día 1

```
-----  
Metros Cavados:  
Emolaba ha cavado 4 metros.  
Beyoncé ha cavado 5 metros.  
Lily614 ha cavado 3 metros.  
Mpia_vf ha cavado 4 metros.  
jtvaldivia ha cavado 5 metros.  
El equipo ha conseguido excavar 21 metros.  
  
Items Encontrados:  
Emolaba consiguió Helado del tipo consumible.  
Beyoncé consiguió Completo del tipo consumible.  
Lily614 no consiguió nada.  
Mpia_vf consiguió PepaBot del tipo tesoro.  
jtvaldivia no consiguió nada.  
Se han encontrado 3 ítems:  
- 2 consumibles.  
- 1 tesoros.  
  
¡¡Durante el día de trabajo ocurrió un Terremoto!!  
La arena final es del tipo rocosa  
Tu equipo ha perdido 2 de felicidad  
  
Lily416 decidió descansar...
```

Figura 4: Ejemplo de Simular día

Después de mostrar lo anterior, y si es que quedan más días de competencia, se deberá volver al [Menú Principal](#). En caso de terminar los días se deberá mostrar el resultado de la competencia junto a los metros excavados totales y [METROS_META](#), indicando si ganaste o no. Finalmente, se deberá volver al [Menú de Inicio](#).

3.2.2. Mostrar Estado Torneo

Al seleccionar esta opción se desplegará la información más relevante del torneo hasta el momento. Como mínimo deberá estar incluido: **el día actual del torneo, el tipo de arena actual, los metros excavados totales, la meta de metros por excavar y el estado de los excavadores**. La información de los excavadores a mostrar es el **nombre, tipo, energía, fuerza, suerte y felicidad**. Si consideras que se debe incluir más información relevante, puedes hacerlo.

*** Estado Torneo ***						

Día actual: 3						
Tipo de arena: Mojada						
Metros excavados: 350 / 900						

Excavadores						

Nombre	Tipo	Energía	Fuerza	Suerte	Felicidad	

Lily614	Docencio	24	20	10	30	
Mpia_vf	Docencio	55	35	15	25	
Beyoncé	Hibrido	81	35	20	40	

Figura 5: Ejemplo de Estado Torneo

Después de mostrar lo anterior se volverá a desplegar el [Menú Principal](#).

3.2.3. Menú ítems

En este menú se va a desplegar información de los ítems que tiene el equipo. Como mínimo debe estar la siguiente información: **Nombre**, **Tipo** y **Descripción**. Al seleccionar ítem, se deben aplicar los efectos respectivos. Si es del tipo **Consumible**, se debe aplicar el efecto del ítem a todos los excavadores. Si es del tipo **Tesoro**, se aplicará el efecto ya sea al equipo o bien la arena según corresponda.

*** Menú Ítems ***		

Nombre	Tipo	Descripción

[1] Empanada	Consumible	Otorga +30 energía / +5 fuerza / +2 felicidad.
[2] DCCaboom!!	Tesoro	Cambia el estado de la arena a rocosa.
[3] Desloratadina	Consumible	Otorga +8 energía / -2 fuerza / +6 felicidad.

[4] Volver		
[X] Salir del programa		
Indique su opción (1, 2, 3, 4 o X):		

Figura 6: Ejemplo de Menú Ítems

En caso de que se seleccione algún ítem, deberá imprimirse en consola los efectos que causó en el **equipo** y la **arena**, para finalmente volver a mostrar el [Menú ítems](#).

3.2.4. Guardar partida

Al seleccionar esta opción se deberán guardar todos los avances que existan del *DCCavaCava* en el archivo [DCCavaCava.txt](#). El formato del archivo queda a tu criterio, siempre y cuando seas capaz de cargarlo al momento de seleccionar la opción **Cargar partida** del [Menú de Inicio](#).

En caso de que el archivo no exista, este debe ser creado, mientras que si el archivo **ya existía** previamente, entonces se **sobrescribirá**. Luego de guardar la partida, se deberá mostrar un mensaje indicando que la partida ha sido guardada exitosamente y se volverá a mostrar el [Menú Principal](#).

4. Entidades

En esta sección se detallarán las entidades que necesitarás para simular *DCCavaCava*. Deberás utilizar conceptos clave de **Programación Orientada a Objetos**, tales como herencia, clases abstractas, polimorfismo, *properties* y relaciones, que pueden ser de agregación o composición. Ten en cuenta que cada uno de estos elementos debe ser incluido en el programa al menos una vez, y deberás descubrir dónde implementarlos según lo propuesto por el enunciado.

Las entidades principales de *DCCavaCava* son **Torneo**, **Arena**, **Excavadores** e **Ítems**. Debes incluir como mínimo las características nombradas a continuación, pero siéntete libre de añadir nuevos atributos y métodos si lo estimas necesario.

4.1. Torneo

Es la entidad general donde se jugará la partida de *DCCavaCava*. En esta tendrás a la arena actual, tu equipo de excavadores y la mochila con ítems. A continuación, se enunciarán las características de la entidad **Torneo**:

- **Arena:** Es el tipo de arena que está actualmente en el terreno de la competencia.
- **Eventos:** Corresponde al conjunto de los eventos que pueden ocurrir durante el torneo.
- **Equipo:** Conjunto de excavadores que tienes a disposición para cavar.
- **Mochila:** Conjunto de ítems que hayas encontrado durante la excavación.
- **Metros cavados:** Corresponde a un `float` que indica la cantidad de metros cavados hasta el momento.
- **Meta:** Corresponde a un `float` que indica la cantidad de `METROS_META` a superar en el torneo.
- **Días transcurridos:** Corresponde a un `int` que tiene como objetivo mostrar el día actual de la competencia.
- **Días totales:** Corresponde a un `int` que indica la cantidad de `DIAS_TOTALES_TORNEO` que durará el torneo.
- **Simular día:** Se encarga de simular el día y mostrar en consola el resumen de lo que ocurrió. Esto se encuentra especificado en [Simular día Torneo](#).
- **Mostrar estado:** Muestra en consola el estado actual de la competencia. Ver detalle en [Mostrar Estado Torneo](#).
- **Ver mochila:** Muestra todos los ítems que tienes a disposición en tu mochila. Ver detalle en [Menú ítems](#).
- **Usar consumible:** Para cada uno de los excavadores dentro del equipo, se aplicará los efectos del consumible seleccionado.
- **Abrir tesoro:** Al seleccionar un tesoro de la mochila, se activará su efecto determinado, alterando el transcurso del torneo.
- **Iniciar evento:** Para cada día habrá una `PROB_INICIAR_EVENTO` donde los excavadores y la arena estarán expuestos a sufrir cambios que podrían cambiar el transcurso del torneo. Existen 3 tipos de eventos:

- **Lluvia:** Este evento tiene una probabilidad de ocurrencia `PROB_LLUVIA` y transforma arena normal en arena mojada, o bien arena rocosa en arena magnética. No tiene efecto sobre la arena mojada ni magnética.
- **Terremoto:** Este evento tiene una probabilidad de ocurrencia `PROB_TERREMOTO` y transforma la arena normal en arena rocosa, o bien arena mojada en arena magnética. No tiene efecto sobre la arena rocosa ni magnética.
- **Derrumbe:** Este evento tiene una probabilidad de ocurrencia `PROB_DERRUMBE` y transforma cualquier tipo de arena en arena normal. Además, el equipo pierde `METROS_PERDIDOS_DERRUMBE` metros de progreso.

Para cambiar de una arena a otra, se debe seleccionar de manera **aleatoria** cualquiera de las arenas del archivo `arenas.csv` filtrando las que son del tipo de arena correspondiente.

Es necesario tener en cuenta que cada evento adverso genera infelicidad a tus excavadores. Por lo tanto, cada excavador pierde `FELICIDAD_PERDIDA` unidades de felicidad luego de que ocurra cualquier evento, **independiente a si cambia el estado actual de la arena** o no.

Además, es importante tener en cuenta que las probabilidades `PROB_LLUVIA`, `PROB_TERREMOTO` y `PROB_DERRUMBE` sumarán 1, ya que primero se debe determinar si ocurrirá un evento, o no, según `PROB_INICIAR_EVENTO` y luego, si un evento ocurre, se debe determinar cuál de los tres será, de acuerdo a sus probabilidades.

4.2. Arena

La arena es el escenario donde se desarrollará la competencia. Cada día los excavadores cavarán una cierta cantidad de metros, la cual se verá afectada dependiendo del **tipo** de arena presente. Existen **4 tipos** distintos de arena: **arena normal**, **arena mojada**, **arena rocosa** y **arena magnética**. Cada tipo de arena tendrá distintas dificultades y contará con eventos únicos que ocurrirán en ella, lo que afectará el estado de la competición.

A continuación se presentan las características y propiedades de la arena:

- **Nombre:** Corresponde a un `str` con el nombre de la arena.
- **Tipo:** Corresponde a un `str` que indica qué tipo de arena es.
- **Rareza:** Corresponde a un `int` entre 1 (menos raro) y 10 (más raro).
- **Humedad:** Corresponde a un `int` entre 1 (menos humedad) y 10 (más humedad).
- **Dureza:** Corresponde a un `int` entre 1 (menos dureza) y 10 (más dureza).
- **Estática:** Corresponde a un `int` entre 1 (con menos estática) y 10 (con más estática).
- **Ítems:** Corresponde al conjunto de todos los *ítems* que pueden aparecer en la arena.
- **Dificultad de la arena:** Dará un valor `float` entre 0 (menos difícil) y 1 (más difícil) que afectará la velocidad de excavación. Este valor estará dado por la siguiente fórmula y siempre debe estar redondeado a los 2 decimales:

$$dificultad\ arena = \text{round}\left(\frac{rareza + humedad + dureza + estática}{40}, 2\right)$$

4.2.1. Tipos de arenas

- **Arena normal**

La arena normal corresponde al tipo de arena más fácil de cavar. Debido a esto, la dificultad de la arena al momento de cavar se ve multiplicada por `POND_ARENA_NORMAL`, el cual será un valor entre `0.1` y `1`, inclusive. Una vez aplicado el ponderador, la dificultad vuelve a redondearse a los 2 decimales.

- **Arena mojada**

Cuando llueve, la arena normal se transforma en arena mojada. Esto genera que algunos ítems se eleven y sea más fácil poder encontrarlos. Debido a esto, **siempre** un excavador va a poder encontrar un ítem. Además, el ítem encontrado tendrá la misma probabilidad ser un **Tesoro** o bien un **Consumible**.

- **Arena rocosa**

Cuando ocurre un terremoto, la arena normal se transforma en arena rocosa. Esta arena es un poco más compleja de cavar que la arena normal, ya que la variante de la dureza afecta más al momento de cavar. Por esta razón, la dificultad de la arena se ve alterada de la siguiente forma:

$$dificultad\ arena = \text{round}\left(\frac{rareza + humedad + 2 \times dureza + estática}{50}, 2\right)$$

- **Arena magnética**

Hay dos formas de obtener arena magnética. Cuando ocurre un terremoto sobre arena mojada y cuando llueve sobre arena rocosa. Este tipo de arena tiene el mismo comportamiento que el de **Arena mojada** y **Arena rocosa**. Sin embargo, dada sus propiedades especiales, cada vez que se inicie el proceso de `Simular día Torneo`, la *humedad* y *dureza* serán valores **aleatorios** entre 1 y 10.

Por último, deberás mantener las propiedades de la arena **siempre** entre 1 y 10 incluso cuando sucedan los eventos de cambio del tipo de arena.

4.3. Excavador

Los excavadores son los que contribuyen al desarrollo de la excavación, al avance del torneo y pueden encontrar tesoros según su suerte. Para poder excavar dispondrán de su energía, la cual se irá gastando mediante por cada día de excavación, para recuperarla deberán descansar o consumir algún ítem.

El excavador tendrá como mínimo las siguientes características:

- **Nombre:** Corresponde a un `str` que indica el nombre del excavador.
- **Edad:** Corresponde a un `int` que indica la edad del excavador. Este valor siempre irá entre `18` y `60` años. La edad afecta la velocidad de excavación.
- **Energía:** Corresponde a un `int` entre `0` y `100`. Esta se puede ver afectada tanto por los días transcurridos, como por los ítems y habilidades utilizadas.
- **Fuerza:** Corresponde a un `int` entre `1` y `10`, se puede ver afectada por los ítems utilizados. La fuerza afecta la velocidad de excavación.
- **Suerte:** Corresponde a un `int` entre `1` y `10`, se puede ver afectada por los ítems utilizados. Este valor afecta la probabilidad de encontrar objetos.

- **Felicidad:** Corresponde a un `int` entre 1 y 10, se puede ver afectada tanto por eventos adversos como por los ítems utilizados. Este valor afecta la velocidad de excavación.

Además, los excavadores son capaces de realizar las siguientes acciones:

- **Cavar:** El excavador cavará una cantidad específica de metros por día que está dado por la siguiente fórmula:

$$\text{metros cavados} = \text{round} \left(\left(\frac{30}{\text{edad}} + \frac{\text{felicidad} + 2 \times \text{fuerza}}{10} \right) \times \frac{1}{10 \times \text{dificultad arena}}, 2 \right)$$

Donde *dificultad arena* corresponde al resultado de la fórmula de **dificultad arena** en la entidad *Arena*.

- **Descansar:** Cuando la energía del excavador llegue a 0, este necesitará descansar una cierta cantidad de días para que la energía del excavador vuelva a ser de 100. La cantidad de días dependerá de la edad como se indica a continuación:

$$\text{días descanso} = \text{int} \left(\frac{\text{edad}}{20} \right)$$

- **Encontrar Ítem:** Cada día los excavadores tendrán la probabilidad de encontrar un *Ítem*. Esta probabilidad estará dada según la siguiente fórmula:

$$\text{probabilidad ítem} = \text{PROB_ENCONTRAR_ITEM} \times \left(\frac{\text{suerte}}{10} \right)$$

Posterior a verificar si el excavador pudo obtener un ítem, si esta probabilidad acierta, entonces el excavador podrá obtener un tesoro con probabilidad de `PROB_ENCONTRAR_TESORO`, o bien un consumible con probabilidad `PROB_ENCONTRAR_CONSUMIBLE`. Ambas constantes deben sumar 1. El tesoro o consumible específico que se adquiere será un ítem **aleatorio** con igual probabilidad entre los que están en la arena.

- **Gastar Energía:** La energía del excavador descende una cantidad determinada por día excavado, la cual está determinada por la siguiente fórmula:

$$\text{energía gastada} = \text{int} \left(\frac{10}{\text{fuerza}} + \frac{\text{edad}}{6} \right)$$

- **Consumir:** Se utiliza un consumible y se obtienen sus beneficios. Dependiendo de la clase, puede ser que algunos de estos beneficios se vean afectados.

4.3.1. Tipos de excavadores

Además, existen tres tipos distintos de excavadores: **ExcavadorDocencio**, **ExcavadorTareo** y **ExcavadorHíbrido**; donde cada uno ellos posee una habilidad exclusiva. Estos se enlistan a continuación:

- **ExcavadorDocencio**

Estos excavadores son expertos en impartir clases sobre excavaciones, debido a esto, al momento de excavar, se activa una acción especial que modifica las características del excavador. En particular, luego de que el excavador termina de cavar, aumenta su felicidad en `FELICIDAD_ADICIONAL_DOCENCIO` unidades y su fuerza en `FUERZA_ADICIONAL_DOCENCIO` unidades. Sin embargo, dado que impartir clases requiere de mucho esfuerzo, además de la cantidad de energía gastada según la fórmula de **Gastar Energía**, su energía disminuye en `ENERGIA_PERDIDA_DOCENCIO` unidades.

■ **ExcavadorTareo**

Estos excavadores son aptos para proponer tareas, debido a esto, tienen grandes conocimientos sobre investigación y planificación de métodos de extracción de arena, lo cual les permite desarrollar habilidades para aprovechar de mejor forma los ítems que recolectan. En específico, al momento de consumir cualquier ítem del tipo **consumible**, tendrán una bonificación de `ENERGIA_ADICIONAL_TAREO` unidades y `SUERTE_ADICIONAL_TAREO` unidades. Sin embargo, dado que ~~corregir tareas~~ la investigación toma mucho tiempo, su edad se ve aumentada en `EDAD_ADICIONAL_TAREO` años y su felicidad disminuye en `FELICIDAD_PERDIDA_TAREO` unidades.

■ **ExcavadorHíbrido**

Estos excavadores son aptos tanto para ofrecer conocimientos como de elaborar tareas, por lo que poseen las mismas habilidades que los ExcavadoresDocencios y los ExcavadoresTareos. Adicionalmente, dado que tienen un gran equilibrio entre ambas entidades, siempre tienen una reserva de energía al momento de cavar. Por esta razón, la energía perdida al momento de excavar se ve reducida a la mitad, y su energía total restante no puede descender de los `20`, por lo que nunca necesitan descansar.

4.4. Ítem

Los ítems juegan un rol fundamental durante el transcurso del torneo. Estos por lo general entregan beneficios para todo tu equipo. ¡Úsalos con cuidado!

En general, todos los ítems tienen las siguientes características:

- **Nombre:** Corresponde a un `str` con el nombre del ítem.
- **Tipo:** Corresponde a un `str` con el tipo de ítem.
- **Descripción:** Corresponde a un `str` que describe brevemente el ítem.

A continuación se enlistan los tipos de ítems existentes en *DCCavaCava*.

4.4.1. Consumibles

Los consumibles pueden alterar tanto positiva como negativamente los atributos del equipo. Contarán con las siguientes características de modificación:

- **Energía:** Cambia en `int` unidades la energía de un excavador.
- **Fuerza:** Cambia en `int` unidades la fuerza de un excavador.
- **Suerte:** Cambia en `int` unidades la suerte de un excavador.
- **Felicidad:** Cambia en `int` unidades la felicidad de un excavador.

Notar que estas características pueden tener tanto valores positivos como negativos.

4.4.2. Tesoros

Son ítems bastante raros de encontrar. Pueden alterar el tipo de arena en la que se cava o también podrían agrandar el tamaño del equipo.

- **Calidad:** Si son de primera calidad (1), entonces corresponde a un tesoro que permite agrandar el equipo. En cambio, si son de segunda calidad (2), corresponderán a tesoros que son capaz de cambiar el tipo de arena. Ver detalles en la sección [Tipos de arenas](#).

- **Cambio:** Indica qué cambio realizará el tesoro. Si es de calidad 1, tendrá un tipo de excavador, mientras que si es de calidad 2, tendrá un tipo de arena. Por ejemplo, si **cambio** es **magnética** cambiará al tipo de arena magnética. Si **cambio** es **híbrido** se agregará un ExcavadorHíbrido al equipo.

5. Archivos

Los siguientes archivos contienen la base de datos de los distintos excavadores, arenas e ítems disponibles para competir en *DCCavaCava*. Cada archivo contiene un encabezado (*header*) en la primera línea que indica el nombre de cada columna, separadas por comas (","). Todos los archivos son necesarios para la correcta ejecución del programa y puedes asumir que el orden de cada columna no variará.

Ten en cuenta que al momento de leer y escribir en los archivos existe la posibilidad de que ciertos caracteres especiales, como la ñ o las tildes no se escriban correctamente. Para solucionar esto, existe un argumento en la función `open()` llamado *encoding* donde pueden especificar el formato de codificación de los caracteres⁵.

Además de los archivos entregados, deberás crear dos archivos:

1. Un archivo `parametros.py`, que contenga como mínimo cada parámetro importante presentado a lo largo del enunciado.
2. Un archivo `DCCavaCava.txt` cuando desees guardar la partida. El formato de este archivo queda totalmente a tu criterio mientras pueda ser leído correctamente al momento de cargar una partida.

A continuación, se detallan los archivos mencionados:

5.1. excavadores.csv

Este archivo contiene todas las opciones de excavadores a reclutar para tu equipo, junto a todas las características descritas en la siguiente tabla:

Nombre	Tipo de Dato	Descripción
Nombre	<code>str</code>	Indica el nombre del excavador
Tipo	<code>str</code>	Indica el tipo de excavador, el cual puede ser de tipo <code>'docencio'</code> , <code>'tareo'</code> o <code>'hibrido'</code> .
Edad	<code>int</code>	Indica la edad del excavador.
Energía	<code>int</code>	Indica la energía del excavador.
Fuerza	<code>int</code>	Indica la fuerza del excavador.
Suerte	<code>int</code>	Indica la suerte del excavador.
Felicidad	<code>int</code>	Indica la felicidad actual del excavador.

Un ejemplo del archivo `excavadores.csv` es el siguiente:

```

1 nombre,tipo,edad,energia,fuerza,suerte,felicidad
2 MatiasMasjuan,docencio,23,80,7,7,9
3 Benjeh,tareo,22,0,1,1,1
4 Beyoncé,hibrido,41,100,10,10,10

```

⁵Para ver ejemplos de uso, pueden visitar la [documentación](#).

5.2. arenas.csv

Este archivo contiene toda la información pertinente a los distintos tipos de arenas que se pueden encontrar en la competencia. Cada arena tiene un nombre, un tipo y una dureza definida de acuerdo a su tipo.

Nombre	Tipo de Dato	Descripción
Nombre	<code>str</code>	Indica el nombre de la arena.
Tipo	<code>str</code>	Indica el tipo de arena. Puede ser <code>'normal'</code> , <code>'mojada'</code> , <code>'rocosa'</code> o <code>'magnetica'</code> .
Rareza	<code>int</code>	Indica la rareza de la arena.
Humedad	<code>int</code>	Indica la humedad de la arena.
Dureza	<code>int</code>	Indica la dureza de la arena.
Estática	<code>int</code>	Indica la estática que contiene la arena.

Un ejemplo del archivo `arenas.csv` es el siguiente:

```
1 nombre, tipo, rareza, humedad, dureza, estatica
2 Arena roja, seca, 2, 3, 1, 2
3 Arena fangosa, mojada, 3, 7, 2, 1
4 Arena volcánica, rocosa, 6, 3, 10, 5
5 Arena eléctrica, magnetica, 8, 9, 8, 10
```

5.3. consumibles.csv

Este archivo contiene todos los consumibles que se pueden encontrar mientras uno se encuentra en el *DCCavaCava*.

Nombre	Tipo de Dato	Descripción
Nombre	<code>str</code>	Indica el nombre del ítem.
Descripción	<code>str</code>	Indica el efecto que posee.
Energía	<code>int</code>	Indica la cantidad de energía que otorga o quita a los excavadores.
Fuerza	<code>int</code>	Indica la cantidad de fuerza que otorga o quita a los excavadores.
Suerte	<code>int</code>	Indica la cantidad de suerte que otorga o quita a los excavadores.
Felicidad	<code>int</code>	Indica la cantidad de felicidad que otorga o quita a los excavadores.

Un ejemplo del archivo `consumibles.csv` es el siguiente:

```
1 nombre, descripcion, energia, fuerza, suerte, felicidad
2 completo, Otorga +15 energia/-4 fuerza/+3 suerte/+10 felicidad, 15, -4, 3, 10
3 mojito, Otorga +3 energia/-5 fuerza/+6 suerte/+8 felicidad, 3, -5, 6, 8
4 hamburguesa de lentejas, Otorga +20 energia/+10 fuerza/+3 suerte/+6 felicidad, 20, 20, 3, 6
```

5.4. tesoros.csv

Este archivo contiene todos los tesoros disponibles en *DCCavaCava*, los cuales tienen efectos únicos dentro del programa.

Nombre	Tipo de Dato	Descripción
Nombre	str	Indica el nombre del ítem.
Descripción	str	Corresponde a la historia del tesoro, junto al efecto que entrega.
Calidad	int	Indica la calidad del tesoro. Puede ser 1 o 2.
Cambio	str	Indica el cambio que realizará según la calidad. Puede ser 'normal', 'mojada', 'rocosa', 'magnetica', 'docencio', 'tareo' o 'hibrido'.

Un ejemplo del archivo `tesoros.csv` es el siguiente:

```
1 nombre,descripción,calidad,cambio
2 Caliz Divino,Cambia el estado de la arena a mojada,2,mojada
3 Apuntes de Gran Docencio,Añade a tu equipo un excavador de tipo Docencio,1,docencio
4 PepaBot,Cambia el estado de la arena a magnetica,2,magnetica
```

5.5. DCCavaCava.txt

Además de los archivos `.csv` sobre las entidades, al momento de guardar la partida en las opciones del [Menú Principal](#), deberás escribir en este archivo toda la información sobre la arena, el equipo, métricas e ítems que tenga en la mochila.

El formato del archivo queda a tu criterio, siempre y cuando seas capaz de guardar toda la información que necesites y cargarlo al momento de seleccionar la opción **Cargar partida** del [Menú de Inicio](#). En el caso de que el archivo ya exista, se debe sobrescribir la información presente.

5.6. parametros.py

Para esta tarea, deberás crear un archivo `parametros.py` y **completarlo con todos los parámetros mencionados a lo largo del enunciado**, los cuales encontrarás en [ESTE_FORMATO](#) y en ese color. Además, debes agregar cualquier valor constante en tu tarea, junto con los *paths* que utilices.

Recuerda que los parámetros deben ser descriptivos y reconocibles:

```
1 CIEN = 100 # mal parámetro
2 DEUDA_TOTAL = 3000 # buen parámetro
3 PROBABILIDAD_EVENTO = 0.2 # buen parámetro
```

Si necesitas agregar algún parámetro que varíe de acuerdo a otros parámetros, una correcta parametrización sería la siguiente:

```
1 PI = 3.14
2 RADIO_ESFERA = 3
3 VOLUMEN_ESFERA = (4/3) * PI * (RADIO_ESFERA ** 3)
```

Dentro del archivo `parametros.py` deberás hacer uso de todos los parámetros almacenados y deberás importarlos correctamente. Si se almacena cualquier información no correspondiente a parámetros, esto tendrá un descuento en tu nota. Por último, no está permitido que un parámetro se encuentre *hardcodeado*⁶, ya que es una mala práctica y su uso conlleva un descuento.

Para esta tarea, el archivo `parametros.py` no se debe ignorar y debes subirlo a tu repositorio. En caso contrario, tu tarea no se podrá corregir.

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**⁷.
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 3 décimas. Deberás indicar en tu README si implementaste alguno de los bonus, y cuáles fueron implementados.

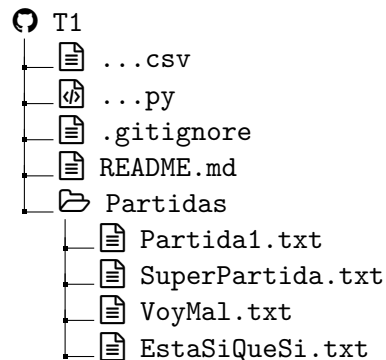
6.1. Guardar Partida (3 décimas)

Para optar a este *bonus* se debe implementar la opción de guardar múltiples partidas en distintos archivos de texto (`.txt`). La opción de **Guardar Partida** debe aparecer dentro del **Menú Principal** y al presionarlo deberá preguntar al usuario el nombre con el cual se quiere guardar la partida **sin considerar la extensión del archivo (`.txt`)**. Una vez entregado el nombre del archivo, se deberá guardar el contenido dentro de la carpeta **Partidas** y se volverá al **Menú Principal**.

Análogamente, el **Menú de Inicio** se debe modificar la opción de **Cargar Partida** mostrando todos los archivos de texto en el directorio correspondiente, **Tareas/T1/Partidas/**. Al seleccionar uno de los archivos, se debe cargar la partida conforme a esa información y mostrar el **Menú Principal**.

```
*** Menú de carga ***
-----
[1] Partida1
[2] SuperPartida
[3] VoyMal
[4] EstaSiQueSi
[X] Salir
```

(a) Ejemplo de Menú de Carga



(b) Vista del repositorio en la carpeta T1/

El formato que se debe utilizar para guardar la partida dentro de los archivos de texto queda a tu criterio, sin embargo, te debes asegurar de que toda la información del equipo de excavación, sus objetos y su progreso se almacene correctamente de modo tal que se pueda restaurar la partida guardada.

⁶*Hard-coding* es la mala práctica de incrustar datos directamente en el código fuente del programa, en vez de obtener los datos de una fuente externa.

⁷Esta nota es sin considerar posibles descuentos.

7. Diagrama de clases

En conjunto con el programa, deberás realizar un diagrama de clases modelando las entidades necesarias del *DCCavaCava*. Este diagrama se entregará en dos ocasiones: la primera en el [Avance de tarea](#), y la segunda en la entrega final. Para esta última, deberás entregar la **versión final** de tu diagrama que **represente fielmente** la modelación de clases implementada en tu programa.

En ambos casos, el diagrama deberá:

- Entregarse en **formato PDF o de imagen**⁸.
- Contener todas las clases junto con sus atributos y métodos. También deberás identificar cuáles clases serán abstractas y cuáles no.
- 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](#) o aplicaciones similares.

Es conveniente adjuntar a tu diagrama un documento con una explicación general de tu modelación. Esto es con el fin de ayudar en la corrección del ayudante a comprender tu 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. Avance de tarea

Para el [Avance de tarea](#) de tu tarea deberás realizar una versión preliminar del [Diagrama de clases](#) que refleje cómo planeas modelar tu programa. A partir de los avances entregados, se brindará un *feedback* general de lo entregado y, además, te permitirá optar por **hasta 2 décimas** adicionales en la nota final de tu tarea.

9. .gitignore

Para esta tarea **deberás utilizar un archivo .gitignore** para ignorar los archivos indicados, este deberá estar dentro de tu carpeta `Tareas/T1/`. Puedes encontrar un ejemplo de `.gitignore` en el siguiente [link](#).

Para esta tarea, los archivos a ignorar corresponden a las bases de datos entregadas para la simulación junto con el enunciado. Es decir, deberás ignorar:

- `Enunciado.pdf`
- `excavadores.csv`
- `arenas.csv`
- `consumibles.csv`
- `tesoros.csv`

Se espera que no se suban archivos autogenerados por las interfaces de desarrollo o los entornos virtuales de Python, como por ejemplo: la carpeta `__pycache__`, `DS_Store`, entre otros.

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.

⁸Cualquier otro formato no será considerado como una entrega válida y no tendrá décimas de avance o puntaje en la tarea.

10. Entregas atrasadas

Entregas con atraso tendrán una penalización en la calificación máxima obtenida acorde a la política de atrasos. Sin embargo, tendrás a disposición cupones para eliminar días de atrasos.

Política de atraso:

- Hasta 1 día de atraso (00:01 a 24:00 hrs de atraso) tendrán calificación máxima 6,0.
- Hasta 2 días de atraso (24:01 a 48:00 hrs de atraso) tendrán calificación máxima 4,0.
- Después de las 48 hrs, no se aceptarán entregas atrasadas y se aplicará calificación mínima 1,0.

Cupones:

- Durante el semestre cada estudiante dispondrá de 2 cupones.
- Cada cupón permite eliminar 1 día de atraso en la tarea recién entregada.

Posterior a la fecha de entrega de la tarea se abrirá un formulario de **Google Form**, en el cual deberán indicar si desean optar por una entrega atrasada y si harán uso de 0, 1 o 2 cupones para la entrega.

11. Importante: Corrección de la tarea

Para esta tarea, el carácter funcional del programa 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 *push* en sus repositorios.

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 archivo `README.md` deberás señalar el archivo y la línea donde se encuentran definidas las funciones o clases relacionados a esos ítems.

Finalmente, si durante la realización de tu tarea se te presenta algún problema o situación que pueda afectar tu rendimiento, no dudes en contactar al ayudante Coordinador de Bienestar o al ayudante de Bienestar de tu sección, puedes hacerlo escribiéndoles a sus respectivos [correos](#).

12. 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 3.10.
- 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. El no incluir este archivo o bien se encuentra vacío conllevaría un [descuento](#) en tu nota.
- Entregas con atraso entre 0 a 24 horas tendrán como calificación máxima 6,0. Entregas entre 24 y 48 horas tendrán calificación máxima 4,0. Más allá de este plazo, la calificación será mínima (1,0).

- Tendrás a tu disposición solo **dos** cupones en el semestre. Cada cupón permite eliminar 1 día de atraso en la tarea recién entregada.
- En ambas situaciones, ya sea entregar atrasado y/o utilizar 0, 1 o 2, deberán rellenar el **formulario** correspondiente que será liberado una vez terminado el plazo oficial de la tarea.
- 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).