Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2233 Programación Avanzada (2020-2)

Tarea 01

Entrega

- **Avance de tarea**
  - **Fecha y hora:** martes 8 de septiembre de 2020, 20:00
  - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T01/`
- **Tarea**
  - **Fecha y hora:** miércoles 16 de septiembre de 2020, 20:00
  - **Lugar:** Repositorio personal de GitHub — Carpeta: `Tareas/T01/`
- `README.md`
  - **Fecha y hora:** jueves 17 de septiembre 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. *DCCumbre Olímpica*

Debido a la pandemia mundial, el Comité Olímpico Internacional (COI) decidió suspender la edición de los Juegos Olímpicos de Tokyo 2020. Para evitar la decepción de tener que esperar un año más, el COI ha llamado al DCC para que programe una competencia que alegre los corazones de la gente en sus casas: La *DCCumbre Olímpica*.

Gracias a tus nuevos conocimientos de **POO**, el DCC ha decidido que tú deberás hacer un programa que permita simular la *DCCumbre Olímpica*. Esta simulación también permite al usuario participar cumpliendo la labor de entrenador de una delegación, el cual debe velar por la victoria de su delegación.

Figura 1: Logo de la *DCCumbre Olímpica*

2. Flujo del programa

Tu misión es crear un programa que permita a un usuario entrenar y dirigir a una delegación para la *DCCumbre Olímpica*, que es un campeonato entre 2 delegaciones con distintas características. Cada delegación se compone inicialmente de 5 deportistas, pudiendo fichar más a medida que avanza el campeonato. El campeonato se simula durante varios días y el usuario, que cumple el rol de entrenador, puede tomar decisiones que impacten el rendimiento de los deportistas de su delegación. La ejecución e interacción del juego serán exclusivamente mediante menús en consola,

El objetivo del usuario es ganar la mayor cantidad de medallas. Existen 4 deportes individuales, cada uno con su propia manera de obtener puntaje: Atletismo, Ciclismo, Gimnasia y Natación. El equipo ganador recibirá una medalla y dinero por cada competencia ganada. El dinero le permitirá realizar acciones. Cada deportista tiene *stats* (atributos) que pueden ser influenciados por entrenamiento o tecnología.

Al iniciar el programa se abrirá el Menú de inicio, donde el usuario podrá empezar una partida o salir. En caso de empezar una partida, podrá ingresar nombre y seleccionar una de las 2 delegaciones a representar, convirtiéndose la delegación no elegida en su oponente durante el campeonato. Luego, el programa debe mostrar el Menú principal, desde el cual se puede simular el día de competencia correspondiente, revisar el estado del campeonato, acceder al Menú Entrenador y salir.

La *DCCumbre Olímpica* se compone de duplas de días: un día de entrenamiento y otro para simular las competencias. En los días de entrenamiento el usuario debe poder, desde el Menú Entrenador, imprimir el estado actual del torneo, fichar deportistas y entrenarlos para el campeonato. Los días de competencia se componen de cuatro competencias seguidas, que son enfrentamientos individuales entre 2 deportistas.

Cada día de competencia, el usuario debe seleccionar a los deportistas que participarán en cada uno de los 4 deportes. Esto implica que cada día de competencia, se podrían ganar hasta 4 medallas, una por cada deporte. El resultado de una competencia dependerá de una fórmula específica para cada deporte y tomará en cuenta las características de los deportistas participantes.

La *DCCumbre Olímpica* tiene una duración de `DIAS_COMPETENCIA`[1] días, valor que siempre será un número par. Luego de que se acabe el último día de competencia, la *DCCumbre Olímpica* se da por finalizada y se deberán guardar los ganadores, medallas y resultados. El equipo ganador será el que tenga más medallas al final del campeonato. Una vez finalizada la *DCCumbre Olímpica*, el usuario podrá tener la opción de salir o realizar una nueva simulación.

3. Menús

El control de la *DCCumbre Olímpica* y la delegación se realiza a través de menús en consola. Los menús deben ser a prueba de **todo tipo de errores** de usuario, y adicionalmente, cada uno debe tener la opción de **volver atrás** y **salir del programa**. A continuación se explicarán los menús mínimos a incluir, pero siéntete libre de añadir más menús si lo consideras necesario.

3.1. Menú de Inicio

Al ejecutar *DCCumbre Olímpica*, se deberá desplegar un menú de bienvenida en donde permita al usuario seleccionar la opción de **Comenzar una nueva partida** o **Salir del programa**.

En caso de seleccionar **Comenzar una nueva partida**, se deberá pedir un nombre de usuario y un nombre de rival que sólo consideren caracteres alfanuméricos sin distinción entre mayúsculas y minúsculas. Luego, se le entregará la opción de escoger una delegación: **DCCrotona** o **IEEEsparta**. Después de haber elegido exitosamente una delegación, se deberá redirigir al **Menú principal**. El nombre de usuario elegido será también el nombre del entrenador de la delegación seleccionada, mientras que el nombre de rival será el del entrenador de la delegación rival. Por otro lado, en caso de seleccionar **Salir del programa**, se deberá finalizar la ejecución del programa.

3.2. Menú Principal

Corresponde al menú principal del programa. Las opciones disponibles pueden dirigir a otros menús o bien ser una acción de la simulación. Una vez resuelta la acción o menú, se debe regresar a este menú, salvo que la opción sea salir del programa.

Se deberá mostrar las siguientes opciones: **Menú entrenador**, **Simular competencias**, **Mostrar estado** y **Salir del programa**.

3.2.1. Menú Entrenador

Esta opción permite acceder al menú de Entrenador. En este menú deberán aparecer todas las acciones que el entrenador puede realizar relacionadas con los **deportistas** de su **delegación**, además de una opción para **Volver al menú anterior** y una para **Salir del programa**.

Las acciones disponibles son: **Fichar**, **Entrenar**, **Sanar**, **Comprar tecnología** y **Usar habilidad especial**.

En caso de seleccionar **Fichar**, se deben mostrar todos los deportistas disponibles, dar la opción de seleccionar alguno que no esté fichado por ninguna delegación y luego agregarlo al equipo de la delegación del usuario.

[1] Las palabras escritas en `ESTE_FORMATO` son parámetros que tendrás que escribir e importar desde el archivo `parametros.py`

Si la opción seleccionada es **Entrenar**, se deben mostrar los deportistas pertenecientes al equipo de la delegación del usuario y permitir seleccionar uno de ellos, para luego seleccionar cuál de sus atributos desea entrenar. Del mismo modo, si se elige **Sanar**, se deberán mostrar los deportistas de la delegación del usuario que estén actualmente lesionados y permitir seleccionar aquel que se desee sanar.

Por otro lado, la opción **Comprar tecnología** da la posibilidad de mejorar la efectividad de los implementos deportivos y médicos de la delegación a cambio de dinero.

Finalmente, la opción de **Usar habilidad especial** sólo podrá utilizarse una vez por partida y sus resultados varían según el tipo de delegación.

Puedes encontrar más información sobre estas acciones en la sección Delegaciones.

Luego de realizarse cualquiera de las acciones antes mencionadas, el programa debe volver a mostrar este menú.

3.2.2. Simular Competencias

Cuando el usuario estime conveniente, puede terminar los entrenamientos del día y realizar la simulación de las competencias del día siguiente.

En el caso que resten días de competencia, al entrar en este menú se mostrará una **lista de deportistas disponibles** de su propia delegación para competir en el primer deporte del día. Luego de escoger uno, la lista se volverá a mostrar y el usuario deberá nuevamente escoger un deportista para participar en la segunda competencia del día. Esto debe repetirse hasta que se hayan seleccionado los deportistas para las cuatro competencias del día.

En el caso de la delegación rival, esta seleccionará a sus deportistas para cada competencia de forma **aleatoria**. Queda a tu criterio decidir si la delegación rival usa o no sus días de entrenamiento fichando deportistas, sanándolos ó entrenándolos.

Luego se procederá a simular las competencias según lo especificado en la sección Campeonato.

Si es que ya no quedan más días para competir, se deben mostrar los resultados finales de la *DCCumbre Olímpica*, mostrar qué delegación fue la ganadora y dar la opción de **Salir del programa** o **Realizar una nueva simulación**.

3.2.3. Mostrar estado

Esta opción deberá imprimir en consola información sobre las delegaciones y sus deportistas, además de algunos datos que permitirán conocer el estado actual de la competencia. Para mayor detalle de qué se debe mostrar en este menú, revisar la sección Campeonato.

4. Entidades

En esta sección se detallan las entidades que necesitarás para simular la *DCCumbre Olímpica*. Para modelarlas correctamente, deberás utilizar conceptos clave de **programación orientada a objetos:** herencia, clases abstractas, polimorfismo, *properties* y relaciones, ya sean de agregación o composición. 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 son **Delegaciones**, **Deportistas**, **Deportes** y el **Campeonato**.

4.1. Delegaciones

Las **delegaciones** son quienes participan en la *DCCumbre Olímpica* y se encargan de dar apoyo en todo momento a los **deportistas**. Están dirigidas por un **entrenador** quien debe crear estrategias para guiar de la mejor manera posible a sus deportistas, y así prepararlos para ganar la mayor cantidad de competencias. Cada delegación puede influir en el estado físico de sus deportistas ya sea **sanando heridas** en caso que se lesionen y **entrenándolos** para mejorar su capacidad física. Las delegaciones poseen atributos, que incluyen:

- **Entrenador:** Es un `str` que corresponde al nombre del entrenador del equipo. El nombre debe estar en formato alfanumérico sin distinción entre carácteres mayúsculas y minúsculas.
- **Equipo:** Una delegación posee un equipo conformado por Deportistas, almacenados en una `list`. Cada delegación tendrá, como mínimo, cinco deportistas dentro de su equipo.
- **Medallas:** Es de tipo `int` y corresponde al total de medallas que posee el equipo. Las medallas son el premio que reciben las delegaciones cuando un deportista gana una competencia. Cada vez que se gana una medalla, esta aumenta en `0.01` la excelencia y respeto de la delegación.
- **Moral:** Es de tipo `int` y corresponde al nivel de ánimo y espíritu del equipo. Puede estar entre `0` y `100`. Se define como el **promedio** de la **moral** de cada deportista del equipo.
- **Dinero:** Es de tipo `int` y corresponde al dinero que posee la delegación. Siempre será un entero positivo. Las delegaciones pueden utilizar el dinero para las distintas acciones que se pueden realizar, y lo podrán obtener a medida que ganan competencias.

Cada delegación tiene fortalezas y debilidades, las cuales influyen directamente en su desempeño en las competencias. Dichas fortalezas y debilidades se expresan a través de los valores deportivos que posee cada tipo de delegación. Estos valores son de tipo `float` y varían entre `0.0` y `1.0`. Cuando se instancia una delegación, algunos atributos como los anteriores deberán ser cargados a partir del archivo `delegaciones.csv`, mientras que otros atributos como los siguientes se determinarán de forma aleatoria[2] dentro de los valores predeterminados para cada tipo de delegación.

- **Excelencia y respeto:** Este atributo influye en la efectividad que tiene el entrenador de sanar y entrenar a su equipo.
- **Implementos deportivos:** Este atributo corresponde al nivel tecnológico de los implementos deportivos para entrenar, y contribuye a un entrenamiento más eficiente.
- **Implementos médicos:** Este atributo influye en la capacidad de recuperación de un deportista luego de una lesión.

4.1.1. Acciones de las delegaciones

Dentro de las responsabilidades de las delegaciones están las siguientes acciones:

- **Fichar deportistas:** Cada delegación tiene la opción de fichar nuevos deportistas para sus equipos, siempre y cuando la moral de la delegación sea mayor a `20`. La delegación deberá pagar un monto por cada fichaje, el cual varía para cada deportista.
- **Entrenar deportistas:** Antes de enfrentar cada día de competencia, la delegación podrá enviar a entrenar a sus deportistas. El costo de enviar a entrenar a un deportista es de `30` DCCoins[3] por cada atributo entrenado y aumenta la moral del deportista en `1`.

[2] Puedes utilizar la función `uniform` del módulo `random`.
[3] Las DCCoins son la moneda oficial de la Cumbre.

- **Sanar lesiones:** Después de que un deportista se lesione, el entrenador puede enviarlo a recuperación, sin embargo le costará `30` DCCoins. Al momento de intentar sanar una lesión, hay una probabilidad de que el deportista se recupere o no. Esta probabilidad está dada por la siguiente fórmula:

$$\text{mín}\left(1, \text{máx}\left(0, \frac{moral\_deportista * (implementos\_medicos + excelencia\_respeto)}{200}\right)\right)$$

  Esta formula te entregará un `float` entre **[0, 1]**[4], el cual debes redondear al primer decimal. Este resultado corresponde a la probabilidad de sanar la lesión, por lo que luego deberás obtener un número aleatorio y ver si este cae bajo la probabilidad calculada.

- **Comprar tecnología:** En caso de tener dinero suficiente, la delegación puede comprar tecnología a un costo de `20` DCCoins cada una, la cuál aumentará la efectividad de sus implementos deportivos y médicos en un `10 %` cada uno.

- **Utilizar habilidad especial:** Cada delegación se especializa en un área especifica, y tiene una habilidad asociada a dicha área. El costo de utilizar la habilidad es de `20` DCCoins.

4.1.2. Tipos de delegaciones

Cada delegación puede pertenecer a uno de estos dos tipos, los cuales tienen rangos diferentes para sus atributos y habilidades especiales:

- **IEEEsparta:** La delegación de IEEEsparta es famosa por tener a los deportistas más fuertes de la *DCCumbre Olímpica*, por lo que al momento de entrenar tienen mayor efectividad que cualquier otra delegación. La fórmula de entrenamiento deportivo se pondera por `1.7`.

  La delegación de IEEEsparta posee como habilidad especial el poder realizar un grito de guerra, con el cual logran subir la moral de todos los deportistas de su equipo al máximo. Esta habilidad puede ser utilizada una sola vez.

  Como era de esperar, los IEEEspartanos se decepcionan mucho al momento de perder, por lo que la moral del deportista disminuye el doble de lo normal cuando pierde una competencia.

  Los valores de sus fortalezas y debilidades varían de la siguiente manera:

  - **Excelencia y respeto:** Entre `0.4` y `0.8`.
  - **Implementos deportivos:** Entre `0.3` y `0.7`.
  - **Implementos médicos:** Entre `0.2` y `0.6`.

- **DCCrotona:** La delegación de DCCrotona es una de las delegaciones más populares de la competencia, es por esto que cada vez que un deportista de este equipo gana una medalla la moral del deportista aumenta el **doble** que en condiciones normales.

  La habilidad especial de esta delegación es **poder obtener una medalla de forma inmediata**, obteniendo todos los beneficios que se otorgan cuando se gana en condiciones normales. Esta habilidad solo se puede utilizar una vez.

  Cada vez que un jugador se lesiona, la delegación debe pagar el doble para poder sanar a sus deportistas.

  Los valores de sus fortalezas y debilidades varían de la siguiente manera:

[4] Incluyendo los extremos.

- **Excelencia y respeto:** Entre `0.3` y `0.7`.
- **Implementos deportivos:** Entre `0.2` y `0.6`.
- **Implementos médicos:** Entre `0.4` y `0.8`.

4.2. Deportistas

Los deportistas son el alma de la competencia. De ellos depende la derrota o la gloria de su **Delegación**, por lo que entrenarán todo lo que puedan para llegar de la mejor forma a la gran *DCCumbre Olímpica* y llevarse el oro. Cabe notar que un deportista puede competir en cualquier deporte con distintos niveles de éxito dependiendo de sus atributos. Los deportistas tienen los siguientes atributos:

- **Velocidad**: Un `int` que va de `0` a `100` que determina qué tan rápido puede moverse. Es crucial para el atletismo, el ciclismo y la natación.
- **Resistencia**: Un `int` que va de `0` a `100` que determina cuánto tiempo puede aguantar el deportista haciendo una determinada actividad física. Esta habilidad es crucial para todos los deportes.
- **Flexibilidad**: Un `int` que va de `0` a `100` que determina cuánto puede estirarse el deportista. Es crucial en el ciclismo, la gimnasia y la natación.
- **Moral**: Un `int` de `0` a `100` que determina la actitud del deportista frente al campeonato. La moral influye en gran parte de las acciones en las que está involucrado.
- **Lesionado**: Un `bool` que indica si el deportista está lesionado o no.
- **Precio**: Un `int` que indica el costo de fichar al deportista.

Además, un deportista es capaz de realizar las siguientes acciones:

- **Entrenar**: El deportista puede ser entrenado en los días anteriores a los de competencia. Existen entrenamientos para cada atributo físico (velocidad, resistencia, y flexibilidad), por lo que se debe elegir cuál usar para el deportista. Cada entrenamiento aumenta en `PUNTOS_ENTRENAMIENTO` el atributo físico elegido.
- **Lesionarse**: Cada vez que un jugador compite en un deporte, existe una probabilidad de lesionarse. Esta probabilidad depende únicamente del valor de **riesgo** del deporte. Si un jugador se lesiona durante una competencia, la pierde automáticamente.

4.3. Deportes

Un deporte es la competencia para la cual entrenan los deportistas. Se caracteriza por tener un conjunto de reglas definidas y es la instancia en que los deportistas dan todo de si por conseguir el triunfo y llevar a su delegación a la cima. Un deporte posee los siguientes atributos:

- **Implemento**: Un `bool` que indica si es necesario el uso de implementos adicionales para competir.
- **Riesgo**: Un `float` entre `0.0` y `1.0` que indica la probabilidad de sufrir una lesión por estar practicando el deporte.

Además posee los siguientes métodos:

- **Validez de competencia**: Se encarga de verificar que se cumplan condiciones mínimas para competir: deben existir 2 deportistas (uno por delegación), no pueden estar lesionados y, de necesitar un implemento, la delegación del deportista debe tener un nivel mayor a `NIVEL_IMPLEMENTOS` de implementos deportivos.

En caso de que alguna Delegación no cumpla alguno de estos requisitos, pierde automáticamente la competencia. Si ninguna Delegación cumple los requisitos necesarios, la competencia se declara como un empate y no se otorgan medallas a ninguna delegación.

- **Calcular ganador**: Pondera las habilidades de los deportistas siguiendo una fórmula específica para cada deporte y luego compara los puntajes obtenidos por los participantes, siendo el deportista con el puntaje más alto el ganador en la disciplina. Si ambos puntajes son iguales, la competencia termina en un empate sin ganadores.

Para esta versión de la *DCCumbre Olímpica*, los deportistas competirán en 4 disciplinas individuales: **Atletismo**, **Ciclismo**, **Gimnasia** y **Natación**.

4.3.1. Atletismo

Es considerado el deporte organizado más antiguo del mundo y abarca numerosas disciplinas, pero sin duda la más esperada en las competencias es la carrera de los 100 metros planos. El atletismo **no requiere implementos adicionales** y el riesgo asociado a esta disciplina es `0.2`.

El ganador de la carrera será el deportista que obtenga el mayor puntaje según la siguiente fórmula:

$$\text{máx}(\texttt{PUNTAJE\_MINIMO}, 0.55 * \texttt{velocidad} + 0.2 * \texttt{resistencia} + 0.25 * \texttt{moral})$$

Donde `velocidad`, `resistencia` y `moral` son los atributos del deportista.

4.3.2. Ciclismo

Una de las variantes de este popular deporte es el ciclismo en ruta, el cual es desarrollado en carretera y por lo mismo el riesgo de lesionarse es de `0.35`. Para competir **sí se requiere de implementos especiales**.

El ganador de la carrera será el deportista que obtenga el mayor puntaje según la siguiente fórmula:

$$\text{máx}(\texttt{PUNTAJE\_MINIMO}, 0.47 * \texttt{velocidad} + 0.36 * \texttt{resistencia} + 0.17 * \texttt{flexibilidad})$$

Donde `velocidad`, `resistencia` y `flexibilidad` son los atributos del deportista.

4.3.3. Gimnasia

Es una de las disciplinas que definen a la *DCCumbre Olímpica*. Los deportistas en esta disciplina destacan por su flexibilidad y manejo del cuerpo. **Sí requiere de implementos especiales** y el riesgo asociado es de `0.3`.

El ganador de la competencia será el deportista que obtenga el mayor puntaje según la siguiente fórmula:

$$\text{máx}(\texttt{PUNTAJE\_MINIMO}, 0.5 * \texttt{flexibilidad} + 0.3 * \texttt{resistencia} + 0.2 * \texttt{moral})$$

Donde `flexibilidad`, `resistencia` y `moral` son los atributos del deportista.

4.3.4. Natación

Es el deporte más practicado en el mundo, posee 4 estilos de nado y **no requiere de implementos adicionales**. El riesgo de este deporte es de `0.25`.

El ganador de la carrera será el deportista que obtenga el mayor puntaje según la siguiente fórmula:

$$\text{máx}(\texttt{PUNTAJE\_MINIMO}, 0.45 * \texttt{velocidad} + 0.3 * \texttt{resistencia} + 0.25 * \texttt{flexibilidad})$$

Donde `velocidad`, `resistencia` y `flexibilidad` son los atributos del deportista.

4.4. Campeonato

El campeonato es la cita deportiva más grande a nivel mundial, por lo que existen una serie de responsabilidades que debe cumplir. Dentro de sus deberes se encuentran el fiscalizar la acción de las Delegaciones, llevar a cabo las competencias de los diferentes Deportes y premiar a los Deportistas ganadores, entre otros. El campeonato deberá contar con los siguientes atributos:

- **Día actual**: Un `int` que indique en que día de competencia actual se encuentra el usuario, donde la cuenta comienza en el día `0`.
- **Medallero**: Un `dict` donde las llaves sean los nombres de los equipos y los valores sean un `int` que indique el número de competencias ganadas que lleva cada delegación. Al concluir los días de competencia, estos valores serán los que indiquen qué delegación se coronará campeona de la *DCCumbre Olímpica*.

Por otro lado, la competencia debe incluir los siguientes métodos:

- **Realizar las competencias del día**:

  Tras cada jornada de entrenamiento, se hará un día de competencia. En cada día de competencia se hará **una competencia por deporte**, es decir, habrá cuatro competencias por día, donde el usuario deberá escoger un deportista por competencia que represente a la delegación. El usuario **puede escoger a un mismo deportista para que actúe en diferentes competencias**, pero se arriesga a que este se lesione durante una de ellas y no pueda competir en las competencias que le queden por disputar, implicando la pérdida automática de dichas competencias.

  Luego de tener a los competidores definidos, las competencias serán simuladas en el mismo orden en el cual se escogieron sus protagonistas, utilizando las fórmulas especificadas para cada deporte en la sección Deportes.

  Finalmente, se deberán escribir los resultados obtenidos en cada competencia en el archivo `resultados.txt`.

  Luego de dictaminar los ganadores de las cuatro competencias, se deberá pasar al día siguiente que corresponderá a un día de entrenamiento.

- **Premiar a los Deportistas y sus Delegaciones**: Concluida una competencia, la organización deberá premiar a aquella delegación que haya resultado victoriosa en la prueba, otorgándole la medalla de ganador. El ganador de la competencia recibirá una bonificación de `20` de moral, que ayudará a subir la moral general de su delegación, mientras que el deportista que no resulte ganador experimentará una baja de `10` en su moral y `0.02` de excelencia y respeto. Por último, la delegación ganadora recibirá `100` DCCoins como premio por haber alcanzado el primer lugar de la competencia.
- **Calcular el nivel de Moral de las Delegaciones**: La moral de cada Delegación deberá calcularse y mostrarse en consola al inicio de cada día de entrenamiento y de competencia. Como se indica en la sección Delegaciones, la fórmula consistirá únicamente en el promedio de la moral de todos los deportistas que pertenecen a cada delegación.
- **Mostrar estado de la Delegación y sus Deportistas**: La organización deberá poder ser capaz de mostrar tanto los atributos importantes de ambas Delegaciones, como los de cada Deportista que las componen. Así, para las Delegaciones, se debe mostrar: su excelencia y respeto, implementos deportivos, implementos médicos y moral. También deberá mostrar la cantidad de medallas de las delegaciones en ese instante y la cantidad de DCCoins que poseen. Por el lado de los Deportistas pertenecientes a cada delegación, deberá mostrar el nombre del deportista y sus atributos: Velocidad, Resistencia, Flexibilidad y el `bool` que indica si el deportista está lesionado o no. Por último, se deberá mostrar el estado actual de la competencia, que incluye el día actual en el cual está, y los

resultados que ha obtenido cada delegación en las diferentes competencias. Puedes usar el formato que desees, mientras la información se presente de forma clara y completa. A continuación se muestra un ejemplo de cómo mostrar el estado:

```
                ESTADO DE LAS DELEGACIONES Y DEPORTISTAS
------------------------------------------------------------------------
IEEEsparta
Entrenador: Lily416
Moral del equipo: 60
Medallas: 5
Dinero: 550

Excelencia y respeto: 0.5
Implementos deportivos: 0.8
Implementos médicos: 0.6

Equipo deportivo
Nombre deportista    | Velocidad | Resistencia | Flexibilidad | Lesión
Usain Bolt                1           0.6           0.5        False
Jyri Tapani Aalto         0.7         0.9           0.7        False
Ami Kondo                 0.3         0.4           0.4        True
Mathias Ntawulikura       0.5         0.4            1         True
Hilary Skaryski           0.9         0.9           0.8        False
************************************************************************
DCCrotona
Entrenador: Gatochico
Moral del equipo: 40
Medallas: 3
Dinero: 350

Excelencia y respeto: 0.3
Implementos deportivos: 0.7
Implementos médicos: 0.4

Equipo deportivo
Nombre deportista    | Velocidad | Resistencia | Flexibilidad | Lesión
Tomás González            0.7         0.5            1         False
Martina Vondrov           0.9         0.3           0.2        True
Hugo Vonlanthen           0.5         0.1           0.8        False
Mikhail Yakovlevich       0.4         0.2           0.4        False
Marianne Vos               1          0.9           0.8        False
------------------------------------------------------------------------
Día 5 : Entrenamiento

Medallero
Deporte         |   IEEEsparta    |  DCCrotona
Atletismo             2               0
Ciclismo              1               1
Gimnasia              0               2
```

```
Natación                2                0
--------------------------------------------------------------------------
```

Figura 2: Ejemplo del estado de las delegaciones y deportistas

5. Archivos

En esta tarea se te entregarán dos archivos de extensión `.csv`[5] con información sobre las delegaciones y los deportistas. Tu programa también deberá ser capaz de generar un archivo `resultados.txt` que sea capaz de almacenar todos los resultados de la *DCCumbre Olímpica*, separados por día de competencia. Por último, deberás crear y utilizar un archivo `parametros.py` en donde deberás almacenar todos los parámetros presentados a lo largo del enunciado.

5.1. `delegaciones.csv`

Este archivo incluye información de las dos delegaciones que participan en la *DCCumbre Olímpica*, que deberás cargar al momento de instanciarlas:

Nombre	Tipo de Dato	Descripción
Delegacion	`str`	Indica el tipo de delegación.
Equipo	`list`	Lista con los deportistas de la delegación, separados por `;`
Medallas	`int`	Cantidad de medallas que tiene la delegación.
Moral	`int`	Indica nivel de ánimo y espíritu de la delegación.
Dinero	`int`	Indica el dinero que posee la delegación.

La primera línea de este archivo corresponderá al *header* o encabezado y este determinará el **orden en que aparecen los datos de las delegaciones**. Considera que el contenido y orden de estas columnas puede variar, por lo que tu programa debe ser capaz de adaptarse a dichos cambios. Por último, tu programa no debe almacenar cambios, por lo que este archivo no debería ser modificado en ningún momento.

Un ejemplo de cómo se vería el archivo es el siguiente:

```
Equipo,Dinero,Moral,Delegacion,Medallas
Usain Bolt;Michael Phelps,300,70.5,IEEEsparta,0
Larisa Latynina;Tomas Gonzalez;Nicolas Massu,400,60,DCCrotona,0
```

5.2. `deportistas.csv`

Este archivo contiene información sobre los deportistas disponibles para participar en la *DCCumbre Olímpica*:

[5] Los archivos de extensión `.csv` son un tipo de documento que representan tablas, en donde las columnas se separan por comas y las filas por saltos de línea.

Nombre	Tipo de Dato	Descripción
Nombre	`str`	Indica nombre del deportista.
Velocidad	`int`	Indica velocidad del deportista.
Resistencia	`int`	Indica resistencia del deportista.
Flexibilidad	`int`	Indica flexibilidad del deportista.
Moral	`int`	Indica la estabilidad emocional del deportista frente a una competencia.
Lesionado	`bool`	Indica si el deportista está lesionado o no.
Precio	`int`	Indica el costo de contratar al deportista.

Al igual que `delegaciones.csv`, la primera línea de este archivo corresponderá al *header* y este determinará el orden en que aparecen los datos de los deportistas. Estos también pueden aparecer en cualquier orden y tu programa deberá adecuarse a ello.

Por último, tu programa no debe almacenar cambios, por lo que este archivo no debería ser modificado en ningún momento.

5.3. `resultados.txt`

Este archivo es una bitácora de lo acontecido en la *DCCumbre Olímpica*. En este deben quedar registrado los datos de lo sucedido en un día de competencias. Estos datos son: `Día, Competencia, Delegación Ganadora y Deportista Ganador`. Un ejemplo de cómo se debería ver este archivo es el siguiente:

```
RESULTADOS DÍA A DÍA DCCUMBRE OLÍMPICA
---------------------------------------
Día: 2

Competencia: Natación
Delegación Ganadora: IEEEsparta
Deportista Ganador: Michael Phelps

Competencia: Atletismo
Delegación Ganadora: DCCrotona
Deportista Ganador: Usain Bolt

Competencia: Gimnasia
Delegación Ganadora: DCCrotona
Deportista Ganador: Tomas Gonzalez

Competencia: Ciclismo
Delegación Ganadora: IEEEsparta
Deportista Ganador: Chris Hoy

***************************************
Día 4:

Competencia: Natación
Delegación Ganadora: DCCrotona
Deportista Ganador: Jenny Thompson
```

```
Competencia: Atletismo
Delegación Ganadora: DCCrotona
Deportista Ganador: Jesse Owens

Competencia: Gimnasia
Delegación Ganadora: IEEEsparta
Deportista Ganador: Simone Biles

Competencia: Ciclismo
Delegación Ganadora: IEEEsparta
Deportista Ganador: Jason Kenny

**************************************
Día 6:
                    .  .  .
```

Figura 3: Ejemplo de Resultados de Días de Competencia

No importa el orden en que aparecen las competencias, pero **sí** importa el orden que aparecen los días. Este archivo se debe actualizar **al término de cada día de competencias**.

5.4. `parametros.py`

Para esta tarea se entregará un archivo llamado `parametros.py`, el cual se encuentra incompleto y debe ser completado. En este se deben encontrar todos los parámetros mencionados a lo largo del enunciado, es decir, palabras y números que se encuentran en `ESTE_FORMATO`, además de todos los *paths* y cualquier otro valor constante que vayas a utilizar en tu código.

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

```
1 CIEN = 100 # mal parámetro
2 POPULARIDAD_MINIMA_DELEGACION = 0 # buen parámetro
3 PROBABILIDAD_ACCIDENTARSE_GIMNASIA = 0.3 # buen parámetro
```

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

```
1 LARGO = 5
2 ANCHO = 6
3 ALTO = 10
4 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*[6] dentro de tu código, también conllevará un descuento, pues esto es una mala práctica.

A diferencia de la tarea anterior, el archivo `parametros.py` **no** se debe ignorar y **debe ser subido a su repositorio**. De lo contrario, no será posible corregir tu tarea.

6. Avance de tarea

En conjunto con el programa, deberás realizar un **diagrama de clases** modelando las entidades necesarias de la *DCCumbre Olímpica*. Este diagrama se entregará en dos ocasiones:

Una versión preliminar que refleje cómo planeas modelar tu programa, que corresponderá al **avance** de esta tarea. A partir de los avances entregados, se te brindará un *feedback* general de lo entregado y además, te permitirá optar por **hasta 2 décimas** adicionales en la nota final de tu tarea.

Luego de esto, junto a la entrega final, deberás entregar una **versión final** de tu diagrama que **represente fielmente** la modelación del problema usada en tu programa.

En ambos casos, el diagrama deberá:

- Entregarse en **formato PDF o de imagen**[7].
- Contener todas las clases junto con sus atributos y métodos. También deberás identificar cuales 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.

7. `.gitignore`

Para esta tarea **deberás crear un archivo** `.gitignore` para ignorar los archivos indicados, este deberá estar dentro de tu carpeta `Tareas/T01/`. Puedes encontrar un ejemplo de `.gitignore` en el siguiente ***link***.

Los archivos a ignorar en esta tarea son:

- `delegaciones.csv`
- `deportistas.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__`.

[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.

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

Para este punto es importante que hagan un correcto uso del archivo `.gitignore`, es decir, el hecho que los archivos indicados no se suban al repositorio , debe ser **debido al uso del archivo `.gitignore`** y no debido a otros medios.

8. Entregas atrasadas

Posterior a la fecha de entrega de la tarea se abrirá un formulario de Google Forms. En caso de que desees que se corrija un *commit* posterior al recolectado, deberás señalar el nuevo *commit* en el *form*.

El plazo para rellenar el *form* será de 24 horas. En caso de que no lo contestes en dicho plazo, se procederá a corregir el *commit* recolectado.

9. 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 hacer *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 jefe de Bienestar a su **correo**.

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`.
- Tu tarea deberá encontrarse subida en la rama `master` de tu repositorio.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier módulo Python está prohibido. Pregunta en la *issue* especial del **foro** si es que es posible utilizar algún módulo en particular.
- Debes adjuntar un archivo `README.md` **conciso y claro**, donde describas los alcances de tu programa, cómo correrlo, los módulos usados, los supuestos hechos, y las referencias a código externo. **Tendrás hasta 24 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).