**Tarea Nº2**

**Evaluación Unidad 2 y 3: Fundamentos de Programación en Python**

**Fecha de entrega:** 28 de junio 2021

**Modalidad:** En grupo

**I. Objetivo.**

El objetivo de la presente tarea es evaluar tu capacidad para:

1. Llevar a cabo un programa completo en el lenguaje Python, utilizando elementos Condicionales, Ciclos, E/S, Strings ([],len,+), Listas, Funciones, uso de módulos mediante import y Clases
2. Resolver un problema, considerando que se cuenta con el procedimiento en lenguaje natural, que se debe implementar para resolverlo.
3. Seguir **en forma precisa la interfaz e instrucciones solicitadas**.

**II. Enunciado.**

El evento más importante del mundo de los juegos acaba de anunciar una nueva versión para este año, en la que cada uno de los países que compone el globo reúne a millares de niños, jóvenes y adultos para enfrentarse en un duelo campal para demostrar quién es el mejor. El evento consiste en que los participantes, denominados entrenadores, se enfrentan entre sí en torneos locales de cada país junto a sus monstruos de bolsillo llamados Pokémon, criaturas con poderes inimaginables que vuelven cada enfrentamiento una experiencia emocionante, tanto para los entrenadores como para los espectadores.

El torneo consiste en que cada entrenador posee como máximo de 6 criaturas de las cuales sólo pueden utilizar 4 en cada enfrentamiento. La dificultad de este evento radica en que actualmente existen 898 tipos de Pokémon y por lo tanto existen muchas combinaciones posibles de equipos:

Cada Pokémon tiene sus propias estadísticas, ventajas y desventajas que lo vuelven único. Las estadísticas base de cada criatura son las siguientes:

* HP o Puntos de vida, Puntos de Ataque, Puntos de Defensa, Puntos de Ataque Especial, Puntos de Defensa Especial, y Puntos de Velocidad.

Las ventajas o desventajas de cada Pokémon al enfrentarse a otro están determinadas por su tipo y el de su contrincante:

* Normal, Fuego, Agua, Eléctrico, Planta, Hielo, Lucha, Veneno, Tierra, Volador, Psíquico, Bicho, Roca, Fantasma, Dragón, Siniestro, Acero, y Hada.

Un entrenador, cuyo nombre es Nabu Go, ha participado en innumerables torneos y ha recopilado la información completa de los Pokémon existentes en un archivo CSV (**pokemon\_data.csv**) y la efectividad de cada tipo de Pokémon (**tabla\_efectividad.csv**), además de un script en Python (**moves.py**) que devuelve el poder de ataque de un movimiento de cada personaje (consultar el **Anexo D** para ver su funcionamiento)

El CSV (delimitado por comas) del archivo pokemon\_data.csv está organizado de la siguiente forma:

**nombre,tipo,puntos\_de\_vida,puntos\_de\_ataque\_fisico\_base,puntos\_de\_defensa\_fisica\_base, puntos\_de\_ataque\_especial\_base,puntos\_de\_defensa\_especial\_base,puntos\_de\_velocidad\_base,movimientos posibles delimitado por ;,**

Ejemplo de los datos:

**charizard,fire,78,84,78,109,85,100,aerialace;aircutter;airslash;attract;bide;blastburn;bodyslam;brickbreak;brutalswing;bulldoze;captivate;confide;counter;curse;cut;defensecurl;defog;dig;doubleedge;doubleteam;dragonbreath;dragonclaw;dragondance;dragonpulse;dragonrage;dragontail;dynamicpunch;earthquake;echoedvoice;ember;endure;facade;fireblast;firefang;firepledge;firepunch;firespin;fissure;flameburst;flamecharge;flamethrower;flareblitz;fling;fly;focusblast;focuspunch;frustration;furycutter;gigaimpact;growl;headbutt;heatwave;hiddenpower;holdhands;honeclaws;hyperbeam;incinerate;inferno;irontail;leer;megakick;megapunch;metalclaw;mimic;mudslap;naturalgift;ominouswind;outrage;overheat;poweruppunch;protect;rage;reflect;rest;return;roar;rockslide;rocksmash;rocktomb;roost;round;sandstorm;scaryface;scratch;secretpower;seismictoss;shadowclaw;skullbash;skydrop;slash;sleeptalk;smokescreen;snore;solarbeam;steelwing;strength;submission;substitute;sunnyday;swagger;swift;swordsdance;tailwind;takedown;thunderpunch;toxic;twister;willowisp;wingattack;workup,**

Nabu lo ha contactado a usted para desarrollar una herramienta​ que permita a los entrenadores planificar mejor sus estrategias en las batallas de torneos competitivos. La idea es que el software pueda calcular el daño aproximado que provocará un ataque considerando el tipo, el poder del ataque, la potencia de ataque del Pokémon y la resistencia del Pokémon rival.

Para lograr el objetivo debe realizar lo siguiente:

1. Debe calcular los puntos estadísticos (Stats) de ambos Pokémon utilizando las siguientes fórmulas:
2. Fórmula para determinar los **puntos de vida** del Pokémon:

Ecuación 1 - Fórmula para calcular los puntos de vida

1. Fórmula para calcular los otros puntos estadísticos (Ataque, Defensa, Ataque Especial, Defensa Especial y Velocidad):

Ecuación 2 - Fórmula para calcular otros puntos estadísticos

Considere que:

* El nivel () de los Pokémon es un valor constante = 50
* Los puntos base () del “stat” de los Pokémon es un valor variable que está en el archivo CSV
* Los puntos individuales () son un valor constante = 31
* Los puntos de esfuerzo () son un valor constante = 250

1. Una vez obtenido los “Stats”, debe calcular la potencia de daño que hará un ataque que seleccione el usuario desde el Pokémon Atacante al Pokémon Rival utilizando la siguiente formula:

Ecuación 3 - Fórmula para calcular el daño aproximado

Considere que:

* Donde es el daño aproximado que provocará el ataque,
* es el nivel del Pokémon atacante, es un valor constante = 50,
* es la potencia del ataque y es un valor variable que entrega la función **get\_move** del script moves.py,
* son los puntos de ataque del Pokémon atacante que se obtiene a partir de la fórmula para calcular los otros “stats” (Ecuación 2). Nota: dependiendo de la categoría del ataque, si especial o físico, el valor que utilizará esta variable será ataque especial o física, respectivamente,
* son los puntos de defensa del Pokémon rival que se obtiene a partir de la fórmula para calcular los otros “stats” (Ecuación 2). Dependiendo de la categoría del atacante, si especial o físico, el valor que utilizará esta variable será defensa especial o física respectivamente,
* es un valor dinámico que se obtiene utilizando la siguiente formula:

Ecuación - Fórmula para calcular el daño aproximado

* es un valor aleatorio entre 0.85 y 1,
* es 1.2 siempre y cuando el ataque ejecutado sea del mismo **tipo** del Pokémon atacante de lo contrario es 1
* El tipo de ataque () determinará si el ataque es efectivo o no, considerando los siguientes valores:
  + 0: No tiene efecto (Ejemplo: Normal contra Fantasma),
  + 0.5: No es efectivo (Ejemplo: Fuego contra Agua),
  + 1: Efectividad normal (Ejemplo: Tierra contra Lucha),
  + 2: Super Efectivo (Ejemplo: Dragón contra Dragón).
* Esta información se puede obtener desde el archivo “**tabla\_efectividad.csv**”, consultar **Anexo C** para ver su estructura.
* **Importante:** Existen Pokémon que tiene un solo tipo, como Blastoise que es solo de tipo Agua(**water**), y otros que tienen 2 tipos, como Charizard que es Fuego(**fire**) y Volador(**flying**). Para esta aplicación considere solo el primer tipo, es decir, para el caso de Charizard, solo tome que es Fuego.
* Si la función **get\_move** entrega como valor 0 no se puede calcular el daño, se debe validar y solicitar al usuario que seleccione otro ataque.

**III. Consideraciones en la programación.**

En la solución que implemente, debe considerar que:

1. El programa debe ser desarrollado en lenguaje Python 3.7 o superior.
2. Su programa solo debe soportar ingreso de números y texto.
3. Debe aplicar los contenidos de la unidad 1, 2 y 3 de la asignatura y módulos que su profesor explicará en clases.
4. El programa debe seguir las reglas del juego indicadas en el enunciado.

**IV. Sobre la entrega, atrasos y faltas a la ética.**

1. Deben formar grupos de máximo de 2 personas.
2. No se aceptará la entrega de tareas atrasadas.
3. Debe subir su trabajo a la plataforma <http://unab.blackboard.com>, en una casilla que se habilitará especialmente para esto. El trabajo debe ser subido por **SOLO UNO** de los integrantes del grupo.
4. Debes subir un archivo .**zip** cuyo nombre debe ser la combinación de los ruts de los integrantes del grupo con formato **rutcompleto1\_rutcompleto2.zip**. Este archivo debe contener los scripts necesarios para ejecutar su solución.

Ejemplo:

Si los integrantes tienen los RUTs **12.345.678-9** y **98.654.321-0**, el archivo se debe llamar: **123456789\_9876543210.zip**

1. No puede intercambiar partes del código de tu solución con otros grupos o compañeros, aunque sean de otras secciones. Tampoco puedes utilizar una solución disponible en Internet para realizar tu tarea. En caso de detectarse copia, los involucrados serán calificados con nota 1.0.

En caso de dudas con el enunciado o si requieres orientación para el desarrollo de su trabajo, por favor consulte con su profesor.

**Anexo A: Ejemplo de cálculo.**

Suponiendo que el usuario escoge a Latias como atacante y a Absol como rival los puntos según el nivel serían los siguientes:

Datos base de los Pokémon:

|  |  |  |
| --- | --- | --- |
| Pokémon Atacante: |  | Pokémon Rival: |
| |  |  | | --- | --- | |  | | | Nombre | Latias | | Puntos de vida (HP) | 80 | | Ataque (Physical attack) | 80 | | Defensa (Physical defense) | 90 | | Ataque especial (Special attack) | 110 | | Defensa especial (Special defense) | 130 | | Velocidad (Speed) | 110 | | VS | |  |  | | --- | --- | |  | | | Nombre | Absol | | Puntos de vida (HP) | 65 | | Ataque (Physical attack) | 130 | | Defensa (Physical defense) | 60 | | Ataque especial (Special attack) | 75 | | Defensa especial (Special defense) | 60 | | Velocidad (Speed) | 75 | |

Luego, aplicamos la fórmula para calcular los puntos de vida (Ecuación 1) y para calcular los otros “stats” (Ecuación 2) para ambos Pokémon:

Latias:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Absol:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Una vez aplicadas las fórmulas obtendremos los stats al **nivel 50** de cada Pokémon:

|  |  |  |
| --- | --- | --- |
| Pokémon Atacante (Nivel 50): |  | Pokémon Rival (Nivel 50): |
| |  |  | | --- | --- | | Nombre | Latias | | Puntos de vida (**HP**) | 172.97 | | Ataque (**Physical** **attack**) | 117.97 | | Defensa (**Physical** **defense**) | 127.97 | | Ataque especial (**Special** **attack**) | 147.97 | | Defensa especial (**Special** **defense**) | 167.97 | | Velocidad (**Speed**) | 147.97 | | VS | |  |  | | --- | --- | | Nombre | Absol | | Puntos de vida (**HP**) | 157.97 | | Ataque (**Physical** **attack**) | 167.97 | | Defensa (**Physical** **defense**) | 97.97 | | Ataque especial (**Special** **attack**) | 112.97 | | Defensa especial (**Special** **defense**) | 97.97 | | Velocidad (**Speed**) | 112.97 | |

Ahora nos disponemos calcular el daño que hará Latias a Absol. Supongamos que Latias usará Pulso Dragón (Dragon Pulse), la Pokédex devolverá lo siguiente:

|  |  |
| --- | --- |
| Movimiento | Pulso Dragón (Dragon Pulse) |
| Poder base (**Power**) | 85 |
| Categoría (**category**) | Especial (**special**) |
| Tipo (**type**) | Dragón (**dragon**) |

La fórmula para calcular el daño (Ecuación 3) junto con la de *Modifier* (Ecuación 4) queda de la siguiente manera (valor random será 0.91):

Aplicando formulas:

→ 1.104

→ 64.57 puntos de daño

Finalmente restamos los puntos de daño a los puntos de HP de Absol:

**Anexo B: Salida esperada del programa.**

Bienvenido al simulador

Ingrese el nombre del primer Pokémon: **latias**

Nombre del Pokémon seleccionado: Latias

Estadísticas base del Pokémon:

- HP = 80

- Ataque = 80

- Defensa = 90

- Ataque especial= 110

- Defensa especial = 130

- Velocidad = 110

Movimientos que puede aprender el pokémon:

0 - aerialace

1 - allyswitch

2 - attract

3 - bodyslam

4 - bulldoze

5 - calm mind

6 - captivate

7 - chargebeam

8 - charm

9 - confide

10 - covet

11 - cut

12 - defog

13 - dive

14 - doubleedge

15 - doubleteam

16 - dracometeor

17 - dragonbreath

18 - dragonclaw

19 - dragonpulse

20 - dreameater

21 - earthquake

22 - endure

23 - energyball

24 - facade

25 - flash

26 - fly

27 - frustration

28 - furycutter

29 - gigaimpact

30 - grassknot

31 - guardsplit

32 - healingwish

33 - healpulse

34 - helpinghand

35 - hiddenpower

36 - honeclaws

37 - hyperbeam

38 - icebeam

39 - icywind

40 - laserfocus

41 - lastresort

42 - lightscreen

43 - magiccoat

44 - magicroom

45 - mimic

46 - mistball

47 - mudslap

48 - natural gift

49 - outrage

50 - protect

51 - psychic

52 - psycho shift

53 - psych up

54 - psyshock

55 - psywave

56 - raindance

57 - recover

58 - reflect

59 - reflecttype

60 - refresh

61 - rest

62 - retaliate

63 - return

64 - roar

65 - roleplay

66 - roost

67 - round

68 - safeguard

69 - sandstorm

70 - secretpower

71 - shadowball

72 - shadowclaw

73 - shockwave

74 - sleeptalk

75 - snore

76 - solarbeam

77 - steelwing

78 - storedpower

79 - substitute

80 - suckerpunch

81 - sunnyday

82 - surf

83 - swagger

84 - swift

85 - tailwind

86 - telekinesis

87 - thunder

88 - thunderbolt

89 - thunder wave

90 - toxic

91 - trick

92 - twister

93 - waterfall

94 - waterpulse

95 - watersport

96 - whirlpool

97 - wish

98 - zenheadbutt

Seleccione un ataque a ejecutar: **19**

El ataque seleccionado es: dragonpulse

Poder de ataque es: 85

El hp al nivel 50 de Latias es 172.97642353760523

El atk al nivel 50 de Latias es 117.97642353760524

El def al nivel 50 de Latias es 127.97642353760524

El spa al nivel 50 de Latias es 147.97642353760523

El spd al nivel 50 de Latias es 167.97642353760523

El spe al nivel 50 de Latias es 147.97642353760523

Ingrese el nombre a atacar Pokémon: **absol**

Nombre del Pokémon seleccionado: Absol

El hp al nivel 50 de Absol es 157.97642353760523

El daño que realizó Latias a Absol fue de: 64.56879167508306

Absol quedó con un HP de: 93.40763186252217

**Anexo C: Estructura Tabla de efectividad**

El archivo “tabla\_efectividad.csv” es un archivo delimitado por comas la cual tiene la siguiente estructura:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Defensa → Ataque ↓ | **Normal** | **Fire** | **Water** | **Electric** | **Grass** | **Ice** | **Fighting** | **Poison** | **Ground** | **Flying** | **Psychic** | **Bug** | **Rock** | **Ghost** | **Dragon** | **Dark** | **Steel** | **Fairy** |
| **Normal** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.5 | 0 | 1 | 1 | 0.5 | 1 |
| **Fire** | 1 | 0.5 | 0.5 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 0.5 | 1 | 0.5 | 1 | 2 | 1 |
| **Water** | 1 | 2 | 0.5 | 1 | 0.5 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 0.5 | 1 | 1 | 1 |
| **Electric** | 1 | 1 | 2 | 0.5 | 0.5 | 1 | 1 | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 0.5 | 1 | 1 | 1 |
| **Grass** | 1 | 0.5 | 2 | 1 | 0.5 | 1 | 1 | 0.5 | 2 | 0.5 | 1 | 0.5 | 2 | 1 | 0.5 | 1 | 0.5 | 1 |
| **Ice** | 1 | 0.5 | 0.5 | 1 | 2 | 0.5 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 0.5 | 1 |
| **Fighting** | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 0.5 | 1 | 0.5 | 0.5 | 0.5 | 2 | 0 | 1 | 2 | 2 | 0.5 |
| **Poison** | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 0.5 | 0.5 | 1 | 1 | 1 | 0.5 | 0.5 | 1 | 1 | 0 | 2 |
| **Ground** | 1 | 2 | 1 | 2 | 0.5 | 1 | 1 | 2 | 1 | 0 | 1 | 0.5 | 2 | 1 | 1 | 1 | 2 | 1 |
| **Flying** | 1 | 1 | 1 | 0.5 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 0.5 | 1 | 1 | 1 | 0.5 | 1 |
| **Psychic** | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 0.5 | 1 | 1 | 1 | 1 | 0 | 0.5 | 1 |
| **Bug** | 1 | 0.5 | 1 | 1 | 2 | 1 | 0.5 | 0.5 | 1 | 0.5 | 2 | 1 | 1 | 0.5 | 1 | 2 | 0.5 | 0.5 |
| **Rock** | 1 | 2 | 1 | 1 | 1 | 2 | 0.5 | 1 | 0.5 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 0.5 | 1 |
| **Ghost** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 0.5 | 1 | 1 |
| **Dragon** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 0.5 | 0 |
| **Dark** | 1 | 1 | 1 | 1 | 1 | 1 | 0.5 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 0.5 | 1 | 0.5 |
| **Steel** | 1 | 0.5 | 0.5 | 0.5 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 0.5 | 2 |
| **Fairy** | 1 | 0.5 | 1 | 1 | 1 | 1 | 2 | 0.5 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 0.5 | 1 |

**Anexo D: Funcionamiento del script moves.py**

El script “moves.py” contiene una función llamada “get\_move(move)” la recibe un parámetro de tipo STR y retorna una lista con el detalle del movimiento seleccionado la cual tiene la siguiente estructura:

[Nombre de movimiento, poder del movimiento(**Power**), tipo del movimiento(**type**), categoría del movimiento(**category**)]

Suponiendo que el usuario escoge a “Dragon Pulse” entonces la función se debe ejecutar de la siguiente manera:

Consola de Python:

>>> from moves import get\_move

>>> get\_move('dragonpulse')

['Dragon Pulse', 85, 'dragon', 'special']

>>>