Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de Computación
IIC2143 – Ingeniería de Software 2021–1

Tarea Pokemon Ruby

ENTREGA: 12 DE ABRIL, 23:59.

1. Indicaciones Generales

Esta tarea es de caracter **opcional**. Las personas que realicen esta tarea obtendrán hasta **5 décimas** extra en su primera evaluación escrita. **Se recomienda fuertemente realizarla**, ya que es una gran oportunidad para aprender Ruby (lenguaje sobre el que se construye el framework Ruby on Rails utilizado durante el curso). **El nombre del archivo principal debe ser main.rb**.

2. Introducción

Los ayudantes de Ingeniería de Software queremos realizar la liga Pokemon ruby para ver quien de nosotros es el mejor entrenador Pokemon. Estamos tan ocupados que no tenemos tiempo para poder realizar esta gran competencia, por eso tu un alumno muy interesado en aprender a usar Ruby, te ofreces como voluntario para ayudarnos a descubrir quién tiene el mejor equipo.

3. Modelación

La modelación presentada es sugerida, pero no es obligatoria, aunque se deben tener **todas las clases** descritas en esta sección.

Clase Pokemon

- id: int
- nombre: str
- vida_maxima: int
- vida_actual: int
- velocidad: int
- defensa: int
- ataque: int

Clase Entrenador

- id: int
- nombre: str
- pokemones: List[Pokemon]
- puntos: int

Clase Combate

- entrenadores: [id_entrenador1, id_entrenador2]
- ganador: int (id del entrenador ganador del combate)
- pokemones_vivos_ganador: List[int] (ids de los Pokemones que quedaron vivos luego de ganar el combate)

Clase Liga

- Atributos
  - entreandores: List[Entrenador]
  - combates: List[Combate]
- Metodos
  - realizar_combates() (La idea es que este método poble el programa con los resultados de los combates. En otras palabras, creará todas las instancias de la clase Combate).

4. Combates

Los combates se realizarán de la siguiente manera:

Primero, cada entrenador combatirá con los demás entrenadores **1 sola vez** (un combate del entrenador 1 vs entrenador 2 es el mismo que el del entrenador 2 vs entrenador 1).

Segundo, durante el combate mismo los Pokemones combatirán en el orden que vienen en la lista de cada entrenador y solo entrará el siguiente cuando el que esta combatiendo se queda sin puntos de vida. El combate es por turnos y el primero en atacar será el que tiene mas velocidad (**ojo que esto es solo para comenzar el combate**, cuando un pokemon muere y otro entra los turnos se mantienen). Si hay empate en las velocidades, comenzara el entrenador con menor id.

Por ejemplo si tenemos un combate entre los siguientes entrenadores:

Entrenador 1

- Pokemon 1
  - velocidad: 2

Entrenador 2

- Pokemon 2
  - velocidad: 5
- Pokemon 3
  - velocidad: 1

Aquí comienza el Pokemon 2 porque tiene mayor velocidad. Si el Pokemon 1 ataca y el Pokemon 2 se muere, si el entrenador 2 aun tuviera pokemones de reserva, este mandará al combate a su siguiente Pokemon (Pokemon 3, en este caso). Ahora le tocaría al Pokemon 3, del entrenador 2, atacar independiente de que su velocidad sea menor que la del Pokemon 1.

- **Tip**: las instancias de la clase Combate pueden crearlas una vez que finaliza el combate, así tienen toda la información de este.

Tercero, en cada turno un Pokemon ataca y el otro recibe daño. La vida resultante del pokemon defensor se calcula como:

$VAPD = VAPD - (APA - \frac{DPD}{4})$

Donde :

- $VAPD = Vida\ Actual\ Pokemon\ Defensor$
- $APA = Ataque\ Pokemon\ Atacante$
- $DPD = Defensa\ Pokemon\ Defensor$

Cuarto, cuando la vida de un Pokemon llega a 0, este es derrotado y el siguiente Pokemon en la lista es lanzado al combate.

Quinto, el combate termina cuando alguno de los dos entrenadores se queda sin Pokemones.

Sexto, los Pokemones recuperan toda la vida perdida una vez finalizado el combate.

5. Sistema de puntos

Cada vez que un entrenador derrota a un Pokemon, este ganará 1 punto. Si el entrenador gana el combate, este ganará 3 puntos (adicionales a los que ganó por derrotar Pokemones).

6. Aclaraciones

Al comienzo la vida actual es igual a la vida máxima del Pokemon, pero esta puede cambiar durante cada combate.

El entrenador perdedor puede ganar puntos por derrotar Pokemones.

7. Métodos

Para comprobar que las interacciones se realizaron de manera correcta, se piden los siguientes métodos los cuales deben escribir sus outputs en un archivo llamado **output.txt**:

- entrenadores_liga(): Este método escribirá en un archivo los nombres y puntos de todos los entrenadores con sus Pokemones siguiendo el siguiente formato:

  COMIENZO ENTRENADORES LIGA

  $\{nombre\_entrenador1, puntos\_entrenador1\}\{nombre\_pokemon1\_entrenador1\}....\{nombre\_pokemon6\_entrenador1\}$

{$nombre\_entrenador2, puntos\_entrenador2$}{$nombre\_pokemon1\_entrenador2$}....{$nombre\_pokemon6\_entrenador2$}

...

{$nombre\_entrenadorN, puntos\_entrenadorN$}{$nombre\_pokemon1\_entrenadorN$}....{$nombre\_pokemon6\_entrenadorN$}

FIN ENTRENADORES LIGA

- top_entrenadores(): Este método escribirá en el archivo el id, nombre y puntos de los 5 entrenadores con mayor puntaje, ordenados de manera descendente según sus puntajes, es decir, el entrenador con mayor puntaje irá primero (en caso de empate el entrenador con menor id irá primero). Debe seguir el siguiente formato:

  COMIENZO TOP ENTRENADORES

  {$id\_entrenador1, nombre\_entrenador1, puntos\_entrenador1$}

  ...

  {$id\_entrenador5, nombre\_entrenador5, puntos\_entrenador5$}

  FIN TOP ENTRENADORES

- top_pokemones(): Este método escribirá en el archivo el nombre y la cantidad de combates que sobrevivieron los 3 pokemones que sobrevivieron a la mayor cantidad de combates, en orden descendente. Este debe seguir el siguiente formato:

  COMIENZO TOP POKEMONES

  {$nombre\_pokemon1, combates\_sobrevividos\_pokemon1$}

  {$nombre\_pokemon2, combates\_sobrevividos\_pokemon2$}

  {$nombre\_pokemon3, combates\_sobrevividos\_pokemon3$}

  FIN TOP POKEMONES

8. Archivos CSV

Para cargar los datos se entregarán los siguientes archivos en formato csv: *pokemon.csv* y *entrenadores.csv*. Estos aechivos tienen el siguiente formato:

pokemon.csv

id,nombre,vida_maxima,ataque,defensa,velocidad

1,Bulbasaur,318,49,49,45

2,Ivysaur,405,62,63,60

3,Venusaur,525,82,83,80

4,Charmander,309,52,43,65

entrenadores.csv

id,nombre,pokemon1,pokemon2,pokemon3,pokemon4,pokemon5,pokemon6

1,Ignacio,1,5,10,2,50,84

2,Barbara,5,24,14,103,66,48

3,Cristóbal,138,12,96,104,72,139

9. Ejecución

Para corregir la tarea, los ayudantes ejecutarán el siguiente comando:

ruby main.rb

En su archivo main.rb deberán ejecutar los métodos necesarios para que el archivo output.txt quede con los tres métodos pedidos en el punto 7 en el siguiente orden:

COMIENZO ENTRENADORES LIGA

...

FIN ENTRENADORES LIGA

COMIENZO TOP ENTRENADORES

...

FIN TOP ENTRENADORES

COMIENZO TOP POKEMONES

...

FIN TOP POKEMONES

Los archivos csv siempre se llamarán de la misma manera, pero estos deben estar en una carpeta llamada *data*. Esta carpeta debe estar en el mismo directorio que el archivo main.rb

Para que quede claro, sus archivos tiene que estar ordenados de la siguiente manera:

| main.rb

| data/

– – | pokemon.csv

– – | entrenadores.csv

Pueden crear todos los archivos .rb que crean necesarios, pero main.rb y la carpeta data deben seguir ese orden.

Luego de ejecutar el programa, su directorio se deberá ver de la siguiente manera:

| main.rb

| data/

– – | pokemon.csv

– – | entrenadores.csv

|output.txt

Importante la corrección se realizará de manera automatizada, por lo que se espera que el formato que utilicen para el archivo de *output* sea exactamente el que se pide en el enunciado. Para facilitar su desarrollo, durante la semana (semana del 29 de marzo) se les entregará un archivo de prueba que pordrán utilizar para verificar que, efectivamente, están cumpliendo con esto.

10. Foro

Cualquier duda que tengan pueden preguntarla creando una issue en el repositorio del curso escribiendo en el título "[Tarea] - Pregunta..."

11. Entrega

La entrega de esta tarea se realizará mediante un *assignment* en Canvas, donde deberán subir un archivo .zip sin archivos .csv ni .txt, dentro del cual deben incluir el o los archivos que utilicen en su programa. Tienen hasta el lunes 12 de Abril a las 23:59 para entregar su tarea.