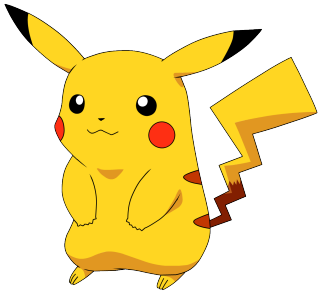
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2233 - PROGRAMACIÓN AVANZADA

Actividad 05

1^o **semestre 2018**
19 de abril

Introducción

Un día común y corriente, el gran Hernán estaba aburrido, por lo que decidió crear un simple juego. Dado que crear una base de datos es demasiado trivial para Hernán, le pidió a uno de sus ayudantes que lo hiciera por él. Como el pobre ayudante tenía mucho trabajo que hacer, decidió ~~robar~~ pedir prestadas **dos bases de datos** para el juego, una con pokémones y otra con entrenadores.



Ejemplo de un pokémon. (Nota: es un Pikachu.)
~~Ojalá que Nintendo no nos demande por esto.~~

Sin embargo, el ayudante no tenía tiempo para hacer las consultas que Hernán le pedía, por lo que decidió pedirte a ti, maestro de la programación funcional, que lo ayudes. Para esto, debes ser capaz de leer los archivos en formato CSV que te entregó el ayudante para, luego, realizar consultas a partir de los datos existentes.

Estructura de archivos CSV

En primer lugar, el repositorio cuenta con dos bases de datos. Además, hay una tabla donde se ilustran los datos que encontrarás en los archivos CSV. Por último, existe una función lista para leer los archivos, que entrega un generador de `namedtuples` con los datos ilustrados a continuación. Estos datos están en formato de *strings*; sin embargo, puedes transformarlos a lo que te resulte adecuado.

Metadata del archivo `pokemondb.csv`

Nombre de columna	Tipo	Descripción
`id`	`int`	Número en la pokédex
`nombre`	`str`	Nombre del pokémon
`tipo_1`	`str`	Primer tipo del pokémon
`tipo_2`	`str`	Segundo tipo del pokémon
`total`	`int`	Suma de las estadísticas del pokémon
`vida`	`int`	Vida base del pokémon
`ataque`	`int`	Ataque base del pokémon
`defensa`	`int`	Defensa base del pokémon
`ataque_especial`	`int`	Ataque especial base del pokémon
`defensa_especial`	`int`	Defensa especial base del pokémon
`velocidad`	`int`	Velocidad base del pokémon
`generacion`	`int`	Generación del pokémon
`legendario`	`bool`	Indica si el pokémon es legendario o no

Metadata del archivo `entrenadoresdb.csv`

Nombre de columna	Tipo	Descripción
`nombre`	`str`	Nombre del entrenador
`tipo_favorito`	`int`	Tipo favorito
`estadistica_favorita`	`str`	Estadística favorita
`legendario`	`bool`	Indica si el entrenador puede usar legendarios o no

Funciones

Para responder a las consultas de Hernán, debes implementar las siguientes funciones en tu código.

- `pokédex_regional(generación, pokémones)`
  Recibe una generación (como un `int` entre 1 y 7) y un **iterable** de pokémones.
  Debe retornar un generador con todos los pokémones que pertenezcan a la generación.

- `obtener_estadística(estadística, pokémon)`
  Recibe una de las seis estadísticas como *string* (*i.e.* `"vida"`, `"ataque"`, `"defensa"`, `"ataque_especial"`, `"defensa_especial"`, `"velocidad"`) y un pokémon en particular.
  Debe retornar, en forma de `int`, el valor de la estadística de ese pokémon.

- `obtener_estadística_promedio(estadística, pokémon)`
  Recibe una de las seis estadísticas como *string* y un **iterable** de todos los pokémones.
  Debe retornar el valor promedio de aquella estadística.

- `pokémones_buena_estadística(estadística, pokémones)`
  Recibe una estadística como *string* y un **iterable** de pokémones.
  Debe retornar un generador con todos los pokémones cuya *estadística* sea mejor que la del promedio en dicha estadística.

- `pokémon_para_entrenador(entrenador, pokémones)`
  Recibe un entrenador y un **iterable** de pokémones.
  Es un generador que retorna, como mucho, a los **seis mejores** pokémones para el equipo del entrenador, ordenados según la estadística favorita del entrenador. Estos pokémones deben cumplir con los requerimientos de `pokémones_buena_estadística` para dicha estadística favorita del entrenador. Además, cualquiera de los tipos del pokémon tiene que ser igual al favorito del entrenador. Por último, debe verificar si es que el entrenador puede utilizar pokémones legendarios.

- `poder_total_entrenador(entrenador, pokémones)`
  Recibe un entrenador y un **iterable** de los pokémones.
  Debe retornar el *poder* del entrenador, que se define como la suma de todas las estadísticas de cada pokémon en su equipo.

  Ejemplo: un entrenador tiene en equipo con Shroomish, Kyogre, Xurkitree, Venonat, Shinx y Oddish. Su *poder* es de 2.423 $(295 + 670 + 570 + 305 + 263 + 320)$. Además, si un entrenador tuviese menos de seis pokémones en su equipo, cada espacio vacío cuenta como 0.

Consultas

Tu código debe ser capaz de recibir las siguientes consultas, imprimiendo los resultados correspondientes.

1) Dado las siguientes combinaciones de `(tipo, estadística)` dadas:

- `("Hoja", "ataque")`
- `("Agua", "defensa_especial")`
- `("Fuego", "ataque_especial")`

El programa debe ser capaz de imprimir los pokémones en común de los entrenadores que tienen el tipo y estadística favorita de esa combinación. Para esto, considere a todos los pokémones disponibles al formar los equipos.

Por ejemplo, si entre los entrenadores que tienen como preferencia el tipo *Fuego* y la estadística favorita como *Velocidad* tuviesen los pokémones Charizard, Arcanine y Moltres en común como parte de su equipo, el programa debería imprimir lo siguiente:

Ejemplo:

> *—— Pokémones en comun: Fuego y Velocidad ——*
> *Charizard*
> *Arcanine*
> *Moltres*

Hint: El uso de `set` y su método `intersection` le podría ser útil.

Uso de ciclos

Debido a que al gran Hernán no le gustó para nada que su ayudante delegara esta actividad a los alumnos, decidió hacerles la vida más difícil: **está prohibido** utilizar `while` y `for` en tus funciones y consultas. Sólo lo puedes usar para listas por comprensión, generadores y para escribir los *prints*. **Cualquier uso de ciclos fuera de estos casos incurre a que la función completa esté incorrecta.**

Notas

- Recuerde que un **iterable** es **cualquier objeto sobre el cual se puede iterar**. En otras palabras, las funciones que reciban iterables no necesariamente deben recibir listas.
- Está fuertemente recomendado realizar la actividad de forma lineal, puesto que es posible que las primeras funciones sean útiles para construir las siguientes.
- No se debe modificar el contenido de los archivos CSV.
- No suba los archivos CSV a su repositorio; de hacerlo, será penalizado. Para lograr esto, debe tener un archivo `.gitignore` en la carpeta `AC05`.
- Puede que las *named tuples* de la librería `collections` sean de gran utilidad.
- Es también útil el uso de `attrgetter` del módulo `operator`.
- Si necesita dos iterables iguales a partir de un único iterable, la función `tee` de la librería `itertools` podría ser de utilidad.

Requerimientos

- (4.60 pts) Funciones
  - (0,70 pts) `pokédex_regional`
  - (0,70 pts) `obtener_estadística`
  - (0,70 pts) `obtener_estadística_promedio`
  - (0,70 pts) `pokémones_buena_estadística`
  - (0,90 pts) `pokémon_para_entrenador`
  - (0,90 pts) `poder_total_entrenador`
- (1.40 pts) Consultas
  - (1.40 pts) Top Pokémones

Entrega

- **Lugar:** En su repositorio de GitHub en la **carpeta** `Actividades/AC05/`
- **Hora:** 16:20