Algoritmos y Programación II
Curso Mendez

.UBAfiuba
FACULTAD DE INGENIERÍA

Atrápalos a Todos!

TP2

Fecha de Presentación: 03/07/2020

Fecha de Vencimiento: 24/07/2020

1. Introducción

La **Pokedex** es una enciclopedia virtual portátil de alta tecnología que los entrenadores Pokémon llevan consigo, para registrar la información de todas las diversas especies Pokémon con las que se encuentran durante su viaje como entrenadores.
La Pokedex funciona de manera simple: cada vez que un entrenador se encuentra con un Pokémon (por ejemplo en una batalla), queda asentada en la enciclopedia la información de la especie a la cual pertenece el Pokémon (numero de identificación, descripción, etc). Adicionalmente se almacena la información particular del Pokémon visto (nivel, nombre, etc), y un historial de Pokémon avistados y capturados por el entrenador.

2. Objetivo

El presente trabajo práctico tiene como finalidad que el alumno se familiarice con las diferentes estructuras de datos dinámicas implementadas, aplicándolas a un problema concreto y poniendo en práctica su uso integrado.

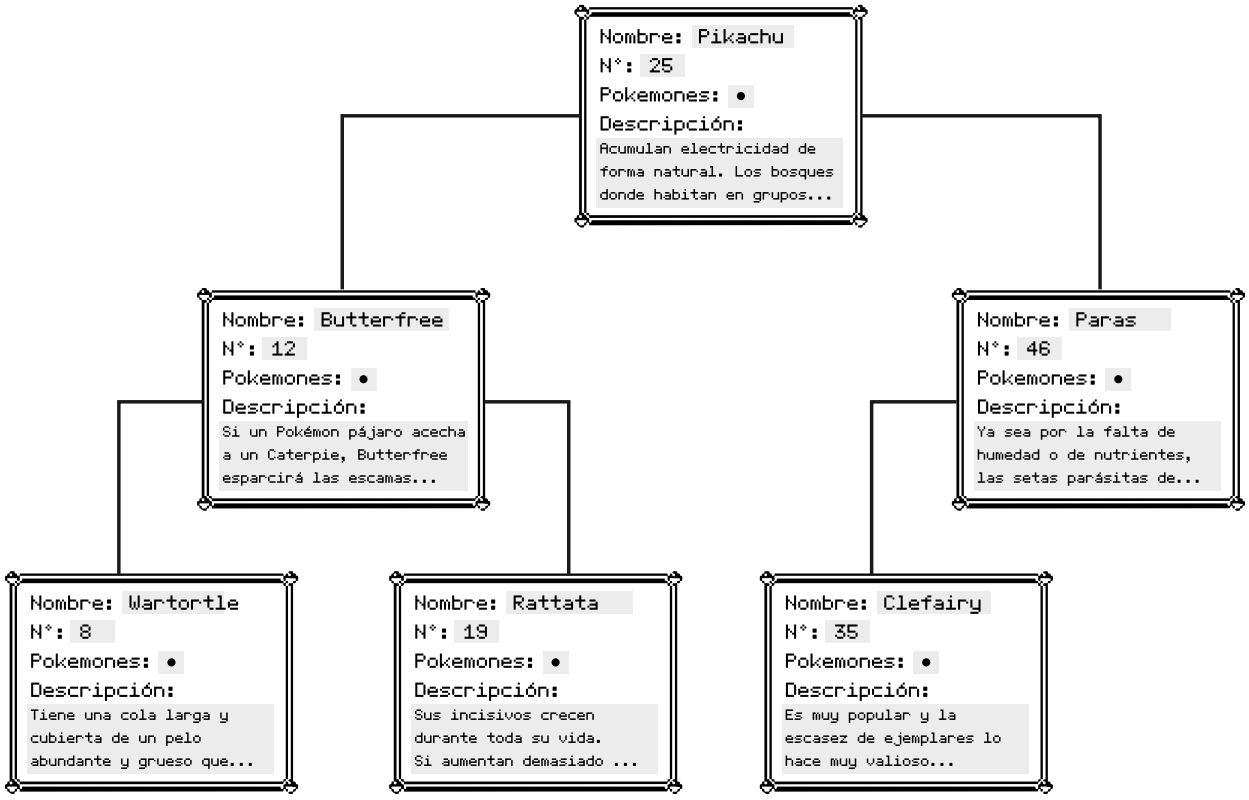
3. Enunciado

Se pide crear el TDA Pokedex y un programa que simule el uso de la misma.

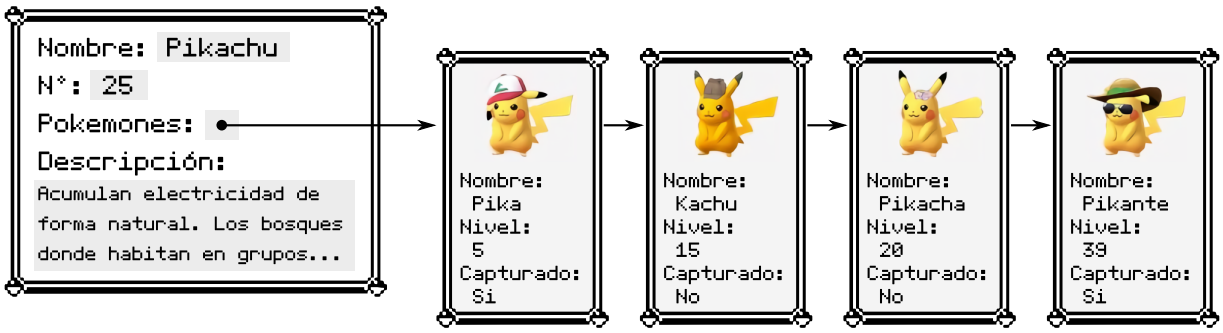
La Pokedex debe contener la información de todas las especies y de los Pokémon particulares vistos por el entrenador dueño de dicho aparato.

Para esto se solicita que se haga uso de las estructuras ABB y Lista implementados con anterioridad (puede agregar primitivas o modificar las existentes, justificando en el readme sin excepción).

Las especies de Pokémon vistas por el entrenador deberán ser almacenadas en un árbol (ordenadas por número de Pokémon) de la siguiente manera:



A su vez, por cada especie Pokémon, se almacena un listado de todos los Pokémon avistados y capturados de dicha especia de la siguiente manera:



La información en la Pokedex podrá ser actualizada de dos formas diferentes: mediante el avistamiento de Pokémon y mediante la evolución de Pokémon.

Cuando se avista un Pokémon, se debe verificar si la especie a la que pertenece está o no almacenada en la Pokedex. Si no existe, debe agregarse la información de la especie para luego almacenar los datos particulares del Pokémon en cuestión. Si la especie del Pokémon ya estaba presente, solamente deben agregarse los datos del Pokémon particular al listado de Pokémon de la especie.

Al evolucionar un Pokémon particular (cambio de especie), dicho Pokémon es removido de el listado de Pokémon conocidos en su especie y deben pasar al listado los Pokémon conocidos de la nueva especie. Por ejemplo, si el entrenador tiene un Pokémon de la especie Pikachu identificado como Pikante, al evolucionar, Pikante deja de ser parte de la lista de Pikachus y pasa a ser parte de la lista de Raichus.

La información de la especie Pikachu no deberá desaparecer de la Pokedex por más de que su lista de Pokémon esté vacía (esto puede ocurrir al evolucionar un Pokémon único en su especie).

Como información extra, la Pokedex almacena en una pila los Pokémon mas recientemente capturados y en una cola los Pokémon mas recientemente avistados. Esta información funciona a modo de historial y es limpiado cada vez que se lo consulta. Por ejemplo si se quieren listar los últimos Pokémon avistados, se mostrará el listado pedido, pero al mismo tiempo estos Pokémon seran removidos del historial.

Por último, existe la posibilidad de prender y apagar la Pokedex. Al apagarse, la información almacenada en memoria se guarda a un archivo (pokedex.txt) y al prenderse la información es recuperada de ese mismo archivo.

Las funcionalidades que se deben implementar son:

```c
/*
 * Crea la Pokedex, reservando la memoria necesaria para la misma.
 * Devolverá una referencia a la pokedex creada o NULL si no se pudo crear.
 */
pokedex_t* pokedex_crear(char entrenador[MAX_NOMBRE]);

/*
 * Función que dado un archivo, deberá cargar en la Pokedex a los
 * Pokémon que fueron tanto capturados como vistos. No verifica si
 * dos Pokémon de la misma especie tienen nombre repetido.
 *
 * Agrega a la pila correspondiente los Pokémon capturados.
 * Agrega a la cola correspondiente los Pokémon avistados.
 *
 * Devuelve 0 si tuvo éxito o -1 si se encuentra algún error durante el proceso.
 */
int pokedex_avistar(pokedex_t* pokedex, char ruta_archivo[MAX_RUTA]);

/*
 * Función que dado un archivo, deberá cargar en la Pokedex a los
 * Pokémon que evolucionaron.
 *
 * La evolución de Pokémon no afecta a la pila de capturados ni a la
 * cola de vistos. Un Pokémon que no está marcado como capturado no
 * puede evolucionar.
 *
 * Devuelve 0 si tuvo éxito o -1 si se encuentra algún error durante
 * el proceso (por ejemplo si no existe la especie, el Pokémon
 * específico o si el Pokémon no fue capturado).
 */
int pokedex_evolucionar(pokedex_t* pokedex, char ruta_archivo[MAX_RUTA]);

/*
 * Procedimiento que muestra los últimos Pokémon que fueron
```

```
 * capturados. Dichos Pokémon deberán ser listados del mas reciente
 * al menos reciente.
 *
 * Luego de invocar esta función, la pila de ultimos capturados queda
 * vacía.
 */
void pokedex_ultimos_capturados(pokedex_t* pokedex);

/*
 * Procedimiento que muestra los últimos Pokémon que fueron
 * vistos. Dichos Pokémon deberán ser listados del mas antiguo al
 * mas reciente.
 *
 * Luego de invocar esta función, la cola de últimos vistos queda
 * vacía.
 */
void pokedex_ultimos_vistos(pokedex_t* pokedex);

/*
 * Muestra la información del Pokémon recibido por parametros (especie
 * y nombre).  En caso de no existir el Pokémon o la especie, se
 * deberá mostrar un mensaje informando que es una especie o Pokémon
 * desconocidos..
 *
 * En caso de recibir un nombre vacío, se deben listar a todos los
 * Pokémon de esa especie.
 *
 */
void pokedex_informacion(pokedex_t* pokedex, int numero_pokemon, char nombre_pokemon[MAX_NOMBRE]);

/*
 * Destruye la estructura de la Pokedex, liberando la memoria que fue
 * reservada para la misma.
 */
void pokedex_destruir(pokedex_t* pokedex);

/*
 * Guarda la información de la pokedex a archivo pokedex.txt. La
 * información debe ser guardada en el formato especificado en el
 * enunciado del TP y adicionalmente deben guardarse de tal forma que
 * al prender la pokedex, la forma de árbol de especies tenga la misma
 * forma que el árbol original.
 *
 * Devuelve 0 en caso de éxito o -1 si hubo algún error.
 */
int pokedex_apagar(pokedex_t* pokedex);

/*
 * Carga la información de la pokedex del archivo pokedex.txt.
 *
 * No se piden validaciones sobre el formato del archivo ya que
 * suponemos que fueron guardados correctamente mediante la función
 * pokedex_apagar.
 *
 * Devuelve la pokedex creada desde el archivo o NULL en caso de error.
 */
pokedex_t* pokedex_prender();
```

A su vez, se cuenta con los siguientes estructuras a utilizar en el presente trabajo:

```
#define MAX_NOMBRE 100
#define MAX_RUTA 100
#define MAX_DESCRIPCION 100

typedef struct especie_pokemon {
        int numero;
        char nombre[MAX_NOMBRE];
        char descripcion[MAX_DESCRIPCION];
        lista_t* pokemones;
} especie_pokemon_t;

typedef struct particular_pokemon {
        char nombre[MAX_NOMBRE];
        int nivel;
        bool capturado;
} particular_pokemon_t;

```

```c
typedef struct pokedex {
        char nombre_entrenador[MAX_NOMBRE];
        lista_t* ultimos_capturados;
        lista_t* ultimos_vistos;
        abb_t* pokemones;
} pokedex_t;
```

Se deberán manejar 3 archivos de texto:

pokedex.txt: Contendrá toda la información de la Pokedex. La primer línea corresponderá al nombre del entrenador, luego, cada línea comenzará con una E si contiene información de una especia o con una P si contiene información de un Pokémon particular.

Las líneas de especie están conformadas por los siguientes campos:

```
E;numero_pokemon;nombre_especie;descripción_especie
```

Las líneas de cada Pokémon particular están conformadas por los siguientes campos:

```
P;nombre_pokemon;nivel;capturado(S/N)
```

Seguido a una especie se encontrarán todos los Pokémon particulares de dicha especie.

Ejemplo:

```
E;25;Pikachu;Pichachu es un pokemon...
P;Pika;13;N
P;Kachu;18;S
...
```

avistamientos.txt: Contendrá toda la información de los Pokémon avistados desde la última actualización.

Las líneas de especie están conformadas por los siguientes campos:

```
numero_pokemon;nombre_especie;descripción_especie;nombre_pokemon;nivel;capturado(S/N)
```

evoluciones.txt: Contendrá toda la información de los Pokémon que evolucionaron desde la última actualización.

Las líneas de especie están conformadas por los siguientes campos:

```
numero_pokemon;nombre_pokemon;numero_pokemon_evolucion;nombre_especie_pokemon;descripcion
```

4. Interacción con el usuario

El programa debe comunicarse con el usuario (y viceversa) a través de un menú con las siguientes opciones:

- **Iniciar Pokedex (tecla I)**: Da inicio a la Pokedex. Deberá leer la infomación del archivo **pokedex.txt**.
- **Guardar Pokedex (tecla G)**: Guarda la Pokedex. Deberá guardar la infomación en el archivo **pokedex.txt**.
- **Salir del programa (tecla S)**: Finaliza la ejecución del programa sin guardar la información.
- **Ayuda (tecla H)**: Informa cuales son los comandos válidos a utilizar en cada momento.
- **Avistar Pokémon (tecla A)**: Actualizará la Pokedex incorporando a ella los Pokémon avistados. La información de los nuevos avistamientos se encuenta en en archivo **avistamientos.txt**.
- **Evolucionar Pokémon (tecla E)**: Actualizará la Pokedex evolucionando los Pokémon que así lo hayan hecho. La información de las nuevas evoluciones se encuenta en en archivo **evoluciones.txt**.
- **Capturas recientes (tecla C)**: Muestra los últimos Pokémon capturados.
- **Vistas recientes (tecla V)**: Muestra los últimos Pokémon vistos.
- **Información especie (tecla M)**: Muestra la información de la especie.
- **Información Pokémon (tecla P)**: Muestra la información de un Pokémon de una determinada especie.

Si la letra ingresada no corresponde a un comando válido, deberá volver a pedirse hasta que sea válida, volviendo a informar cuales son los comandos disponibles en ese momento.

Los comandos válidos en determinado momento dependen del estado de la Pokedex. Por ejemplo, si la Pokedex no está iniciada, los únicos comandos válidos son Iniciar Pokedex, Salir del programa y Ayuda.

5. Resultado esperado

Se espera que el trabajo creado sea compilado sin errores con la siguiente línea:

```
gcc *.c -Wall -Werror -Wconversion -std=c99 -o pokedex
```

Y, ser corrido con **Valgrind**:

```
valgrind --leak-check=full --track-origins=yes --show-reachable=yes ./pokedex
```

Se brindarán los siguientes archivos de ejemplo como punto de partida. Se recomienda crear y hacer un backup de los suyos para probar el trabajo práctico:

pokedex.txt

```
Alumno Sin Nombre
E;Pikachu;25;Pikachu es uno de los pokémon que tiene la apariencia de un pequeño ratón.
P;pika;5;S
P;kachu;15;N
P;Pikacha;20;N
P;Pikante;39;S
E;Charmander;4;La llama que tiene en la punta de la cola arde según sus sentimientos.
P;Char;23;N
P;Mander;5;S
P;Agumon;13;N
```

avistamientos.txt

```
25;Pikachu;Pikachu es un raton...;Kachu;15;N
46;Paras;Paras es un bichito...;Parrrras;12;S
25;Pikachu;Pikachu es un raton...;Pikacha;20;N
12;Butterfree;Butterfree es una mariposa...;Danaus;23;S
19;Rattata;Rattata es otra rata...;Colagusano;30;N
35;Clefairy;Clefairy es muy rosa...;Cleffinha;13;S
25;Pikachu;Pikachu es un raton...;Pikante;39;S
8;Wartortle;Wartortle es la tortuga...;Trotu;10;N
```

evoluciones.txt

```
25;Kachu;26;Raichu;Los Raichus también son ratones... ¡Cuantos ratones!
8;Trotu;9;Blastoise;Los Blastoise tienen dos cañones de agua en su espalda...
19;Colagusano;20;Raticate;Los Rattata viven en las praderas...
35;Cleffinha;36;Clefable;Los Clefable tambien son rosados...
```

6. Entrega

La entrega deberá contar con todos los archivos necesarios para compilar y ejecutar correctamente el TP. Se recomienda que la misma conste de un archivo .c en donde transcurra el flujo principal del programa, y otro que contenga las implementaciones de la biblioteca brindada por la cátedra.

Dichos archivos deberán formar parte de un único archivo **.zip** el cual será entregado a través de la plataforma de corrección automática **Kwyjibo**.

El archivo comprimido deberá contar además con:

- Un **README.txt** que contenga:
  - Una breve introducción sobre el funcionamiento del trabajo presentado.
  - Una explicación de como compilarlo (línea de compilación) y como ejecutarlo (línea de ejecución).
- Archivos **pokedex.txt**, ***avistamientos.txt*** y **evoluciones.txt** de ejemplos para probar el trabajo.
- El enunciado.

7. Referencias

- https://www.pokemon.com/el/pokedex/
- https://pokemon.fandom.com/es/wiki/Estadisticas
- http://www.juegosdb.com/guia-del-entrenador-pokemon-caracteristicas-de-un-pokemon-nintendo-ds-nintendo-3ds/
- https://es.wikipedia.org/wiki/Pokémon