

Laboratorio 1: Sistemas Operativos

Profesor: Viktor Tapia

Ayudante de cátedra: Martín Rodríguez y Tomás Rebolledo

Ayudante de Tarea: Javiera Cárdenas y Nicolás Toro

13 de Marzo 2024

1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso y ejecución de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

2 Contexto

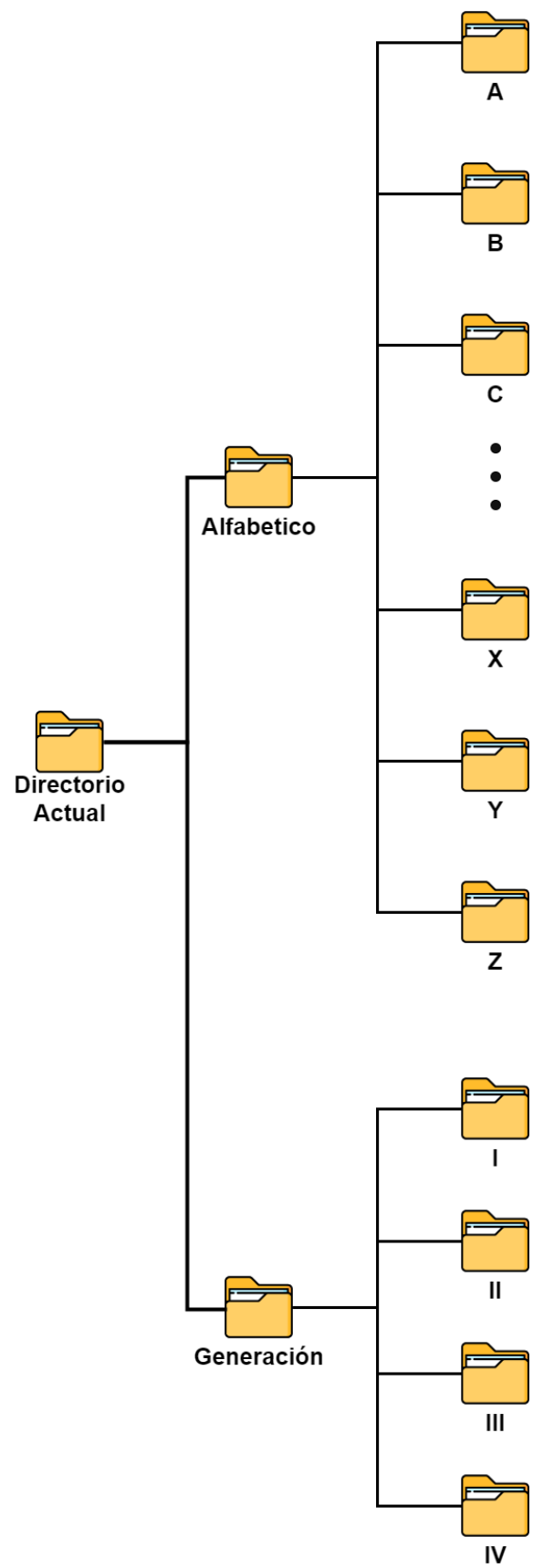
El año pasado, Ash Ketchum de pueblo paleta, se convirtió en el flamante ganador de la liga Pokémon. Luego de obtener el trofeo, decidió enviar a todos sus Pokémon al rancho del profesor Oak para que pudieran disfrutar de un merecido descanso. Sin embargo, Ash tenía tantos Pokémon que el profesor decidió asignarlos a sectores específicos según las condiciones de cada región y llevar un registro detallado de ellos. El profesor Oak solicita tu ayuda para completar esta importante tarea.

3 Tarea

Se requiere desarrollar un programa que organice cada Pokémon, representado por un archivo .png, en diferentes directorios según ciertos parámetros, ya sea su nombre o número de Pokédex.

Los directorios deben establecerse siguiendo un criterio de clasificación de los archivos. En primer lugar, se organizarán alfabéticamente en un directorio para facilitar el registro, asignando cada archivo a su directorio correspondiente según la letra inicial de su nombre. Posteriormente, se crearán en otro directorio copias de estos archivos, clasificándolos según la generación a la que pertenezca cada Pokémon, asignando cada archivo al directorio adecuado basándose en el número asociado a cada archivo.

Para una comprensión más clara, se adjunta una imagen que ilustra la estructura de los directorios en la próxima página.



3.1 Nombre archivos

Recibirá múltiples archivos .png que presentan sus nombres en el siguiente formato:

NombrePokemon_Número.png

3.2 Registro

Por último, se solicita crear un archivo *RegistroPokemon.txt*, que incluya la cantidad de Pokémon que comienzan con cada letra del abecedario y la cantidad de Pokémon que hay en cada generación, siguiendo el siguiente formato:

Generación
I – CantidadI
II – CantidadII
III – CantidadIII
IV – CantidadIV
Alfabético
A – CantidadA
B – CantidadB
C – CantidadC
○
○
○
X – CantidadX
Y – CantidadY
Z – CantidadZ

3.3 Consideraciones

- Se dará un archivo .zip con todos los archivos (imágenes) de los pokemones a utilizar.
- ***HINT*** Las generaciones van desde los números 1-151 para la primera, 152-251 para la segunda, 252-386 para la tercera y 387-493 para la cuarta.

4 Presentación Aleatoria

Para cada tarea, se seleccionarán grupos al azar para presentar su tarea frente a ayudantes y eventualmente profesor, recibiendo una ponderación del 80% y 20% entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente.

5 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los nombres de los archivos.
- Instrucciones generales de compilación y uso.

6 Consideraciones Generales

- Se deberá trabajar en **parejas**. Se deberá entregar en el repositorio de **Github** habilitado a mas tardar el día 3 de Abril de 2023 a las 23:59 horas. Se descontarán 10 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje C. Se asume que usted sabe programar en este lenguaje, ha tenido vivencias con el, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- Las tareas serán ejecutadas en **Linux**, cualquier tarea que no se pueda ejecutar en dicho sistema operativo, partirá de nota máxima 60.
- Las preguntas deben ser hechas por Aula a traves del foro. De esta forma los demás grupos pueden beneficiarse en base a la pregunta.
- Si no se entrega README o MAKE, o si su programa no funciona, la nota es 0 hasta la corrección.
- Se descontarán hasta 50 puntos por:
 - Mala implementación del Makefile.
 - No respetar el formato de entrega.
 - Solicitar edición de código para el correcto funcionamiento del Laboratorio.