

9 DE SEPTIEMBRE DE 2022

TAREA PROGRAMADA 1
MANIPULACIÓN DE ARCHIVOS SML/NJ

JEAN HUNT
LENGUAJES DE PROGRAMACIÓN
Ingeniería en computación




Tabla de contenido

<i>Manual de usuario</i>	1
Compilación y ejecución	2
Tutorial	3
1. Programa analizador.....	3
2. Programa creador	7
<i>Documentación general</i>	9
1. Descripción del problema	10
1.1. Creador	10
1.2. Analizador	10
2. Diseño del programa:	11
2.1. Decisiones de diseño	12
2.2. Algoritmos usados	12
3. Librerías utilizadas	13
4. Análisis de resultados	14

Ilustración 1:Compilación y ejecución.....	2
Ilustración 2:Llamada al método main.....	2
Ilustración 3:Ejecutar analizador.....	3
Ilustración 4:Ejecutar creador.....	3
Ilustración 5:Menú analizador.....	3
Ilustración 6:Top ascendente por rango.....	4
Ilustración 7:Detalles por altura.....	4
Ilustración 8:Detalles por posición.....	5
Ilustración 9: Detalles por clase.....	5
Ilustración 10:Detalles por orden.....	5
Ilustración 11:Resumen.....	6
Ilustración 12:Menu creador.....	7
Ilustración 13:Agregar espécimen.....	7
Ilustración 14:Es espécimen agregado.....	8
Ilustración 15:Limpiar indice.....	8
Ilustración 16:Algoritmos analizador.....	12
Ilustración 17:Algoritmos creador.....	13
Ilustración 19:Objetivos alcanzados.....	14

Manual de usuario

Compilación y ejecución

Este programa de manipulación de archivos funciona a través de la consola de comandos, no utiliza ninguna librería externa al lenguaje por lo que lo único que se necesita para ejecutar el programa es tener el compilador de Standard ML of New Jersey.

La guía para la instalación y ejecución del compilador se encuentra en el siguiente enlace: <http://smlnj.org/install/>

Una vez instalado el compilador, para ejecutar el programa seguiremos los siguientes pasos.

1. Abrir una terminal de comandos, ya sea integrada o de terceros.
2. Dirigirse a la ruta en donde se encuentra el archivo proyecto.
3. Compilar el programa haciendo uso del siguiente comando:

sml "nombreArchivo.sml"

en este caso:

sml analizador.sml ó sml creador.sml

Nota: Puede iniciar el programa desde cualquier directorio, pero tendrá que ingresar la ruta completa del archivo, por ejemplo:

sml C://escritorio/programa/analizador.sm

4. Llamar a la función main

C:\Users\jeanp\Desktop\TEC\[2022][2][Semestre]\Lenguajes\Tareas\[T1][JeanHunt]\T1_lenguajes_SII_2022>sml analizador.sml

Ilustración 1: Compilación y ejecución

```
- main();
Ingrese la ruta del archivo: |
```

Ilustración 2: Llamada al método main

Todos los datos que sean solicitados deberán ser ingresados mediante la consola de comandos, por ejemplo, selección mediante índices numéricos o entrada por teclado de algunos valores.

Tutorial

Es sistema está conformado de dos programas, un creador y un analizador. Para ejecutar uno de ellos se deberá ingresar el comando respectivo en consola y llamar al método main() como se muestra en la siguiente imagen:

```
C:\Users\jeanp\Desktop\TEC\[2022][2][Semestre]\Lenguajes\Tareas\[T1][JeanHunt]\T1_lenguajes_SII_2022>sm1 analizador.sm1
```

Ilustración 3:Ejecutar analizador

```
C:\Users\jeanp\Desktop\TEC\[2022][2][Semestre]\Lenguajes\Tareas\[T1][JeanHunt]\T1_lenguajes_SII_2022>sm1 creador.sm1
```

Ilustración 4:Ejecutar creador

1. Programa analizador

Una vez ejecutada la función main() el programa solicitará la ruta del archivo que desea analizar, este archivo deberá tener el siguiente formato en cada línea:

RANKING AVISTAMIENTOS,CLASE,ORDEN,ESPECIE,ALTURA_LARGO,PESO

```
- main();
Ingrese la ruta del archivo: FaunaCostarricense.csv

1.Mostrar top ascendente por rango
2.Mostrar detalles animales por altura
3.Mostrar detalles especies
4.Mostrar especies por clase
5.Cantidad de especies por orden
6.Resumen
7.Salir

Escoja una opcion: |
```

Ilustración 5:Menú analizador

Si el archivo existe, el analizador extraerá los datos y mostrará un menú al usuario como se muestra en la anterior imagen.

1.1. Mostrar top ascendente por rango

Al seleccionar esta opción se solicitará un valor de inicio y uno de final. El programa mostrará los detalles de las especies dentro de ese rango ordenados de manera ascendente tal y como se muestra en la siguiente imagen:

Escoja una opcion: 1

Inicio: 4

Final: 10

RANKING	AVISTAMIENTOS	CLASE	ORDEN	ESPECIE	ALTURA_LARGO	PESO
5		Reptiles	Serpientes	Terciopelo	3.5	30
6		Mamifero	Carnivoro	Comadreja	1.3	25
7		Aves	Rapaces	Buho Blanco	0.75	10
8		Reptiles	Cocodrilos	Caiman	3.3	45
9		Aves	Lapas	Lapa Verde	1.1	17
10		Reptiles	Serpientes	Boa	5.1	55

Ilustración 6: Top ascendente por rango

1.2. Mostrar detalles animales por altura

Al seleccionar esta opción se solicitará un valor de altura. El programa mostrará los detalles de todas aquellas especies cuya altura o lago supere el valor indicado y se ordenaran de manera descendente tal y como se muestra en la siguiente imagen:

Escoja una opcion: 2

Altura a filtrar: 1.3

RANKING	AVISTAMIENTOS	CLASE	ORDEN	ESPECIE	ALTURA_LARGO	PESO
10		Reptiles	Serpientes	Boa	5.1	55
5		Reptiles	Serpientes	Terciopelo	3.5	30
8		Reptiles	Cocodrilos	Caiman	3.3	45
13		Mamifero	Carnivoro	Zorro Gris	1.7	35
1		Aves	Rapaces	Zopilote	1.5	23

Ilustración 7: Detalles por altura

1.3. Mostrar detalles especies por posición

Al seleccionar esta opción se solicitará una posición en el ranking. El programa mostrará los detalles de la especie que se encuentre en la posición con el valor indicado:

Posición del ranking: 8

RANKING	AVISTAMIENTOS	CLASE	ORDEN	ESPECIE	ALTURA_LARGO	PESO
8		Reptiles	Cocodrilos	Caiman	3.3	45

Ilustración 8: Detalles por posición

1.4. Mostrar especies por clase

Al seleccionar esta opción se solicitará un valor de clase. El programa mostrará los detalles de todas aquellas especies que pertenezcan a la clase indicada tal y como se muestra en la siguiente imagen:

Escoja una opción: 4

Clase a filtrar: mamifero

RANKING	AVISTAMIENTOS	CLASE	ORDEN	ESPECIE	ALTURA_LARGO	PESO
2		Mamifero	Carnivoro	Mapache	0.8	27
3		Mamifero	Murcielagos	Murcielago pescador	0.5	1
4		Mamifero	Murcielagos	Murcielago vespertino	0.3	0.75
6		Mamifero	Carnivoro	Comadreja	1.3	25
11		Mamifero	Carnivoro	Gato Silvestre	1	20
12		Mamifero	Roedores	Guatuzá	0.7	11
13		Mamifero	Carnivoro	Zorro Gris	1.7	35

Ilustración 9: Detalles por clase

1.5. Cantidad de especies por orden

Al seleccionar esta opción se solicitará un valor de orden. El programa mostrará los detalles de todas aquellas especies que pertenezcan al orden indicado tal y como se muestra en la siguiente imagen:

Escoja una opción: 5

Orden a filtrar: carnivoro

RANKING	AVISTAMIENTOS	CLASE	ORDEN	ESPECIE	ALTURA_LARGO	PESO
2		Mamifero	Carnivoro	Mapache	0.8	27
6		Mamifero	Carnivoro	Comadreja	1.3	25
11		Mamifero	Carnivoro	Gato Silvestre	1	20
13		Mamifero	Carnivoro	Zorro Gris	1.7	35

Ilustración 10: Detalles por orden

1.6. Resumen

Al seleccionar esta opción el analizador generará un resumen de la información almacenada en el índice y mostrará:

- Cantidad de especies distintas
- La especie con el nombre más largo
- Cantidad de especies pequeñas (0 a 2.5)
- Cantidad de especies medianas (2.6 a 5)
- Cantidad de especies grandes (mayores a 5)
- Cantidad de especies pro cada clase

Dicha información la desplegará en consola de la siguiente manera:

```
Escoja una opcion: 6

Cantidad de especies: 15
Especie con el nombre mas largo: Murcielago vespertino
Cantidad de especies pequenas: 12
Cantidad de especies medianas: 2
Cantidad de especies grandes: 1

Cantidad de especies por clase

Clase | Cantidad |
-----|-----|
Aves | 5 |
Mamifero | 7 |
Reptiles | 3 |
```

Ilustración 11: Resumen

1.7. Salir

Terminará la ejecución del programa al seleccionar esta opción.

2. Programa creador

Una vez ejecutada la función `main()` el programa solicitara la ruta del archivo que desea crear o modificar, este archivo tendrá el siguiente formato en cada línea:

RANKING AVISTAMIENTOS,CLASE,ORDEN,ESPECIE,ALTURA_LARGO,PESO

```
- main();
Ingrese la ruta del archivo:  pruebaCreador.txt

1.Agregar especimen
2.Limpiar indice
3.Salir
Escoja una opcion: 
```

Ilustración 12:Menu creador

2.1. Agregar espécimen

Al seleccionar esta opción, se le solicitaran los datos ranking, clase, orden, especie, altura y peso del espécimen que desea agregar, una vez insertados estos datos presione enter y se creara una nueva entrada en el archivo que contiene el índice tal y como se muestra en las siguientes dos imágenes:

```
Escoja una opcion: 1

Ranking: 2
clase: Reptiles
orden: Serpientes
especie: Boa
Altura: 5.1
Peso: 55
```

Ilustración 13:Agregar espécimen

```
1  RANKING AVISTAMIENTOS,CLASE,ORDEN,ESPECIE,ALTURA_LARGO,PESO
2  2,Reptiles,Serpientes,Boa,5.1,55
```

Ilustración 14:Espécimen agregado

2.2. Limpiar índice

Al seleccionar esta opción se eliminarán todas las entradas del archivo índice dejando únicamente los títulos de las columnas. Al finalizar la ejecución de este método este será el archivo resultado:

```
1  RANKING AVISTAMIENTOS,CLASE,ORDEN,ESPECIE,ALTURA_LARGO,PESO
```

Ilustración 15:Limpiar indice

2.3. Salir

Terminará la ejecución del programa al seleccionar esta opción.

Documentación general

1. Descripción del problema

Se deberán implementar dos programas en SML/NJ que tendrán como funcionalidad lo siguiente:

1.1. Creador

Tiene como finalidad la administración del índice por medio de dos funciones:

- a) Agregar un nuevo espécimen en el archivo del índice. Solicitará al usuario la posición, la clase, el orden, la especie, la altura y el peso. Se incluye al final del archivo una línea con los valores separados por coma (,).
- b) Limpiar el Índice: requiere que se borren todos los registros (especies) del Índice para empezar la carga “desde cero”. Se limpia el archivo.

1.2. Analizador

Analiza un documento de texto sobre la fauna en Costa Rica, y da cierta información sobre su contenido. Al iniciar el programa, debe pedir al usuario que introduzca la ruta del archivo a analizar (por ejemplo /tmp/Fauna.csv) y luego debe mostrar un menú con las siguientes opciones;

- a) Mostrar top (ascendente) dentro de un rango indicado por el usuario: Muestra el ranking de avistamiento (este es el dato de ordenamiento del top), clase, orden, especie, altura/largo y peso. La información debe salir titulada y tabulada. El usuario indica una posición máxima y una mínima.
- b) Mostrar animales con altura o largo mayor al valor indicado por el usuario (ordenado descendente): Muestra el ranking de avistamiento, clase, orden, especie,

altura-largo (este es el dato de ordenamiento del top) y peso. La información debe salir titulada y tabulada. El usuario indica un valor inferior.

- c) Detalles de las especies por ranking: El usuario debe ingresar la posición de la especie en el ranking, y se muestra toda la información disponible para todas las especies en esa posición (ranking de avistamiento, clase, orden, especie, altura-largo y peso). La información debe salir titulada y tabulada.
- d) Especies por Clase: retorna el detalle de todas las especies que tienen una clase indicada, se muestra el ranking, especie y altura.
- e) Cantidad de especies por orden: retornar cantidad de especies que tienen un orden indicado.
- f) Resumen: la cantidad especies, la especie con el nombre más largo (especie), cantidad de especies pequeñas (alturas de 0 a 2.5), cantidad de especies medianas (alturas de 2.6 a 5), cantidad de especies altas (alturas mayores a 5) y cantidad de especies por clase.

El programa debe correr al llamarse la función main().

***Nota:** La descripción anterior fue extraída de la especificación correspondiente a esta tarea programada

2. Diseño del programa:

El programa fue desarrollado en el lenguaje de programación SML/NJ. La interfaz del programa será únicamente a través de la consola de comandos y mediante entrada del teclado.

Existen 2 archivos que componen el programa:

- **Creador.sml:** Contiene todas las funcionalidades relacionadas al sistema *creador* solicitado.
- **Analizador.sml:** Contiene todas las funcionalidades relacionadas al sistema *analizador* del índice

2.1. Decisiones de diseño

Antes y durante el desarrollo del proyecto se tomaron las siguientes decisiones:

- El programa será desarrollado en lenguaje sml/nj
- Se utilizará vscode como entorno de desarrollo.
- La navegación por menus se hará mediante índices indicados en el mismo.
- El programa se desarrollará únicamente para ser ejecutado en consola.
- Únicamente se utilizaran librerías ya incluidas en el lenguaje.
- Se utilizará github como control de repositorio.

2.2. Algoritmos usados

En la siguiente imagen, se encuentra un listado con las declaraciones de los algoritmos utilizados en los programas:

Analizador.sml

```

1  fun getLine ()
2  fun splitter sep s
3  fun elimNewlines s
4  fun printRes(lista)
5  fun listOf(nil)
6  fun abrirArchivo ()
7  fun delete_ith ([], k)
8  fun topAscendenteRango(lista)
9  fun detallesAltura(lista)
10 fun especiesRanking(lista)
11 fun especiesClase(lista)
12 fun especiesOrden(lista)
13 fun eliminarEspeciesRep []
14 fun eliminarRep []
15 fun resumen (lista)
16 fun imprimirMenu (lista)
17 fun main ()

```

Ilustración 16: Algoritmos analizador

Creador.sml

```

1  fun getLine ()
2  fun splitter sep s
3  fun elimNewlines s
4  fun limpiarArchivo (ruta)
5  fun esInt s
6  fun esReal s
7  fun solicitarNum (s)
8  fun agregarEntrada (ruta)
9  fun imprimirMenu (ruta)
10 fun main ()

```

Ilustración 17: Algoritmos creador

3. Librerías utilizadas

Para el desarrollo del programa únicamente se utilizaron librerías y estructuras nativas del lenguaje sml/nj la cuales se enlistan a continuación:

- **Int:** Uso y manipulación de datos de tipo entero
- **String:** Uso y manipulación de cadenas de texto
- **List:** Manipulación de arreglos de datos.
- **Format:** Formatos para la salida de texto al momento de imprimir en consola.
- **Real:** Uso y manipulación de datos numéricos con decimales
- **TextIO:** Entrada y salida de texto

4. Análisis de resultados.

En la siguiente tabla se muestra una lista de objetivos para los cuales se evaluará si fueron cumplidos o no, en caso de no cumplirse también se indicará la razón del porqué.

Objetivo	Alcanzado	Razón
Cumplir con las funcionalidades del programa solicitadas	Si	
Obtener una alta rigidez durante la ejecución	Si	
Documentación interna del programa	Si	.
Documentación externa del programa	Si	

Ilustración 18:Objetivos alcanzados

En el siguiente enlace se puede encontrar el repositorio que contiene la tarea
https://github.com/JPHuntV/T1_lenguajes_SII_2022.git