



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
ORGANIZACIÓN DE COMPUTADORES - CCPG 1049
PROYECTO – PRIMERA EVALUACIÓN

COMBATE POKÉMON

Estructura del proyecto

Para la elaboración del proyecto se decidió atomizar cada una de las funciones para que cada función no posea un número exorbitante de líneas de código y para que sea mucho más fácil de entender al momento de leer el código. Las funciones fueron separadas en diferentes archivos y el archivo principal se llama **main.asm** y es donde se encuentra el main del programa.

Casos especiales

Debido a la estructura de la matriz de factores de ataque brindada como recurso a usar. Existe un caso en donde el ataque es nulo para ambos pokémons, dando como resultado un bucle infinito ya que la condición de salida para la batalla pokémon es que la vida de alguno de los dos pokémons llegue a cero, pero al no hacerse daño esto jamás ocurre. Para recrear este caso solo basta con efectuar una batalla entre un pokémon de tipo “normal” como uno de tipo “ghost”.

Manejo de información

En cuanto a manejo de los recursos especificados en el proyecto para usar, se tiene los siguientes:

- **Archivo de texto PokeTypes.txt:** Este archivo contiene 108 líneas y en cada una de sus líneas posee el nombre del pokémon con su respectivo tipo separado por una coma. El recurso brindado posee por defecto una tabulación al final de cada línea.

Este archivo posee cierta peculiaridad de acuerdo al sistema operativo en el cual se esté trabajando. Si se está trabajando en Windows la estructura del archivo de texto posee las características de un archivo de texto de Windows o DOS: *nombrePokemon,TipoPokemon\t\r\n*

En cambio si se trabajaba bajo un sistema operativo de distribución de Linux como Ubuntu, el archivo posee las características de un archivo de texto de Unix: *nombrePokemon,TipoPokemon\t\n [1]*

Cabe recalcar que la lectura del archivo se validó basado en la lectura de un archivo de texto con estructura de Unix.

Además los datos obtenidos del archivo, son limpiados para ser guardados posteriormente en un buffer al cual se accede mediante dos arreglos de punteros paralelos, uno que apunta a los nombres y el otro apunta a los tipos correspondientes, para facilitar el manejo de los datos.

- **Arreglo de tipos sin repetir:** Este arreglo posee los tipos de pokemones sin repetir disponibles en nuestro programa. Este arreglo posee relación con la matriz de factores ya que los ejes de dicha matriz son paralelas a este arreglo respectivamente. Se decidió embeber al arreglo en el código, en la sección .data de la función batalla como se muestra en la *Ilustración 1*

```
t1:      .ascii "normal"
t2:      .ascii "fight"
t3:      .ascii "flying"
t4:      .ascii "poison"
t5:      .ascii "ground"
t6:      .ascii "rock"
t7:      .ascii "bug"
t8:      .ascii "ghost"
t9:      .ascii "steel"
t10:     .ascii "fire"
t11:     .ascii "water"
t12:     .ascii "grass"
t13:     .ascii "electric"
t14:     .ascii "psychic"
t15:     .ascii "ice"
t16:     .ascii "dragon"
t17:     .ascii "dark"
t18:     .ascii "fairy"
types:   .word t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18
```

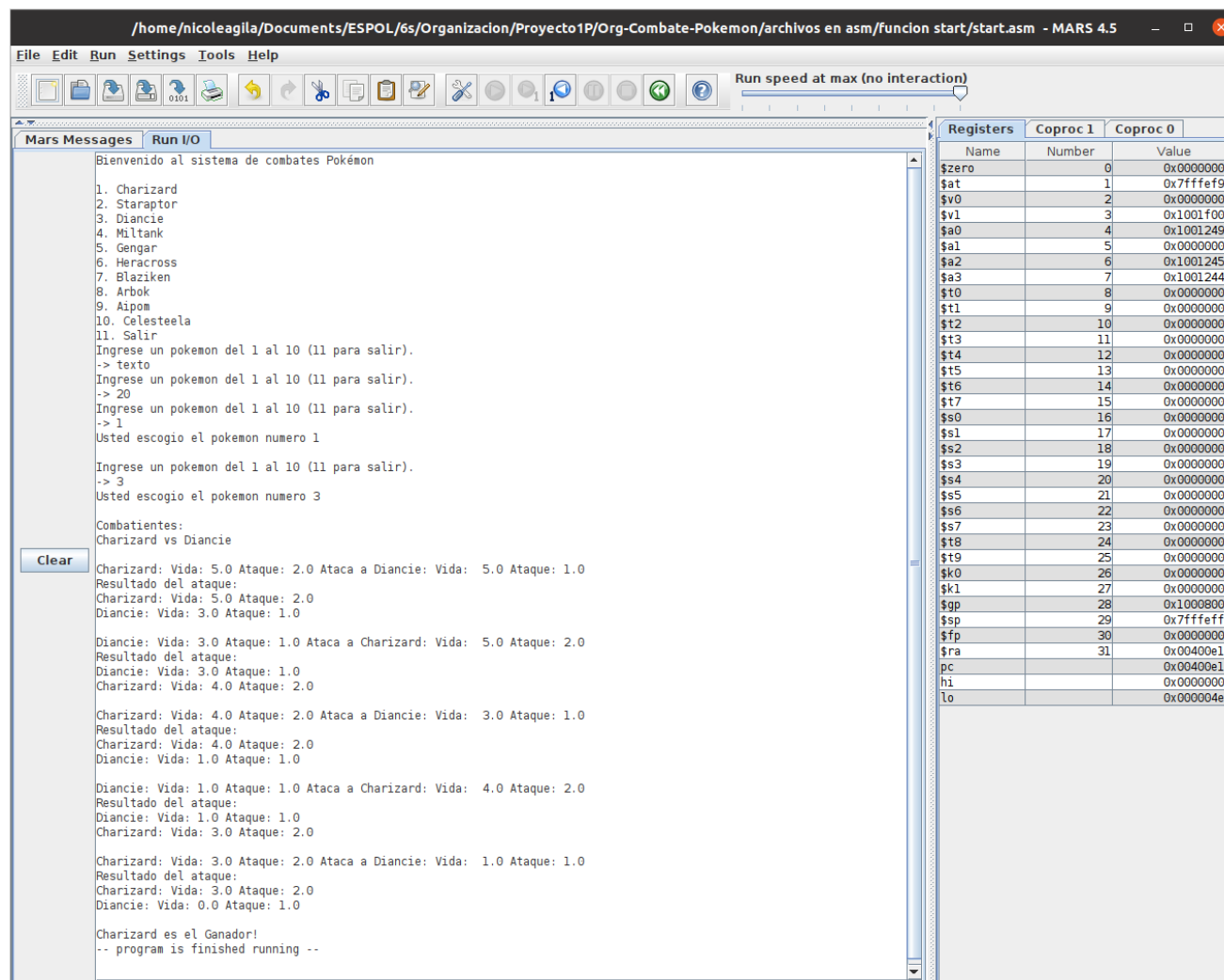
Ilustración 1, Arreglo de tipos

- **Matriz de factores de ataque:** Esta matriz de flotantes posee el factor de ataque dependiendo de cada tipo de pokémon. Cada fila corresponde al tipo pokémon que ataca y cada columna corresponde al tipo pokémon atacado. Las filas de esta matriz y sus columnas, guardan relación con el arreglo de tipos sin repetir. Para un fácil manejo de estos factores, se decidió embeber la matriz de flotantes en la sección .data de la función batalla como se muestra en la *ilustración 2* y para acceder a los datos se usa la siguiente fórmula: **$\text{direccionMemoria} = \text{direccionBase} + (\text{indiceFila} \times \text{numeroColumnas} + \text{indiceColumna}) \times \text{tamanoDato}$** [2]

```
mFactor: .float 1.0, 1.0, 1.0, 1.0, 1.0, 0.5, 1.0, 0.0, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
.float 2.0, 1.0, 0.5, 0.5, 1.0, 2.0, 0.5, 0.0, 2.0, 1.0, 1.0, 1.0, 1.0, 0.5, 2.0, 1.0, 2.0, 0.5
.float 1.0, 2.0, 1.0, 1.0, 1.0, 0.5, 2.0, 1.0, 0.5, 1.0, 1.0, 2.0, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0
.float 1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 1.0, 0.5, 0.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0
.float 1.0, 1.0, 0.0, 2.0, 1.0, 2.0, 0.5, 1.0, 2.0, 2.0, 1.0, 0.5, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0
.float 1.0, 0.5, 2.0, 1.0, 0.5, 1.0, 2.0, 1.0, 0.5, 2.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0
.float 1.0, 0.5, 0.5, 0.5, 1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 2.0, 0.5
.float 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 0.5, 1.0
.float 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 0.5, 0.5, 0.5, 1.0, 0.5, 1.0, 2.0, 1.0, 1.0, 2.0
.float 1.0, 1.0, 1.0, 1.0, 1.0, 0.5, 2.0, 1.0, 2.0, 0.5, 0.5, 2.0, 1.0, 1.0, 2.0, 0.5, 1.0, 1.0
.float 1.0, 1.0, 1.0, 1.0, 2.0, 2.0, 1.0, 1.0, 1.0, 2.0, 0.5, 0.5, 1.0, 1.0, 1.0, 0.5, 1.0, 1.0
.float 1.0, 1.0, 0.5, 0.5, 2.0, 2.0, 0.5, 1.0, 0.5, 0.5, 2.0, 0.5, 1.0, 1.0, 1.0, 0.5, 1.0, 1.0
.float 1.0, 1.0, 2.0, 1.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 0.5, 0.5, 1.0, 1.0, 0.5, 1.0, 1.0
.float 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 0.5, 1.0, 1.0, 1.0, 1.0, 0.5, 1.0, 1.0, 0.0, 1.0
.float 1.0, 1.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, 0.5, 0.5, 0.5, 2.0, 1.0, 1.0, 0.5, 2.0, 1.0, 1.0
.float 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 0.0
.float 1.0, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 0.5, 0.5
.float 1.0, 2.0, 1.0, 0.5, 1.0, 1.0, 1.0, 1.0, 0.5, 0.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0
```

Ilustración 2, Matriz de factores

Capturas de pantalla del funcionamiento



The screenshot shows the MARS 4.5 emulator window. The title bar indicates the file path: `/home/nicoleagila/Documents/ESPOL/6s/Organizacion/Proyecto1P/Org-Combate-Pokemon/archivos en asm/funcion start/start.asm - MARS 4.5`. The menu bar includes File, Edit, Run, Settings, Tools, and Help. The toolbar contains various icons for file operations and execution. The 'Run speed at max (no interaction)' slider is visible.

The 'Mars Messages' tab is active, showing the following text:

```

Bienvenido al sistema de combates Pokémon
1. Charizard
2. Staraptor
3. Diancie
4. Miltank
5. Gengar
6. Heracross
7. Blaziken
8. Arbok
9. Aipom
10. Celesteela
11. Salir
Ingrese un pokemon del 1 al 10 (11 para salir).
-> texto
Ingrese un pokemon del 1 al 10 (11 para salir).
-> 20
Ingrese un pokemon del 1 al 10 (11 para salir).
-> 1
Usted escogio el pokemon numero 1

Ingrese un pokemon del 1 al 10 (11 para salir).
-> 3
Usted escogio el pokemon numero 3

Combatientes:
Charizard vs Diancie

Charizard: Vida: 5.0 Ataque: 2.0 Ataca a Diancie: Vida: 5.0 Ataque: 1.0
Resultado del ataque:
Charizard: Vida: 5.0 Ataque: 2.0
Diancie: Vida: 3.0 Ataque: 1.0

Diancie: Vida: 3.0 Ataque: 1.0 Ataca a Charizard: Vida: 5.0 Ataque: 2.0
Resultado del ataque:
Diancie: Vida: 3.0 Ataque: 1.0
Charizard: Vida: 4.0 Ataque: 2.0

Charizard: Vida: 4.0 Ataque: 2.0 Ataca a Diancie: Vida: 3.0 Ataque: 1.0
Resultado del ataque:
Charizard: Vida: 4.0 Ataque: 2.0
Diancie: Vida: 1.0 Ataque: 1.0

Diancie: Vida: 1.0 Ataque: 1.0 Ataca a Charizard: Vida: 4.0 Ataque: 2.0
Resultado del ataque:
Diancie: Vida: 1.0 Ataque: 1.0
Charizard: Vida: 3.0 Ataque: 2.0

Charizard: Vida: 3.0 Ataque: 2.0 Ataca a Diancie: Vida: 1.0 Ataque: 1.0
Resultado del ataque:
Charizard: Vida: 3.0 Ataque: 2.0
Diancie: Vida: 0.0 Ataque: 1.0

Charizard es el Ganador!
-- program is finished running --
  
```

The 'Registers' tab is also visible, showing a table of registers and their values:

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x7ffff94
\$v0	2	0x0000000a
\$v1	3	0x1001f000
\$a0	4	0x10012492
\$a1	5	0x00000001
\$a2	6	0x10012454
\$a3	7	0x1001244c
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffcfc
\$fp	30	0x00000000
\$ra	31	0x00400e10
pc		0x00400e18
hi		0x00000000
lo		0x000004ec

Ilustración 3, Prueba de funcionamiento 1

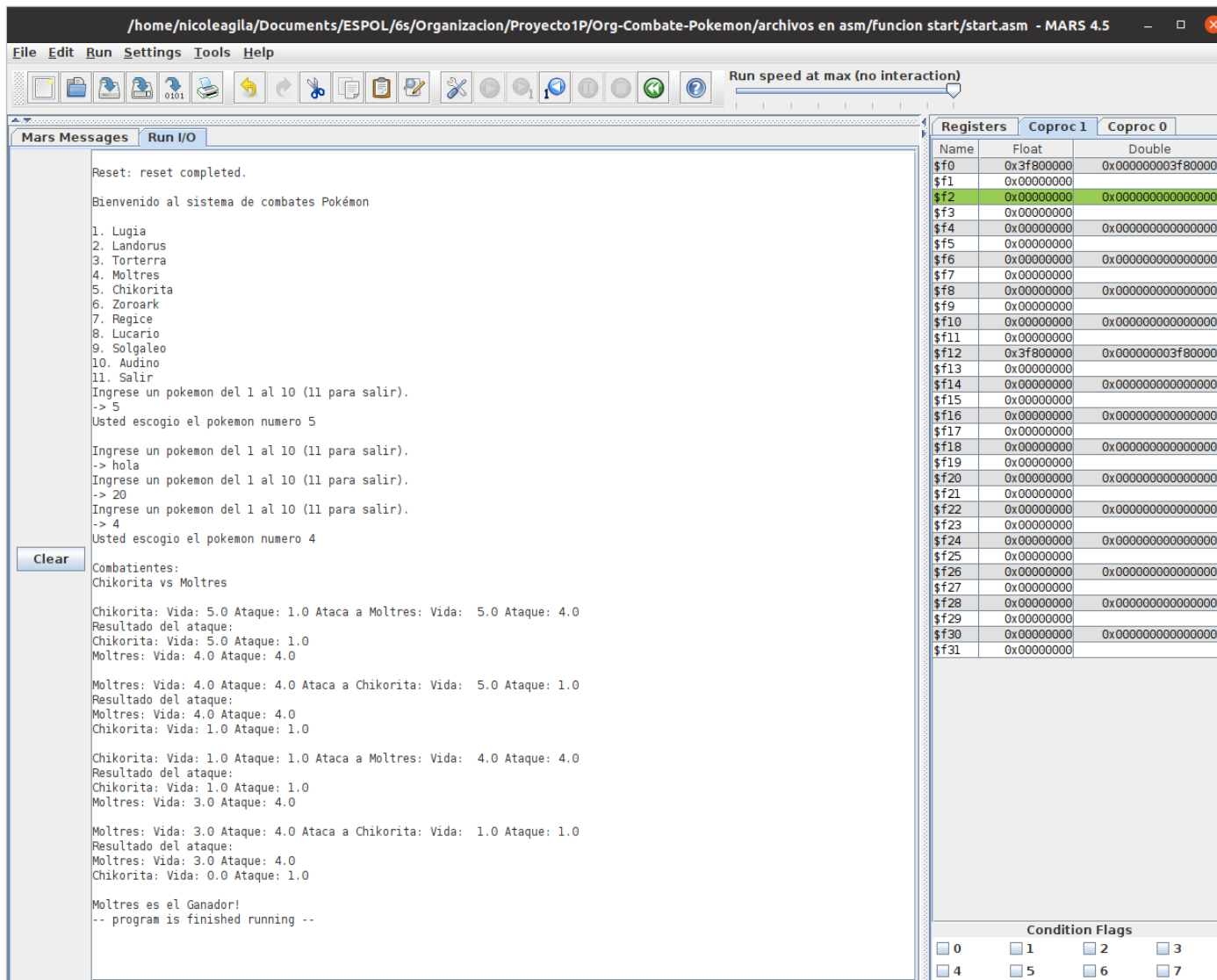


Ilustración 4, Prueba de funcionamiento 2

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
ORGANIZACIÓN DE COMPUTADORES - CCPG 1049
PROYECTO – PRIMERA EVALUACIÓN

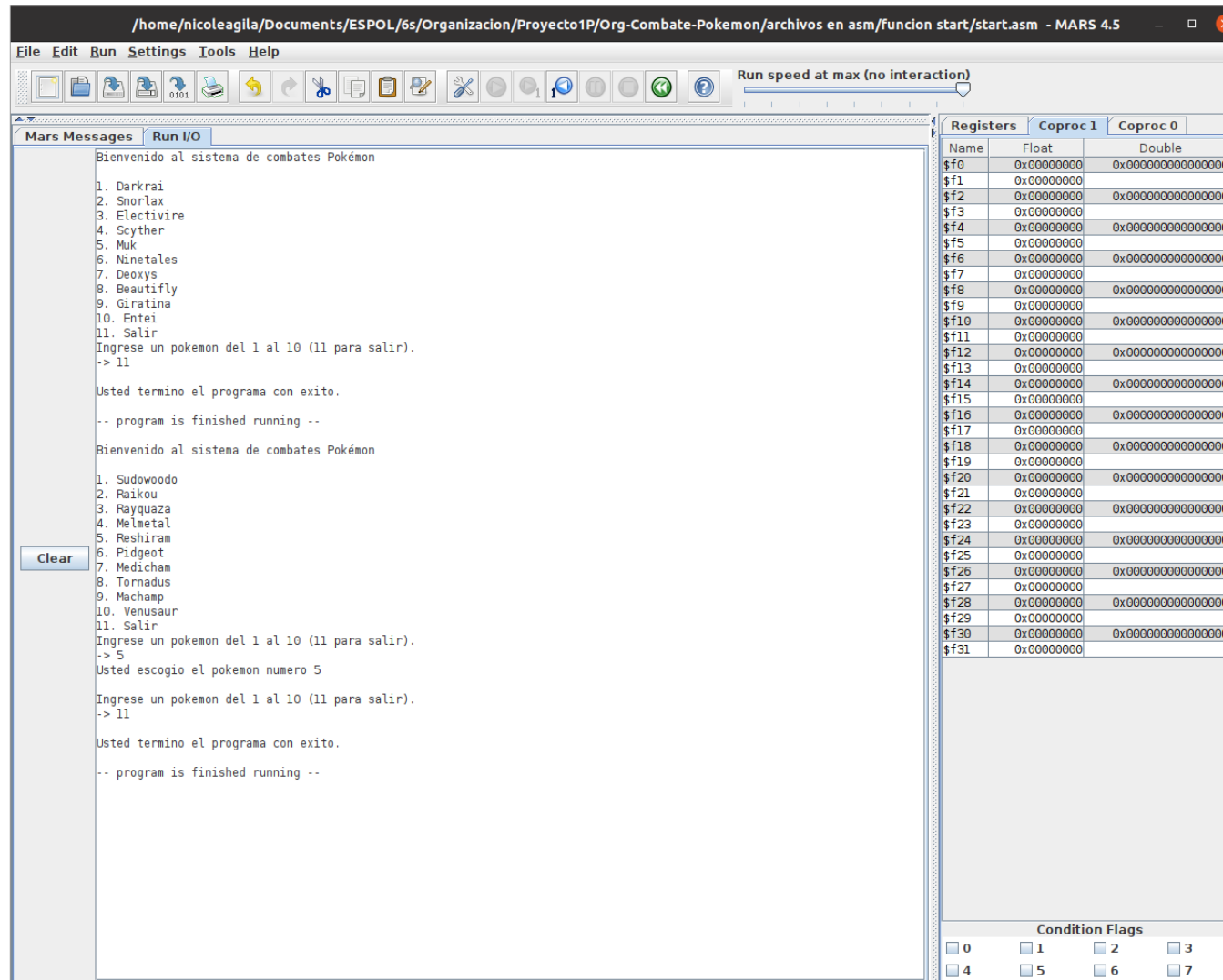


Ilustración 5, Prueba de funcionamiento 3

Referencias

- [1] Juan and J. M. C. C. dice: “Cambiar los saltos de linea rn de Windows a n de Linux, Unix, Mac OSX,” *Indalcasa*, 25-Oct-2012.
[Online]. Available: <https://www.indalcasa.com/linux/cambiar-los-saltos-de-linea-rn-de-windows-a-n-de-linux-unix-mac-osx/>.
- [2] Manzano, B. and Villamar, L., 2020. *Matrices En Ensamblador MIPS*. [ebook] Available at:
https://espolec.sharepoint.com/sites/Section_11448/Documentos%20compartidos/General/matricesProyecto.pdf [Accessed 18 July 2020].

Bibliografias

- MIPS Instruction Reference*. [Online]. Available: <http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html>.
- MIPS syscall functions available in MARS*. [Online]. Available:
<http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>.
- Assembly Language Coding Standards*. [Online]. Available:
<https://www.ee.ryerson.ca/~kclowes/stand-alone/CodingStandards/CodingStdAsm/index.html>.
- “Learn X in Y minutes Where X=MIPS Assembly,” *Learn MIPS Assembly in Y Minutes*. [Online]. Available:
<https://learnxinyminutes.com/docs/mips>