

Reporte Final: Práctica de Organización de computadores  
Pérez Pérez, Juan David.  
jperezp2@eafit.edu.co  
Universidad EAFIT

*Resumen* Este documento pretende dar a conocer todos los aspectos involucrados en la práctica llevada a cabo en la asignatura: Organización de Computadores.

*Índice de Términos*

Matriz, Saltos de caballo, posiciones de memoria, contenido, contador.

## I. INTRODUCCIÓN

En este informe muestra todo el planteamiento e instrucciones que se desarrollaron para el debido funcionamiento de este proyecto, con sus respectivas características funcionales y no funcionales.

## II. MARCO TEÓRICO

### A. Fase de revisión

En la primera parte del desarrollo de la práctica se comenzó por comprender el funcionamiento de la práctica, para poder llevar a cabo la debida lógica para desarrollar el proyecto.

Esto teniendo en cuenta que el entorno o lenguaje en el cual se hace este proyecto es desconocido, se procede a hacer una búsqueda exhaustiva sobre la sintaxis que maneja **SIMUPROC**.

### B. Fase de desarrollo

Se da continuación a la lógica propuesta de cómo realizar las acciones en la matriz 4x4, en esta lógica están propuestas varias ideas; una de ellas es inicializar una matriz con 16 posiciones en cada una de las cuales se asignara un valor de cero.

Esto se hace con el fin de poder comparar con valores numéricos cada posición para saber si esta se encuentra disponible para realizar un movimiento

o salto de caballo según las posibilidades de movimiento en la casilla en que se encuentre el programa.

Antes de realizar esta serie de movimientos el usuario tiene acceso a un menú el cual indica determinadas opciones con las que debe cumplir el programa las cuales son:

- Establecer una posición inicial
- Realizar el llenado de la matriz
- Mostrar el resultado en pantalla

Además de esto se establece un contador de movimientos el cual nos indica en que movimiento se encuentra el programa, con lo cual se sabe que recorrido realizó para llenar completamente la matriz.

### C. Fase Final

Se realizan cambios sobre el cómo debe devolverse a determinada casilla el programa si se encuentra en una posición en la cual no es posible realizar algún movimiento, también se crea una opción en el menú para salirse del programa en ejecución.

Además de un análisis o prueba de escritorio previa para verificar el correcto funcionamiento del programa y una mejora en cuanto a las líneas de código usadas.

Se realiza una verificación en cuanto a las líneas de código ya que a medida que se necesitaban añadir instrucciones en el programa, se podía producir una modificación en las posiciones de memoria; de tal forma que no se traslaparan instrucciones

### III. INDICACIONES ÚTILES

#### A. Código de algunas funciones

##### Menú

```
MSG *****
MSG *   Menu Template   *
MSG *****
MSG * 1) Posicion inicial *
MSG * 2) Llenado de la matriz*
MSG * 3) Mostrar el contenido*
MSG * 4) Salir          *
MSG *****
LDT   ;Ingreso del dato para posicion
CMP 113 ;Comparar con posición 113 (1)
JEQ 130 ;Si son iguales, saltar a 113
CMP 114 ;Comparar con posición 114 (2)
JEQ 340 ;Si son iguales, saltar a 340
CMP 115 ;Comparar con posición 115 (3)
JEQ 800 ;Si son iguales, saltar a 800
CMP 116 ;Comparar con posicion 116 (4)
JEQ 890 ;Si son iguales, saltar a 890
```

##### Movimientos posibles

```
#370
;Movimientos casilla 1

;Chequeo de HALT
MOV AX,250
CMP 240
JEQ 800

;salto a 7
MOV AX,7
CMP 215
JEQ 524
;salto a 10
MOV AX,00A
CMP 215
JEQ 536
```

##### Devolverse

```
;Acciones para devolverse

#700
;Compara a donde se devuelve
MOV AX,260
CMP 1
JEQ 370
CMP 2
JEQ 37D
CMP 3
JEQ 38D
```

##### Mostrar en Pantalla

```
;Mostrar Matriz en pantalla
#800
MSG *** MATRIZ COMPLETA ***
MSG Primera fila
MOV AX,001
EAP
MOV AX,002
EAP
MOV AX,003
EAP
MOV AX,004
EAP
```

#### B. Salida del programa

El programa muestra como salida la matriz con sus respectivos valores en cada casilla indicando el orden en el que visitó cada posición, esto se muestra de forma secuencial.

```
Fila 1
Casilla 1
Casilla 2
Casilla 3
Casilla 4
```

...

Las casillas se enumeran de izquierda a derecha

#### IV. CONCLUSIONES

Como resultado de toda la investigación, planteamiento y desarrollo realizado en este documento, se puede ver el cómo se llevó a cabo el proyecto realizado mediante un lenguaje que nos da una noción de *assembler* con lo cual vimos cómo realizar un tipo de programación mediante posiciones de memoria.

Por otro lado se ve la lógica implementada para este proyecto aprovechando la programación mediante posiciones de memoria para llevar a cabo el recorrido de la matriz.

#### REFERENCIAS

- [1] Sitio web del lenguaje  
“<https://sites.google.com/site/simuproc/instrucciones-soportadas>”.