

# **Manual de usuario CH-Máquina**

Luis Felipe Vélez García

Universidad Nacional de Colombia

Administración de sistemas informáticos

Sistemas operativos

Manizales, Caldas.

2023

## Introducción

El CH-Máquina es un software básico de simulación de sistema operativo que se ejecuta en una computadora. A través de él, se pueden ejecutar programas con la extensión .CH, los cuales deben estar escritos en un lenguaje específico llamado CH-Lenguaje, estos programas deben basarse en operaciones básicas que se encuentran en cualquier otro lenguaje de programación. La administración de la memoria es la característica principal del software, la cual se asemeja al funcionamiento de una memoria en un sistema operativo real. Además, es importante destacar que el programa también cuenta con la capacidad de validar la sintaxis de los programas escritos en CHLenguaje para evitar errores en el código y mejorar la eficiencia de los procesos. Esta validación sintáctica es una herramienta valiosa para garantizar que los programas se ejecuten sin problemas y logren los resultados deseados.

La ejecución de los programas en el CH-Máquina se realiza de manera secuencial. Esto significa que el software procesa las instrucciones de los programas en el orden en que se han escrito, lo que garantiza que las tareas se realicen en el momento y lugar adecuado. La ejecución secuencial es un proceso fundamental en la programación y se encuentra presente en la mayoría de los lenguajes de programación existentes.

#### Requisitos:

- Se recomienda utilizar Google Chrome, Mozilla Firefox o Microsoft Edge como navegadores web para una mayor compatibilidad con el software.

#### Guía de Inicio Rápido:

Esta guía rápida tiene como objetivo brindarte una introducción sencilla al CH-Máquina.

#### Paso 1: Iniciar el CH-Máquina

1. Encuentra y abre el programa CH-Máquina en la carpeta de aplicaciones o en el menú de inicio.
2. La pantalla de inicio del CH-Máquina se abrirá automáticamente.

#### Paso 3 Seleccionar el algoritmo de planificación de procesos:

- Desplegar el menú de se planificación y seleccionar el que se requiera.
- Si la elección de planificación de proceso es "Round Robin", seleccionar el valor del Quantum (el valor por defecto es de 5).

#### Paso 2: Cargar el Archivo .CH

1. Haz clic en "Cargar" para abrir la ventana de selección de archivos.
2. Selecciona el archivo .CH y haz clic en "Abrir".
3. Si hay errores de sintaxis, aparecerá una ventana emergente con la lista de errores.
4. Ingresa el valor de prioridad del archivo.

#### Paso 4: Ejecutar el Archivo Cargado

- Para ejecutar paso a paso, haz clic en "Paso a Paso". Para ejecución automática, haz clic en "Normal" (puedes pausar la ejecución en cualquier momento haciendo clic en "Pausa").
- También puedes visualizar los valores asignados a la memoria haciendo clic en "Muestra de Memoria".

## Funcionalidades

El CH-Máquina cuenta con una serie de características clave que lo hacen destacar, incluyendo:

- **Gestión Eficiente de Memoria:** Administra la memoria de manera eficiente para optimizar el rendimiento del sistema y garantizar un uso efectivo de los recursos disponibles.

- **Validación de Sintaxis del Código:** Verifica la sintaxis del código para asegurar que esté escrito de forma correcta y coherente, ayudando a los usuarios a evitar errores comunes de programación.
- **Compilación y Ejecución de Programas:** Permite a los usuarios compilar y ejecutar sus programas de manera fácil y rápida, facilitando el proceso de desarrollo y prueba de software.
- **Visualización de Resultados en Pantalla:** Muestra los resultados de la ejecución de programas directamente en pantalla. Esta función es especialmente útil para programas que realizan cálculos complejos o manipulan grandes volúmenes de datos, ya que permite a los usuarios ver la salida generada por el programa de manera clara y comprensible.

Estas funcionalidades combinadas hacen del CH-Máquina una herramienta poderosa para los desarrolladores, proporcionando un entorno de programación eficiente y efectivo.

### **Construcción de programas**

Para escribir de forma correcta el programa, se deben tener en cuenta las siguientes reglas:

- El código puede tener comentarios por líneas, los cuales se denotarán por dos backslash (//) en las dos primeras posiciones de la instrucción, de igual manera se podrán insertar líneas en blanco entre instrucciones del programa.
- Se podrán realizar operaciones entre valores enteros y reales, los resultados intermedios se manejarán como reales y el resultado final obedecerá al tipo de variable que almacena el resultado.
- El programa utilizará un acumulador para registrar los valores de los cálculos y recibirá como nombre reservado "acumulador".
- Las posiciones de memoria que almacenen datos (variables) tendrán un nombre asociado que iniciara con una letra y máximo tendrá 255 caracteres sin espacios intermedios. Estas variables deberán ser creadas antes de ser usada.
- El sistema es sensible a case, es decir diferenciara letras mayúsculas de minúsculas.
- Se podrán realizar operaciones entre valores enteros y reales, los resultados intermedios se manejarán como reales y el resultado final obedecerá al tipo de variable que almacena el resultado (realizando truncamiento si es necesario).
- Las instrucciones constarán de 2 partes; el código de la operación y el(los) operando(s) dependiendo el tipo de instrucción.

Las instrucciones que se pueden utilizar son las siguientes:

Operación	Descripción
cargue almacene	Cárguese/copie en el acumulador el valor almacenado en la variable indicada por el operando.

	<p>Guarde/copie el valor que hay en el Acumulador en la posición de memoria que corresponda a la variable indicada por el operando.</p>
nueva	<p>Crea una nueva variable cuyo nombre es el especificado en el primer operando, en el segundo operando definirá el tipo de variable (C Cadena/alfanumérico, I Entero, R Real/decimal, L lógico o booleano (1 Verdadero o 0 Falso), un tercer operando establecerá un valor de inicialización.</p> <p>A cada variable se le asignará automáticamente una posición en la memoria. Las variables deberán estar definidas antes de ser utilizadas. Las variables no inicializadas tendrán por defecto el valor cero para reales y enteros, espacio para cadenas, 0 para lógicos. El separador de decimales es el punto.</p>
lea	Lee por teclado/pantalla el valor a ser asignado a la variable indicado por la variable referida en el operando.
sume	Incrementa/sume al valor del acumulador el valor indicado por la variable señalada por el operando.
reste	Decrementa/reste del valor del acumulador el valor indicado por la variable que señala el operando.
multiplique	Multiplique el valor del acumulador por el valor indicado por la variable señalada por el operando.
divida	Divida el valor del acumulador por el valor indicado por la variable señalada por el operando. El divisor deberá ser una cantidad diferente de cero.
potencia	Eleve el acumulador a la potencia señalada por el operando (los exponentes pueden ser valores enteros, positivos o negativos)
modulo	Obtenga el módulo al dividir el valor del acumulador por el valor indicado por la variable señalada por el operando.
concatene	Genere una cadena que una la cadena dada por el operando a la cadena que hay en el acumulador (Operando alfanumérico). El contenido del acumulador deberá tratarse como cadena en caso de ser numérico.
elimine	Genere una subcadena que elimine cualquier aparición del conjunto de caracteres dados por el operando de la cadena que se encuentra en el acumulador (operando alfanumérico).
extraiga	Genere una subcadena que extraiga los primeros caracteres (dados por el operando con valor numérico) de la cadena que se encuentra en el acumulador.
Y	Produce una operación lógica Y (AND) entre el primer operando y el segundo operando que son variables lógicas y la almacena en el tercer operando.
O	Produce una operación lógica O (OR) entre el primer operando y el segundo operando que son variables lógicas y la almacena en la variable del tercer operando.
NO	Produce una operación de negación lógica para el primer operando que es una variable lógica y el resultado se almacena en la variable del segundo operando.
muestre	Presente por pantalla el valor que hay en la variable indicada por el operando, si el operando es acumulador muestre el valor del acumulador.

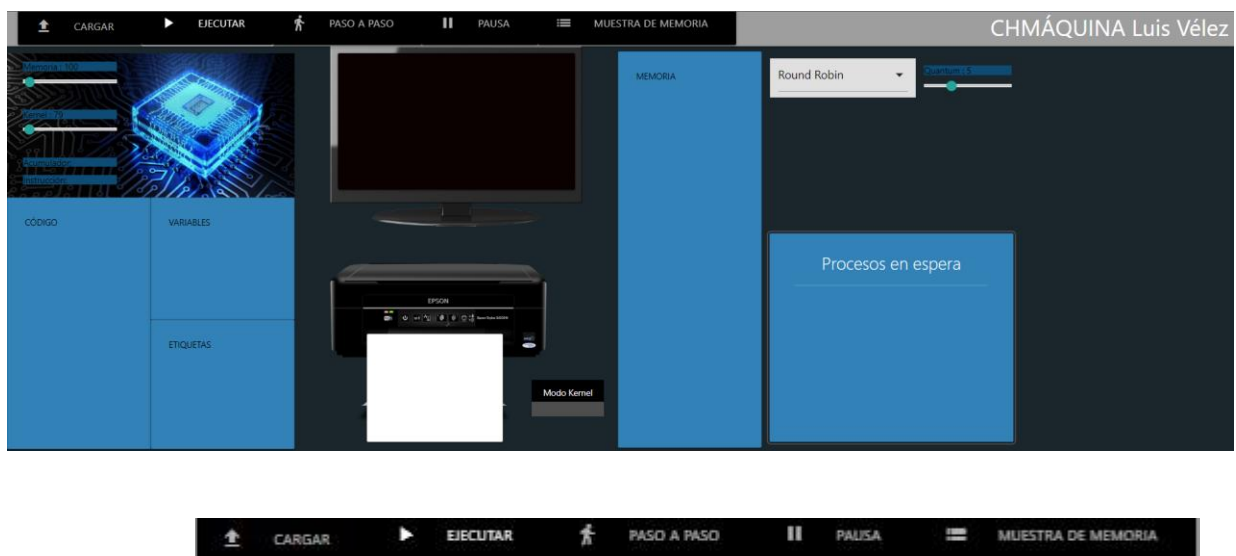
imprima	Presente por la impresora el valor que hay en la variable indicada por el operando, si el operando es acumulador muestre el valor del acumulador.
retorne	El programa termina; debe ser la última instrucción del programa y tiene opcionalmente un operando numérico entero.
*	
vaya	Salte a la instrucción que corresponde a la etiqueta indicada por el operando y siga la ejecución a partir de allí.
*	
vayasi	Si el valor del acumulador es mayor a cero salte a la instrucción que corresponde a la etiqueta indicada por el primer operando y continúe la ejecución a partir de allí. Si el valor del acumulador es menor a cero salte a la instrucción que corresponde a la etiqueta indicada por el segundo operando y continúe la ejecución a partir de allí. o Si el acumulador es cero salte a la siguiente instrucción adyacente a la instrucción vayasi y siga la ejecución a partir de allí.

---

**etiqueta** La etiqueta es un nombre que opcionalmente se le puede asignar a una instrucción en el programa para evitar trabajar con las posiciones en memoria de las instrucciones y poder utilizar un nombre simbólico independiente de su ubicación. Crea una nueva etiqueta cuyo nombre es el especificado en el primer operando y a la cual le asignará automáticamente la posición indicada en el segundo operando (esta será la posición relativa de la instrucción a la que se le asigna este nombre con respecto a la primera instrucción del programa). Las instrucciones que definen etiquetas podrán definirse en cualquier posición del programa, pero en todo caso antes de la instrucción retorne.

## Interfaz gráfica

La interfaz gráfica del CH-Máquina se compone de varios elementos que permiten al usuario interactuar con el software de manera efectiva



- Cargar: Permite cargar los programas para ser ejecutados.

- Paso a paso: Modo de ejecución que permite visualizar uno a uno los pasos que se llevan a cabo.
- Muestra de memoria: Muestra el estado de la memoria en cualquier momento que se solicite.
- Pausa: Permite detener la ejecución del programa en cualquier momento que se solicite.

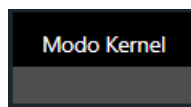


**Memoria:** Asigna el tamaño seleccionado a la memoria.

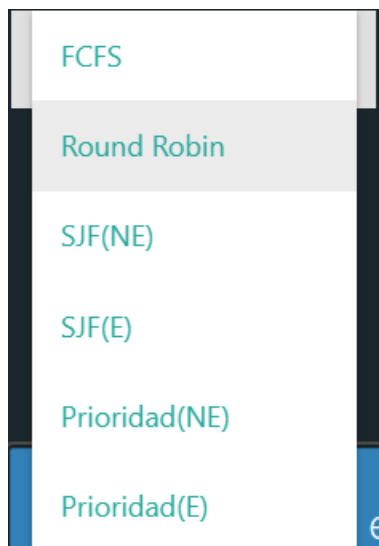
**Kernel:** Asigna el tamaño seleccionado al kernel del sistema.

**Acumulador:** Muestra el valor de la variable acumulador, durante toda la ejecución del programa.

**Instrucción:** Muestra la instrucción que se está ejecutando, durante toda la ejecución del programa.

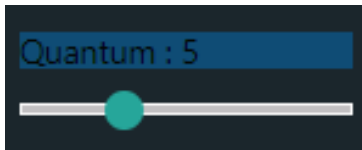


Muestra el modo en que esta (Kernel o Usuario)



## Botón de selección de

**planificación:** Muestra el modo en el que se encuentra el software.



**Quantum:** Barra para seleccionar el quantum(Round Ronin)



**Monitor:** Presenta resultados de la ejecución de programas.





**Impresora:** Imprime los resultados de la ejecución de programas.

CÓDIGO	VARIABLES
	ETIQUETAS

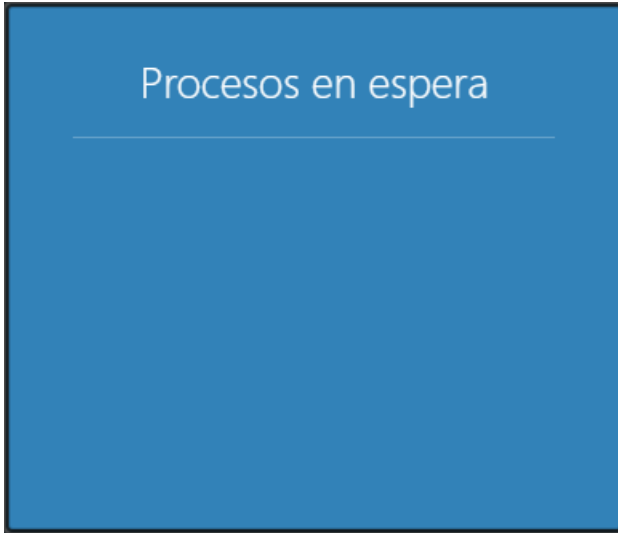
**Código:** Muestra el código del programa cargado junto a su ubicación en memoria.

**Variables:** Muestra las variables del programa cargado junto a su ubicación en memoria y el ID de documento.

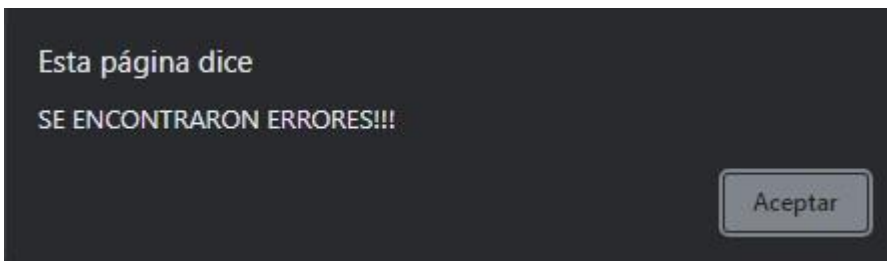
**Etiquetas:** Muestra las etiquetas del programa cargado junto a su ubicación en memoria.

MEMORIA

**Memoria:** Muestra la ocupación de las posiciones de la memoria.



Panel que muestra los diferentes procesos que se ejecutan



Ventana de errores cuando el archivo .CH tiene errores