**Manual de usuario CH-Máquina**

Alejandro Álvarez Patiño

Universidad Nacional de Colombia sede Manizales

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 CH-Lenguaje 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.

**Índice**

[Introducción 1](#_Toc129728694)

[Requisitos 4](#_Toc129728695)

[Guía de inicio rápido 4](#_Toc129728696)

[Funcionalidades 5](#_Toc129728697)

[Construcción de programas 5](#_Toc129728698)

[Interfaz gráfica 7](#_Toc129728699)

# **Requisitos**

* Navegador web: se recomienda Google Chrome, Mozilla Firefox o Microsoft Edge para una mejor compatibilidad con el software.

# **Guía de inicio rápido**

Esta guía de inicio rápido tiene como objetivo proporcionar una introducción rápida y sencilla al CH-Máquina.

Paso 1: Iniciar el CH-Máquina.

* Buscar y abrir el programa CH-Máquina en la carpeta de aplicaciones o en el menú de inicio.
* La pantalla de inicio del CH-Máquina se abrirá automáticamente.

Paso 2: Cargar el archivo .CH:

* Hacer clic en el botón “cargar” para abrir la ventana de selección de archivos.
* Seleccionar el archivo .CH y dar clic en abrir.
* Si la sintaxis es incorrecta, se abrirá una ventana emergente con la lista de errores.

Paso 3: Ejecutar el archivo cargado:

* Si se desea ejecutar línea por línea, dar clic en el botón “paso a paso”. Si se desea ejecutar de forma automática, dar clic en el botón “Normal” (El software permite pausar en cualquier momento su ejecución, dar clic en botón “pausa”).
* El software permite realizar la visualización de los valores asignados a la memoria, dar clic en el botón “Muestra de memoria”.

¡Felicidades! Ahora estás listo para utilizar el CH-Máquina para crear y ejecutar programas en un entorno.

# **Funcionalidades**

Entre las funcionalidades principales se incluyen la gestión eficiente de la memoria, la validación de la sintaxis del código, la compilación y ejecución de programas. Efectivamente, el CH-Máquina presenta los resultados de la ejecución de programas en pantalla. Esto permite al usuario visualizar la salida generada por el programa, lo que es particularmente útil para programas que realizan cálculos o manipulan datos.

## **Construcción de programas**

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

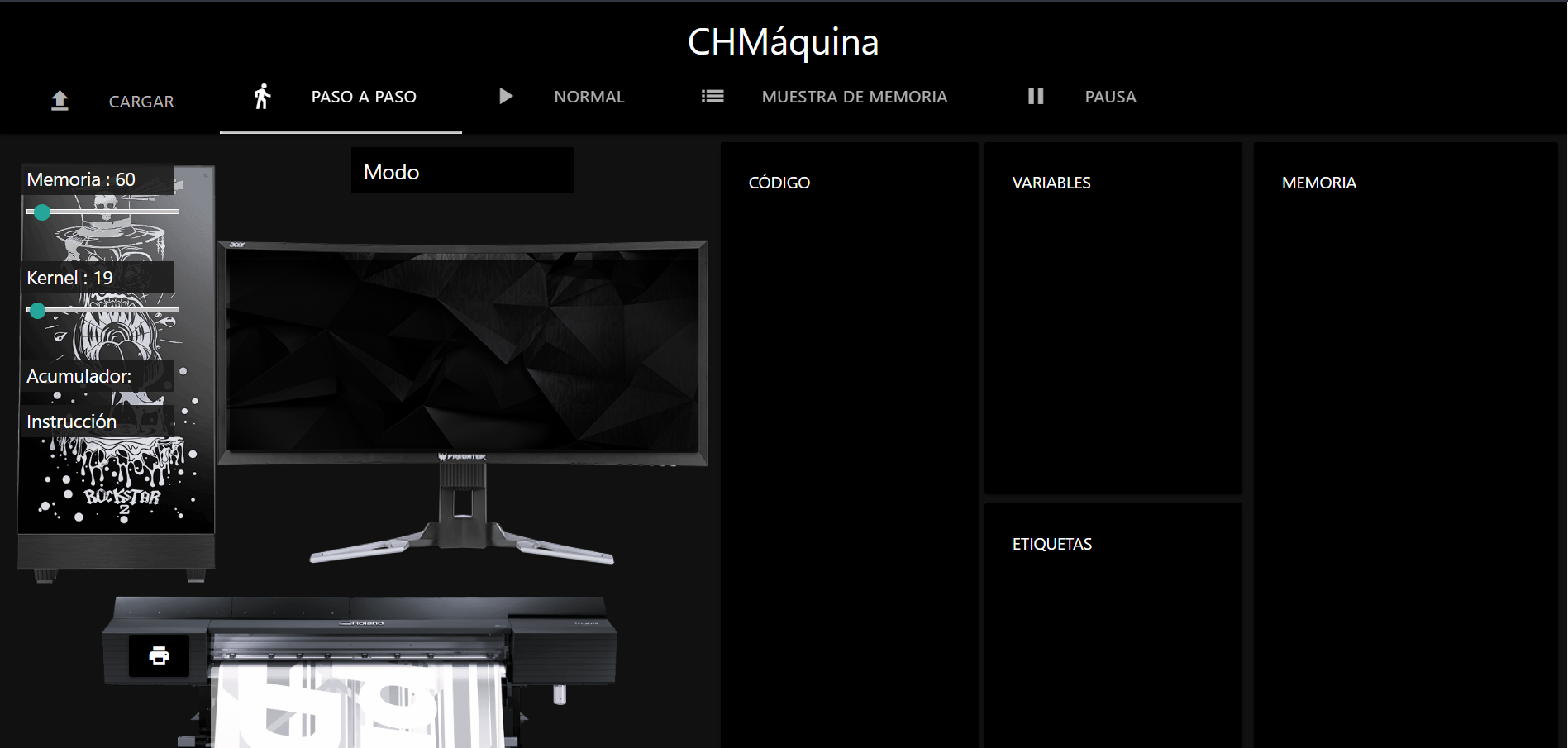
* El código puede tener comentarios por líneas, los cuales se denotarán por dos
* backslash (//) en las dos primeras posiciones de la instrucción, de igual manera se podrán insertar líneas en blanco entre instrucciones del programa.
* Se podrán realizar operaciones entre valores enteros y reales, los 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.

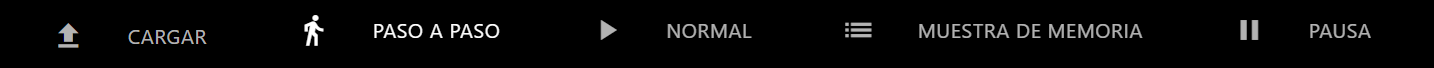
Las instrucciones que se pueden utilizar los las siguientes:

|  |  |
| --- | --- |
| Operación | Descripción |
| cargue | Cárguese/copie en el acumulador el valor almacenado en la variable indicada por el operando. |
| almacene | Guarde/copie el valor que hay en el Acumulador en la posición de memoria que corresponda a la variable indicada por el operando. |
| nueva | 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.  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. |
| lea | Lee por teclado/pantalla el valor a ser asignado a la variable indicado por la variable referida en el operando. |
| sume | Incremente/sume al valor del acumulador el valor indicado por la variable señalada por el operando. |
| reste | Decremente/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 continue 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 continue 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. |

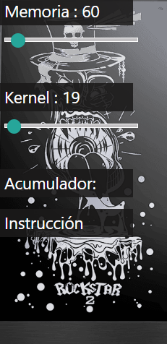
## **Interfaz gráfica**

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.
* Normal: Modo de ejecución automática.
* 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.

* Imagen que contiene Patrón de fondo

  Descripción generada automáticamenteMuestra el modo en el que se encuentra el software.

Imagen que contiene interior, tabla, computadora, escritorio

Descripción generada automáticamente

* Monitor: Presenta resultados de la ejecución de programas.



* Imprime los resultados de la ejecución de programas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

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

Texto

Descripción generada automáticamente

* Muestra la ocupación de las posiciones de la memoria.