

# Manual de usuario ChMaquina

Jhon Alejandro Franco Gómez

jhon.1702011356@ucaldas.edu.co

0000013167

Universidad de Caldas

Manizales, Caldas

## Tabla de contenido:

3.....	Interfaz de usuario
7.....	Explicación editor CH
8.....	Cómo escribir en lenguaje CH
10.....	Ejemplos de programas CH

## Interfaz gráfica:

Cuando se le da inicio a la máquina, se encontrará con un título en la parte superior que dice 'ChMáquina', y debajo de este habrá una imagen de Tux, la mascota del kernel Linux.

Abajo de esta imagen encontrará una serie de botones el cual se detallan a continuación:



## Establecer valores:

Este botón es el primero que se debe accionar antes de iniciar la ejecución del ChMáquina, puesto que define los valores de memoria y kernel que se utilizaran durante la ejecución.

## Cargar .CH:

Este botón es el medio por el cual se pueden cargar archivos .ch para el funcionamiento de la máquina; al clicar en "Seleccionar archivo" se desplegará su explorador de archivos del entorno donde esté ejecutando la máquina. Tenga en cuenta las siguientes recomendaciones:

- Los nombres de las variables en cada programa no deben repetirse en la memoria, es decir, cada programa tiene sus propias variables sin repetirse.
- Asegúrese de que tiene espacio suficiente en la memoria para cargar las instrucciones.

- Valide la sintaxis de cada programa, de lo contrario no podrá añadirlo a la memoria.

## Mostrar mapa de memoria / Ocultar mapa de memoria:

Como su nombre indica, estos botones permiten desplegar y ocultar un mapa donde podrá visualizar la memoria de la máquina.

## Memoria:

Esto no es un botón, es un campo numérico donde puede ingresar la cantidad de memoria que desea asignarle a la máquina para su ejecución. Su valor por defecto es 120

## Kernel:

Otro campo numérico donde se deberá ingresar la cantidad que ocupará el Kernel en la memoria. SIEMPRE debe ser menor que la memoria. Y en lo recomendable dejar un buen espacio de memoria disponible para añadir programas.

## Iniciar:

Este botón permite darle ejecución a la máquina con las instrucciones cargadas en la memoria.

## FCFS:

Este botón da inicio al ChMáquina utilizando el algoritmo de ordenamiento de procesos First Come First Served, el cual permite la ejecución de los programas cargados en memoria, ordenándolos por tiempo de llegada sin discriminar por tiempo que tarda cada uno en ejecutarse.

## SJF No Expropiativo:

Este botón da inicio al ChMáquina utilizando el algoritmo de ordenamiento de procesos Shortest Job First, el cual toma en cuenta el tiempo que

tarda cada programa en ejecutarse, para ordenarlos de menor a mayor y que de esta manera lo más rápidos se puedan ejecutar primero.

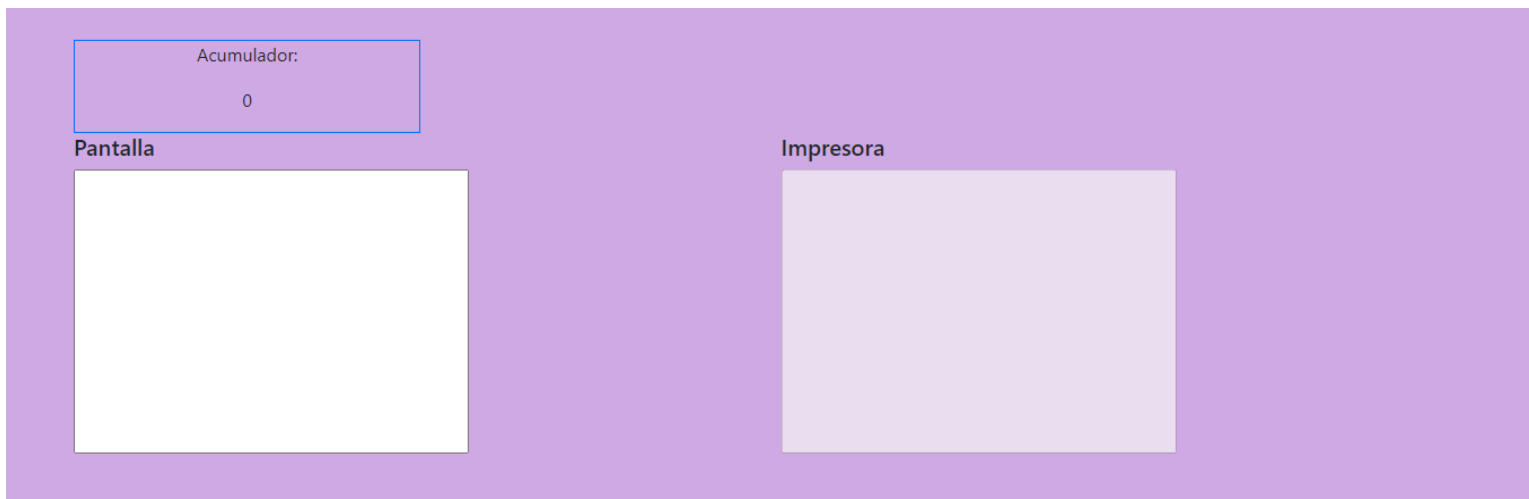
## Ejecución paso a paso:

Ese botón genera la ejecución de de la máquina, pero con una serie de notificaciones que indican lo que está sucediendo durante la ejecución.

## Crear programa CH:

Este botón redirecciona a un editor de programa, donde se le permite al usuario crear su propio programa .ch

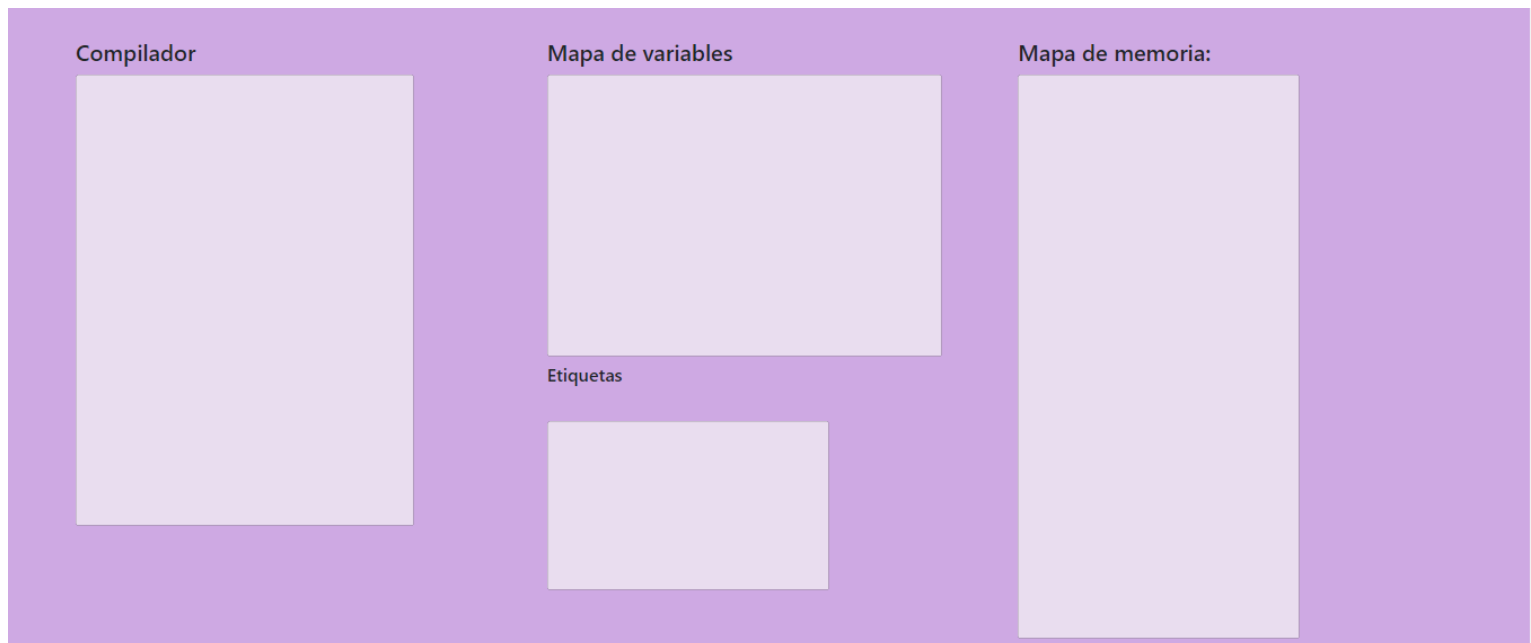
Si nos desplazamos hacia abajo nos encontraremos con un acumulador, y dos secciones que representan la pantalla y la impresora.



La función del acumulador es llevar cuenta de los procesos algorítmicos que realicen las instrucciones dadas por los programas .ch, palabras reservadas del lenguaje CH como cargue, lea, sume, reste, multiplique, divida, potencia, etc, son usadas para modificar el valor de este acumulador y operar con el.

La función de los campos *Pantalla* e *Impresora* son para mostrar información, la cual también se manipula mediante funciones como imprima o muestre.

Continuando con el desplazamiento, podrá encontrar una última sección donde se ven los siguientes campos



### Compilador:

Este campo muestra las instrucciones escritas en los archivos .ch y que están cargados en la memoria.

### Mapa de variables:

Este campo, como su nombre indica, es el campo donde se van a mostrar todas las variables creadas y cargadas en memoria, en este campo solo se mostrará el nombre de las variables.

### Mapa de memoria:

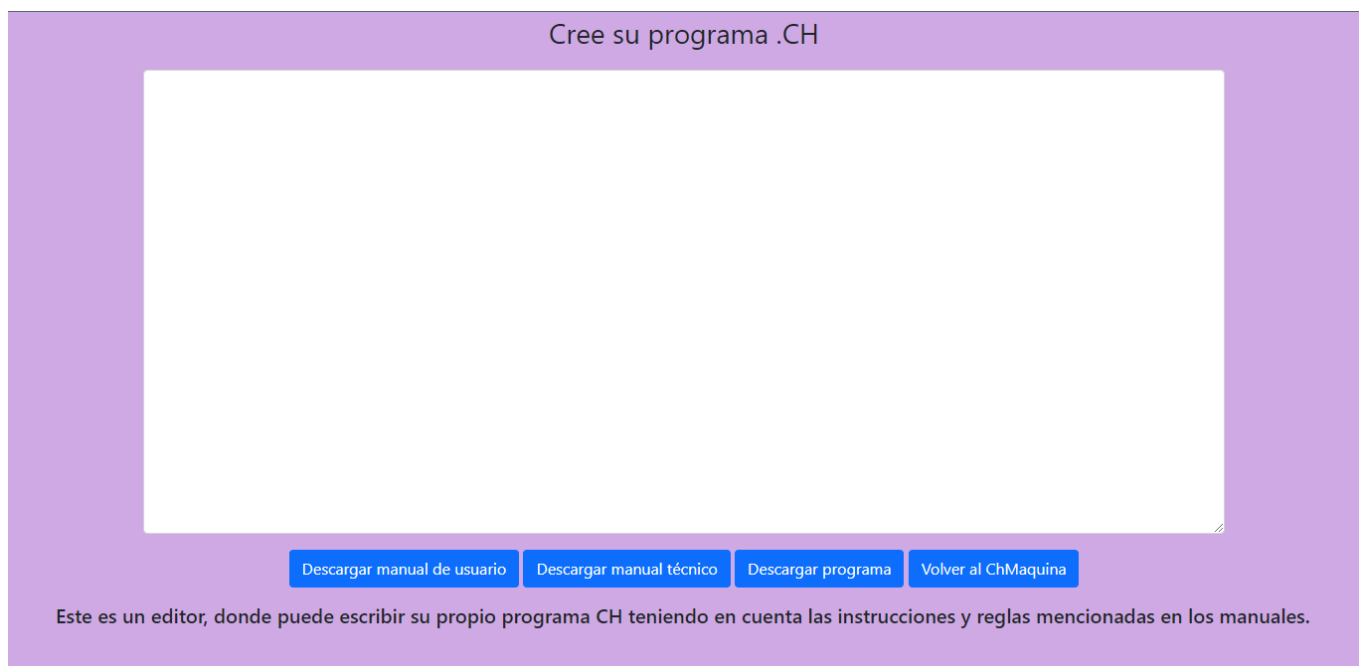
Este mapa permite al usuario conocer las posiciones de memoria que tienen las instrucciones cargadas

## Etiquetas:

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.

## Editor CH:



## Interfaz del editor:

Al ingresar al editor mediante el botón "Crear programa .ch", el sistema nos mostrará un editor de texto en el que podemos escribir un programa para el ChMáquina usando como lenguaje de programación el lenguaje CH; el cual se explica a continuación.

## El lenguaje de programación CH:

Se compone de instrucciones muy simples.

Cada línea del archivo .ch es una instrucción, la cual le indicará al programa cómo actuar y qué acciones realizar; a continuación se dispondrá de una lista que describe la función de cada instrucción.

Adicionalmente se pondrán un par de ejemplos de programas CH para dar mayor claridad.

Tenga en cuenta que las palabras reservadas del lenguaje deben estar escritas en minúsculas; a excepción de las instrucciones **Y**, **O**.

**Cargue:** Cargar 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 variable indicada por el operando.

**Lea:** Lee por teclado el valor a ser asignado en la variable indicada por la variable referida en 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, I = Entero, R = Real, L = Lógico (1 = Verdadero o 0 = Falso)), un tercer operando establecerá un valor de inicialización.. Las variables no inicializadas en el tercer operando tienen por defecto el valor cero (0) para reales y enteros, espacio para cadenas, 0 para lógicos.

**Sume:** Incremente el valor del acumulador en el valor indicado por la variable señalada por el operando.

**Reste:** Decrementa el acumulador en 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).

**Módulo:** 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).

**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.

**N0:** 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:** Se usa la impresora para mostrar el valor que hay en la variable indicada por el operando, si el operando es acumulador muestre el valor del acumulador.

Vaya: Salte a la instrucción que corresponde a la etiqueta indicada por el operando y siga la ejecución a partir de allí.

Vayasi: Salte a la instrucción que corresponde a la etiqueta indicada por el operando y siga la ejecución a partir de allí, si y sólo si:

- Si el valor del acumulador es mayor a cero, salte a la instrucción que corresponde a la etiqueta indicada por el primer operando.
- Si el valor del acumulador es menor a cero, salte a la instrucción que corresponde a la etiqueta indicada por el segundo operando.
- Si el valor del acumulador es igual a cero, salte a la siguiente instrucción.

Etiqueta: Crea una nueva etiqueta, donde el primer operando es el nombre que tendrá la etiqueta y el segundo operando es la posición de instrucción que almacena la etiqueta. La posición es un valor numérico el cual tiene un rango definido por la cantidad de líneas que tiene el programa.

Retorne: Es la última instrucción de TODOS los programas .CH, asegúrese de incluirla para dar por terminado el programa y que el ChMáquina pueda continuar con la correcta ejecución de los demás programas cargados.

### **Ejemplos:**

```
nueva x I  
lea x  
muestre x  
retorne
```

```
nueva prueba2 I 15  
imprima prueba2  
retorne
```