

---

# Manual Técnico ChMaquina

---

## Manual Técnico del Programa de Simulación de Sistema Operativo

### 1. Introducción

Este manual técnico está destinado a proporcionar información detallada sobre el programa de simulación de sistema operativo, el cual fue programado utilizando JavaScript y HTML. El programa permite cargar archivos de tipo .ch, que contienen instrucciones que simulan "programas", y luego ejecutar estas instrucciones paso a paso o de corrido. Además, el programa realiza una validación sintáctica para asegurar que se cumplan las reglas establecidas para una correcta sintaxis.

### 2. Validación de archivos

El programa solo permite la carga de archivos con extensión .ch, lo que se logra mediante la validación de una función programada. La validación se realiza utilizando la extensión del archivo y solo permite cargar archivos de texto con esta extensión.

### 3. Validación sintáctica

El programa realiza una validación sintáctica de las instrucciones contenidas en los archivos cargados. La validación se realiza línea por línea, utilizando una función programada que guarda las palabras reservadas del lenguaje de programación en arreglos. La función utiliza la función split para dividir cada línea en palabras y luego valida cada palabra para garantizar que cumpla con las reglas sintácticas.

### 4. Ejecución del programa

Una vez que se valida la sintaxis, el programa se carga en memoria y se ejecuta. El programa permite ejecutar las instrucciones paso a paso o de corrido. Para la ejecución de las instrucciones, el programa extrae la primera posición del string que contiene la instrucción y busca en los arreglos de palabras reservadas para determinar la acción correspondiente. A continuación, se ejecuta la función programada que corresponde a la acción mencionada. Este proceso se repite hasta que se han ejecutado todas las líneas de cada programa.

---

## 5. Creación de variables y etiquetas

Antes de ejecutar el programa, se debe cargar las variables y etiquetas mediante la función cargar variables y etiquetas. Esta función filtra el array buscando las palabras reservadas para la creación de variables y etiquetas, y para cada una de ellas genera un JSON que se guarda en un array. El JSON contiene el nombre de la variable o etiqueta, la posición en memoria (que se determina después de cargar las instrucciones) y la posición en la lista de instrucciones.

## 6. Memoria

El programa cuenta con dos arreglos unidimensionales para el manejo de la memoria: el arreglo de memoria y el arreglo kernel.

El arreglo de memoria es donde se almacenan tanto las instrucciones del programa como las variables que se crean durante la ejecución. Este arreglo es unidimensional y se maneja como un arreglo lineal, es decir, cada posición de memoria se almacena secuencialmente a partir de la posición 0.

El arreglo kernel es también unidimensional y tiene un tamaño fijo. Una vez que se inicia el programa, la memoria en la posición 0 contiene la variable acumulador seguida del contenido del arreglo kernel. El arreglo kernel contiene instrucciones y variables que se ejecutan en todo momento, como por ejemplo la variable acumulador.

Cada línea de instrucción en los archivos cargados por el usuario ocupa un espacio en memoria, esto significa que cada línea se almacena en una posición de memoria específica del arreglo de memoria. Las variables creadas durante la ejecución del programa también ocupan un espacio en memoria.

Es importante tener en cuenta que la memoria es un arreglo lineal, es decir, no cuenta con estructuras de datos complejas como listas enlazadas o árboles, sino que se maneja de manera secuencial. Por lo tanto, es necesario tener un cuidado especial al manejar la memoria para evitar errores en la ejecución del programa.

## 7. Consideraciones adicionales

El programa permite cargar varios archivos siempre y cuando estos no superen el tamaño de memoria. Además, el programa está diseñado para ser compatible con JavaScript y HTML, lo que lo hace altamente personalizable y fácil de integrar en otros proyectos.