



2 DE AGOSTO DE 2021

GETIME
DOCUMENTACIÓN OFICIAL DE LA VERSIÓN 1.0.0

AUTHOR
ORGANISM – SHIELD ENCODERS



Contenido

AVISO.....	2
.....	2
ABSTRACTO	1
FUNCIONAMIENTO A GRANDES RASGOS	1
DEPENDENCIAS	1
USAR PROGRAMA DESDE CUALQUIER RUTA.....	1
EXPLICACIÓN	1
PASOS	1
MANUAL DE USO.....	2
LEER ANTES DE USAR	2
PARA DESARROLLADORES.....	3
ALGORITMO GENERALIZADO (Comprender el por qué se diseñó de esta manera)	3
PROBLEMA	3
SOLUCION.....	4
DIAGRAMA DE FLUJO DEL ALGORITMO GENERAL (Para codificarlo en otro lenguaje)	5
DIAGRAMA DE FLUJO DE LA V 1.0.0.....	6

AVISO

Si eres desarrollador y quieres codificar este programa en otro lenguaje, entonces te interesan los apartados: [PARA DESARROLLADORES](#), [DIAGRAMA DE FLUJO DEL ALGORITMO GENERAL](#).

ABSTRACTO

Este programa fue desarrollado por un equipo independiente de programadores con propósito de crear una herramienta con dependencias nativas del sistema operativo **WINDOWS**, que sea capaz de **calcular el tiempo de ejecución de un ejecutable**, en otras palabras, esta herramienta fue escrita en **MS-DOS** también conocido como **BATCH**, por lo que no es necesario instalar nada para poder utilizarla ya que el intérprete viene preinstalado por defecto en los sistemas operativos **WINDOWS**, además la herramienta es fácil de modificarse para alterar su comportamiento ya que se trata de un lenguaje interpretado.

FUNCIONAMIENTO A GRANDES RASGOS

El programa utiliza la pseudo variable de entorno **TIME**, para obtener la hora del sistema antes de llamar el ejecutable a fin de establecer la hora en la que se inicia la ejecución. Una vez el ejecutable termina su ejecución, el programa vuelve a obtener la hora del sistema para determinar en qué momento finalizó la ejecución, para posteriormente hacer uso de un **algoritmo que resta las unidades de tiempo inicial de las unidades de tiempo final**, lo que da como **resultado la duración de la ejecución**. Al final muestra la hora en la que el programa empezó a ejecutarse, seguido de la hora en la que el programa finalizó su ejecución y la duración.

TIME = HORAS:MINUTOS:SEGUNDOS.CENTISEGUNDOS → 01:34:56.91

DEPENDENCIAS

- **CMD.EXE**

USAR PROGRAMA DESDE CUALQUIER RUTA

UNA VEZ TENGAS EL ARCHIVO **gettime.bat** en tu sistema, te conviene seguir las siguientes instrucciones para poder utilizar el programa desde cualquier ruta en la que te encuentres (si piensas descargar más herramientas creadas en **batch**, esto te va a ser de mucha ayuda):

EXPLICACIÓN

La variable de entorno **PATH** es la que contiene las rutas a las que el sistema va en busca ejecutables al momento de que tú escribes un comando para ejecutarlo, es decir cuando escribes una palabra en el **cmd.exe** y das **"ENTER"**, el programa va a las rutas que contiene **PATH** para buscar un archivo de nombre igual a la palabra que escribiste, y si no encuentra coincidencias, entonces te dice que tu palabra no se reconoce como comando interno, externo, archivo por lotes ejecutable o programa. Entonces nuestro objetivo es agregar una nueva ruta la cual va a contener nuestros **batch**, a esa variable para que el sistema también la tome en cuenta al momento de buscar.

PASOS

- Crea una carpeta en la ruta que quieras para que guardes ahí tus **BATCH**.
- Abre un símbolo de sistema y modifica la variable de entorno **PATH**.
 - Para que la modificación quede almacenada para el usuario actual

- **setx Path** "%Path%;C:\Users\ruta\de ejemplo\\"", no importa si la ruta tiene espacios de igual manera servirá y asegúrate que tenga \ al final.
 - Para que la modificación quede almacenada para todos los usuarios del sistema
 - **setx Path** "%Path%;C:\Users\ruta\de ejemplo\ /M"
- Abre una nueva instancia de cmd.exe (símbolo de sistema) y visualiza el contenido de la variable **Path**, si todo fue correcto, ahora deberías poder ver la ruta que añadiste al final de todo el contenido de la variable.
 - **set Path**

Ahora debes poder llamar a los programas alojados en la ruta agregada, desde cualquier otra ruta.

MANUAL DE USO

getime [-a] <nombre_Del_Ejecutable>

opciones:

- a Habilita el soporte para las secuencias de escape ANSI (MUESTRA LOS MENSAJES COLOREADOS). (El número de compilación de tu sistema debe ser 1909 o mayor). Si puedes ver el texto en color verde que se muestra en la consola cuando se te enseña el manual entonces tu sistema soporta esta opción.
- d Se activa el modo Demostración.
No mide el tiempo de otro programa, sino que mide el tiempo de ejecución de sí mismo sin ANSI.
Si se activa esta opción, no importa si otra opción ha sido activada o si el nombre de un ejecutable válido ha sido especificado, se anulan todo comportamiento y se activa el modo demostración.

LEER ANTES DE USAR

- El programa **no** cuenta con una versión de **GUI** por el momento y usa el **cmd.exe** como intérprete.
- Fue desarrollado en **BATCH**.
- Se recomienda que se ejecute en una versión de Windows igual o mayor a **1909** (puedes la versión ejecutando el comando **winver** → compilación del SO) ya que el programa usa secuencias de escape **ANSI** para colorear ciertas palabras que se muestran por pantalla, EN CASO de que su sistema no cuente con soporte para secuencias de escape ANSI, no active la opción.
- Comunicarse a OrgAnismSupport@Outlook.com para recibir soporte técnico.
- El programa solamente puede medir la duración de la ejecución de un programa hasta 23 horas, si su medición requiere de días para finalizar este programa no le es de mucha utilidad.

PARA DESARROLLADORES

- Si lo que te interesa es **extraer la parte funcional del código** que **obtiene los tiempos y calcula la duración**, ya que el programa no se hizo pensando en la modularidad, sino que se optó por hacer etiquetas dentro del mismo, para simular el comportamiento de funciones, la parte funcional está en las líneas
 - **126** -> obtiene el tiempo inicial.
 - **137** -> llama el ejecutable
 - **166** -> obtiene el tiempo final
 - **187-190** -> obtiene las unidades de tiempo inicial por separado (centisegundos, segundos, minutos, hora).
 - **194-201** -> condicionales para arreglar el bug octal (leer línea **420** del programa)
 - **204-207** -> obtiene las unidades de tiempo final por separado.
 - **211-218** -> condicionales para arreglar el bug octal
 - **221-231** -> condicionales para determinar si es necesaria la conversión de unidades en caso de que alguna de las unidades de tiempo inicial sea mayor a su correspondiente del tiempo final.
 - **234-247** -> conversión de centisegundos y segundos de tiempo final, además de

ALGORITMO GENERALIZADO (Comprender el por qué se diseñó de esta manera)

Este apartado expone el **algoritmo** lo más **generalizado** posible para que pueda ser codificado en cualquier lenguaje, la forma en la que se presenta **el problema y su solución**.

PROBLEMA

Se requiere que, dada **la hora inicial** de la ejecución de un programa y **la hora final**, se determine **la duración de la ejecución** del mismo.

$$\text{Duración} = \text{TiempoFinal} - \text{TiempoInicial}$$

Pero antes de comenzar a operar, es necesario verificar si alguna de las unidades de tiempo inicial es mayor a su correspondiente del tiempo final, ya que si es el caso la fórmula nos puede dar unidades negativas por ejemplo:

centisegundos del tiempo final > centisegundos del tiempo inicial = duracion en centisegundos.

Pongamos el ejemplo de ***centisegundosIniciales = 98, centisegundosFinales = 51***

Entonces si efectuamos la resta sería ***Duración = 51 - 98 = -47*** lo que para el usuario significa que la ejecución del programa duró **-47** centisegundos.

SOLUCION

1. Para obtener la duración de un proceso, basta con conocer el tiempo inicial y el tiempo final de la ejecución, es decir, qué hora era cuando se inició la ejecución y qué hora era cuando se finalizó.
2. Evitar el problema de números negativos:

En caso de que alguna de las unidades correspondientes al inicial, sean mayores a las del final entonces hay que efectuar la conversión de unidades para poder operar con el algoritmo convencional.

Nuestro objetivo es que las unidades del tiempo final sean todas mayores o iguales a las del tiempo inicial para poder restar las del inicial de las del final sin que nos den números negativos.

La conversión consiste en aumentar las unidades del tiempo final en tantas unidades equivale de su orden superior inmediato y disminuir en 1 unidad su unidad final de orden superior inmediato, es decir si es el caso en el que los centisegundos iniciales sean mayores a los finales entonces se hace lo siguiente:

suma 100 unidades a los centisegundos finales (aumentamos las unidades del tiempo final en tantas unidades equivale de su orden superior inmediato, en este caso los segundos) **y resta 1 unidad a los segundos finales** (restamos 1 unidad de los segundos, ya que representan el orden superior inmediato de los centisegundos)

de esta manera se obtiene el equivalente de segundos en centisegundos.

Retomando el ejemplo anterior:

tiempo inicial = 1 segundo con 98 centisegundos, y tiempo final = 2 segundos con 51 centisegundos

también puede ser representado como:

tiempo inicial = 0 segundos con 198 centisegundos y tiempo final 0 segundos con 251 centisegundos (ya que 1 segundo equivale a 100 centisegundos)

una vez hecha la conversión, se puede operar tranquilamente:

251 centisegundos – 198 centisegundos = 053 centisegundos.

Entonces el programa duró 53 centisegundos en ejecución.

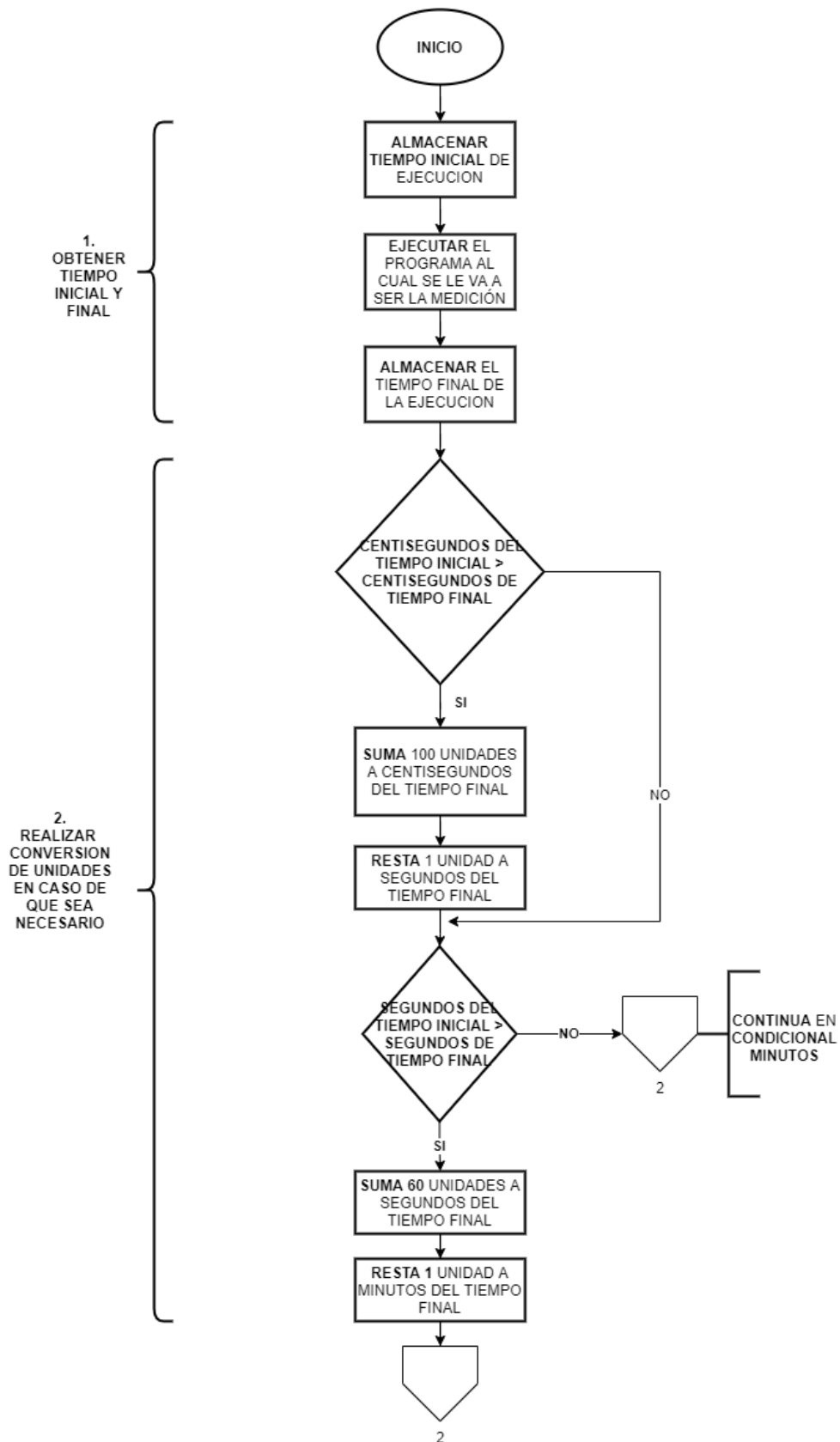
Esto se repite con cada una de las unidades a **excepción** de las **horas**.

3. Por último, podemos aplicar la fórmula tranquilamente

$$\text{Duración} = \text{TiempoFinal} - \text{TiempoInicial}$$

- a. Hora inicial \rightarrow 1:20:34.78
- b. Hora final \rightarrow 1:75:51.99 \rightarrow 2:15:51.99
- c. Duración \rightarrow 0:55:17.21

DIAGRAMA DE FLUJO DEL ALGORITMO GENERAL (Para codificarlo en otro lenguaje)



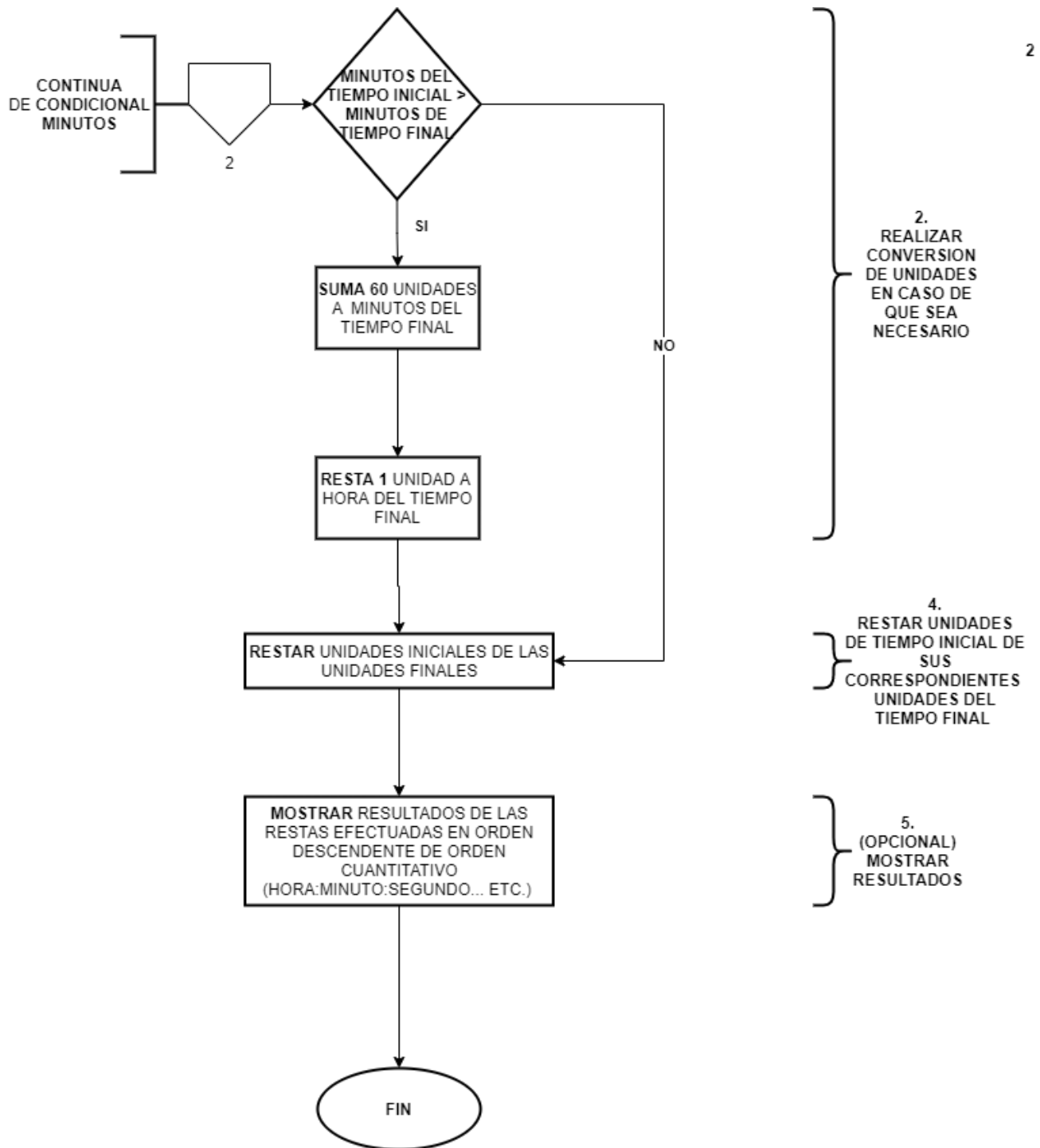


DIAGRAMA DE FLUJO DE LA V 1.0.0
(VER SIGUIENTE PÁGINA)