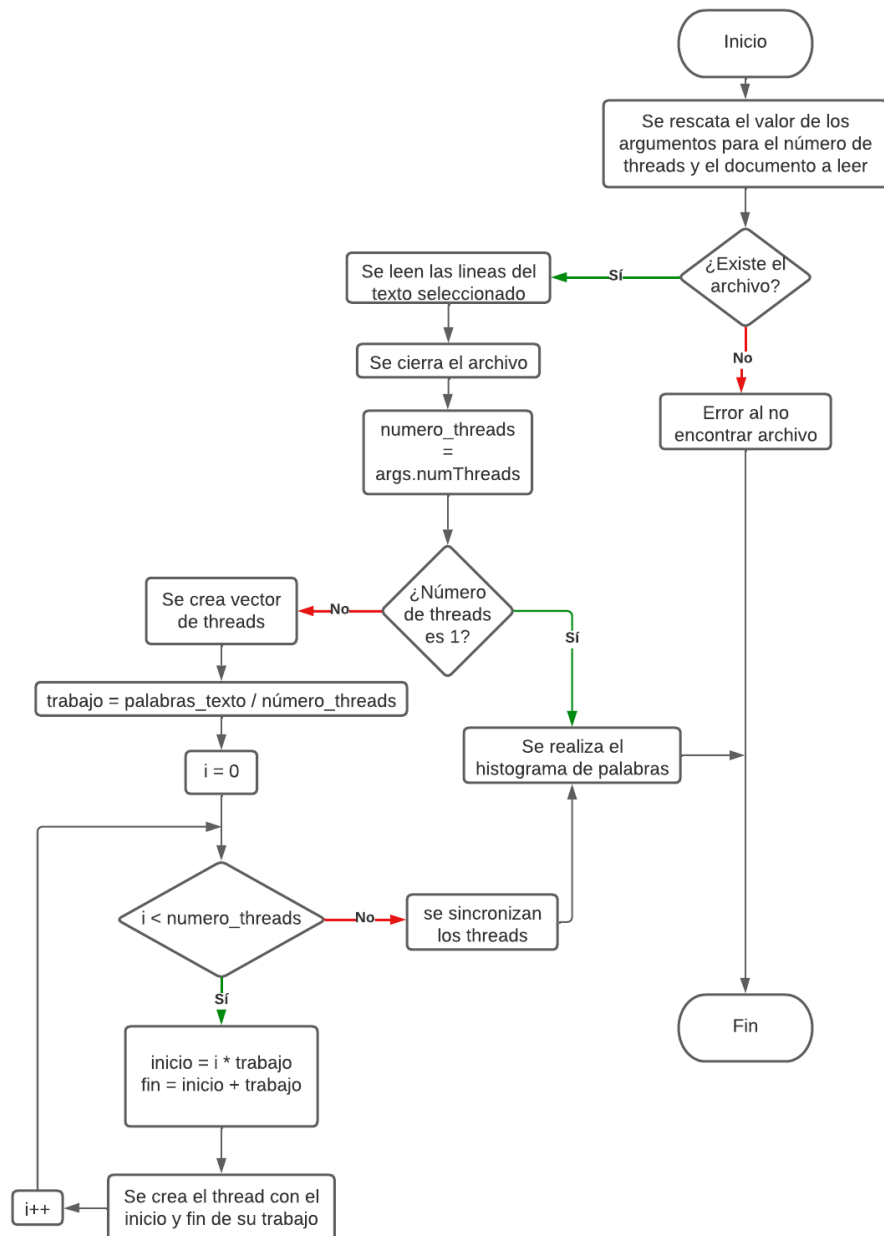


Diagrama de Flujo

Erik Chandia Ulloa, erik.chandia@alumnos.uv.cl

Iam Gallano Vasquez, iam.gallano@alumnos.uv.cl

Vicente Gonzáles Contreras, vicente.gonzales@alumnos.uv.cl



El diagrama de flujo presentado muestra el recorrido que realiza cada vez que se ejecuta el código, e inicia capturando los valores de argv entregados a la hora de iniciar el programa. El objetivo de la captura de argumentos es poder tener un número variable de threads, además de indicar el archivo que se leerá. A continuación, se verifica la existencia del archivo entregado, en caso de que este no exista, el programa terminará con un mensaje de error. Ya verificada la existencia del archivo, se procede a leer el contenido de este y se almacena en una variable para poder cerrar el archivo. Ya teniendo el contenido del archivo, se asigna el número de threads que se ingresaron por argumentos, y se verifica si el número indicado es mayor que 1. En caso de que se haya seleccionado 1 solo thread, el programa funcionara de forma secuencial, de la misma forma como lo hacia originalmente, pero en caso de que hayan mas threads, se crea un vector que los almacenará. Una vez indicado que el número de threads es mayor que 1, se calcula el trabajo que debe realizar cada hilo, para que estos tengan la misma cantidad de trabajo repartida, y dicho trabajo lo calculamos como el total de palabras del archivo dividido el número de threads. A continuación, se crea un ciclo que les asigna el inicio y el final a cada hilo, esto con el objetivo de que los threads no lean las mismas palabras. El inicio se calcula como multiplicando el trabajo de cada hilo por el número ID del hilo, y el fin se calcula sumándole el trabajo al inicio del hilo. Ya habiendo configurado los threads, estos se sincronizan y se les asigna la función de histograma, lo que muestra en pantalla el resultado final del código y lo lleva al termino de este.