



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por nombre alumno
en Universidad de Burgos — 26 de junio
de 2019

Tutor: nombre tutor

Índice general

Índice general	I
Índice de figuras	II
Índice de tablas	III
Apéndice A Documentación técnica de programación	1
A.1. Introducción	1
A.2. Estructura de directorios	1
A.3. Manual del programador	2
A.4. Compilación, instalación y ejecución del proyecto	3
A.5. Pruebas del sistema	5
Apéndice B Documentación de usuario	7
B.1. Introducción	7
B.2. Requisitos de usuarios	7
B.3. Instalación	8
B.4. Manual del usuario	8

Índice de figuras

A.1. Así luce la ventana de Spyder	3
A.2. Configuración del fichero setup.py	4
B.1. Vista de la ventana principal	9
B.2. Vista de la ventana principal	10
B.3. Mensaje de error	10
B.4. Mensaje de advertencia	11
B.5. Aparición del botón tras la elección de un archivo .py	11
B.6. Ventana donde elegir operaciones	13
B.7. Ejemplo de la Listbox de procesadores	14
B.8. Ejemplo gráfica resultante análisis individual	15
B.9. Ejemplo gráfica resultante recalculada	16
B.10. Mensaje de error por elegir solo un fichero	17
B.11. Gráfico de barras resultante	18
B.12. Gráfico de puntos resultante	19

Índice de tablas

Apéndice A

Documentación técnica de programación

A.1. Introducción

En esta sección se explicaran los conceptos necesarios para poder ponerse a trabajar con este proyecto:

Lo primero antes de nada seria descargarse la herramienta desde aquí:

Se tiene que descargar desde aquí:

[https://github.com/Guillecal/TFG-Herramienta_para_medir_la_eficiencia_de_codigo_python/tree/master/Prueba%20TFG\](https://github.com/Guillecal/TFG-Herramienta_para_medir_la_eficiencia_de_codigo_python/tree/master/Prueba%20TFG)

A.2. Estructura de directorios

Los archivos necesarios están metidos en la carpeta Pruebas TFG, dentro de esta esta el archivo principal Vent.py que hace de controlador y vista, y este llama al interprete que se encuentra dentro de la carpeta byterun. Dentro de esta nos interesa el archivo pyvm2.py que es donde se realizo el upgrade, ademas de ser la partemas importante del interprete.

APÉNDICE A. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

A.3. Manual del programador

Se empezara preparando el entorno de trabajo para trabajar con el proyecto:

Python

Para el desarrollo de la herramienta se utilizó la versión de Python 2.7.16, es recomendable descargar esta versión para evitar algún tipo de incoherencia.

Link de descarga: [https://github.com/Guillecal/TFG-Herramienta_para_medir_la_eficiencia_de_codigo_python/tree/master/Prueba%20TFG\](https://github.com/Guillecal/TFG-Herramienta_para_medir_la_eficiencia_de_codigo_python/tree/master/Prueba%20TFG)

Seguido a esto es necesario instalar las bibliotecas utilizadas dentro del programa. Por ello es necesario comprobar si se tiene instalado primero el sistema de gestión pip, el cual es esencial para poder instalar las bibliotecas faltantes. Para este proyecto se utilizó la versión 19.1.1, pero en este caso la versión no debería afectar, hay que descargar el contenido que se encuentra en la siguiente dirección, guardar como get-pip.py:

Link Descarga: <https://bootstrap.pypa.io/get-pip.py>

Despues de de esto en el símbolo de sistemas hay que ejecutar el comando:
python get-pip.py

Cuando ya se tiene pip instalado ya solo sería meter os siguientes comandos por el símbolo de sistema:

```
pip install matplotlib  
pip install python-tk
```

IDE

Como entorno de desarrollo se podría se puede utilizar el IDE de Python, pero para el desarrollo de la herramienta se ha utilizado Spyder.

Link descaga: <https://www.spyder-ide.org/>

A.4. COMPILACIÓN, INSTALACIÓN Y EJECUCIÓN DEL PROYECTO

Una vez descargado e instalado este IDE ya viene listo para empezar poder trabajar, pero en caso de que se requiera, se pueden cambiar algunas configuraciones según el gusto de cada uno desde las pestañas Ver y Herramientas.

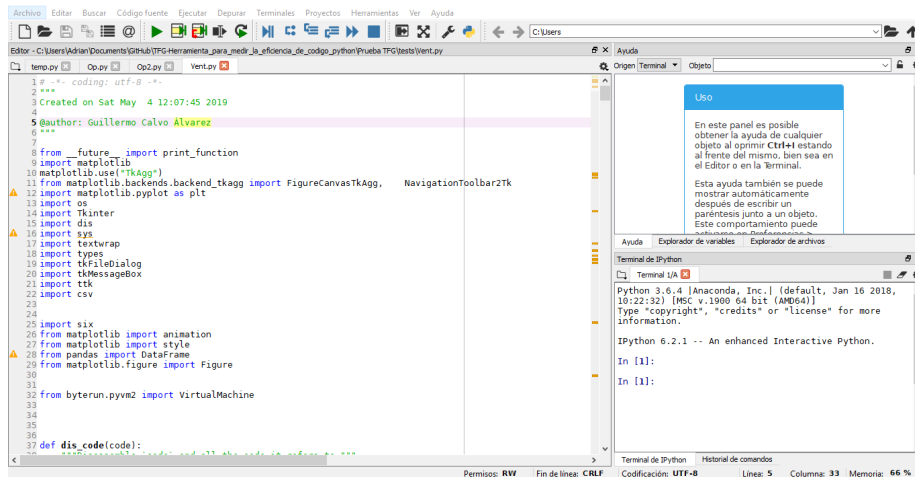


Figura A.1: Así luce la ventana de Spyder

Esta es una recomendación, si se está acostumbrado utilizar otro tipo de IDE que sirva para el lenguaje Python no habría ningún problema.

Github Desktop

Esta es la herramienta utilizada para gestionar mejor el repositorio donde se encuentra el trabajo. Se puede descargar desde aquí:

Link: <https://desktop.github.com/>

Una vez descargado e instalado, la primera vez que lo ejecutamos, podemos clonar desde aquí el proyecto desde el repositorio

A.4. Compilación, instalación y ejecución del proyecto

No se puede compilar al ser Python el lenguaje del código, ya que este es un lenguaje interpretado. Tampoco es necesaria una instalación, por cual lo

APÉNDICE A. DOCUMENTACIÓN TÉCNICA DE PROGRAMACIÓN

único que queda por hacer es ejecutar el código. Hay dos maneras de hacer esto:

- A través de la línea de comandos
- A través de un archivo ejecutable

Línea de comandos

Esta forma es muy sencilla. Simplemente hay abrir el Símbolo de sistema (Línea de comandos) y navegar hasta el directorio donde se encuentra el archivo Vent.py, esto se puede hacer fácilmente con el comando CD para cambiar de directorio y dir para mostrar los que contiene el directorio actual. Una vez encontrado el directorio simplemente hay que ejecutar el comando: Python Vent.py

Archivo ejecutable

Simplemente sería clicar en el archivo .exe que se encuentra dentro de la carpeta dist.

Pero para crear este archivo es necesario previamente haber configurado un fichero llamado setup.py que se tiene que encontrar en el mismo directorio que el archivo que se quiere hacer ejecutable, en nuestro caso es dentro de la carpeta prueba TFG. Setup.py se ha configurado de la siguiente manera:

```
# -*- coding: utf-8 -*-  
  
from distutils.core import setup  
import py2exe  
import matplotlib  
  
setup(console=['Vent.py'],  
      data_files=matplotlib.get_py2exe_datafiles())
```

Figura A.2: Configuración del fichero setup.py

Una vez configurado hay que dirigirse al directorio donde se encuentran desde el símbolo de sistema y ejecutar el comando "setup.py py2exe". Con esto debería generarse una carpeta llamada "distz" dentro de esta se encontraría el ejecutable

Como advertencia hay que comentar que podría ser necesario la descarga de un archivo .dll si no se tienen dentro de los archivos de python.

A.5. Pruebas del sistema

Haré vídeos cuando acabe los cambios en el código.

Apéndice B

Documentación de usuario

B.1. Introducción

En esta sección se explicara lo necesario para comprender como realizar un buen funcionamiento de la herramienta a través de unos pasos.

B.2. Requisitos de usuarios

Antes de nada hay que señalar una serie de requisitos previos para que se pueda ejecutar la herramienta:

1. Tener un ordenador con una version de Python 2.7 o posterior. junto con las librerías:.
2. Tener algún tipo de aplicación capaz de crear, leer y escribir archivos de-formado .csv (Recomendando MMicrosoft Excel o Apache OpenOffice Calc en su defecto)
3. Tener disponible algún tipo de archivo .py para analizar.

No son unos requisitos muy exigentes y una vez cumplidos ya se podría proceder con la instalación sin problemas.

B.3. Instalación

Realmente la herramienta no requiere de ningún tipo de instalación. Simplemente habría que descargar el proyecto del siguiente enlace:

https://github.com/Guillecal/TFG-Herramienta_para_medir_la_eficiencia_de_codigo_python

Se descargara un archivo.zip, cuyo contenido hay que extraerlo. Se puede hacer en el directorio que se quiera y una vez hecho simplemente habría que ejecutar el fichero con formato .exe.

Pero antes de esto hay que asegurarse de tener algún tipo de fichero.csv en la carpeta Prueba TFG dentro del proyecto, ya que si no, la herramienta no funcionara como es debido. En principio debería haber algún archivo. csv en el proyecto descargado desde la dirección anterior, pero por si acaso es mejor hacer esta revisión.

B.4. Manual del usuario

Una vez preparados los ficheros y los requisitos previos como esta herramienta no requiere de ningún tipo de configuración posterior, para poder iniciar de forma fácil la herramienta, hay creado un ejecutable que simplemente hay que clicar para iniciar. Cabe destacar que la interfaz de esta herramienta tiene todos los textos en inglés, a pesar de esto en este manual nos referiremos a las distintas opciones en su traducción al español.

Una vez este iniciado, muestra una interfaz bastante simple, que nos da a elegir entre tres opciones, Análisis individual, Análisis Comparativo o salir, aquí dependiendo del tipo de uso que se quiera hacer la herramienta, se elegirá uno u otro.

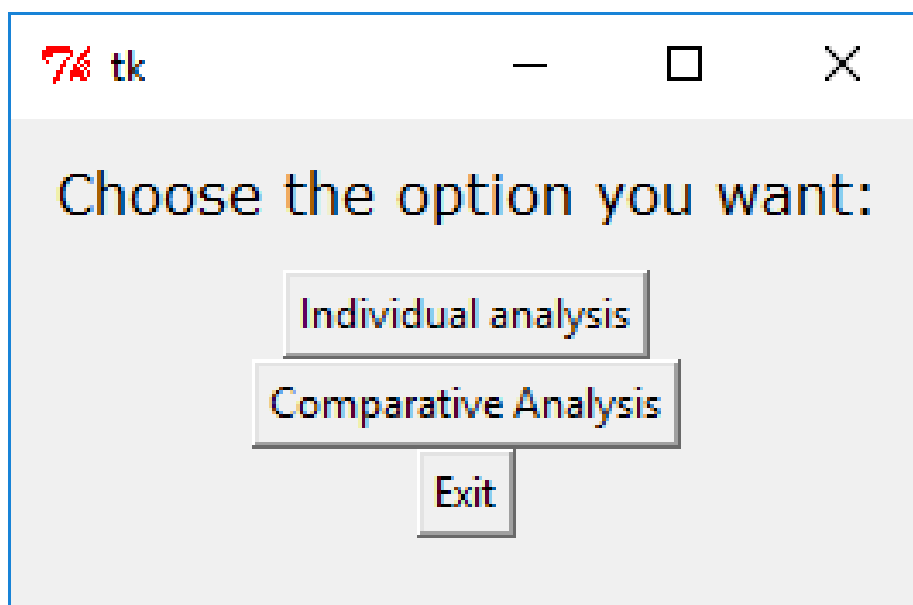


Figura B.1: Vista de la ventana principal

En el caso de querer ver el porcentaje de eficiencia que proporciona cada tipo de operación Por otra parte si lo que se quiere es hacer una comparación de eficiencia entre ficheros para determinar cuál es el que menos ciclos de reloj tarda en ejecutarse, el análisis comparativo es el que debes pulsar.

Análisis individual

Si pulsamos el análisis individual, lo primero que nos saldrá será una ventana con dos botones, buscar fichero y volver a la ventana principal.

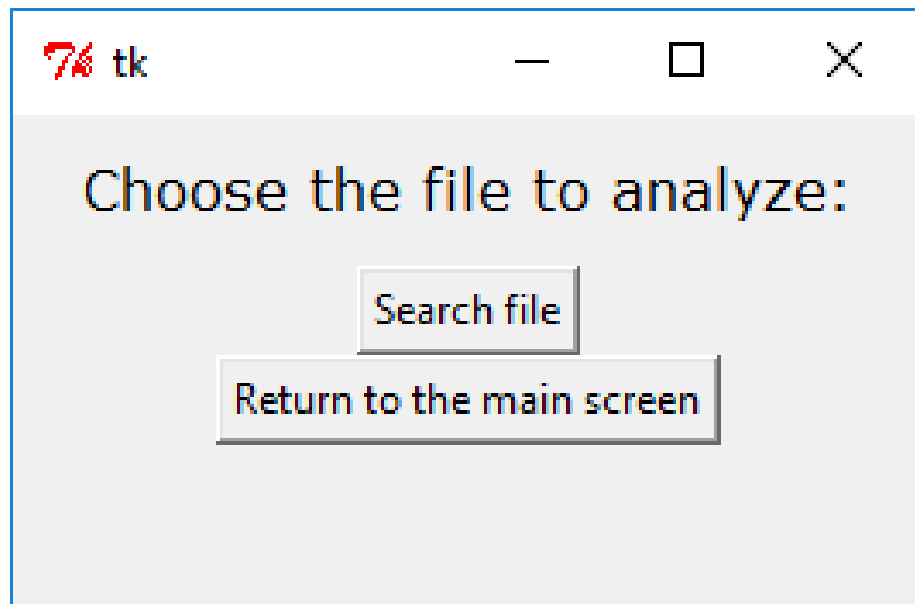


Figura B.2: Vista de la ventana principal

El primero de estos nos permite buscar el fichero que deseamos analizar a través de un explorador de archivos, el archivo seleccionado debe tener un formato.py, ya que si no nos saldrá un mensaje de error indicando que es un tipo de formato no valido.

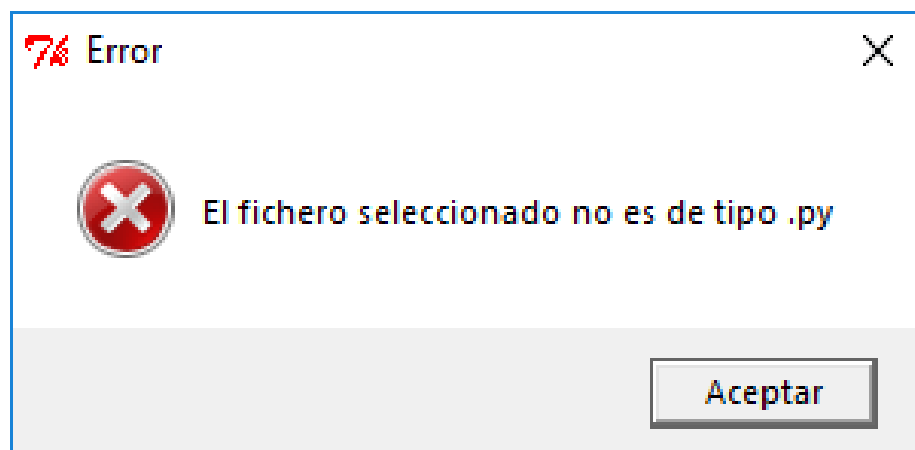


Figura B.3: Mensaje de error

Algo parecido aparecerá si no elegimos ningún fichero, como advertencia

saldrá también un mensaje

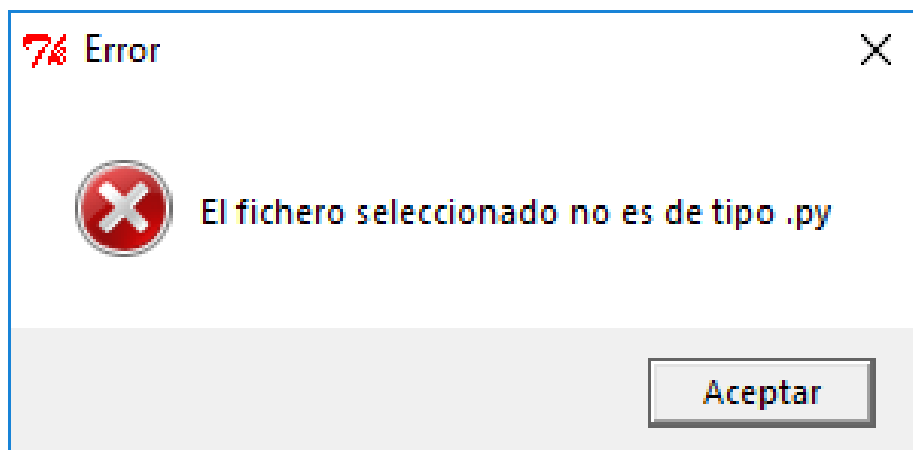


Figura B.4: Mensaje de advertencia

Si por el contrario pulsamos el botón de volver a la ventana principal, esto hará que aparezca de nuevo la ventana que se mostró al principio.

Una vez seleccionado el archivo .py aparecerá un nuevo botón en el que pondrá analizar fichero. A veces puede que tarde un poco en mostrar el fichero, porque internamente la herramienta tiene que hacer una serie de cálculos.

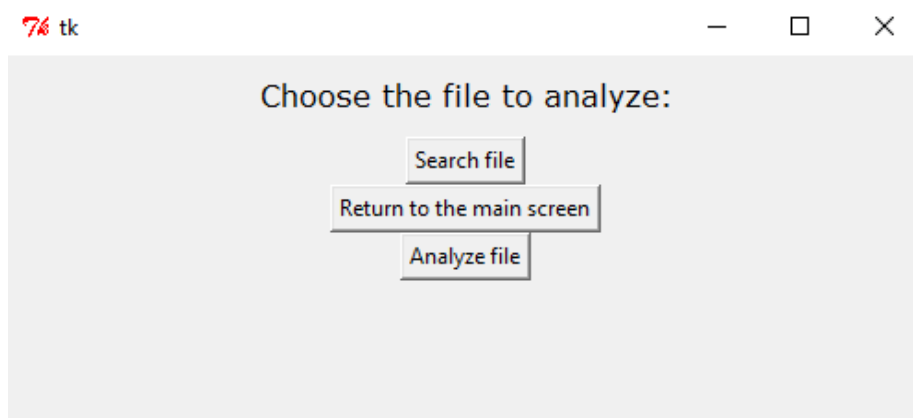


Figura B.5: Aparición del botón tras la elección de un archivo .py

Si pulsamos ese botón, nos mostrara la ventana de análisis. En esta ventana se muestran los resultados del análisis hecho al fichero que ha sido elegido en la ventana previa. Cuenta con una serie de opciones que hará variar los resultados que serán mostrados.

Echando un vistazo de arriba abajo nos encontramos con las siguientes opciones:

Primero al igual que en la ventana anterior a esta, nos encontramos con un botón que nos deja volver a la ventana inicial. Por si queremos realizar algún tipo de análisis más.

Despues hay un boton que nos lleva a otra ventana donde se pueden ver unos checkboxes que representan los diferentes tipos de operaciones encontrados en el análisis del fichero seleccionado, que por defecto al principio saldrán todos marcados. Esto indica que operaciones van a ser mostradas en la gráfica de los resultados, si hay algún tipo de operación que no queramos que salga tan solo tendríamos que clicar en el checkbox de la operación para quitarle la marca. Cuando queden marcadas aquellas operaciones que deseemos hay un boton para volver a la vista anterior en la parte de abajo.



Figura B.6: Ventana donde elegir operaciones

Seguido a esto hay una Listbox que deja elegir el procesador con el que se quiere ponderar a las operaciones. En esta lista se muestran los procesadores que tengamos disponibles en la carpeta Prueba TFG. Así que si queremos

que aparezcan mas tipos de procesadores tan solo habría que crear más procesadores teóricos ahí.

(Recomendación de copiar un procesador ya creado y tan solo tener que cambiar los valores según se desee)

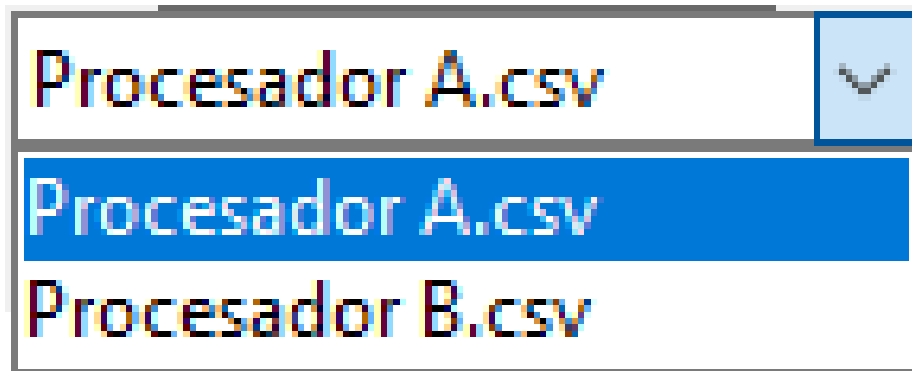


Figura B.7: Ejemplo de la Listbox de procesadores

Una vez configurados todos estos valores anteriores, habría que pulsar el botón de abajo Mostrar resultados, y tras esto saldría la gráfica resultante.

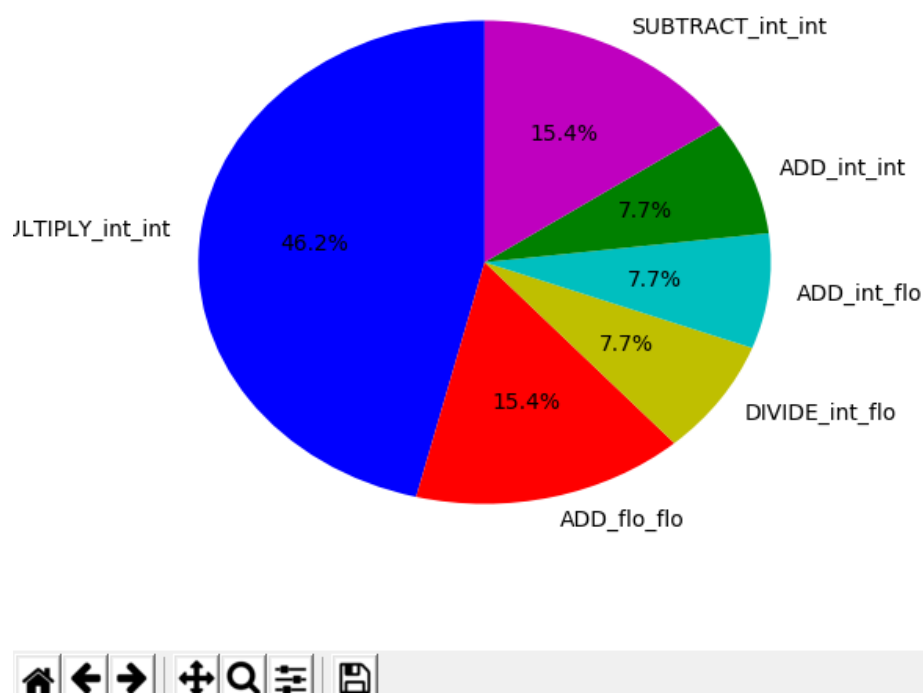


Figura B.8: Ejemplo gráfica resultante análisis individual

Hay que tener en cuenta que la configuración puede ser cambiada aunque ya se haya mostrado una gráfica y si se vuelve a pulsar el botón mostrar resultados, vuelve a recargar la gráfica con la nueva configuración que haya sido elegida.

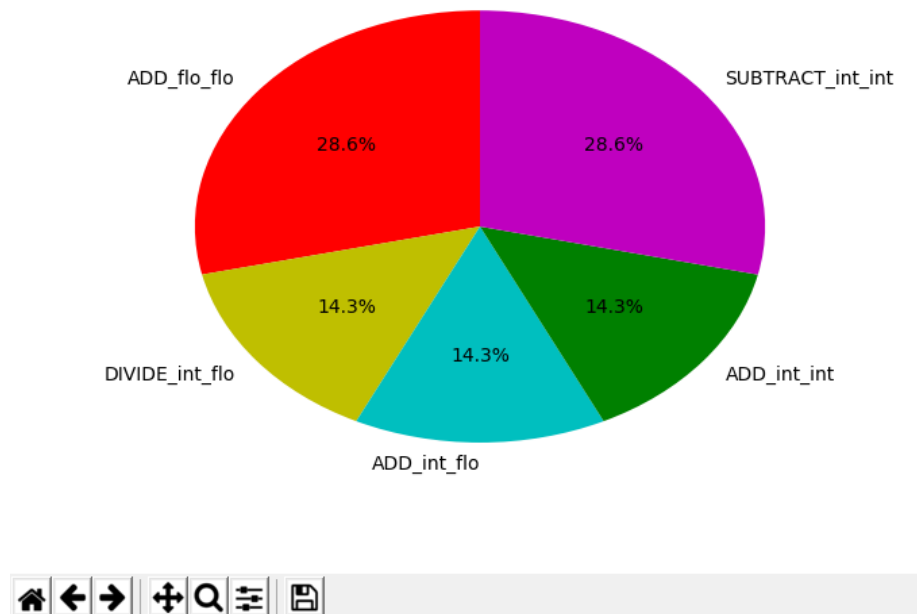


Figura B.9: Ejemplo gráfica resultante recalculada

En este caso se ha quitado la marca de la multiplicación de enteros y vemos como este desaparece de la gráfica. Esto mismo se podría hacer también con la listbox de los procesadores.

La gráfica muestra los porcentajes de ciclos de reloj que consume cada tipo de operación. Cada uno diferenciado por un color e indicado con el nombre de la operación y el nombre del tipo de los dos operadores.

Como se puede ver en la ultima imagen en la parte de abajo hay una serie de botones que nos permiten hacer un par de interacciones con el gráfico como por ejemplo ampliar la zona deseada o poder guardarla.

Análisis Comparativo

Pulsando el botón de análisis comparativo, Muestra una ventana con dos tipos de opciones, Buscar ficheros y volver a la ventana inicial. Esta ventana es prácticamente idéntica a la primera que sale en el análisis individual (Figura x). Pero la diferencia que hay en esta se trata del botón buscar ficheros, con el cual sale un explorador de archivos pero con el que hay que

seleccionar mas de un fichero. Al igual que en el análisis individual solo se pueden elegir ficheros con formato .py pero además no se puede elegir solo 1 fichero, ya que se hace alguna de estas dos cosas saldrá un mensaje de error.

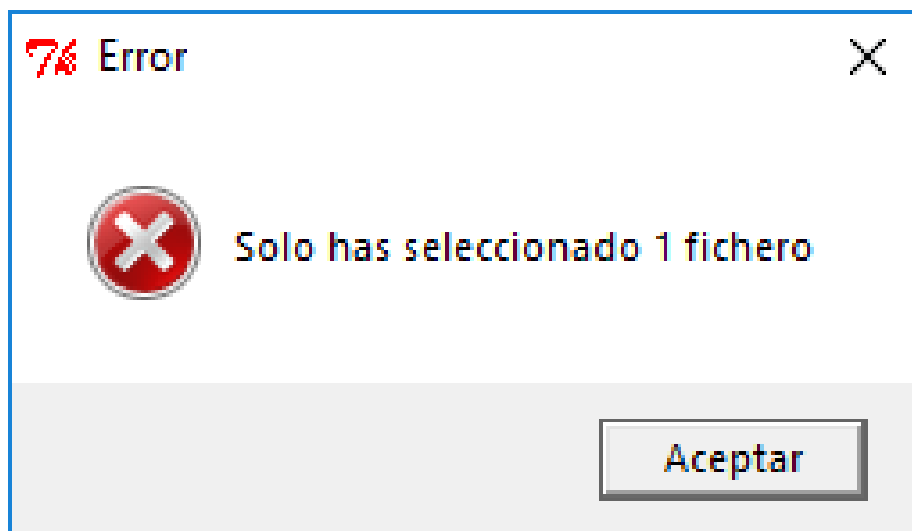


Figura B.10: Mensaje de error por elegir solo un fichero

Una vez seleccionado el fichero saldrá el botón de Analizar, y este nos pasará a la ventana de análisis.

Lo primero que vemos en la ventana de análisis es el botón de volver al menú principal en caso de que deseemos hacer el otro tipo de análisis o el mismos con otros ficheros.

Seguido hay una listbox que para indicar el procesador teórico con el que se desea ponderar las operaciones sacadas por el interprete.

Seguido a esto hay un boton que nos lleva a otra ventana donde se hay una serie de checkboxes, pero a diferencia de el análisis individual, aquí muestra todas las diferentes operaciones que aparecen en todos los ficheros metidos. Así que puede haber operaciones del chebox que se encuentren solo en un fichero.

Una vez elegidos los parámetros esta el botón mostrar resultados que nos muestra la grafica resultante, según los parámetros establecidos. La gráfica se puede volver a recalcular sin salirse de la ventana volviendo a pulsar el botón

de mostrar resultados y mostrara la gráfica según como estén los parámetros.

La gráfica muestra los ciclos de reloj totales que suman todas las operaciones en cada uno de los ficheros seleccionados, a través de una gráfica de barras, si se han seleccionado menos de 10 ficheros y en caso contrario, a través de un gráfico de puntos. Los nombres de los ficheros en la gráfica son sustituidos por letras en orden alfabético. Y como se ha indicado antes para ver el nombre de cada uno hay que buscar en la celda de texto que fichero está asociado a cada nombre.

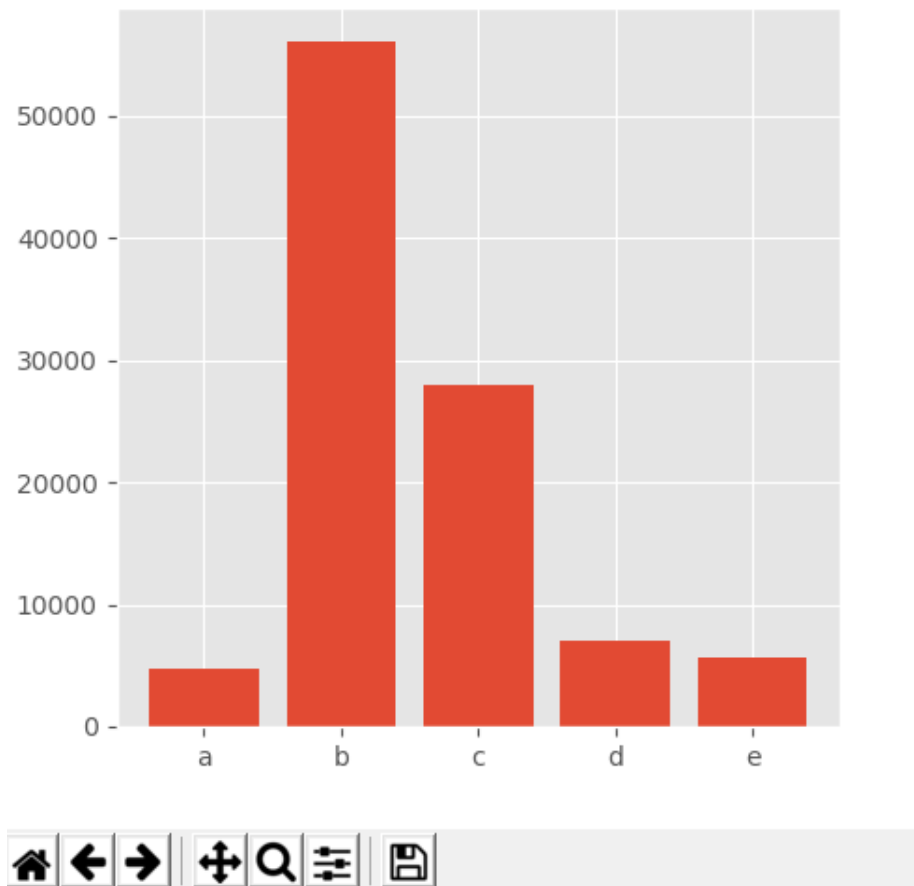


Figura B.11: Gráfico de barras resultante

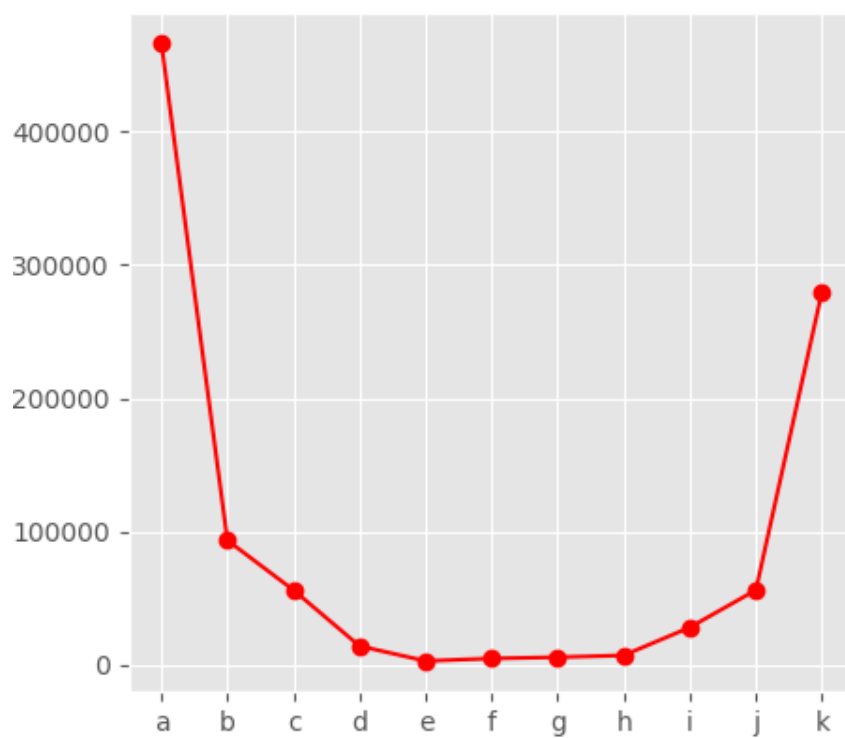


Figura B.12: Gráfico de puntos resultante