



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Herramienta para medir la
eficiencia de código en
Python
Documentación Técnica**



Presentado por Guillermo Calvo Álvarez
en Universidad de Burgos — 3 de julio
de 2019

Tutor: Jesús Enrique Sierra García

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	3
Apéndice B Especificación de Requisitos	7
B.1. Introducción	7
B.2. Objetivos generales	7
B.3. Catalogo de requisitos	7
B.4. Especificación de requisitos	8
Apéndice C Especificación de diseño	13
C.1. Introducción	13
C.2. Diseño de datos	13
C.3. Diseño procedimental	13
C.4. Diseño arquitectónico	14
Apéndice D Documentación técnica de programación	17
D.1. Introducción	17
D.2. Estructura de directorios	17
D.3. Manual del programador	18

D.4. Compilación, instalación y ejecución del proyecto	20
D.5. Pruebas del sistema	21
Apéndice E Documentación de usuario	23
E.1. Introducción	23
E.2. Requisitos de usuarios	23
E.3. Instalación	24
E.4. Manual del usuario	24

Índice de figuras

B.1. Diagrama de casos de uso	12
C.1. Diagrama secuencial de la ejecución de la herramienta	14
D.1. Así luce la ventana de Spyder	19
D.2. Configuración del fichero setup.py	21
E.1. Vista de la ventana principal	25
E.2. Vista de la ventana principal	25
E.3. Mensaje de error	26
E.4. Mensaje de advertencia	26
E.5. Aparición del botón tras la elección de un archivo .py	27
E.6. Ventana donde elegir operaciones	28
E.7. Ejemplo de la Listbox de procesadores	28
E.8. Ejemplo gráfica resultante análisis individual	29
E.9. Ejemplo gráfica resultante recalculada	29
E.10. Mensaje de error por elegir solo un fichero	30
E.11. Gráfico de barras resultante	31
E.12. Gráfico de puntos resultante	32

Índice de tablas

A.1. Coste de personal	4
A.2. Coste del hardware	4
A.3. Coste del software.	4
A.4. Costes Totales	5

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En esta sección se plantea la forma de estimar los costes del proyecto de la manera mas precisa. Los costes se dividen en tres: costes de tiempo, de trabajo y monetarios.

Se determinan los recursos necesarios para llevar este proyecto a un entorno laboral. Aunque hay partes de este proyecto mas fácil de planificar que otras, porque por ejemplo las partes de investigación son algo que pueden prolongarse si no se encuentran los conocimientos deseados.

El plan de proyecto se compone en las siguientes dos partes:

- Planificación temporal: Esta planificación sirve para ajustar los tiempos que tiene que tardar cada parte del proyecto. El tiempo que tarde cada parte puede variar según el nivel de complejidad y la cantidad de tareas que tenga.
- Estudio de viabilidad: Esta planificación es la que determina si el proyecto se podrá trasladar al marco laboral. Muestra los costes monetarios y las licencias a las que este sujeto el proyecto.

A.2. Planificación temporal

En el desarrollo del proyecto se utilizó la metodología Kanban para organizar el flujo de trabajo, combinada con una gestión con la cual el desarrollo del proyecto se dividió en 4 etapas diferentes:

Investigación

La etapa mas temprana del proyecto, aquí es cuando hay que recoger información sobre las herramientas y temas que se hacen uso en la herramienta. A pesar de ser la primera etapa, puntualmente se podría volver a realizar una tarea de este tipo en un desarrollo mas avanzado.

Tareas:

- Conocimientos sobre la eficiencia.
- Información sobre interpretes.
- Búsqueda de un interprete.
- Procesar como funciona el interprete elegido.
- Información sobre herramientas del desarrollo como por ejemplo Spyder, Tkinter o Latex

Desarrollo

Esta es la etapa donde una vez ya se tienen los conocimientos se crea la herramienta, esta puede ser la etapa mas larga junto con la de investigación por los diversos contratiempos que pueden surgir.

Tareas:

- Upgrade del interprete.
- Funciones de llamado a ficheros e interprete.
- Implementación de análisis.
- Fabricación de la interfaz.
- Implementacion de resultados a través de gráficas.

Pruebas

En esta etapa una vez hechas las funcionalidades principales del desarrollo, se trata de hacer comprobaciones para ver los posibles fallos o carencias del diseño y realizar las mejoras o soluciones.

Tareas:

- Comprobar funcionalidades principales.
- Comprobar disposición de la interfaz.
- Realizar soluciones a los problemas que surjan.
- Implementación de mejoras o funcionalidades secundarias.

Documentación

En la etapa final se trata de documentar todo el proceso que se ha realizado a lo largo del proyecto. Esta etapa a pesar de ser la última, debe hacerse alguna tarea previamente a medida que evoluciona la tarea para ir guardando la información que luego se plasmará en la documentación.

Tareas:

- Guardar datos relevantes sobre el progreso.
- Realización de la Memoria.
- Añadir anexos.
- Realización del artículo.

A.3. Estudio de viabilidad

Viabilidad económica

Aquí se plantean los costes y beneficios en el caso de que el proyecto se realizase en un entorno laboral.

Los costes se dividen en 3 tipos:

Coste de personal

Son los costes que supone tener a un desarrollador contratado en el 2019, teniendo en cuenta tanto su salario neto, la cotización a la seguridad social y la retención del IRPF durante una duración de 4 meses.

Concepto	Coste
Salario neto	1000 €
Seguridad social	150 €
IRPF	236 €
Salario bruto	1386 €
Total 4 meses	5544 €

Tabla A.1: Coste de personal

Costes de infraestructura hardware

En el caso de este proyecto el coste en hardware tan solo sería para costear un ordenador donde poder desarrollar y ejecutar la herramienta.

Concepto	Coste	Amortización
Ordenador	600	42 €

Tabla A.2: Coste del hardware

Costes de infraestructura software

Estos costes son los que supone las licencias del software que se necesita para hacer el desarrollo del proyecto. En caso de este proyecto tan solo habría que costear una licencia del sistema operativo.

Concepto	Coste	Amortización
Windows 10 Pro	259	64,25 €

Tabla A.3: Coste del software.

Costes Totales

Concepto	Coste
Personal neto	5544 €
Hardware	42 €
Software	64.25 €
Total	5650.25 €

Tabla A.4: Costes Totales

Como se puede ver en la tabla A4 el total de los costes para hacer el desarrollo de esta herramienta ascendería hasta los 5650.25 €.

Costes Totales

Si la herramienta se comercializase los beneficios dependerían del tipo de distribución que se llevase a cabo y la acogida por parte de los usuarios. El perfil de usuario que se busca con esta herramienta se asemeja al de un desarrollador, por lo tanto podrían comercializarse versiones enfocadas a empresas y otras para particulares.

Viabilidad legal

Se han investigado las licencias de las herramientas utilizadas en el desarrollo para saber la viabilidad del proyecto.

He aquí las licencias:

- Matplotlib: BSD.
- Latex: licencia LPPL Version1.3c.
- Spyder: BSD.
- ByteRun: Por la descripción del autor todo indica que tiene una licencia GNU-3.0.
- Mostrar esa información de una manera agradable y entendible.

Como se ve en el listado anterior ninguna de las licencias es un impedimento para la distribución del la herramienta.

Para la herramienta se opta por una licencia GNU-3.0, ya que esta permite usar, estudiar, compartir y modificar el software siempre que se mantenga la licencia.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En esta sección se van a explicar los diferentes objetivos de la herramienta junto sus respectivos requisitos.

B.2. Objetivos generales

Los objetivos generales de la herramienta son:

- Recoger la elección del usuario sobre el tipo de análisis y los ficheros que desea analizar.
- Identificar los diferente tipos de operaciones al ejecutar los ficheros a través del intérprete.
- Enseñar los resultados de los análisis a través de gráficas.

B.3. Catalogo de requisitos

Ahora se mostraran los requisitos, tanto funcionales como no funcionales, de los objetivos generales de la herramienta:

Requisitos funcionales

- RF-1 Elegir análisis.

- RF-2 Elegir ficheros.
- RF-3 Análisis.
- RF-4 Elegir parámetros.
- RF-5 Muestra resultados.

Requisitos no funcionales

- RNF-1 Fiabilidad: la herramienta debe poder cumplir con sus funcionalidades principales siempre y cuando se cumplan ciertas condiciones.
- RNF-2 Rendimiento: la herramienta debe responder en unos tiempos aceptables para poder hacer un uso práctico.
- RNF-3 Usabilidad: La herramienta debe tener un tiempo de aprendizaje corto utilizando una interfaz ergonómica y un manual de usuario.
- RNF-4 Escalabilidad: La herramienta debe tener un patrón de diseño que facilite su escalabilidad y mantenimiento.
- RNF-5 Modularidad: La herramienta debe ser flexible y permitir ampliar nuevos desarrollos a través de la modularidad
- RNF-6 Procesadores y ficheros: La herramienta necesita tener disponible un fichero .csv con la estructura de un procesador teórico y un fichero .py que analizar.

B.4. Especificación de requisitos

C1: Elegir Análisis

Versión 1.0

Autor: Guillermo Calvo Álvarez

Descripción: Seleccionar el tipo de análisis que se desea realizar para así poder acceder a la elección de ficheros

Precondición: Haber iniciado la herramienta

Secuencia Normal:

1. Elegir ficheros para analizar.
2. Ejecutar el código de los ficheros a través del intérprete.

Postcondicion Tras elegir una opción la actual ventana cambiará a otra.

Excepciones Se cierra la herramienta sin hacer una elección.

Importancia Alta

Comentarios

C2: Elegir Ficheros

Versión 1.0

Autor: Guillermo Calvo Álvarez

Descripción: Seleccionar los ficheros que se deseen analizar a través del explorador de archivos

Precondición: Haber seleccionado el tipo de análisis

Secuencia Normal:

1. Seleccionar el botón de buscar fichero.
2. Ir al directorio donde se encuentren los ficheros deseados a través del explorador.
3. Seleccionar los ficheros.

Postcondición Empezará automáticamente el análisis de los ficheros seleccionados.

Excepciones No elegir ningún fichero en cualquier análisis, elegir mas de un fichero en el análisis individual o elegir uno en el análisis múltiple.

Importancia Alta

Comentarios

C3: Análisis

Versión 1.0

Autor: Guillermo Calvo Álvarez

Descripción: La herramienta automáticamente ejecuta los ficheros seleccionados y guardando las operaciones que encuentra en la matriz de operaciones

Precondición: Haber seleccionado los ficheros que se quieren analizar

Secuencia Normal:

1. La ventana llama al interprete.
2. El interprete ejecuta el ByteCode.

3. Si la instrucción que se encuentra es una operación la guarda en la matriz de operaciones.
4. Al terminar devuelve los valores obtenidos.

Postcondicion Sale un mensaje indicando el final de la ejecución y aparece un nuevo botón en la ventana actual..

Excepciones Si el fichero seleccionado tiene alguna nomenclatura que el interprete no entienda hará que se detenga el proceso.

Importancia Alta

Comentarios

C4: Elegir Parámetros

Versión 1.0

Autor: Guillermo Calvo Álvarez

Descripción: Pulsar el botón analizar fichero o comparar ficheros, esto hace que se muestre una nueva ventana donde se deben elegir los parámetros deseados para mostrar en las gráficas finales

Precondición: Tiene que salir un mensaje confirmando que la ejecución del interprete a sido un éxito

Secuencia Normal:

1. Pulsar el botón analizar fichero o comparar ficheros.
2. Seleccionar a través de una listbox el procesador teórico deseado.
3. Pulsar el botón ver operaciones y seleccionar las operaciones deseadas.

Postcondicion

Excepciones Si el fichero seleccionado tiene alguna nomenclatura que el interprete no entienda hará que se detenga el proceso.

Importancia Alta

Comentarios

C6: Muestra resultados

Versión 1.0

Autor: Guillermo Calvo Álvarez

Descripción: Pulsar el botón mostrar resultados para enseñar el gráfico resultante teniendo en cuenta los parámetros establecidos previamente

Precondición: Tiene que estar ajustados los parametros para que la gráfica muestre la información que se desea

Secuencia Normal:

1. Pulsar el botón mostrar resultados.
2. Mostrar la gráfica resultante.

Postcondicion Queda visualizado en forma de gráfica todos los datos del análisis hecho a los ficheros

Excepciones Si se pulsa el botón sin tener seleccionado ningún procesador no mostrara nada.

Importancia Alta

Comentarios

Actores

El único actor es el usuario final que quiere hacer el análisis, interactúa de forma constante con la herramienta para poder llevar a cabo su función.

Casos de Uso

Diagrama de casos de uso de la herramienta:

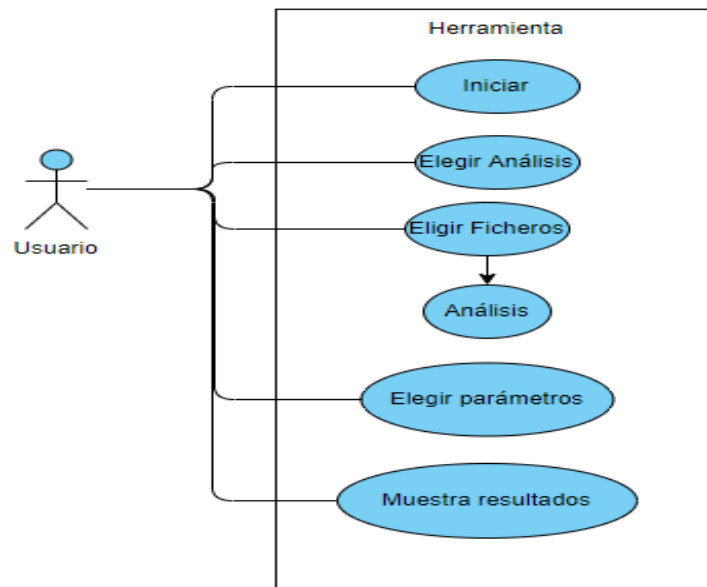


Figura B.1: Diagrama de casos de uso

Apéndice C

Especificación de diseño

C.1. Introducción

En esta sección se explican todos los aspectos relevantes sobre el diseño de la herramienta como el diseño de la arquitectura o los tipos de datos que se utilizan.

C.2. Diseño de datos

Los datos que se utilizan en la herramienta son los siguientes:

- Archivos.py: Se utilizan los códigos hechos en lenguaje python de estos archivos para hacer su análisis.
- Archivos.csv: Se forma una matriz dentro de estos archivos para hacer la ponderación de operaciones.
- ByteCode: Es el código intermedio que surge tras compilar el código de alto nivel de los archivos.py.

C.3. Diseño procedimental

A continuación se verán las interacciones entre los elementos de la herramienta:

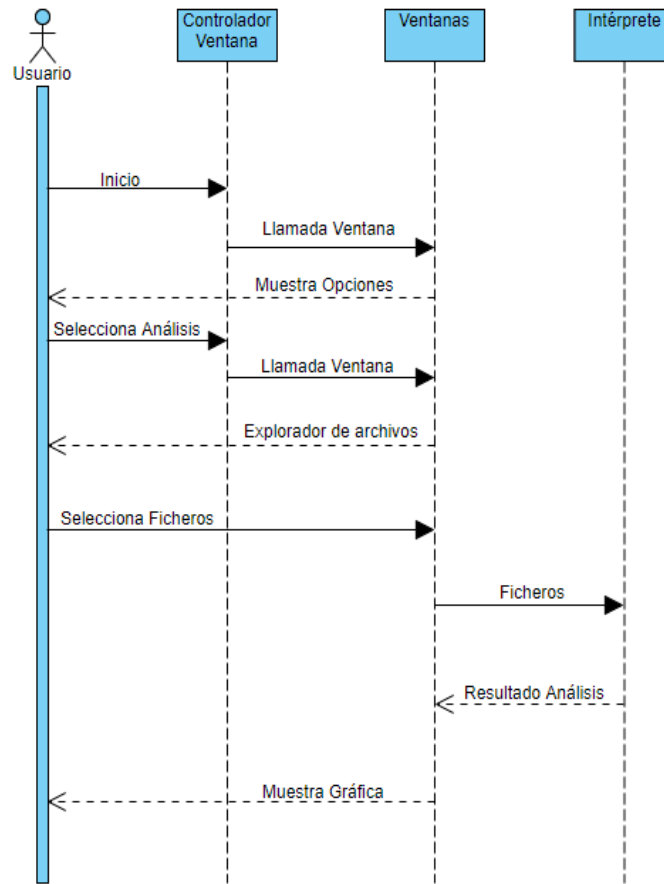


Figura C.1: Diagrama secuencial de la ejecución de la herramienta

Como se puede ver en el diagrama, el usuario tiene una interacción constante con la herramienta en toda su ejecución.

C.4. Diseño arquitectónico

El diseño arquitectónico de la herramienta se basa en un sistema de ventanas que se llaman las unas a la otras y mientras tanto hay un controlador que gestiona todas estas llamadas. También hay ciertas ventanas en concreto que son las encargadas de hacer llamadas al Intérprete cuando es necesario las cuales son:

- VentanaAnálisis.

- VentanaMultiple.

Apéndice D

Documentación técnica de programación

D.1. Introducción

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

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)

D.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 realizó el upgrade, ademas de ser la parte más importante del intérprete.

D.3. Manual del programador

Se empezará 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 que se tengan. 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>

Después de de esto en el símbolo de sistemas hay que ejecutar el comando:

- pip python get-pip.py

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

- pip install matplotlib
- pip install python-tk

IDE

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

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

Una vez descargado e instalado este IDE ya viene listo para poder empezar a 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.

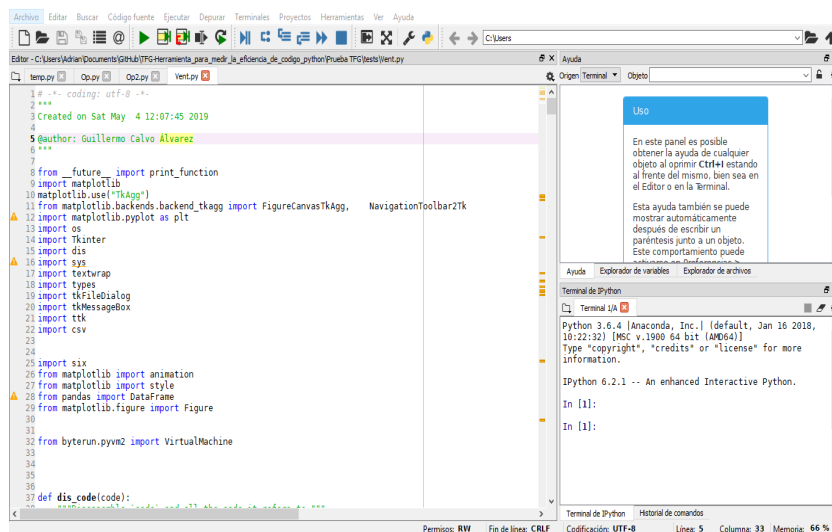


Figura D.1: Así luce la ventana de Spyder

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

D.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 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 que 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 D.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 encontrará 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.

D.5. Pruebas del sistema

Se han realizado dos vídeos, uno haciendo una prueba del análisis individual y otro haciendo el análisis múltiple. A parte también se comentan los contenidos de los archivos .csv

Apéndice E

Documentación de usuario

E.1. Introducción

En esta sección se explica lo necesario para comprender como hacer el funcionamiento de la herramienta a través de unos pasos.

E.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.

E.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.

E.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 Múltiple o salir, aquí dependiendo del tipo de uso que se quiera hacer la herramienta, se elegirá uno u otro.

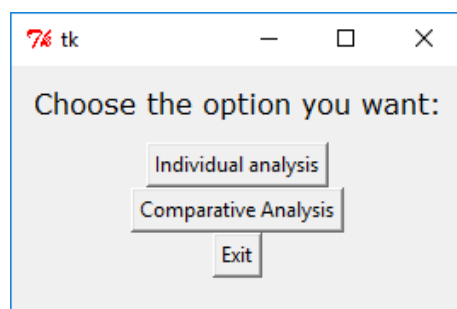


Figura E.1: Vista de la ventana principal

En el caso de querer ver el porcentaje de eficiencia que proporciona cada tipo de operación de un fichero, el análisis individual es la opción adecuada. Por otra parte si lo que se quiere es hacer una comparación de eficiencia entre ficheros, el análisis Múltiple 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.

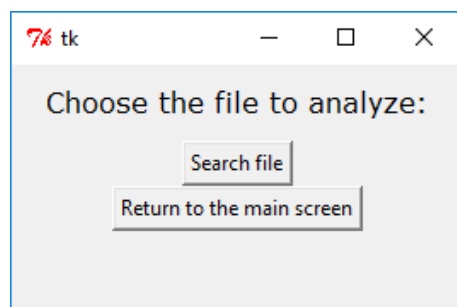


Figura E.2: Vista de la ventana principal

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 válido.

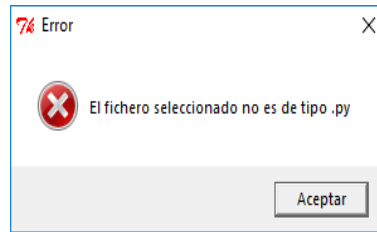


Figura E.3: Mensaje de error

Algo parecido aparecerá si no elegimos ningún fichero, como advertencia saldrá también un mensaje

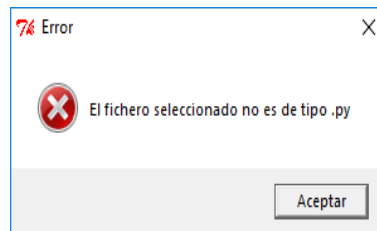


Figura E.4: Mensaje de advertencia

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.

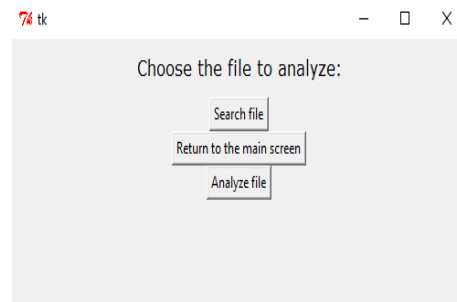


Figura E.5: Aparición del botón tras la elección de un archivo .py

Si pulsamos ese botón, nos mostrará 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:

1. 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.
2. Después hay un botón 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 botón para volver a la vista anterior en la parte de abajo.

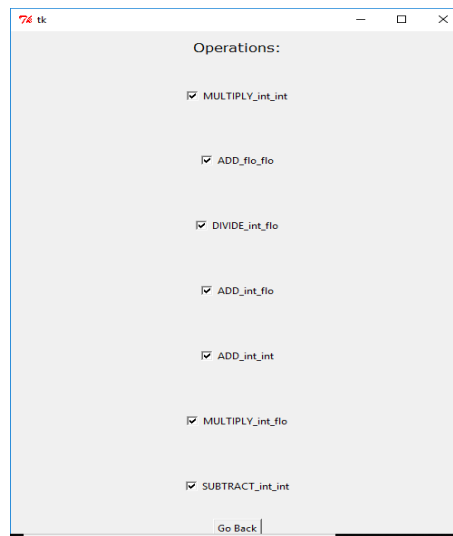


Figura E.6: Ventana donde elegir operaciones

3. 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.

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

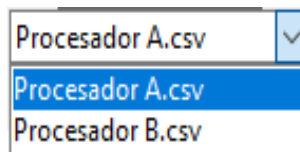


Figura E.7: Ejemplo de la Listbox de procesadores

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

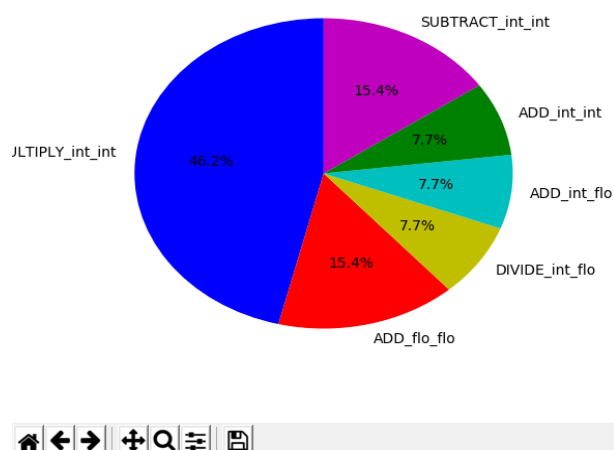


Figura E.8: Ejemplo gráfica resultante análisis individual

- 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.

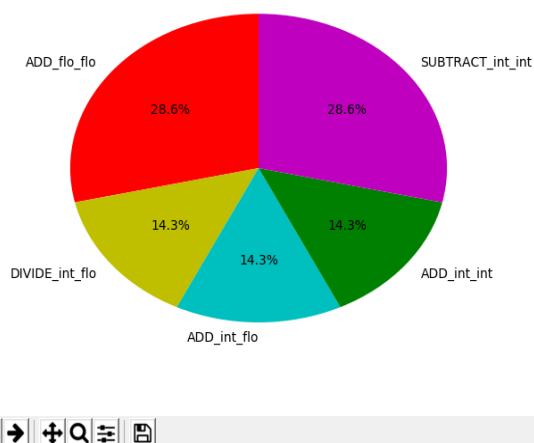


Figura E.9: Ejemplo gráfica resultante recalculada

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 última 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 Múltiple

Pulsando el botón de análisis Múltiple, 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 B2). 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.

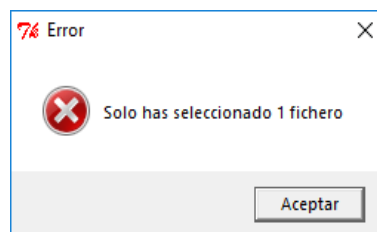


Figura E.10: Mensaje de error por elegir solo un fichero

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 mismo con otros ficheros.

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

Seguido a esto hay un botón que nos lleva a otra ventana donde hay una

serie de checkboxes, pero a diferencia del análisis individual, aquí muestra todas las diferentes operaciones que aparecen en todos los ficheros. Así que puede haber operaciones del checkbox que se encuentren solo en un fichero. Una vez elegidos los parámetros está el botón mostrar resultados que nos muestra la gráfica 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.

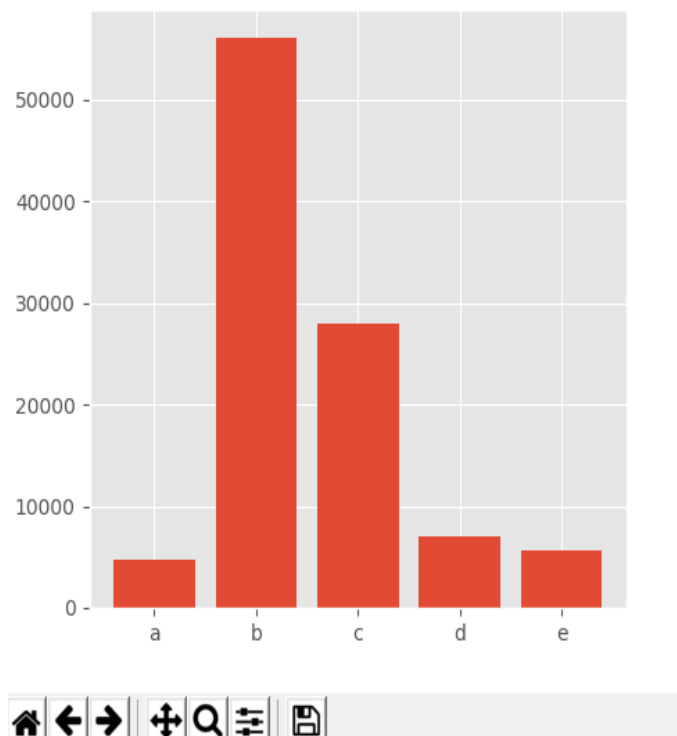


Figura E.11: Gráfico de barras resultante

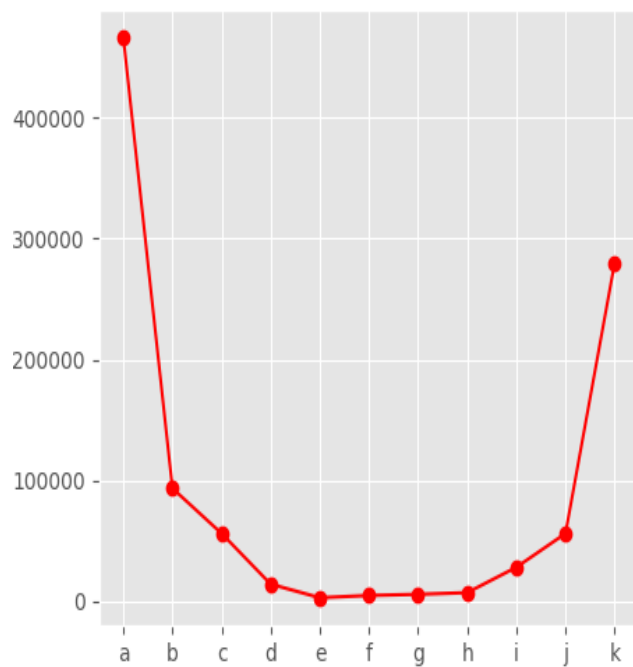


Figura E.12: Gráfico de puntos resultante