



Lenguajes y Autómatas

Juan Pablo Rosas Baldazo

Reporte de proyecto final Traductor Java-Python

Integrantes del equipo

Alfonso Guerrero Contreras # de control 16480211

Marlene Garza Guzmán # de control 16480135

Darcy Zuzet Lona Cruz # de control 16480037

06/12/2018

Introducción

En la programación, escoger el lenguaje correcto es fundamental, este puede facilitar la codificación o hacer muy difícil el desarrollo de un sistema o aplicación.

En este documento se explicará un traductor de lenguaje java, que es orientado a objetos a Python, un lenguaje multipropósito.

Veremos pruebas de velocidad para comprobar el tiempo de ejecución (traducción), así como algo de teoría sobre cómo se usan las herramientas utilizadas.

Tabla de contenido

Introducción.....	2
Descripción.....	4
Abrir un archivo de texto, dividirlo en líneas y guardarlo en una lista.	4
Tabla de simbolos reservados	4
Traductor	4
Función de control	6
Tabla de descripción de funciones	6
Pruebas	6
Pruebas de Alfonso	7
Pruebas de Darcy	8
Pruebas de Marlene	9
Conclusión.....	10
Fuentes	11

Descripción

Abrir un archivo de texto, dividirlo en líneas y guardarlo en una lista.

Primero creamos la función “AbrirArchivoTxt”, donde creamos una variable temporal para guardar lo que contiene el archivo, para ello usamos la función `open(dirección, r)`, donde `dirección` es el directorio en donde esta guardado el archivo y `r` es para especificar lectura.

```
def AbrirArchivoTxt():
```

```
    codigoJava = open(r"C:\Users\Angel\Desktop\java.txt", 'r')
```

El siguiente paso consiste en separar por renglones para crear una lista de ellos que nos ayudara a manipular los datos.

```
    listaDeLineas=codigoJava.readlines()
```

Tabla de simbolos reservados

Mas que una tabla es una lista de las cadenas que identificara el programa y que reemplazara, siempre y cuando exista una equivalencia en lenguaje Python.

```
replace('public static void main(String[] args) {', '')
replace('public static void', 'def')
replace('public class prueba{', '')
replace('{', ':')
replace('int ', '')
replace('}', '')
replace('System.out.println', 'print')
replace('\n', '')
replace(';', '')
replace('\t', '')
```

Traductor

Para e traductor, creamos la función “Traductor”, que recibirá la lista de renglones que guardamos del archivo.txt

```
def Traductor(listaDeLineas):
```

Usamos un ciclo for para recorrer la lista, además guardaremos cada línea en una variable temporal para que sea más fácil de manipular

```
for i in listaDeLineas:
```

```
    linea = i
```

Ahora solo pasaremos la línea por la lista de reemplazos para eliminar o cambiar las palabras reservadas y símbolos.

```
    linea = linea.replace('public static void main(String[]  
        args) {', '')  
    linea = linea.replace('public static void', 'def')  
    linea = linea.replace('public class prueba{', '')  
    linea = linea.replace('{', ':')  
    linea = linea.replace('int ', '')  
    linea = linea.replace('}', '')  
    linea = linea.replace('System.out.println', 'print')  
    linea = linea.replace('\n', '')  
    linea = linea.replace(';','')  
    linea = linea.replace('\t','')
```

Como hemos eliminado todas las identaciones y Python las necesita para una correcta sintaxis, las colocaremos donde hace falta, para eso nos ayudaremos de una bandera

```
    bandera = False
```

La indentación se coloca en el renglón siguiente al de una línea terminada en “:”, por lo que, mientras que la línea no termine en “:” la bandera será falsa, pero si esta lo hace, la bandera se cambiara a verdadera y a la siguiente línea se le agregara la indentación.

```
        if bandera == True:  
            linea = '    ' + linea  
            bandera = False  
        elif ':' in linea:  
            bandera = True
```

Después imprimimos la línea traducida:

```
    print(linea)
```

Función de control

Creamos una función de para llamar a las funciones anteriores

```
def CombortidorDeCodigo():  
    listaDeLineas = AbrirArchivoTxt()  
    Traductor(listaDeLineas)
```

y ejecutamos esta función para iniciar el programa importando la función time para medir el tiempo de ejecución.

```
CombortidorDeCodigo()
```

Tabla de descripción de funciones

Función	Descripción
Open	La función clave para trabajar con archivos.
Open –“ r “	Abre un archivo para leer, error si el archivo no existe
Return	Su principal función es tomar las entradas dadas y devolver un resultado
def	Usada para llamar a una función
Replace	Devuelve una copia de la cadena en la que las ocurrencias de la antigüedad se han reemplazado por nuevas

Pruebas

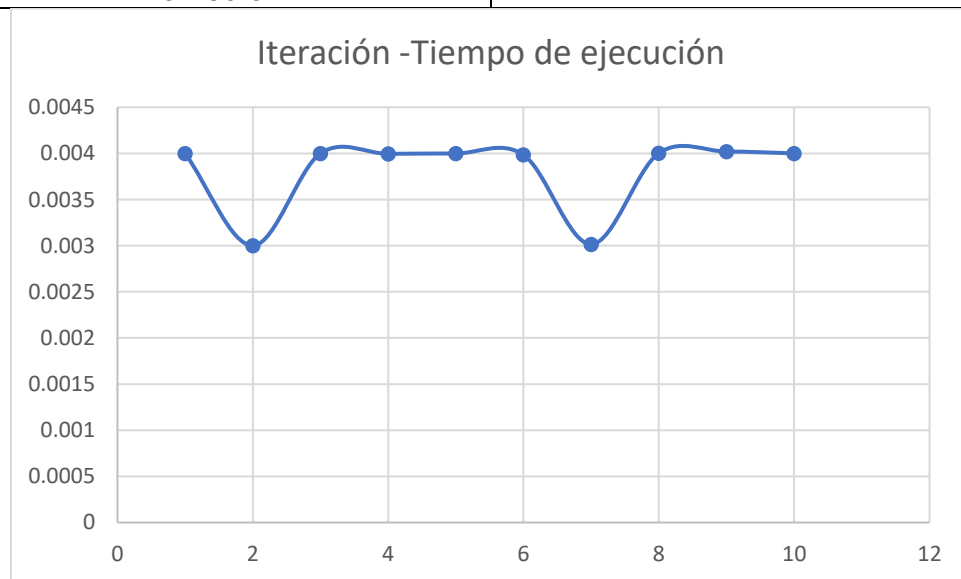
Para probar el tiempo de ejecución de la aplicación, se ejecutará varias veces seguidas con diferentes equipos (los de los 3 miembros del equipo) para observar el desempeño con diferentes recursos de hardware.

Pruebas de Alfonso

El equipo en que se harán las pruebas es un laptop HP Pavilon x360

- Procesador AMD A8-6410 APU 2.00GHz
- Graficos Rendon R5 Graphics
- Windows 10 Home de 64 bits
- 4 GB de memoria RAM

Iteración	Tiempo
1	0.003999471664428711
2	0.003000974655151367
3	0.003999948501586914
4	0.003995180130004883
5	0.003999471664428711
6	0.0039844512939453125
7	0.003015279769897461
8	0.004002094268798828
9	0.0040204524993896484
10	0.004000663757324219
Promedio	



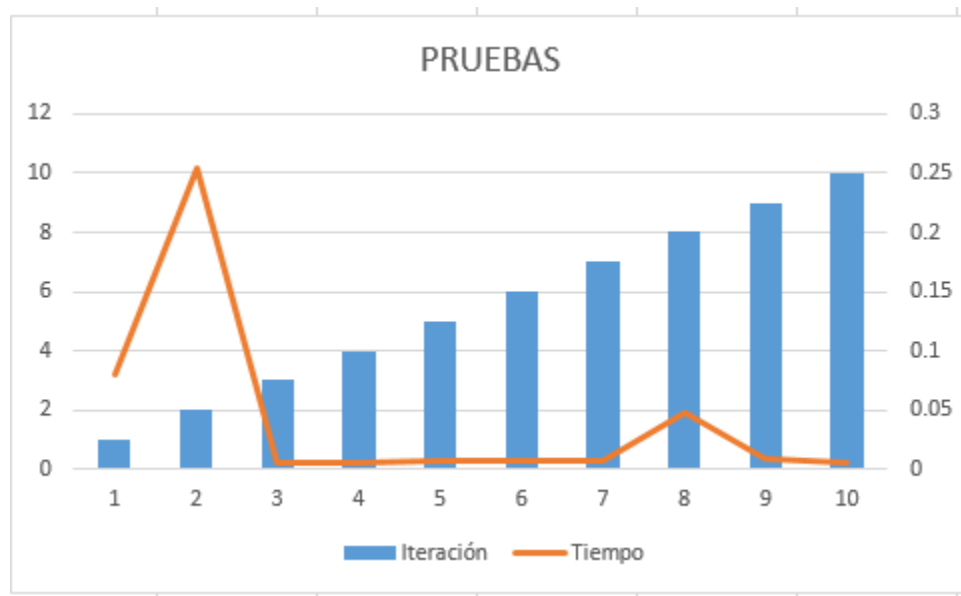
Como podemos observar, la variación en milisegundos es muy baja

Pruebas de Darcy

El equipo en que se harán las pruebas es un laptop ASUS

- Procesador Intel® Celeron® 1.60 GHz
- Windows 10 Home de 64 bits
- 2 GB de memoria RAM

Iteración	Tiempo
1	0.07966470718383789
2	0.25347447395324707
3	0.0059926509857177734
4	0.005988121032714844
5	0.00699162483215332
6	0.006989479064941406
7	0.006987094879150391
8	0.04792165756225586
9	0.007986307144165039
10	0.005988597869873047



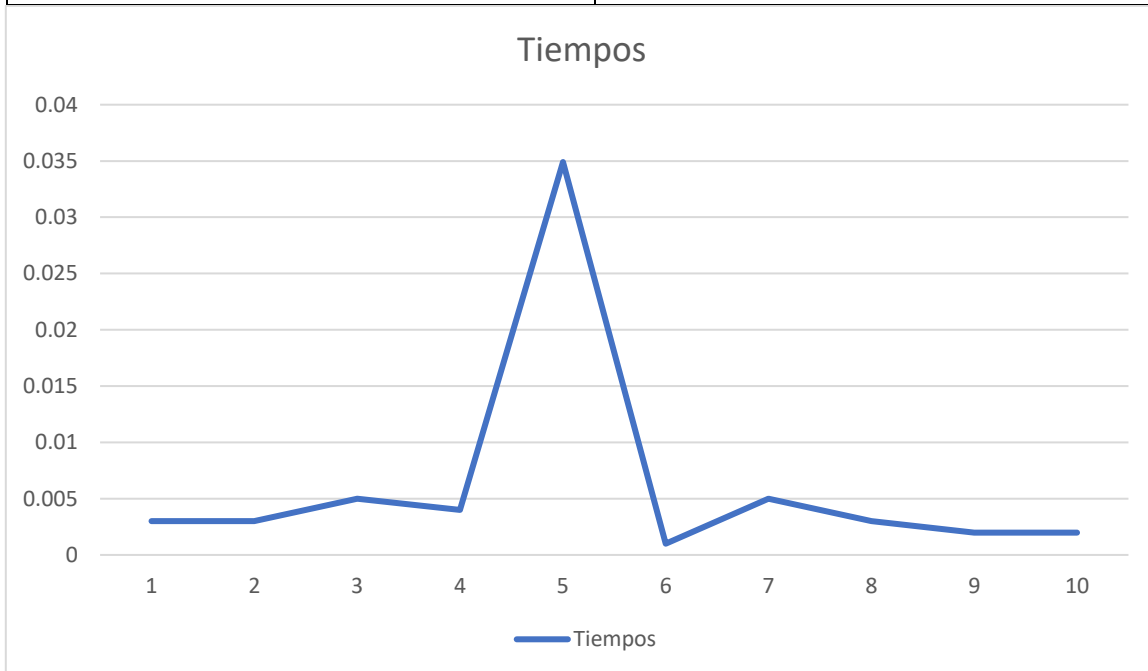
Se observa que entre más se realizan las pruebas tarda más en ejecutarse el programa.

Pruebas de Marlene

El equipo en que se harán las pruebas es un laptop Lenovo ideapad 320

- Procesador Intel Core 5i 7th Generación
- Windows 10 Home de 64 bits
- 12 GB de RAM

Iteración	Tiempo
1	0.002992868423461914
2	0.002992391586303711
3	0.0049877166748046875
4	0.003988742828369141
5	0.034906864166259766
6	0.0009963512420654297
7	0.004987955093383789
8	0.0029916763305664062
9	0.0019948482513427734
10	0.0019948482513427734
Promedio	0.00628343



Como se puede observar, la variación aumento en gran medida cuando llego a la iteración número 5 que se puede considerar como la parte media de las pruebas, esta se mostró con un mayor aumento en el tiempo se estabilizo logrando resultados similares que la primera parte, teniendo pocas variaciones entre ellas.

Conclusión

En conclusión, un traductor de un lenguaje de programación a otro es bastante complejo de diseñar, sin duda requiere de constantes actualizaciones, así como de un dominio y entendimiento de ambos lenguajes.

Sin embargo, como un prototipo y de modo didáctico, es una excelente forma de practicar el pensamiento crítico.

Fuentes

<https://docs.python.org/3/library/>

https://www.w3schools.com/python/python_functions.asp

https://www.tutorialspoint.com/python/string_replace.htm

<https://stackoverflow.com/questions/7129285/what-is-the-purpose-of-the-return-statement>

https://www.w3schools.com/python/python_file_handling.asp

<https://programminghistorian.org/es/lecciones/trabajar-con-archivos-de-texto>