



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)



TEORIA COMPUTACIONAL

NOMBRE Y NÚMERO DE LA PRÁCTICA:

- PRÁCTICA 1. INTRODUCCIÓN A PYTHON

NOMBRE DEL ALUMNO:

- SANTOS MÉNDEZ ULISES JESÚS

NOMBRE DEL MAESTRO:

- JORGE LUIS ROSAS TRIGUEROS

FECHA DE REALIZACIÓN:

- 16/10/2020

FECHA DE ENTREGA:

- 23/10/2020

Marco Teórico

Alfabetos y Lenguajes

Alfabeto:

Conjunto no vacío y finito de símbolos. A estos símbolos también se les suele llamar letras del alfabeto. Se denota por la letra griega Σ . Para denotar alfabetos se tiende a utilizar letras mayúsculas como A, B, C,..., para los símbolos se emplean letras minúsculas.

Por ejemplo:

$$\Sigma^1 = \{a, b, c, \dots, z\}$$

Palabra:

Secuencia finita de símbolos de un alfabeto. Lo correcto es hablar de “palabras definidas sobre un alfabeto”.

Palabra vacía:

Es una palabra que no tiene ningún símbolo y se representa como ε . Esta tiene una longitud de cero.

Longitud de una palabra:

Es el número de símbolos que componen la palabra. Se representa utilizando dos barras ($||$).

Subcadena:

Subsecuencias de símbolos consecutivos de una palabra, a menudo se usan las palabras factor o infijo.

- Si w, x son palabras, se dice que ex es **prefijo** de w , si para alguna cadena z se obtiene que $w=xz$. Un **prefijo propio** es un prefijo que no es igual a la cadena completa.
- Si w, t , son palabras, se dice que t es sufijo de w si para alguna cadena r se obtiene que $w=rt$. Un **sufijo propio** es un sufijo que no es igual a la cadena completa.
- El orden de los símbolos importa en las palabras, debido a que por definición una palabra es una secuencia.

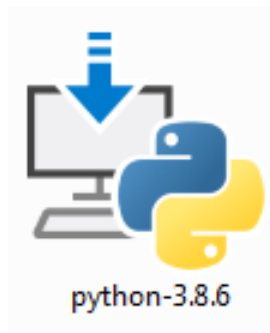
Material y Equipo:

-PC (véase figura 1)¹



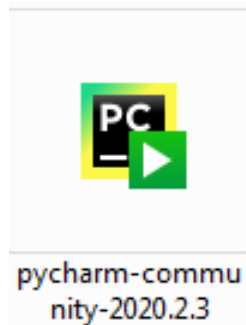
(Figura 1)

-Python 3 (Python 3.8.6) (véase figura. 2)²



(Figura 2)

-IDE (Pycharm) (véase figura 3)³



(Figura 3)

¹ Figura 1: PC o cualquier computadora para el desarrollo de la práctica

² Ejecutable para instalar Python 3.8.6

³ Ejecutable de IDE Pycharm

Desarrollo de la práctica

- 1) Se pidió investigar y responder el siguiente cuestionario por medio de fuentes bibliográficas en internet.

¿Quién desarrolló Python?

Guido Van Rossum.

¿Por qué se llama “Python”?

Por la afición de Van Rossum al grupo Monty Python que son un grupo de comediantes británicos.

Versión actual de Python.

3.9

Explicar el término “Pythonico”

Es todo código que sigue los principios de Python de legibilidad y transparencia. Sigue la serie de máximas, consejos de programación que se supone que hay que seguir en la medida de lo posible.

Diferencia entre lista, tupla y diccionario

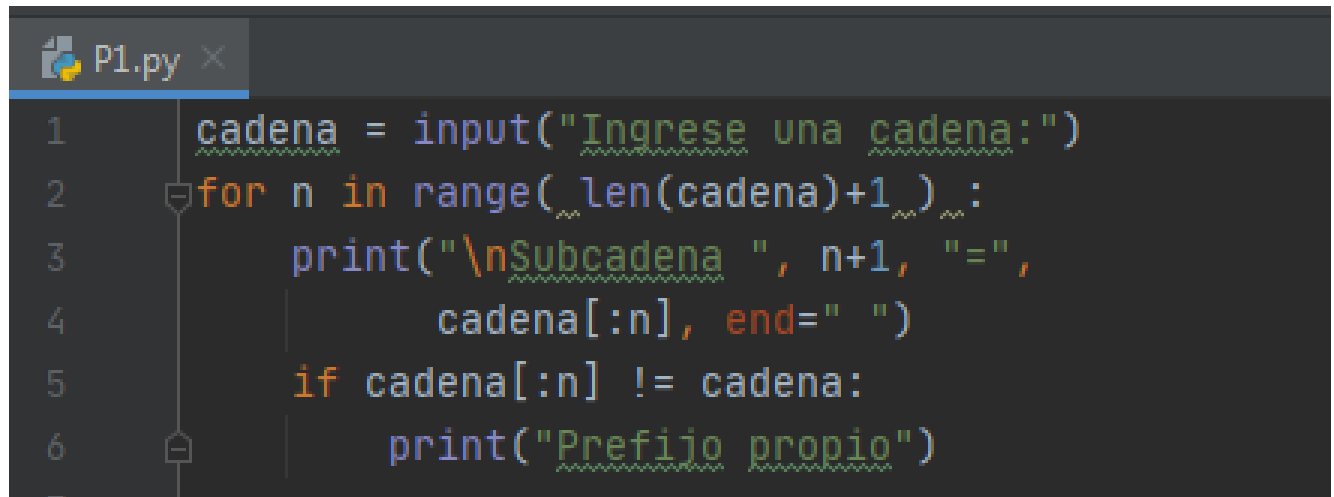
Lista: Es un tipo de colección ordenada. Siendo equivalente a lo que en otros lenguajes se le conoce como arrays.

Tuplas: La diferencia con las listas estriba en que las tuplas no poseen estos mecanismos de modificación a través de funciones tan útiles, además sus valores son inmutables es decir sus valores no se pueden modificar una vez creada y tiene un tamaño fijo, las tuplas son más ligeras que las listas.

Diccionarios: Los diccionarios, también llamados, matrices asociativas, deben su nombre a que son colecciones que relacionan una clave y un valor.

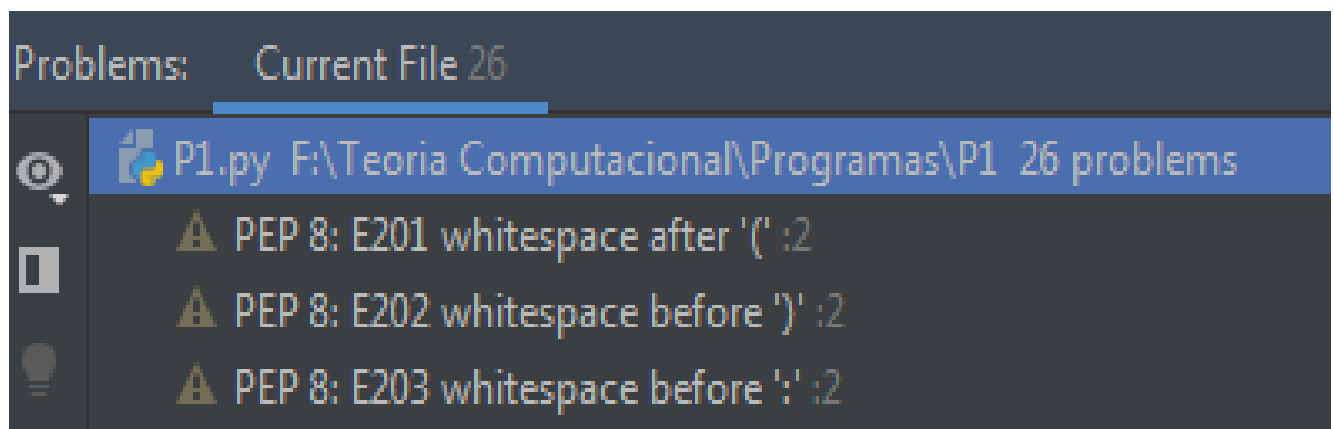
La diferencia principal entre los diccionarios y las listas o las tuplas es que a los valores almacenados en un diccionario se les accede no por su índice, porque de hecho no tienen orden, sino por su clave, utilizando de nuevo el operador []-

- 2) Se realizó un fragmento del programa para la parte de prefijos (Véase Figura 4)⁴, la dificultad principal fue el comprender la sintaxis y algunos comandos de Python (Véase Figura 5)⁵, ya que la IDE marcaba algunas advertencias sobre espacios e indentación, se lograron resolver las dudas buscando la sintaxis de Python.



```
P1.py x
1  cadena = input("Ingrese una cadena:")
2  for n in range(len(cadena)+1):
3      print("\nSubcadena ", n+1, "=",
4          cadena[:n], end=" ")
5      if cadena[:n] != cadena:
6          print("Prefijo propio")
```

(Figura 4)



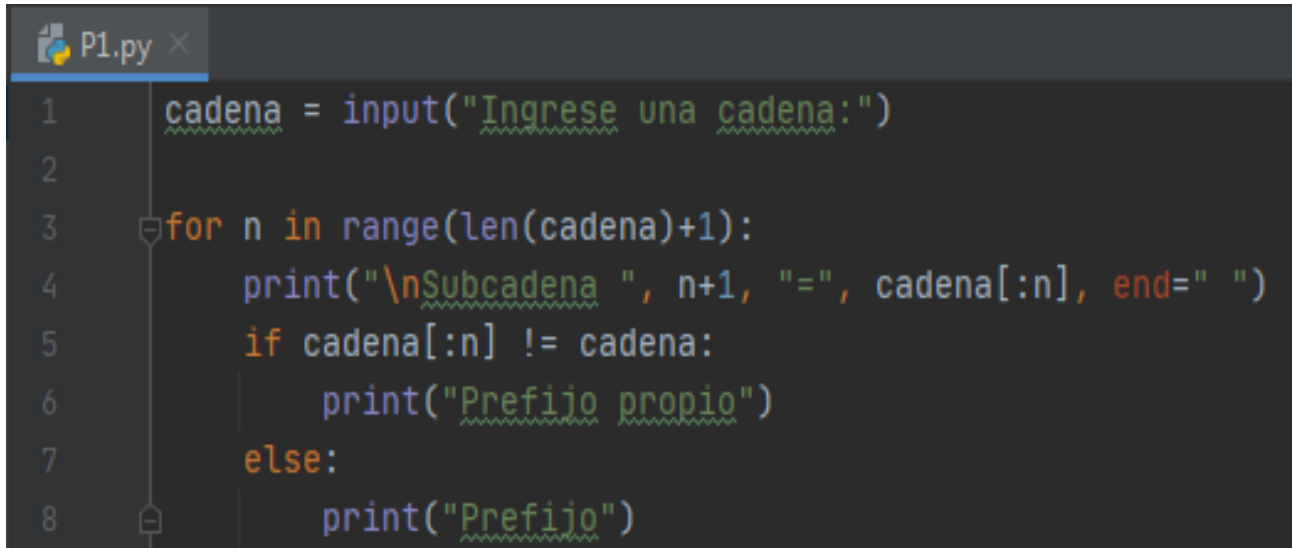
```
Problems: Current File 26
P1.py F:\Teoria Computacional\Programas\P1 26 problems
PEP 8: E201 whitespace after '(' :2
PEP 8: E202 whitespace before ')' :2
PEP 8: E203 whitespace before ':' :2
```

(Figura 5)

⁴ Fragmento código Python para conocer sufijos y sufijos propios de una cadena.

⁵ Errores o advertencias por espacios en la escritura de una función de Python.

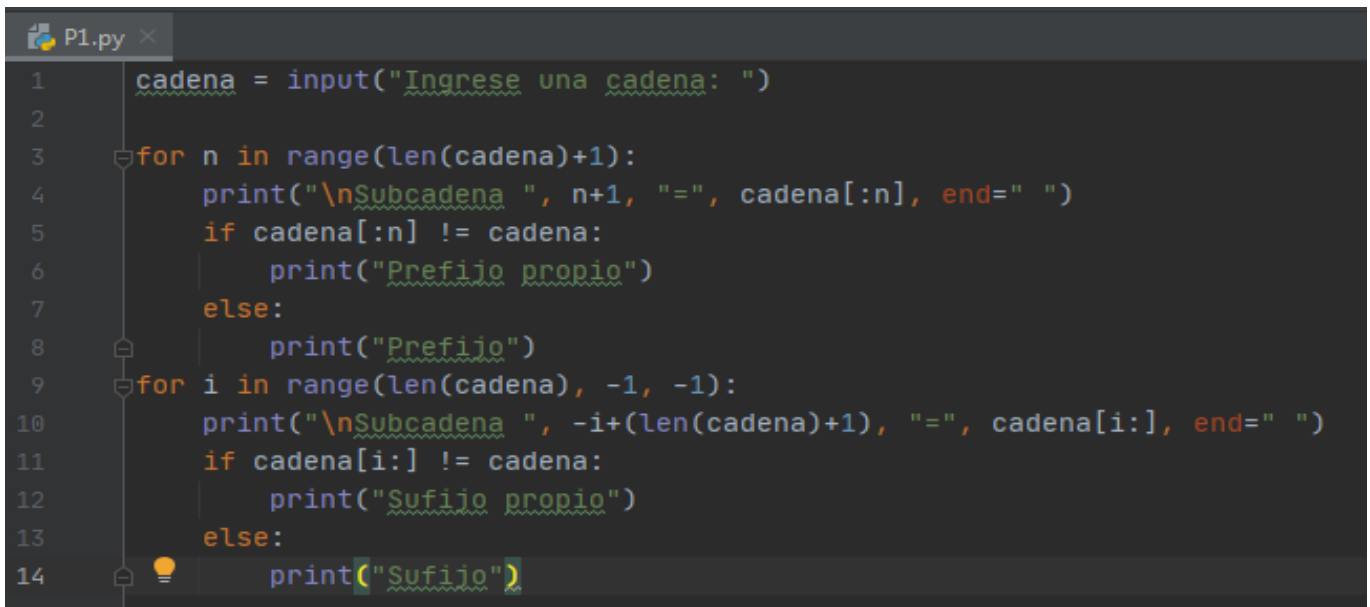
- 3) Después de corregir las advertencias, el código tenía un formato mucho más ordenado, indicando que se debe de llevar un orden y tener cuidado con espacios mal asignados porque puede perjudicar al programa.(Véase Figura 6)⁶.



```
1  cadena = input("Ingrese una cadena:")
2
3  for n in range(len(cadena)+1):
4      print("\nSubcadena ", n+1, "=", cadena[:n], end=" ")
5      if cadena[:n] != cadena:
6          print("Prefijo propio")
7      else:
8          print("Prefijo")
```

(Figura 6)

- 4) Se completó el código anexando la parte dedicada a sufijos de las subcadenas. (Véase Figura 7)⁷.



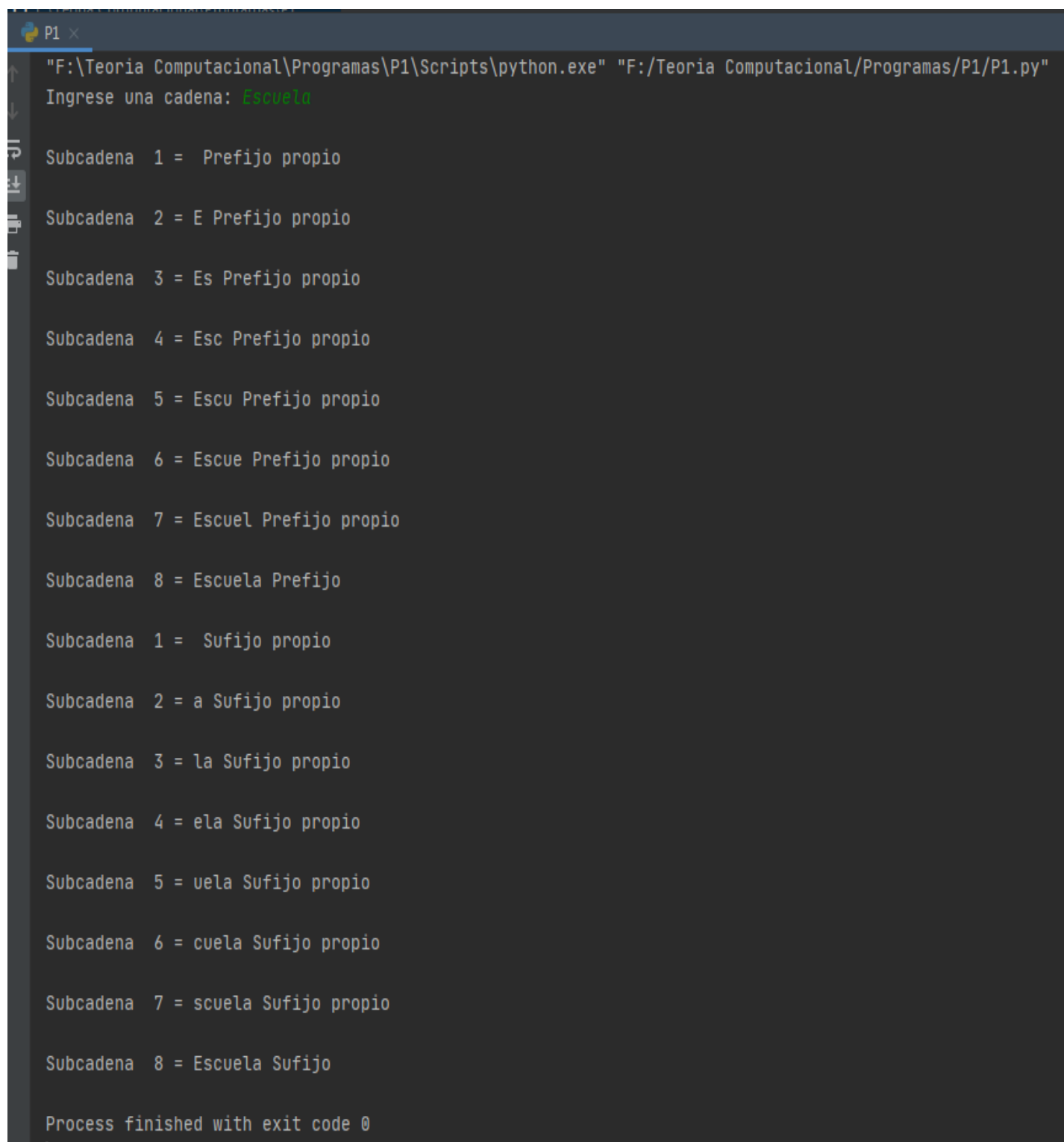
```
1  cadena = input("Ingrese una cadena: ")
2
3  for n in range(len(cadena)+1):
4      print("\nSubcadena ", n+1, "=", cadena[:n], end=" ")
5      if cadena[:n] != cadena:
6          print("Prefijo propio")
7      else:
8          print("Prefijo")
9  for i in range(len(cadena), -1, -1):
10     print("\nSubcadena ", -i+(len(cadena)+1), "=", cadena[i:], end=" ")
11     if cadena[i:] != cadena:
12         print("Sufijo propio")
13     else:
14         print("Sufijo")
```

(Figura 7)

⁶ Código de Python corregido siguiendo la sintaxis correcta.

⁷ Código de Python completo para la práctica donde se incluyen Prefijos, prefijos propios, sufijos y sufijos propios.

- 5) Se compiló y se ejecutó el programa con la cadena propuesta en la práctica (Escuela). (Véase Figura 8)⁸.



```
P1 x
"F:\Teoria Computacional\Programas\P1\Scripts\python.exe" "F:/Teoria Computacional/Programas/P1/P1.py"
Ingrese una cadena: Escuela

Subcadena 1 = Prefijo propio
Subcadena 2 = E Prefijo propio
Subcadena 3 = Es Prefijo propio
Subcadena 4 = Esc Prefijo propio
Subcadena 5 = Escu Prefijo propio
Subcadena 6 = Escue Prefijo propio
Subcadena 7 = Escuel Prefijo propio
Subcadena 8 = Escuela Prefijo
Subcadena 1 = Sufijo propio
Subcadena 2 = a Sufijo propio
Subcadena 3 = la Sufijo propio
Subcadena 4 = ela Sufijo propio
Subcadena 5 = uela Sufijo propio
Subcadena 6 = cuela Sufijo propio
Subcadena 7 = scuela Sufijo propio
Subcadena 8 = Escuela Sufijo
Process finished with exit code 0
```

(Figura 8)

⁸ Ejecución del programa, salida de consola metiendo la cadena Escuela y se muestran todas las subcadenas.

Conclusiones:

En conclusión esta práctica fue necesaria para conocer el IDE y el mismo lenguaje de Python, lo que pude observar en la práctica fue que a diferencia de otros lenguajes de programación Python tiene ciertas reglas que deben seguirse así como una sintaxis más cómoda que la que maneja C o Java, una sugerencia para la mejora de la práctica es la lectura y la visualización de videos para comprender ciertas funciones que Python brinda para que el código que se desarrolla sea "Pythonico".

Bibliografía:

- *Practique la teoría de autómatas y lenguajes formales*, Leonardo Alonso Hernández Rodríguez, 2010.
- *Python para todos*, Raúl González Duque, 2007.
- *Introducción a la programación en Python*, Andrés Marzal, Isabel García, 2010.
- *Teoría de autómatas y lenguajes formales*, Serafín Moral.
- *Teorías de Autómatas y lenguajes formales*, Elena Jurado Málaga, 2008.