INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

TEORÍA COMPUTACIONAL

PRÁCTICA 02

ALUMNO: HERNÁNDEZ CASTELLANOS CÉSAR URIEL

PROFESOR: ROSAS TRIGUEROS JORGE LUIS

FECHA DE REALIZACIÓN: 24 DE AGOSTO DEL 2017

FECHA DE ENTREGA: 27 DE AGOSTO DEL 2017

Marco teórico.

Funciones con cadenas en Python.

Concatenar

Este término significa juntar cadenas de caracteres. El proceso de concatenación se realiza mediante el operador de suma (+). Ten en cuenta que debes marcar explícitamente dónde quieres los espacios en blanco y colocarlos entre comillas.

En este ejemplo, la cadena de caracteres “mensaje1” tiene el contenido “Hola Mundo”

```
mensaje1 = 'Hola' + ' ' + 'Mundo'
print(mensaje1)
-> Hola Mundo
```

Multiplicar

Si quieres varias copias de una cadena de caracteres utiliza el operador de multiplicación (*). En este ejemplo, la cadena de caracteres mensaje2a lleva el contenido “Hola” tres veces, mientras que la cadena de caracteres mensaje2b tiene el contenido “Mundo”. Ordenemos imprimir las dos cadenas.

```
mensaje2a = 'Hola ' * 3
mensaje2b = 'Mundo'
print(mensaje2a + mensaje2b)
-> Hola Hola Hola Mundo
```

Añadir

¿Qué pasa si quieres añadir material de manera sucesiva al final de una cadena de caracteres? El operador especial para ello es compuesto (+=).

```
mensaje3 = 'Hola'
mensaje3 += ' '
mensaje3 += 'Mundo'
print(mensaje3)
-> Hola Mundo
```

Métodos para cadenas de caracteres: buscar, cambiar

En adición a los operadores, Python trae preinstalado docenas de métodos que te permiten hacer cosas con las cadenas de caracteres. Solos o en combinación, los métodos pueden hacer casi todo lo que te imagines con las cadenas de caracteres. Puedes usar como referencia la lista de métodos de cadenas de caracteres (String Methods) en el sitio web de Python, que incluye información de cómo utilizar correctamente cada uno. Para asegurar que

tengas una comprensión básica de métodos para cadenas de caracteres, lo que sigue es una breve descripción de los utilizados más comúnmente.
Extensión

Puedes determinar el número de caracteres en una cadena utilizando el método len. Acuérdate que los espacios en blanco cuentan como un carácter.

```
mensaje4 = 'hola' + ' ' + 'mundo'
print(len(mensaje4))
-> 10
```

Encontrar

Puedes buscar una sub-cadena en una cadena de caracteres utilizando el método find y tu programa te indicará el índice de inicio de la misma. Esto es muy útil para procesos que veremos más adelante. Ten en mente que los índices están numerados de izquierda a derecha y que el número en el que se comienza a contar la posición es el 0, no el 1.

```
mensaje5 = "Hola Mundo"
mensaje5a = mensaje5.find("Mundo")
print(mensaje5a)
-> 5
```

Si la sub-cadena no está presente el programa imprimirá el valor -1.

```
mensaje6 = "Hola Mundo"
mensaje6a = mensaje6.find("ardilla")
print(mensaje6a)
-> -1
```

Minúsculas

A veces es útil convertir una cadena de caracteres a minúsculas. Para ello se utiliza el método lower. Por ejemplo, al uniformar los caracteres permitimos que la computadora reconozca fácilmente que “Algunas Veces” y “algunas veces” son la misma frase.

```
mensaje7 = "HOLA MUNDO"
mensaje7a = mensaje7.lower()
print(mensaje7a)
-> hola mundo
```

Convertir las minúsculas en mayúsculas se logra cambiando .lower() por upper().
Reemplazar

Si necesitas cambiar una sub-cadena de una cadena se puede utilizar el método replace.

```
mensaje8 = "HOLA MUNDO"
```

```
mensaje8a = mensaje7.replace("L", "pizza")
print(mensaje8a)
-> HOpizzaA MUNDO
```

Cortar

Si quieres cortar partes que no quieras del principio o del final de la cadena de caracteres, lo puedes hacer creando una sub-cadena. El mismo tipo de técnica te permite separar una cadena muy larga en componentes más manejables.

```
mensaje9 = "Hola Mundo"
mensaje9a = mensaje9[1:8]
print(mensaje9a)
-> ola Mun
```

Puedes sustituir las variables por números enteros como en este ejemplo:

```
mensaje9 = "Hola Mundo"
startLoc = 2
endLoc = 8
mensaje9b = mensaje9[startLoc: endLoc]
print(mensaje9b)
-> la Mun
```

Esto hace mucho más simple usar este método en conjunción con el método find como en el próximo ejemplo, que busca la letra “d” en los seis primeros caracteres de “Hola Mundo” y correctamente nos dice que no se encuentra ahí (-1). Esta técnica es mucho más eficaz en cadenas largas -documentos enteros, por ejemplo. Observa que la ausencia de un número entero antes de los dos puntos significa que queremos empezar desde el principio de la cadena. Podemos usar la misma técnica para decirle al programa que pase hasta el final de la cadena de caracteres dejando vacío después de los dos puntos. Y recuerda que la posición del índice empieza a contar desde 0, no desde 1.

```
mensaje9 = "Hola Mundo"
print(mensaje9[:5].find("d"))
-> -1
```

Existen más, pero los métodos para cadenas de caracteres anteriores son un buen comienzo. Fíjate que en el ejemplo anterior utilizamos corchetes en vez de paréntesis. Esta diferencia en los símbolos de la sintaxis es muy importante. Los paréntesis en Python son utilizados generalmente para llevar un argumento a una función. De tal manera que cuando vemos algo como:

```
print(len(mensaje7))
```

quiere decir que se lleva la cadena de caracteres “mensaje7” a la función len y entonces enviar el valor resultante de esa función a la declaración print para ser impresa. Una función

puede ser llamada sin un argumento, pero de todas formas tienes que incluir un par de paréntesis vacíos después del nombre de la función. Vimos un ejemplo de ello también.

```
mensaje7 = "Hola Mundo"
mensaje7a = mensaje7.lower()
print(mensaje7a)
-> Hola Mundo
```

Esta declaración le dice a Python que aplique la función lower a la cadena mensaje7 y guarde el valor resultante en la cadena mensaje.

VIM

Vim es una versión mejorada del editor de texto Vi, el cual, fue creado en 1976 por Bill Joy que tomó recursos de ed y ex, dos editores de texto para Unix. Vim, fue presentado en el año 1991 y desde entonces no ha dejado de experimentar mejoras.

Vim, como su antecesor vi, se utiliza desde un terminal en modo texto. Se controla por completo mediante el teclado. Esto es en parte a causa de que Vi fue desarrollado amediados de la década de 1970, cuando los terminales se comunicaban con un ordenador principal (host) mediante una conexión en serie.

Material y equipo.

El material utilizado en la práctica es el siguiente:

Herramientas de software:

* Oracle VM VirtualBox
* Ubuntu 17.04
* Python 3.6.2
* eric6 Web Browser (QtWebKit)

Herramientas de hardware:

* Computadora personal

Desarrollo de la práctica.

En la segunda práctica de laboratorio, se llevó a cabo la resolución de problemas que involucran cadenas de texto, dichas soluciones del problema se implementaron en el lenguaje de programación Python, los problemas son los siguientes:

1. Escriba un programa de Python para calcular la longitud de una cadena.

```
def calcularLongitudDeCadena(cadena):
        return len(cadena)
```

2. Escriba un programa de Python para contar el número de caracteres (frecuencia de caracteres) en una cadena.
Cadena de ejemplo: 'google.com '
Resultado previsto: {'o': 3, 'g': 2, '': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}

```
def contarNumeroDeCaracter(cadena):
        miDiccionario = {}
        for caracter in cadena:
                if miDiccionario.has_key(caracter):
                        miDiccionario[caracter] += 1
                else:
                        miDiccionario[caracter]= 1
        return miDiccionario
```

3. Escriba un programa de Python para obtener una cadena hecha de los 2 primeros y los 2 últimos caracteres de

una determinada cadena. Si la longitud de cadena es menor que 2, devuelve la cadena vacía.
Cadena de ejemplo: 'w3resource'
Resultado Esperado: 'w3ce'
Cadena de ejemplo: 'w3'
Resultado Esperado: 'w3w3'
Cadena de ejemplo: 'w'
Resultado esperado: '' (Cadena vacía)

```
def obtenerNuevaCadena(cadena):
        return cadena[:2]+cadena[-2:]
```

4. Escriba un programa de Python para obtener una cadena de una cadena dada, donde todas las apariciones de su primer carácter se han cambiado a '$', excepto el propio primer carácter.
Cadena de ejemplo: 'regresar'
Resultado esperado: 'regesa'

```
def reemplazarCaracter(cadena):
        cadenanueva = MutableString(cadena.replace(cadena[0],'$'))
        cadaux = MutableString(cadena)
        cadenanueva[0] = cadaux[0]
        return cadenanueva
```

5. Escriba un programa de Python para obtener una sola cadena de dos cadenas dadas, separadas por un espacio, además de intercambiar los dos primeros caracteres de cada cadena.
Cadenas de ejemplo: 'abc', 'xyz'
Resultado Esperado: 'xyc abz'

```
def obtenerInicioDeCadena(miCadena):
        return miCadena[:2]

def obtenerNuevaCadenaCinco(miCadenaUno, miCadenaDos):
        cadena = MutableString(miCadenaUno)
        cadena[0] = ''
        cadena[0] = ''
        cadenaaux = obtenerInicioDeCadena(miCadenaDos) + cadena
        return cadenaaux
```

6. Escribe una función de Python que tome una lista de palabras y devuelva la longitud de la más larga.

```
def obtenerCadenaMasLarga(miLista):
        r = ''
        for i in miLista:
                if len(i)>len(r):
                        r = i
        return r
```

7. Escriba un programa de Python para quitar el carácter de una posición dada de una cadena no vacía.

```
def eliminarCaracter(indice, cadena):
        miCadena = MutableString(cadena)
        miCadena[indice]=''
        return miCadena
```

8. Escriba un programa de Python para cambiar una cadena dada a una nueva cadena donde se intercambiaron los caracteres primero y último.

```
def invertirCaracter(cadena):
        aux = MutableString(cadena)
        aux[0] = cadena[-1]
        aux[-1] = cadena[0]
        return aux
```

9. Escriba un programa de Python para eliminar los caracteres que tienen valores de índice impar de una cadena dada.

```
def eliminarIndiceImpar(cadena):
        cad = MutableString(cadena)
        for i in range(len(cadena)):
                if i%2!=0:
                        cad[i]=''

        return cad
```

10. Escriba un programa de Python para contar las ocurrencias de cada palabra en una oración determinada.

```
def contarOcurrenciaDeOracion(cadena):
        miLista = cadena.upper().split()
        miDiccionario = {}
        for cadena in miLista:
                if miDiccionario.has_key(cadena):
                        miDiccionario[cadena] += 1
                else:
                        miDiccionario[cadena]= 1
        return miDiccionario
```

El problema que presentó la primera parte de la práctica, no fue mas que las dificultades comunes al aprender un nuevo lenguaje de programación, pero nada que no se pueda resolver con un poco de investigación.

La segunda parte de la práctica, el principal problema fue el desconocimiento de algunas palabras presentadas en el tutorial del idioma inglés, en cuanto a la parte técnica no resulto problema alguno.

Diagramas, gráficas y pantallas.

Imagen 1.0 Ejecución de los programas solicitados.

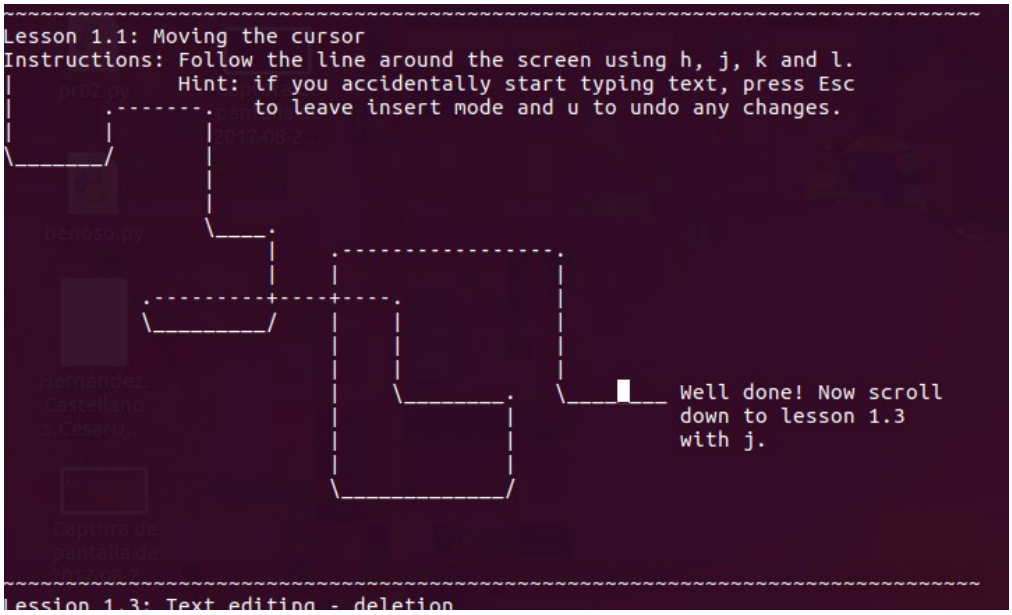

Imagen 1.1 Lección de VIM (Seguimiento de un camino por medio de h,j,k y l)

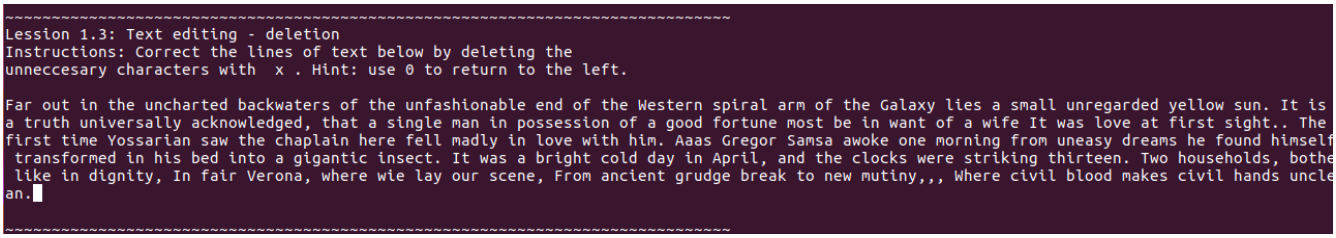

Imagen 1.2 Lección de VIM (Eliminar caracteres innecesarios)

```
Lesson 1.4: Text editing - insertion
Make the pairs of lines match up by inserting the missing text.
Hint: if you're having trouble with the escape key, try Ctrl-[ instead.

Oliver was the victim of a systematic course of treachery and deception.
Oliver was the victim of a systematic course of treachery and deception.

Now, Mr. Bumble was a fat man and a choleric;
Now, Mr. Bumble was a fat man and a choleric;

This was no very great consolation to the child.
This was no very great consolation to the child.

'Please, sir, I want some more.'
'Please, sir, I want some more.'

'We refuse to sanction these indenture,'  said the old gentleman
'We refuse to sanction these indentures,' said the old gentleman

Here the positon  of affairs had not at all improved.
Here the position of affairs had not at all improved.
```

Imagen 1.3 Lección de VIM (Completar las oraciones, por medio del modo insertar)

```
Lesson 1.5: Text editing - appending
Complete the lines by filling in the missing word, then pressing A
to append to the line. Remember, return the cursor to the left with 0.


"I am afraid, my dear Watson, that most of your conclusions were erroneous.
"I am afraid, my dear Watson, that most of your conclusions were erroneous.

I laughed incredulously as Sherlock Holmes leaned back in his settee
I laughed incredulously as Sherlock Holmes leaned back in his settee

"Mr. Holmes, they were the footprints of a gigantic hound!"
"Mr. Holmes, they were the footprints of a gigantic hound!"

"Do you mean that yout wife and you wish to leave?"
"Do you mean that your wife and you wish to leave?"

"I heard it distinctly, and I am sure that it was really the sob of a woman."
"I heard it distinctly, and I am sure that it was really the sob of a woman."
```

Imagen 1.4 Lección de VIM (Completar las oraciones)

Conclusiones y recomendaciones.

Python se caracteriza por su sencillez y elegancia, me resultó sorprendente el como Python requiere de pocas lineas de programación en comparación con otros lenguajes que es necesario escribir una gran cantidad de lineas, en Python es posible realizar funciones en una única linea sin perder nada de claridad.

En cuanto a los nuevos aprendizajes adquiridos sobre VIM, puedo decir que VIM resulta ser una eficiente herramienta de desarrollo debido a su sencillez que esta maneja, lo que se traduce a un desarrollo ágil.

Referencias.

[1]D. Phillips, C. Giridhar and S. Kasampalis, *Python*. Birmingham: Packt Publishing, 2016.

[2]"2.1.5.1 Métodos de las cadenas", *Pyspanishdoc.sourceforge.net*, 2017. [Online]. Available: http://pyspanishdoc.sourceforge.net/lib/string-methods.html. [Accessed: 28- Aug- 2017].

[3]"6.1. Operaciones con cadenas (Algoritmos de Programación con Python)", *Librosweb.es*, 2017. [Online]. Available: http://librosweb.es/libro/algoritmos_python/capitulo_6/operaciones_con_cadenas.html. [Accessed: 28- Aug- 2017].

[4]"Strings functions en Python | luauf.com", *Luauf.com*, 2017. [Online]. Available: https://luauf.com/2008/08/24/strings-functions-en-python/. [Accessed: 28- Aug- 2017].

[5]"Vim (text editor)", *En.wikipedia.org*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Vim_(text_editor). [Accessed: 28- Aug- 2017].