

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

## FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

### CCPG1001 - FUNDAMENTOS DE PROGRAMACIÓN

#### PRIMERA EVALUACIÓN - I TÉRMINO 2016-2017/ Junio 28, 2016

**Nombre:** \_\_\_\_\_ **Matrícula:** \_\_\_\_\_

**Paralelo:** \_\_\_\_\_

COMPROMISO DE HONOR: Al firmar este compromiso, reconozco que el presente examen está diseñado para ser resuelto de manera individual, que puedo usar un lápiz o esferográfico; que sólo puedo comunicarme con la persona responsable de la recepción del examen; y, cualquier instrumento de comunicación que hubiere traído, debo apagarlo y depositarlo en la parte anterior del aula, junto con algún otro material que se encuentre acompañándolo. Además no debo usar calculadora alguna, consultar libros, notas, ni apuntes adicionales a los que se entreguen en esta evaluación. Los temas debo desarrollarlos de manera ordenada.

Firmo el presente compromiso, como constancia de haber leído y aceptado la declaración anterior. "Como estudiante de ESPOL me comprometo a combatir la mediocridad y actuar con honestidad, por eso no copio ni dejo copiar".

\_\_\_\_\_  
Firma

### TEMA 1 (20 PUNTOS)

La **lista** mostrada en el ejemplo contiene los URLs de diferentes sitios Web que han sido visitados.

Los URLs normalmente se repiten y corresponden algunas veces a universidades de Ecuador y otros países.

Note que los URLs no diferencian entre mayúsculas y minúsculas.

Por ejemplo:

`www.espol.edu.ec` y `www.ESPOL.edu.EC`  
corresponden al mismo sitio.

```
lista = ["www.espol.edu.ec",  
        "www.google.com",  
        "www.sri.gob.ec",  
        "www.fiec.espol.edu.ec",  
        "www.uess.edu.ec",  
        "www.FIEC.espol.edu.ec",  
        "www.fict.espol.edu.ec",  
        "www.fcnm.Espol.edu.ec",  
        "www.ucsg.edu.ec",  
        "www.Stanford.edu",  
        "www.harvard.edu",  
        "www.stanford.edu",  
        "www.UCSG.edu.ec",  
        ....  
        "www.google.com.ec",  
        "www.facebook.com",  
        "www.opensource.org",  
        "www.educacionbc.edu.mx"]
```

Escriba un programa en Python que dada una **lista** realice lo siguiente:

a. Muestre los nombres o siglas de las universidades que aparecen en la lista (sin repetir).

Del el ejemplo mostrado, la salida sería:

En la lista aparecen 6 universidades:

- 1.) ESPOL
- 2.) UESS
- 3.) UCSG
- 4.) STANFORD
- 5.) HARVARD
- 6.) EDUCACIONBC

b. Muestre la cantidad y los nombres/siglas de universidades de Ecuador que aparecen en la lista.

Del ejemplo mostrado, la salida sería:

En la lista aparecen 3 universidades de Ecuador

- 1.) ESPOL
- 2.) UESS
- 3.) UCSG

c. Dado un usuario y el nombre o sigla de la universidad, imprima el correo electrónico asignado.

Por ejemplo:

Ingrese el usuario: `rafael.bonilla`  
Ingrese el nombre/sigla de la universidad: `UCSG`  
El correo electrónico del usuario es:  
`rafael.bonilla@ucsg.edu.ec`

## TEMA 2. (30 PUNTOS)

Una empresa registra los **sitios** de internet **visitados** por sus empleados y los **minutos** de navegación en una lista de tamaño  $t$  semejante a la mostrada. Cada registro se almacena usando una cadena con el formato: 'empleado|sitio visitado|minutos':

```
visitados = [ 'maria2|www.facebook.com|160', 'xavi7|www.eluniverso.com|50',  
              'jose15|www.sri.gob.ec|30', 'maria2|www.twitter.com|30',  
              'xavi7|www.inec.gob.ec|10', 'maria2|www.espol.edu.ec|50',  
              'jose15|www.sri.gob.ec|120', 'xavi7|www.sri.gob.ec|20',  
              ..., 'maria2|www.twitter.com|20' ]
```

Para analizar los datos, la empresa dispone de las listas para  $n$  **empleados** y  $p$  sitios de **trabajo**:

```
empleados = ['maria2', 'jose15', ..., 'xavi7']  
trabajo = ['www.espol.edu.ec', 'www.inec.gob.ec', ..., 'www.sri.gob.ec']
```

Con las listas **visitados**, **empleados** y **trabajo** descritas, elabore un programa en Python para obtener los siguientes reportes:

- Genere una lista con los sitios que **no son de trabajo** que han sido visitados por los empleados. La lista obtenida debe contener los  $q$  los elementos **sin repetir**.
- Tabule el **tiempo** total de visita para cada empleado ( $n$ ) y cada sitio visitado ( $m=p+q$ ) en un **arreglo de NumPy** de dos dimensiones que represente lo mostrado en el ejemplo:

tipo	trabajo			no son de trabajo			
Tiempo	www.espol.edu.ec	www.inec.gob.ec	www.sri.gob.ec	www.facebook.com	www.eluniverso.com	...	www.twitter.com
maria2	50	0	0	160	0	...	50
jose15	0	0	150	0	0	...	0
xavi7	0	10	20	0	50	...	0

### TEMA 3. (40 puntos)

Se dispone de un **arreglo de NumPy** que representa el **tiempo**[empleado,sitio] de navegación en Internet de una empresa expresada en un *arreglo* de *nxm*. El arreglo es similar al resultado del tema anterior, además también se dispone de las listas **empleados** y **trabajo**.

La empresa requiere elaborar un reporte de las visitas a internet que incluya los siguientes datos:

- a. Tiempo total de **uso de Internet**
- b. Tiempo total de uso de Internet **por empleado**
- c. Tiempo total de visita **por sitio**
- d. Tiempo total de visita **por** sitio de **trabajo**
- e. Tiempo total de visita **por** sitios que **no son de trabajo**
- f. El nombre del **empleado** que **más tiempo** ha pasado en sitios que **no son de trabajo**
- g. El sitio de **trabajo** que **más tiempo** ha sido usado
- h. Si el proveedor de Internet cobra un valor de *5 centavos* por minuto de visita a los sitios de *trabajo* y el *doble* para los *otros sitios*. Calcule el total a **pagar** en dólares.
- i. ¿Cuántos **empleados** han visitado cada **sitio**?

Desarrolle un programa en Python que calcule los resultados a las preguntas anteriores y muestre por pantalla las respuestas.

#### TEMA 4 (10 PUNTOS)

a. (5 puntos) Dada la secuencia de instrucciones de la derecha, realice la prueba de escritorio para las siguientes variables:

cuantos	donde	lista

```

mensaje = 'No basta saber, se debe también aplicar. No es
suficiente querer, se debe también hacer. Goethe(1749-1832)'

largo=len(mensaje)
cual='be'
cuantos=0
lista=[]

donde=-1
i=0
while (i<largo):
    donde=mensaje[i:].find(cual)
    if (donde>0):
        cuantos=cuantos+1
        i=i+donde+1
        lista.append(donde)
    else:
        i=i+1

print (cuantos)
print (lista)

```

b. ¿Cuál es el resultado luego de ejecutar el siguiente código? Justifique su respuesta mostrando cómo cambian de valor de **i** y **lista2** durante la ejecución del código. (5 puntos)

```

lista = [5,3,2,6,7,34,1,23,5,6]
lista2 = []
for i in range(1,len(lista)):
    if (lista[i-1] <= lista[i]) and (lista[i] >= lista[i+1]):
        lista2.append(lista[i])
print(lista2)

```

- A. [34, 23]
- B. [3, 34, 23]
- C. [3, 6, 7, 34, 23]
- D. IndexError: list index out of range

---//---

#### Cheat Sheet. Funciones y propiedades de referencia en Python.

Librería Numpy para <b>arreglos</b> :	para <b>listas</b> :	para <b>cadenas</b> :
np.array((numRows,numCols),dtype= ) <b>arreglos</b> .shape <b>arreglos</b> .reshape() numpy.sum( <b>arreglos</b> ) numpy.mean( <b>arreglos</b> ) <b>arreglos</b> .sum(axis=1)	<b>listas</b> .append(...) <b>listas</b> .count(...) <b>listas</b> .index(...) <b>listas</b> .pop() elemento in <b>listas</b>	<b>cadenas</b> .islower() <b>cadenas</b> .isupper() <b>cadenas</b> .lower() <b>cadenas</b> .upper() <b>cadenas</b> .split(...) <b>cadenas</b> .find(...) <b>cadenas</b> .count(...)