

Buenos Aires, 10 de julio de 2023

TEMA 4  
Hoja 1 de 8

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	
TEL:	DOCENTE (nombre y apellido):
AULA:	

Duración del examen: 1:30h. Completar con **letra clara, mayúscula e imprenta**. El examen consta de 13 preguntas de opción múltiple. Cada pregunta tiene una y sólo una respuesta correcta.

Las respuestas deben completarse en la siguiente matriz:

Opción	EJ. 1	EJ. 2	EJ. 3	EJ. 4	EJ. 5	EJ. 6	EJ. 7	EJ. 8	EJ. 9	EJ. 10	EJ. 11	EJ. 12	EJ. 13
1									X				
2						X	X					X	
3		X	X								X		
4	X			X	X			X		X			X

**¡ATENCIÓN!** Las respuestas sólo se considerarán válidas si se encuentran en la matriz. De haber diferencias entre la opción seleccionada en el ejercicio y en la matriz, se considerará como válida la de la matriz.

<b>Ejercicio 0104</b>			
<b>Dado el siguiente fragmento de código:</b>			
<pre> if ("a" in nombre and "i" not in nombre):     print(1) elif ("a" in nombre or "i" in nombre):     print(2) else:     print(3) if not (len(nombre) == 5):     print(4) </pre>			
<b>¿Qué valor tiene que tener la variable nombre para que se imprima por pantalla exactamente un 3 y un 4?</b>			
1.	"maria"		1.
2.	"marlen"		2.
3.	"fito"		3.
4.	Ninguna de las opciones cumple con lo pedido	X	4.

<b>Ejercicio 0204</b>			
<b>Dado el siguiente fragmento de código:</b>			
<pre> i = 1 while i &lt;=3:     i+=1     print('vuelta')  for j in range(2,5):     print('vuelta')  while i&lt;2:     i+=1     print('vuelta')                 </pre>			
<b>¿Cuántas veces se imprime la palabra "vuelta"?</b>			
1.	4 (cuatro)		1.
2.	5 (cinco)		2.
3.	6 (seis)	X	3.
4.	7 (siete)		4.

<b>Ejercicio 0304</b>			
<b>Se quiere hacer un programa le pida al usuario ingresar su edad hasta que ingrese una edad válida. Para que una edad sea considerada válida debe cumplir con las siguientes condiciones:</b>			
<ul style="list-style-type: none"> <li>- Debe ser un entero.</li> <li>- Debe ser mayor o igual a 18 y menor o igual a 120.</li> </ul>			
<b>¿Cuál de las siguientes versiones <u>NO</u> cumple el funcionamiento pedido?</b>			
1.	<pre> edad=-1 while edad&lt;18 or edad&gt;120:     try:         edad=int(input('Edad: '))         if edad&lt;18 or edad&gt;120:             print('Seleccionar la edad entre 18 y 120')     except ValueError:         print('Sólo números enteros')                 </pre>		1.
2.	<pre> edad_valida = False edad = input('Edad: ') while not edad_valida:     try:         edad = int(edad)         if edad &gt;= 18 and edad &lt;= 120:             edad_valida = True         else:             print('Seleccionar la edad entre 18 y 120')             edad = input('Edad: ')     except ValueError:         print('Sólo números enteros')         edad = input('Edad: ')                 </pre>		2.
3.	<pre> edad=int(input('Edad: ')) while edad&lt;18 or edad&gt;120:     try:         if edad&lt;18 or edad&gt;120:             print('Seleccionar la edad entre 18 y 120')         edad=int(input('Edad: '))     except ValueError:         print('Sólo números enteros')                 </pre>	X	3.
4.	<pre> edad_valida=False while not edad_valida:     try:         edad=int(input('Edad: '))         if edad in range(18, 121):             edad_valida=True         else:             print('Seleccionar la edad entre 18 y 120')     except ValueError:         print('Sólo números enteros')                 </pre>		4.

**Ejercicio 0404**

**Dadas las siguientes variables:**

```
lista1 = [8, 5, 10]
lista2 = [21, 42, 9]
lista3 = []
```

**¿Cuál de los siguientes fragmentos de código deja a la lista3 con más de 3 elementos luego de ejecutarse?**

1.	<pre>for i in range(3):     if lista1[i] &gt; 10:         lista3.append(lista1[i])     elif lista2[i] &gt; 10:         lista3.append(lista2[i])</pre>		1.
2.	<pre>for i in lista1:     if i &gt;= 10:         lista3.append(i) for i in lista2:     if i &lt;= 10:         lista3.append(i)</pre>		2.
3.	<pre>for i in lista1:     if i &gt; 10:         lista3.append(i)  lista3 += lista2 lista3.remove(8)</pre>		3.
4.	<pre>for i in lista2:     if i &gt; 10:         lista3.append(i)  lista3 += lista1 lista3.remove(8)</pre>	X	4.

**Ejercicio 0504**

**Dado el siguiente programa:**

```
def elige(dicci):
    print('Seleccioná')
    for op in dicci:
        print(op, '.', dicci[op])

    seleccion = input()
    while seleccion in dicci.keys():
        seleccion = input()
    return seleccion

menu = {'0': 'Pasta', '1': 'Pollo', '2': 'Carne', '3': 'Veggie'}
opc = elige(menu)
print('Elegiste', menu[opc])
```

**¿Cuál de las siguientes afirmaciones es verdadera?**

<b>1.</b>	<p><b>Se imprime</b></p> <p>0 . Pasta 1 . Pollo 2 . Carne 3 . Veggie</p> <p><b>y permite que el usuario seleccione sólo opciones válidas. Luego devuelve la opción al programa principal.</b></p>	<b>1.</b>
<b>2.</b>	<p><b>Se imprime</b></p> <p>0 . Pasta 1 . Pollo 2 . Carne 3 . Veggie</p> <p><b>y permite que el usuario seleccione cualquier opción sin validar. Luego la devuelve al programa principal. Si la opción no existe, se imprime 'Elegiste Pasta'.</b></p>	<b>2.</b>
<b>3.</b>	<p><b>Se imprime</b></p> <p>1 . Pasta 2 . Pollo 3 . Carne 4 . Veggie</p> <p><b>y permite que el usuario seleccione cualquier opción. Luego devuelve la opción al programa principal. Si la opción no existe, arroja error.</b></p>	<b>3.</b>
<b>4.</b>	<p><b>Ninguna de las afirmaciones es verdadera</b></p>	X <b>4.</b>

<b>Ejercicio 0604</b>			
<p><b>Dado el siguiente programa:</b></p> <pre>aulas = {     203: ['Matias', 'Tomás', 'Andrea'],     13: ['Bautista', 'Ana'],     116: ['Zoe', 'Miguel'] }</pre> <p># Acá falta un fragmento de código</p> <pre>for aula in aulas:     print(aulas[aula][-1])</pre> <p><b>Cual de los siguientes fragmentos de código se puede poner en lugar del comentario para que por pantalla se imprima:</b></p> <p>Tomás Ana Miguel</p>			
1.	<pre>for aula in aulas:     if aula&lt;200:         aulas[aula].sort(reverse=False)</pre>		1.
2.	<pre>for aula in aulas:     if aula&gt;200:         aulas[aula].sort()</pre>	X	2.
3.	<pre>for aula in aulas:     aulas[aula] = ['Tomás', 'Ana', 'Miguel']</pre>		3.
4.	Ninguna de las opciones cumple con lo pedido		4.

<b>Ejercicio 0704</b>			
<p><b>Dado el siguiente programa</b></p> <pre>lista_1 = [4, 3, 1, 2, 5] lista_2 = [10, -15, 5, 20, 15] suma = 0 for i in range(x, 4):     j = lista_1[i]     suma += lista_2[j] print(suma)</pre> <p><b>¿Cuál tiene que ser el valor de x para que se imprima 10 ?</b></p>			
1.	0		1.
2.	1	X	2.
3.	2		3.
4.	Ninguna de las opciones es correcta		4.

<b>Ejercicio 0804</b>			
<p><b>Una empresa de videojuegos cuenta con un archivo que contiene por cada fila: código de uno de sus juegos, código de jugador, nivel alcanzado. El path del archivo está guardado en la variable file_name.</b></p> <p><b>Por ejemplo:</b>                      567994;29345;Platinado                      567994;29340;Platinado                      587721;29346;Completo                      567990;29346;Platinado</p> <p><b>Se desea ver en pantalla un listado de códigos de juegos, sin repeticiones, que hayan sido platinados al menos una vez.</b>  <b>¿Cuál de los siguientes programas cumple con lo pedido ?</b></p>			
<b>1.</b>	<pre>def platino(line):     return "Platinado" in line  file = open(file_name, "r") lineas = file.readlines() file.close()  platinados = {} for linea in lineas:     if platino(linea):         platinados[linea.split(";")[0]] = linea.split(";")[2]     else:         platinados[linea.split(";")[0]] = linea.split(";")[1]  for jugador in platinados:     print(jugador)</pre>	<b>1.</b>	
<b>2.</b>	<pre>def platino(line):     return "Platinado" in line  file = open(file_name, "r") lineas = file.readlines() file.close()  platinados = [] for linea in lineas:     if platino(linea):         platinados.append(linea.split(";")[1])  for jugador in platinados:     print(jugador)</pre>	<b>2.</b>	
<b>3.</b>	<pre>def platino(line):     return "Platinado" in line  file = open(file_name, "r") lineas = file.readlines() file.close()  platinados = [] for linea in lineas:     if platino(linea) and linea.split(";")[1] not in platinados:         platinados.append(linea.split(";")[1])  for jugador in platinados:     print(jugador)</pre>	<b>3.</b>	
<b>4.</b>	Ninguna de las opciones hace lo solicitado.	<b>X</b>	<b>4.</b>

<b>Ejercicio 0904</b>						
<b>Se tiene el siguiente DataFrame:</b>						
	nombre	artista	año	género	playlist	cant_reproducciones
0	Living la vida loca	Ricky Martin	1999	pop	1	5.0
1	La bestia Pop	PR y RR	1985	rock	2	NaN
2	Automático	María Becerra	2022	reggaetón	1	3.0
3	Stay	Rihanna	2012	pop	3	NaN
4	De música ligera	Soda Stereo	1990	rock	2	10.0

**Si se quiere cambiar la columna 'playlist' para que, en vez de tener números, contenga los títulos:**  
 - "Movidos" para la playlist 1  
 - "Rock nacional" para la 2  
 - "Románticos" para la 3.  
**¿Cuál es la forma correcta de hacer este cambio?**

1.	<code>p = df['playlist'].map({1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'}) df['playlist'] = p</code>	X	1.
2.	<code>df['playlist'] = df['playlist'].apply(map, {1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'})</code>		2.
3.	<code>df[df['playlist'].map({1: 'Movidos', 2: 'Rock nacional', 3: 'Románticos'})]</code>		3.
4.	<code>if (df['playlist'] == 1): df['playlist'] = "Movidos" elif (df['playlist'] == 2): df['playlist'] = "Rock nacional" else: df['playlist'] = "Románticos"</code>		4.

<b>Ejercicio 1004</b>			
<b>Usando el mismo Dataframe del ejercicio 9, se quiere obtener el promedio de año de todas las canciones.</b>			
<b>¿Cuál de las siguientes opciones cumple con lo pedido?</b>			
1.	<code>df[df['año']].mean()</code>		1.
2.	<code>df['año'].count().mean()</code>		2.
3.	<code>df[df['año']].mean().count()</code>		3.
4.	<code>df['año'].mean()</code>	X	4.

<b>Ejercicio 1104</b>			
<b>Considerando el DataFrame del punto 9 y el siguiente código:</b>			
<code>df['cant_reproducciones'].notnull()</code>			
<b>¿Cuál de las siguientes afirmaciones es correcta?</b>			
1.	El código no agrega una columna al dataframe original. Sólo muestra una columna con los valores False (si la canción tiene cant_reproducciones) o True (si no tiene).		1.
2.	El código agrega la columna 'e' al dataframe original. Los valores de la columna van a ser False si la canción tiene cant_reproducciones, True si no tiene.		2.
3.	El código no agrega una columna al dataframe original. Sólo muestra una columna con los valores True (si la canción tiene cant_reproducciones) o False (si no tiene).	X	3.
4.	El código agrega la columna 'e' al dataframe original. Los valores de la columna van a ser True si la canción tiene cant_reproducciones, False si no tiene.		4.

<b>Ejercicio 1204</b>			
<b>¿Cuál de estas afirmaciones es incorrecta?</b>			
1.	La función <code>pie</code> puede o no recibir un arreglo con labels para mostrar qué simboliza cada parte mostrada en el gráfico.		1.
2.	La función <code>scatter</code> recibe un arreglo de tuplas que representan una serie de puntos en el plano cartesiano.	X	2.
3.	La función <code>scatter</code> es muy parecida a la función <code>plot</code> , sólo que <code>scatter</code> no une los puntos con líneas.		3.
4.	Para la función <code>plot</code> , a mayor cantidad de puntos, nuestra línea va a ser más suave.		4.

<b>Ejercicio 1304</b>			
<p><b>Se tiene el siguiente programa</b></p> <pre>def solo_voc(x):     txt=''     for char in x.lower():         if char in 'aeiouáéíóú':             txt+=char     return txt  def pocas(x):     return len(x)&gt;3  a=['Ana', 'Eugenia', 'Milagros', 'Ema', 'Inés'] lista1=list(filter(pocas,a)) lista2=list(map(solo_voc,lista1))</pre> <p><b>¿Qué contenido tendrá lista2 al ejecutarlo?</b></p>			
1.	['ana', 'ema', 'inés']		1.
2.	['aa', 'ea', 'ié']		2.
3.	['eueia', 'iao']		3.
4.	['eueia', 'iao', 'ié']	X	4.