

**PENSAMIENTO COMPUTACIONAL (90)**

**.UBAXXI**

**TEMA 3**

EXAMEN: Julio 2024

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

Duración del examen: 1:30h.

- ✓ Escribir claramente el nombre en todas las páginas.
- ✓ El examen consta de 10 preguntas de opción múltiple.
- ✓ Cada pregunta tiene una y sólo una respuesta correcta.
- ✓ Las respuestas seleccionadas deben consignarse en la siguiente matriz de opciones.
- ✓ **Sólo se considerarán las respuestas anotadas en la matriz.**
- ✓ Las preguntas de la 1 a la 7 inclusive permiten acumular 1 punto (si son correctas), de la 8 a la 10 cada una acumula 2 puntos o 0.
- ✓ La nota final se calcula de acuerdo a la siguiente función:

$$\text{nota}(\text{puntos}) = (\text{puntos} // 7 + 1) \% 2 * (\text{puntos} + 1) // 2 + (\text{puntos} // 7 \% 2) * (\text{puntos} - 3)$$

Función nota(puntos) tabulada:

Puntos	1 o 2	3 o 4	5 o 6	7	8	9	10	11	12	13
Nota	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>

**Matriz de Respuestas**

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 1 Pto	Ej 7 1 Pto	Ej 8 2 Ptos	Ej 9 2 Ptos	Ej 10 2 Ptos	
<b>1</b>											<b>1</b>
<b>2</b>											<b>2</b>
<b>3</b>											<b>3</b>
<b>4</b>											<b>4</b>

**¡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 esta última.

<b>D103 – 1 Pto</b>			
¿Cuál de los siguientes códigos evalúa <b>más</b> condiciones si <b>material='algodón'</b> , <b>largo=2.5</b> y <b>tipo='gruesa'</b> ?			
<b>1</b>	<pre>if tipo=='gruesa':     print('Soporta peso alto') elif tipo=='media' and material=='plástico':     print('Soporta peso moderado') elif largo&lt;1:     print('Soporta peso bajo') elif largo&gt;2 and material=='plástico':     print('Soporta peso medio')</pre>		<b>1</b>
<b>2</b>	<pre>if tipo=='media' or material=='plástico':     print('Soporta peso moderado') elif largo&gt;2 and material=='plástico':     print('Soporta peso medio') elif largo&gt;0.5 and largo&lt;3 and material in ('U25','algodón'):     print('Soporta peso alto') else:     print('Baja resistencia')</pre>		<b>2</b>
<b>3</b>	<pre>if largo&gt;0.5 and largo&lt;3 and material in ('U25','algodón'):     print('Soporta peso alto') elif tipo=='media' or material=='plástico':     print('Soporta peso moderado') elif largo&gt;2 and material=='plástico':     print('Soporta peso medio') elif largo&lt;1:     print('Soporta peso medio') else:     print('Baja resistencia')</pre>		<b>3</b>
<b>4</b>	<pre>if tipo=='gruesa':     print('Soporta peso alto') if tipo=='media' and material=='plástico':     print('Soporta peso moderado') if largo&lt;1:     print('Soporta peso bajo') if largo&gt;2 and material=='plástico':     print('Soporta peso medio') if largo&gt;0 and largo&lt;3 and material in ('acero','U25'):     print('Soporta peso alto')</pre>	<b>X</b>	<b>4</b>

<b>0203 – 1 Pto</b>			
¿Cuál programa <b>no</b> muestra exactamente 3 *?			
<b>1</b>	<pre>i=1 while i&lt;=3:     for letra in 'Mafalda':         print('*')     i+=1</pre>	<b>X</b>	<b>1</b>
<b>2</b>	<pre>for elem in (3,4,6,7,10):     if elem%2==0:         for i in range(1,2):             print('*')</pre>		<b>2</b>
<b>3</b>	<pre>for i in range(9,7):     j=1     while j&lt;5:         k=2         while k&lt;8:             print('*')             k+=1         j+=1     k=3     while k&gt;0:         print('*')         k-=1</pre>		<b>3</b>
<b>4</b>	<pre>for i in range(7,2,-3):     print('*') for j in range(1,3):     if j%2!=0:         print('*')</pre>		<b>4</b>

0303 – 1 Pto					
Dado el siguiente DataFrame <i>stock</i> :					
	Código	Descripción	Existencia	Pr Unit	Unidad
0	25	Silicona transparente	5.0	2300	pomo
1	3	Clavos autoperforantes	230.0	600	gramo
2	14	Lija nro 6	25.0	235	unidad
3	11	Lija nro 3	36.0	235	unidad
4	5	Tornillos Philips ½	NaN	980	unidad
5	107	Tornillo común 3/4	320.0	670	gramo
6	29	Martillo carpintero	NaN	8700	unidad
7	102	Cal viva	1400.0	270	gramo
¿Qué sentencia produce el siguiente resultado?					
	Código	Descripción	Existencia	Pr Unit	Unidad
4	5	Tornillos Philips ½	NaN	980	unidad
6	29	Martillo carpintero	NaN	8700	unidad
1	<code>stock[stock['Existencia'].isnull()]</code>				X 1
2	<code>stock[stock['Unidad']=='gramo']</code>				2
3	<code>stock.head(2)</code>				3
4	<code>stock['Unidad'].value_counts()</code>				4

0403 – 1 Pto			
<p>¿Cuál es la función <b>suprime</b> adecuada para el siguiente programa?</p> <pre>def suprime (...):     -     -     -  #Ppal pal1='camiseta' pal2='SETA' pal3='se' print(suprime(pal2,pal1)) print(suprime(pal3,pal2))</pre> <p>La salida debería ser: <b>cami</b> <b>TA</b></p> <p><b>Nota:</b> El método <b>index()</b> devuelve la posición donde inicia un texto dentro de otro <b>Ej:</b> <b>'Una complejidad Absoluta'.index('pleji') -&gt; 7</b></p> <p>El método <b>pop()</b> recibe una posición y elimina ese elemento en una lista <b>Ej:</b> <b>[5,0,1,2].pop(2) -&gt; [5,0,2]</b></p> <p>El método <b>join()</b> devuelve un texto con los elementos (str) de una lista <b>Ej:</b> <b>'-'.join(['a','e','o']) -&gt; 'a-e-o'</b></p> <p>El método <b>replace()</b> devuelve un texto con el primer argumento remplazado por el segundo <b>Ej:</b> <b>'Living la vida loca'.replace('vi','VU')-&gt; 'LiVUng la VUda loca'</b></p> <p>El método <b>count()</b> devuelve la cantidad de apariciones del argumento en una secuencia <b>Ej:</b> <b>'Living la vida loca'.count('vi')-&gt; 2</b></p>			
1	<pre>def suprime(txt1,txt2):     resp=txt2     if txt2 in txt1:         txt1.replace(txt2, '')     return resp</pre>		1
2	<pre>def suprime():     if txt2.count(txt1)==1:         txt2=txt2.replace(txt1, '')     return txt2</pre>		2
3	<pre>def suprime(txt1,txt2):     resp=txt2     if txt1.lower() in txt2.lower():         lista=list(txt2)         inicia=txt2.lower().index(txt1.lower())         for i in range(len(txt1)):             lista.pop(inicia)         resp=''.join(lista)     return resp</pre>	X	3
4	<pre>def suprime(txt2):     resp=txt2     if txt1.lower() in txt2.lower():         lista=list(txt2)         inicia=txt2.lower().index(txt1.lower())         for i in range(len(txt1)):             lista.pop(inicia)         resp=''.join(lista)</pre>		4

<b>0503 – 1 Pto</b>			
¿Qué muestra el siguiente programa?			
<pre> a=['enero', '', 'marzo', '', 'mayo', '', 'julio',   'agosto', '', 'octubre', '', 'diciembre'] b=['ENE', 'FEB', 'MAR', 'ABR', 'MAY', 'JUN',   'JUL', 'AGO', 'SEP', 'OCT', 'NOV', 'DIC'] cortos=['sep', 'Ter', 'DIC'] for c in cortos:     if c.upper() in b:         posi=b.index(c.upper())         if a[posi]!='':             print(a[posi])         else:             print('No tiene 31 días')     else:         print('No es un mes')                 </pre>			
<b>Nota:</b>			
El método <b>index()</b> devuelve la posición donde se encuentra un elemento dentro de una lista			
<b>Ej:</b>			
<b>[0,0,1,0,2].index(1) -&gt; 2</b>			
<b>1</b>	sep ter dic		<b>1</b>
<b>2</b>	DIC		<b>2</b>
<b>3</b>	No tiene 31 días No es un mes diciembre	<b>X</b>	<b>3</b>
<b>4</b>	ENERO AGOSTO DICIEMBRE		<b>4</b>

0603 – 1 Pto			
<p>En el siguiente programa:</p> <pre>def selecciona (...):     -     -     -  #Ppal palabras=['astucia', 'AZULADO', 'perMEABLE', 'Disponible'] adjetivos=list(filter(selecciona, palabras)) print(adjetivos)</pre> <p>La función <b>selecciona()</b> debe indicar si la palabra recibida es un adjetivo (terminado en ada, ado, able o ible). De ese modo el programa mostrará el siguiente resultado:</p> <pre>['AZULADO', 'perMEABLE', 'Disponible']</pre> <p>¿Cuál versión de <b>selecciona()</b> <b>NO</b> realiza lo esperado?</p>			
1	<pre>def selecciona(pal):     pal=pal.lower()     termina=('ada','ado','able','ible')     adj=False     for t in termina:         if t in pal:             adj=True     return adj</pre>		1
2	<pre>def selecciona(pal):     esta=True     termina=('ada','ado','able','ible')     i=0     while i&lt;len(termina):         if termina[i] in pal:             esta=True             i+=10     return not esta</pre>	X	2
3	<pre>def selecciona(pal):     esta=False     for t in ('ada','ado','able','ible'):         if t in pal.lower():             esta=True     return esta</pre>		3
4	<pre>def selecciona(pal):     esta=0     termina=('ada','ado','able','ible')     i=0     while i&lt;len(termina) and esta==0:         if termina[i] in pal.lower():             esta+=1             i+=1     return esta==1</pre>		4

0703 – 1 Pto			
<p>¿Cuál de los siguientes programas muestra el contenido de una lista desde el principio hasta la primera aparición u ocurrencia de un número? Si el número no está, debe mostrar la lista completa.</p> <p><b>Nota:</b> El método <b>index()</b> devuelve la posición de un elemento en una secuencia. Si el elemento <b>no está</b> produce <b>ValueError</b> y aborta la ejecución</p> <p><b>Ej:</b> <b>[2,2].index(3) -&gt; ValueError</b> <b>[2,2].index(2) -&gt; 0</b></p>			
1	<pre>num=0 lista=[1,1,4,5,6,2] indice=lista.index(num) try:     print(lista) except ValueError:     print(lista[0:indice])</pre>		1
2	<pre>num=0 lista=[1,1,4,5,6,2] try:     indice=lista.index(indice)     print(lista[0:indice]) elif ValueError:     print(lista[0])</pre>		2
3	<pre>num=0 lista=[1,1,4,5,6,2] if:     indice=num.index(lista)     print(lista[0:indice]) else:     print(lista)</pre>		3
4	<pre>num=0 lista=[1,1,4,5,6,2] try:     indice=lista.index(num)     print(lista[0:indice]) except ValueError:     print(lista)</pre>	X	4

0803 – 2 Ptos			
<p>¿Qué muestra el siguiente programa?</p> <pre>def convierte(t):     verbo=t[:len(t)-4]+'ar'     return verbo.upper()  adjetivos=['divisaBle', 'RAZONable', 'compensABLE', 'ALcanzable'] infinitivos=list(map(convierte,adjetivos)) print(infinitivos)</pre> <p><b>Nota:</b> Se puede seleccionar la primera parte de una string haciendo <b>s[:x]</b> que toma los primeros caracteres hasta antes del de la posición <b>x</b></p> <p><b>Ej:</b> <b>'tornavías'[:4] -&gt; 'torn'</b></p>			
1	Divis razon compens alcanz		1
2	'arARarAR'		2
3	['DIVISAR', 'RAZONAR', 'COMPENSAR', 'ALCANZAR']	X	3
4	Compensable-COMPENSAR		4

<b>0903 – 2 Ptos</b>			
<p>Dado el siguiente programa:</p> <pre>def abre (arch,modo) :     return open (arch,modo)  arch1=abre ('carrera1.txt',...) #carrera1.txt listaAl=arch1.readlines() arch1.close() lisParc=[] carre=listaAl[0] for alu in listaAl[1:]:     datos=alu.split(',')     lisParc.append(datos[1]+' '+datos[0]+';'+carre)  arch1=abre ('carrera2.txt',...) #carrera2.txt listaAl=arch1.readlines() arch1.close() carre=listaAl[0] for alu in listaAl[1:]:     datos=alu.split(',')     lisParc.append(datos[1]+' '+datos[0]+';'+carre)  arch2=abre ('parcial.txt',...) #parcial.txt for alu in lisParc:     arch2.write (alu) arch2.close()</pre> <p>Que arma un listado para un parcial con alumnos de 2 carreras</p> <p>Contenido de <b>carrera1.txt</b>:          Química          Analía, Ricardes, 2021          Juliana, Ortesano, 2023          Pedro, Fuentes, 2023</p> <p>Contenido de <b>carrera2.txt</b>:          Industrial          Mariano, Quimsey, 2024          Dante, Drexler, 2022          Ignacio, Artez, 2023          Raquel, Venturini, 2024</p> <p>¿Cuáles deberían ser los modos de apertura para cada archivo de forma que el programa funcione correctamente?</p>			
<b>1</b>	carrera1.txt modo 'r+' carrera2.txt modo 'r+' parcial.txt modo 'w'	<b>X</b>	<b>1</b>
<b>2</b>	carrera1.txt modo 'w' carrera2.txt modo 'w+' parcial.txt modo 'r'		<b>2</b>
<b>3</b>	carrera1.txt modo 'r' carrera2.txt modo 'a' parcial.txt modo 'r'		<b>3</b>
<b>4</b>	carrera1.txt modo 'w+' carrera2.txt modo 'a' parcial.txt modo 'r'		<b>4</b>

1003 – 2 Ptos			
<p>¿Qué salida produce el siguiente programa?</p> <pre> verduras={1:'papa',2:'zanahoria',3:'lechuga',            4:'tomate',5:'cebolla',6:'arvejas'} ensaladas=['mixta','rusa','de la casa','candice'] ingredientes=[[0,3],[0,4],[0,5],[2,1],               [2,4],[2,2],[1,1],[1,2],[1,6]] carta={} for par in ingredientes:     ensa=ensaladas[par[0]]     lleva=verduras[par[1]]     if ensa.upper() in carta:         carta[ensa.upper()].append(lleva)     else:         carta[ensa.upper()]=[lleva] print('Ensaladas') for ensa in carta:     print(ensa,':')     print(*carta[ensa])                     </pre> <p><b>Nota:</b> Si se coloca * delante de una secuencia en un <b>print</b>, se muestran todos los elementos de la misma, uno al lado del otro y separados por espacio.</p> <p><b>Ej:</b> <b>print(*['una','dos','tres']) -&gt; una dos tres</b></p>			
1	Ensaladas mixta de la casa rusa candice		1
2	Ensaladas MIXTA : lechuga tomate cebolla DE LA CASA : papa tomate zanahoria RUSA : papa zanahoria arvejas	X	2
3	Ensaladas lechuga: mixta tomate: mixta de la casa cebolla: mixta papa: de la casa rusa zanahoria: de la casa rusa arvejas: rusa		3
4	Ensaladas candice: []		4



**Talón de Control para el Alumno**

	Ej 1 1 Pto	Ej 2 1 Pto	Ej 3 1 Pto	Ej 4 1 Pto	Ej 5 1 Pto	Ej 6 1 Pto	Ej 7 1 Pto	Ej 8 2 Ptos	Ej 9 2 Ptos	Ej 10 2 Ptos	
1											1
2											2
3											3
4											4