

## PENSAMIENTO COMPUTACIONAL (90)

UBAXXI

## TEMA 5

EXAMEN: SEGUNDO PARCIAL

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:

Puntos	1 o 2	3 o 4	5 o 6	7	8	9	10	11	12	13
<b>Nota</b>	<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>0105 – 1 Pto</b>			
<p>¿Cuál de los siguientes códigos valida adecuadamente que ingrese un nombre finalizado con vocal? Debe detectar el error y garantizar un dato válido</p> <p><b>Ejs válidos: elena, Pedro, SANDRA</b></p> <p><b>Ejs inválidos: javier, Inés, 9ema</b></p> <p><b>Nota:</b> Un índice negativo (-i) aplicado a una secuencia hace referencia al elemento de la posición len(secuencia)-i</p> <p><b>Ej:</b> <b>a=[1,23,56,7]</b> <b>a[-2] -&gt; 56</b></p>			
<b>1</b>	<pre>def vocFin(n):     voc='aeiouáéíóú'     return n[-1] not in voc  #PPal nom=input('Nombre finalizado en vocal: ') while not vocFin(nom) or not nom.isalpha():     nom=input('Nombre finalizado en vocal: ')</pre>		<b>1</b>
<b>2</b>	<pre>def vocFin(n):     voc='aeiouáéíóú'     return n[-1].lower() in voc  #PPal nom=input('Nombre finalizado en vocal: ') while not vocFin(nom) or not nom.isalpha():     nom=input('Nombre finalizado en vocal: ')</pre>	<b>X</b>	<b>2</b>
<b>3</b>	<pre>def vocFin(n):     voc='aeiouáéíóú'     return n[-1] not in voc  #PPal nom=input('Nombre finalizado en vocal: ') while vocFin(nom) and not nom.isalpha():     nom=input('Nombre finalizado en vocal: ')</pre>		<b>3</b>
<b>4</b>	<pre>def vocFin(n):     voc='aeiouáéíóú'     return n[-1].upper() in voc  #PPal nom=input('Nombre finalizado en vocal: ') if vocFin(nom) and nom.isalpha():     print('Ok') else:     print('Ingreso Inválido, hasta pronto!')</pre>		<b>4</b>

<b>0205 – 1 Pto</b>			
¿Cuál es la salida correcta del siguiente programa?			
<pre>def nomDia(d):     dias={1:'domingo',2:'lunes',3:'martes',4:'miércoles',           5:'jueves',6:'viernes',7:'sábado'}     return dias[d]  #PPal dias=[1,5,2,6,7] nombres=list(map(nomDia,dias)) print(nombres)</pre>			
<b>1</b>	[1, 5, 2, 6, 7] [3, 4]		<b>1</b>
<b>2</b>	['SAB', 'VIE', 'JUE', 'MIE', 'MAR', 'LUN', 'DOM']		<b>2</b>
<b>3</b>	'DOMINGO'		<b>3</b>
<b>4</b>	['domingo', 'jueves', 'lunes', 'viernes', 'sábado']	<b>X</b>	<b>4</b>

<b>0305 – 1 Pto</b>			
¿Cuál es la salida del siguiente programa?			
<pre>def mes30(m):     de30dias=(4,6,9,11)     return m in de30dias  #PPal meses=[1,5,6,2,7] de30=list(filter(mes30,meses)) print(de30)</pre>			
<b>1</b>	[4, 6, 9, 11]		<b>1</b>
<b>2</b>	[6]	<b>X</b>	<b>2</b>
<b>3</b>	'abril'		<b>3</b>
<b>4</b>	[]		<b>4</b>

<b>0405 – 1 Pto</b>			
<p>¿Qué contenido tendrá el archivo <b>destacados.txt</b> al finalizar la ejecución del programa si el archivo <b>deportistas.txt</b> tiene el siguiente contenido?</p> <p>Contenido de <b>deportistas.txt</b>:</p> <pre>lionel,messi,1 lucha,aymar,3 julián,álvarez,1 manu,ginobili,5</pre> <p>Programa a ejecutarse:</p> <pre>deportes={1:'Futbol',2:'Patín',3:'Hokey',           4:'Tenis',5:'Basquet',6:'Ciclismo'} arch=open('deportistas.txt','r') lista=arch.readlines() arch.close() nom=[] for dep in lista:     d=dep.split(',')     nom.append(d[1]+' '+d[0]+' '+deportes[int(d[2])].upper()+'\n') arch=open('destacados.txt','w') arch.writelines(nom) arch.close()</pre> <p><b>Nota:</b> El método <b>split()</b> devuelve una lista con las partes de un texto tomando como separador el argumento</p> <p><b>Ej:</b> <code>'yo soy argentina'.split(' ') -&gt; ['yo', 'soy', 'argentina']</code></p>			
1	ginobili manu,BASQUET		1
2	messi lionel,FUTBOL aymar lucha,HOKEY álvarez julián,FUTBOL ginobili manu,BASQUET	X	2
3	lionel,messi,1 lucha,aymar,3 julián,álvarez,1 manu,ginobili,5		3
4	1 3 1 5		4

<b>0505 – 1 Pto</b>			
<p>¿Qué <b>no</b> debería ser <b>estructura</b> para que la siguiente instrucción se ejecute sin problemas?</p> <pre>estructura.append(0)</pre>			
1	Una lista de strings		1
2	Una tupla de enteros	X	2
3	Una lista vacía		3
4	Una lista de enteros		4

<b>D605 – 1 Pto</b>			
<p>Para averiguar la posición o número de orden de un alumno en una lista de nombres. ¿Cuál función impide que aborte la ejecución del mismo si se ingresa el nombre <b>pedro</b>?</p> <pre>def donde (...): - - -</pre> <pre>alumnos=['Andreoli, Julieta','Vargas, Ulises','Fiquet, Paulo'] alum=input('Nombre: ') pos=donde(alumnos,alum) print(alum,pos)</pre> <p><b>Nota:</b> El método <b>index()</b> devuelve la posición de un elemento en una lista. Si el elemento no está, se aborta la ejecución del programa</p> <p><b>Ej:</b> <b>[1,2,3].index(3) -&gt; 2</b></p>			
<b>1</b>	<pre>def donde(alumnos,alum):     indice=alumnos.index(alum)     cartel='está en la posición '+str(indice+1)     return cartel</pre>		<b>1</b>
<b>2</b>	<pre>def donde(alumnos,alum):     try:         indice=alumnos.index(alum)         cartel='está en la posición '+str(indice+1)     except:         cartel='NO ESTÁ EN LA LISTA'     return cartel</pre>	<b>X</b>	<b>2</b>
<b>3</b>	<pre>def donde(alumnos):     if:         indice=alumnos.index(alum)         cartel='está en la posición '+str(indice+1)     elif:         cartel='NO ESTÁ EN LA LISTA'     return cartel</pre>		<b>3</b>
<b>4</b>	<pre>def donde(alumnos,alum):     valido=False     while not valido:         indice=alumnos.index(alum)         if indice in range(len(alumnos)):             cartel='está en la posición '+str(indice+1)             valido=True     return cartel</pre>		<b>4</b>

0705 – 1 Pto																										
Para el DataFrame <b>vue</b> de pandas, que contiene:																										
	fila	asiento	zona	pax																						
0	5	A	4	tripulación																						
1	16	A	2	None																						
2	16	B	2	Marcelo Uriondo																						
3	16	C	2	Delsy Anchorena																						
4	23	D	1	None																						
5	23	B	1	Juana Nazar																						
<b>Nota:</b>																										
<b>None</b> es la constante nula. Si aparece <b>None</b> en algún campo de un DataFrame es porque ese campo está vacío, no tiene contenido (equivale a <b>NaN</b> ).																										
¿Qué contendrá <b>vue1</b> después de la siguiente operación?																										
<code>vue1=vue.loc[1:3,['zona','fila']]</code>																										
<b>1</b>			<table border="1"> <thead> <tr> <th></th> <th>pax</th> <th>zona</th> </tr> </thead> <tbody> <tr><td>0</td><td>tripulación</td><td>4</td></tr> <tr><td>1</td><td>None</td><td>2</td></tr> <tr><td>2</td><td>Marcelo Uriondo</td><td>2</td></tr> <tr><td>3</td><td>Delsy Anchorena</td><td>2</td></tr> <tr><td>4</td><td>None</td><td>1</td></tr> <tr><td>5</td><td>Juana Nazar</td><td>1</td></tr> </tbody> </table>		pax	zona	0	tripulación	4	1	None	2	2	Marcelo Uriondo	2	3	Delsy Anchorena	2	4	None	1	5	Juana Nazar	1		<b>1</b>
	pax	zona																								
0	tripulación	4																								
1	None	2																								
2	Marcelo Uriondo	2																								
3	Delsy Anchorena	2																								
4	None	1																								
5	Juana Nazar	1																								
<b>2</b>		<table border="1"> <thead> <tr> <th>fila</th> <th>asiento</th> <th>zona</th> <th>pax</th> </tr> </thead> <tbody> <tr><td>0</td><td>5</td><td>A</td><td>4</td></tr> <tr><td>1</td><td>16</td><td>A</td><td>2</td></tr> </tbody> </table>	fila	asiento	zona	pax	0	5	A	4	1	16	A	2		<b>2</b>										
fila	asiento	zona	pax																							
0	5	A	4																							
1	16	A	2																							
<b>3</b>		<table border="1"> <thead> <tr> <th>fila</th> <th>asiento</th> <th>zona</th> <th>pax</th> </tr> </thead> <tbody> <tr><td>0</td><td>5</td><td>A</td><td>4</td></tr> </tbody> </table>	fila	asiento	zona	pax	0	5	A	4		<b>3</b>														
fila	asiento	zona	pax																							
0	5	A	4																							
<b>4</b>		<table border="1"> <thead> <tr> <th>zona</th> <th>fila</th> </tr> </thead> <tbody> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>2</td></tr> </tbody> </table>	zona	fila	1	2	2	2	3	2	<b>X</b>	<b>4</b>														
zona	fila																									
1	2																									
2	2																									
3	2																									

0805 – 2 Ptos					
¿Cuáles modos de apertura deben emplearse con los archivos <b>datos.txt</b> y <b>otro.txt</b> en el siguiente programa para que no salte error?					
<pre> arch=open('datos.txt',...) filas=arch.readlines() for fil in filas:     print(fil) arch.close()  arch=open('otro.txt',...) linea=arch.readline() print(linea) arch.write('Fin\n') arch.close()                     </pre>					
<b>1</b>	open() de datos.txt	'w+'	open() de otro.txt	'a'	<b>1</b>
<b>2</b>	open() de datos.txt	'a'	open() de otro.txt	'r'	<b>2</b>
<b>3</b>	open() de datos.txt	'r+'	open() de otro.txt	'r+'	<b>X</b> <b>3</b>
<b>4</b>	open() de datos.txt	'a'	open() de otro.txt	'w'	<b>4</b>

<b>0905 – 2 Ptos</b>			
Para el DataFrame <b>vue</b> de pandas, que contiene:			
	<code>fila</code>	<code>asiento</code>	<code>zona</code> <code>pax</code>
0	5	A	4 tripulación
1	16	A	2 None
2	16	B	2 Marcelo Uriondo
3	16	C	2 Delsy Anchorena
4	23	D	1 None
5	23	B	1 Juana Nazar
<b>Nota:</b>			
<b>None</b> es la constante nula. Si aparece <b>None</b> en algún campo de un DataFrame es porque ese campo está vacío, no tiene contenido (equivale a <b>NaN</b> ).			
¿Qué operación produce el siguiente resultado?			
	<code>fila</code>	<code>asiento</code>	<code>zona</code> <code>pax</code>
1	16	A	2 None
4	23	D	1 None
<b>1</b>	<code>vue.groupby('fila')['asiento'].max()</code>		<b>1</b>
<b>2</b>	<code>vue[['pax','asiento','fila']]</code>		<b>2</b>
<b>3</b>	<code>vue[vue['pax'].isnull()]</code>		<b>X 3</b>
<b>4</b>	<code>vue[['pax','zona']]</code>		<b>4</b>

<b>1005 – 2 Ptos</b>			
En el siguiente programa:			
<pre> multas={'AC 335 FQ':[],'AD 358 DD':[15644,33207],         'AB 124 EG':[],'AB 358 FH':[89000]} pat=input('Ingrese patente: ').upper() if pat in multas:     ..... # línea a completar      print('El dominio',pat,'tiene \$',total,           'en concepto de deuda por infracciones') </pre>			
Que calcula el importe total de multas adeudadas de una patente. ¿Cuál debería ser la línea faltante en el código?			
La salida final, para un ingreso <b>ad 358 dd</b> debería ser:			
El dominio AD 358 DD tiene \$ 48851 en concepto de deuda por infracciones			
<b>1</b>	<code>total=multas[pat][0]</code>		<b>1</b>
<b>2</b>	<code>total= 0</code>		<b>2</b>
<b>3</b>	<code>total=sum(multas[pat])</code>		<b>X 3</b>
<b>4</b>	<code>total=multas[1]+multas[2]</code>		<b>4</b>

**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	
<b>1</b>											<b>1</b>
<b>2</b>											<b>2</b>
<b>3</b>											<b>3</b>
<b>4</b>											<b>4</b>