

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
Nota	1	2	3	4	5	6	7	8	9	10

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	
1											1
2											2
3											3
4											4

¡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.

0105 – 1 Pto			
<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>Ejs válidos: elena, Pedro, SANDRA</p> <p>Ejs inválidos: javier, Inés, 9ema</p> <p>Nota: Un índice negativo (-i) aplicado a una secuencia hace referencia al elemento de la posición len(secuencia)-i</p> <p>Ej: a=[1,23,56,7] a[-2] -> 56</p>			
1	<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>		1
2	<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>	X	2
3	<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>		3
4	<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>		4

0205 – 1 Pto

¿Cuál es la salida correcta del siguiente programa?

```
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)
```

1	[1, 5, 2, 6, 7] [3, 4]		1
2	['SAB', 'VIE', 'JUE', 'MIE', 'MAR', 'LUN', 'DOM']		2
3	'DOMINGO'		3
4	['domingo', 'jueves', 'lunes', 'viernes', 'sábado']	X	4

0305 – 1 Pto

¿Cuál es la salida del siguiente programa?

```
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)
```

1	[4, 6, 9, 11]		1
2	[6]	X	2
3	'abril'		3
4	[]		4

0405 – 1 Pto			
<p>¿Qué contenido tendrá el archivo destacados.txt al finalizar la ejecución del programa si el archivo deportistas.txt tiene el siguiente contenido?</p> <p>Contenido de deportistas.txt:</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>Nota: El método split() devuelve una lista con las partes de un texto tomando como separador el argumento</p> <p>Ej: <code>'yo soy argentina'.split(' ') -> ['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

0505 – 1 Pto			
<p>¿Qué no debería ser estructura 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

D605 – 1 Pto			
<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 pedro?</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>Nota: El método index() devuelve la posición de un elemento en una lista. Si el elemento no está, se aborta la ejecución del programa</p> <p>Ej: [1,2,3].index(3) -> 2</p>			
1	<pre>def donde(alumnos,alum): indice=alumnos.index(alum) cartel='está en la posición '+str(indice+1) return cartel</pre>		1
2	<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>	X	2
3	<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>		3
4	<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>		4

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

0805 – 2 Ptos					
¿Cuáles modos de apertura deben emplearse con los archivos datos.txt y otro.txt 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>					
1	open() de datos.txt	'w+'	open() de otro.txt	'a'	1
2	open() de datos.txt	'a'	open() de otro.txt	'r'	2
3	open() de datos.txt	'r+'	open() de otro.txt	'r+'	X 3
4	open() de datos.txt	'a'	open() de otro.txt	'w'	4

0905 – 2 Ptos			
Para el DataFrame vue 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
Nota:			
None es la constante nula. Si aparece None en algún campo de un DataFrame es porque ese campo está vacío, no tiene contenido (equivale a NaN).			
¿Qué operación produce el siguiente resultado?			
	fila	asiento	zona pax
1	16	A	2 None
4	23	D	1 None
1	vue.groupby('fila')['asiento'].max()		1
2	vue[['pax','asiento','fila']]		2
3	vue[vue['pax'].isnull()]		X 3
4	vue[['pax','zona']]		4

1005 – 2 Ptos			
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 ad 358 dd debería ser:			
El dominio AD 358 DD tiene \$ 48851 en concepto de deuda por infracciones			
1	total=multas[pat][0]		1
2	total= 0		2
3	total=sum(multas[pat])		X 3
4	total=multas[1]+multas[2]		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