

## PENSAMIENTO COMPUTACIONAL (90)

UBAXXI

## TEMA 3

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>0103 – 1 Pto</b>		
<b>0203 – 1 Pto</b>		
¿Cuál de los siguientes códigos valida adecuadamente que se ingrese un nombre sin tilde?		
Debe detectarse la correcta del siguiente programa?		
<p>Ejs válidos: <b>Ursula, Inés, Gema</b>                  Ejs inválidos: <b>Pedro, JUAN</b></p> <pre> def nomMes(mes):     meses=[1:'enero',2:'febrero',3:'marzo',4:'abril',5:'mayo',6:'junio',7:'julio',8:'agosto',9:'septiembre',10:'octubre',11:'noviembre',12:'diciembre']     def tilde(n):         return meses[meses.index(n)].upper()         voc='AEIOU'         marca=False     #PPal     i=0     meses=[10,5,2,6,7,9]     while not marca and i&lt;len(meses):         nombres=list(map(tilde,meses[i:i+1]))         if voc[i] in nombres:             print(nombres)             marca=True         i+=1     Nota: return marca                 </pre>		
	<b>X</b>	<b>1</b>
El método <b>upper()</b> devuelve el texto todo en mayúsculas		
Ej: nom=input('Nombre sin acentos: ') a=tilde(nom) or not nom.isalpha(): a.upper()>'HOLA' Nombre sin acentos: ')		
<b>1</b> def tilde(n): voc='AEIOU' for letra in voc: if letra in n.lower(): marca=True return marca		
<b>1</b>	<b>X</b>	<b>1</b>
<b>2</b> ['OCTUBRE', 'MAYO', 'FEBRERO', 'JUNIO', 'JULIO', 'SEPTIEMBRE'] for letra in voc: if letra in n.lower(): marca=True return marca		
<b>2</b>		<b>2</b>
<b>3</b> [9, 7, 6, 5, 10] else: marca=False return marca		
<b>3</b>		<b>3</b>
<b>4</b> #PPal nom=input('Nombre sin acentos: ') while tilde(nom) or not nom.isalpha(): nom=input('Nombre sin acentos: ')		
<b>3</b> def tilde(n): voc='ÁÉÍÓÚ' marca=False for letra in n.lower(): if letra in voc: marca=True return True		
<b>4</b> #PPal nom=input('Nombre sin acentos: ') while tilde(nom) or not nom.isalpha(): nom=input('Nombre sin acentos: ')		
<b>4</b> def tilde(n): voc='ÁÉÍÓÚ' marca=False if n.lower() in voc: marca=True return marca		
<b>4</b> #PPal nom=input('Nombre sin acentos: ') if nom.isalpha() or tilde(nom): print('Válido') else: print('Inválido')		
<b>4</b>		

<b>0303 – 1 Pto</b>			
¿Cuál es la salida del siguiente programa?			
<pre>def noLaborable(d):     finde=(6,7)     return d in finde  #PPal # 1 es lunes y 7 es domingo dias=[1,5,4,6,2,7] sinBanco=list(filter(noLaborable,dias)) print(sinBanco)</pre>			
<b>1</b>	[1, 2]		<b>1</b>
<b>2</b>	[6, 7]	<b>X</b>	<b>2</b>
<b>3</b>	5		<b>3</b>
<b>4</b>	[]		<b>4</b>

<b>0403 – 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> <p>lionel,messi,futbol  gaby,sabattini,tenis  rafa,nadal,tenis  chapu,nocioni,basquet</p> <p>Programa a ejecutarse:</p> <pre> arch=open('deportistas.txt','r') lista=arch.readlines() arch.close() nombres=[] for dep in lista:     depor=dep.split(',')     nombres.append(depor[1]+'*****\n') arch=open('destacados.txt','w') arch.writelines(nombres) 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>  'yo soy argentina'.split(' ') -&gt; ['yo', 'soy', 'argentina']</p>			
<b>1</b>	<b>messi*****  sabattini*****  nadal*****  nocioni*****</b>	<b>X</b>	<b>1</b>
<b>2</b>	<b>Rafa,  Nadal,  Tenis  Chapu,  Nocioni,  Basquet</b>		<b>2</b>
<b>3</b>	<b>Lionel,,  gaby,,  rafa,,  chapu,,</b>		<b>3</b>
<b>4</b>	<b>lionel,messi,futbol  gaby,sabattini,tenis  rafa,nadal,tenis  chapu,nocioni,basquet</b>		<b>4</b>

<b>0503 – 1 Pto</b>			
¿Qué debería ser <b>estructura</b> para que la siguiente instrucción se ejecute sin problemas?			
dato=estructura['Hola']			
<b>1</b>	<b>Un diccionario</b>	<b>X</b>	<b>1</b>
<b>2</b>	<b>Una lista de lista</b>		<b>2</b>
<b>3</b>	<b>Una tupla</b>		<b>3</b>
<b>4</b>	<b>Una string</b>		<b>4</b>

<b>0603 – 1 Pto</b>			
Para averiguar el camarote de un viaje en barco asignado a un pasajero se dispone del siguiente programa. ¿Cuál función impide que aborte la ejecución del mismo si se ingresa el nombre <b>Alicia</b> ?			
<pre>def cabina (...): - - -  barco={'Ana':15,'Ulises':2,'Flores':1,       'Ignacio':15, 'Agustín':17,'Paulina':22} print('Averiguá en qué camarote viaja') pasajero=input('Nombre: ') print(pasajero,cabina(barco,pasajero))</pre>			
<b>1</b>	<pre>def cabina(nom):     except:         camarote=b[nom]         cartel='viaja en camarote '+str(camarote)     try:         cartel='NO VIAJA EN ESTE BARCO'     return cartel</pre>		<b>1</b>
<b>2</b>	<pre>def cabina(nom,b):     camarote=b[nom]     cartel='viaja en camarote '+str(camarote)     return cartel</pre>		<b>2</b>
<b>3</b>	<pre>def cabina(b,nom):     camarote=b[nom]     try:         cartel='viaja en camarote '+str(camarote)     except:         cartel='NO VIAJA EN ESTE BARCO'     return nom</pre>		<b>3</b>
<b>4</b>	<pre>def cabina(b,nom):     try:         camarote=b[nom]         cartel='viaja en camarote '+str(camarote)     except:         cartel='NO VIAJA EN ESTE BARCO'     return cartel</pre>	<b>X</b>	<b>4</b>

0703 – 1 Pto							
Para el DataFrame <b>vue</b> de pandas, que contiene:							
	nombre	dni	fila	asiento	tarifa		
0	Paz, Orlando	25889145	14	C	flex		
1	Manzur, Jimena	15226223	11	D	promo		
2	Hermenegildo, Juan	19002136	6	A	promo		
3	Decquer, Mariqana	41100107	22	C	base		
4	Tasz, Martín	36214875	14	F	flex		
¿Qué contendrá <b>vue1</b> después de la siguiente operación?							
vue1=vue[vue['asiento']!='C']							
1	nombre	dni	fila	asiento	tarifa		
	1 Manzur, Jimena	15226223	11	D	promo		
	2 Hermenegildo, Juan	19002136	6	A	promo	X	1
	4 Tasz, Martín	36214875	14	F	flex		
2	dni	tarifa					
	0 25889145	flex					
	1 15226223	promo					
	2 19002136	promo					2
	3 41100107	base					
	4 36214875	flex					
3	nombre	dni	tarifa				
	0 Paz, Orlando	25889145	flex				
	1 Manzur, Jimena	15226223	promo				
	2 Hermenegildo, Juan	19002136	promo				3
	3 Decquer, Mariqana	41100107	base				
4	nombre	dni	fila	asiento	tarifa		
	0 Paz, Orlando	25889145	14	C	flex		4

0803 – 2 Ptos			
¿Cuáles modos de apertura deben emplearse con el archivo <b>notas.txt</b> en el siguiente programa para que no salte error?			
<pre> arch=open('notas.txt',...) #primer open() filas=['1\n','2\n','3\n'] arch.writelines(filas) arch.close()  arch=open('notas.txt',...) #segundo open() arch.write('Fin\n') arch.close()                     </pre>			
1	Primer open() 'r'	Segundo open() 'r+'	1
2	Primer open() 'w'	Segundo open() 'a'	X 2
3	Primer open() 'r'	Segundo open() 'a'	3
4	Primer open() 'a'	Segundo open() 'r'	4

0903 – 2 Ptos			
Para el DataFrame <b>vue</b> de pandas, que contiene:			
	nombre	dni	fila asiento tarifa
0	Paz, Orlando	25889145	14 C flex
1	Manzur, Jimena	15226223	11 D promo
2	Hermenegildo, Juan	19002136	6 A promo
3	Decquer, Mariqana	41100107	22 C base
4	Tasz, Martín	36214875	14 F flex
¿Qué operación produce el siguiente resultado?			
6			
1	<code>vue['fila'].min()</code>	X	1
2	<code>vue.loc[:, ['fila', 'asiento']]</code>		2
3	<code>vue[vue['tarifa'].isnull()]</code>		3
4	<code>vue.iloc[2:]</code>		4

1003 – 2 Ptos			
En el siguiente programa:			
<pre>txt='Me picaron MAS mosquitos QUE EN EL VERANO' traductor={'e':''), 'o':'%', 'a':'*', 'i':'/', '            'u':'@', 'm':'&amp;', ' ':'+-'} for letra in txt:     if letra in traductor:         ..... # línea a completar print(txt)</pre>			
Que traduce los caracteres del texto que están definidos en <b>traductor</b> . ¿Cuál debería ser la línea faltante en el código? La salida final debería ser:			
M) +-p/c*r%n+-MAS+-&%sq@/t%s+-QUE+-EN+-EL+-VERANO			
1	<code>txt.replace(traductor[letra], '1')</code>		1
2	<code>txt=txt.replace(letra, traductor[letra])</code>	X	2
3	<code>If letra in txt:</code>		3
4	<code>letra=traductor[letra]</code>		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