

## PENSAMIENTO COMPUTACIONAL (90)

UBAXXI

## TEMA 2

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>0102 – 1 Pto</b>			
¿Cuál de los siguientes códigos valida adecuadamente que se ingrese un nombre iniciado con consonante? Debe detectar el error y garantizar un dato válido <b>Ejs válidos: Lucía, Pedro, JUAN</b> <b>Ejs inválidos: Úrsula, Inés, 9ema</b>			
<b>1</b>	<pre>voc='AEIOUÁÉÍÓÜ' nom=input('Nombre que comience con consonante: ') if nom[1].isalpha() or nom[0] not in voc:     print('Inválido') else:     print('Vale')</pre>		<b>1</b>
<b>2</b>	<pre>voc='AEIOUÁÉÍÓÜ' nom=input('Nombre que comience con consonante: ') while not nom.isalpha():     nom=input('Nombre que comience con consonante: ') while nom.lower() not in voc:     nom=input('Nombre que comience con consonante: ')</pre>		<b>2</b>
<b>3</b>	<pre>voc='AEIOUÁÉÍÓÜ' nom=input('Nombre que comience con consonante: ') while nom.isalpha() and nom[2] not in voc:     nom=input('Nombre que comience con consonante: ')</pre>		<b>3</b>
<b>4</b>	<pre>voc='AEIOUÁÉÍÓÜ' nom=input('Nombre que comience con consonante: ') while nom[0].upper() in voc or not nom.isalpha():     nom=input('Nombre que comience con consonante: ')</pre>	<b>X</b>	<b>4</b>

<b>0202 – 1 Pto</b>			
¿Cuál es la salida correcta del siguiente programa?			
<pre>def nomMes(mes):     meses={1:'ENE',3:'MAR',4:'ABR',2:'FEB',6:'JUN',5:'MAY',            7:'JUL',8:'AGO',9:'SEP',12:'DIC',10:'OCT',11:'NOV'}     return meses[mes]  #PPal meses=[11,5,4,6,7,8] nombres=list(map(nomMes,meses)) print(nombres)</pre>			
<b>1</b>	['NOV', 'MAY', 'ABR', 'JUN', 'JUL', 'AGO']	<b>X</b>	<b>1</b>
<b>2</b>	[]		<b>2</b>
<b>3</b>	['ene', 'feb', 'mar', 'abr']		<b>3</b>
<b>4</b>	[9, 3, 12]		<b>4</b>

<b>0302 – 1 Pto</b>			
¿Cuál es la salida del siguiente programa?			
<pre>def habiles(d):     finde=(6,7)     return d not in finde  #PPa1 # 1 es lunes y 7 es domingo dias=[1,5,4,6,2,7] hayBanco=list(filter(habiles,dias)) print(hayBanco)</pre>			
<b>1</b>	[1, 5, 4, 2]	<b>X</b>	<b>1</b>
<b>2</b>	[0, 0, 0, 0, 0, 0]		<b>2</b>
<b>3</b>	7		<b>3</b>
<b>4</b>	6		<b>4</b>

<b>0402 – 1 Pto</b>			
¿Qué contenido tendrá el archivo <b>nomAmigos.txt</b> al finalizar la ejecución del programa si el archivo <b>amigos.txt</b> tiene el siguiente contenido?			
Contenido de <b>amigos.txt</b> :			
<p>Juan,Álvarez,juanjo12@gmail.com          Ana María,Paz,anitapaz@yahoo.com.ar          Rodo,García,rgarcia04@gmail.com</p>			
Programa a ejecutarse:			
<pre>arch=open('amigos.txt','r+') agenda=arch.readlines() arch.close() nombres=[] for amigo in agenda:     nom=amigo.split(',')     nombres.append(nom[1]+'\\n') arch=open('nomAmigos.txt','w') arch.writelines(nombres) arch.close()</pre>			
<b>Nota:</b>			
El método <b>split()</b> devuelve una lista con las partes de un texto tomando como separador el argumento			
<b>Ej:</b>			
'yo soy argentina'.split(' ') -> ['yo', 'soy', 'argentina']			
<b>1</b>	juan ana maría rodo		<b>1</b>
<b>2</b>	Ana María ,Paz ,anitapaz@yahoo.com.ar		<b>2</b>
<b>3</b>	Álvarez Paz García	<b>X</b>	<b>3</b>
<b>4</b>	juanjo12@gmail.com ,anitapaz@yahoo.com.ar ,rgarcia04@gmail.com		<b>4</b>

<b>0502 – 1 Pto</b>			
¿Qué debe ser <b>estructura</b> para que la siguiente instrucción se ejecute sin problemas?			
<code>estructura['Hola']=1</code>			
<b>1</b>	<b>Un diccionario</b>	<b>X</b>	<b>1</b>
<b>2</b>	<b>Una 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>0602 – 1 Pto</b>			
Con un tablero de Batalla Naval de 4x4 con 7 barcos de un casillero cargados (X es barco; blanco es agua). ¿Cuál función impide que aborte la ejecución del programa si los ingresos son <b>fila=5</b> y <b>columna=10</b> ?			
<pre>def tiro(...):     -     -     -  tablero=[[' ','X','X',' '],[' ',' ',' ',' '],           ['X',' ','X','X'],['X',' ',' ','X']] print('Batalla naval') fila=int(input('Fila: ')) col=int(input('Columna: ')) print(tiro(tablero,fila,col))</pre>			
<b>1</b>	<pre>def tiro(t,i,j):     except:         casillero=t[i][j]         if casillero==X:             resp='Hundido'         else:             resp='Agua'     else:         resp='Fuera del Mar'     return resp</pre>		<b>1</b>
<b>2</b>	<pre>def tiro(t,i,j):     try:         casillero=t[i][j]         if casillero=='X':             resp='Hundido'         else:             resp='Agua'     except:         resp='Fuera del Mar'     return resp</pre>	<b>X</b>	<b>2</b>
<b>3</b>	<pre>def tiro(i,j):     casillero=t[i][j]     if casillero=='X':         resp='Hundido'     else:         resp='Agua'</pre>		<b>3</b>
<b>4</b>	<pre>def tiro():     try:         casillero=t[i][j]         if casillero=='X':             resp='Hundido'         else:             resp='Agua'     elif i&lt;j:         resp='Fuera del Mar'</pre>		<b>4</b>

0702 – 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?			
<code>vue1=vue[vue['asiento']=='C']</code>			
1	dni	tarifa	1
	0 25889145	flex	
	1 15226223	promo	
	2 19002136	promo	
	3 41100107	base	
	4 36214875	flex	
2	nombre	dni	tarifa
	0 Paz, Orlando	25889145	flex
	1 Manzur, Jimena	15226223	promo
	2 Hermenegildo, Juan	19002136	promo
	3 Decquer, Mariqana	41100107	base
3	nombre	dni	fila asiento tarifa
	0 Paz, Orlando	25889145	14 C flex
	1 Manzur, Jimena	15226223	11 D promo
	3 Decquer, Mariqana	41100107	22 C base
	4 Tasz, Martín	36214875	14 F flex
4	nombre	dni	fila asiento tarifa
	0 Paz, Orlando	25889145	14 C flex
	3 Decquer, Mariqana	41100107	22 C base
			X 4

0802 – 2 Ptos			
¿Cuáles modos de apertura deben emplearse con el archivo <b>notas.txt</b> en el siguiente programa para que funcione correctamente?			
<pre> arch=open('notas.txt',...) #primer open() datos=arch.readlines() print(datos) arch.close()  arch=open('notas.txt', ...) #segundo open() datos.append('F\n') for fila in datos:     arch.write(fila) arch.close() </pre>			
1	Primer open() 'r'	Segundo open() 'w'	X 1
2	Primer open() 'a'	Segundo open() 'w'	2
3	Primer open() 'r+'	Segundo open() 'r'	3
4	Primer open() 'w'	Segundo open() 'w'	4

11/07/2024 DataFrame **vue** de pandas, que contiene:

Tema 2 – Pag 6

	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?

	fila	asiento
0	14	C
1	11	D
2	6	A
3	22	C
4	14	F

1	<code>vue.head(1)</code>		1
2	<code>vue.loc[:, ['fila', 'asiento']]</code>	X	2
3	<code>vue[vue['tarifa'].isnull()]</code>		3
4	<code>vue.iloc[2:]</code>		4

<b>1002 – 2 Ptos</b>			
<p>En el siguiente programa:</p> <pre>txt='Me temo que no es NECESARIO TODO ' contLetras={} for letra in txt:     if letra.isalpha():         if letra.lower() in contLetras:             ..... # línea a completar         else:             contLetras[letra.lower()]=1 print(txt) print('Lista de letras y ocurrencias') for letra in contLetras:     print(letra+'/'+letra.upper(),':',contLetras[letra])</pre> <p>Que cuenta las ocurrencias en el texto de cada una de sus letras (ignorando mayúsculas y minúsculas). ¿Cuál debería ser la línea faltante en el código?</p> <p>La salida final debería ser:</p> <pre>Me temo que será NECESARIO COMPRENDER bien todo Lista de letras y ocurrencias m/M : 3 e/E : 9 t/T : 2 o/O : 5 q/Q : 1 u/U : 1 s/S : 2 r/R : 4 á/Á : 1 n/N : 3 c/C : 2 a/A : 1 i/I : 2 p/P : 1 d/D : 2 b/B : 1</pre>			
<b>1</b>	contLetras[letra] = letra-1		<b>1</b>
<b>2</b>	contLetras[letra.lower()] += 1	<b>X</b>	<b>2</b>
<b>3</b>	Letra = 0		<b>3</b>
<b>4</b>	contLetras[letra] = 10		<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>