

PENSAMIENTO COMPUTACIONAL (90)

.JEA.XI

TEMA 1

EXAMEN: SEGUNDO PARCIAL – PRIMER CUATRIMESTRE 2025

APELLIDO:	CALIFICACIÓN:
NOMBRE:	
DNI (registrado en SIU Guaraní):	
E-MAIL:	NOTA Y FIRMA DOCENTE (no rellenar)
TEL:	
AULA:	

Duración del examen: 1:30h.

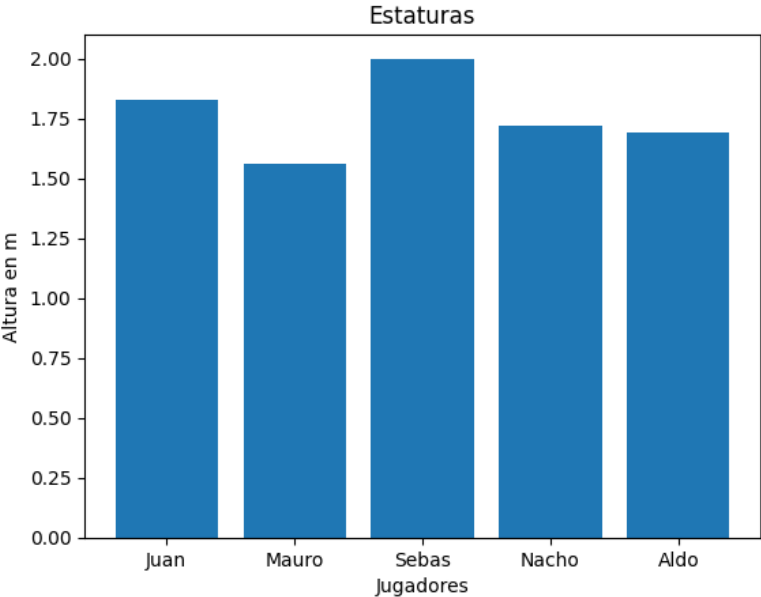
- ✓ Escribir claramente el nombre en todas las páginas.
- ✓ El examen consta de 9 preguntas de opción múltiple.
- ✓ Cada pregunta tiene una y sólo una respuesta correcta.
- ✓ Las respuestas seleccionadas deben consignarse en la siguiente matriz de opciones.
- ✓ **Sólo se considerarán las respuestas anotadas en la matriz.**
- ✓ Las preguntas de la 1 a la 5 inclusive permiten acumular 1 punto (si son correctas), de la 6 a la 9 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

[illegible]

Ejercicio 1 – Tema 1			1 Pto
Cuál de los siguientes diccionarios tiene todas sus claves numéricas (tipo int, float o complex)?			
A	{'1':2, '23':5, '0':13}		A
B	{1:'uno', 2:'dos', True:'V', False:'F'}		B
C	{0: '3', 2:66, 5:3.45, 1:[-2,5]}	X	C
D	{('1', '0'):2, ('3','5'):44}		D

Ejercicio 2 – Tema 1			1 Pto
¿Cuál de las siguientes instrucciones debería usar (junto a otras) para generar un gráfico como el siguiente con matplotlib? Los datos a representar son:			
<pre>altura = [1.83, 1.56, 2.00, 1.72, 1.69] jugadores = ['Juan', 'Mauro', 'Sebas', 'Nacho', 'Aldo']</pre>			
			
A	ax.scatter(altura, jugadores)		A
B	ax.barh(jugadores)		B
C	ax.set_title(altura)		C
D	ax.bar(jugadores, altura)	X	D

Ejercicio 3 – Tema 1			1 Pto																												
Dado el Dataframe <i>resfrios</i> :																															
<table><tr><th></th><th>mes</th><th>humedad</th><th>cantidad</th></tr><tr><td>0</td><td>enero</td><td>60</td><td>2</td></tr><tr><td>1</td><td>febrero</td><td>75</td><td>23</td></tr><tr><td>2</td><td>marzo</td><td>79</td><td>32</td></tr><tr><td>3</td><td>abril</td><td>68</td><td>31</td></tr><tr><td>4</td><td>mayo</td><td>62</td><td>45</td></tr><tr><td>5</td><td>junio</td><td>55</td><td>61</td></tr></table>					mes	humedad	cantidad	0	enero	60	2	1	febrero	75	23	2	marzo	79	32	3	abril	68	31	4	mayo	62	45	5	junio	55	61
	mes	humedad	cantidad																												
0	enero	60	2																												
1	febrero	75	23																												
2	marzo	79	32																												
3	abril	68	31																												
4	mayo	62	45																												
5	junio	55	61																												
¿Cuál de las siguientes instrucciones debería usar para producir la siguiente salida?																															
<table><tr><th></th><th>mes</th><th>humedad</th><th>cantidad</th></tr><tr><td>0</td><td>enero</td><td>60</td><td>2</td></tr><tr><td>1</td><td>febrero</td><td>75</td><td>23</td></tr><tr><td>3</td><td>abril</td><td>68</td><td>31</td></tr></table>					mes	humedad	cantidad	0	enero	60	2	1	febrero	75	23	3	abril	68	31												
	mes	humedad	cantidad																												
0	enero	60	2																												
1	febrero	75	23																												
3	abril	68	31																												
A	resfrios.tail(3)		A																												
B	resfrios[['cantidad', 'mes']]		B																												
C	resfrios.loc[resfrios.index[[1,3,5]]]		C																												
D	resfrios[resfrios['cantidad']<32]	X	D																												

Ejercicio 4 – Tema 1			1 Pto
Dado el archivo carreras.txt, que contiene alumnos anotados en distintas carreras: Quimica Analia,Ricardes,2021 Juliana,Ortesano,2023 Pedro,Fuentes,2023 Industrial Jose,Agustoni,2021 Ana,Mauner,2022 Andrea,Carragno,2022 Civil Marta,Leiva,2021 ¿Con cuál conjunto de modos de apertura funcionará adecuadamente el siguiente programa? <pre>def abre(archivo,modo): nombre=open(archivo,modo) return nombre alumnos=abre('carreras.txt',...) #modo 1? tit=alumnos.readline() tit=tit.strip('\n').lower() archivo=tit+'-alumnos.txt' lista=abre(archivo,...) #modo 2? for lin in alumnos: if ',' in lin: lista.write(lin) else: lista.close() tit=lin.strip('\n').lower() archivo=tit+'-alumnos.txt' lista=abre(archivo,...) #modo 3? alumnos.close() lista.close()</pre> Al finalizar quedarán 3 archivos adicionales: quimica-alumnos.txt, industrial-alumnos.txt y civil-alumnos.txt, que contendrá cada uno los alumnos anotados de esa carrera en el archivo original.			
A	modo 1 'r' modo 2 'w' modo 3 'r'		A
B	modo 1 'a' modo 2 'r' modo 3 'r'		B
C	modo 1 'r' modo 2 'w' modo 3 'w'	X	C
D	modo 1 'w' modo 2 'w' modo 3 'w'		D

Ejercicio 5 – Tema 1		1 Pto	
¿Cuál programa realiza un ingreso validado de un número par? ✓ No se debe aceptar un número que sea impar ✓ El programa no debe dar fallo si se ingresa un dato que no sea entero ✓ El programa debe insistir en la solicitud hasta obtener lo esperado			
A	<pre>for i in range(3): numPar=int(input('Ingresa un número par: ')) if numPar<0 or numPar%2!=0: print('No es par') print(numPar)</pre>		A
B	<pre>pide=True while pide: try: numPar=int(input('Ingresa un número par: ')) if numPar>0 and numPar%2==0: pide=False else: print('No es par') except ValueError: print('Debe ser número entero') print(numPar)</pre>	X	B
C	<pre>pide=False while pide: try: numPar=int(input('Ingresa un número par: ')) if numPar<0 or numPar%2!=0: pide=False else: print('No es par') except ValueError: print('Debe ser número entero') print(numPar)</pre>		C
D	<pre>try: numPar=int(input('Ingresa un número par: ')) except: print('No es un número') if numPar>0 and numPar%2==0: print(numPar)</pre>		D

Ejercicio 6 – Tema 1				2 Ptos																													
Dado el Dataframe <i>resfrios</i> :																																	
<table><thead><tr><th></th><th>mes</th><th>humedad</th><th>cantidad</th></tr></thead><tbody><tr><td>0</td><td>enero</td><td>60</td><td>2</td></tr><tr><td>1</td><td>febrero</td><td>75</td><td>23</td></tr><tr><td>2</td><td>marzo</td><td>79</td><td>32</td></tr><tr><td>3</td><td>abril</td><td>68</td><td>31</td></tr><tr><td>4</td><td>mayo</td><td>62</td><td>45</td></tr><tr><td>5</td><td>junio</td><td>55</td><td>61</td></tr></tbody></table>							mes	humedad	cantidad	0	enero	60	2	1	febrero	75	23	2	marzo	79	32	3	abril	68	31	4	mayo	62	45	5	junio	55	61
	mes	humedad	cantidad																														
0	enero	60	2																														
1	febrero	75	23																														
2	marzo	79	32																														
3	abril	68	31																														
4	mayo	62	45																														
5	junio	55	61																														
¿Qué salida produce la siguiente instrucción?																																	
<pre>print(resfrios['humedad'].sum())</pre>																																	
A	2	marzo	79	32	A																												
	4	mayo	62	45																													
	5	junio	55	61																													
B	75				B																												
C	399			X	C																												
D	0	enero	60	2	D																												
	1	febrero	75	23																													
	2	marzo	79	32																													
	3	abril	68	31																													

Ejercicio 7 – Tema 1		2 Ptos	
Dado el siguiente programa:			
<pre>def edita1(pal): return pal[1:].upper() def edita2(pal): return pal[0].upper() def edita3(pal): return len(pal)>5 palabras=['casa','auto','banco','cartel','semáforo'] # línea de código faltante print(nuevaLista)</pre>			
¿Cuál debería ser la línea de código faltante para que la salida sea la siguiente?			
<pre>['ASA', 'UTO', 'ANCO', 'ARTEL', 'EMÁFORO']</pre>			
A	nuevaLista=list(map(edita2,palabras))		A
B	nuevaLista.append(map(edita1,palabras))		B
C	nuevaLista=list(map(edita1,palabras))	X	C
D	nuevaLista=list(map(edita3,palabras))		D

Ejercicio 8 – Tema 1			2 Ptos
Dado el siguiente programa: <pre>estaciones=... #?? primSemes=[0,10000,25000,67500,48000,33000,29000] print('Ventas Colección Sweaters Invierno 2025') for i in range(1,len(primSemes)): print(estaciones[i][0],'\$',primSemes[i], estaciones[i][1].upper())</pre> ¿Cuál de las opciones de la estructura estaciones funcionará correctamente para producir la siguiente salida? Ventas Colección Sweaters Invierno 2025 enero \$ 10000 VERANO febrero \$ 25000 VERANO marzo \$ 67500 VERANO-OTOÑO abril \$ 48000 OTOÑO mayo \$ 33000 OTOÑO junio \$ 29000 OTOÑO-INVIERNO			
A	<pre>estaciones=['enero','verano','febrero','verano','marzo', 'verano-otoño','abril','otoño','mayo', 'otoño','junio','otoño-invierno']</pre>		A
B	<pre>estaciones={1:['enero','verano'],2:['febrero','verano'], 3:['marzo','verano-otoño'],4:['abril','otoño'], 5:['mayo','otoño'], 6:['junio','otoño-invierno']}</pre>	X	B
C	<pre>estaciones={'enero':'verano','febrero':'verano', 'marzo':'verano-otoño','abril':'otoño', 'mayo':'otoño','junio':'otoño-invierno'}</pre>		C
D	<pre>estaciones={'0':['enero','verano'], '1':['febrero','verano'], '2':['marzo','verano-otoño'], '3':['abril','otoño'], '4':['mayo','otoño'], '5':['junio','otoño-invierno']}</pre>		D

Ejercicio 9 – Tema 1			2 Ptos
¿Cuál es la salida del siguiente programa? <pre>def edita (txt): return txt.upper() def patron(txt): return ('á' in txt) or ('ó' in txt) ciudades=['anillaco','san javier','la dársena','fiambalá'] mostrar=list(filter(patron,ciudades)) salida=list(map(edita,mostrar)) print(salida)</pre>			
A	<pre>['anillaco','san javier','la dársena', 'fiambalá']</pre>		A
B	<pre>['ANILLACO','SAN JAVIER']</pre>		B
C	<pre>[]</pre>		C
D	<pre>['LA DÁRSENA','FIAMBALÁ']</pre>	X	D