

<b>1.a)</b>	<b>20 puntos</b>	
def buscar_nombre(rut):	1 pto	<b>Observación:</b>  Una línea está correcta si está en la posición correcta (funcionalmente) y con la indentación correcta.
arch = open('pacientes.txt')	1 pto	
for line in arch:	1 pto	
r, n, e = line.strip().split(':')	1 pto	
if rut == r:	1 pto	
arch.close()	1 pto	
return n	1 pto	
<b>1.b)</b>		
def pacientes_dia(dia, mes, anio):	1 pto	
fecha = map(str,(dia, mes, anio))	1 pto	
fecha = '-'.join(fecha)	1 pto	
lista_n = []	1 pto	
arch = open('atenciones.txt')	1 pto	
for line in arch:	1 pto	
r, f, c = line.strip().split(':')	1 pto	
if fecha == f:	1 pto	
nombre = buscar_nombre(r)	1 pto	
lista_n.append(nombre)	1 pto	
arch.close()	1 pto	
lista_n = list(set(lista_n))	1 pto	
return lista_n	1 pto	

Pregunta 2:		40 pts			
Item	0 Pto	3 Ptos	6Ptos	10 Ptos	12 Ptos
Manejo de archivos	No demuestra tener noción de cómo usar archivos o nunca cierra los archivos abiertos.	Al menos en una función utiliza de forma completamente correcta archivos. Cierra los archivos de forma incorrecta. Abre los archivos con el nombre de los archivos del ejemplo y no con el parámetro de la función.	Usa correctamente los archivos en al menos 2 de los casos, abre y cierra donde corresponde, utiliza el nombre recibido como parámetro y no el nombre de los ejemplos.		
Función en_Comuna()	No demuestra nociones de cómo resolver el problema	Demuestra tener una noción de qué hacer, tiene la lógica de la solución pero comete a los más 2 errores en el proceso. Ej: no inicializar la lista, errores al agregar elementos en la lista, errores en el procesamiento del texto.	Comete algún error importante o 2 errores leves. Ej: no crea una lista, se equivoca al ingresar un elemento a la lista, uso incorrecto de los parámetros de la función, entre otros.	Retorna correctamente el resultado esperado.	
Función promedio_Restaurante()	No demuestra nociones de cómo resolver el problema	Demuestra tener una noción de qué hacer, tiene la lógica de la solución pero con a lo más 3 errores en el proceso. Ej: no inicializa las variables de suma y contar, no transforma a número la cantidad de estrellas, realiza división entera, uso incorrecto del split, etc.	Comete algún error importante o 2 errores leves. Ej: olvida el strip, división entera y no real para el promedio, hace strip sobre split.	Comete un error leve. Ej: utiliza strip sin paréntesis, split por ':' pero sin comillas, división entera y no real para el promedio.	Retorna correctamente el resultado esperado
Función elMejorDeLaComuna()	No demuestra nociones de cómo resolver el problema	Demuestra tener una noción de qué hacer, tiene la lógica de la solución pero con varios errores en el proceso. Ej: no inicializa variable de máximo y nombre restaurant, uso incorrecto de las funciones anteriores.	Comete algún error importante o 2 errores leves. Ej: inicializa el máximo en un valor muy grande, uso incorrecto de los parámetros de la función, no considera el caso de que la comuna no tenga restaurants.	Comete un error leve. Ej: equivoca el signo al comparar el máximo,	Retorna correctamente el resultado esperado
<b>Descuentos</b>		<b>0 Pto</b>		<b>-4 Puntos</b>	
Forma y expresiones	No comete errores graves con respecto al lenguaje; a lo más errores atribuibles más a descuidos que ignorancia (por ejemplo, en una ocasión olvidar los dos puntos al final de la línea, confunde = y == o confunde () y [] o , ocupa ≤ en lugar de <=, pero el resto de las veces lo hace bien)		Demuestra no entender construcciones importantes del lenguaje (por ejemplo: no indenta, le pone una condición al else, confunde el string x con la variable x, asigna valor a variables de forma incorrecta 4=x), consistentemente no le pone los paréntesis finales a d.items() o d.values(). Un descuento de -4 por tipo de error.		

Pregunta 3: 40 pts				
Item	0 Pto	4 Ptos	7 Ptos	10 Ptos
Obtener los apellidos de los candidatos	No sabe como acceder a los candidatos o los errores son mayores a sus aciertos.	Algorítmicamente obtiene los apellidos de los candidatos pero comete a lo más 3 errores en su implementación. Ej: no cierra el archivo, agrega comillas o un .txt a la variable con el nombre del archivo.	Obtiene los apellidos de los candidatos pero comete a lo más 2 errores en su implementación. Ej: uso incorrecto del split, no incluye comillas en el carácter de separación del split, cierra el archivo en la ubicación incorrecta.	Accede correctamente al apellido de los candidatos.
Obtener los tweets por cada candidato	No demuestra nociones de cómo obtener los tweets de cada candidato	Algorítmicamente obtiene los tweets de los candidatos pero comete a lo más 3 errores en su implementación. Ej: compara incorrectamente si el candidato es parte del tweet, no incluye tweets ya incluidos para otro candidato, no cierra el archivo, agrega comillas a la variable con el nombre del archivo.	Obtiene los tweets de los candidatos pero comete a lo más 2 errores en su implementación. Ej: uso incorrecto del split, no concidera mayúsculas y minúsculas en el calse, abre y cierra el archivo en la ubicación incorrecta impidiendo una nueva lectura del mismo.	Obtiene de forma correcta los Tweets de cada candidato, sin errores
Ordenar tweets por fecha	No demuestra nociones o no ordena los tweets de algún candidato por fecha de publicación.	Algorítmicamente ordena los tweets pero comete a lo más 2 errores en su implementación. Ej: asigna sort() o reverse() a una variable, no agrega correctamente las líneas a una lista, ordena por otro elemento que no es la fecha.	Ordena los tweets por candidato pero comete algún error en su implementación. Ej: ordena del más reciente al más antiguo, al hacer un append de una tupla olvida el paréntesis de la misma.	Ordena correctamente los tweets de un candidato.
Generar los archivos por cada candidato	No demuestra nociones de cómo generar los archivos	Algorítmicamente intenta crear los archivos por candidato pero comete a lo más 3 errores en su implementación. Ej: nombra de forma incorrecta los archivos, olvida ocupar el método 'w' en el open, no escribe la línea en el formato que se indicó.	Crea los archivos por candidato pero comete a lo más 2 errores menores en su implementación. Ej: olvida pasar a minúscula el nombre del candidato, olvida el '.txt', olvida el '\n' al final de la línea.	Genera de forma correcta los archivos por candidato.
<b>Descuentos</b>		<b>0 Pto</b>	<b>-4 Puntos</b>	
Sintaxis y expresiones	No comete errores graves con respecto al lenguaje; a lo más errores atribuibles más a descuidos que ignorancia (por ejemplo, en una ocasión olvidar los dos puntos al final de la línea, confunde = y == o confunde () y [] o , ocupa ≤ en lugar de <=, pero el resto de las veces lo hace bien)		Demuestra no entender construcciones importantes del lenguaje (por ejemplo: no indenta, le pone una condición al else, hace while con rango, confunde el string x con la variable x, asigna valor a variables de forma incorrecta 4=x). Un descuento de -4 por tipo de error.	