

<b>Nombre Programa</b>
AyudanteNomina
<b>Descripción</b>
Programa que calcula la nomina de una empresa a partir de un archivo de texto con los nombres y salarios, por ejemplo nombres.txt. Al final guarda la nomina en el archivo liquidacion.txt. Se guarda registro de errores en errores.txt y registro de operación en archivo log.txt.
<b>Versión</b>
2.0 PUNTO 4
<b>Autor (Email Institucional - Nombre CamelCase)</b>
<a href="mailto:enorenaa@unal.edu.co">enorenaa@unal.edu.co</a> - Esteban Noreña Arroyave
<b>Fecha (Mes Día de Año)</b>
Septiembre 21 de 2015

Entradas		
#	Descripción	
E	1	Archivo con nomina con extensión .txt, que contiene los nombres y el salario base para cada cada empleado.
E	2	
E	3	
E	4	
E	5	

Validaciones		
#	Descripción	
Va	1	Haya como mínimo un argumento de entrada que debe ser el archivo de nomina.
Va	2	Validar que el argumento, sea el nombre de un archivo válido y que su extensión sea .txt.
Va	3	Validar que el número de líneas sea igual 2.
		Validar que cada línea tenga la estructura:
		NOMBRE_COMPLETO*SALARIO_BASE*CARGO
Va	4	
Va	5	Validar que el nombre tenga mínimo 2 palabras y máximo 5 palabras.
Va	6	Validar que el salario sea de tipo numérico.
Va	7	
Va	8	
Va	9	
Va	10	

Salidas		
#	Descripción	

		<p>Archivo con el valor de la nómina calculada llamado liquidacion.txt.</p> <p>La primera linea del archivo contendra el promedio para cada cargo</p> <p>Cada línea del archivo de liquidación tendrá la siguiente estructura de 15 columnas, 13 de ellas calculadas como liquidación:</p> <p>NOMBRE_COMPLETO*SALARIO_BASE</p> <p> </p> <p>VALOR_AUXILIO_TRANSPORTE_EFECTIVO</p> <p> </p> <p>VALOR_CESANTIAS*VALOR_INTERESES_CESANTIAS*</p> <p>VALOR_PRIMA*VALOR_VACACACIONES</p> <p> </p> <p>VALOR_ARL*VALOR_SALUD_EMPRESA*VALOR_PENSION_EMPRESA</p> <p> </p> <p>VALOR_SALUD_EMPLEADO*VALOR_PENSION_EMPLEADO*</p> <p>VALOR_FONDO_SOLIDARIDAD</p> <p> </p> <p>COSTO_TOTAL_EMPLEADO (SALARIO_BASE + TOTAL APROPIACIONES + AUXILIO_TRANSPORTE_EFECTIVO)</p> <p> </p> <p><b>S 1</b> SALARIO NETO EMPLEADO (SALARIO_BASE - TOTAL DEDUCCIONES + AUXILIO_TRANSPORTE_EFECTIVO)</p>
<b>S</b>	<b>2</b>	Archivo con listado de errores llamado errores.txt.
<b>S</b>	<b>3</b>	Archivo con registro de operación de programa llamado log.txt.
<b>S</b>	<b>4</b>	
<b>S</b>	<b>5</b>	

Verificaciones	
#	Descripción
<b>Ve 1</b>	Verificar que existan los archivos errores.txt y log.txt
<b>Ve 2</b>	Si el archivo liquidacion.txt fue creado, verificar que tenga la estructura requerida que ha sido especificada en S1.

Ve	3	
Ve	4	
Ve	5	

Funciones				
#	Nombre	Descripción	Argumentos	Retorno
1	validar_linea	Valida que una linea del archivo cumpla con las validaciones Va3, Va4, Va5.	linea a validar, numero de la linea	<p>Retorna un array con la siguiente estructura:</p> <p>[True   False][Nombre Completo][Salario]</p> <p>[True   False] -&gt; True si la linea es valida, False lo contrario.</p> <p>[Nombre Completo] -&gt; Se almacena el nombre si la linea es valida, de lo contrario va vacío.</p> <p>[Salario] -&gt; Se almacena el salario como cadena de texto si la linea es valida, de lo contrario va vacío.</p>
2	guardar_error	Guarda un mensaje de error en el archivo errores.txt	Mensaje a guardar.	Nada.
3	guardar_log	Guarda un mensaje de registro en el archivo log.txt	Mensaje a guardar.	Nada.
4	terminar_programa	Finaliza el programa y guarda mensaje de error.	Mensaje de razón de finalización.	Nada.
5	crear_archivo	Crear un archivo con el nombre especificado.	Nombre de archivo.	Booleano. True para creacion exitosa, False para creacion fallida.
6	leer_lineas_archivo	Lee las lineas de un archivo. Internamente se encarga de abrir y cerrar el archivo para cada lectura.	Nombre de archivo.	Array que contiene las lineas leidas del archivo.
7	escribir_linea_archivo	Escribe una linea al final del archivo. Internamente se encarga de abrir y cerrar el archivo para cada lectura.	Nombre de archivo, Linea a escribir.	Booleano. True para escritura exitosa, False para escritura fallida.

Variables			
#	Nombre	Uso	Tipo
1	nombre_archivo_errores	Contiene nombre del archivo para guardar errores.	string
2	nombre_archivo_registro	Contiene nombre del archivo para guardar registro de operacion.	string
3	nombre_archivo_liquidacion	Contiene nombre del archivo para guardar liquidacion calculada.	string
4	numero_minimo_lineas	Numero minimo de lineas que debe contener el archivo de nomina	int
5	Multiples constantes para calculo salario minimo	Constantes como el salario minimo aplicable, auxilio de transporte, porcentaje riesgos laborales.	ints
6	cantidad_argumentos	Guarda el numero de argumentos proporcionados	int
7	nombre_archivo_nomina	Guarda el nombre del archivo de nomina para validarlo posteriormente	string
8	lineas_archivo_nomina	Guarda las lineas leidas del archivo de nomina	string list
9	numero_lineas_nomina	Guarda la cantidad de lineas del archivo de nomina	int
10	nomina	Array que guarda la informacion de nomina.	array
11	liquidacion	Array que guarda la informacion de liquidacion de nomina.	array
12	total_<<cargo>>	Variable que llevara el acumulado de cada cargo	int
13			
14			
15			

Algoritmo	
Paso	Descripción
1	Crear archivos errores.txt y log.txt, para guardar cada error que suceda y guardar el registro de la ejecución del programa, respectivamente.
2	Realizar lectura de argumentos de línea de comandos. Realizar validaciones de entrada Va1 y Va2.
3	Si se cumplen Va1 y Va2, realizar lectura de cantidad de numero de lineas de archivo. Validar Va3.
4	Si se cumple Va3, realizar Ciclo.
5	Antes de ciclo definir variable tipo array (nomina), de tamaño numero_lineas_archivo x 2 columnas, que contendrá las lineas que cumplen con las validaciones. La longitud del array debe ser igual al numero de lineas del archivo.
6	Inicio de ciclo 1, desde 1 hasta número de líneas (numero_lineas_archivo), variable indice de ciclo: x.
7	Leer línea x, validar Va4, Va5, Va6 para la línea x. Si se cumplen las validaciones, marcar como válida la línea x. Si la línea es buena, guardarla en el arreglo nomina, guardando el nombre en nomina[x][0] y guardando el salario en nomina[x][1]. Si la línea no cumple, terminar el programa guardando el error, indicando el número de la línea del archivo que está mal y solicitando al usuario que corrija el archivo de la nómina para poder continuar. Todos las líneas del archivo deben cumplir con las validaciones. La validación y lectura de campos de cada línea se realizará con la función F1, validar_linea.
8	Fin de ciclo 1
9	Después de haber leído todo el archivo y haberlo puesto en memoria en el array nomina, empezar a hacer el calculo de la liquidacion de nomina en otro ciclo. Antes se debe crear una variable de tipo array numerico llamada liquidacion y tendrá las dimensiones numero_lineas_archivo x 13 columnas. Al final del ciclo 2 se pregunta de que cargo es para sumar al total de cada uno
10	Inicio de ciclo 2, desde 1 hasta numero_lineas_archivo, variable indice de ciclo: z
11	Leer nomina[z][1] y guardarlo en la variable local salario_base.
12	Guardar en liquidacion[z][0] el valor a aplicar de auxilio de transporte. Si el salario_base es menor o igual a 2 salario minimos se le da auxilio de transporte, de resto el valor es 0.
13	Guardar en liquidacion[z][1] el valor de cesantias.

<b>14</b>	Guardar en liquidacion[z][2] el valor de intereses sobre cesantias.
<b>15</b>	Guardar en liquidacion[z][3] el valor de prima de servicios.
<b>16</b>	Guardar en liquidacion[z][4] el valor de vacaciones.
<b>17</b>	Guardar en liquidacion[z][5] el valor de ARL.
<b>18</b>	Guardar en liquidacion[z][6] el valor de salud por parte del empleador.
<b>19</b>	Guardar en liquidacion[z][7] el valor de pensión por parte del empleador.
<b>20</b>	Guardar en liquidacion[z][8] el valor de salud para el empleado.
<b>21</b>	Guardar en liquidacion[z][9] el valor de salud para el empleado.
<b>22</b>	Guardar en liquidacion[z][10] el valor de aporte al fondo de solidaridad de pensiones. Si el salario base es mayor o igual a 4 salarios minimos, este valor es el 1% del salario base, de resto el valor es 0.
<b>23</b>	Guardar en liquidacion[z][11] el costo total para la empresa.
<b>24</b>	Guardar en liquidacion[z][12] el salario neto para el empleado.
<b>25</b>	Fin de ciclo 2
<b>26</b>	Despues de tener calculada la liquidacion, crear el archivo liquidacion.txt y guardarla en éste.
<b>27</b>	Antes del ciclo 3 se añade la primera linea con los totales de cada cargo
<b>28</b>	Inicio ciclo 3, desde 1 hasta numero_lineas_archivo, variable indice de ciclo: w.
<b>29</b>	Guardar en archivo linea w con el formato definido.
<b>30</b>	Fin ciclo 3
<b>31</b>	Después de finalizar el ciclo cerrar el archivo liquidacion.txt
<b>32</b>	
<b>33</b>	
<b>34</b>	
<b>35</b>	
<b>36</b>	
<b>37</b>	
<b>38</b>	
<b>39</b>	
<b>40</b>	
<b>41</b>	
<b>42</b>	
<b>43</b>	
<b>44</b>	



45	
46	
47	
48	
49	
50	
51	

¿Qué sabe como implementar del algoritmo propuesto?	
Conocimiento	Descripción
1	Hacer ciclos en Python.
2	Leer archivo en Python.
3	Crear una variable.
4	Crear archivo en Python.
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

¿Qué no sabe y debe consultar en Google?	
Pregunta	Descripción
1	Hacer un split de una cadena de texto.
2	Qué es un array y cómo definirlo en Python.
3	Cómo dar formato a números flotantes como strings en Python.
4	Validar argumentos de línea de comandos.
5	Hacer un ciclo for en Python.
6	Inicializar array de múltiples dimensiones en Python.
7	
8	
9	
10	