

<b>Nombre Programa</b>
AyudanteNomina
<b>Descripción</b>
Programa que calcula la nomina de una empresa a partir de dos archivos de texto con los nombres y salarios, por ejemplo nombres.txt y salarios.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>
2.0.1
<b>Autor (Email Institucional - Nombre CamelCase)</b>
<a href="mailto:jposadah@unal.edu.co">jposadah@unal.edu.co</a> -Jacobó Posada Hoyos
<b>Fecha (Mes Día de Año)</b>
Septiembre 01 de 2015

Entradas		
#	Descripción	
E	1	Archivo con nombres con extensión .txt, que contiene los nombres de cada empleado.
E	2	Archivo con salarios con extensión .txt, que contiene los salarios de cada empleado.
E	3	
E	4	
E	5	

Validaciones		
#	Descripción	
Va	1	Validar que haya como mínimo dos argumentos de entrada que deben ser los archivos nombres y salarios.
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 de cada archivo sea igual 2 y que este sea igual entre ambos.
Va	4	Validar que el nombre tenga mínimo 2 palabras y máximo 5 palabras.
Va	5	Validar que el contenido del archivo salarios contenga solo caracteres numéricos y que sea mayor que cero.
Va	6	
Va	7	
Va	8	
Va	9	

Salidas		
#	Descripción	

		<p>Archivo con el valor de la nómina calculada llamado liquidacion.txt. 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.	
<b>Ve 3</b>		
<b>Ve 4</b>		
<b>Ve 5</b>		

Funciones				
#	Nombre	Descripción	Argumentos	Retorno
1	validar_linea_nombres	Valida que una linea del archivo cumpla con las validaciones.	linea a validar, numero de la linea	<p>Retorna un array con la siguiente estructura:</p> <p>[True   False][Nombre Completo]</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>
2	validar_linea_salarios	Valida que una linea del archivo cumpla con las validaciones.	linea a validar, numero de la linea	<p>Retorna un array con la siguiente estructura:</p> <p>[True   False][Salario]</p> <p>[True   False] -&gt; True si la linea es valida, False lo contrario.</p> <p>[Salario] -&gt; Se almacena el salario como cadena de texto si la linea es valida, de lo contrario va vacío.</p>
3	guardar_error	Guarda un mensaje de error en el archivo errores.txt	Mensaje a guardar.	Nada.
4	guardar_log	Guarda un mensaje de registro en el archivo log.txt	Mensaje a guardar.	Nada.
5	terminar_programa	Finaliza el programa y guarda mensaje de error.	Mensaje de razón de finalización.	Nada.
6	crear_archivo	Crear un archivo con el nombre especificado.	Nombre de archivo.	Booleano. True para creacion exitosa, False para creacion fallida.

7	leer_lineas_archivo	Lee las líneas de un archivo. Internamente se encarga de abrir y cerrar el archivo para cada lectura.	Nombre de archivo.	Array que contiene las líneas leídas del archivo.
8	escribir_linea_archivo	Escribe una línea al final del archivo. Internamente se encarga de abrir y cerrar el archivo para cada lectura.	Nombre de archivo, Línea a escribir.	Booleano. True para escritura exitosa, False para escritura fallida.
9				

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 deben contener los archivos nombres y salarios.	int
5	Multiples constantes para calculo salario minimo	Constantes como el salario minimo aplicable, auxilio de tranporte, porcentaje riesgos laborales.	ints
6	cantidad_argumentos	Guarda el numero de argumentos proporcionados	int
7	nombre_archivo_nombres	Guarda el nombre del archivo de nombres para validarlo posteriormente	string
8	nombre_archivo_salarios	Guarda el nombre del archivo de salarios para validarlo posteriormente	string
9	lineas_archivo_nombres	Guarda las lineas leidas del archivo de nombres	string list
10	lineas_archivo_salarios	Guarda las lineas leidas del archivo de salarios	string list
11	numero_lineas_nombres	Guarda la cantidad de lineas del archivo de nombres	int
12	numero_lineas_salarios	Guarda la cantidad de lineas del archivo de salarios	int
13	nomina	Array que guarda la informacion de nomina.	array
14	liquidacion	Array que guarda la informacion de liquidacion de nomina.	array
15			
16			
17			
18			

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 en cada uno de los archivos, validar Va4 en el archivo nombres y Va5 en el archivo salarios, 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 los archivos para poder continuar. Todos las líneas de los archivos 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 cada uno de los archivos y haberlos puesto en la 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.
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.
14	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>	Inicio ciclo 3, desde 1 hasta numero_lineas_archivo, variable indice de ciclo: w.
<b>28</b>	Guardar en archivo linea w con el formato definido.
<b>29</b>	Fin ciclo 3
<b>30</b>	
<b>31</b>	
<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>	
<b>45</b>	



46	
47	
48	
49	
50	

¿Qué sabe como implementar del algoritmo propuesto?	
Conocimiento	Descripción
1	Hacer ciclos en Python.
2	Leer archivo en Pyhon.
3	Crear una variable.
4	Crear archivo en Python.
5	Hacer un split de una cadena de texto.
6	Crear un array y definirlo
7	Dar formato a numeros flotantes como strings en Python.
8	Validar argumentos de linea de comandos.
9	Inicializar array de multiples dimensiones.
10	
11	
12	
13	
14	
15	

¿Qué no sabe y debe consultar en Google?	
Pregunta	Descripción
1	Como eliminar un archivo .txt
2	
3	
4	
5	
6	
7	
8	
9	
10	