

Nombre Programa
AyudanteNomina
Descripción
Programa que calcula la nomina de una empresa a partir de un archivo de texto con los nombres y salarios, por ejemplo nommina.txt, lee las apropiaciones y deducciones para calculo de liquidacion desde archivos apropiaciones.txt y deducciones.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.
Versión
2.0.6
Autor (Email Institucional - Nombre CamelCase)
cdviverosd@unal.edu.co - Cristhian Danilo Viveros Delgado
Fecha (Mes Día de Año)
Septiembre 23 de 2015

Entradas		
#	Descripción	
E	1	Archivo con nomina con extensión .txt, que contiene los nombres y el salario base para cada empleado.
E	2	Archivo con apropiaciones con extensión .txt que contiene el nombre de la apropiación y el porcentaje de esta
E	3	Archivo con deducciones con extensión .txt que contiene el nombre de la deducción y el porcentaje de esta
E	4	
E	5	
Validaciones		
#	Descripción	
Va	1	Haya como mínimo tres argumento de entrada que debe ser el archivo de nomina, de apropiaciones y de deducciones.
Va	2	Validar que los argumentos, sean el nombre de un archivo válido y que su extensión sea .txt.
Va	3	Validar que el número de líneas de nomina sea mayor que 2.
Va	4	Validar que cada línea del archivo de nomina tenga la estructura: NOMBRE_COMPLETO*SALARIO_BASE
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	Validar que cada línea del archivo de apropiaciones tenga la estructura: APROPIACION*PORCENTAJE_APROPIACION
Va	8	Validar que el porcentaje de apropiacion sea numerico (flotante)
Va	9	Validar que cada línea del archivo de apropiaciones tenga la estructura: DEDUCCION*PORCENTAJE_DEDUCCION
Va	10	Validar que el porcentaje de deduccion sea numerico (flotante)

S	1	<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 VALOR_AUXILIO_TRANSPORTE_EFECTIVO VALOR_CESANTIAS*VALOR_INTERESES_CESANTIAS* VALOR_PRIMA*VALOR_VACACACIONES VALOR_ARL*VALOR_SALUD_EMPRESA*VALOR_PENSION_EMPRESA VALOR_SALUD_EMPLEADO*VALOR_PENSION_EMPLEADO* VALOR_FONDO_SOLIDARIDAD COSTO_TOTAL_EMPLEADO (SALARIO_BASE + TOTAL APROPIACIONES + AUXILIO_TRANSPORTE_EFECTIVO) SALARIO NETO EMPLEADO (SALARIO_BASE - TOTAL DEDUCCIONES + AUXILIO_TRANSPORTE_EFECTIVO)</p>
S	2	Archivo con listado de errores llamado errores.txt.
S	3	Archivo con registro de operación de programa llamado log.txt.
S	4	
S	5	
Verificaciones		
	#	Descripción
Ve	1	Verificar que existan los archivos errores.txt y log.txt
Ve	2	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,	linea a validar, numero de la	Retorna un array con la siguiente estructura: [True False][Nombre Completo][Salario]
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	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	Nombre de archivo, Linea a escribir.	Booleano. True para escritura exitosa, False para escritura fallida.
8	validar_linea_apr	Valida que una linea del archivo cumpla con las validaciones Va7, Va8	linea a validar, numero de la linea	Retorna un array con la siguiente estructura: [True False][apropiacion][Porcentaje apropiacion]
9	validar_linea_ded	Valida que una linea del archivo cumpla con las validaciones Va9, Va10	linea a validar, numero de la linea	Retorna un array con la siguiente estructura: [True False][Deducccion][Porcentaje deducccion]
10				
11				
12				

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	nombre_archivo_apropiaciones	guarda el nombre de archivo de apropiaciones para validarlo posteriormente	string
13	lineas_apropiaciones	Guarda las lineas leidas del archivo de apropiaciones	string list
14	numero_lineas_apropiaciones	Guarda la cantidad de lineas del archivo de apropiaciones	int
15	nombre_archivo_deducciones	guarda el nombre de archivo de deducciones para validarlo posteriormente	string
16	lineas_deducciones	Guarda las lineas leidas del archivo de deducciones	string list
17	numero_lineas_deducciones	Guarda la cantidad de lineas del archivo de deducciones	int
18	apropiaciones	Array que guarda la informacion de apropiaciones.	array
18	deducciones	Array que guarda la informacion de deducciones.	array

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 líneas 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 líneas que cumplen con las validaciones. La longitud del array debe ser igual al numero de líneas 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	Inicio de ciclo 2, desde 1 hasta número de líneas (numero_lineas_apropiacion), variable indice deciclo: m.
10	Leer línea m, validar Va7, Va8 para la línea m. Si se cumplen las validaciones, marcar como válida la línea m. Si la línea es buena, guardarla en el arreglo apropiaciones, guardando el porcentaje de apropiacion en apropiaciones[m][0]. 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 apropiaciones 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 F8, validar_linea_apr.
11	Fin de ciclo 2
12	Inicio de ciclo 3, desde 1 hasta número de líneas (numero_lineas_deduccion), variable indice de ciclo: n.
13	Leer línea n, validar Va9, Va10 para la línea n. Si se cumplen las validaciones, marcar como válida la línea n. Si la línea es buena, guardarla en el arreglo deducciones, guardando el porcentaje de deduccion en deducciones[n][0]. 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 deducciones 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 F9, validar_linea_ded.
14	Después de haber leído todos los archivos y haberlos puesto en memoria en el array nomina, array apropiaciones y array deducciones, se definen los porcentajes para cada apropiacion y deduccion para realizar los calculos correspondientes

15	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.
16	Inicio de ciclo 4, desde 1 hasta numero_lineas_archivo, variable indice de ciclo: z
17	Leer nomina[z][1] y guardarlo en la variable local salario_base.
18	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.
19	Guardar en liquidacion[z][1] el valor de cesantias.
20	Guardar en liquidacion[z][2] el valor de intereses sobre cesantias.
21	Guardar en liquidacion[z][3] el valor de prima de servicios.
22	Guardar en liquidacion[z][4] el valor de vacaciones.
23	Guardar en liquidacion[z][5] el valor de ARL.
24	Guardar en liquidacion[z][6] el valor de salud por parte del empleador.
25	Guardar en liquidacion[z][7] el valor de pensión por parte del empleador.
26	Guardar en liquidacion[z][8] el valor de salud para el empleado.
27	Guardar en liquidacion[z][9] el valor de salud para el empleado.
28	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.
29	Guardar en liquidacion[z][11] el costo total para la empresa.
30	Guardar en liquidacion[z][12] el salario neto para el empleado.
31	Fin de ciclo 4
32	Despues de tener calculada la liquidacion, crear el archivo liquidacion.txt y guardarla en éste.
33	Inicio ciclo 5, desde 1 hasta numero_lineas_archivo, variable indice de ciclo: w.
34	Guardar en archivo linea w con el formato definido.
35	Fin ciclo 5
36	Después de finalizar el ciclo cerrar el archivo liquidacion.txt
37	
38	
39	
40	
41	
42	
43	
44	
45	

46	
47	
48	
49	
50	

¿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	Qué es un array y cómo definirlo en Python.
6	Cómo dar formato a números flotantes como strings en Python.
7	Validar argumentos de línea de comandos.
8	Hacer un ciclo for en Python.
9	Inicializar array de múltiples dimensiones en Python.
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	
3	
4	
5	
6	
7	
8	
9	
10	