

Algoritmos y Estructuras de Datos
Examen Final 17/12/2020 – Regulares 2015+ [Python y PE]

Una empresa de servicios agropecuarios necesita un programa que le permita operar con los diferentes servicios que tiene contratados. De cada **Servicio**, se tiene una clave de identificación (un número entero), la descripción o nombre del servicio (una cadena), el nombre del cliente que contrató ese servicio, el monto facturado al cliente por el servicio (un valor en coma flotante), un número entre 0 y 24 que indica el tipo de servicio (por ejemplo: 0: administración de campo, 1: siembra y cosecha de soja, etc.), y otro número pero entre 0 y 4 para indicar el mecanismo de pago (por ejemplo: 0: por transferencia bancaria, 1: por débito a la cuenta del cliente en la empresa, etc.).

En base a lo anterior, desarrollar un programa completo que OBLIGATORIAMENTE disponga al menos de dos módulos:

- En el primer módulo, definir la clase Servicio que represente al registro a usar en el programa, y las funciones básicas para operar con registros de ese tipo.
- En el segundo módulo, incluir el programa principal y las funciones generales que sean requeridas. Aplique las validaciones que considere necesarias.

El programa debe basarse en un menú de opciones para desarrollar las siguientes tareas:

1. Generar un arreglo de registros que contenga los datos de todos los servicios. Puede generarlo en forma manual o aleatoria. No se exige que el arreglo se mantenga ordenado a medida que se carga.
2. Muestre el arreglo generado, a razón de un registro por línea en la consola de salida. Este listado debe salir ordenado de menor a mayor, por clave de identificación de los servicios.
3. Mostrar un listado de los servicios cuyo mecanismo de pago sea 1 o 4, pero de tal forma que el listado salga ordenado de menor a mayor por descripción o nombre de los servicios.
4. Determine si existe en el arreglo un servicio en el que la clave de identificación coincida con el valor **c** que se carga por teclado. Si existe, muestre sus datos completos. Si no existe, informe con un mensaje. La búsqueda **debe** detenerse al encontrar el primer registro que cumpla los requisitos pedidos.
5. Determine el monto acumulado que se ha facturado para cada combinación posible de los posibles tipos de servicios con cada tipo posible de mecanismo de pago (un total de $25 * 5 = 125$ acumuladores). Muestre sólo los resultados que sean diferentes de cero.
6. Defina una segunda clase de registro con el nombre **Factura**, de forma que contenga tres campos: uno para el tipo de servicio, otro para el tipo de mecanismo de pago, y otro para el importe acumulado para ese servicio y ese mecanismo de pago. Genere un archivo binario con registros de tipo **Factura**, en el que se grabe toda la información que generó en el punto 5. Es decir, el archivo debe tener un registro cuyo campo tipo servicio valga 0, el campo tipo de mecanismo de pago valga 0, y el campo importe acumulado valga el acumulador que se calculó para el tipo 0 y el mecanismo 0 en el punto 5 anterior, y así seguir con cada combinación de tipo de servicio y mecanismo de pago que haya acumulado en el punto 5.
7. Muestre el archivo generado en el punto anterior, a razón de un registro por línea en la pantalla. Pero muestre solo los registros cuyo importe sea diferente de cero.
8. Al seleccionar esta opción, el programa debe pedir que el usuario cargue por teclado un texto que representa un reclamo realizado por un cliente. Puede asumir que el usuario cargará el texto finalizando con un punto, y que las palabras del texto se separarán entre ellas con un (y solo un) espacio en blanco. Procesar ese texto a razón de un carácter por vuelta de ciclo, y determinar cuántas palabras del texto contienen una o más veces la secuencia "mi".

Criterios generales de evaluación:

- a.) Desarrollo del programa completo, incluyendo al menos los dos módulos pedidos, el menú correctamente planteado, funciones bien diseñadas y parametrizadas (cuando sea apropiado), validaciones cuando sean aplicables y estilo de código fuente: **[máximo: 2 puntos (8.3% del puntaje)]**
- b.) Desarrollo correcto de los ítems 1 y 2: **[máximo: 4 puntos (16.6% del puntaje)]**
- c.) Desarrollo correcto del ítem 3: **[máximo: 4 puntos (16.7% del puntaje)]**
- d.) Desarrollo correcto del ítem 4: **[máximo: 4 puntos (16.7% del puntaje)]**
- e.) Desarrollo correcto del ítem 5: **[máximo: 2 puntos (8.3% del puntaje)]**
- f.) Desarrollo correcto de los ítems 6 y 7: **[máximo: 4 puntos (16.7% del puntaje)]**
- g.) Desarrollo correcto del ítem 8: **[máximo: 4 puntos (16.7% del puntaje)]**

Para aprobar el parcial, el alumno debe llegar a un **total acumulado de al menos 60% del puntaje (es decir, alrededor de 14.4 puntos acumulados)**, pero obligatoriamente debe estar desarrollado el programa, funcionando y operativo.

NOTA	PORCENTAJE	CALIFICACIÓN
1		Insuficiente
2		Insuficiente
3		Insuficiente
4		Insuficiente
5		Insuficiente
6	60% a 68%	Aprobado
7	69% a 77%	Bueno
8	78% a 86%	Muy Bueno
9	87% a 95%	Distinguido
10	96% a 100%	Sobresaliente