

## Examen de trabajos practicos - 04/06/2024

### 1. Archivos Secuenciales

Considere que se tiene un archivo que contiene información de los préstamos otorgados por una empresa financiera que cuenta con varias sucursales. Por cada préstamo se tiene la siguiente información: número de sucursal donde se otorgó, DNI del empleado que lo otorgó, número de préstamo, fecha de otorgación y monto otorgado. Cada préstamo otorgado por la empresa se considera como una venta. Además, se sabe que el archivo esta ordenado por los siguientes criterios: código de sucursal, DNI del empleado y fecha de otorgación (en ese orden).

Se le solicita definir las estructuras de datos necesarias y escribir el módulo que reciba el archivo de datos y genere un informe en un archivo de texto con el siguiente formato.

Informe de ventas de la empresa

Sucursal <número sucursal>

Empleado: DNI <DNI empleado>

Año	Cantidad de ventas	Monto de ventas
-----	--------------------	-----------------

.....	.....	.....
-------	-------	-------

Totales	<Total ventas empleado>	<Monto total empleado>
---------	-------------------------	------------------------

Empleado: DNI <DNI empleado>

...

Cantidad total de ventas sucursal: .....

Monto total vendido por sucursal: .....

Sucursal <número sucursal>

## Examen de trabajos practicos - 04/06/2024

...

Cantidad de ventas de la empresa: .....

Monto total vendido por la empresa: .....

Notas:

- \* El archivo de datos se debe recorrer solo una vez.
- \* Para determinar el año de otorgación de un préstamo, puede asumir que existe una función `extraerAño(fecha)` que, a partir de una fecha dada, devuelve el año de la misma.
- \* En la generación del archivo de texto solo interesa que aparezca la información requerida, NO es necesario que se incluyan los espacios en blanco o tabulaciones que se incluyen en el informe de modo como ejemplo.

### 2. Archivos Directos

Realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada registro tiene capacidad para dos claves. El número natural indica el orden de llegada de las mismas. Deberá explicar los pasos que realiza en cada operación y dibujar los estados sucesivos correspondientes.

1	JavaScript	10100111	2	C#	10101010
3	HTML	00111110	4	NoSQL	01101111
5	Python	01101101	6	Rust	11110000
7	SQL	01011001	8	Perl	01010111
9	Java	00110100	10	PHP	11100011