

## 2.3. Archivos Indexados

### Ejercicio 2.3.1.a

Dado un fichero secuencial de Facturas, ordenado por Nro. de Cliente y Nro. de Factura, con la siguiente estructura:

**FACTURAS** Ordenado por Nro\_Cliente, Nro\_Factura

Nro_Cliente	Nro_Factura	Fecha	Importe
-------------	-------------	-------	---------

Se desea un listado con el siguiente detalle:

Nro. Cliente	Nombre Cliente	Numero Factura

Los datos del cliente se encuentran en un fichero indexado por Nro. de Cliente, que tiene la siguiente estructura:

**CLIENTES** Indexado por Nro\_Cliente

Nro_Cliente	Nombre	DNI	CUIT	Domicilio
-------------	--------	-----	------	-----------

### Ejercicio 2.3.1.b

Dado un fichero secuencial de Facturas, ordenado por Nro. de Cliente y Nro. de Factura, con la siguiente estructura:

**FACTURAS** Ordenado por Nro\_Cliente, Nro\_Factura

Nro_Cliente	Nro_Factura	Fecha	Importe
-------------	-------------	-------	---------

Se desea un listado con el siguiente detalle:

Nro. Cliente	Nombre Cliente	Total Facturado	Cantidad de Facturas

Los datos del cliente se encuentran en un fichero indexado por Nro. de Cliente, que tiene la siguiente estructura:

#### **CLIENTES** Indexado por Nro\_Cliente

Nro_Cliente	Nombre	DNI	CUIT	Domicilio
-------------	--------	-----	------	-----------

### Ejercicio 2.3.2

Una empresa dispone de un fichero secuencial con datos de sus empleados, ordenado por número de sucursal, y categoría, con los siguientes datos:

#### **EMPLEADOS** Ordenado por Nro\_Sucursal, Categoria

Nro_Sucursal	Categoría (A,B,C)	Nombre_Empleado	Cod_Curso	Tecnico
◀				▶

Y un fichero con datos de cursos, indexado por código de curso:

#### **CURSO** Indexado por Cod\_Curso

Cod_Curso	Nombre	Fecha	Cant_Horas
-----------	--------	-------	------------

Emitir un listado informando:

1. Para cada empleado: sucursal, categoría, nombre del empleado y nombre del curso que debe realizar.
2. Por sucursal, categoría y toda la empresa:
  - a. Total empleados técnicos
  - b. Total empleados no técnicos
  - c. Total empleados

### Ejercicio 2.3.3

Los automovilistas pasan por el peaje del Puente Gral. Belgrano y deben pagar según su categoría, pero además, si ya han pasado previamente dentro del día tienen pase libre.

Teniendo en cuenta el archivo siguiente, construya el algoritmo que realice lo que corresponda: genere el comprobante, indicando el importe a pagar o emita un mensaje indicando que ya pasó anteriormente. Además indique cuales deberían ser los datos de entrada.

#### **PEAJE** Indexado por Patente, Fecha

Patente XXX-NNN	Fecha N(8)	Ult_Hora N(4)	Costo xxx,xx
-----------------	------------	---------------	--------------

#### Costo por categoría

Categoría	Costo
1	1,20
2	2,50
3	4,00
4	5,00

#### Ejercicio 2.3.4

Crear un algoritmo que simule el trabajo de una caja de supermercado. El algoritmo debe permitir imprimir el ticket de compra y realizar el descuento de stock del producto. Al generar el comprobante del ticket debe guardar los datos en los archivos TICKET y DETALLE\_TICKET (el cliente es: "consumidor final" y el NroTicket se genera automáticamente, mediante la función OBTENER\_TICKET). Archivos:

#### **PRODUCTOS** Indexado por Cod\_Prod

Cod_Prod	Nombre	Stock	Precio
----------	--------	-------	--------

#### **TICKET** Indexado por Nro\_Ticket

Nro_Ticket	Fecha	Cliente
------------	-------	---------

**DETALLE\_TICKET** Indexado por Nro\_Ticket

Nro_Ticket	Nro_Linea	Cod_Prod	Cantidad
------------	-----------	----------	----------

Comprobante:

Empresa: .....	CUIT: .... - ..... - ...	Fecha: ... / ... / ...
Cliente .....		
Producto	Cantidad	Subtotal
.....	.....	.....
.....	.....	.....
.....	.....	.....
Total		.....

**Ejercicio 2.3.5**

Para poder comprar dólares en una entidad bancaria al precio oficial (\$9,40), el beneficiario, debe tener un ingreso promedio en los últimos 12 meses equivalente a dos veces el sueldo mínimo vital y móvil (el cual en la actualidad es de \$5000 por mes). Luego, con el sueldo del mes actual, solo se permite comprar por un importe no superior al 30% del mismo.

Por ej.: si una persona tiene un sueldo de \$10.000, desde septiembre del año pasado, al dia de hoy esa persona cumple la condición para comprar y puede comprar dólares por un monto máximo de \$3.000 (equivalente a U\$S 319).

Para ello se cuenta con dos archivos indexados:

**CABECERA\_SUELDO** Indexado por DNI, Periodo, Nro\_Recibo

DNI N(8)	Periodo N(6)	Nro_Recibo N(15)	ApyNom AN(50)	Empresa AN(50)
----------	--------------	------------------	---------------	----------------

El periodo está representado por 6 caracteres numéricos dispuestos de forma de año/mes (aaaamm). El número de recibo (Nro\_Recibo) es único. Una persona puede tener varios recibos de

sueldo. La clave de este archivo es DNI, Periodo y Nro\_Recibo.

### **RECIBO** Indexado por Nro\_Recibo

Nro_Recibo N(15)	Concepto N(8)	Tipo (0..2)	Monto N(15,2)
------------------	---------------	-------------	---------------

El campo tipo puede contener los siguientes valores: 0 - Sueldo básico, 1 – Otros Ingresos, 2 – Descuentos. Para calcular el sueldo mínimo se suman el sueldo (tipo 0), y los otros ingresos (tipo 1), NO se restan los descuentos. La clave es Nro\_Recibo.

Dado el escenario descripto, se pide escribir dos algoritmos:

1. a. Que el empleado del Banco ingrese un número de documento de algún interesado en comprar dólares, y le devuelva si está habilitado o no para comprar y, en caso positivo, cuál es el monto máximo en pesos que se le autoriza.
- b. Si el interesado desea comprar, solicite el monto en pesos que destinará a la compra, el cual deberá ser descontado de su cuenta. Los datos de la cuenta están en un archivo indexado con la siguiente estructura (indexado por DNI):

DNI N(8)	Nro_Cuenta N(25)	Saldo N(15,2)
----------	------------------	---------------

- c. Los puntos a) y b) se repiten hasta que el operador (empleado del Banco) indique que desea finalizar.
2. a. Procesar peticiones de compra, de acuerdo a un archivo de entrada de peticiones, evaluando si es posible realizar la operación o no. Si no es posible, indicar cual es el motivo: 1 – No tiene el ingreso promedio suficiente, o 2 – Pide más del 30 % de su sueldo actual.

### **PETICIONES**

DNI N(8)	Cant_Soli N(15,2)
----------	-------------------

- b. El resultado de la evaluación se debe grabar en un nuevo archivo de salida con el siguiente formato:

### **SALIDA**

DNI N(8)	Cant_Soli N(15,2)	Pudo (si,no)	Error 1..2
----------	-------------------	--------------	------------

## Ejercicio 2.3.6

Una Municipalidad debe liquidar las patentes de su parque automotor para el cuarto trimestre del año e imprimir un padrón de cobros y deudas, con cortes de importe por grupo, categoría y año de fabricación.

Los archivos son:

#### **AUTOS** Ordenado por CLAVE

Grupo N(2)	Categoría 1..50	Anio_Fab N(4)	Nro_Patente N(8)	DNI AN(8)
◀				▶

#### **DEUDAS** Indexado por CLAVED

Nro_Patente N(8)	Anio_Deuda N(4)	Trimestre N(1)	Importe N(5,2)
------------------	-----------------	----------------	----------------

Para el trimestre actual, la cuota a abonar viene en el siguiente archivo:

#### **CUOTAS** Indexado por CLAVEC

Grupo N(2)	Categoría 1..50	Anio_Fab N(4)	Importe N(5,2)
------------	-----------------	---------------	----------------

Antes de imprimir el renglón correspondiente a cada nro de patente se debe verificar si existen deudas pendientes, en cuyo caso se sumaran todos los importes adeudados y se consignarán en la columna de deudas.

#### **PADRÓN**

CLAVE	DNI	APYNOM	DOMIC	DEUDA	4to TRIMESTRE
....	...	.....	....	\$ XXX,XX	\$ XXX,XX
TOTAL				\$ XXX,XX	\$ XXX,XX

#### Ejercicio 2.3.7

---

Crear un algoritmo que imprima un reporte como el que se indica, incluyendo totales por Obra Social y Clínica de liquidaciones a médicos. Los archivos que intervienen son:

**LIQUIDACIONES** Ordenado por O.S. y Clinica

O.S.	Clinica	Nro_Leg	Mes	Anio	Bruto	Descuento_AFIP	Des
◀							▶

Neto = Bruto – (Desc\_AFIP + Desc\_DGR)

**OBRAS\_SOCIALES** Indexado por Cod\_Os

Cod_OS	Nombre

**CLÍNICAS** Indexado por Cod\_Cli

Cod_Cli	Nombre

**MÉDICOS** Indexado por Nro\_Leg

Nro_Leg	ApyNom	Especialidad	DNI

Reporte:

**Obra Social:**

Clinica:

Médicos

Nro Legajo

Nombre

Neto

.....

.....

.....

.....

.....

.....

Total Clínica

.....

Total Obra Social

.....

**Ejercicio 2.3.8**

Los alumnos que desarrollaban el proyecto UTNBook (ejercicio 2.23) han decidido modificar parte de la aplicación para que esta tenga mayor interacción con el usuario. Es así que decidieron eliminar el archivo de Notificaciones, provocando así que la inserción nuevos amigos, la eliminación de amistades y la escritura en los muros de otras personas esté a cargo del usuario.

Con estas consideraciones trabajaron con el siguiente archivo:

#### **AMIGOS** Indexado por Cod\_Usuario y Cod\_Amigo

Cod_Usuario	Cod_Amigo	Fecha_Amistad	Mensaje_Muro
-------------	-----------	---------------	--------------

El proceso para agregar o eliminar amigos y escribir en los muros de estos es ahora el siguiente: El usuario, al iniciar la sesión, ingresa su Código de Usuario y el Código del amigo. Luego, puede seleccionar tres opciones distintas: Agregar Amigo; Eliminar Amigo y Escribir en el Muro.

Si selecciona Agregar Amigo se realizan las acciones necesarias para incorporar esta nueva amistad al usuario, y en caso de no poder realizarse, se muestra un mensaje por pantalla explicando el motivo que imposibilita la amistad. (por ej. Cod\_Usuario inexistente, Amistad entre Usuario y Amigo ya existente, etc.). En cambio, si selecciona Eliminar Amigo el algoritmo deberá eliminar la amistad en caso de existir, en caso contrario informar el error por pantalla.

Por último, al seleccionar Escribir en el Muro el usuario debe ingresar el mensaje que desea escribir a su amigo y luego el algoritmo deberá registrar este mensaje. De ser necesario informar algún error que pueda ocurrir (por ej. Amistad entre el Usuario y Amigo inexistente, etc).

#### Ejercicio 2.3.9

---

Pepsico S.A.I.C. desea que Ud. realice el algoritmo para poner al corriente los saldos de sus clientes y el stock de la empresa. Para ello cuentan con:

#### **CLIENTES** Indexado por ClientelID

ClienteID	Cliente_Nombre	Cliente_CUIT	Cliente_Saldo
-----------	----------------	--------------	---------------

#### **DETALLE\_CLIENTES** Indexado por Nro\_Operacion

Nro_Operacion	Fecha_Operacion	ClienteID	Importe	Factura_Numero
---------------	-----------------	-----------	---------	----------------

#### **VENTAS** Ordenado por Venta\_Numero

Venta_Número	Venta_Fecha	Factura_Número	ProductoID	Cantidad_Venta
◀				▶

**PRODUCTOS** Indexado por ProductoID

ProductoID	Stock	Producto_Detalle	Costo_Unitario

Se debe actualizar el saldo del cliente y además agregar el detalle de la compra que figura en el archivo de Ventas al archivo Detalle\_Clientes, por cada venta realizada se deberá descontar la cantidad vendida del stock.

Por final del proceso se desea saber el total de productos vendidos, y un listado de los clientes con su saldo actualizado.

**Ejercicio 2.3.10**

Una empresa dedicada a la venta de combustibles para automotores anunció una promoción mediante la cual los clientes pueden acceder a descuentos importantes. En cada carga los clientes suman 10 puntos por cada \$100 de combustible, y una vez que el cliente llega a los 100 puntos, este los puede canjear por un descuento de \$100 en el importe total. La cantidad de puntos que tiene cada cliente está almacenado en el archivo indexado PUNTOS el cual tiene el siguiente formato:

**PUNTOS** Indexado por DNI

DNI	Cant_Puntos	Ult_Carga

Los datos de cada cliente se encuentran almacenados en el archivo CLIENTE también indexado con el siguiente formato

**CLIENTE** Indexado por DNI

DNI	Ape_Nom	Edad	Ciudad

Se pide realizar el algoritmo que permita:

1. Realizar la carga interactiva de las cargas de combustible (ingresando monto), actualizando la cantidad de Puntos y la Fecha de Última Carga en el archivo PUNTOS. Si no existe el Cliente, se lo debe inicializar en ambos archivos con los datos correspondientes.

2. Permitir al cliente aplicar descuentos a las cargas realizadas siempre y cuando tenga puntos suficientes (sólo se puede canjear 100 puntos por carga).
3. Determinar la cantidad de cargas realizadas de usuarios no existentes.

### Ejercicio 2.3.11

---

Una empresa dedicada a la venta de combustibles para automotores anunció una promoción mediante la cual los clientes pueden acceder a beneficios importantes. En cada carga los clientes suman 10 puntos por cada \$100 de combustible y dependiendo del medio de pago se le aplica un factor multiplicador, el cual se encuentra en un arreglo (ya está cargado). Los medios de pago posibles son: Efectivo, Débito o Crédito. La cantidad de puntos que tiene cada cliente están almacenados en el archivo indexado PUNTOS el cual tiene el siguiente formato:

#### **PUNTOS** Indexado por DNI

DNI	Cant_Puntos	Ult_Carga
-----	-------------	-----------

Los datos de cada cliente se encuentran almacenados en el archivo CLIENTE también indexado con el siguiente formato

#### **CLIENTE** Indexado por DNI

DNI	Ape_Nom	Edad	Ciudad
-----	---------	------	--------

Se pide realizar el algoritmo que permita:

1. Actualizar el archivo PUNTOS a partir del siguiente archivo secuencial (puede haber más de una carga por cliente), agregando los clientes que no existen en los archivos anteriores cuando sea necesario

#### **CARGAS** Indexado por DNI

DNI	Medio_Pago	Ape_Nom	Fecha	Ciudad
-----	------------	---------	-------	--------

1. Determinar la cantidad de cargas mayor a \$1000
2. Informar la cantidad de Clientes nuevos cargados