

### Trabajo Práctico #1 : Metodología-DP-ECP-Módulos-ED: Registro, Arreglo y Archivo texto.

Se requiere de un proceso que realice la simulación de compra de artículos de un cliente en un supermercado. Para ello, se cuenta con los siguientes archivos de datos:

a) **Articulos.Txt**: desordenado, **máx. 10000 artículos**, conteniendo cada línea los siguientes datos:

Cód. Art. (int máx. 8 dígit.)	Cod.Rubro (short 2 dígit.)	Descripción Art. (str30)	Stock Actual (ushort, 4 dígit.)
Precio Unitario (float 6.2)	Uni.Medida (str10)	Porc. Ofertas (short x 14)	

**NOTA:** el campo **Porc. Ofertas** son 7 pares indicando pos. par: tipo descuento, pos. impar: porcentaje descuento.

b) **IndDescripArt.Txt**: ordenado por Descripción de Artículos, conteniendo:

Descripción Art.	Posición Art. (int)	Estado (bool)
------------------	---------------------	---------------

**NOTA:** el campo Estado valor 0 (falso) indica Cod.Art. baja lógica, valor 1 (true) indica Cod.Art. activo.

c) **Rubros.Txt**: ordenado por Cód.Rubro, con 15 rubros, conteniendo los siguientes datos:

Cód. Rubro	Descripción Rubro (str20)
------------	---------------------------

d) **ListaCompras.Txt**: sin orden, **máximo 100**, cada una de las líneas contiene los siguientes datos:

Descripción Art.	Cant. Requerida (short 2 dígit.)
------------------	----------------------------------

**Observación:** En cada **struct** (registro mejor dicho) se deben indicar solamente los campos indicados.

#### Se pide:

- Volcar y Generar en la memoria **RAM** estructuras estáticas (Tablas):
  - Volcar a una primer tabla todos los datos del archivo **IndDescripArt.Txt**.
  - Volcar a una segunda tabla todos los datos del archivo **ListaCompras.Txt**.
  - Volcar a una tercer tabla conteniendo: *Cód.Rubro* (con repetición) y su *Pos.Art.*, luego, ordenar por el campo *Cód.Rubro*, del archivo **Articulos.Txt**. Esta tabla contiene la misma cantidad de componentes que la tabla del archivo **IndDescripArt**. del punto 1.a).
- Procesar la **tabla de Lista de Compras** recorriendo secuencialmente y por cada *descripción* de artículo buscarlo en la **tabla de IndDescripArt**, si el estado es activado se deberá restar el stock actual, **actualizando** en el archivo el *stock actual*; se pueden presentar dos casos: 1) el stock Actual es mayor o igual al solicitado o 2) es menor en este caso se satisface parcialmente la solicitud comprada y se deberá indicar la cantidad efectiva comprada. Si el *estado* esta en baja lógica en la tabla correspondiente se deberá indicar el valor cero en el campo *cantidad comprada*.
- Emitir el ticket con igual criterio que la **Lista de Compras** de acuerdo al siguiente diseño:

#### **Datos de la Cabecera son:**

KOTTO Yo te reconozco SUC 170 XXXXXX...X 9999 XX...X C.U.I.T. 99-99999999-9 Fecha: nomdia 99/99/9999 Hora: 99:99:99 Nro. Ticket: 9999-99999999 Nro. Caja: 9999 ----- F A C T U R A - B ORIGINAL -----
--

**Datos del cuerpo son, en formato tabular (en columnas):**

4 x \$ 745.32		
Galletitas Media Tarde x 3 pack		
99999999	\$	9999.99
Jub. 4	\$	-999.99
5 x \$ 3962.04		
Cuadril novillito kgs		
99999999	\$	99999.99
Marca. 3	\$	-9999.99
3 x \$ 3845.12		
Coca-Cola 225 litros		
19967859	\$	11535.36
MercPago 6	\$	-2883.84
15 x \$ 643.23		
Galletitas Express pack x 3 gramos		
62937159	\$	9648.45
2 x \$ 4908.45		
Alfajor TERRABUSI 6 unidades pack		
99999999	\$	9999.99
Comunid. 5	\$	-9999.99
SubTot. sin descuentos....:	\$	999999.99
Descuentos por promociones:	\$	-99999.99
TOTAL	\$	999999.99

#### Datos del pie son:

Su pago con Tipo Pago:	\$ 999999.99
Su vuelto:	\$ 9.99
GRACIAS POR SU COMPRA	
Para consultas, sugerencias o reclamos comunicarse al correo infoKotto.com.ar	

**NOTA:** El cuerpo del ticket se compone de:

cant. x pre.Uni  
Descripción del art Unid.Med.  
Cód. Art. Imp.Tot.Item  
Tipo descuento nroDesc. Imp.Descto

4. Emitir el Listado de Artículos ordenado por **Cód. Rubro** según el siguiente diseño:

#### Listado de Artículos ordenados por Código de Rubro

##### Cod. Rubro: 1 LACTEOS

Cod.Art.	Descripcion	Stk.	Pre.Uni.	U.Med.	TD %	TD %	TD %	TD %	TD %	TD %	TD %	TD %
58791254	Helados bombom	25	802.65	gramos	1	20	1	20	1	15	4	15
74141296	Leche Serenisima con Vit.A+D	38	785.09	litros	6	20	3	20	2	15	4	15
5978451	yogur entero c/colchon durazno	58	2041.02	litros	2	20	3	20	0	15	2	15
3236875	Queso reggianito	23	1873.30	kilo	4	20	3	20	1	15	3	15

##### Cod. Rubro: 3 ROPA

Cod.Art.	Descripcion	Stk.	Pre.Uni.	U.Med.	TD %	TD %	TD %	TD %	TD %	TD %	TD %	TD %
----------	-------------	------	----------	--------	------	------	------	------	------	------	------	------

##### Cod. Rubro: 5 BEBIDAS CON ALCOHOL

Cod.Art.	Descripcion	Stk.	Pre.Uni.	U.Med.	TD %	TD %	TD %	TD %	TD %	TD %	TD %	TD %
80196873	Cerveza AmsTel Lager	250	3421.87	litros	2	20	2	20	6	15	2	15
13102456	Cerveza Schneider	250	936.05	gramos	1	20	3	20	5	15	1	15
78678241	Vino Toro Malbec	250	809.09	litros	2	20	2	20	4	15	2	15

##### Cod. Rubro: 8 BEBIDAS SIN ALCOHOL

Cod.Art.	Descripcion	Stk.	Pre.Uni.	U.Med.	TD %	TD %	TD %	TD %	TD %	TD %	TD %	TD %
19967859	Coca-Cola 225	81	3845.12	litros	2	20	4	10	2	15	3	10

En el bloque principal, solo se establecen las invocaciones a los módulos, y declarar las variables pertenecientes a este módulo y sus tipos de datos.

El bloque principal debe contener las siguientes acciones:

```
main() {  
    Declarar las variables utilizadas en el bloque ppal().  
  
    Abrir (Articulos,IndDescripArt,Rubros,ListaCompras);  
    VolcarArchivos(lista de parámetros que correspondan); // indicados por el grupo de trabajo.  
    ProcCompras(lista de parámetros que correspondan);  
    EmitirTicket(lista de parámetros que correspondan);  
    EmitirArt_x_Rubro(lista de parámetros que correspondan);  
    Cerrar (Articulos,IndDescripArt,Rubros,ListaCompras);  
    return 0;  
}
```

### **Observaciones, restricciones y recursos disponibles:**

Utilizar las siguientes funciones, se indican los prototipos, invocando en donde sea necesario:

- **bool LeerSuf** (modo &id, sid &id), el cual **lee una componente de datos** del archivo y los almacena en una estructura interna. La función retorna un boolean, verdadero, si la lectura fue exitosa, caso contrario, falso; *modo* puede ser *fstream* o *ifstream* según corresponda, *sid* indica un tipo de estructura de datos. Crear una función de lectura para cada archivo que se lee. Suf luego de Leer en el nombre de la función es un sufijo para el archivo que se lee, el cual debe ser reemplazado Suf por el nombre apropiado, p.e. *LeerArt*, *LeerRub*...
- void **CabeceraTicket**(int &ds) se debe invocar dentro del módulo **EmitirTicket**. El parámetro *ds* se deberá utilizar en el cuerpo del ticket, que indica el día de la semana, 1:dom., 2:lun., ... 7:sáb.
- void **PieTicket**(float impTot, float impTotDesto, float impTotConDesto) se debe invocar dentro del módulo **EmitirTicket**.
- void **OrdxBur** (tid tbl, tid card), ordena tbl *Rubros* en *Artículos* con repetición por descripción.
- void **IntCmb** (id &elem1, id &elem2), *intercambia ambos elementos*.
- void **ActLinea**(modo &id, sid id) o void **ActLinea**(modo &id, sid id, short posArt) que debe *actualizar en la línea que corresponda* el nuevo stock actual. Se debe grabar c/u. de los datos contenidos en la línea.
- int **BusBinVec**(tbl id, tid clv, tid ult) que busca el valor *clv* en la tabla de *IndDescrip* y **que** retorna la posición encontrada o -1 si no se encontró.
- string **Replicate**(char car, unsigned n), retorna una cadena con n veces car.
- long **GetTime**(int hora,int min, int seg). La función retorna la hora larga, como un solo número en el formato hhhmmss. Además en sus parámetros devuelve la hora, los min. y los segundos .
- long **GetDate**(int year,int mes,int dia,int diaSem). La función retorna la fecha larga, como un solo número en el formato aaaammdd. Además en sus parámetros devuelve el año, el mes, el día y el día de la semana. Esta función se encuentra en OBTENER LA FECHA Y HORA DEL SISTEMA del apunte del prof. Hugo Cuello Teoría y Práctica del Lenguaje C/C++ ANEXOS

**Espacio en disco:** Solo para generar el archivos de salida **Ticket.Txt** en formato texto.

**Espacio para arrays y registros:** Lo necesario que requiera este proyecto.

**Espacio en memoria dinámica:** 0 bytes.

**Accesos a los archivos:** un solo recorrido secuencial, para leer en *Artículos*, *IndDescripArt* y en *ListaCompras*. Además acceso al azar para leer y/o grabar en *Articulos* en 3 (tres) instancias; una para actualizar el saldo actual, dos para emitir el ticket y 3 para emitir listado de rubros .

**Bloque Principal:** sólo invocaciones a módulos, según lo establecido anteriormente.

**Paradigma de Programación:** Solo se aceptará el Paradigma Imperativo Procedural, Programación Estructurada y Modular.

**Optimización:** dado que el uso de ciclos afecta el tiempo de ejecución de un proceso, se evaluará la eficiencia en el uso de los mismos.

Utilizar nombres significativos para los identificadores, dibujos para las estructuras de datos a utilizar, rotulando cada elemento, tamaño, breve leyenda de cómo se generan y estado inicial, respetar esos nombres para utilizarlos en el algoritmo. Preparar una muestra de datos para los archivos de datos e imprimirla, para ejecutar el programa. En el disco solo contendrán los archivos “**TP1V\_K1\_\_G\_-Apellido Nombre.cpp**”, y los archivos de datos indicados anteriormente como así también el archivo de salida todos ubicados en la carpeta raíz del disco. Ejemplo: TP1V1\_K1023G3\_PEREZ JUAN.CPP

### **Cada grupo debe crear su propia muestra de datos para los archivos, artículos de un supermercado.**

Se deben utilizar constantes con nombres para indicar cantidades.

El Trabajo Práctico deberá ser entregado de acuerdo a las pautas indicadas más abajo, el cual se aprobará si reúne los requerimientos solicitados en **tiempo y forma (tres fechas máximas)**: A: Aprobado, N: No Aprobado. **(Se debe respetar el orden indicado a continuación)**:

1. Entregar en carpeta tamaño A4 de tapa transparente y con sujetador de gancho perfectamente alineadas para las hojas lo siguiente: **(no se aceptan hojas sueltas ni otro tipo de carpeta)**
2. Carátula con los datos de los integrantes del TP, la cantidad de alumnos por grupo del mismo curso se determinará en clase no superando 5 grupos como máximo.
3. Esta misma hoja que establece el enunciado del problema a resolver. Cada grupo elegirá un líder del proyecto, que será el responsable de realizar las entregas del TP. Si un líder abandona la cursada, se deberá elegir otro líder.
4. Diseñar las estructuras de Datos graficándolas indicando con rótulos apropiados, cada elemento, su tamaño en bytes y las variables utilizadas. Las estructuras de datos a graficar son:
  - a. *El diseño de los registros de cada uno de los archivos.*
  - b. *Otras estructuras de datos que considere necesarias para poder realizar el proceso solicitado. Algunas serán explicadas en clase, como complemento a este documento.*
5. Graficar el Bloque Principal.
6. Graficar cada uno de los módulos –funciones- a utilizar, cabecera y cuerpo.
7. Construir una muestra de datos, para los archivos de datos, la cual se la utilizará para probar el Algoritmo. En la hoja impresa a entregar debe haber rótulos apropiados, pero, NO en los archivos de Datos, el cual contendrán solamente, los datos. Ver detalle del formato más abajo.
8. Emitir según la muestra establecida, los resultados esperados, siempre acompañada de los rótulos apropiados, según formato de salida indicados anteriormente.
9. Codificación del Algoritmo completo en el Lenguaje C++, emitiendo números de líneas. Usar Code-Blocks. Las primeras líneas serán de comentario indicando: **Nombre del programa, fecha entrega, Nro. versión, breve comentario del objetivo del programa, datos del curso, nombre del día, turno, nro. del grupo e integrantes (Apellido, Nombre). Nombre del compilador: Borland C++ V.5.5**

### **Formato del archivo de datos Articulos.Txt**

Se deberá ajustar la escritura de los datos respetando a raja tabla este formato en donde cada columna representa un dato y su tipo de dato y ancho:

Cada columna representa de izquierda a derecha lo siguiente:

**codArt codRub descrip stkAct preUni UniMed T % T % ... T %** (7 veces)

9(8) 99 X(30) 9(4) 9(5).99 9(10) 9 99 9 99 ... 9 99

Los números enteros o reales ajustado a la derecha, las cadenas ajustadas a la izquierda. Si una cadena tiene menos caracteres de los indicados, se deberá rellenar con espacios en blanco a derecha.

### **Ejemplos:**

```
12406297 23 Naranjas de jugo      54 1526.28 gramos  2 20 3 10 5 10 1 25 7 15 6 25 2 10
345678 145 Leche Cindor          2451 630.86 cc    1 5 4 15 4 15 3 20 4 10 6 25 2 15
7451932 1 Fanta 2.25             153 3542.09 litros 5 15 3 20 5 20 4 15 2 20 6 25 3 10
```

Para los ingresos de los datos de cada archivo de texto, utilizar el editor Code-Blocks.

Se utilizará este formato semejante a los demás archivos.

La salida de los resultados debe estar dirigida a un archivo de texto con el nombre **Ticket.Txt** se debe utilizar la sentencia **freopen**, para redirigir la salida de la pantalla a archivo de texto al utilizar **cout**.

La cantidad de datos de muestra para cada archivo deberán ser los siguientes:

Articulos.Txt: entre 40 y 45 líneas.

IndDescripArt.Txt: entre 40 y 45 líneas, con la misma cantidad que en Artículos.Txt.

Rubro.Txt: 15 líneas, si o si.

ListaCompras.Txt: entre 25 y 30 líneas.

Cada línea representa un conjunto de datos que reúne todos los datos solicitados para cada archivo.

En las muestras de datos contemplar, todos los casos posibles, con respecto a las cantidades:

- Stock insuficiente para algunos artículos.
- Sin Stock, es decir cero.
- Estado cero (*false*) algunos artículos y en Estado uno (*true*) varios.

Las opciones para las promociones son 7:

**SinPromo, Promo, Marca, Jub., Comu., MercPago, ANSES**, para saber si un artículo tiene o no promoción compararlo con la cadena "**SinPromo**". Utilizar una selección múltiple para asignar el nombre de la promo a la variable. Se aclara que el tipo de promo y su porcentaje se utilizará un array de tipo *short* con 14 (catorce) posiciones en donde las posiciones pares serán los tipos de promociones y las posiciones impares indicarán el porcentaje.

**Se deben leer todos los datos de cada línea en los archivos, sin importar si algunos datos no se utilicen en el proceso.**

Dar nombres de identificadores representativos a su uso, es decir, con significado.

Cada nueva entrega además del nombre indicado para el archivo del código en C++ irá acompañado de la versión entregada, iniciando la primera entrega con el sufijo V1, luego la segunda entrega V2, y así sucesivamente.

**(S.E. ú O.)**