


Redictado Taller de programación 2020

CLASE 5

Merge de Listas –Caso Especial



```
Program HolaMundo;  
Begin  
  writeln('Hola mundo');  
end.
```



Supongamos que Pepe y Moni mantienen sus gastos ordenados por tipo de consumo (Supermercado, Ropa, Impuestos, etc)



Gastos de Moni



Gastos de Pepe



Si quisieran conocer cuánto dinero han gastado, en total, en cada tipo de consumo



Gastos de Moni

Merge

Un caso ESPECIAL



Totales por Consumo



Gastos de Pepe



¿Sirve aplicar el mismo proceso de Merge que vimos?



Gastos de Moni

Merge

Un caso ESPECIAL



Gastos de Pepe



NO, no obtengo totales de cada tipo de consumo



Comenzamos a trabajar...

Dame el monto total
gastado de un tipo de
consumo

Antes de darle el total
debo sumar los montos de
un mismo consumo!!!

CASO ESPECIAL

Totales por
Consumo



Gastos de Moni

Impuestos 500

Peluquería 40

Peluquería 200

Ropa 700

Gastos de Pepe

Impuestos 300

Peluquería 50

Salidas 300

Salidas 200



Comenzamos a trabajar...

Totales por
Consumo

Impuestos 800



Impuestos 500

Impuestos 300

Gastos de Moni

Peluquería 40

Peluquería 200

Ropa 700

Gastos de Pepe

Peluquería 50

Salidas 300

Salidas 200

Antes de darle el total
debo sumar los montos de
un mismo consumo!!!



Si pensamos en el proceso de
sumar los montos de un
mismo consumo...

MoniBuscaMinimo

mientras (tipo consumo mínimo es válido)

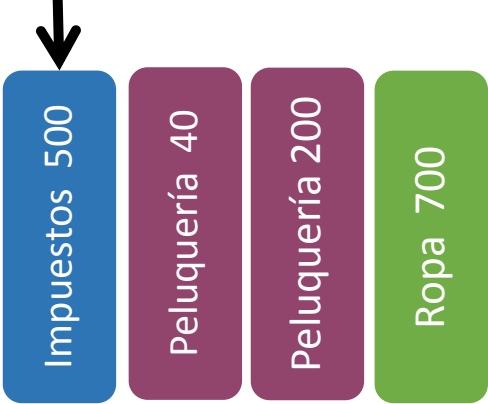
```
guardar tipo del consumo "actual" (mínimo)
inicializar monto total del consumo "actual"
mientras (tipo consumo mínimo = tipo consumo "actual")
    sumar el gasto al monto total
MoniBuscaMinimo
```

PepeGuardaMinimo (tipo "actual" y monto total)





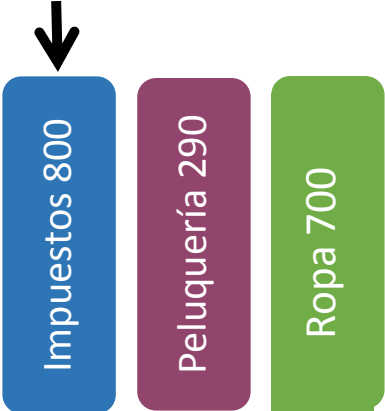
Lista 1



Lista 2



Lista Nueva



MoniBuscaMinimo

mientras (tipo consumo mínimo es válido)

```
guardar tipo del consumo "actual" (mínimo)
inicializar monto total del consumo "actual"
mientras (tipo consumo mínimo = tipo consumo "actual")
    sumar el gasto al monto total
MoniBuscaMinimo
```

PepeGuardaMinimo (tipo "actual" y monto total)

| Mínimo | Impuestos 500 | | Impuestos 300 | | Peluquería 40 | Peluquería 200 | Peluquería 50 | Ropa 700 | VALOR ALTO |
|--------------------|---------------|----------------|---------------|--|---------------|----------------|----------------|----------|------------------|
| Tipo Actual | Impuestos | | Peluquería | | | | Ropa | | |
| Monto Total Actual | 0 | 500 | 800 | | 0 | 40 | 240 | 290 | 0 700 |



PepeGuardaMinimo

es tan solo un agregar atrás en la lista nueva



MoniBuscaMinimo

tiene una solución similar a la del
módulo DeterminarMinimo ya visto

¿En qué se diferencian?



Actividad en máquina

ACTIVIDAD 1

Suponiendo que 4 integrantes de una familia llevan sus gastos de manera similar a Pepe y Mónica.

Realizar un **merge acumulador** de 4 listas que contienen los gastos ordenados por tipo de consumo de cada integrante.



Actividad en máquina

Tareas:

- a) Crear el programa **ProgramaAcumulador.pas**
- b) Generar las 4 listas que contienen los gastos ordenados por tipo de consumo. De cada consumo se conoce el tipo (entero), la fecha y el monto del gasto.
- c) Mostrar las listas generadas.

IMPORTANTE: Para realizar esta tarea deben reutilizar los módulos implementados anteriormente.



Actividad en máquina

Tareas:

d) Adaptar los tipos de datos utilizados por el **procedure** **AgregarAlFinal** ya trabajado.



Actividad en máquina

Tareas:

e) Adaptar los tipos de datos utilizados en el **procedure Determinar_minimo** ya visto.



Actividad en máquina

Tareas:

f) Realizar un módulo que haga un **merge** entre las 4 listas de consumos, acumulando el monto total de cada consumo. Utilizar los módulos implementados en d) y e).

- El módulo se llamará **mergeAcumulador**



Actividad en máquina

Tareas:

g) En el programa principal invocar al módulo **mergeAcumulador** enviándole las 4 listas creadas previamente. Imprimir en pantalla cada tipo de consumo y su monto total gastado.

Enviar a través de la Mensajería de Ideas, el archivo `ProgramaAcumulador.pas` al docente asignado al grupo.





Actividad en máquina

ACTIVIDAD 2

Retomar la Actividad 2 de la clase anterior para:

- a) Generar una nueva lista que totalice la cantidad de ventas de cada producto de la perfumería. Esta lista debe estar ordenada por código de producto de forma ascendente. Deben utilizarse las cinco listas ya creadas de esa actividad.
- b) Incorporar un módulo recursivo que devuelva el código de producto con menor cantidad vendida, a partir de la lista generada en a).

