



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





Gastos de Moni



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



Vierge
Un caso ESPECIAL

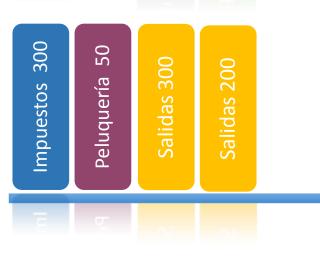
Gastos de Moni

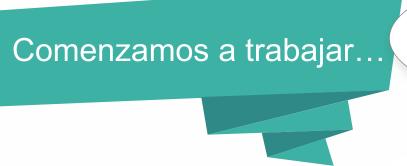
Impuestos 500
Peluquería 40
Peluquería 200
Ropa 700

Totales por Consumo



Gastos de Pepe





Dame el monto total gastado de un tipo de consumo Antes de darle el total debo sumar los montos de un mismo consumo!!!

500

Impuestos

ÉSPECIAL

700

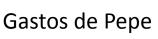
Ropa



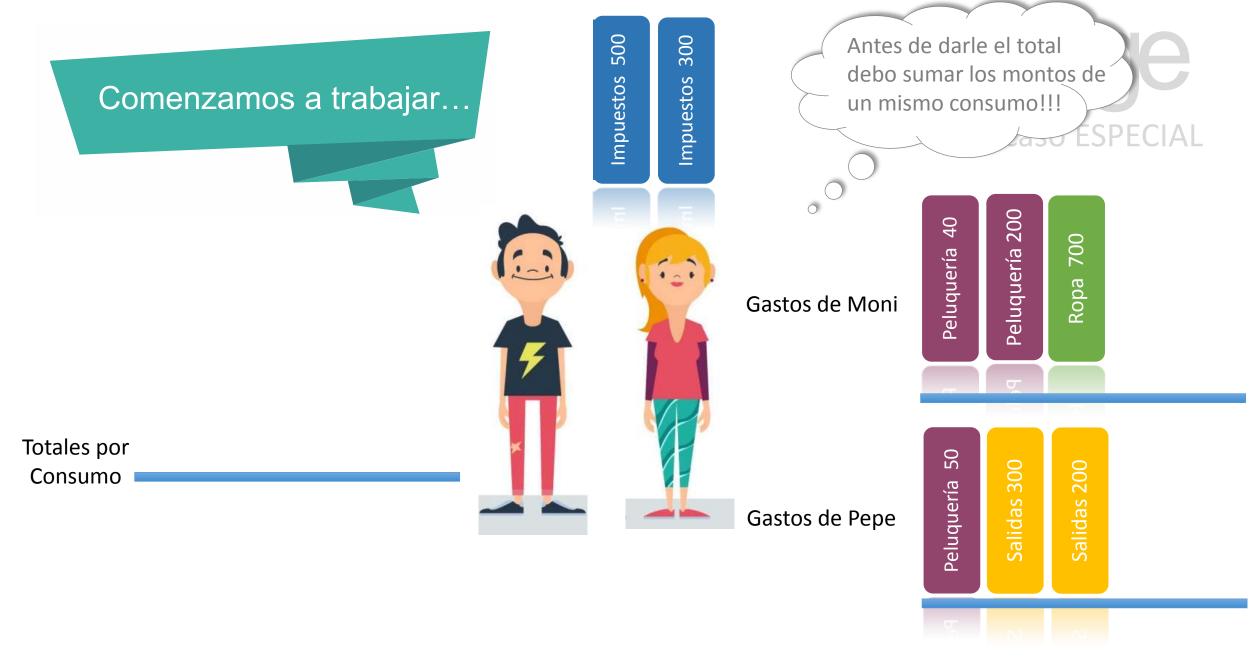
Gastos de Moni

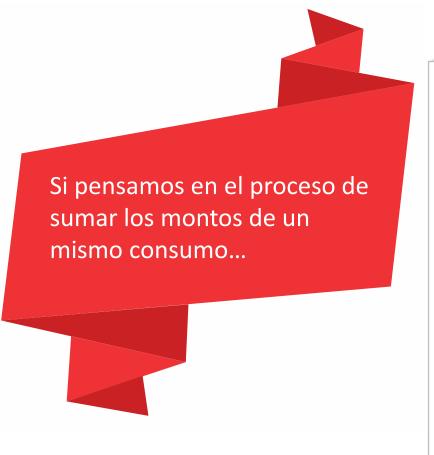
Peluquería 40 Peluquería 200

Totales por 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)



PepeGuardaMinimo

es tan solo un agregar atrás en la lista nueva



MoniBuscaMinimo

tiene una solución similar a la del módulo DeterminarMinimo ya visto





Objetivo de la actividad:

Realizar un **merge acumulador** de 2 listas que contienen los gastos ordenados por tipo de consumo.

Tareas:

- a) Crear el programa Programa Acumulador
- b) Generar las dos listas que contienen los gastos ordenados por tipo de consumo. De cada consumo se conoce el tipo de consumo, la fecha y el monto del gasto.
- c) Mostrar las listas generadas.

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

Tareas:

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



Tareas:

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



Tareas:

- f) Realizar un módulo que haga un **merge** entre las 2 listas de consumos, acumulando el monto total de cada consumo. Utilizar los módulos implementados en d) y e).
 - El módulo se llamará mergeAcumulador



Tareas:

e) En el programa principal invocar al módulo **mergeAcumulador** enviándole las 2 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 Adicional:

Una perfumería tiene cuatro sucursales. Cada sucursal dispone en papel de la información de sus ventas.

Implementar un programa que:

- a) Genere cuatro listas ordenadas por código de producto de forma ascendente con las ventas de cada sucursal de la perfumería. De cada venta se lee fecha de venta, código del producto y cantidad vendida. La lectura finaliza con código de producto igual a -1.
- b) Genere 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.