

# **PROGRAMACIÓN I**

AÑO 2025

1

## Corte de Control

# EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX`. Se pide informar la cantidad de inmuebles por Localidad.

Localidad La Plata  
Tipo `Depto`  
cantHab:2  
cantBaños 1  
precio 15.200

Localidad Gonnet  
Tipo `Casa`  
cantHab:3  
cantBaños 2  
precio 23.000

Localidad La Plata  
Tipo `Casa`  
cantHab:5  
cantBaños 3  
precio 55.400

Localidad La Plata  
Tipo `Casa`  
cantHab:1  
cantBaños 2  
precio 18.000

Localidad XXX



**La Plata 3, Gonnet 1**

**¿Cómo  
hago?**



¿Qué cambio y consideraciones  
debo tener si quiero informar la  
cantidad de propiedades por  
localidad?

# EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX`. Se pide informar la cantidad de inmuebles por Localidad.

Localidad La Plata  
Tipo `Depto`  
cantHab:2  
cantBaños 1  
precio 15.200

Localidad La Plata  
Tipo `Casa`  
cantHab:5  
cantBaños 3  
precio 55.400

Localidad La Plata  
Tipo `Casa`  
cantHab:1  
cantBaños 2  
precio 18.000

Localidad Gonnet  
Tipo `Casa`  
cantHab:3  
cantBaños 2  
precio 23.000

Localidad XXX



**La Plata 3, Gonnet 1**

**NECESITO que los datos  
vengan “organizados” por  
el criterio que se quiere  
informar**

# EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX`. Se pide informar la cantidad de inmuebles por Localidad. **Los datos se leen organizados por localidad.**

Localidad La Plata  
Tipo `Depto`  
cantHab:2  
cantBaños 1  
precio 15.200

Localidad La Plata  
Tipo `Casa`  
cantHab:5  
cantBaños 3  
precio 55.400

Localidad La Plata  
Tipo `Casa`  
cantHab:1  
cantBaños 2  
precio 18.000



**La Plata 3**

Localidad Gonnet  
Tipo `Casa`  
cantHab:3  
cantBaños 2  
precio 23.000



**Gonnet 1**

**¿Cómo  
hago?**

Localidad XXX

# EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuya localidad es `XXX`. Se pide informar la cantidad de inmuebles por Localidad. **Los datos se leen organizados por localidad.**

```
Leo un inmueble (inmu)
```

```
Mientras (no sea el último) do
```

```
begin
```

```
    guardo la localidad actual
```

```
    inicializo contador de la localidad (cantI)
```

```
    while (no sea el ultimo) y (sea la misma localidad) do
```

```
        begin
```

```
            cuento un inmueble mas (cantI)
```

```
            leo un inmueble (inmu)
```

```
        end;
```

```
        informo cantidad de lugares por localidad (cantI)
```

```
    end;
```

```
End.
```

# EJERCITACIÓN



```
Program uno;  
Type  
inmueble = record  
    tipo: string;  
    cantHab: integer;  
    cantBaños: integer;  
    precio: real;  
    localidad: string;  
end;
```

```
Procedure leerInmueble (var i: inmueble);  
begin  
end;
```

```
Var  
    inmu: inmueble;  
    locActual: string;  
    cantI: integer;
```

¿Por qué se  
inicializa cantI en  
ese lugar?

```
Begin  
    leerInmueble (inmu);  
    while (inmu.localidad <> `XXX`) do  
        begin  
            locActual := inmu.localidad;  
            cantI := 0;  
            while (inmu.localidad = locActual) do  
                begin  
                    cantI := cantI + 1;  
                    leerInmueble (inmu);  
                end;  
            write (`La cantidad de inmuebles es`, cantI);  
        end;  
    End.
```

¿Es correcto leer una  
sola vez dentro del  
segundo while?

¿Qué cambio si quiero imprimir  
el nombre de la localidad?



# EJERCITACIÓN



```
Begin
  leerInmueble (inmu);
  while (inmu.localidad <> `XXX`) do
    begin
      locActual:=inmu.localidad;
      cantI:=0;
      while (inmu.localidad = locActual) do
        begin
          cantI:= cantI + 1;
          leerInmueble (inmu);
        end;
      write (`La cantidad de inmuebles es`, cantI,
        `y la localidad es`, inmu.localidad);
    end;
  End.
```

```
Begin
  leerInmueble (inmu);
  while (inmu.localidad <> `XXX`) do
    begin
      locActual:=inmu.localidad;
      cantI:=0;
      while (inmu.localidad = locActual) do
        begin
          cantI:= cantI + 1;
          leerInmueble (inmu);
        end;
      write (`La cantidad de inmuebles es`, cantI,
        `y la localidad es`, locActual);
    end;
  End.
```



**¿La dos soluciones  
cumplen lo pedido?**

**¿Qué cambios debo realizar  
para informar la localidad  
con más inmuebles?**

# EJERCITACIÓN



Escriba un programa que lea inmuebles hasta leer un inmueble cuyo precio es -1. Se pide informar la cantidad de inmuebles por Localidad. **Los datos se leen organizados por localidad.**

Localidad La Plata  
Tipo `Depto`  
cantHab:2  
cantBaños 1  
precio 15.200

Localidad La Plata  
Tipo `Casa`  
cantHab:5  
cantBaños 3  
precio 55.400

Localidad La Plata  
Tipo `Casa`  
cantHab:1  
cantBaños 2  
precio 18.000

Localidad Gonnet  
Tipo `Casa`  
cantHab:3  
cantBaños 2  
precio 23.000

precio -1



**La Plata 3, Gonnet 1**

# EJERCITACIÓN



```
Begin
leerInmueble (inmu);
while (inmu.precio <> -1) do
  begin
    locActual:=inmu.localidad;
    cantI:=0;
    while (inmu.precio <> -1) and
      (inmu.localidad = locActual) do
      begin
        cantI:= cantI + 1;
        leerInmueble (inmu);
      end;
    write (`La cantidad de inmuebles es`, cantI,
      `y la localidad es`,locActual);
  end;
End.
```

**Válida**

*¿Qué cambios debo realizar para  
informar la localidad con más  
inmuebles?*

```
Begin
leerInmueble (inmu);
while (inmu.precio <> -1) do
  begin
    locActual:=inmu.localidad;
    cantI:=0;
    while (inmu.localidad = locActual) do
      begin
        cantI:= cantI + 1;
        leerInmueble (inmu);
      end;
    write (`La cantidad de inmuebles es`, cantI,
      `y la localidad es` inmu.localidad);
  end;
End.
```

**Inválida ¿Por qué?**

# Ejercitación



Un supermercado requiere el procesamiento de los productos que dispone. De cada producto se conoce su código, nombre, marca, stock y precio unitario. El procesamiento finaliza con el código -1 y los productos de igual marca se leen consecutivamente.

Se requiere informar:

- La cantidad en stock de productos de cada marca

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				

*¿Qué significa que los productos de igual marca se leen consecutivamente?*

**Resultados:**

- SanCor 350
- Ala 20
- Matarazzo 70
- Quilmes 100



Un supermercado requiere el procesamiento de los productos que dispone. De cada producto se conoce su código, nombre, marca, stock y precio unitario. El procesamiento finaliza con el código -1 y los productos de igual marca se leen consecutivamente.

Se requiere informar:

 La cantidad en stock de productos de cada marca.

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				

```
Leer Datos del producto
mientras haya productos en el super
    inicializar total por marca
    mientras sea la misma marca
        actualizar total por marca
    leer otro producto
fin mientras
Mostrar total por marca
fin mientras
```

¿Analizamos los datos?

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				

Leer Datos producto

mientras haya productos en el super

    inicializar total por marca

    mientras sea la misma marca

        actualizar total por marca

        leer otro producto

    fin mientras

    Mostrar total por marca

fin mientras

#### Type

```
cadena15 = string [15];
```

```
producto= Record
```

```
    codigo: integer;
```

```
    nombre: cadena15;
```

```
    marca: cadena15;
```

```
    stock: integer;
```

```
    precio: real
```

```
End;
```

```

Program otro_corte;
type ...
var prod: producto;
    cant: integer;
begin
  LeerProducto (prod);
  While (prod.codigo <> -1) do
  begin
    cant:= 0;
    marcaActual:= prod.marca;
    while (marcaActual = prod.marca) do
    begin
      cant := cant + prod.stock;
      LeerProducto (prod);
    end;
    write (marcaActual, ' ', cant);
  end;
end.

```

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				

¿Problema...?

```
Program otro_corte;
type ...
var prod: producto;
    cant: integer;
begin
  LeerProducto (prod);
  While (prod.codigo <> -1) do
  begin
    cant:= 0;
    marcaActual:= prod.marca;
    while (prod.codigo <> -1) and (marcaActual = prod.marca) do
    begin
      cant := cant + prod.stock;
      LeerProducto (prod);
    end;
    write (marcaActual, ' ', cant);
  end;
end.
```

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				



# Ejercitación

Si ahora el problema nos pide, además, que informe:

- Los nombres de los productos con stock en cero.

Los códigos de los productos que tienen exactamente dos dígitos iguales a 5.

La cantidad de productos cuyo precio es mayor a \$40.

Codigo	Nombre	Marca	Stock	Precio
1000	Leche	SanCor	100	20
1100	Yoghurt	SanCor	200	25
5500	Manteca	SanCor	50	30
5055	Detergente	Ala	0	55
4500	Jabón en Polvo	Ala	20	200
2400	Fideos	Matarazzo	35	30
3000	Ravioles	Matarazzo	35	80
5250	Cerveza	Quilmes	100	50
-1				

## Resultados:

- Nombres productos con stock en cero: Detergente
- Códigos con dos dígitos iguales a 5: 5500, 5250
- Cantidad de productos con precio mayor a \$40: 4



Un supermercado requiere el procesamiento de los productos que dispone. De cada producto se conoce su código, nombre, marca, stock y precio unitario. El procesamiento finaliza con el código -1 y los productos de igual marca se leen consecutivamente. Se pide además que informe:

- Stock por marca
- Los nombres de los productos con stock en cero.
- Los códigos de los productos que tienen exactamente dos dígitos iguales a 5.
- La cantidad de productos cuyo precio es mayor a \$40.

**inicializar contador > \$40**

Leer Datos producto

mientras haya productos en el super

    inicializar total por marca

    mientras sea la misma marca

        actualizar total por marca

**Si stock=0 entonces mostrar nombre**

**Si código tiene 2 dígitos 5 entonces mostrar código producto**

**Si precio > 40 entonces aumentar contador > \$40**

    leer otro producto

fin mientras

Mostrar total por marca

fin mientras

**Mostrar contador > \$ 40**

**¿Qué módulos podemos implementar?**

**¿Escribimos el programa en Pascal?**

**inicializar contador > \$40**

Leer Datos producto

mientras haya productos en el super

    inicializar total por marca

    mientras sea la misma marca

        actualizar total por marca

**Si stock=0 entonces mostrar nombre**

**Si código tiene 2 dígitos 5 entonces mostrar código producto**

**Si precio > 40 entonces aumentar contador > \$40**

    fin mientras

    Mostrar total por marca

fin mientras

**Mostrar contador > \$ 40**

Un supermercado requiere el procesamiento de los productos que dispone. De cada producto se conoce su código, nombre, marca, stock y precio unitario. El procesamiento finaliza con el código -1 y los productos de igual marca se leen consecutivamente. Se pide además que informe:

- Stock por marca
- Los nombres de los productos con stock en cero.
- Los códigos de los productos que tienen exactamente dos dígitos iguales a 5.
- La cantidad de productos cuyo precio es mayor a \$40.

```
{programa principal}
var
  prod:producto;
  totalst, c40:integer;
  aux:cadena10;

begin
  c40:=0;
  leerProducto(prod);
  while (prod.codigo<>-1) do begin
    totalst:=0;
    aux:=prod.marca;
    while (prod.codigo<>-1) and (aux=prod.marca) do begin
      totalst:= totalst+prod.stock;
      if (prod.stock=0) then writeln (prod.nombre);
      if (digitos5(prod.codigo)) then writeln (prod.nombre);
      if (prod.precio>40) then c40:= c40+1;
      leerProducto(prod)
    end;
    writeln ('El total stock marca ', aux, ' es: ', totalst);
  end;
end.
```

```

program CortedeControl;
Type cadena10=string[10];
        producto=record
                codigo:integer;
                nombre:cadena10;
                marca:cadena10;
                stock:integer;
                precio:real;
        end;
procedure leerProducto(var p:producto);
begin
    readln(p.codigo);
    if (p.codigo<>-1) then begin
        readln(p.nombre);
        readln(p.marca);
        readln(p.stock);
        readln(p.precio);
    end;
end;
function digitos5 (num:integer): boolean;
var cant:integer;
begin
    cant:=0;
    while (num<>0) and (cant<3) do begin
        if (num mod 10 = 5) then cant:= cant+1;
        num:= num div 10;
    end;
    digitos5:= cant=2;
end;

```

```

{programa principal}
var
    prod:producto;
    totalst, c40:integer;
    aux:cadena10;

begin
    c40:=0;
    leerProducto(prod);
    while (prod.codigo<>-1) do begin
        totalst:=0;
        aux:=prod.marca;
        while (prod.codigo<>-1) and (aux=prod.marca)
        do begin
            totalst:= totalst+prod.stock;
            if (prod.stock=0)
            then writeln (prod.nombre);
            if (digitos5(prod.codigo))
            then writeln (prod.nombre);
            if (prod.precio>40)
            then c40:= c40+1;
            leerProducto(prod)
        end;
        writeln ('Total stock marca ', aux, ' es: ', totalst);
    end;
end.

```