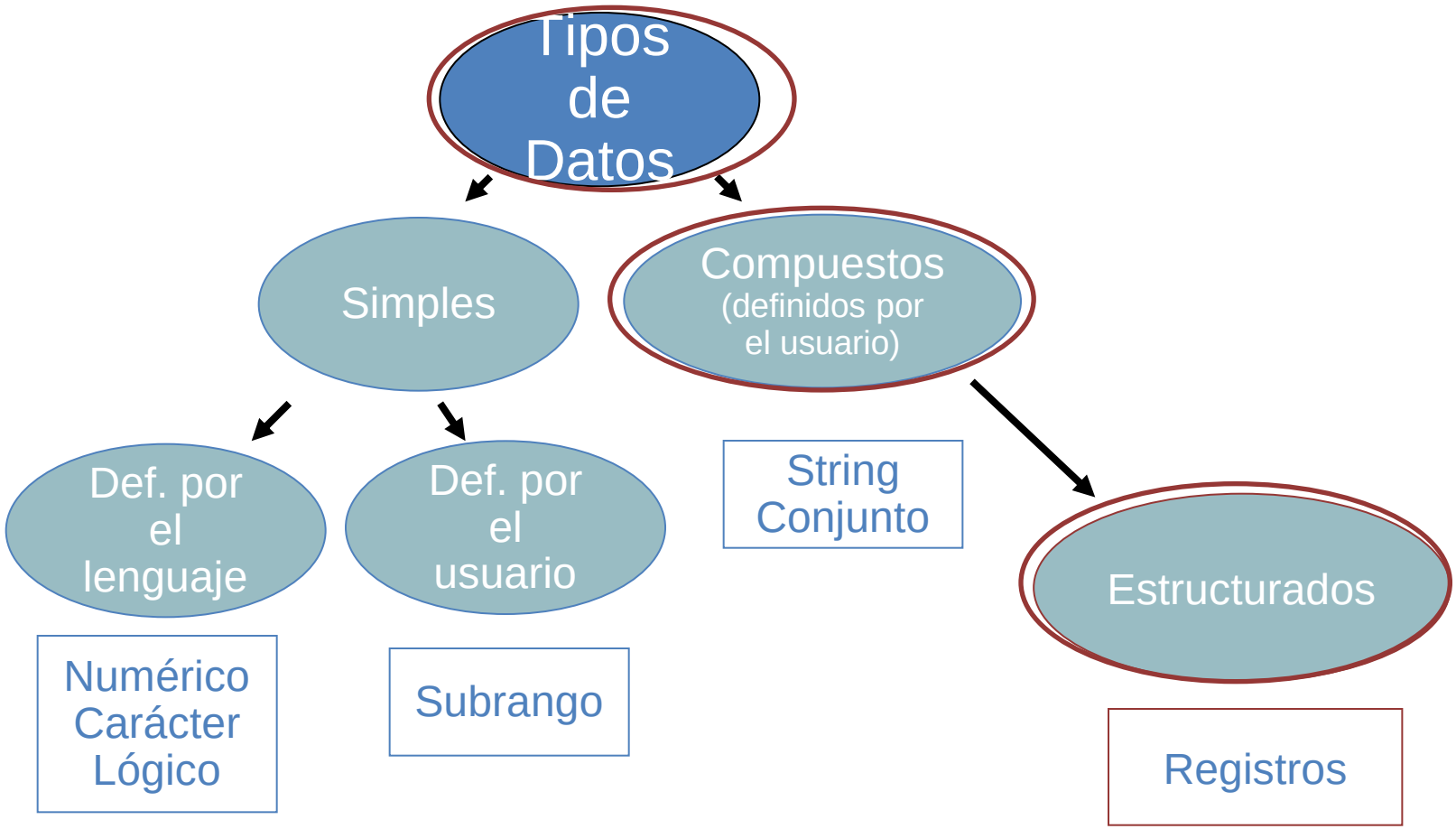
Programación I

TEMAS de la CLASE

1. Tipo de dato REGISTRO
   - Concepto
   - Declaración en Pascal
   - Ejercitación
2. Corte de control
   - Ejercitación con registros

Trabajaremos con el tipo de dato estructurado REGISTRO



Tipo de dato Registro – Características

Un registro es una estructura de datos que cumple con:

- Los valores pueden ser de diferentes tipos, esto convierte a un registro en una estructura de datos **heterogénea**.
- El almacenamiento ocupado por un registro es fijo, por esto, un registro es una estructura **estática**.
- El **acceso** a sus componentes (campos) es **directo**. Debe referenciarse a través de su nombre.

Nombre
DNI
Fecha Nac.
NroLegajo
Sexo
Sueldo
Antigüedad

Nombre
Nro. Alumno
Datos Personales
Materias que cursa
Materias aprobadas

Origen
Destino
Fecha envío
Mensaje

Código
Marca
Nombre
Precio
Fecha de vencimiento

Tipo de dato Registro – Declaración en Pascal

- Se identifica el nombre del tipo como registro (RECORD).
- Se especifica el nombre y tipo de los

campos individuales que componen el tipo. La

lista de campos se encierra entre las palabras claves record y end.

- Cada uno de los campos tiene un identificador. Los campos pueden ser nombrados individualmente, como variables ordinarias.
- Los campos pueden ser de cualquier tipo predefinido o definidos por el usuario (estáticos).

```
Type
 nombre = record
              Campo1: tipoA;
              Campo2: tipoB;
              .....
          end;
Var
 nombre-variable:nombre;
```

Tipo de dato Registro – Declaración en Pascal

Código
Marca
Nombre
Precio
Fecha de vencimiento

Observemos el campo fecha

```
Type
 cadena15 = string [15];


 producto= Record
             codigo: integer;
          Nombre: cadena15;
             Marca: cadena15;
             FechaVenc?????
             Precio: real;
            End;
Var
  prod: producto;
```

Tipo de dato Registro – Declaración en Pascal

Código
Marca
Nombre
Precio
Fecha de

El tipo fecha puede ser definido como un registro

```
Type
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
```

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;

Var
  prod: producto;
```

¿Cuál es la ocupación en memoria para prod?

Tipo de dato Registro – Acceso a los campos

- Para acceder a los campos de un registro, se necesita especificar tanto el nombre del registro como el del campo que interesa.
- Esto se denomina calificar al campo.
- En Pascal se hace de la siguiente forma:

Nombre-Registro.nombre-campo

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
              codigo: integer;
              nombre: cadena15;
              marca: cadena15;
              fechaVenc: fecha;
              Precio: real;
             End;

Var
  prod: producto;
```

Prod.nombre

Prod.precio

Prod.fechaVenc.dia

prod.marca

Operaciones aplicadas a los campos de un registro

- Dado que el Registro es un tipo de dato estructurado, las operaciones deberán aplicarse a cada uno de los campos que lo componen.
- Como los campos de un registro son variables de algún tipo, las operaciones posibles sobre un campo son las permitidas para el tipo de dato correspondiente.

- **Asignación**
- **Comparación**
- **Lectura/escritura**

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;
Var prod: producto;
```

Operaciones aplicadas a los campos de un registro

Asignación

prod.precio :=20

prod.nombre := 'Yerba'

prod.marca := 'SanCor'

Prod.fechaVenc.año:=2019

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;
Var prod: producto;
```

Operaciones aplicadas a los campos de un registro

Comparación

```
if prod.precio=20 then …

    if (prod.precio > 30) and
       (prod.precio <50)  then …

while (prod.marca= 'SanCor') do begin

if (prod.fechaVenc.año) =2018 then …
```

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;
Var prod: producto;
```

Operaciones aplicadas a los campos de un registro

Lectura

```
Read (prod.marca);
Read (prod.codigo);
Read (prod.fechaVenc.mes);
```

Escritura

```
write (prod.marca);
write (prod.nombre);
write (prod.fechaVenc.año);
```

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;
Var prod: producto;
```

Observar que el orden en que se leen los campos del registro podría no coincidir con el orden especificado en la declaración del tipo empleado ¿Por qué?

Operaciones permitidas para el tipo Registro

La única operación permitida entre registros es la de Asignación, **siempre y cuando**, los dos registros estén definidos del mismo tipo.

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
              codigo: integer;
              nombre: cadena15;
              marca: cadena15;
              fechaVenc: fecha;
              Precio: real;
             End;
Var prod1, prod2: producto;
Begin
   …….
    prod2 := prod1;
  ……..
```

¿Cómo se comparan variables de tipo registro?

No pueden realizarse comparaciones entre registros completos, es decir para saber si dos registros son iguales, se debe evaluar cada uno de los campos.

```
Type
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
```

Para comparar dos registros puede realizarse una función de la forma:

```
function iguales (f1, f2 : fecha) : boolean;
  begin
    iguales:= (f1.dia=f2.dia) and (f1.mes=f2.mes) and (f1.año = f2.año);
  end;
```

Ejemplo para la lectura del registro Producto

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
             codigo: integer;
             nombre: cadena15;
             marca: cadena15;
             fechaVenc: fecha;
             Precio: real;
            End;
Var prod: producto;
```

```
Procedure LeerProducto (Var prod: producto);
 begin
   Readln (prod.codigo);
   Readln (prod.nombre);
   Readln (prod.marca);
   Readln (prod.fechaVenc.dia);
   Readln (prod.fechaVenc.mes);
   Readln (prod.fechaVenc.año);
   Readln (prod.precio);
end;
```

¿Podríamos pensar en un procedimiento de lectura para el registro fecha?

```
Procedure LeerFecha (Var f: fecha);
 begin
   Readln (f.día);
   Readln (f.mes);
   Readln (f.año);
end;
```

Ejercitación

Se leen 100 datos correspondientes a los productos de un supermercado. Obtener un listado con los nombres de los productos con precio entre 25 y 50 pesos e informar la cantidad de productos de marca ‘Ala’.

Se puede utilizar la declaración del tipo producto vista anteriormente…

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
              codigo: integer;
              nombre: cadena15;
              marca: cadena15;
              fechaVenc: fecha;
              Precio: real;
            End;
```

Se leen 100 datos correspondientes a los productos de un supermercado. Obtener un listado con los nombres de los productos con precio entre 25 y 50 pesos e informar la cantidad de productos de marca 'Ala'.

```
Inicializar cantidad
Repetir 100
  Leer producto
  si precio entre 25 y 50 entonces
                          muestro nombre
  si marca = Ala then incremento cantidad
Fin
Mostrar cantidad
```

```
Procedure LeerProducto (Var prod: producto);
  Procedure LeerFecha (Var f:fecha);
   begin
     Readln (f.dia);
     Readln (f.mes);
     Readln (f.año);
   end
begin
    Readln (prod.codigo);
    Readln (prod.nombre);
    Readln (prod.marca);
    LeerFecha (prod.fechaVenc);
    Readln (prod.precio);
end;
```

```
Program ejemplo1;
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
             end
   producto = Record
                codigo: integer;
                nombre: cadena15;
                marca: cadena15;
                fechaVenc:fecha;
                Precio: real;
               End;

{implementación módulo LeerProducto}

var prod: producto; cant, i: integer;

begin   {Programa principal}
 cant:= 0;
 for i:= 1 to 100 do begin
   LeerProducto (prod);
   if (prod.precio >=25 and prod.precio <=50)
                 then Writeln (prod.nombre);
   if (prod.marca = 'Ala') then cant := cant +1;
 end;
 write (cant)
end.
```

Ejercitación

Se leen datos correspondientes a los productos de un supermercado. La lectura finaliza con código igual a -1. Obtener un listado con los nombres de los productos con precio entre 25 y 50 pesos e informar la cantidad de productos de marca ‘Ala’.

Se puede utilizar la declaración del tipo producto vista anteriormente…

```
Type
  cadena15 = string [15];
  días= 1..31;
  meses = 1..12;
  anios = 1900..2100
  fecha = record
            día: dias;
            mes: meses;
            año: anios;
          end;
  producto= Record
              codigo: integer;
              nombre: cadena15;
              marca: cadena15;
              fechaVenc: fecha;
              Precio: real;
            End;
```

Se leen datos correspondientes a los productos de un supermercado. La lectura finaliza con código igual a -1. Obtener un listado con los nombres de los productos con precio entre 25 y 5[illegible] 'Ala'.

```
Inicializar cantidad
Leer producto
Mientras producto <> -1
    si precio entre 25 y 50 entonces
                              muestro nombre
   si marca = Ala then incremento cantidad
   leer otro producto
Fin
Mostrar cantidad
```

```
Procedure LeerProducto2 (Var prod: producto);
 Procedure LeerFecha (Var f:fecha);
   begin
     Readln (f.dia);
     Readln (f.mes);
     Readln (f.año);
   end
begin
  Readln (prod.codigo);
  if (prod.código <> -1)
  then begin
         Readln (prod.nombre);
         Readln (prod.marca);
         LeerFecha (prod.fechaVenc);
         Readln (prod.precio);
       end;
end;
```

```
Program ejemplo2;
Type
    cadena15 = string [15];
    días= 1..31;
    meses = 1..12;
    anios = 1900..2100
    fecha = record
             día: dias;
             mes: meses;
             año: anios;
            end
    producto = Record
                   codigo: integer;
                   nombre: cadena15;
                   marca: cadena15;
                   fechaVenc:fecha;
                   Precio: real;
               End;

{implementación módulo LeerProducto2}

var prod:producto; cant: integer;

begin   {Programa principal}
  cant:= 0;
  LeerProducto2 (prod);
  While (prod.codigo <> -1)  do begin
   if (prod.precio >=25 and prod.precio <=50)
           then Writeln (prod.nombre);
   if (prod.marca = 'Ala') then cant := cant +1;
   LeerProducto2 (prod);
  end;
  write (cant)
end.
```

TEMAS de la CLASE

2 Corte de Control

Ejercitación con Registros

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;
```

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*

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.

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

¿Qúe 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
      leer otro producto
    fin mientras
    Mostrar total por marca
fin mientras
Mostrar contador > $ 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.
```