

PROGRAMACIÓN I

TEORÍA – CECILIA SANZ

A solid green horizontal bar at the bottom of the slide.

Temas

- ✓ Repaso de Conjuntos
- ✓ String – Definición - Ejemplos

Tipos de datos definidos por el usuario-CONJUNTO

Realice un programa que lea caracteres hasta leer el carácter '@',
La secuencia está formada por palabras, al finalizar informe la
cantidad de palabras con al menos 3 vocales minúsculas.

casa

auxilio

resplandor

materia

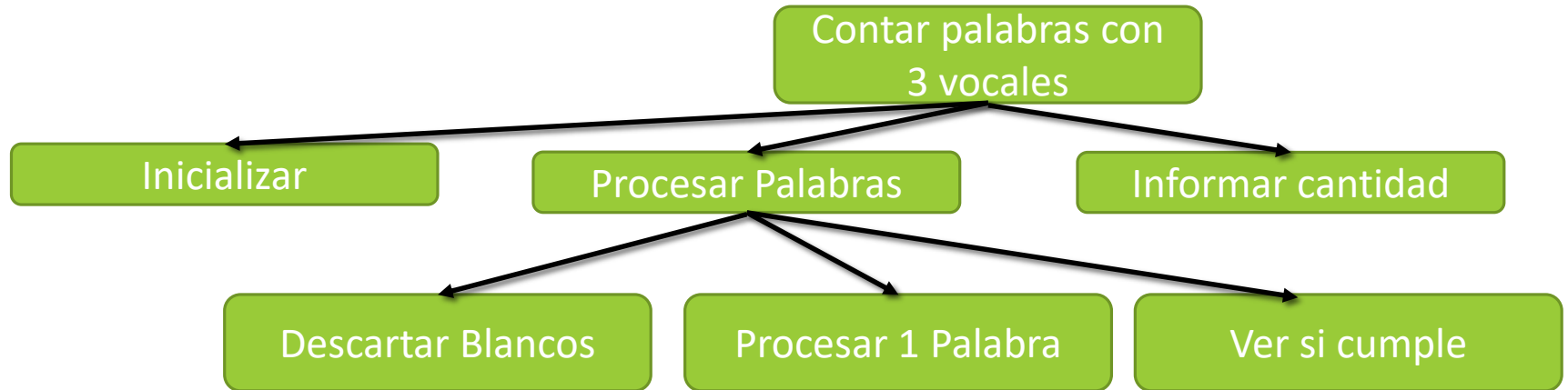
pelea



4


Tipos de datos definidos por el usuario-CONJUNTO

Realice un programa que lea caracteres hasta leer el carácter '@', La secuencia está formada por palabras, al finalizar informe la cantidad de palabras con al menos 3 vocales minúsculas.



Tipos de datos definidos por el usuario-CONJUNTO

```
Program cuatro;  
type  
  letras = set of char;  
{acá van los módulos}  
var  
  vocales:letras;  
  cant,contp:integer; letra:char;  
Begin  
  vocales:=['a','e','i','o','u'];  
  contp:=0;  
  read(letra);
```



Inicializar

```
while (letra <> '@') do {Procesar Palabras}  
Begin  
  cant:=0;  
  descartarBlancos(letra); {descatar Blancos}  
  procesarPalabra(letra, cant, vocales); {procesar Palabra}  
  If (cant >= 3) then contp:=contp+1; {ver si cumple}  
end;  
writeln ('Cantidad de palabras que cumplen: ', contp); {Informar}  
End.
```

Tipos de datos definidos por el usuario-CONJUNTO

```
Procedure descartarBlancos(var car:char);
```

```
Begin
```

```
  while ( car = ' ') do
```

```
    readln(car);
```

```
End;
```

```
Procedure procesarPalabra (var car: char; var cant:  
integer; vocales:letras);
```

```
Begin
```

```
  while (car <> '@') and (car<>' ')do
```

```
    begin
```

```
      if(car in vocales)then cant:= cant+1;
```

```
      readln (car);
```

```
    end;
```

```
End;
```

```
while (letra <> '@') do
```

```
Begin
```

```
  cant:=0;
```

```
  descartarBlancos(letra);
```

```
  procesarPalabra(letra, cant);
```

```
  If (cant >= 3) then contp:=contp+1;
```

```
end;
```

```
  writeln ('Cantidad de palabras que  
cumplen: ', contp);
```

```
End.
```

Tipos de datos definidos por el usuario-CONJUNTO

Realice un programa que lea caracteres hasta leer el carácter '@',
La secuencia está formada por palabras, al finalizar informe la
cantidad de palabras con al menos 3 vocales en minúsculas
distintas.

casa

auxilio

resplandor

materia

pelea



3

Tipos de datos definidos por el usuario-CONJUNTO

```
Program cuatro;  
type  
  conjletras = set of char;  
var  
  vocalesPal, vocales:conjletras;  
  cant, contp:integer; letra:char;  
Begin  
  vocales:=['a','e','i','o','u'];  
  contp:=0;  
  readln(letra);
```

```
  while (letra <> '@') do  
    Begin  
      cant:=0; vocalesPal:=[]; descartarBlancos(letra);  
      while (letra <> '@') and (letra<>' ' )do  
        begin  
          if(letra in vocales) and (not (letra in vocalesPal))  
            then begin  
              vocalesPal:=vocalesPal+[letra];  
              cant:= cant+1;  
            end;  
          readln (letra);  
        end;  
      If cant>=3 then contp:=contp+1;  
    End;  
    Writeln(contp);  
  End.
```

Esto deberías
er
módulo
Se deja
aquí
para
explicar
en clase

VER PROGRAMA
PASCAL

STRING

```
mostrar("hola mundo!");
```

comillas

Mensaje

X

hola mundo!

OK

Tipos de datos definidos por el usuario-STRING

- Un tipo de dato string es una sucesión de caracteres de un largo determinado, que se almacenan en un área contigua de la memoria.
- Existe en la mayoría de los lenguajes como un tipo predefinido.
- No es un dato simple.

Type

identificador = string[k]; donde k es la longitud máxima del string

identificador = string; se supone como 255 caracteres como máximo

STRING- Declaración

Tipos de datos definidos por el usuario-STRING

TYPE cadena1 = string [10];
 cadena2= string [25];
 fecha = string [8];
 día = string [2];

VAR h1, h2, h3 : cadena1;
 h4 : cadena2;
 h5,h6 : string [14];
 fecha1, fecha2 : fecha;

EJEMPLOS

STRING- Asignación

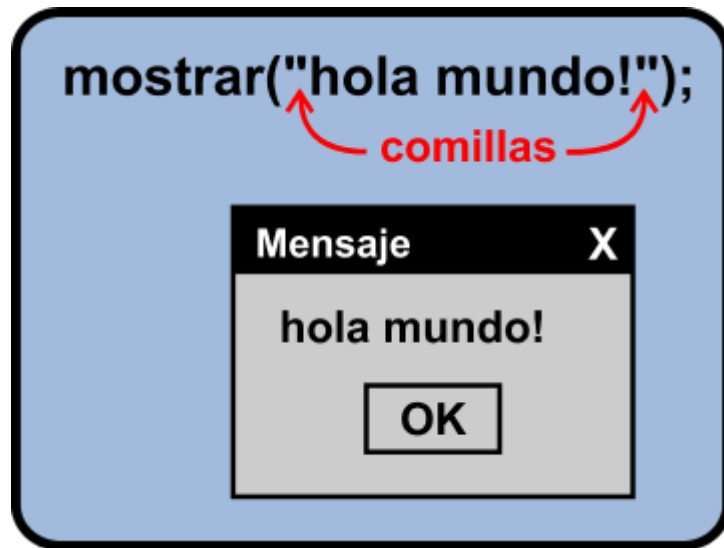
Tipos de datos definidos por el usuario-STRING

Asignación:

```
Program uno;  
Type  
  cadena= string[20];  
Var  
  c1,c2:cadena;  
Begin  
  c1:= 'casa';  
  c2:= c1;  
End.
```

Si cuando se asigna contenido a C1 este supera los 20 caracteres el mismo es truncado (a 20 caracteres en este caso).

STRING- Comparación



Tipos de datos definidos por el usuario-STRING

Comparación por igualdad

```
Program uno;  
Type  
  cadena= string[20];  
Var  
  c1,c2:cadena;  
Begin  
  if (c1 = 'casa') then...  
  while (c2 = c1) do...  
End.
```

Se evalúa si las longitudes de los strings a comparar son iguales, si es así compara el contenido. Si las longitudes no son iguales no se compara el contenido.

Tipos de datos definidos por el usuario-STRING

Comparación por distinto

Program uno;

Type

cadena= string[20];

Var

c1,c2:cadena;

Begin

if (c1<> 'casa') then...

while (c2= c1) do...

End.

Se evalúa si las longitudes de los strings a comparar son iguales, si no es así se devuelve verdadero. Si las longitudes son iguales se compara el contenido.

OPERACIONES

Tipos de datos definidos por el usuario-STRING

Comparación por menor

Program uno;

Type

cadena= string[20];

Var

c1,c2:cadena;

Begin

C1:='amor';

if (c1<= 'casa') then...

while (c2= c1) do...

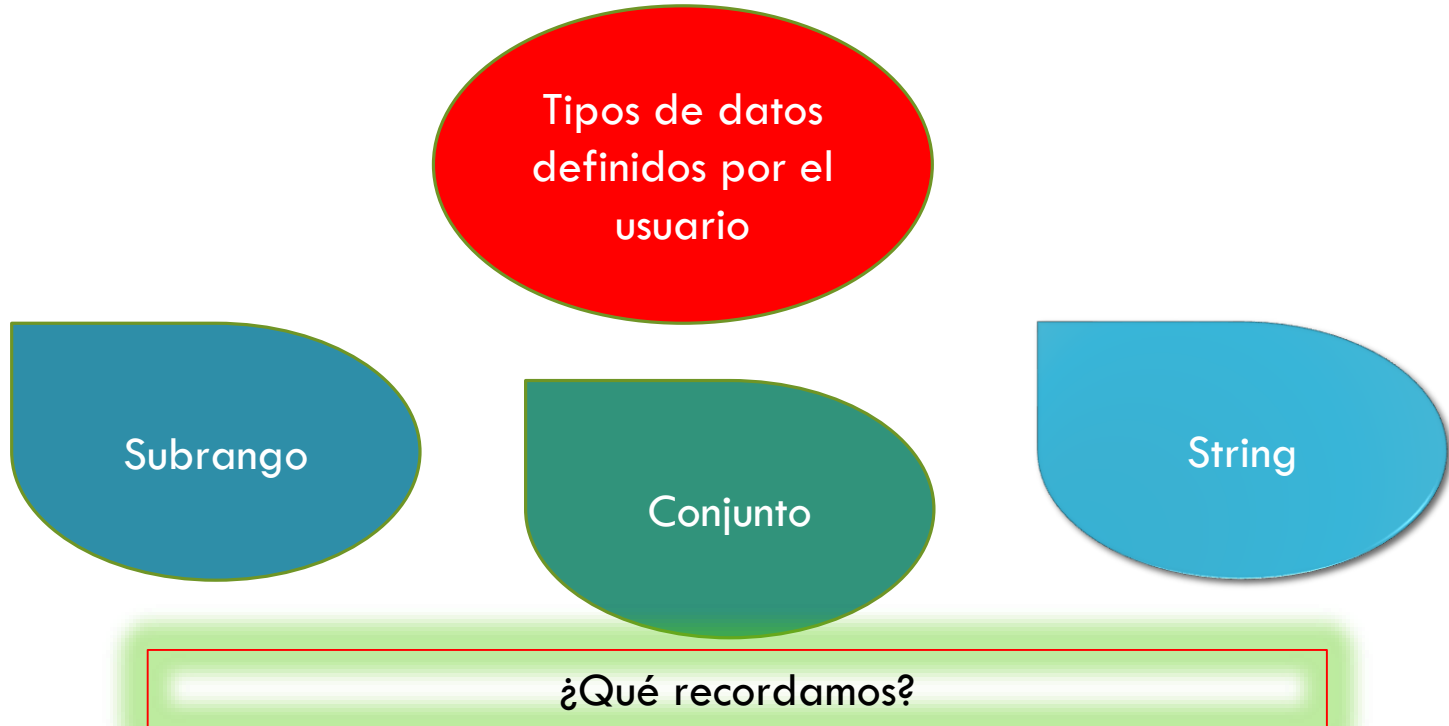
End.

Se utiliza el orden lexicográfico

OPERACIONES

REPASO

Tipos de datos definidos por el usuario vistos



Tipos de datos definidos por el usuario-PREGUNTAS

Repaso:

- ✓ ¿Cómo se definen los datos de tipo subrango?
- ✓ ¿Qué operadores pueden utilizarse con datos de tipo subrango?
- ✓ ¿A qué tipos de datos simples pueden aplicarse el concepto de subrango?
- ✓ ¿Cómo es el esquema general de un programa que incorpora la declaración de tipos?