



Conceptos de Algoritmos Datos y Programas



CADP – TEMAS

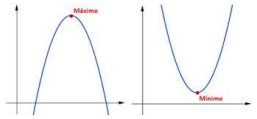


● Máximos y mínimos

● EJERCICIO – PREGUNTAS FINALES

CADP – MAXIMOS y MINIMOS

MAXIMOS



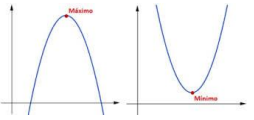
Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.

- Qué tipo de valor es el promedio?
- Cuál es la condición de fin?
- Cómo verifico que es el mejor promedio?

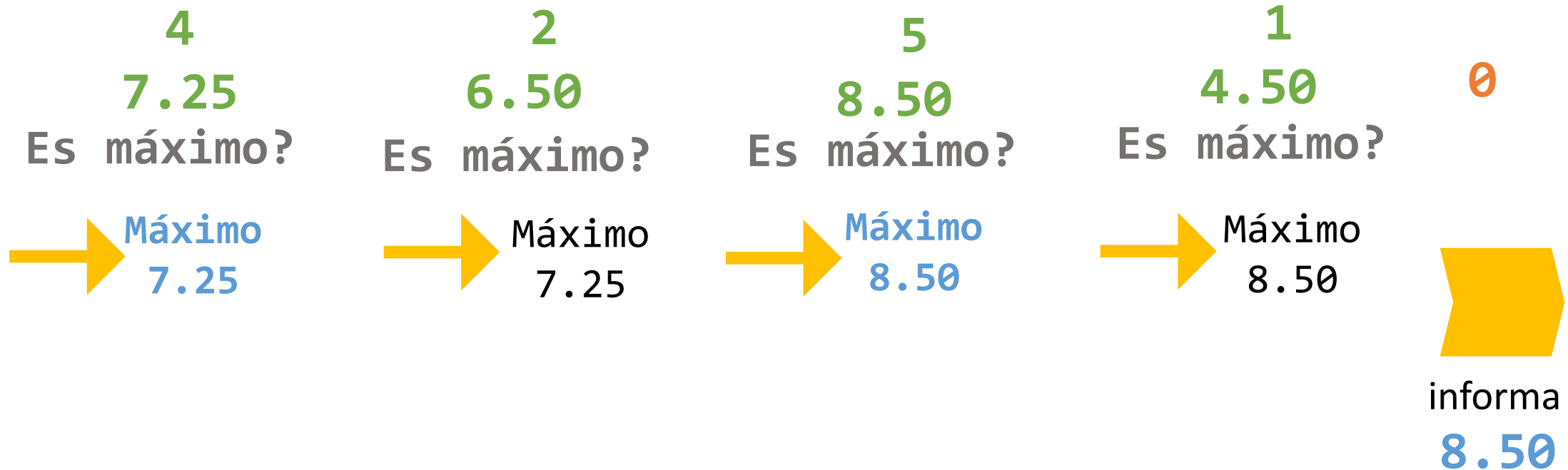
La forma de trabajar es teniendo en cuenta la manera natural de pensarlo, por cada número (o valor) a tener en cuenta se verifica si es máximo y en caso de serlo se actualiza el máximo que se tiene hasta el momento.

CADP – MAXIMOS y MINIMOS

MAXIMOS

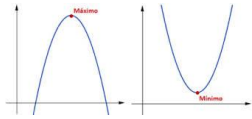


Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.



CADP – MAXIMOS y MINIMOS

MAXIMOS



Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;  
var
```

```
    prom:real;  
    alu:integer;
```

```
begin
```

```
    Leo un promedio (prom y un alumno alu);  
    while (no sea la condición de fin) do
```

Cómo verifico begin

si es máximo? verificar si es máximo

si (es máximo) entonces

actualizar máximo

**Cómo actualizo
el máximo?**

Leo un promedio (prom y un alumno alu);

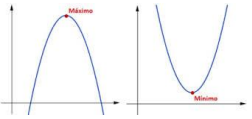
```
end;
```

```
write ("El mejor promedio es:",    );
```

```
end.
```

CADP – MAXIMOS y MINIMOS

MAXIMOS



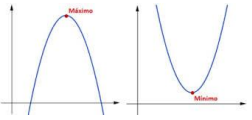
Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.

Cuál es el error?

```
Program uno;  
var  
    prom:real;alu:integer;  max:real;  
begin  
    read(prom);  
    read(alu);  
    while (prom <> 0) do  
        begin  
            If (prom >= max) then  
                max:= prom;  
            read (prom);  
            read(alu);  
        end;  
  
        write ("El mejor promedio es:", max );  
    end.
```

CADP – MAXIMOS y MINIMOS

MAXIMOS



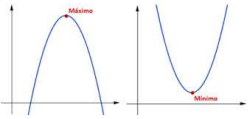
Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.

Qué modifico si quiero saber el número del alumno con mejor promedio?

```
Program uno;  
var  
    prom:real;alu:integer; max:real;  
begin  
    read(prom);  
    read(alu);  
    max:=-1; ←  
    while (prom <> 0) do  
        begin  
            If (prom >= max) then  
                max:= prom;  
            read(prom);  
            read(alu);  
        end;  
    write ("El mejor promedio es:", max );  
end.
```

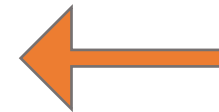
CADP – MAXIMOS y MINIMOS

MAXIMOS

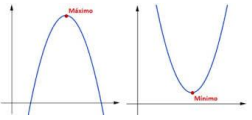


Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;
var
    prom:real;alu:integer; max:real;maxalu:integer;
begin
    read(prom);
    read(alu);
    max:= -1;
    while (prom <> 0) do
        begin
            If (prom >= max) then begin
                max:= prom;
                maxalu:= alu;
            end;
            read(prom);
            read(alu);
        end;
        write ("El mejor alumno es:", maxalu );
    end.
```



CADP – MAXIMOS y MINIMOS **MAXIMOS -Recordar**



Utilizar una variable que representará al máximo.

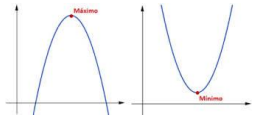


Inicializar la variable antes de comenzar la lectura de los datos. El máximo en un valor bajo.

Actualizar la variable máximo cuando corresponda

CADP – MAXIMOS y MINIMOS

MAXIMOS



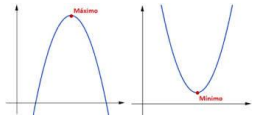
```
Program uno;  
var  
    cantidad,num,max_cant,max_num:integer;  
begin  
    max_cant:= 0;  
    read(num); read(cantidad);  
    while (num <> 80) do  
        begin  
            if (cantidad > max_cant) then begin  
                max_num:= num;  
                max_cant:= cantidad;  
            end;  
            read(num); read(cantidad);  
        end;  
        write ("El número con mas cantidad es :",max_num);  
    end.
```

Qué imprime si
se lee?

15	0
23	0
8	0
80	16

CADP – MAXIMOS y MINIMOS

MINIMOS



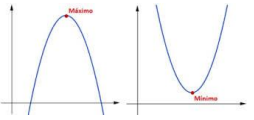
Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más bajo.

- Qué tipo de valor es el promedio?
- Cuál es la condición de fin?
- Cómo verifico que es el peor promedio?

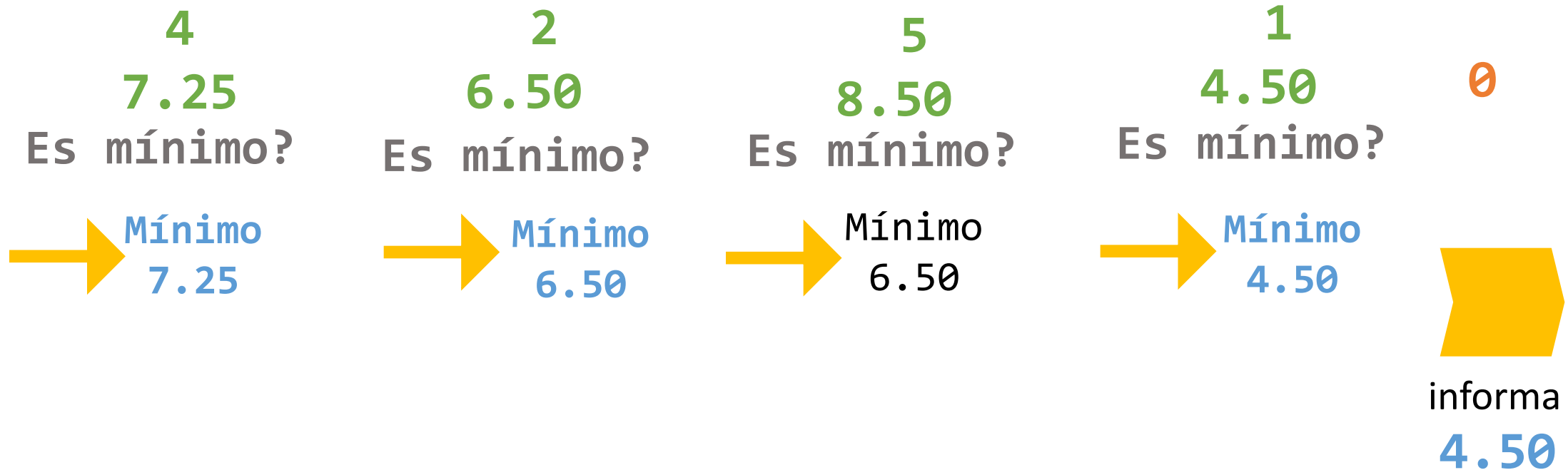
La forma de trabajar es teniendo en cuenta la manera natural de pensarlo, por cada número (o valor) a tener en cuenta se verifica si es mínimo y en caso de serlo se actualiza el mínimo que se tiene hasta el momento.

CADP – MAXIMOS y MINIMOS

MINIMOS

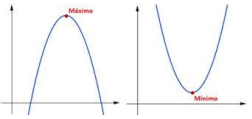


Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más bajo.



CADP – MAXIMOS y MINIMOS

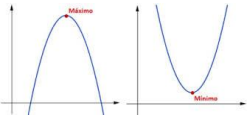
MAXIMOS



Realizar un programa que lea número de alumno y promedio hasta leer un promedio igual a 0. Al finalizar informar el promedio más bajo.

```
Program uno;
var
    prom:real;alu:integer; min:real;
begin
    read(prom);
    read(alu);
    min:=11;
    while (prom <> 0) do
        begin
            If (prom <= min) then
                min:= prom;
            read(prom);
            read(alu);
        end;
    write ("El mejor promedio es:", min );
end.
```

CADP – MAXIMOS y MINIMOS **RECORDAR**



Utilizar una variable que representará al mínimo.

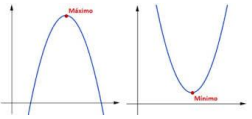


Inicializar la variable antes de comenzar la lectura de los datos. El mínimo en un valor alto.

Actualizar la variable mínimo cuando corresponda

*Qué modifico si quiero saber
el promedio máximo y el
promedio mínimo?*

CADP – MAXIMOS y MINIMOS **AMBOS**

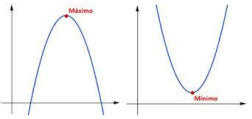


**Ambas soluciones
son correctas?**

```
Program opcion1;
var
prom,max,min:integer;
begin
  max:= -1;
  min= 11;
  read(prom);
  while (prom <> 0) do
    begin
      if (prom >= max) then
        max:= prom
      else
        if (prom <= min) then
          min:= prom;
        read(prom)
      end;
    write (max,min);
  end.
```

```
Program opcion2;
var
prom,max,min:integer;
begin
  max:= -1;
  min= 11;
  read(prom);
  while (prom <> 0) do
    begin
      if (prom >= max) then
        max:= prom
      if (prom <= min) then
        min:= prom;
      read(prom)
    end;
    write (max,min);
  end.
```

CADP – MAXIMOS y MINIMOS



```
Program uno;  
Var  
    max,valor:integer;  
begin  
    max:=-999;  
    read(valor);  
    while (valor <> 0) do  
        begin  
            if (valor >= max) then  
                max:= valor  
            read (valor);  
        end;  
    write (max);  
end.
```

```
Program dos;  
Var  
    min,max,valor:integer;  
begin  
    min:= 9999; max:=-999;  
    read(valor);  
    while (valor <> 0) do  
        begin  
            if (valor >= max) then  
                max:= valor  
            else  
                if (valor <= min) then  
                    min:= valor;  
                read (valor);  
            end;  
        write (min,max);  
    end.
```

```
Program tres;  
Var  
    min,max,valor:integer;  
begin  
    min:= 9999; max:=-999;  
    read(valor);  
    while (valor <> 0) do  
        begin  
            if (valor >= max) then  
                max:= valor  
            if (valor <= min) then  
                min:= valor;  
            read (valor);  
        end;  
    write (min,max);  
end.
```



Qué imprime cada programa si se leen los siguientes valores?

-6, 5, 31, 5, 50, -6, 50, 20, 0



Conceptos de Algoritmos Datos y Programas



CADP – TEMAS



● Tipos de Datos definidos por el usuario

● EJERCICIO – PREGUNTAS FINALES

CADP – TIPOS DE DATOS- CLASIFICACION



SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE

COMPUESTO

DEFINIDO POR EL LENGUAJE

DEFINIDO POR EL PROGRAMADOR

string

Integer
real
char
boolean

Son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

Permiten definir nuevos tipos de datos a partir de los tipos simples.

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



HASTA AHORA

Hemos trabajado los tipos de datos simples o compuestos que se pueden considerar estándar en la mayoría de los lenguajes de programación.



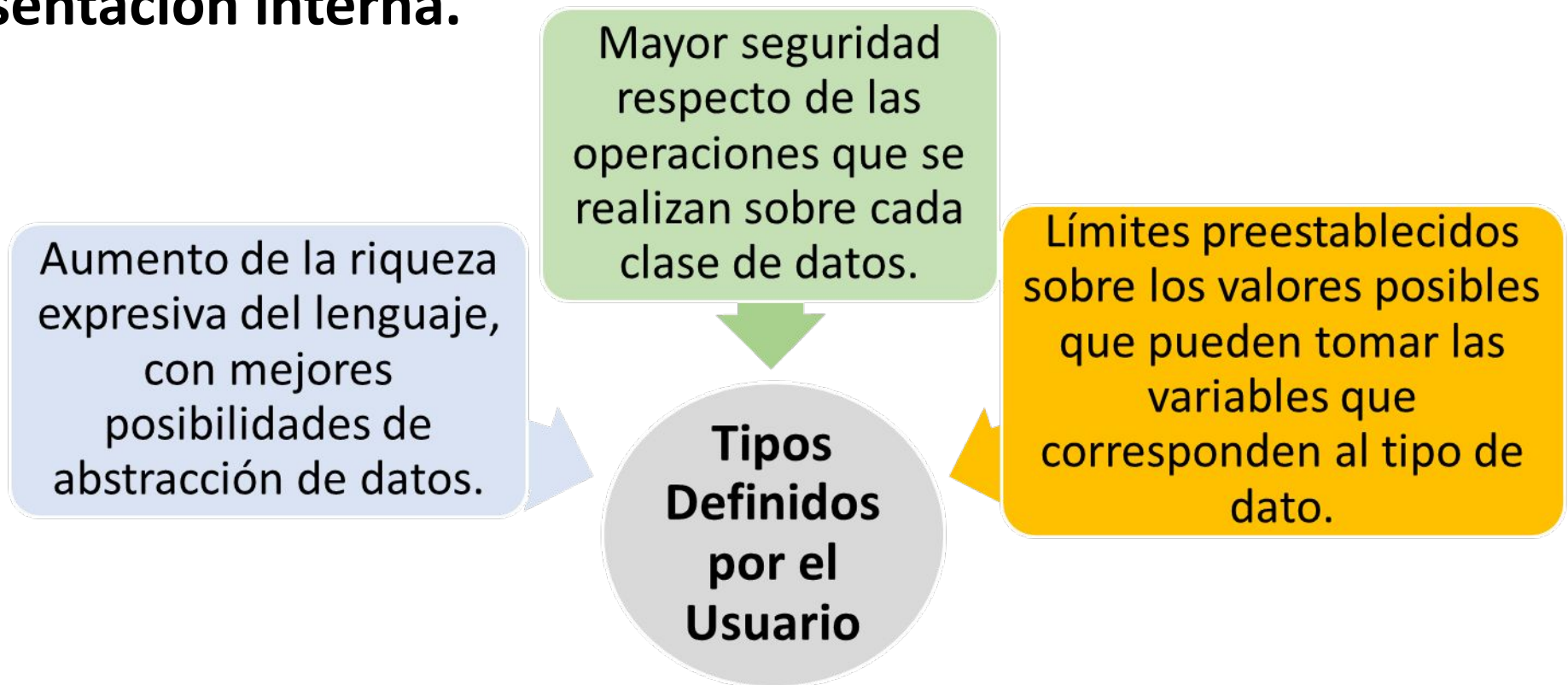
Esto significa que el conjunto de valores de ese tipo, las operaciones que se pueden efectuar y su representación están definidas y acotadas por el lenguaje.

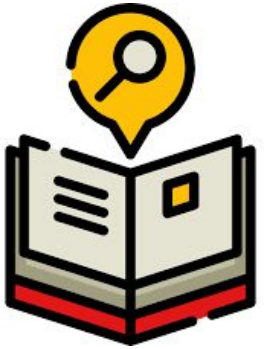
Esta definición nos indica que podemos requerir la representación de elementos “NO de tipo estándar” que se asocien con el fenómeno real a tratar.

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



Un aspecto muy importante en los lenguajes de programación es la capacidad de especificar y manejar datos no estándar, indicando valores permitidos, operaciones válidas y su representación interna.



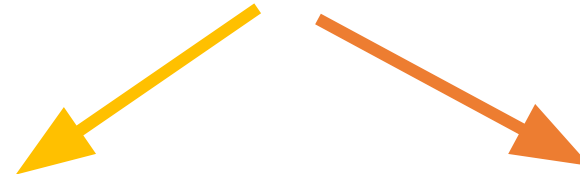


TDU (Tipos de Datos definidos por el usuario)

Un tipo de dato definido por el usuario es aquel que no existe en la definición del lenguaje, y el programador es el encargado de su especificación.

Type

```
identificador = tipo;
```



Un tipo estandar

Un tipo definido por el
lenguaje

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



```
program uno;
```

```
Const
```

```
...
```

```
Type
```

```
    nuevotipo1 = tipo;
```

```
Módulos
```

```
Var
```

```
    x: nuevotipo1;
```

```
...
```

```
Begin
```

```
...
```

```
End.
```

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



```
Program nombre;
```

```
Const
```

```
    N = 25;
```

```
    pi = 3.14;
```

Type

```
    nuevotipo1 = integer;
```

```
módulos {luego veremos como se  
    declaran}
```

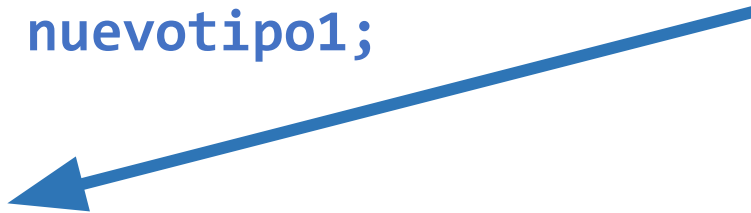
```
var
```

```
edad: integer;
```

```
peso: real;
```

```
valor: nuevotipo1;
```

```
begin
```



La variable valor:

- puede tener asignado cualquier valor permitido para los enteros.
- Puede utilizar cualquier operación permitida para los enteros.
- NO puede relacionarse con ninguna otra variable que no sea de su mismo tipo



CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO

```
Program nombre;
```

```
Const
```

```
    N = 25;
```

```
    pi = 3.14;
```

```
Type
```

```
    nuevotipo1 = integer;
```

```
var
```

```
    edad: integer;
```

```
    peso: real;
```

```
    valor: nuevotipo1;
```

```
begin
```

```
    valor:= 8;  read (valor); if (valor MOD 3 = ...) then ...
```

```
    edad:= valor;  edad:= edad + valor
```



```
end.
```

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



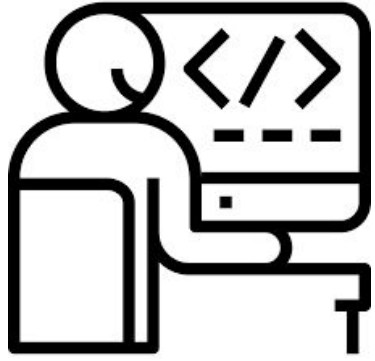
VENTAJAS



Flexibilidad: en el caso de ser necesario modificar la forma en que se representa el dato, sólo se debe modificar una declaración en lugar de un conjunto de declaraciones de variables.

Documentación: se pueden usar como identificador de los tipos, nombres autoexplicativos, facilitando de esta manera el entendimiento y lectura del programa.

Seguridad: se reducen los errores por uso de operaciones inadecuadas del dato a manejar, y se pueden obtener programas más confiables..



Conceptos de Algoritmos Datos y Programas



CADP – TEMAS



- Tipos de Datos definidos por el usuario
- Tipos de Datos subrango
- EJERCICIO – PREGUNTAS FINALES

CADP – TIPOS DE DATOS- CLASIFICACION



SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE

COMPUESTO

DEFINIDO POR EL LENGUAJE

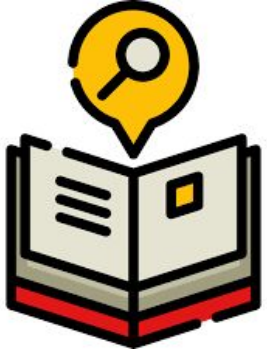
DEFINIDO POR EL PROGRAMADOR

string

Integer
real
char
boolean

Son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

Permiten definir nuevos tipos de datos a partir de los tipos simples.



SUBRANGO

Es un tipo ordinal que consiste de una sucesión de valores de un tipo ordinal (predefinido o definido por el usuario) tomado como base.

- Es simple.
- Es ordinal
- Existe en la mayoría de los lenguajes

*Cómo se
declara?*

*Cómo se
usa?*

CADP – TIPOS DE DATOS- CLASIFICACION



SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE

COMPUESTO

string

DEFINIDO POR EL
LENGUAJE

DEFINIDO POR EL
PROGRAMADOR

subrango

Integer
real
char
boolean

Son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

Permiten definir nuevos tipos de datos a partir de los tipos simples.

CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



SUBRANGO

```
program uno;
```

```
Const
```

```
...
```

```
Type
```

```
    tipo1 = valor1..valor2;
```

```
var
```

```
    x,y: tipo1;
```

```
Begin
```

```
...
```

```
End.
```


CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



SUBRANGO

```
program uno;
```

```
Const
```

```
...
```

```
Type
```

```
    notas = 0..10;
```

```
    letras = 'a'..'1';
```

```
var
```

```
    not1,not2: notas;
```

```
    let:letras;
```

```
Begin
```

```
...
```

```
End.
```

**Qué operaciones
puedo hacer con
cada variable?**

**Se puede hacer un
subrango de
reales?**



SUBRANGO – OPERACIONES

Operaciones Permitidas

- Asignación
- Comparación
- Todas las operaciones permitidas para el tipo base

Operaciones NO permitidas

- Depende del tipo base.

CADP – EJERCITACION



Programa 1: Se pide realizar que lea caracteres (suponga sólo minúsculas) hasta leer el carácter 'z'. Al finalizar informar cuantos caracteres leídos estaban entre 'a' y 'h'; cuántos entre 'i' y 'n' y cuántos entre 'ñ' y 'y'.

- Estructura de control principal del programa?
- Necesito definir algún tipo de datos?
- Cómo cuento según el valor?

CADP – EJERCITACION



Qué tiene de malo esta solución?

Qué puede pasar?

```
Program uno;
var
  primera,segunda,tercera :integer;
  letra:char;
Begin
  primera:= 0;  segunda:=0; tercera:=0;
  read (letra);
  while (letra <> 'z') do
    begin
      if ((letra = 'a') or (letra = 'b') or ...) then
        primera:= primera + 1
      else
        if ((letra = 'i') or (letra = 'j') or ...) then
          segunda:= segunda + 1
        else
          tercera:= tercera + 1;
        read (letra);
      end;
    write (primera,segunda,tercera);
  end.
```

CADP – EJERCITACION



```
Program uno;
Type
  letras = 'a'..'z';
var
  primera,segunda,tercera:integer;
  letra:letras;
Begin
  primera:= 0;  segunda:=0; tercera:=0;
  read (letra);
  while (letra <> 'z') do
    begin
      if ((letra = 'a') or (letra = 'b') or ...) then
        primera:= primera + 1
      else
        if ((letra = 'i') or (letra = 'j') or ...)
then
          segunda:= segunda + 1
        else
          tercera:= tercera + 1;
        read (letra);
      end;
    write (primera,segunda,tercera);
```

Se puede
mejorar?

CADP – EJERCITACION



```
Program uno;  
Type  
  letras = 'a'..'z';  
var  
  primera,segunda,tercera:integer;  
  letra:letras;  
Begin  
  primera:= 0;  segunda:=0; tercera:=0;  
  read (letra);  
  while (letra <> 'z') do  
    begin  
      case letra of  
        'a'..'h': primera:= primera + 1;  
        'i'..'n': segunda:= segunda + 1;  
        'ñ'..'y': tercera:= tercera + 1;  
      end;  
      read (letra);  
    end;  
    write (primera,segunda,tercera);  
  end.  
end.
```