



Conceptos de Algoritmos Datos y Programas



CADP – Temas de la clase de hoy

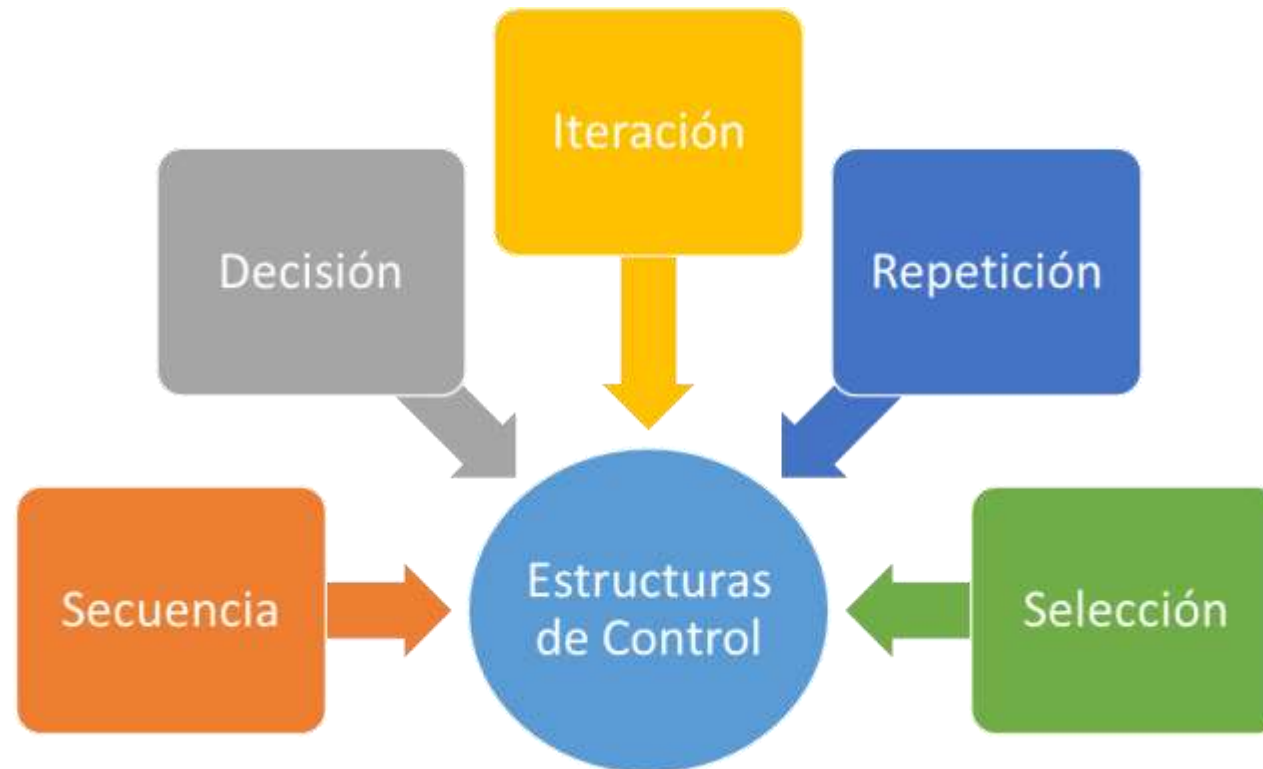


- Estructuras de control
- Cálculo de máximos y mínimos
- Tipos de datos definidos por el usuario

CADP – ESTRUCTURAS DE CONTROL



Todos los lenguajes de programación tienen un conjunto mínimo de instrucciones que permiten especificar el control del algoritmo que se quiere implementar. Como mínimo deben contener: secuencia, decisión e iteración.



CADP – ESTRUCTURAS DE CONTROL REPETICION



Realizar un programa que lea caracteres hasta leer el carácter '='. Al finalizar informe cuantos caracteres mayúsculas, cuántos minúscula, cuántos dígitos, y cuántos caracteres especiales ha leído.

- Qué tipo de datos leo?
- Cuál es la condición de fin?
- Cómo identifico que carácter es?

'A'		
'0'		Minúsculas 1
'B'		Mayúsculas 2
'x'		Dígitos 1
'!'	informa	Especiales 2
'\$'		
'='		

CADP – ESTRUCTURAS DE CONTROL REPETICION



```
Program uno;
var
  car:char;
  cmin, cmay, cnum, cesp:integer;
begin
  read (car);
  cmin:=0; cmay:=0; cnum:=0; cesp:=0;

  while (car <> '=') do begin
    if (car = 'a') or (car = 'b')... or (car = 'z') then
      cmin:= cmin + 1;
    if (car = 'A') or (car = 'B')... or (car = 'Z') then
      cmay:= cmay + 1;
    if (car = '0') or (car = '1')... or (car = '9') then
      cnum:= cnum + 1;
    if (car = '@') or (car = '!')... or (car = '*') then
      cesp:= cesp + 1;
    read (car);
  end;
  write (...);
End.
```

Con lo que
sabemos hasta
ahora

Se puede
mejorar?

CADP – ESTRUCTURAS DE CONTROL REPETICION



```
Program uno;
var
    car:char; cmin, cmay, cnum, cesp:integer;
begin
    read (car);
    cmin:=0; cmay:=0; cnum:=0; cesp:=0;
    while (car <> '=') do begin
        if (car = 'a') or (car = 'b')... or (car = 'z') then
            cmin:= cmin + 1
        else
            if (car = 'A') or (car = 'B')... or (car = 'Z') then
                cmay:= cmay + 1
            else
                if (car = '0') or (car = '1')... or (car = '9') then
                    cnum:= cnum + 1
                else if (car = '@') or (car = '!')... or (car = '*') then
                    cesp:= cesp + 1;
            read (car);
        end;
    write (...);
End.
```

Con lo que
sabemos hasta
ahora

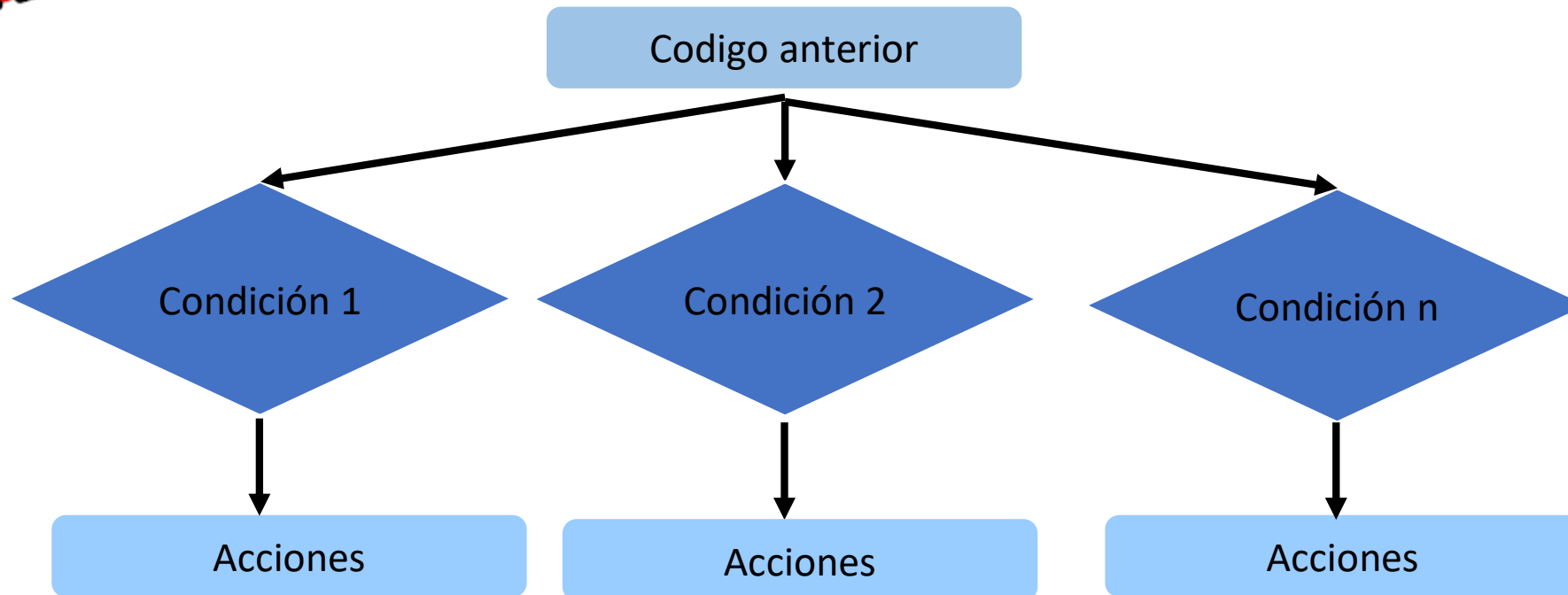
Se puede
mejorar?

CADP – ESTRUCTURAS DE CONTROL



SELECCION

Permite realizar distintas acciones dependiendo del valor de una variable de tipo ordinal.



Cómo es la sintaxis?

CADP – ESTRUCTURAS DE CONTROL SELECCION



```
case (variable) of
  condicion1: accion1;
  condición 2: acción2;
  ....
  condición n: acción n;
end;
```



más de una acción

```
case (variable) of
  condicion1: accion1;
  condición 2: begin
    acción2;
    accion3;
  end;
  ....
  condición n: accion;
end;
```


CADP – ESTRUCTURAS DE CONTROL SELECCION



```
Program uno;  
var  
    car:char; cmin,cmay,cnum,cesp:integer;  
begin  
    read (car);  
    cmin:=0; cmay:=0; cnum:=0; cesp:=0;  
    while (car <> '=') do begin  
        case car of  
            car = 'a': cmin:= cmin + 1;  
            ...  
            car = 'z': cmin:= cmin + 1;  
            car = 'A': cmay:= cmay + 1;  
            ...  
            car = 'Z': cmay:= cmay + 1;  
            car = '0': cmay:= cnum+ 1;  
            ...  
            car = '9': cmay:= cnum + 1;  
            else cesp:= cesp + 1;  
        end;  
        read (car);  
    end;  
    write ()  
End
```

*Se puede
mejorar?*

CADP – ESTRUCTURAS DE CONTROL SELECCION



```
Program uno;  
var  
    car:char;  
    cmin,cmay,cnum,cesp:integer;  
begin  
    read (car);  
    cmin:=0; cmay:=0; cnum:=0; cesp:=0;  
    while (car <> '=') do begin  
        case car of  
            'a'.. 'z': cmin:= cmin + 1;  
            'A'.. 'Z': cmay:= cmay + 1;  
            '0'.. '9': cnum:= cnum+ 1;  
            else  cesp:= cesp + 1;  
        end;  
        read (car);  
    end;  
    write (  
End.
```

CADP – ESTRUCTURAS DE CONTROL SELECCION



- La variable del case debe ser de tipo ordinal
- Las opciones deben ser disjuntas

CADP – Estructuras de control



Problema: se leen valores de alturas de personas, hasta leer la altura 1.59. Informar la cantidad de personas que miden entre 1.00 y 1.30; la cantidad de personas que miden entre 1.31 y 1.50; la cantidad de personas que miden entre 1.51 y 1.89 y las que miden más de 1.89

El alumno 1: utiliza una variable real para leer las alturas y cuatro contadores para contar la cantidad de personas en cada rango.

Además utiliza un while como estructura de control principal y adentro utiliza un case que incluye los rangos de alturas para saber cual contador sumar. Al final informa los valores de los contadores.

El alumno 2: utiliza una variable real para leer las alturas y cuatro contadores para contar la cantidad de personas en cada rango.

Además utiliza un while como estructura de control principal y adentro utiliza un if con else para saber cual contador sumar. Al final informa los valores de los contadores.

CADP – MAXIMOS y MINIMOS

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más alto.

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

7.25

6.50

8.50

4.50

0

Es
máximo?
→ Máximo
7.25

Es
máximo?
→ Máximo
7.25

Es máximo?
→ Máximo
8.50

Es máximo?
→ Máximo
8.50



informa

8.50

CADP – MAXIMOS y MINIMOS

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;  
var  
    prom:real;  
begin  
    Leo un promedio (prom);  
    while (no sea la condición de fin) do  
        begin  
            verificar si es máximo  
            Leo un promedio (prom);  
        end;  
  
        write ("El mejor promedio es:",      );  
end.
```

*Cómo
verifico si es
máximo?*

CADP – MAXIMOS y MINIMOS

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;  
var  
    prom:real;    max:real;  
begin  
    Leo un promedio (prom);  
    while (no sea la condición de fin) do  
        begin  
            verificar si es máximo    If (prom >= max) then  
                                     max:= prom;  
            Leo un promedio (prom);  
        end;  
    write ("El mejor promedio es:", max );  
end.
```

Cuál es el error?

CADP – MAXIMOS y MINIMOS

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;
```

```
var
```

```
    prom:real;    max:real;
```

```
begin
```

```
    max:= -1;
```

```
    Leo un promedio (prom);
```

```
    while (no sea la condición de fin) do
```

```
        begin
```

```
            verificar si es máximo
```

```
            If (prom >= max) then
```

```
                max:= prom;
```

```
            Leo un promedio (prom);
```

```
        end;
```

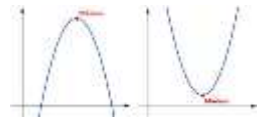
```
        write (“El mejor promedio es:”, max );
```

```
end.
```


CADP – MAXIMOS y MINIMOS



MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más alto.

```
Program uno;
```

```
var
```

```
    prom:real;    max:real;
```

```
begin
```

```
    max:= -1;
```

```
    read(prom);
```

```
    while (prom <> 0) do
```

```
        begin
```

```
            If (prom >= max) then
```

```
                max:= prom;
```

```
                read (prom);
```

```
            end;
```

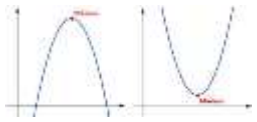
```
        write ("El mejor promedio es:", max );
```

```
end.
```

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

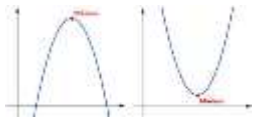
CADP – MAXIMOS y MINIMOS

MAX-MIN



```
Program uno;  
var  
    prom,max:real; nro_alu:integer; max_nro_alu:integer;  
begin  
    max:= -1;  
    read(prom); read(nro_alu);  
    while (prom <> 0) do  
        begin  
            if (prom >= max) then begin  
                max:= prom;  
                max_nro_alu:= nro_alu;  
            end;  
            read(prom); read(nro_alu);  
        end;  
        write ("El mejor promedio es:",max,"y el alumno es", max_nro_alu);  
    end.
```

CADP – MAXIMOS y MINIMOS **RECORDAR**



Utilizar una variable que representará al máximo.

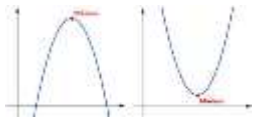


Inicializar la variable antes de comenzar la lectura de los datos. El máximo en un valor bajo.

Actualizar la variable máximo cuando corresponda

CADP – MAXIMOS y MINIMOS

MAX-MIN



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

Qué imprime si
se lee?

23

8

23

5

0

80

CADP – MAXIMOS y MINIMOS

MAX-MIN



```
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

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más bajo.

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

7.25

Es mínimo?

→ Mínimo
7.25

6.50

Es mínimo?

→ Mínimo
6.50

8.50

Es mínimo?

→ Mínimo
6.50

9.50

Es mínimo?

→ Mínimo
6.50

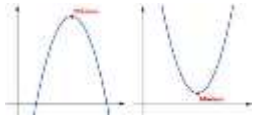
0



informa
6.50

CADP – MAXIMOS y MINIMOS

MAX-MIN



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más bajo.

```
Program uno;  
Var  
    min:real;  
    prom:real;  
begin  
    min:= 11;  
    read(prom);  
    while (prom <> 0) do  
        begin  
            If (prom <= min) then  
                min:= prom;  
            read (prom);  
        end;  
        write ("El peor 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



Realizar un programa que lea promedios de alumnos leer un promedio igual a 0. Al finalizar informar el promedio más bajo y el mas alto.

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

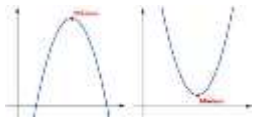


Funcionan las dos?

Si se leen los valores a
continuación que
imprime cada una

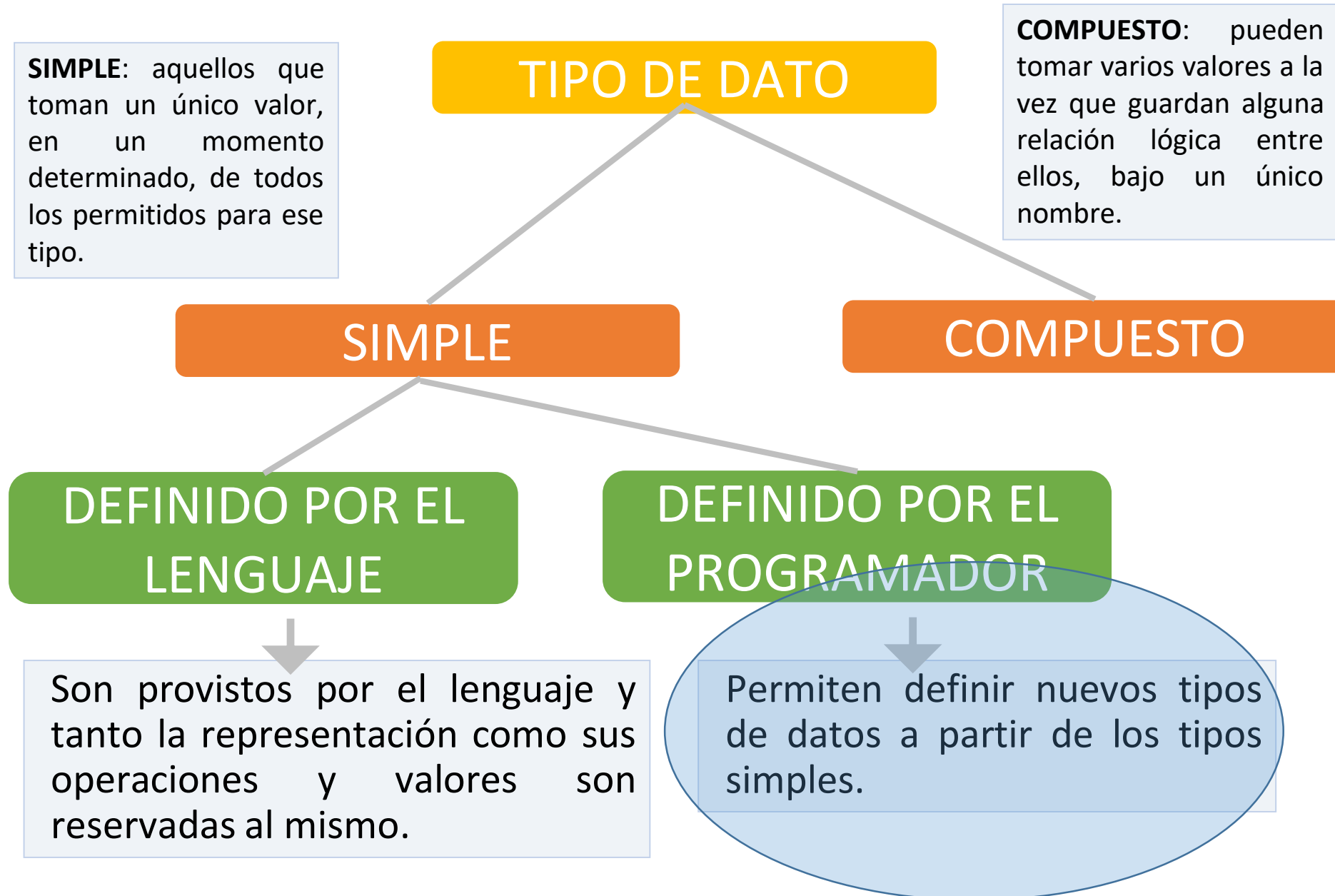
5.50
10
8.50
0

MAX-MIN



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

CADP – Tipos de Datos - Clasificación

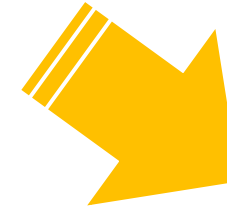


CADP – Tipos de Datos definidos por el programador



HASTA AHORA

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



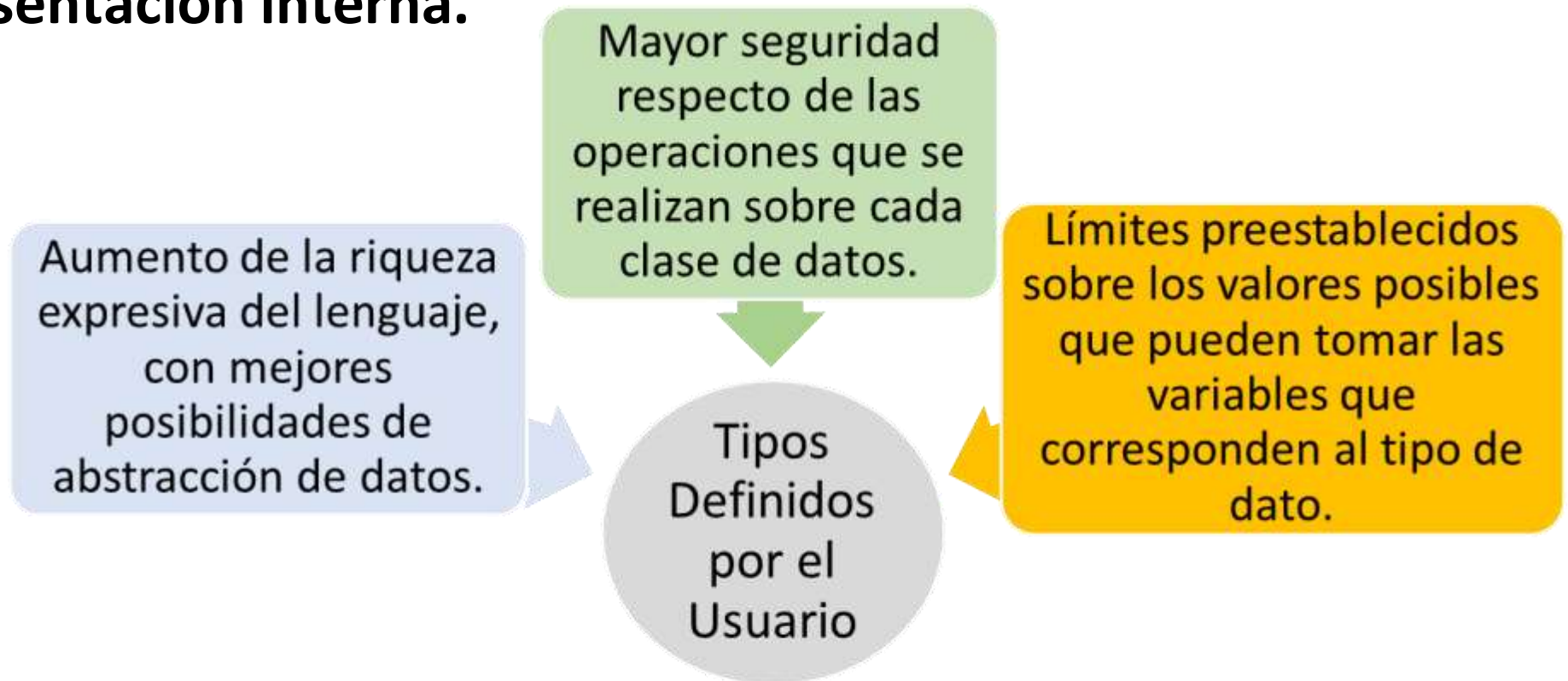
Esto significa que el conjunto de valores de ese tipo, las operaciones que se pueden efectuar y su representación están definidas y acotadas por el lenguaje.

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 programador



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.



CADP – Tipos de Datos definidos por el programador



TIPO 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;
```

Dónde se declara?

Un tipo estandar

**Un tipo definido
por el lenguaje**

CADP – Tipos de Datos definidos por el programador



TIPO DE DATOS DEFINIDOS POR EL USUARIO (TDDU)

```
program uno;
```

```
Const
```

```
...
```

```
Type
```

```
    identificador = tipo;
```

```
Módulos
```

```
Var
```

```
    x: identificador;
```

```
...
```

```
Begin
```

```
...
```

```
End.
```

CADP – Tipos de Datos definidos por el prog. (TDDU)

```
program uno;  
Const  
  ...  
Type  
  numeritos = integer;  
var  
  num1,num2: numeritos;  
  num3: integer;  
Begin  
  ...  
End.
```

Por ser un tipo predefinido, no
hace falta especificar valores ni
operaciones



Qué operaciones
puedo hacer con
num1 y num2?

Qué operaciones
puedo hacer con
num3?

Y con num1 y
num3?

CADP – Tipos de Datos definidos por el prog. (TDDU)



```
program uno;  
Const  
    ...  
Type  
    numeritos = integer;  
var  
    num1,num2: numeritos;  
    num3: integer;  
Begin  
    read (num1);  
    num2:= num1 + 4;  
    read (num3);  
    num2:= num 2 DIV 10;  
End.
```

```
program uno;  
Const  
    ...  
Type  
    numeritos = integer;  
var  
    num1,num2: numeritos;  
    num3: integer;  
Begin  
    num3:= num2 + 3;  
End.
```

**NO puede hacerse ninguna operación
que los relacione**

CADP – Tipos de Datos definidos por el prog. (TDDU)



VENTAJAS DE LOS TDDU



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..

CADP – Tipos de Datos definidos por el prog. (TDDU)



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 definidos por el prog. (TDDU)



SUBRANGO

```
program uno;  
Const  
    ...  
Type  
    nombre = valor1..valor2;  
var  
    x,y: nombre;  
  
Begin  
    ...  
End.
```

CADP – Tipos de Datos definidos por el prog. (TDDU)



SUBRANGO

```
program uno;
```

```
Type
```

```
    mayusculas = 'A'..'Z';
```

```
    primer_decena = 1..10;
```

```
    caracDigitos = '0'..'9';
```

```
var
```

```
    letra   : mayusculas;
```

```
    número : primer_decena;
```

```
    digitos: caracDigitos;
```

```
Begin
```

```
    ...
```

```
End.
```

Qué
operaciones
tienen
permitidas?

CADP – Tipos de Datos definidos por el prog. (TDDU)



SUBRANGO – OPERACIONES PERMITIDAS

Operaciones Permitidas

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

Operaciones NO permitidas

- Depende del tipo base.

CADP – Tipos de Datos definidos por el prog. (TDDU)



Realizar un programa que lea para cada uno de 30 alumnos las notas de 5 exámenes rendidos en los 5 días de la semana. Informe para cada alumno el día que obtuvo mas nota.

- Cómo represento la nota de un alumno?
- Cuál es la condición de fin?
- Cómo informo por alumno?

CADP – Tipos de Datos definidos por el prog. (TDDU)



Realizar un programa que lea para cada uno de 30 alumnos las notas de 5 exámenes finales rendidos en los 5 días de la semana. Informe para cada alumno el día que obtuvo mas nota.

Alumno

nota 1 nota 2 nota 3 nota 4 nota 5

Alumno 1

7	4	8	6	2		3
					informa	

Alumno 2

10	4	6	6	8		1
					informa	

CADP – Tipos de Datos definidos por el prog. (TDDU)



Para cada alumno

 Inicializo su máximo

 Para cada día

 Leo nota de examen

 Si (la nota supera el máximo)

 actualizo el máximo

 Informo el día máximo

CADP – Tipos de Datos definidos por el prog. (TDDU)



```
program uno;  
var  
    nota,alumno,max,dia,dia_max: integer;  
Begin  
    for alumno:= 1 to 30 do begin  
        max:=-1;  
        for dia:= 1 to 5 do begin  
            read (nota);  
            if (nota >= max) then  
                begin  
                    max:= nota;  
                    dia_max:= dia;  
                end;  
            end;  
            write (“el día máximo es”, dia_max);  
        end; // de todos los alumnos  
End.
```

Funciona?

*Se puede
mejorar?*



```
program uno;  
Type  
  notas = -1..10;  
var  
  nota,max: notas;  
  alumno,dia,dia_max: integer;  
Begin  
  for alumno:= 1 to 30 do begin  
    max:= -1;  
    for dia:= 1 to 5 do begin  
      read (nota);  
      if (nota >= max) then  
        begin  
          max:= nota;      dia_max:= dia;  
        end;  
      end;  
    write ("el día máximo es", dia_max);  
  end;  // de todos los alumnos  
End.
```

*Podría definir más
subrangos?*

*Qué modifico si quiero
informar en vez de día
1, lunes?*

```

program uno;
Type
  notas = -1..10;
var
  nota,max: notas;
  alumno,dia,dia_max: integer;
Begin
  for alumno:= 1 to 30 do begin
    max:= -1;
    for dia:= 1 to 5 do begin
      read (nota);
      if (nota >= max) then
        begin
          max:= nota;      dia_max:= dia;
        end;
      write ("el día máximo es", dia_max);
    end;
  end; // de todos los alumnos

```

Clase 2 **End.**

```

case dia_max of
  1: write ("Lunes");
  2: write ("Martes");
  3: write ("Miércoles");
  4: write ("Jueves");
  5: write ("Viernes");
end;
end;
end;
End.

```



**Si día_max y dia fueran
de tipo subrango puedo
usar la misma solución?**