



Conceptos de Algoritmos Datos y Programas

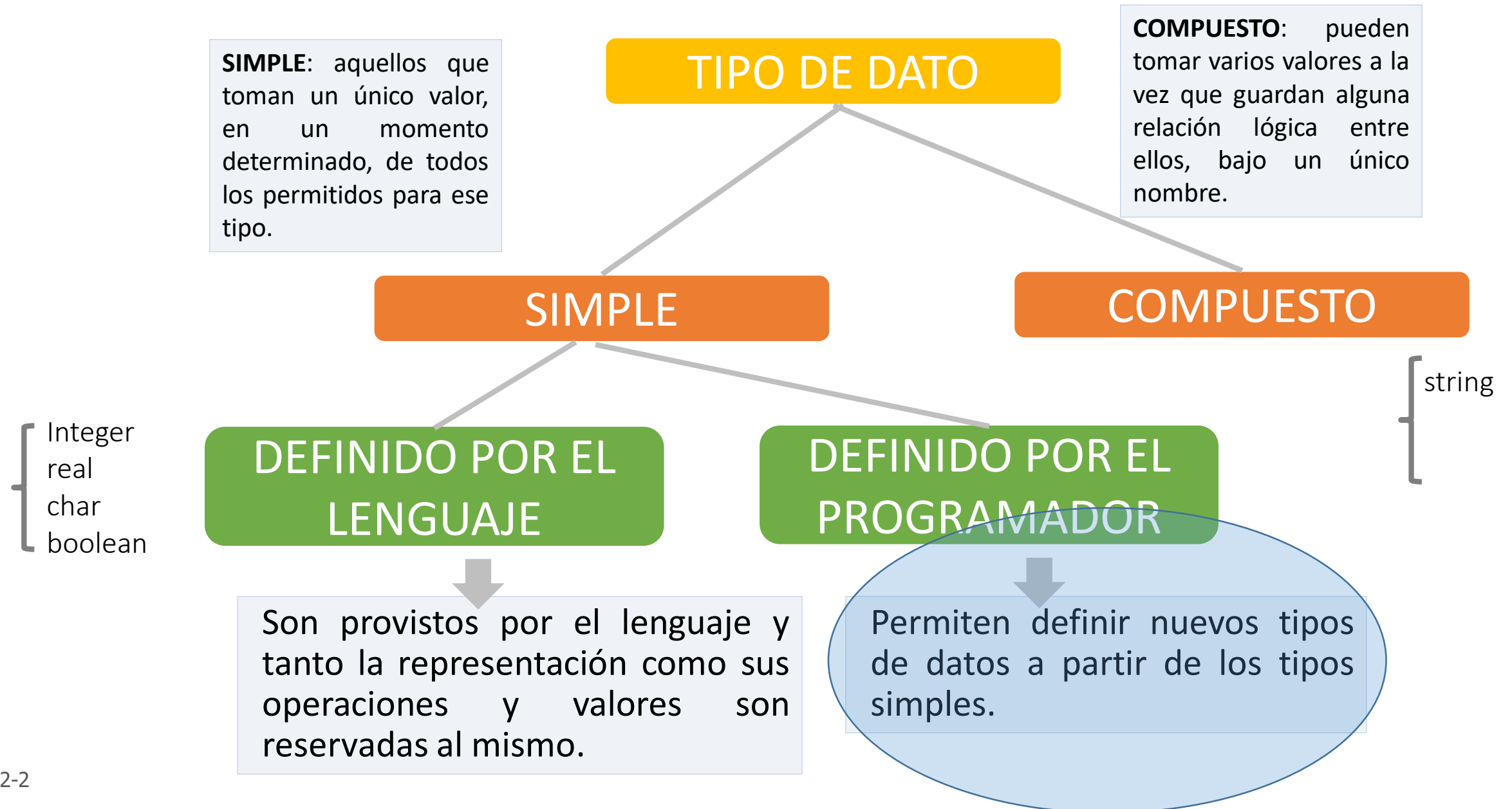
CADP – TEMAS



● Tipos de Datos definidos por el usuario

● EJERCICIO – PREGUNTAS FINALES

CADP – TIPOS DE DATOS- CLASIFICACION

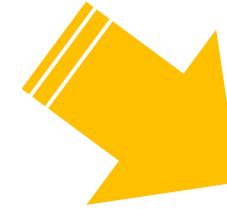


CADP – TIPOS DE DATOS DEFINIDOS POR EL USUARIO



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.

Aumento de la riqueza expresiva del lenguaje, con mejores posibilidades de abstracción de datos.

Mayor seguridad respecto de las operaciones que se realizan sobre cada clase de datos.

Límites preestablecidos sobre los valores posibles que pueden tomar las variables que corresponden al tipo de dato.

**Tipos
Definidos
por el
Usuario**



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: identificador;
```

```
...
```

```
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;
```

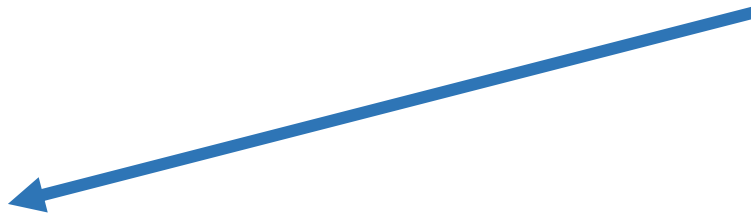
```
valor: nuevotipo1;
```

```
begin
```

```
end.
```

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.
```



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..