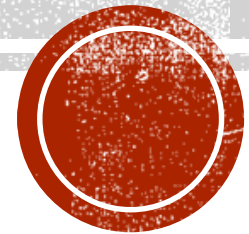


ALGORÍTMICA Y PROGRAMACIÓN II

TIPOS DE DATOS - SIMPLES



TIPOS DE DATOS - SIMPLES

- Previa escritura de un programa
 - Se deberá identificar y estudiar el conjunto de algoritmos involucrados
 - Se deberán identificar los tipos de datos y estructuras a utilizar
- ¿Por qué son importantes los tipos y estructuras de datos?
 - Obtener programas sencillos y eficientes



TIPOS DE DATOS - SIMPLES

- Dato

- Objetos de un programa que me permiten representar la información.
- Solamente consideraremos aquellos de interés para la resolución del problema.

- Abstracción de datos

- Separar las propiedades lógicas de los datos de su implementación.



TIPOS DE DATOS - SIMPLES

- Encapsulamiento de datos
 - El sistema operativo y los lenguajes de programación se encargan de convertir el mundo del programador (modelo de la realidad) y el modelo del hardware.
 - Encapsulan los datos ocultando niveles de implementación.

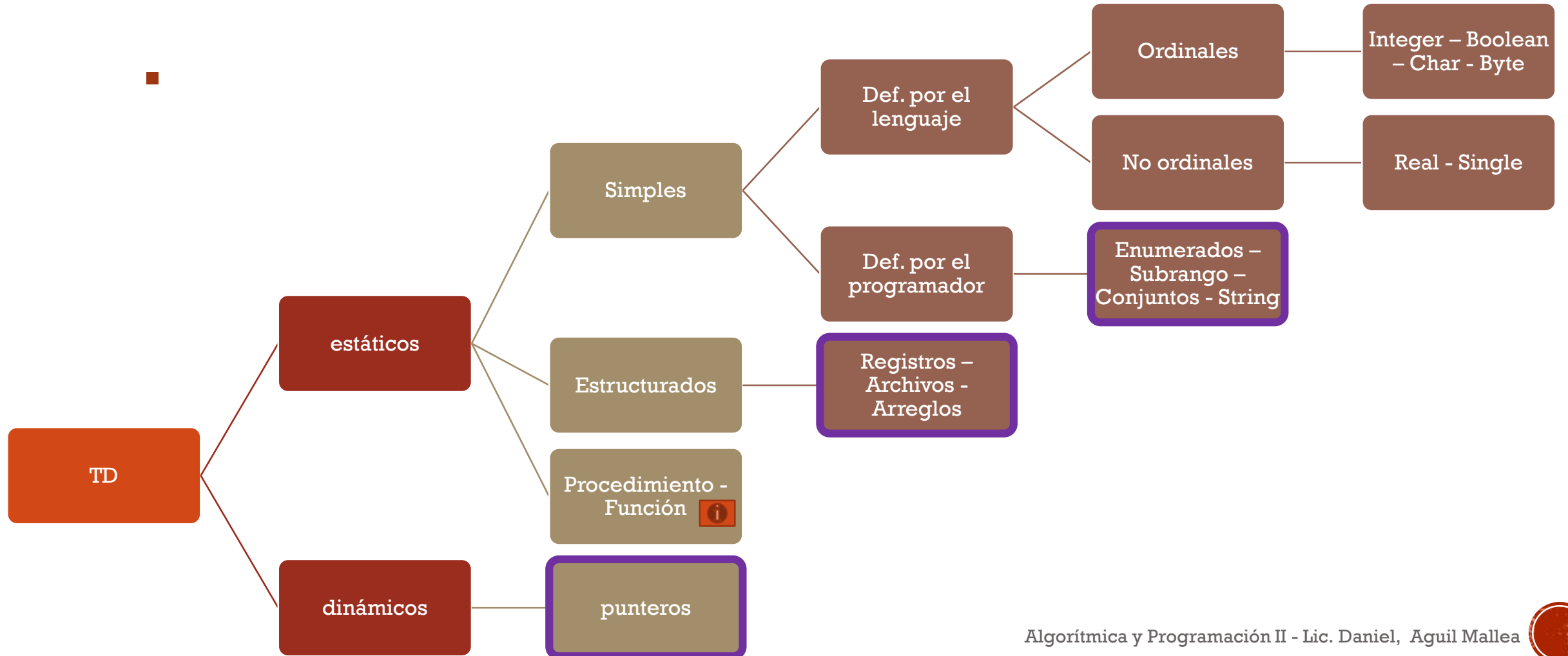


TIPOS DE DATOS - SIMPLES

- Tipo de dato
 - Rango de valores posibles
 - Operaciones
 - Representación interna.



TIPOS DE DATOS - SIMPLES



TIPOS DE DATOS - SIMPLES

- Ventajas de utilizar tipos de datos
 - Flexibilidad : el cambio en su definición me modifica todas las variables.
 - Documentación : el nombre nos lleva a pensar de que se trata.
 - Seguridad : permite realizar prueba de código (con o sin directivas del compilador).



TIPOS DE DATOS - SIMPLES

■ Subrango

- Podemos indicar los posibles valores que pueden tomar las variables.
- Como cuando definimos los índices de un arreglo.

Type

Tletra = 'a'...'z';

Tdia = 1..7;



TIPOS DE DATOS - SIMPLES

- Subrango

- Hay que tener en cuenta la configuración del compilador al momento de pensar que sucede al intentar almacenar un valor fuera del rango establecido.

- $\{\$R+\}$ o $\{\$R-\}$



TIPOS DE DATOS - SIMPLES

- Subrango - Ejercicio

Crear un programa que pida un dato entre el 2010 al 2015 y muestre la toneladas producidas ese año en Argentina. El dato debe ser de tipo subrango, y las Toneladas deben estar guardados en un array de tipo real.



TIPOS DE DATOS - SIMPLES

■ Enumerado

- Permite al programador especificar con mayor detalle los valores que puede tomar una variable.
- Representan valores ordinales (puedo utilizar funciones sucesor `succ()` o precedente `pred()`)

Type

`TDiaLaboral = (Lunes, Martes, Miércoles, Jueves, Viernes);`

`Tmes = (enero, febrero, marzo, abril, mayo, junio);`



TIPOS DE DATOS - SIMPLES

■ Enumerado

- En las implementaciones modernas se pueden leer o escribir directamente (antes se debían realizar conversiones).

Type

Tdia = (lunes, martes, miercoles);

Var

dia:Tdia;

...

Readln(dia); Writeln(dia);



TIPOS DE DATOS - SIMPLES

■ Conjunto

- Una colección de elementos de un mismo tipo, en nuestro caso será un valor ordinal. (char, byte, subrango, enumerado)
- Existía una restricción que el conjunto no podía tener más de 256 valores.

```
Type
  TLetras = Set of char;
...
Letras := [ ];
Letras := ['a','b'];
```



TIPOS DE DATOS - SIMPLES

■ Conjunto - operaciones

Operador	Función
+	Unión – genera un conjunto
-	Diferencia – genera un conjunto
*	Intersección – genera un conjunto
In	True si el elemento se encuentra en el conjunto
=	True si ambos conjuntos son iguales
<>	True si ambos conjuntos son distintos
$A \leq B$	True si el conjunto A esta incluido en B
$A \geq B$	True si el conjunto B esta incluido en A



TIPOS DE DATOS – SIMPLES – EJ1

Type

```
Tletras= set of char;
```

Var

```
lminusculasValidas,
```

```
lmayusculasValidas, letrasValidas: Tletras;
```

Begin

```
minusculasValidas := ['a', 'b', 'c', 'd'];
```

```
mayusculasValidas := ['F', 'H', 'K', 'M'];
```

```
letrasValidas := minusculasValidas + mayusculasValidas;
```

Repeat

```
  writeln('Introduce una letra...');
```

```
  readLn(letra);
```

```
  if not (letra in letrasValidas) then
```

```
    writeln('No aceptada!');
```

```
  until letra in letrasValidas;
```

end.



TIPOS DE DATOS - SIMPLES

▪ String

➤ Enfoque tradicional

- Arreglo de caracteres
- Máxima longitud 255 símbolos (0 + 1..255)
- Operaciones: concatenar (+) – asignar (:=) – comparar (=, >, <)
- American Standard Code for Information Interchange (ASCII) - American citizen



Type

```
TTextoCorto = String[10];
```

...

Var

```
descripcion:TTextoCorto;
```

descripcion := 'hola!'



TIPOS DE DATOS - SIMPLES

■ String

➤ Funciones provistas

- Concat – Copy – Length – Pos – Upcase

➤ Procedimientos provistos

- Insert – Str – Val

Val(texto, Num , cod);

Cant := length(texto);

Existen más funciones y procedimiento para manejo de cadenas en librerías adicionales.

```
program strUtilidades;  
uses strUtils;  
var texto:String;
```

```
begin  
  RANDOMIZE;  
  texto := ifThen(random(2) = 0,  
                  'random = 0',  
                  'random = 1');  
  writeln(texto);  
end.
```



TIPOS DE DATOS - SIMPLES

■ String

➤ Enfoque Moderno

- Extensión ASCII (7-bits) >> ISO8859-1 (8-bits)
 - ShortString (alias de string) – {\$H-} – tiene limite –
 - AnsiString (alias de String) – {\$H+} – sin limite – es tratado como un puntero – se almacena en el heap
- Unicode >> utf8 – utf16
 - UnicodeString – WideString – utf8String
- [International Organization for Standardization](#)
- [Unicode Consortium](#)



TIPOS DE DATOS – SIMPLES – EJ2

```
program cadenas;
```

```
uses crt;
```

```
var
```

```
    s : String[10];
```

```
    i : byte;
```

```
BEGIN
```

```
    s := 'hola!';
```

```
    writeln('longitud: ', ord(s[0]));
```

```
    writeln('longitud: ', length(s));
```

```
    writeln('texto: ', s);
```

```
    writeln('-----');
```

```
// chr(97) = 'a' - chr(98) = 'b'
```

```
for i := 1 to 10 do
```

```
    s[i] := chr(96 + i);
```

```
//
```

```
if(s = 'abcdefghij')then
```

```
    writeln('BIEN')
```

```
else
```

```
    writeln('QUE PASO?');
```

```
//
```

```
    writeln(s);
```

```
END.
```



TIPOS DE DATOS - SIMPLES

■ Ejercicio

Crear un programa que dado un texto ingresado por el usuario informe la cantidad de vocales que posee.



TIPOS DE DATOS – SIMPLES – EJ3

```
VAR
    Frase:String;
    i,c:integer;
BEGIN
    c:=0;
    write('Ingrese una frase : ');
    readln(Frase);

    FOR i:=1 TO Ord(Frase[0]) DO
        IF Frase[i] IN ['a','e','i','o','u','A','E','I','O','U'] THEN
            c+=1;
    Writeln('La Frase ingresada tiene ',c,' vocales');
END.
```



BIBLIOGRAFÍA UTILIZADA

- Algoritmos, Datos y Programas – Armando De Gusti
- Algoritmos y programación – Cristobal Pareja Flores
- [http://wiki.freepascal.org/Character and string types](http://wiki.freepascal.org/Character_and_string_types)
- <https://lazarus-ccr.sourceforge.io/docs/rtl/strutils/index-5.html>
- <https://es.wikipedia.org/wiki/UTF-8>
- <https://perlgeek.de/en/article/encodings-and-unicode>
- <http://www.unicode.org/>

