

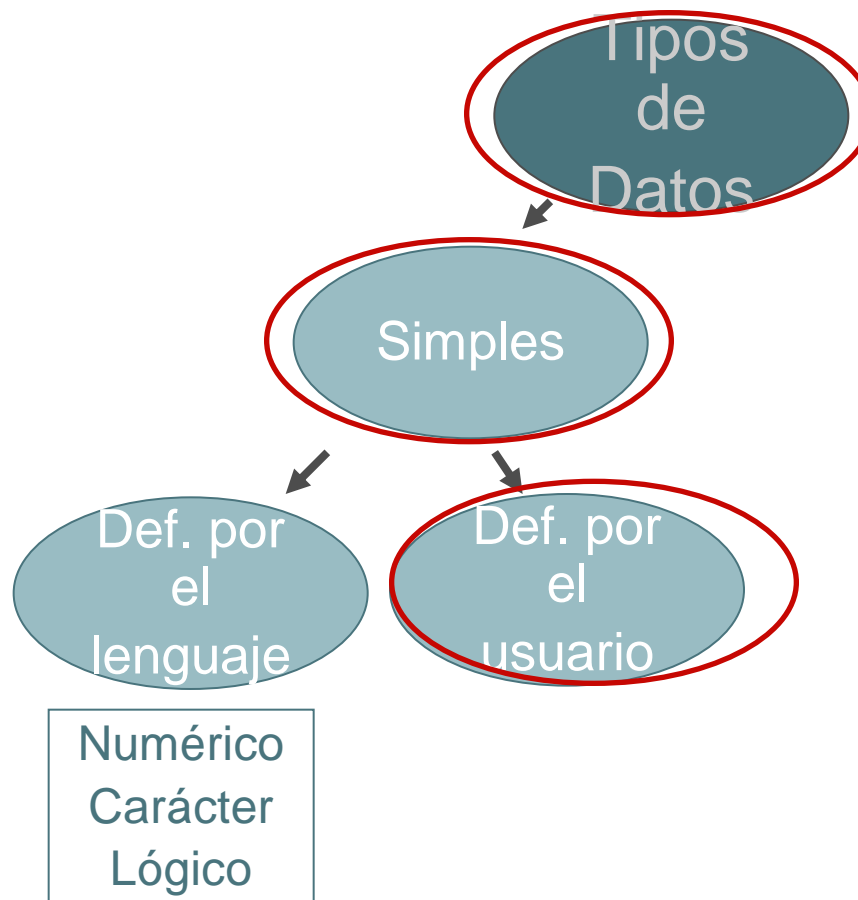
# Programación I

TEMAS  
de la  
CLASE

- 1 Tipos de datos definidos por el usuario
- 2 Tipo de dato estructurado/  
Estructura de datos

# Clasificación de Tipos De datos

**Hasta aquí presentamos los tipos de datos simples** que son aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.



# 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.

Esto permite:

- 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 de datos definidos por el usuario

## ¿Qué ventajas tiene DECLARAR tipos?

- **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.

# 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.

**Sintéticamente entonces un Tipo significa una clase de datos que tiene asociado:**

- Un rango de valores posibles.
- Una forma de representación.
- Un conjunto de operaciones permitidas.
- Un conjunto de condiciones de valores permitidos que se pueden verificar.

# Tipos de datos definidos por el usuario

En Pascal, los tipos deben ser declarados antes de ser usados.

La declaración de tipos se hace a través de la palabra clave TYPE de la siguiente forma:

```
TYPE identificador = tipo;
```

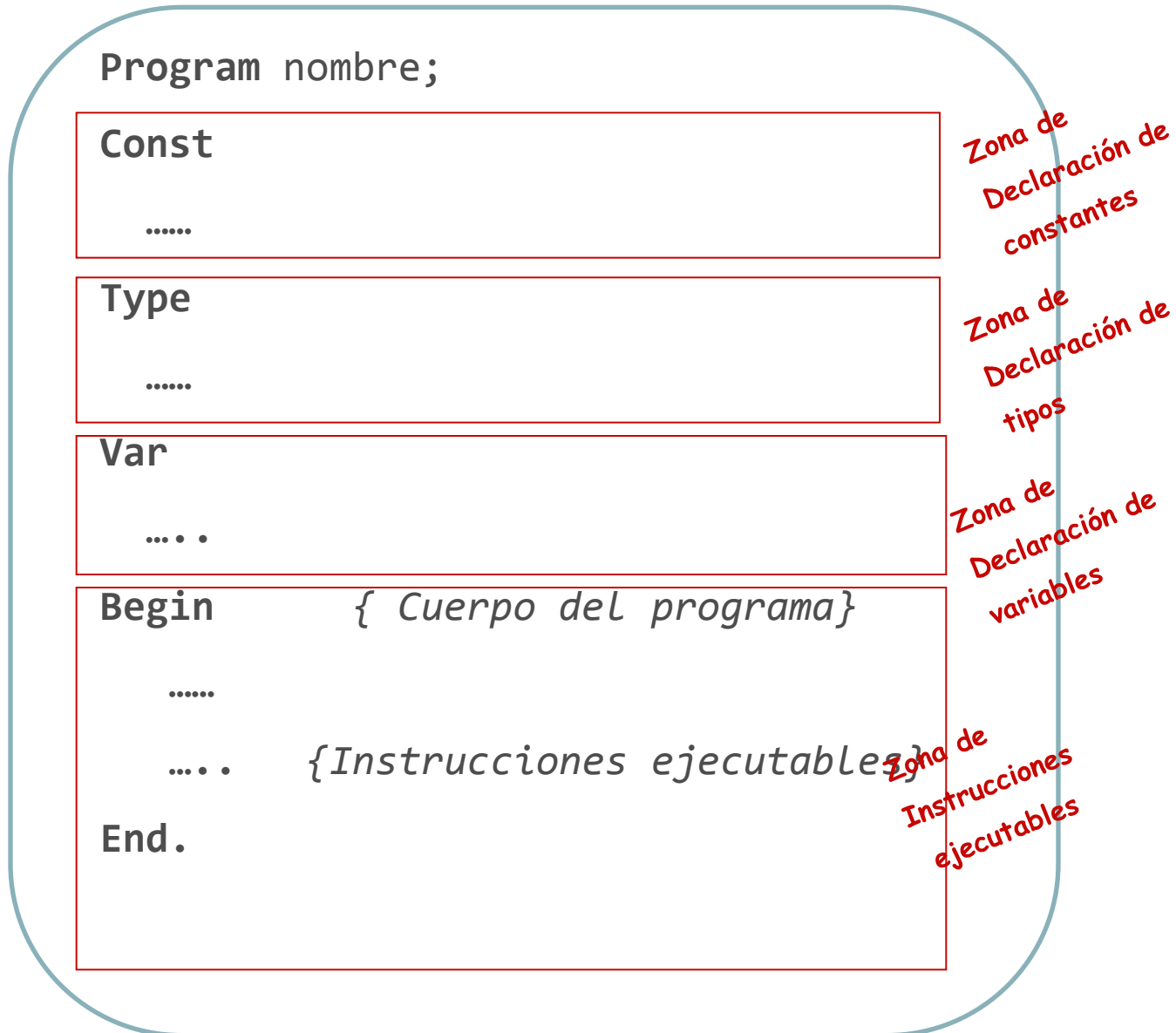


Nombre con que se conocerá al tipo de dato en el programa.



Puede ser un tipo estándar o alguno de los tipos de datos definidos por el usuario.

# Esquema general de un programa que usa tipos definidos por el usuario

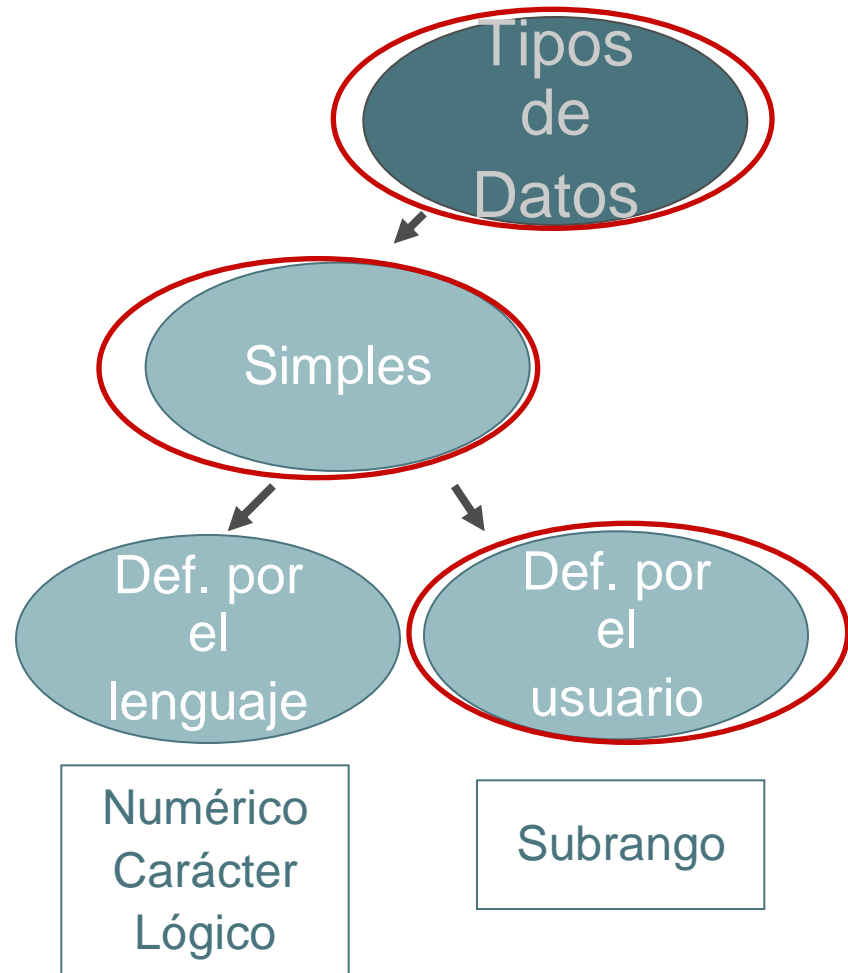




# Clasificación de los Tipos de Datos

**Recordemos la clasificación de los tipos de datos ya vista...**

Ahora vamos a comenzar a trabajar con un tipo de dato **simple y definidos por el usuario**.



# Tipo de dato definido por el usuario: SUBRANGO

Un tipo de dato **subrango** es un tipo simple y ordinal que consiste de una sucesión de valores extraídos de un tipo ordinal base.

Para declarar un tipo **SUBRANGO** se deben especificar los valores inicial y final de la sucesión, separados por dos puntos seguidos:

```
Type  identificador = valor inicial .. valor final;
```

# Tipo de dato definido por el usuario: SUBRANGO

**Program** nombre;

**Const**

fin = 1999;

**Type**

siglo\_pasado= 1900 .. Fin;

mayusculas = 'A' .. 'Z';

**Var**

letra: mayusculas;

año: siglo\_pasado;

**Begin**            { *Cuerpo del programa*}

...

...            {*Instrucciones ejecutables*}

**End.**

# Tipo SUBRANGO - Operaciones

- Las **operaciones** de un tipo de dato subrango se heredan del tipo base.

Los tipos base de los que se pueden definir subrangos son:

- ✓ Enteros
- ✓ Caracteres

- La ocupación en memoria estará condicionada por el tipo base

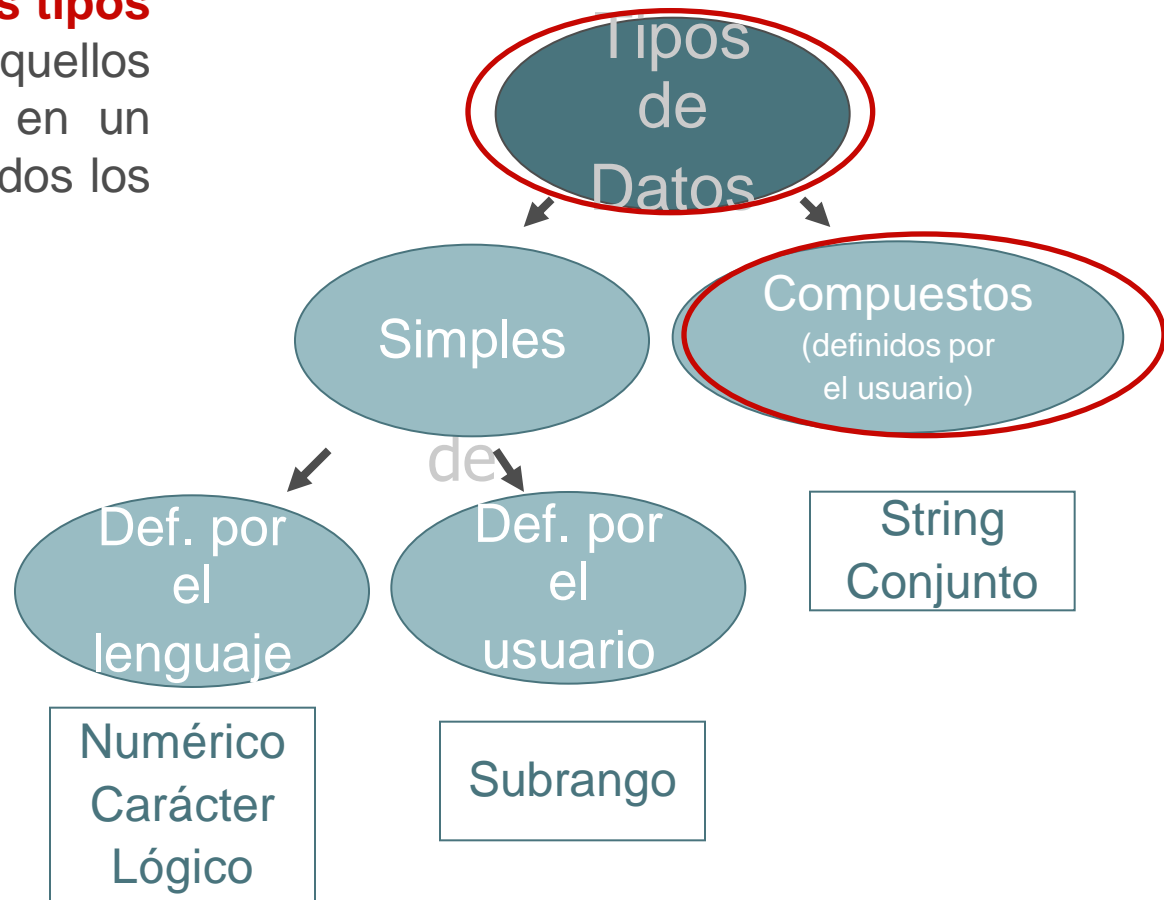
```
Program ejemplo;  
Const  minimo = 1;  
        maximo = 500;  
  
Type  
    rango = minimo .. maximo;  
    meses = 1 .. 12;  
    letras = 'A' .. 'z';  
  
Var  
    dato1, dato2 : rango;  
    descanso : meses;  
    letra: letras;  
  
Begin  
    dato1 := dato1 Div dato2;  
    descanso:= 1;  
    Read (letra);  
    ...  
End.
```



Ejercicio 1: Realice un programa que lea edades de 20 alumnos de la facultad e informe el promedio de edades y la edad más grande.

# Clasificación de Tipos De datos

- **Hasta aquí presentamos los tipos de datos simples** que son aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.



- **Ahora comenzaremos a trabajar con los tipos de datos compuestos** que son aquellos que pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos.

# Tipo de dato definido por el usuario: STRING

Un tipo de dato string es una sucesión de caracteres de longitud determinada.

```
TYPE identificador = string [ longitud ];
```

- Longitud es el número máximo de caracteres que puede contener el dato.
- En PASCAL cuando no se especifica la longitud ese identificador podrá contener como máximo 255 caracteres.
- La cantidad de memoria que utiliza una variable está determinada por su longitud. Recordar que cada carácter ocupa 1 byte.

# Tipo de dato definido por el usuario: STRING

**Program** nombre;

**Type**

cadena10 = string [10];

cadena25 = string [25];

fecha = string [8];

día = string [2];

**Var**

nom1, nom2, nom3 : cadena10;

apellido : cadena25;

fecha1, fecha2 : fecha;

**Begin** { *Cuerpo del programa*}

.....

.... {*Instrucciones ejecutables*}

**End.**



# Tipo de dato STRING: Operaciones

Las operaciones permitidas son:

- **Asignación ( $:=$ )**
- **Entrada/Salida (Read / write)**
- **De relación ( $>, <, =, \dots$ )**



Ejercicio 2: Realice un programa que lea nombres y DNI de personas hasta leer el nombre 'Ana', que debe procesarse. Se debe informar la cantidad personas cuyo DNI es par e informar el nombre de las personas cuyo DNI tiene exactamente 3 dígitos con valor 5.

# Tipo CONJUNTO - Características

- Desde el punto de vista informático un tipo conjunto representará una colección de datos simples (además los datos que estarán guardados en el conjunto deben ser de tipo ordinal), sin repetición y limitada por la implementación en cada lenguaje o sistema operativo.
- No necesariamente existe en la mayoría de los lenguajes.
- Es un tipo de datos compuesto.
- No es un tipo de datos ordinal.

# Tipo CONJUNTO - Características

- Se pueden tener conjuntos de valores cuyos tipos serán integer, boolean o char.
- En la implementación de Pascal el conjunto no puede tener más de 255 elementos (en la práctica esto no se tendrá en cuenta).
- No permite operaciones de lectura - escritura.
- Permite la asignación, unión, intersección, pertenencia, diferencia

# Tipo CONJUNTO – Definición en Pascal

## Type

identificador = **set of** *tipo ordinal*;

letras = set of char;

## Var

conjLetras: letras;

# Tipo CONJUNTO – Operaciones

## ■ Asignación

```
Program conj1;  
Type  
  letras = set of char;  
Var  
  letras1, letras2, letras3: letras;  
  
Begin  
  letras1:= [];  
  letras2:= ['a'..'f'];  
  letras3:= letras2;  
  ...  
End.
```

# Tipo CONJUNTO – Operaciones

## ■ Unión

- ❑ Se representa con el signo + y da como resultado otro conjunto.
- ❑ En este conjunto resultado aparecen los elementos de los dos conjuntos y aquellos elementos repetidos aparecen una vez.

```
Program conj2;  
Type  
    conjcar = set of char;  
Var  
    carac1,carac2: conjcar;  
  
Begin  
    carac1:= ['E', '9'];  
    carac2:= ['a'] + carac1;  
    ...  
End.
```

# Tipo CONJUNTO – Operación Intersección

## ■ Intersección

□ Se representa con el signo \* y da como resultado otro conjunto. En el conjunto resultado aparecen solamente los elementos comunes a los dos conjuntos.

```
Program conj3;
```

```
Type
```

```
  conjcar = set of char;
```

```
Var
```

```
  carac1, carac2: conjcar;
```

```
Begin
```

```
  carac1:= ['E', '9'];
```

```
  carac2:= ['a'] * carac1;
```

```
  ...
```

```
End.
```



# Tipo CONJUNTO – Operación Diferencia

## ■ Diferencia

- Se representa con el signo - y da como resultado otro conjunto.
- Este conjunto resultado contiene los elementos que están en el primer conjunto y no están en el segundo.

**Program** conj4;

**Type**

conjcar = set of char;

**Var**

carac1,carac2: conjcar;

**Begin**

carac1:= ['E', '9'];

carac2:= ['a','E'] - carac1;

...

**End.**

# Tipo CONJUNTO – Operación Pertenencia

## ■ Pertenencia

- ❑ Se representa con el operador `in` y da como resultado un valor lógico.
- ❑ Esta operación devuelve verdadero si el elemento está en el conjunto y falso en caso contrario.

```
Program conj5;
```

```
Type
```

```
    conjcar = set of char;
```

```
Var
```

```
    carac1,carac2: conjcar;
```

```
Begin
```

```
    carac1:= ['E', '9'];
```

```
    if ('a' IN carac1) then
```

```
        ...
```

```
End.
```

# Tipo CONJUNTO – Operaciones

## ■ Comparación

❑ Se pueden usar los operadores relacionales para determinar si un conjunto está incluido en otro ( $\leq$ ), si son distintos ( $\neq$ ) ó iguales ( $=$ ).

**Program** conj6;

**Type**

conjcar = set of char;

**Var**

carac1, carac2: conjcar;

**Begin**

carac1:= ['E', '9'];

carac2:= ['E', '9', 'F'];

if (carac1  $\leq$  carac2)

then ...

**End.**



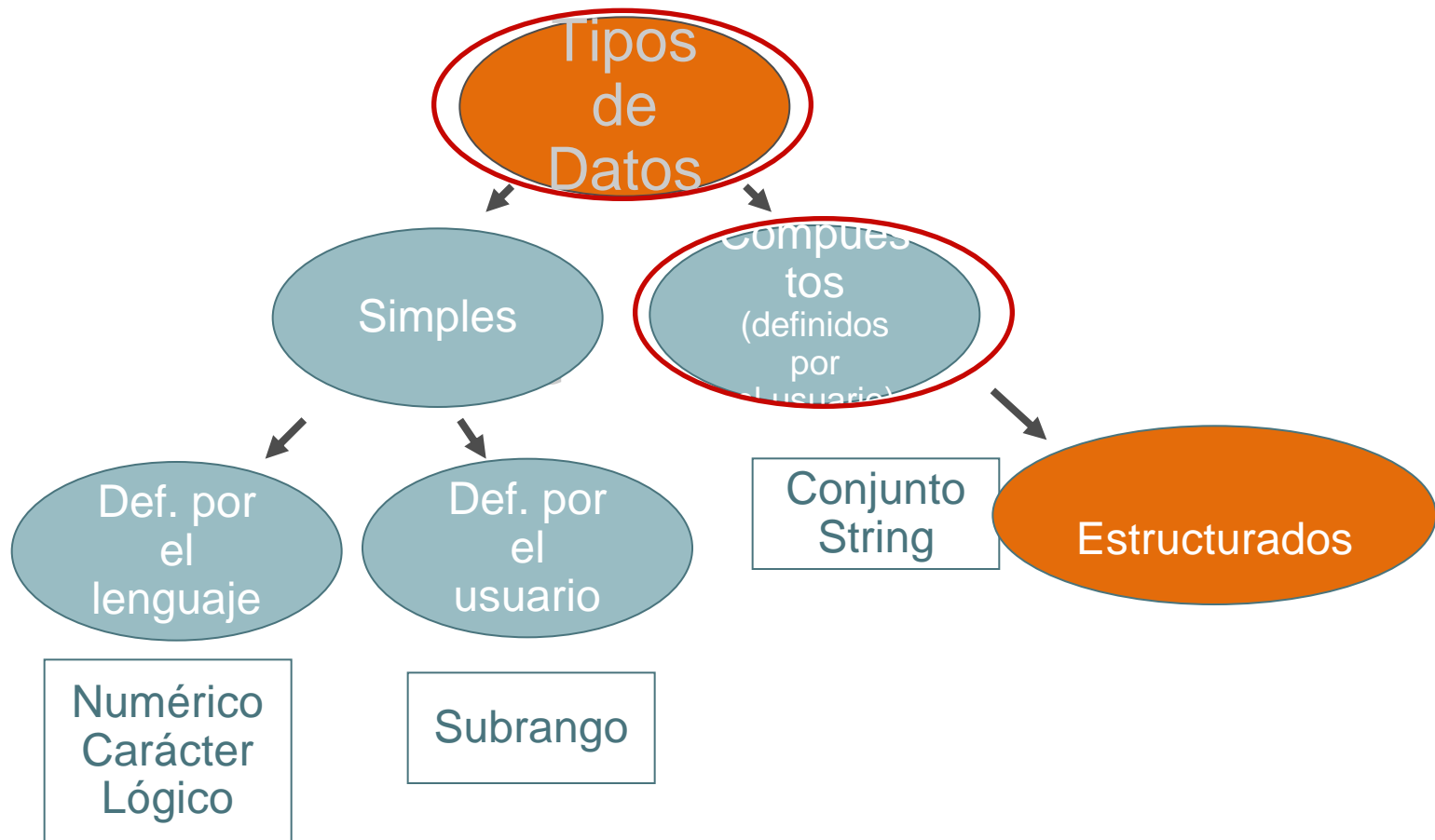
Ejercicio 3: Realice un programa que lea caracteres hasta leer el carácter '@', al finalizar informe la cantidad de consonantes minúsculas y la cantidad de vocales minúsculas leídas.

1

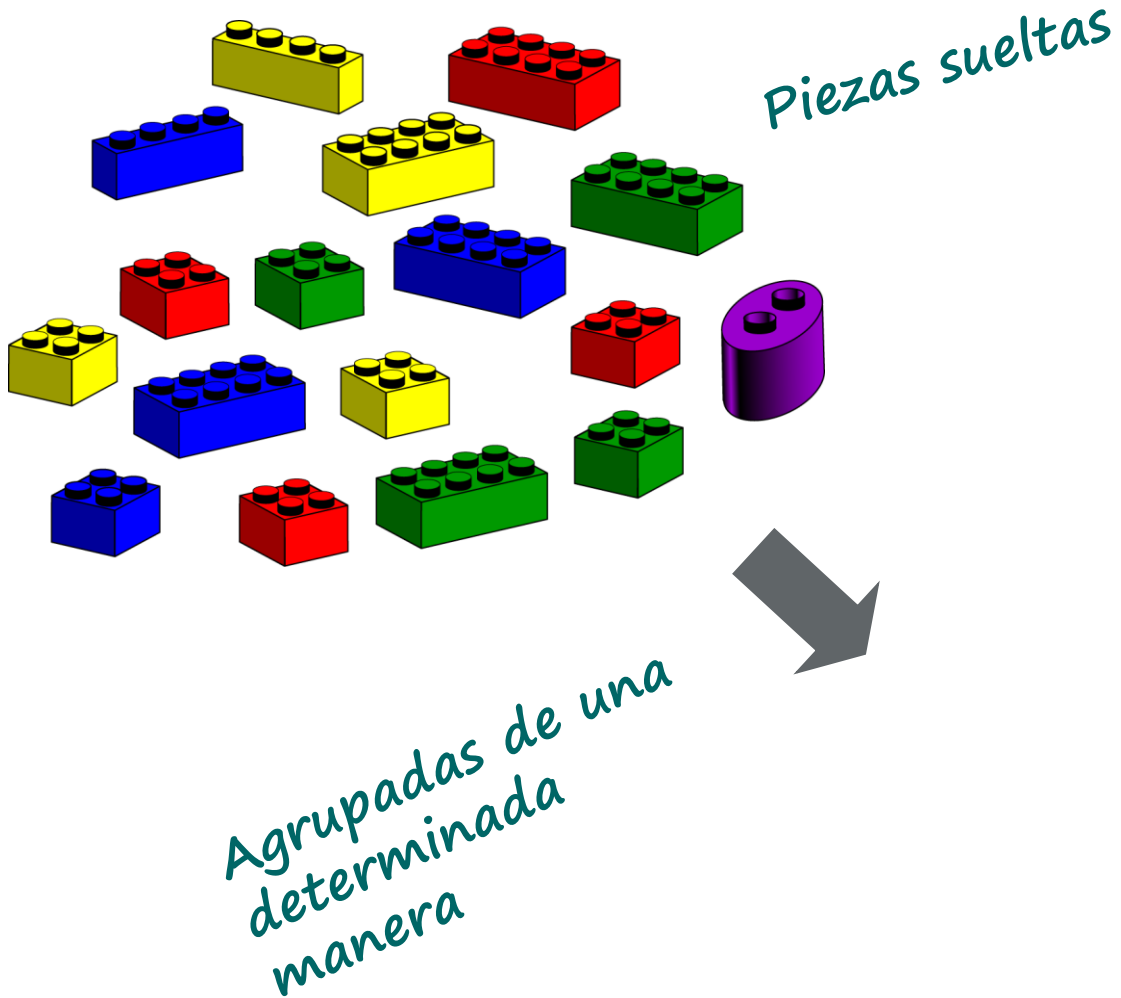
## Tipo de dato estructurado/ Estructura de datos

Concepto

Clasificación



# Estructuras de datos – Motivación



Mafalda

# Estructuras de datos

Una estructura de datos es un conjunto de variables (que podrían ser de distinto tipo) relacionadas entre sí y que se puede operar como un todo, bajo un nombre único.

Esto nos va permitir representar los elementos del mundo real, que generalmente son más complejos que un número entero o una palabra.



# Estructuras de datos

Representar los datos de **empleados** de una empresa. Se identifica a través del nombre, el número de documento, la fecha de nacimiento, el número de legajo, el sexo, el sueldo, la antigüedad, etc.



Representar los datos de un **producto** del **supermercado**. Se debería identificar el código, la marca, la identificación, el precio, la fecha de vencimiento, etc.

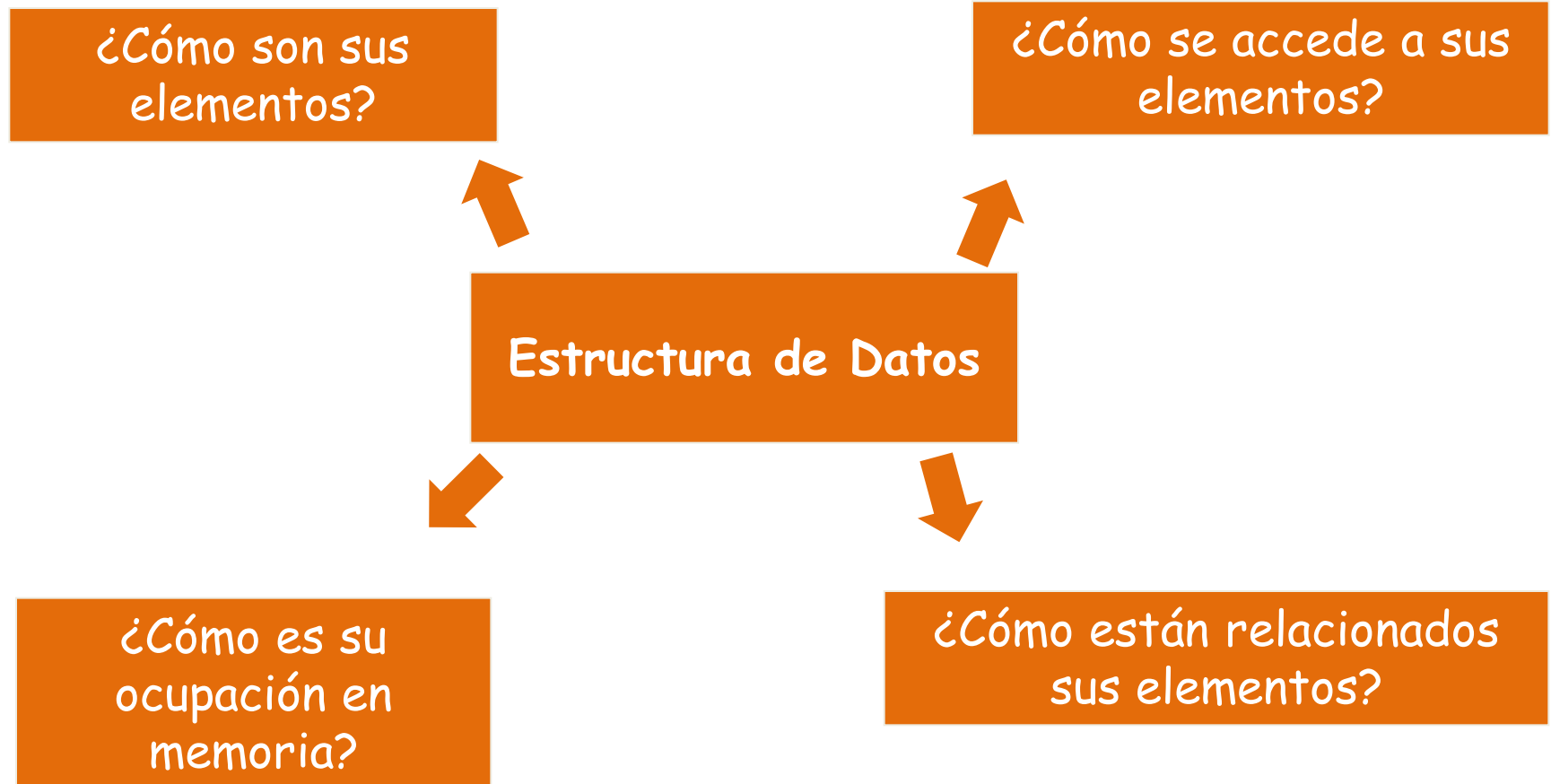


Representar la **lista de clubes de un torneo de fútbol**. Se deberían considerar los nombres de los equipos de fútbol.

Huracán  
Independiente  
Lanús  
San Lorenzo  
Arsenal  
Banfield  
Colón  
Newells  
Gimnasia (LP)  
Estudiantes  
Vélez  
Olimpo  
Quilmes  
San Martín (SJ)

# Estructuras de datos - Clasificación

Las estructuras de datos se pueden clasificar desde distintos puntos de vista:



# Estructuras de datos - Clasificación

De acuerdo a los tipos de datos que se pueden almacenar en la estructura:



Cola de personas

## Estructura de Datos

Homogénea

Heterogénea



Los objetos del escritorio



Los Envases Plásticos



Los Animales domésticos



Las partes de mi computadoras

# Estructuras de datos - Clasificación

Las estructuras de datos pueden clasificarse, de acuerdo al tipo de datos que la componen en homogéneas y heterogéneas.

Una estructura de datos se dice **homogénea** si los datos que la componen son todos del mismo tipo.

Una estructura de datos se dice **heterogénea** si los datos que la componen son de distinto tipo.



Cola de personas



Los Envases Plásticos



Los Animales domésticos



Las partes de mi computadoras



Los objetos del escritorio

# Estructuras de datos - Clasificación

De acuerdo a la ocupación de memoria las estructuras pueden ser:



Las butacas del teatro

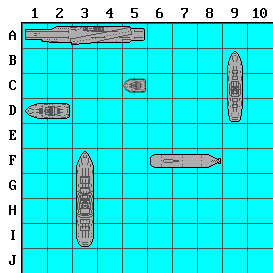
## Estructura de Datos

Estática

Dinámica



Una pila de canastos



El cuadro para el juego de la Batalla Naval



La fila de personas frente a la ventanilla de pagos

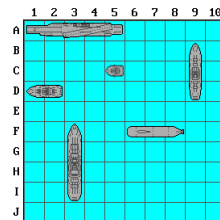
# Estructuras de datos - Clasificación

Las estructuras de datos pueden clasificarse, de acuerdo a la ocupación de memoria en estáticas y dinámicas.

Una estructura de datos se dice **estática** si la cantidad de elementos que contiene es fija, es decir que la cantidad de memoria que ocupa no varía durante la ejecución del programa.



Las butacas del teatro



El cuadro para el juego de la Batalla Naval

Una estructura de datos se dice **dinámica** si la cantidad de elementos que contiene es variable, y por lo tanto la cantidad de memoria ocupada puede cambiar durante la ejecución de un programa.



La pila de canastos

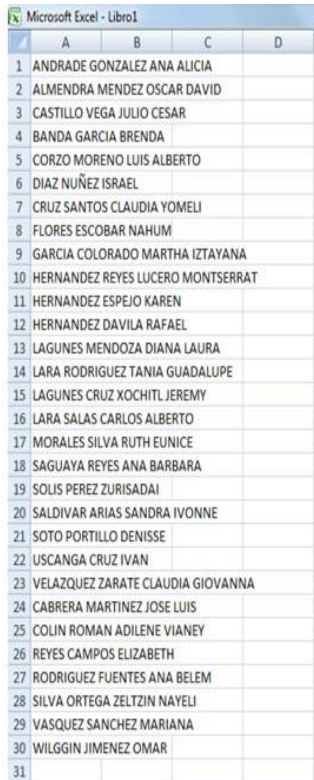


La fila de personas frente a la ventanilla de pagos



# Estructuras de datos - Clasificación

De acuerdo al acceso a sus elementos, las estructuras pueden ser:



	A	B	C	D
1	ANDRADE GONZALEZ ANA ALICIA			
2	ALMENDRA MENDEZ OSCAR DAVID			
3	CASTILLO VEGA JULIO CESAR			
4	BANDA GARCIA BRENDA			
5	CORZO MORENO LUIS ALBERTO			
6	DIAZ NUÑEZ ISRAEL			
7	CRUZ SANTOS CLAUDIA YOMELI			
8	FLORES ESCOBAR NAHUM			
9	GARCIA COLORADO MARTHA IZTAYANA			
10	HERNANDEZ REYES LUCERO MONTSERRAT			
11	HERNANDEZ ESPEJO KAREN			
12	HERNANDEZ DAVILA RAFAEL			
13	LAGUNES MENDOZA DIANA LAURA			
14	LARA RODRIGUEZ TANIA GUADALUPE			
15	LAGUNES CRUZ XOCHITL JEREMY			
16	LARA SALAS CARLOS ALBERTO			
17	MORALES SILVA RUTH EUNICE			
18	SAGUAYA REYES ANA BARBARA			
19	SOLIS PEREZ ZURISADAI			
20	SALDIVAR ARIAS SANDRA IVONNE			
21	SOTO PORTILLO DENISSE			
22	USCANGA CRUZ IVAN			
23	VELAZQUEZ ZARATE CLAUDIA GIOVANNA			
24	CABRERA MARTINEZ JOSE LUIS			
25	COLIN ROMAN ADILENE VIANEY			
26	REYES CAMPOS ELIZABETH			
27	RODRIGUEZ FUENTES ANA BELEM			
28	SILVA ORTEGA ZELTZIN NAYELI			
29	VASQUEZ SANCHEZ MARIANA			
30	WILGGIN JIMENEZ OMAR			
31				



## Estructura de Datos

De acceso  
secuencial

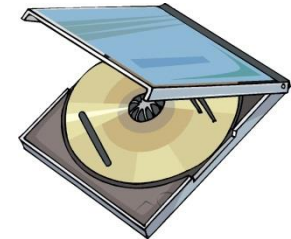
De acceso  
directo



Seleccionar una  
aplicación determinada



Localizar una imagen en un rollo de película



Seleccionar una canción determinada

# Estructuras de datos - Clasificación

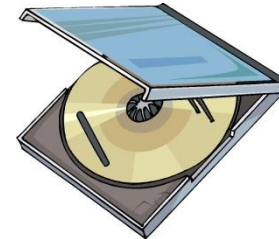
Las estructuras de datos pueden clasificarse, de acuerdo a como se accede a sus elementos, de acceso secuencial o directo.

Una estructura de datos se dice de **acceso secuencial**, si para acceder a un elemento particular se debe respetar un orden predeterminado, por ejemplo, pasando por todos los elementos que le preceden.



Localizar una imagen en un rollo de película

Una estructura de datos se dice de **acceso directo**, si se puede acceder a un elemento particular, directamente, sin necesidad de pasar por los anteriores a él, por ejemplo, indicando una posición.



Seleccionar una canción determinada



# Estructuras de datos - Clasificación

De acuerdo a la relación entre sus elementos, las estructuras pueden ser:



Productos a cobrar en la caja del supermercado

## Estructura de Datos

Lineales

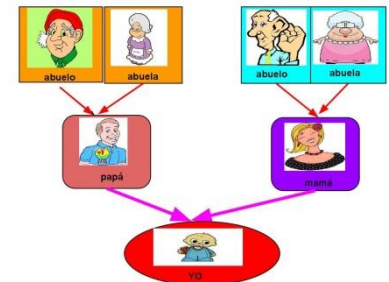
No lineales



Las rutas que unen localidades de la Pcia de Bs. As.



Los vehículos que pasan por la cabina de peaje (cada uno tiene uno antes y uno después)



Nuestro árbol genealógico

# Estructuras de datos - Clasificación

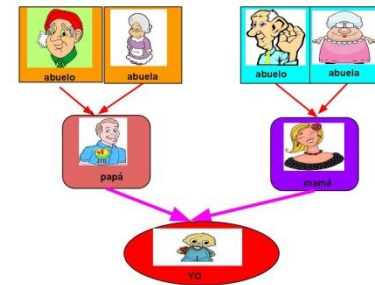
Las estructuras de datos pueden clasificarse, de acuerdo a su linealidad en Lineales y No Lineales.

Una estructura de datos se dice **lineal** cuando está formada por ninguno, uno o varios elementos que guardan una relación de adyacencia ordenada donde a cada elemento le sigue uno y le precede uno, **solamente**.



Productos a cobrar en la  
caja del supermercado

Una estructura de datos se **dice No lineal** si para un elemento dado pueden existir 0, 1 ó mas elementos que le suceden y 0, 1 ó mas elementos que le preceden.



Nuestro arbol genealógico

# Estructuras de datos - Resumen

Las estructuras de datos se pueden clasificar desde distintos puntos de vista:

