



Registros

Arreglos de Registros



Registros

Un *Registro* es una estructura de datos que permite almacenar una colección finita de datos no necesariamente del mismo tipo.

A cada una de las partes que forman la nueva estructura de datos se la denomina **campo**.

Un registro, al igual que un arreglo, ocupa celdas consecutivas de memoria.



Registros

Definición del tipo

```
registro <identificador registro>
{
    <tipo> <identificador de campo1>
    <tipo> <identificador de campo2>
}
```

Declaración de variables

```
<identificador registro> <identificador Variable-Registro>
```



Registros

Ejemplo

Suponiendo que se desea definir una variable de tipo registro que contenga el nombre, número de documento y nota obtenida en un parcial por un alumno.

Declaración del tipo registro:

Registro **alumno**

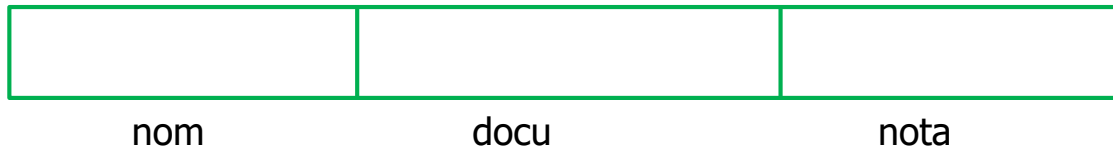
```
{ cadena nom  
  entero docu  
  real nota  
}
```

Declaración de la variable tipo registro:

alumno alu

Esquemáticamente, la variable alu:

alu





Registros

Acceso a los campos de un registro

Los campos de un registro son accedidos utilizando el identificador de campo en vez de la posición relativa como sucede en los arreglos, de la siguiente forma:

<Identificador de registro> . <Identificador de campo>

Esta expresión recibe el nombre de selector de campo, identificado con el símbolo “.” (punto)

Ejemplo:

alu.nom

alu.docu

alu.nota



Registros

Operaciones sobre registros

Las operaciones básicas sobre un registro son: lectura, escritura y asignación.

Lectura y Escritura

Por ser un registro una estructura compuesta, se deben efectuar las operaciones de **lectura y/o escritura individualmente** para cada uno de sus campos.

Leer **alu.nom**, **alu.docu**
Escribir **alu.nota**

Asignación

La asignación entre registros está permitida. Si **a** y **r** son variables registros del **mismo tipo**, la acción **a=r** copia todos los valores asociados con el registro **r** al registro **a**.
Si se declaran dos variables **alu** y **egresado** del tipo alumno:

alumno alu, egresado

La acción de asignación de registros equivale a:

egresado = alu;

egresado.nom = alu.nom
egresado.docu = alu.docu
egresado.nota = alu.nota

Registros

Anidamiento de Registros

Los campos de un registro pueden ser a su vez un registro, en este caso se trabajará con registros anidados. Si se considera el campo domicilio en el registro alumno anterior

Registro **alumno**

{ cadena **nom**
entero **docu**

Registro **domi**

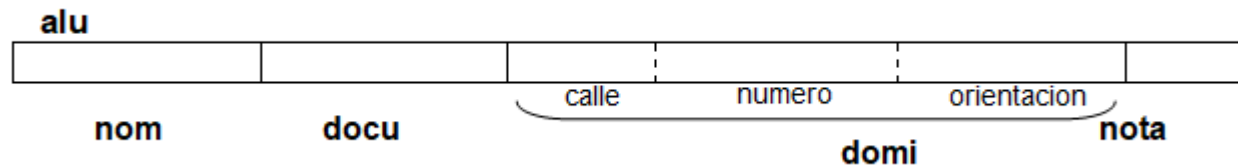
{ cadena calle
entero numero
caracter orientacion
}

real **nota**

}

alumno alu

Esquema de la variable registro **alu**:





Registros

Arreglos como campos de un registro

Los campos de un registro pueden contener arreglos.

Suponer que en lugar de una única nota, se quiere registrar las notas correspondientes a 5 evaluaciones realizadas al alumno.

La estructura que permitirá almacenar esta información es un registro como:

```
Registro alumno
{
  cadena nom
  entero docu
  registro domi
    {
      cadena calle
      entero numero
      caracter orientacion
    }
  real nota [5]
}
alumno alum
```

Para leer o mostrar en forma secuencial las notas se debe utilizar la acción Para..FinPara. Así por ejemplo si se quiere almacenar las notas en el registro:

```
Para i desde 0 hasta 4
  Leer alum.nota [i]
finpara
```


Arreglo de Registros

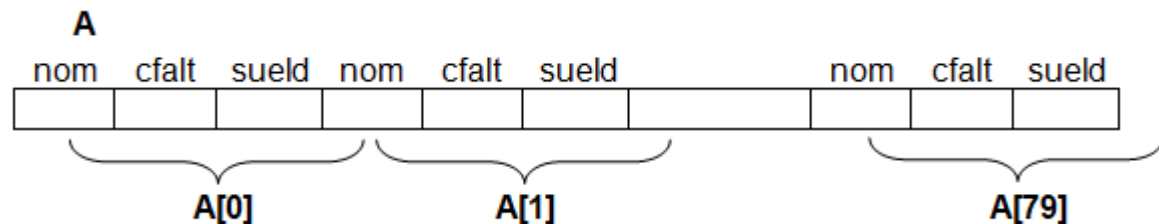
Paso 1: se define el tipo registro:

```
registro empleado  
{ cadena nom  
  entero cfalt  
  real sueld  
}
```

Paso 2: se declara la variable arreglo:

empleado A[80]

Gráficamente la variable A:





Arreglo de Registros

Acceso a las componentes

Algoritmo Arreglo_Registro

```
registro empleado
    { cadena nom
      entero cfalt
      real sued
    }

Void Carga (empleado Ar[80] )
    comienzo
    entero i
        Para i desde 0 hasta 79
            Leer Ar[i].nom, Ar[i]. cfalt, Ar[i].sued
        finpara
    retorna()
fin
```

.....

// **Algoritmo Principal**

Comienzo
empleado A[80]
Carga (**A**)

.....
.....

fin