

6. Vectores

Fundamentos de Informática

Dpto. Lenguajes y Sistemas Informáticos

Curso 2012 / 2013



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniaritzako
Unibertsitate Eskola
Vitoria-Gasteiz



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Índice

Vectores

1. Introducción
2. Declaración
3. Acceso
4. Paso de parámetros
5. Recorrido completo
6. Recorrido de búsqueda
7. Inicialización

1. Introducción

- **Necesidad**
 - Agrupar datos **del mismo tipo** bajo un mismo nombre
 - Vectores, matrices, tablas, arrays, listas, ...
- **Acceso**
 - Se accede a los elementos por medio de un **índice**
 - Las operaciones sobre vectores se hacen **elemento a elemento** (e.g. no se puede copiar un vector de golpe)
- **Declaración**
 - Se declaran especificando los **índices**
 - A menudo no conocemos cuántos elementos vamos a necesitar por lo que nos curaremos en salud
- **Paso de parámetros**
 - Los vectores se pasan siempre por **referencia**

2. Declaración (I)

- Especificar el intervalo de índices
 - Se especifica el índice inferior y el superior
 - Ejemplo (7 elementos):

Dim v(-1 To 5) As Double

-1	0	1	2	3	4	5
48.12	-256.901	18.001	-243.7	8.231	0.3818	4.21231

- Ha de ser un intervalo válido
 - Ejemplo error:

Dim v(10 To 1) As Integer



2.1 Declaración (II)

- **Especificar el índice máximo (alternativa)**
 - Podemos declarar un vector indicando el índice máximo
 - El índice mínimo es el 0
 - Ejemplo (**7 elementos**):

```
Dim ds(6) As String
```

0	1	2	3	4	5	6
"lunes"	"martes"	"miércoles"	"jueves"	"viernes"	"sábado"	"domingo"

- **Ha de ser un intervalo válido**
 - Ejemplo error:

```
Dim v(-1) As String
```



3. Acceso

- **Elemento**

- Accederemos a un elemento indicando su índice en el vector entre paréntesis

- Ejemplo:

```
pctRes.Print v(5)
```

- **Índices extremos del vector**

- Visual Basic nos permite conocer los índices de los elementos extremos de un vector:

- **LBound** (*LowerBound*): índice inferior
- **UBound** (*UpperBound*): índice superior

- Frecuentemente no ocuparemos todo el vector por lo que el elemento en la posición **UBound** estará sin definir

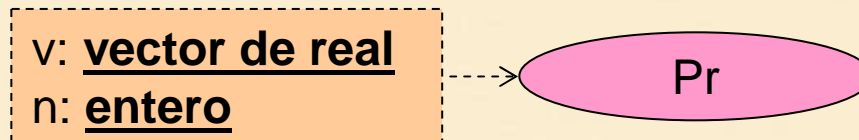
4. Paso de parámetros (I)

- **Por referencia**
 - El paso de parámetros por valor implicaría la copia de todos los elementos, lo cual suele ser lento e innecesario
 - La mayoría de los lenguajes de programación impiden el paso de vectores por valor. Si el programador desea obtener una copia puede realizarla y enviarla por referencia
- **Diagrama de flujo**
 - Para diseñar la cabecera de un subprograma habrá que poner atención a si se leen los datos previos del vector (parámetro de entrada) y/o se modifica el valor de los elementos del vector (parámetro de salida)

4. Paso de parámetros (II)

- Visual Basic

- Al pasar los vectores por referencia **no está claro** si son de entrada y/o de salida
- Se especifica que un parámetro es un vector (abriendo y cerrando paréntesis) pero no se especifica el tamaño ni los índices
- Se puede llamar a un subprograma con vectores de distinto tamaño
- Ejemplo de cabecera de un subprograma que recibe un vector (e.g. para mostrar su contenido en un picture box):



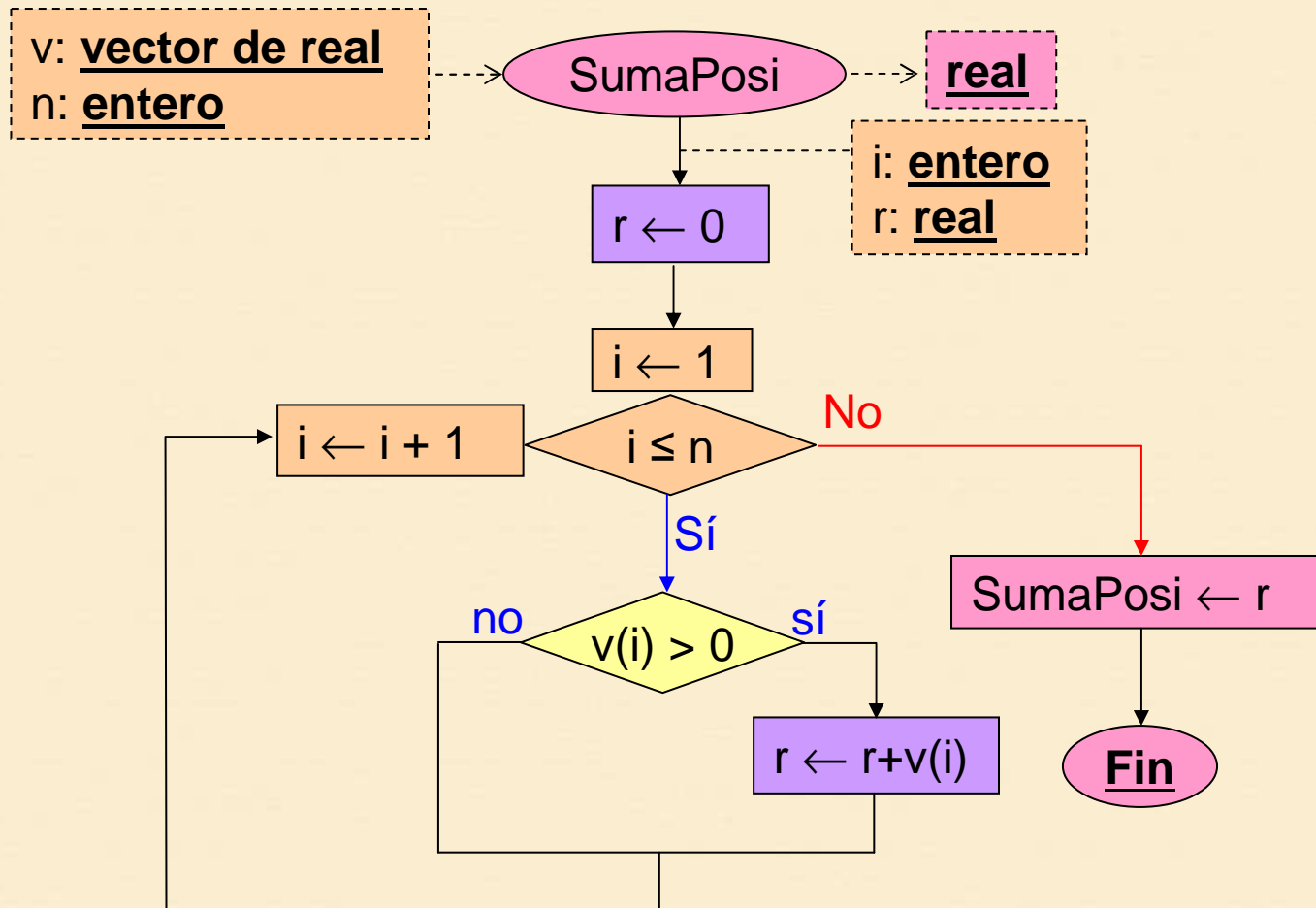
Sub Pr (ByRef tbl () As Double , ByVal n As Integer)



5. Recorrido completo

- **Ejemplos**
 - Sumar los elementos de un vector
 - Contar los elementos del vector que cumplen una propiedad
 - Realizar una operación sobre todos los elementos del vector
- **Modelo general de resolución**
 - Utilizaremos un bucle **For** que recorra todos los n elementos
- **Visual Basic**
 - De 1 a n ó de 0 a $n-1$
 - De $ini = \text{Lbound}(v)$ a $\text{Lbound}(v) + n - 1$

5.1 Sumar los números positivos



5.2 Sumar los números positivos

```
Function SumaPosi (ByRef v() As Double, ByVal n As Integer) As Double
```

```
  Dim i As String
```

```
  Dim r As Double
```

```
  r = 0
```

```
  For i = 1 To n Step 1
```

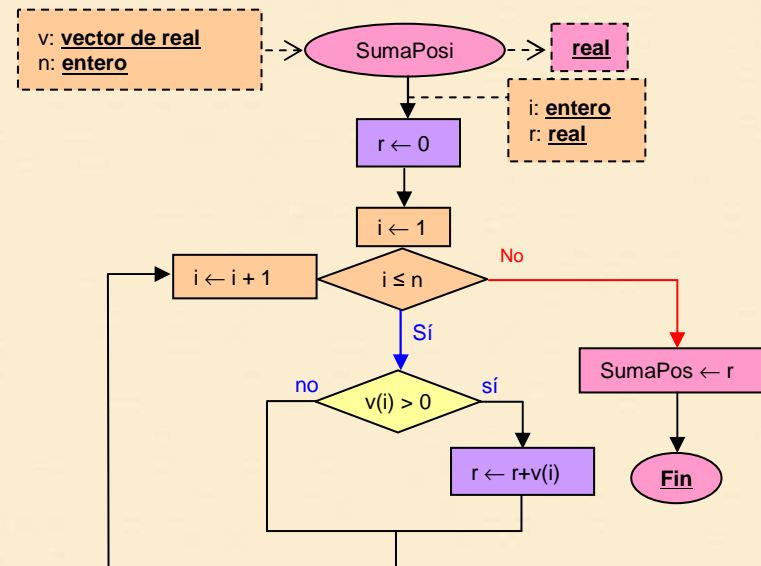
```
    If v(i) > 0 Then  
      r = r + v(i)
```

```
    End If
```

```
  Next i
```

```
  SumaPosi = r
```

```
End Sub
```

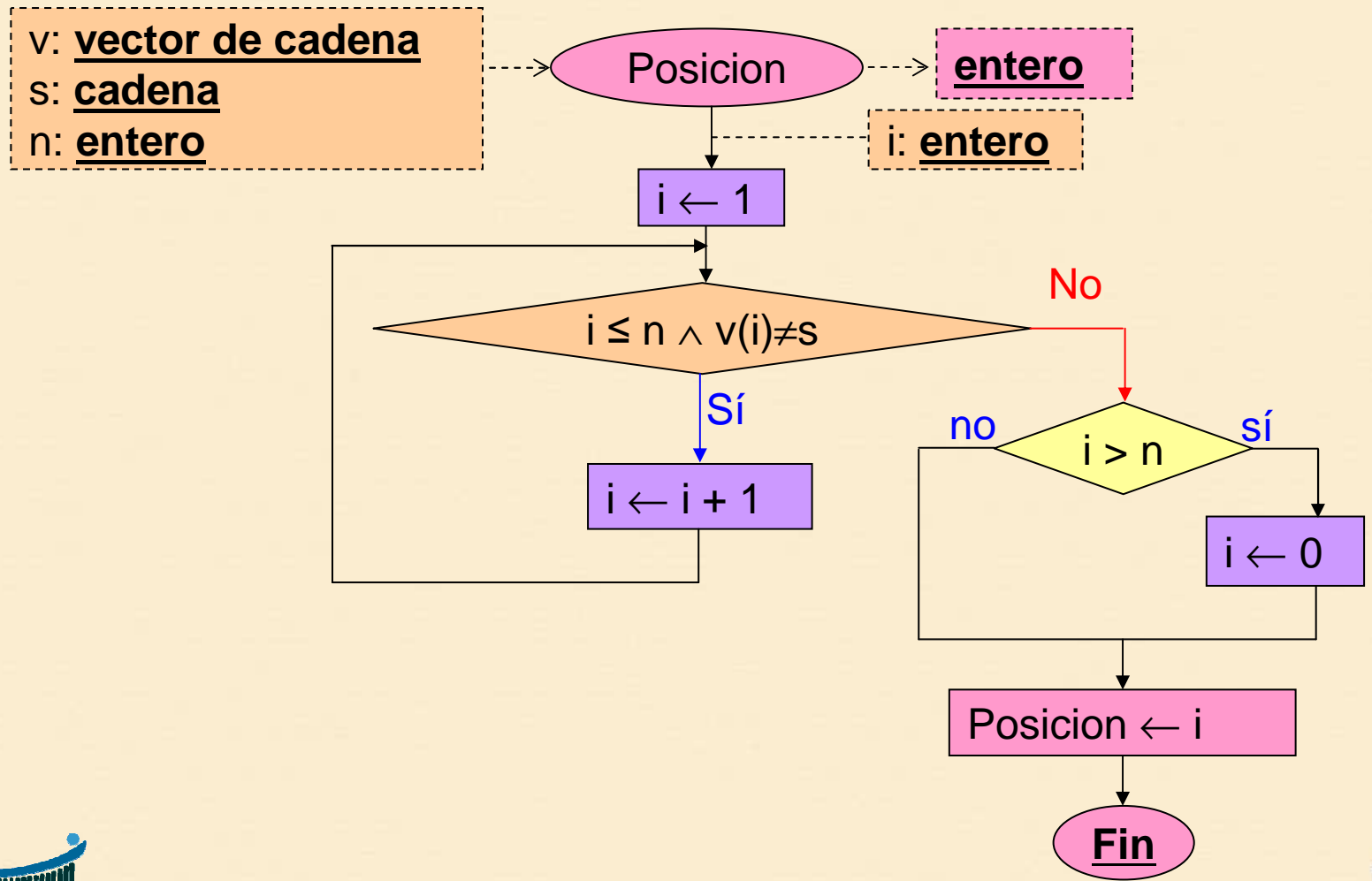


6. Recorrido de búsqueda

- **Ejemplos**
 - Buscar la posición de un número o una cadena en un vector
 - Buscar el primer elemento que cumple una propiedad
- **Modelo general de resolución**
 - Hay dos condiciones de parada:
 1. Hemos encontrado el elemento que buscábamos
 2. Hemos agotado todas las posibilidades
 - Convertiremos el bucle For en un bucle While que recorra todos los elementos o pare si encuentra lo que buscaba
- **Visual Basic**
 - De 1 a n And Not Cond
 - De ini=LBound(v) a LBound(v)+ n-1 And Not Cond



6.1 Buscar un nombre en lista



6.2 Buscar un nombre en una lista

```
Function Posicion (ByRef v() As String, ByVal s As String, ByVal n As Integer) As Integer
```

```
Dim i As String
```

```
i = 1
```

```
While i <= n And v(i) <> s
```

```
    i = i + 1
```

```
Wend
```

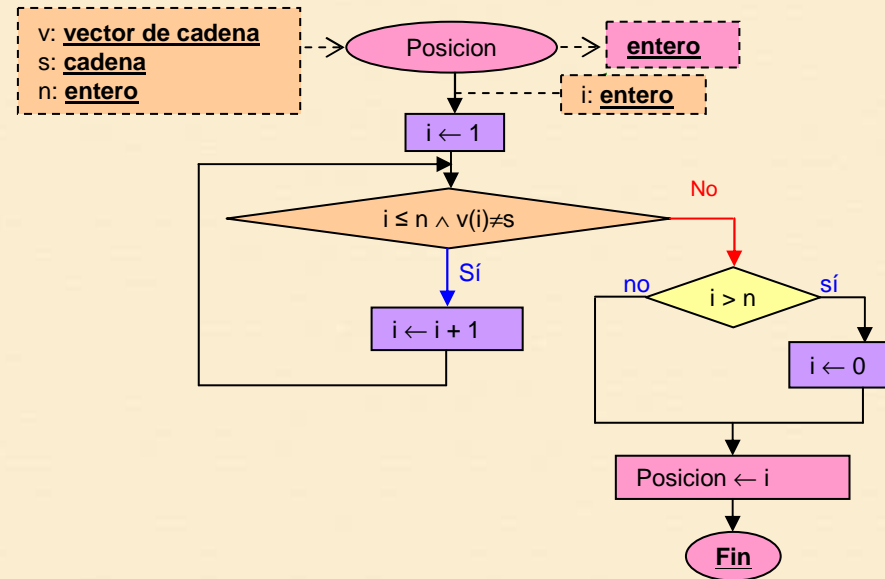
```
If i > n Then
```

```
    i = 0
```

```
End If
```

```
Posicion = i
```

```
End Sub
```



7. Inicialización de un vector

- **Enunciado**
 - Inicializar un vector a valores aleatorios entre 0 y 10
- **Funciones nuevas**
 - **Rnd**
 - Devuelve un número aleatorio de distribución uniforme en $[0, 1)$
 - Parámetros: no tiene
 - **Randomize**
 - Inicializa la semilla de números aleatorios
 - Parámetros: no tiene
- **Descripción**
 - Para poder hacer pruebas vamos a utilizar un subprograma que nos rellene los vectores a valores aleatorios. Así no tenemos que andar nosotros introduciéndolos. La distribución de **Rnd** nos permitirá verificar la corrección de los resultados.



7.1 Inicialización

```
Sub Inicializa (ByRef v() As Double)  
    Dim i As String  
    For i = LBound(v) To UBound(v) Step 1  
        v(i) = Rnd * 10  
    Next i  
End Sub
```


8. Programa principal

- **Enunciado**
 - Inicializar un vector a valores aleatorios entre 0 y 10, calcula la media y muestra el resultado en pantalla
- **Descripción**
 - El programa principal será el que declare el vector. El resto de los subprogramas lo utilizarán, pasándose los parámetros por referencia
 - Podemos reutilizar el subprograma de inicialización del apartado anterior

8.1 Programa principal

```
Sub CalculaMedia_Click ()  
    Dim notas (1 To 200) As Double  
    Dim m As Double  
    Inicializa (notas)  
    m = Media (notas, 200)  
    MsgBox CStr (m)  
End Sub
```