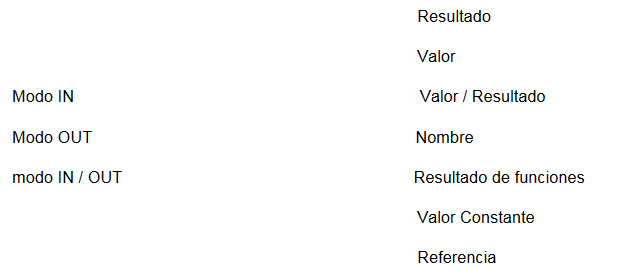
|  |  |
| --- | --- |
| \* n | rga |
|  | Pto retorno |
|  | EE |
|  | ED |
|  | VR |

# 1.

* Parametro
  + Es una forma de compartir datos entre diferentes unidades
* Parametro real
  + Es un valor u otra entidad que se pasa a un procedimiento o función
* Parametro formal
  + El parámetro es una expresión (u otra construcción) que produce un argumento. Es un identificador a través del cual un procedimiento puede acceder a un argumento
* Ligadura posicional
  + Los parámetros se corresponden con la posición que ocupan en la lista
* Ligadura por palabra clave o nombre
  + Los parámetros se corresponden con el nombre por lo tanto pueden estar colocados indistintamente en la lista

# 2.



# 3.

|  |  |
| --- | --- |
| Tipo de pasaje de parámetros | Lenguaje |
| IN / OUT (Referencia si no primitivo; valor resultado si primitivo), IN (valor constante), OUT (Resultado) | ADA |
| IN (valor), IN/OUT (referencia) | C |
| IN (valor) | Ruby |
| IN (valor) | JAVA |
| IN(valor) | Python |

b)En ADA los parámetros formales por valor no se pueden modificar puesto que son tratados como constantes, mientras que en pascal si se pueden modificar aunque estos cambios no se vean reflejados en el parámetro real

c) IN / OUT (Referencia si no primitivo; valor resultado si primitivo), IN (valor constante), OUT (Resultado)

# 4

## b.

### Cadena estatica

#### ii) valor

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = ~~2~~ 6 |
|  | i = 1 |
|  | J = 3 |
|  | VR |

Imprime: 1 3 6

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = 5 |
|  | VR |

Imprime: 1 3 5

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ~~1~~ 5 |
|  | y = ~~3~~ 5 |
|  | VR |

Imprime: 5 5 1 3 6

#### iii) ​Valor Resultado– Estatica

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = ~~2~~ 6 |
|  | i = ~~1~~ 5 |
|  | J = ~~3~~ 5 |
|  | VR |

Imprime: 5 5 6

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = 5 |
|  | VR |

imprime: 5 5 5

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ~~1~~ 5 |
|  | y = ~~3~~ 5 |
|  | VR |

Imprime: 5 5 1 3 6

#### iv) ​Nombre

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = ~~2~~ 6 |
|  | i = ~~1~~ 5 |
|  | j = ~~3~~ 5 |
|  | VR |

Imprime: 5 5 6

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = 5 |
|  | VR |

imprime: 5 5 5

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ^i-Dos |
|  | y = ^j-Dos |
|  | VR |

Imprime: 5 5 5 5 6

#### v) Resultado

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = 1 |
|  | j = 5 |
|  | VR |

Imprime:

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = 5 |
|  | VR |

imprime:

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = |
|  | y = |
|  | VR |

Imprime:

### Cadena dinámica

#### i) Referencia

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = ~~1~~ 5 |
|  | j = ~~5~~ 8 |
|  | VR |

Imprime: 5 8 2

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = ~~5~~ 9 |
|  | VR |

Imprime: 5 8 9

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = (Ref i-\*1) |
|  | y = (Ref j-\*1) |
|  | VR |

Imprime: 5 8 5 8 9

#### ii) Valor

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = 1 |
|  | j = 3 |
|  | VR |

Imprime: 1 3 2

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = ~~5~~ 9 |
|  | VR |

Imprime:1 3 9

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ~~1~~ 5 |
|  | y = ~~3~~ 8 |
|  | VR |

Imprime: 5 8 1 3 9

#### iii) Valor Resultado

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = ~~1~~ 5 |
|  | j = ~~3~~ 8 |
|  | VR |

Imprime: 5 8 2

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = ~~5~~ 9 |
|  | VR |

Imprime:5 8 9

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ~~1~~ 5 |
|  | y = ~~3~~ 8 |
|  | VR |

Imprime: 5 8 1 3 9

#### iv) Nombre

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = ~~1~~ 5 |
|  | j = ~~3~~ 8 |
|  | VR |

Imprime: 5 8 2

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = ~~5~~ 9 |
|  | VR |

Imprime: 5 8 5

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = ^i-Dos |
|  | y = ^j-Dos |
|  | VR |

Imprime: 5 8 5 8 9

#### iv) Resultado

|  |  |
| --- | --- |
| \* 1 | Rga Main |
|  | Pto retorno idk |
|  | m = 2 |
|  | i = 1 |
|  | j = 3 |
|  | VR |

Imprime:

|  |  |
| --- | --- |
| \* 2 | Rga Dos |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | M = 5 |
|  | VR |

Imprime: 5 8 5

|  |  |
| --- | --- |
| \* 3 | Rga Recibe |
|  | Pto retorno \*2 |
|  | EE \*1 |
|  | ED \*2 |
|  | x = |
|  | y = |
|  | VR |

Da error al intentar usar las variables

Imprime:

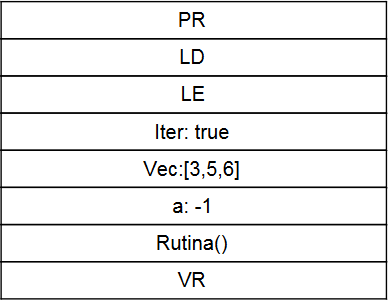
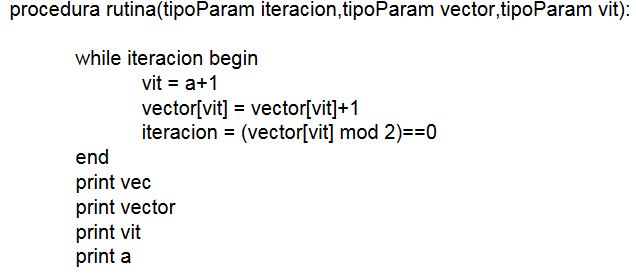
## c.

sí, en los tipos “Resultado” en la primera y segunda línea del proceso “Recibe” utiliza las variables sin inicializarlas

## d.

prácticamente, solo cambia que se modifica la “m” del proceso “Dos” en lugar de la “m” del proceso “Main”

# 5



rutina(iter,vec,a)

## a.

(4,6,7), (4,6,7), 2, 2 -> **Referencia**

## b.

(3,5,6),(4,6,7), 2, 2

## c.

(3,5,6),(5,5,6), 0, -1 -> **Valor**

valor

|  |  |
| --- | --- |
| \* 2 | Rga rutina |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | Iteración = ~~t~~ ~~t~~ f |
|  | Vector = [~~3~~ ~~4~~ 5, 5, 6] |
|  | Vit = ~~-1~~ ~~0~~ 0 |
|  | VR |

Imprime: [3,5,6] [5,5,6] 0 -1

Referencia

|  |  |
| --- | --- |
| \* 1 | Rga main |
|  | Pto retorno \*1 |
|  | EE |
|  | ED |
|  | Iter= ~~t~~ ~~t~~ ~~t~~ f |
|  | Vec = [~~3~~ 4, ~~5~~ 6, ~~6~~ 7] |
|  | a= ~~-1~~ ~~0~~ ~~1~~ 2 |
|  | Rutina() |
|  | VR |

|  |  |
| --- | --- |
| \* 2 | Rga rutina |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | Iteración = Ref(iter-Main) |
|  | Vector = Ref(Vec-Main) |
|  | Vit = Ref(a-Main) |

Imprime: [4, 6, 7] [4, 6, 7] 2 2

Nombre

|  |  |
| --- | --- |
| \* 1 | Rga main |
|  | Pto retorno \*1 |
|  | EE |
|  | ED |
|  | Iter= ~~t t t~~ f |
|  | Vec = [~~3~~ 4, ~~5~~ 6, ~~6~~ 7] |
|  | a= ~~-1~~ ~~0 1~~ 2 |
|  | Rutina() |
|  | VR |

|  |  |
| --- | --- |
| \* 2 | Rga rutina |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | Iteración = ^iter-Main |
|  | Vector = ^Vec-Main |
|  | Vit = ^a-Main |

Imprime: [4, 6, 7] [4, 6, 7] 2 2

Valor resultado

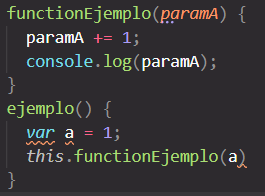
|  |  |
| --- | --- |
| \* 1 | Rga main |
|  | Pto retorno \*1 |
|  | EE |
|  | ED |
|  | Iter= t |
|  | Vec = [3, 5, 6] |
|  | a= -1 |
|  | Rutina() |
|  | VR |

|  |  |
| --- | --- |
| \* 2 | Rga rutina |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | Iteración = t t f |
|  | Vector = [3 4 5, 5, 6] |
|  | Vit = -1 0 0 |

Imprime:

# 6

## Valor entero

al ser parametros por nombre

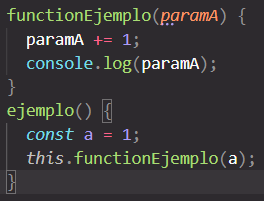
cuando se hace la instrucción “paramA+=1”

lo que en realidad se ejecuta es “a+=1”

lo mismo, al imprimir se reemplaza “paramA” por “a”

quedando “console.log(a)” e imprimiendo “2”

## Una constante

al ser parametros por nombre

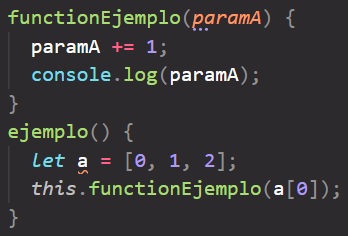
cuando se hace la instrucción “paramA+=1”

lo que en realidad se ejecuta es “a+=1” pero como “a” es una constante aca daría un error.

De no estar esa línea el “console.log(paramA)” quedaría

“console.log(a)” que se imprimiría bien, dando “1” como resultado

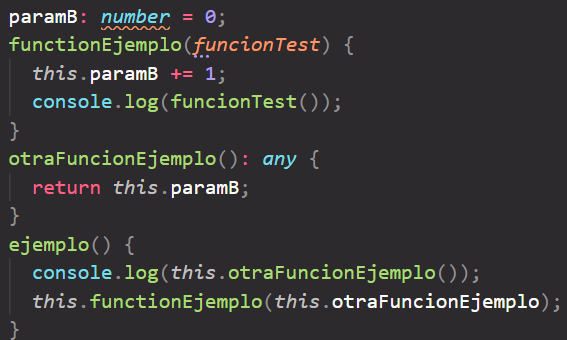
## Un elemento de un arreglo

en este ejemplo al ejecutar “paramA+=1” se ejecuta realmente “a[0]+=1” quedando el vector: “[1,1,2]”

luego el imprimir quedaría

“console.log(a[0])”, imprimiendo: 1

## una expresión

en este ejemplo al pasarse una función, al ejecutarse la línea “console.log(funcionTest())” realmente se ejecuta

“console.log(this.otraFuncionEjemplo());”

Imprimiendo 1

Se puede ver que en el primer console.log() se imprimirá 0 puesto que aun no se incremento la variable global

# 7.

## Cadena estatica

|  |  |
| --- | --- |
| \* n | rga |
|  | Pto retorno |
|  | EE |
|  | ED |
|  | VR |

|  |  |
| --- | --- |
| \* 1 | Rga Uno |
|  | Pto retorno |
|  | y = ~~(1 .. 6)~~ ~~(1 .. 5)~~ ~~1~~ 2 |
|  | Z := ~~2~~ ~~3~~ 3 |
|  | R1 = (2,2,~~2~~ 3,~~2~~ 4,2,2) |
|  | R2 = (1,~~1~~ ~~2~~ ~~3~~ 4,~~1~~ ~~3~~ 5,1,1) |
|  | Procedure Dos |
|  | VR |

Llama a dos con (r1( y + r2( y )), r2( z ), y, z)

r1( y + r2( y )) = 2 + 1 = 2 = r1(3) = 2

r2( z ) = 1

y = 1

z = 2

imprime: (2,2,3,4,2,2)

imprime: (1, 4, 5,1,1)

|  |  |
| --- | --- |
| \* 2 | Rga Dos-Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | X = ^ r1( y + r2( y )) |
|  | T = ^ r2( z ) |
|  | Io = ref(y) |
|  | Y = ~~2~~ 3 (vr) |
|  | VR |

x := x + 1; -> r1( y + r2( y )) = r1( y + r2( y )) + 1

r1(2) = 2 + 1 = 3

t := t + 1; -> r2( z ) = r2( z ) + 1 =r2(2)=r2(2)+1

x := x + 2; -> r1( y + r2( y )) = r1( y + r2( y )) + 2

r1( y + r2( y )) = 2 + r2(2) = 2 + 2

|  |  |
| --- | --- |
| \* 3 | Rga Dos - Dos |
|  | Pto retorno \*2 |
|  | EE \*2 |
|  | ED \*2 |
|  | T1 = ^t (^r2(z)) |
|  | VR |

t1 := t1 + 1; -> r2(z)++

t := t + 1; -> r2(z)++

t1 := t1 + 2; -> r2(z)+=2

t := t + 2; -> r2( z )+=2

|  |  |
| --- | --- |
| \* 4 | Rga Tres |
|  | Pto retorno \*3 |
|  | EE \*3 |
|  | ED \*3 |
|  | VR |

## Cadena dinamica

|  |  |
| --- | --- |
| \* 1 | Rga Uno |
|  | Pto retorno |
|  | Y = 1..6 1..5 1 2 |
|  | Z = 2 3 |
|  | R1 = (2,2,~~2~~ 3,~~2~~ 4,2,2) |
|  | R2 = (1,~~1~~ ~~2~~ ~~3~~ 4,~~1 3~~ 5,1,1) |
|  | Procedure Dos |
|  | VR |

Imprime: (2,2,3,4,2,2)

Imprime: (1,4, 5,1,1)

|  |  |
| --- | --- |
| \* 2 | Rga Dos-Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | X = ^ r1( y + r2( y )) |
|  | T = ^ r2( z ) |
|  | Io = ref(y) |
|  | Y = 2 3 (vr) |
|  | VR |

|  |  |
| --- | --- |
| \* 3 | Rga Dos-Dos |
|  | Pto retorno \*2 |
|  | EE \*2 |
|  | ED \*2 |
|  | T1 = ^t |
|  | VR |

|  |  |
| --- | --- |
| \* 4 | Rga Tres |
|  | Pto retorno\*3 |
|  | EE \*3 |
|  | ED \*3 |
|  | VR |

8.

Son modelos de ejecución que se aplican cuando lo que se pasa como parámetro es una subrutina. Todo aplica a resolver las referencias no locales dentro de esa rutina que se ejecuta.

Cuando aparece una referencia a algo no local y se sigue el esquema shallow, es muy similar a seguir la cadena dinámica. Es decir, la referencia se busca en quien invocó, y así siguiendo la cadena hasta que se encuentra la variable.

Si el esquema es deep, es muy similar a seguir la cadena estática, es decir, se busca en quien la contiene.

Shadow = cadena dinamica

Deep = cadena estatica

## deep

|  |  |
| --- | --- |
| \* 1 | Rga A |
|  | Pto retorno |
|  | X = 0 |
|  | y = “a” |
|  | procedure B |
|  | procedure C |
|  | function D |
|  | VR |

# 9.

Asumimos que da igual si es estatico o dinamico

## nombre

|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [~~1~~ 3,~~1~~ 2,~~1~~ 0,1,1] |
|  | x = ~~3~~ ~~0~~ ~~1~~ 2 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 3 2 0 1 1

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | m = ^a(x) |
|  |  |
|  | VR |

### B - nombre

|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [~~1~~ 3,~~1~~ 2,~~1~~ 0,1,1] |
|  | x = 3 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 3 2 0 1 1

No se modifica la x de main

Opera sobre a(x)

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | m = ^a(x) |
|  | x=~~0 1~~ 2 |
|  | VR |

## Referencia

|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [1,1,~~1 3 2~~ 3,1,1] |
|  | x = ~~3 0 1~~ 2 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 1 1 3 1 1

Opera directamente sobre a(3)

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  |  |
|  | m = ref(a(3)) |
|  | VR |

### Referencia - B

|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [1,1,~~1 3 2~~ 3,1,1] |
|  | x = 3 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 1 1 3 1 1

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | X= ~~0 1~~ 2 |
|  | m = ref(a(3)) |
|  | VR |

## Valor resultado

|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [1,~~1~~ 4,~~1~~ 0,1,1] |
|  | x = ~~3 0 1~~ 2 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 1 4 0 1 1

Opera sobre una variable propia y luego refleja ese valor en a(x)

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  |  |
|  | m = ~~1~~ 3 4 |
|  | VR |

### Valor resultado - B

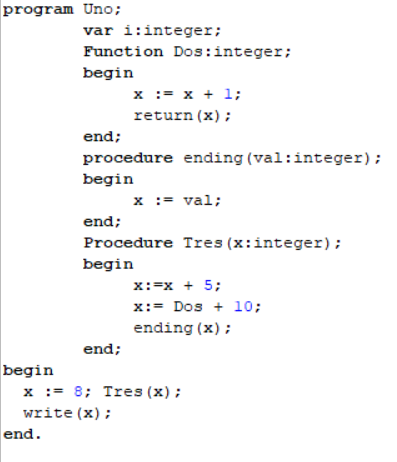
|  |  |
| --- | --- |
| \* 1 | rga Main |
|  | Pto retorno |
|  | a= [1,1 4,1 0,1,1] |
|  | x = 3 |
|  | I = (1..5) |
|  | Procedure Uno |
|  | VR |

Imprime: 1 4 0 1 1

No cambia nada

|  |  |
| --- | --- |
| \* 2 | rga Uno |
|  | Pto retorno \*1 |
|  | EE \*1 |
|  | ED \*1 |
|  | X = 0 1 2 |
|  | m = 1 3 4 |
|  | VR |

# 10.

la solución fue implementar una nueva función que escriba sobre la variable global el valor que quedo en la x local a Tres al final de la ejecucion

b) por favor matenme