



aprenderaprogramar.com

# Problemas con resolución intuitiva. Ordenar una serie de números. Ejercicio resuelto en pseudocódigo. (CU00261A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha última actualización: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 60 del Curso Bases de la programación Nivel II

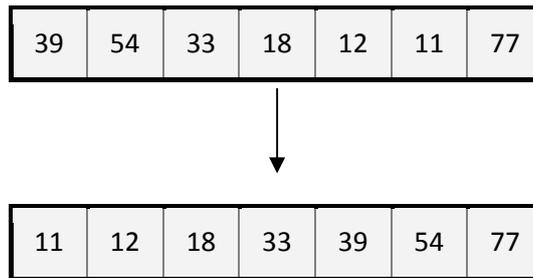
24

## PROBLEMAS CON RESOLUCIÓN INTUIDA. ORDENAR UNA SERIE DE NÚMEROS

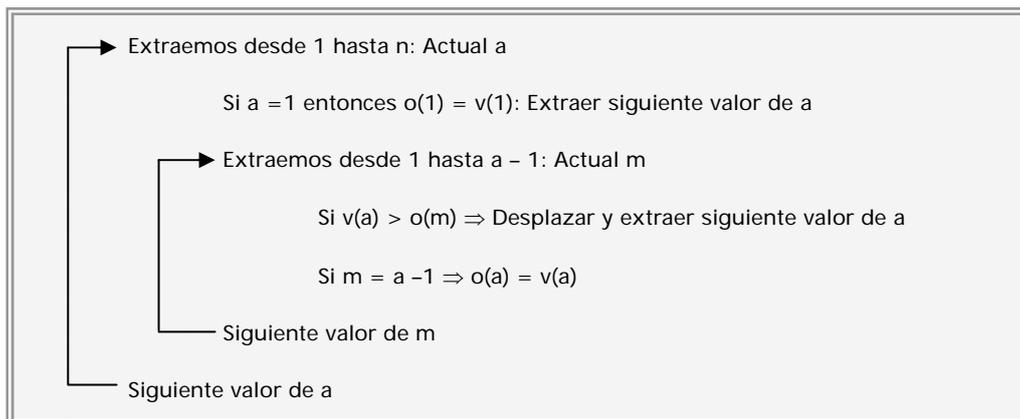
### EJERCICIO

Crear el pseudocódigo y el diagrama de flujo para un programa que ordene una serie de  $n$  números.

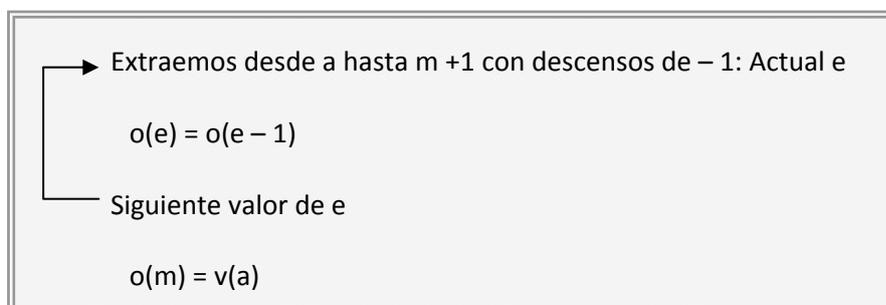
Supongamos lo siguiente: tenemos una serie de números y queremos ordenarla (de menor a mayor).



El procedimiento de resolución a utilizar debe estar basado en este planteamiento esquemático:



Procedimiento para desplazar:



Supondremos que los datos se leen desde un archivo que contiene  $n$  (número de valores) y los valores como  $dato(1), dato(2), \dots, dato(n)$ .

## SOLUCIÓN

### PROGRAMA ORDEN01 [Pseudocódigo aprenderaprogramar.com]

#### Variables

Enteras: E, n

Reales: dato()

#### 1. Inicio

#### 2. Mientras E <> 3 Hacer

2.1 Mostrar "1. Ver datos desordenados.

2. Ordenar.

3. Salir."

2.2 Pedir E

#### 2.3 Si E = 1 Entonces

Llamar Extraer

**FinSi**

#### 2.4 Si E = 2 Entonces

**Si n <> 0 Entonces**

Llamar Ordenar(dato)

**SiNo**

Mostrar "No hay datos que ordenar"

**FinSi**

**FinSi**

**Repetir**

#### 3. Fin

---

#### Módulo Extraer

#### Variables

Enteras: i

1. Leer n

2. Redimensionar dato(n)

#### 3. Desde i = 1 hasta n Hacer

Leer dato(i)

Mostrar "dato(", i, ")=", dato(i)

**Siguiente i**

**FinMódulo**

**Módulo Ordenar(v(): Reales)**

**Variables**

Enteras: a, m

Reales: o()

1. Redimensionar o(n)

2. Desde a = 1 hasta n Hacer

2.1 Si a = 1 Entonces

o(1) = v(1)

**FinSi**

2.2 Desde m = 1 hasta a – 1 Hacer

2.2.1 Si v(a) > o(m) Entonces

Llamar Desplazar

SalirDesde

**FinSi**

2.2.2 Si m = a – 1 Entonces

o(a) = v(a)

**FinSi**

**Siguiente m**

**Siguiente a**

3. Mostrar “Valores ordenados”

4. Desde a = 1 hasta n Hacer

Mostrar o(a)

**Siguiente**

**FinMódulo**

**Módulo Desplazar**

**Variables**

Enteras: j

1. Desde j = a hasta m + 1 Paso = – 1 Hacer

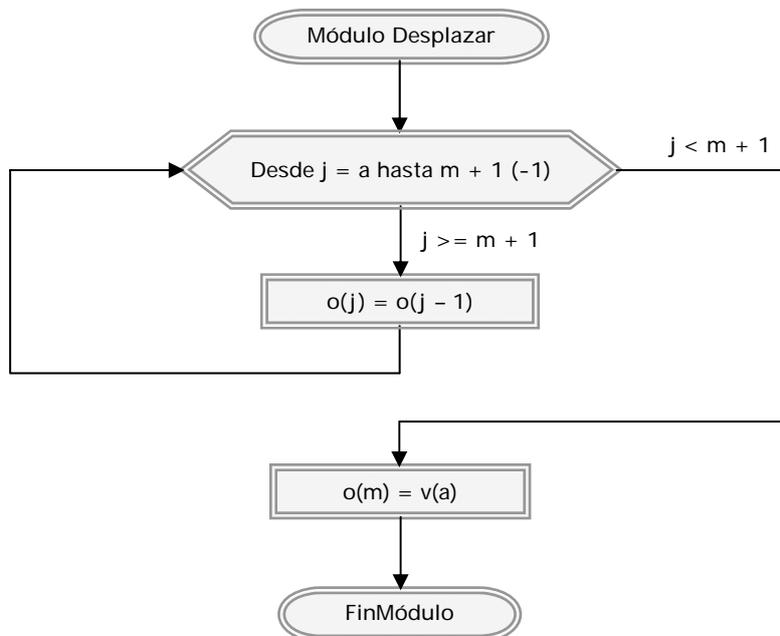
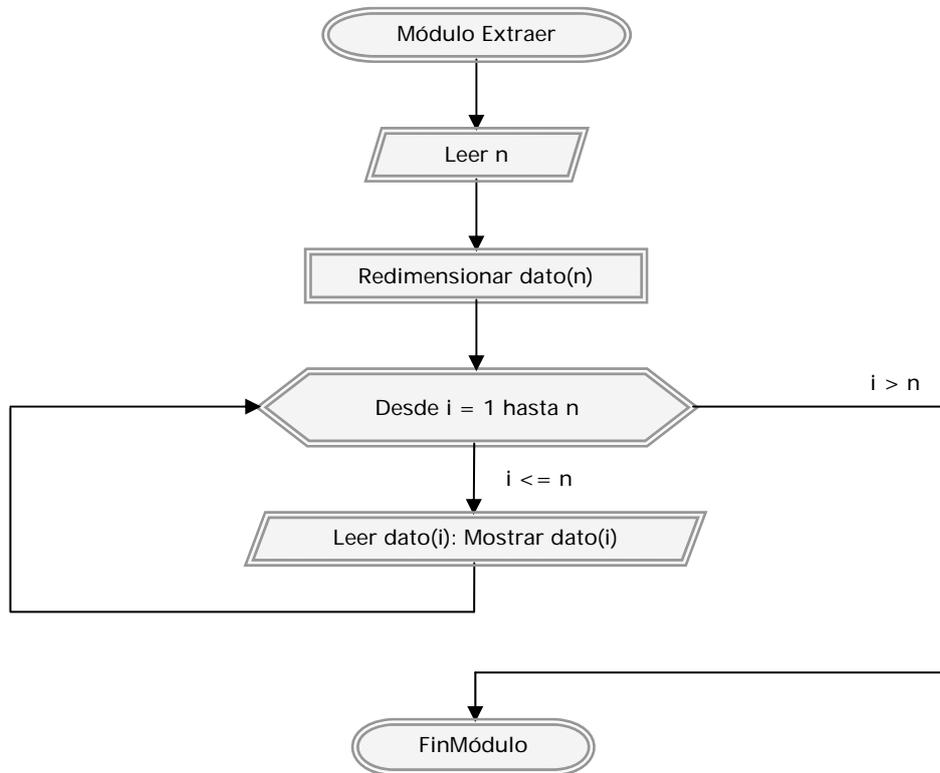
o(j) = o(j – 1)

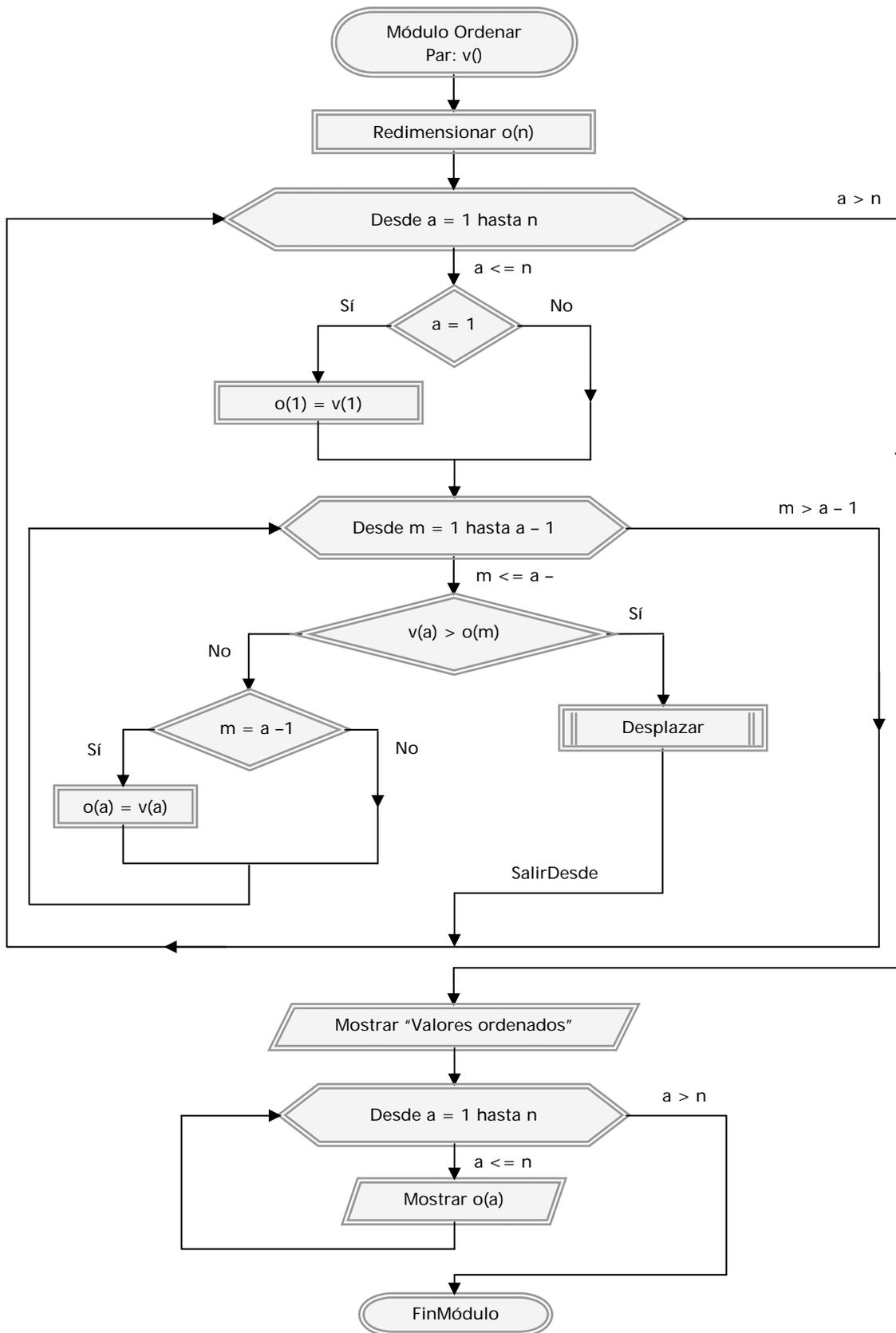
**Siguiente j**

2. o(m) = v(a)

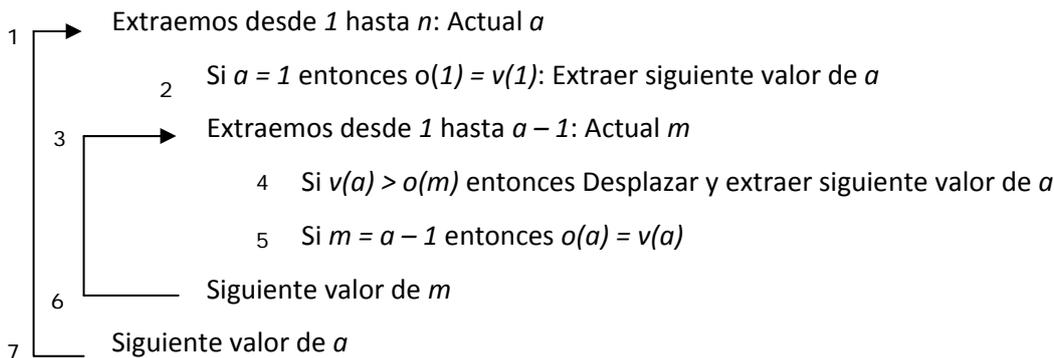
**FinMódulo**







**Comentarios:** Hay una curiosidad en la traslación de lo planteado como procedimiento esquemático a seguir y lo que finalmente hemos plasmado en pseudocódigo en este algoritmo. En el planteamiento habíamos escrito:



Teníamos planteados dos puntos con “Extraer siguiente valor de  $a$ ”, en las líneas 2 y 4. La correspondencia con el algoritmo es:

Línea 2 planteamiento → Sin correspondencia

Línea 4 planteamiento → Línea 2.2.1 Módulo *Ordenar(SalirDesde)*

¿Por qué lo previsto en la línea 2 del planteamiento no se corresponde con una instrucción de control directo de flujo? Aparte de que podríamos rediseñar el código cambiando a la línea 2 del planteamiento de sitio, si quisiéramos respetar su textualidad nos bastaría introducir una instrucción *IrA* para saltar a la línea 7, ya que un *SalirDesde* no cumpliría nuestro objetivo, que es pasar al siguiente valor de  $a$  sin entrar al bucle  $m$ . ¿Por qué no se ha hecho? Pues simplemente porque no era necesario, y ya hemos dicho que conviene evitar a toda costa las instrucciones de control directo de flujos y en particular la instrucción *IrA*. ¿Por qué no es necesaria? Si analizamos el algoritmo, comprobamos que para evitar la entrada en el bucle  $m$  cuando  $a$  vale 1 no hace falta dar un salto, porque la propia definición del bucle  $m$  impide esa posibilidad. Es decir, cuando  $a$  vale 1 no se llega a entrar en el bucle  $m$  de forma natural.

La situación es:

$$a = 1$$

$$\text{Desde } m = 1 \text{ hasta } a - 1$$


---


$$a = 1$$

$$\text{Desde } m = 1 \text{ hasta } 0$$

Al cumplirse la condición de salida  $m > a - 1$  no hay entrada en el bucle (recordar la forma de trabajar de la Instrucción *Desde ... Siguiente*) con lo cual es innecesaria una instrucción de salto en ese punto. Dicha instrucción podría ponerse, pero en este caso lo único que haría sería complicar la lectura y comprensión.

El *SalirDesde* que se ha empleado se ha considerado oportuno y por eso figura ahí, aunque podrían buscarse otras soluciones: cada cual que escoja la que estime más conveniente.

**Próxima entrega: CU00262A**

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)