

## Bloque 2. Tema 1

### Nociones de algoritmos y acción

#### Objetivos:

- Conocer los conceptos de entorno, acción y procesador
- Familiarizarse con la noción de algoritmo
- Comprender en qué consiste el análisis descendente

## Tema 1.

### Nociones de algoritmo y acción

1. Introducción.
2. Nociones de entorno, acción y procesador.
3. Acciones primitivas; descomposición de una acción.
4. Noción de análisis descendente.
5. Noción de algoritmo.

#### Bibliografía:

- "Introducción a la programación. ". J. Biondi y G. Clavel. Tomo 1: Algorítmica y lenguajes. Manuales de Informática Masson Edit. Masson. Barcelona.

### 1.1. Introducción

**¿Cómo se puede resolver un problema de forma automática utilizando un ordenador/lenguaje de programación?**

Debo preguntarme:

- ¿Sé resolver el problema?  
Es imprescindible.
- ¿Necesito inmediatamente el ordenador?  
No.
- ¿Es necesario algún procesador?  
Sí, pero puede ser virtual.
- ¿Cómo tendríamos que darle las órdenes al procesador?  
Paso a paso.

*Este procedimiento suele denominarse **protocolo** para resolver un problema (en general) o **algoritmo** (en el caso particular de la programación).*

### Algunos ejemplos

En cualquier situación será necesario enunciar un algoritmo o protocolo si deseamos que un procesador realice una acción que no sea elemental para él/ella:

- Poner en marcha una caldera de gas,
- preparar un compuesto químico,
- cocinar una tortilla de seis huevos,
- calcular la media de tres números,
- devolver el cambio de una cantidad,
- calcular la solución de un conjunto de ecuaciones, . . .

## 1.2. Nociones de entorno, acción y procesador

- **Entorno:** Conjunto de utensilios u objetos necesarios o disponibles para resolver un problema.  
*(Contexto en el que resolvemos el problema)*
- **Acción:** Suceso que modifica el entorno.  
*(Tarea/s que podemos realizar para solucionarlo)*
- **Procesador:** Entidad capaz de entender un enunciado y ejecutar el trabajo indicado.  
*(Quien realiza el trabajo)*

### Ejemplo: ¿Cómo calcular la media de tres números con una calculadora?

1. **Entorno:** una calculadora.
2. **Acciones comprensibles por el procesador:** pulsar teclas/números de la calculadora
3. **Procesador:** cualquier persona adulta.

### Solución:

1. Pulsa [C] para borrar
2. Escribe primer número
3. Pulsa [+]
4. Escribe segundo número
5. Pulsa [+]
6. Escribe tercer número
7. [Pulsa [=]]
8. Pulsa [÷]
9. Escribe [3]
10. Pulsa [=]

### Ejemplo: ¿Cómo preparar una tortilla de seis huevos?

1. **Entorno:** una mesa, un fogón, una sartén, un plato, un tenedor, aceite, una cesta con huevos, un cubo de basura.
2. **Acciones comprensibles por el procesador:** coger un huevo, romper un huevo en un plato, batir huevos, echar aceite, poner en la sartén, retirar de la sartén, poner al fuego, retirar del fuego, tirar las cáscaras, encender el fuego;  
  
puede contestar a ¿está caliente?, ¿está hecha tortilla?
3. **Procesador:** cualquier persona adulta.

### Solución: ¿cómo hacer una tortilla de seis huevos?

1. romper seis huevos en un plato
2. batir los huevos
3. calentar aceite en la sartén
4. si el aceite está caliente, poner en la sartén [el contenido del plato]
5. si la tortilla está hecha, retirar la tortilla del fuego
6. Tirar las cáscaras

Tipos de acciones que puede entender un procesador:

- en función de su complejidad (primitivas o compuestas)
- según se tenga en cuenta el estado del entorno o no (condicionales o incondicionales)

### 1.3. Acciones primitivas: descomposición de una acción

- **Acción primitiva:** Acción para un procesador concreto cuyo enunciado es suficiente para que pueda ser ejecutada sin información adicional

Una acción que no sea primitiva (se denominará **compuesta**) debe descomponerse en acciones más sencillas

En el ejemplo de la tortilla:

- {romper seis huevos en el plato, calentar aceite en la sartén, retirar la tortilla del fuego, tirar las cáscaras} no son acciones primitivas para el procesador

### 1.3. Acciones primitivas: descomposición de una acción

- romper seis huevos en el plato
  - Hacer 6 veces:
    - coger un huevo;
    - romperlo en el plato
- calentar aceite en la sartén
  - echar aceite en la sartén
  - encender el fuego
  - poner la sartén en el fuego
- retirar la tortilla del fuego
  - retirar la sartén del fuego
  - retirar [contenido] de la sartén

### 1.4. Noción de análisis descendente

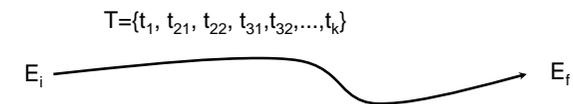
Dado un trabajo  $T$  asociado a la resolución de un enunciado, el análisis descendente busca una secuencia de acciones primitivas que lo realicen.

El trabajo  $T$  debe transformar el entorno inicial del problema  $E_i$ , asociado al enunciado, en el entorno final deseado,  $E_f$ , que se corresponda con la solución al problema.



El procedimiento de análisis descendente (o **análisis top-down**) consiste en descomponer cada tarea no elemental (el trabajo  $T$ ) en un conjunto de tareas de menor complejidad.

Este proceso debe repetirse con cada nueva tarea generada hasta que todas las tareas sean elementales.



Cada una de las tareas intermedias,  $t_m$ , tiene un entorno inicial,  $E_{im}$ , y un entorno final,  $E_{fm}$ , siendo  $E_{i1} = E_i$  y  $E_{fk} = E_f$

Además, para cada par de tareas consecutivas debe verificarse:  $E_{fm} = E_{i(m+1)}$

Solución al problema: ¿cómo cocinar una tortilla de seis huevos (cont.)?

Vamos a suponer que disponemos de la siguiente acción genérica para todo procesador:

*repetir*

*conjunto de acciones primitivas  
hasta que se cumpla una condición*

**Ejemplo de análisis descendente para la acción:  
romper seis huevos en un plato.**

{ cómo poner seis huevos en una mesa }

1. repetir

- a) coge huevo de la cesta
- b) ponlo en la mesa

hasta que haya 6 huevos

{ Romper los seis huevos }

2. repetir

- a) coge un huevo de la mesa,
- b) rómpelo, echando el contenido en el plato,
- c) tira las cáscaras a la basura

hasta que no queden huevos en la mesa

## 1.5. Noción de algoritmo

**Algoritmo:** Secuencia de acciones primitivas a realizar por un procesador para resolver una tarea concreta.

**Programación:** La resolución de problemas usando los lenguajes de programación sobre un ordenador.

- Conlleva codificar o traducir un algoritmo que resuelve un problema en algún lenguaje de programación

**Ejemplos:**

- Propón un algoritmo que sea capaz de devolver el cambio de una compra en euros.
- Realiza un algoritmo para la agrupación de un documento de 10 páginas que ha sido enviado a un servicio de fotocopiado.

### Ejercicio: Devolver el cambio de una cantidad

**Entorno:** una caja con monedas de { 1, 2, 5, 10, 20 y 50 } céntimos y monedas/billetes de { 1, 2, 5, 10, 20, 50 } euros

**Acciones:**

- coger monedas/billetes;
- coger monedas de un cajón;
- ordenar las monedas de mayor a menor;
- elegir una moneda [de mayor/menor valor];
- calcular la diferencia entre dos cantidades;
- Comprobar diferencia entre dos valores;
- Comprobar valor [de una moneda o del cambio]

**Procesador:** cualquier persona adulta

- **Realiza un algoritmo para la agrupación de un documento de 10 páginas que ha sido enviado a un servicio de fotocopiado.**

Las condiciones del trabajo son:

- Hay 10 paquetes. Cada uno tiene aproximadamente 100 copias de cada hoja.
- No están ordenados por hojas.
- Procesador = Individuo.
- **Entorno** = 10 paquetes de hojas numeradas, escritorio donde están apiladas, una grapadora, 10 mesas numeradas, donde cada mesa puede recibir las copias asociadas a la hoja que tiene el mismo número que la mesa.
- **Funciones/acciones que entiende el procesador:** coger y poner, ir de una mesa a otra, ir a una mesa de un número concreto, reconocer presencia o ausencia de números en una hoja, reconocer presencia o ausencia de hojas/paquetes de hojas en una mesa, grapar.