

## **TEMA 1. INTRODUCCIÓN A LA ARQUITECTURA VON NEUMANN**

- 1.1. [Sistema computador](#)
- 1.2. [Principio de funcionamiento de un computador](#)
  - 1.2.1. [Descripción de elementos](#)
  - 1.2.2. [Desarrollo de una instrucción de procesamiento](#)
- 1.3. [Niveles de estudio del computador](#)

*Este tema introduce la estructura básica de la **máquina de programa almacenado** (máquina de Von Neumann) y su principio de funcionamiento. Se describen sus componentes básicos y fundamentales: Unidad Central de Proceso (Unidad de Control y Unidad Operativa), Memoria Central (o Memoria Principal) y Unidades de Entrada/Salida. Y, se describe el desarrollo de una instrucción de procesamiento, diferenciando las distintas fases que la componen.*

*Dada la complejidad de un computador, se aborda desde distintas ópticas los niveles de detalle a los que pueden ser estudiados. Se describen los niveles de estudio según Bell y Newell (niveles estructurales), según Levy (niveles de interpretación) y según Blaauw (niveles conceptuales).*



## 1.1. SISTEMA COMPUTADOR

Con una visión muy general, un *COMPUTADOR* u *ORDENADOR* se compone de una *Memoria Central* o *Memoria Principal*, que contiene programas y datos, de una *Unidad Central de Proceso*, que ejecuta el programa, y de *Unidades de Entrada/Salida* (unidades de E/S) para los intercambios de información con el exterior. A veces se le denomina *SISTEMA INFORMÁTICO* al conjunto formado por el computador y los elementos que estén conectados a él.

Para tener una primera aproximación al sistema computador, se describen, de forma muy resumida, las partes principales y básicas del mismo.

### LA UNIDAD CENTRAL DE PROCESO Y LA MEMORIA CENTRAL

El *programa* es registrado en memoria antes de comenzar su ejecución. Esta memoria, gracias a la cual puede ejecutarse el programa, se llama **Memoria Central** o **Memoria Principal** y, en torno suyo se organiza el resto de las diferentes unidades de la máquina.

La Memoria Principal almacena dos clases de informaciones:

- a) *Instrucciones* del programa (o informaciones descriptoras del tratamiento a realizar), que la máquina deberá ejecutar.
- b) *Operandos* o *datos* (o informaciones a tratar), con los cuales efectuará la máquina los tratamientos dictados por las instrucciones.

Las dos clases de información tienen su correspondencia física en las siguientes unidades peculiares de la máquina:

- a) La *Unidad de Control*, también llamada Unidad de Instrucciones o Unidad de Gobierno, para las informaciones descriptoras.
- b) La *Unidad Operativa* (Unidad Aritmética y Lógica), o Unidad de Proceso, para las informaciones a tratar.

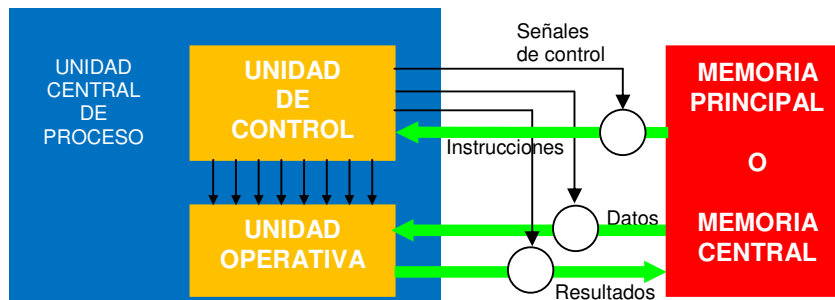
Esquemáticamente, las principales funciones de la *Unidad de Control* son:

- Leer de la Memoria Principal la nueva instrucción a ejecutar.
- Analizar dicha instrucción y establecer las conexiones eléctricas correspondientes dentro de la Unidad Operativa.
- Leer de la Memoria Principal los datos implicados por la instrucción.
- Desencadenar el tratamiento de dichos datos en la Unidad Operativa.

- Eventualmente, almacenar el resultado en la Memoria Principal.

La *Unidad Operativa* opera con los datos que recibe siguiendo órdenes de la *Unidad de Control*.

El conjunto formado por la Unidad de Control y la Unidad Operativa forman un todo en prácticamente todos los sistemas computadores. A este conjunto se le llama *Unidad Central*, *Unidad Central de Proceso* o *Procesador Central*. Para algunos autores, la Unidad Central de Proceso comprende también la Memoria Principal.



**Figura 1.1.** Organización general de la Unidad Central de Proceso y la Memoria Principal.

### UNIDADES DE E/S Y UNIDADES PERIFÉRICAS

La máquina puede ejecutar un programa inicialmente registrado en la Memoria Principal, sobre datos registrados en la misma Memoria Principal y almacenar los resultados en dicha memoria a medida que se obtienen. Ahora es preciso dotarla de medios para comunicarse con el exterior: este es el papel reservado a las *Unidades Periféricas*.

Existen dos grandes clases de *Unidades Periféricas*:

- Las *Unidades de Comunicación* (impresora, pantalla, teclado, etc.) que permiten el diálogo con el exterior.
- Las *Unidades de Almacenamiento* (discos, cintas magnéticas, etc.), cuyas capacidades de almacenamiento son muy superiores a la de la Memoria Principal.

Las Unidades Periféricas se conectan a la Unidad Central de Proceso o directamente a la Memoria Principal, a través de unidades especializadas en la gestión de las transferencias de información. Estas unidades de intercambio de información con el exterior se llaman *Unidades de E/S*. La Unidad de Control se comunica con estas Unidades de E/S mediante instrucciones que pueden ser específicas de E/S o generales de transferencia, dependiendo del sistema computador.

En resumen, un computador u ordenador es un conjunto ensamblado de unidades diferentes, cuyo funcionamiento viene dictado por el programa registrado en la Memoria Central o Principal. La Unidad de Control gobierna la ejecución de las operaciones pedidas por el programa; si la operación es un cálculo, es la Unidad Operativa quien la realiza; si es una transferencia de información con el exterior, se opera realizando transferencias con las Unidades de E/S.

Hemos dicho que en la memoria se guardan de forma ordenada los datos y las instrucciones que han de ser ejecutadas por los restantes componentes del sistema para obtener los resultados deseados. Sin embargo, el computador ha de poseer la capacidad de romper la secuencia de las instrucciones y producir saltos en el programa de acuerdo con los resultados que se van obteniendo en la ejecución de dicho programa.

La notable aportación de John Von Neumann<sup>1</sup> con el concepto de *MÁQUINA DE PROGRAMA ALMACENADO* ha constituido la base de los planteamientos, que aún se siguen aplicando a la mayoría de los computadores. Dicho concepto se fundamenta en los tres principios siguientes:

1. Máquina electrónica digital, que trabaja con información codificada en binario.
2. Programa almacenado en memoria.
3. Posibilidad de provocar una ruptura de la secuencia ordenada de instrucciones en un programa.

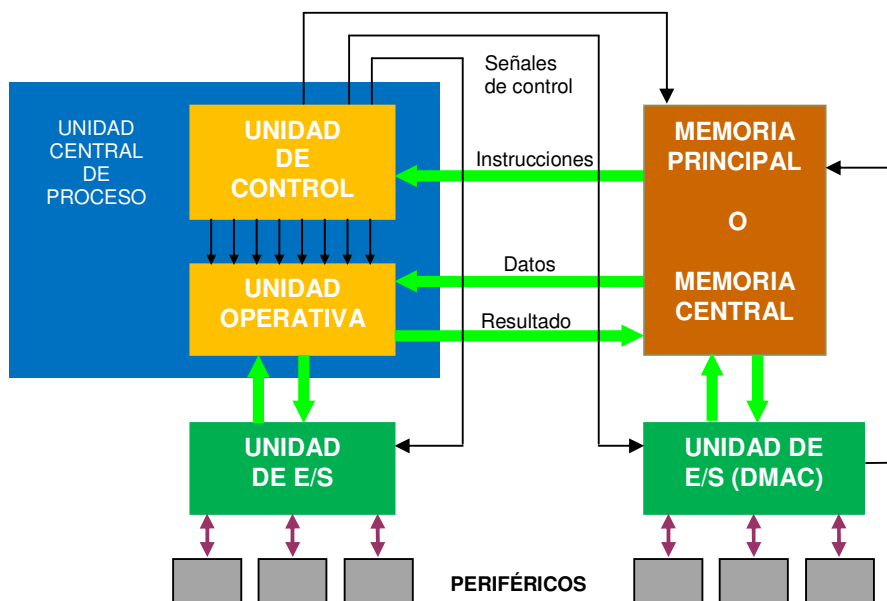


Figura 1.2. Esquema general de un computador.

<sup>1</sup> Matemático de origen húngaro nacionalizado norteamericano.

## 1.2. PRINCIPIO DE FUNCIONAMIENTO DE UN COMPUTADOR

Vamos a dar un paso más en la descripción del computador, y particularmente en el de tipo Von Neumann, para tener una comprensión intuitiva de sus mecanismos.

### 1.2.1. Descripción de elementos

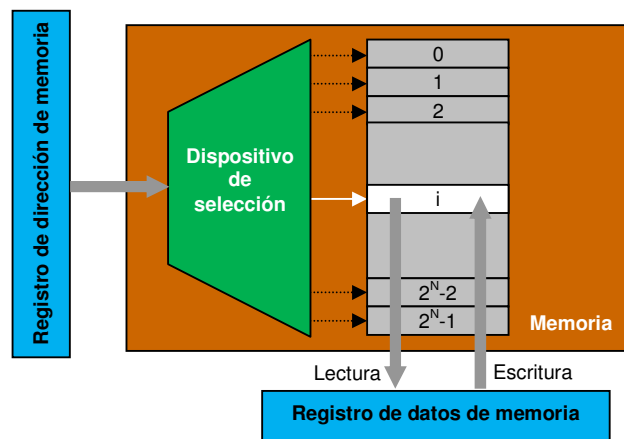
En este apartado se hace referencia a los elementos básicos que forman parte de todo sistema computador, como son: los registros, la Memoria Principal, el programa, la Unidad Operativa, la Unidad de Control, las Unidades de E/S, las Unidades Periféricas y las Interrupciones.

#### REGISTROS

Durante el tratamiento de la información (instrucciones y datos) pueden conservarse temporalmente en elementos de memoria llamados *registros*. Debido a una orden, procedente en general de la Unidad de Control, una información puede ser transferida de un registro a otro, sin que esta transferencia altere el contenido del primero.

#### MEMORIA PRINCIPAL

Podemos considerar la Memoria Principal como un conjunto de *células*, cada una con posibilidad de almacenar una información, denominada *palabra de memoria*, del tipo dato o del tipo instrucción. Las células están numeradas, diferenciándolas la Unidad de Control por un número asociado a la misma, llamado *dirección*. La Unidad de Control puede pedir *LEER* el contenido de una célula de una dirección determinada o *ESCRIBIR* una nueva información en una célula de dirección determinada. Para realizar estas operaciones, la Unidad de Control proporciona a un registro asociado a la Memoria Principal, *Registro de Dirección de Memoria*, la dirección de la célula implicada. El dispositivo de selección de memoria analiza el contenido del Registro de Dirección de Memoria y sensibiliza la célula implicada, bien para una lectura, bien para una escritura. Si se tratase de una lectura, la información almacenada en la célula será transferida a un segundo registro, también asociado a la Memoria Principal, *Registro de Datos de Memoria*. En el caso de una escritura, previamente habrá sido preciso cargar ese mismo Registro de Datos de Memoria con la información que se quiera transferir a la célula en cuestión. La operación de escritura destruye la información almacenada, sustituyéndola por la nueva información del Registro de Datos de Memoria.



**Figura 1.3.** Lectura y escritura en memoria en la posición de memoria  $i$ .

### PROGRAMA

El programa se compone de instrucciones, que suponemos almacenadas secuencialmente en la memoria. Instrucciones que tengan que ser ejecutadas unas a continuación de otras serán almacenadas en direcciones consecutivas de la memoria: finalizada la ejecución de una instrucción de dirección  $A$ , el computador se encadena automáticamente sobre la instrucción de dirección  $A+1$ ; excepto en el caso de ruptura de secuencia.

Esquemáticamente distinguimos tres grandes clases de instrucciones:

- *Instrucciones de procesamiento sobre operandos en memoria.* Comprende fundamentalmente:
  - Operaciones aritméticas
  - Operaciones lógicas
  - Operaciones de transferencia con la memoria
- *Instrucciones de ruptura de secuencia,* que permiten romper el encadenamiento secuencial de las instrucciones y saltar a otra parte del programa obligatoriamente o si se satisfacen determinadas condiciones. Dos tipos de ruptura:
  - Ruptura incondicional
  - Ruptura condicional
- *Instrucciones de intercambio,* que permiten el trasiego de informaciones entre el computador y el medio exterior (a través de las Unidades de E/S). Hay sistemas computadores que no cuentan con instrucciones especiales para las operaciones de E/S; la comunicación entre la Unidad Central de Proceso y las

Unidades de E/S la realizan haciendo uso de las mismas instrucciones que emplean para la comunicación con la Memoria Principal (operaciones de transferencia con la memoria).

### UNIDAD OPERATIVA

La Unidad Operativa o Unidad Aritmética y Lógica opera con los datos que recibe siguiendo órdenes de la Unidad de Control.

La instrucción ha de indicar a la Unidad de Control la operación que debe realizar el sistema y la dirección de los operandos con los que se debe realizar aquella. Por lo tanto estará constituida por un cierto número de bits divididos en dos campos: el *campo de código de operación* y el *campo de dirección*.



**Figura 1.4.** Formato de una instrucción.

El *código de operación* indica la operación que debe realizar el sistema. Las operaciones pueden utilizar un sólo operando (operación monádica), por ejemplo la detección de paridad, o, en el caso más general, dos operandos (operación diádica), por ejemplo la suma aritmética. En este último caso, al realizar la instrucción correspondiente, el sistema ha de conocer:

- La dirección de ambos operandos.
- La dirección del resultado.
- La dirección de la próxima instrucción.

Para definir estas informaciones y tomando siempre como referencia una instrucción que implique una operación diádica, existen varias formas. La forma más inmediata consiste en incluir estas informaciones de forma explícita en la instrucción, dividiendo el campo de dirección en cuatro subcampos, tal como se indica en la [Figura 1.5](#).



**Figura 1.5.** Instrucción de un computador de cuatro direcciones.

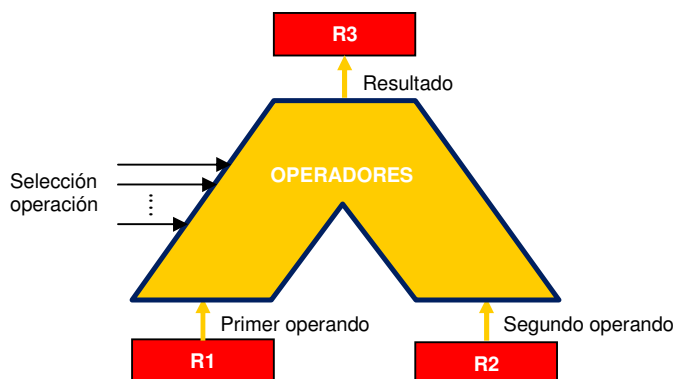
El computador que tiene instrucciones con esa estructura se denomina *Computador de Cuatro Direcciones*. Esta solución presenta como principal inconveniente el que la instrucción necesita un número elevado de bits.

Pero, si como se ha indicado anteriormente las instrucciones se almacenan en posiciones consecutivas de memoria y en el orden en que deben efectuarse, se puede prescindir del campo de dirección que indica dónde se encuentra la próxima instrucción y, sustituirlo por un registro situado en la Unidad Central de Proceso cuyo contenido se va incrementando después de realizada cada instrucción (*Registro Contador de Programa* – Program Counter – PC). Así, de esa forma, se obtiene la instrucción correspondiente al denominado *Computador de Tres Direcciones*, representada en la [Figura 1.6](#).



**Figura 1.6.** Instrucción de un computador de tres direcciones.

La Unidad Operativa capaz de ejecutar esta operación, tanto en el computador de cuatro como en el de tres direcciones, puede ser la representada en la [Figura 1.7](#).



**Figura 1.7.** Unidad Operativa para una máquina de cuatro o de tres direcciones.

En esta máquina, una instrucción de suma por ejemplo, necesita de cuatro accesos a la Memoria Principal, que permiten sucesivamente: buscar la instrucción, después el primer operando y el segundo operando y, por último, almacenar el resultado.

Aparte de las instrucciones que se producen en secuencia, sin cambiar el orden, el computador ha de ser capaz de tomar decisiones, es decir, modificar la secuencia de operaciones en función del resultado obtenido al ejecutar la anterior instrucción. Para ello ha de existir, dentro del conjunto de instrucciones, una cuyo campo de dirección indique la dirección de la próxima instrucción a realizar. A esta instrucción se le denomina de salto o de ruptura de secuencia y puede ser incondicional o condicional.



**Figura 1.8.** Instrucción de salto incondicional.





**Figura 1.9.** Instrucción de salto condicional.

La condición no necesita indicarse obligatoriamente aparte en un campo exclusivo, puede indicarse en el propio código de operación de la instrucción.

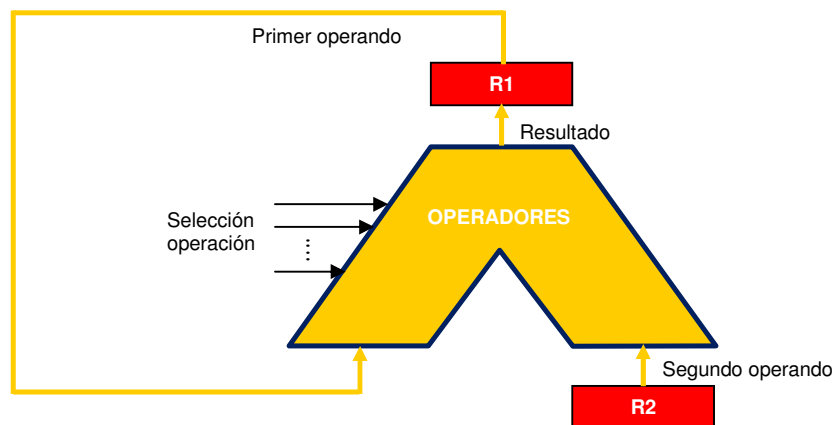
Cuando la Unidad de Control ejecuta una instrucción de salto condicional, observa el valor del biestable de condición correspondiente, definido según el resultado de la instrucción operativa anterior, y rompe o no la secuencia según cuál sea su valor lógico. La ruptura de secuencia se produce cambiando el contenido del Contador de Programa por el campo de dirección de la siguiente instrucción.

Volviendo al formato de una instrucción que implica una operación diádica, el computador de tres direcciones sigue presentando el inconveniente de necesitar un gran número de bits en el campo de dirección. Por ello, si el resultado se almacena en la misma dirección que uno de los operandos, se obtiene el *Computador de Dos Direcciones*.



**Figura 1.10.** Instrucción de un computador de dos direcciones.

La Unidad Operativa correspondiente al computador de dos direcciones se muestra en la [Figura 1.11](#).



**Figura 1.11.** Unidad Operativa para una máquina de dos direcciones.

La búsqueda de una reducción todavía mayor del número de bits de una instrucción condujo al computador de un sólo campo de dirección. La instrucción de un *Computador de Una Dirección* solamente selecciona un operando en memoria, [Figura 1.12](#); por lo

tanto, el otro operando ha de estar en un registro especial situado en la Unidad Central de Proceso, que suele almacenar también el resultado de la operación y por ello se llama *Registro Acumulador*.



**Figura 1.12.** Instrucción de un computador de una dirección.

En el *Computador de Una Dirección* son necesarias instrucciones que permitan, tanto cargar el *Registro Acumulador* con el contenido de una dirección de memoria (instrucción de carga del acumulador – LOAD Acumulador), como que permitan almacenar en una dirección de memoria el contenido del *Registro Acumulador* (instrucción de almacenamiento del acumulador – STORE Acumulador). El formato de la instrucción LOAD Acumulador se representa en la [Figura 1.13](#). De igual forma, el formato de la instrucción STORE Acumulador viene representado en la [Figura 1.14](#).

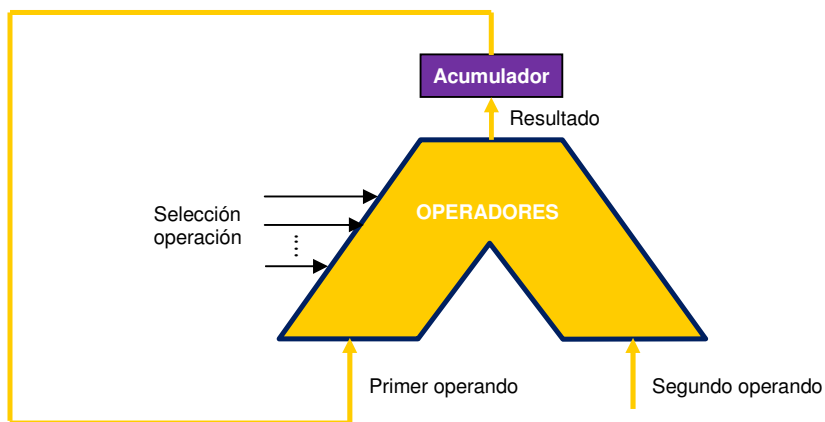


**Figura 1.13.** Instrucción de carga en el acumulador el contenido de una posición de memoria.



**Figura 1.14.** Instrucción de almacenamiento del contenido del acumulador en una dirección de memoria.

La Unidad Operativa correspondiente a un computador de una dirección puede ser la representada en la [Figura 1.15](#).



**Figura 1.15.** Unidad Operativa de una máquina de una dirección.

En la Unidad Operativa de la máquina de una dirección, el acumulador sustituye a los registros R1 y R3 de la Unidad Operativa de la máquina de tres direcciones. El segundo operando puede almacenarse durante la operación en el Registro de Datos de Memoria asociado a la Memoria Principal, no necesitándose de esta forma el registro R2.

En el computador de una dirección, por ejemplo una operación de suma de dos operandos que se encuentran en sendas posiciones de memoria y que queremos almacenar su resultado en una tercera dirección de memoria, necesita de tres instrucciones: la *primera*, para cargar el primer operando en el acumulador; la *segunda*, para sumar el segundo operando con el contenido del acumulador; y, la *tercera*, para almacenar en memoria el contenido del acumulador. Cada una de estas tres instrucciones implicará un código de operación y una dirección; las instrucciones se muestran en la [Tabla 1.1](#).

Código de Operación		Dirección
Instrucción primera	Carga del Acumulador	Dirección primer operando
Instrucción segunda	Suma con el Acumulador	Dirección segundo operando
Instrucción tercera	Almacenamiento del Acumulador	Dirección del resultado

**Tabla 1.1.** Instrucciones para sumar dos datos en memoria principal y almacenar el resultado en la misma memoria principal, para un Computador de Una Dirección.

En el *Computador de Una Dirección*, la complejidad de la Unidad Central de Proceso resulta mínima comparada con la del resto de computadores de más direcciones. Por esta razón, este computador es el que más se ha desarrollado. En los siguientes apartados se hará referencia a los elementos básicos e indispensables para entender el funcionamiento de este tipo de máquinas.

### UNIDAD DE CONTROL

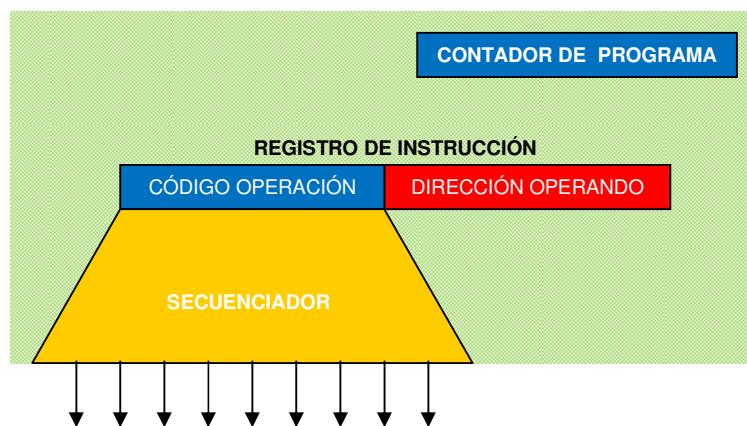
La *Unidad de Control*, es la unidad que extrae y analiza las instrucciones de la Memoria Principal. Para ello, la Unidad de Control necesita de dos registros de propósito especial:

- *Registro Contador de Programa* (Program Counter).
- *Registro de Instrucción* (Instruction Register).

El *Registro Contador de Programa* contiene la dirección de la Memoria Principal donde se encuentra la próxima instrucción a ejecutar. Su nombre se debe a que, a

excepción de las rupturas de secuencia, este registro ve aumentado su contenido al realizar cada instrucción, como ya se ha comentado anteriormente.

El *Registro de Instrucción* contiene durante su ejecución la instrucción extraída de la memoria. El Registro de Instrucción para una máquina de una dirección tiene dos partes perfectamente diferenciadas: una para el Código de Operación (CÓD. OP.) y otra que contiene la Dirección del operando (DIR. OP.).



**Figura 1.16.** Esquema representativo de la Unidad de Control.

La Unidad de Control comporta, además, un órgano llamado *Generador de Secuencias* (*Secuenciador*) quien, tras “analizar” el Código de Operación, distribuye las órdenes mediante un conjunto de señales de control al resto de unidades del computador: Memoria Principal, Unidad Operativa, etc., para hacer ejecutar las distintas fases de la instrucción. Este órgano es el que le impone el carácter de “sistema secuencial” tanto a la Unidad de Control como a la Unidad Central de Proceso.

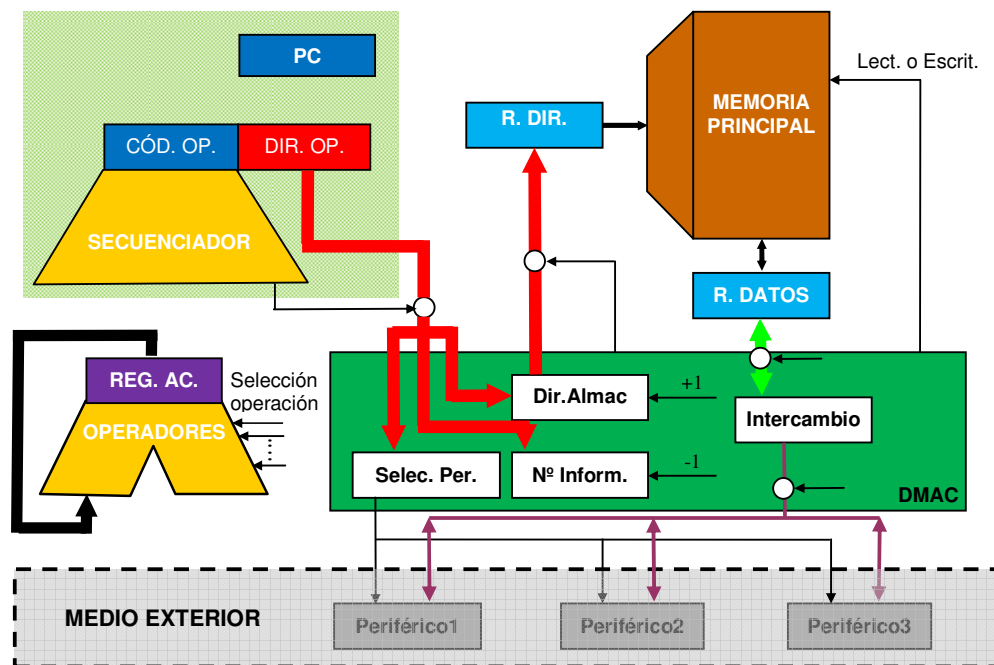
### UNIDADES DE E/S

Las Unidades de E/S actúan como “intermediarios” entre las Unidades Periféricas (en el exterior), y la Unidad Central de Proceso y la Memoria Principal. Aunque no todas las Unidades de E/S son capaces de gestionar sin ayuda de la Unidad Central de Proceso la transferencia de un conjunto de informaciones entre las Unidades Periféricas y la Memoria Principal tanto en un sentido como en otro, existen determinadas Unidades de E/S que pueden realizar independientemente y paralelamente a la ejecución de un programa por parte de la Unidad Central de Proceso, dichas transferencias. Para este caso, las informaciones transferidas se almacenan o se leen de forma secuencial; y, para inicializar tal transferencia, especiales instrucciones de E/S (dependiendo del sistema pueden ser o no ser especiales) deben suministrar a estos dispositivos especiales de E/S, denominados *Controladores de Acceso Directo a Memoria* (DMAC), por una parte, la

dirección de la Unidad Periférica implicada y, por otra, la dirección de la Memoria Principal para el almacenamiento/lectura de la primera información así como el número de informaciones a transferir. Una vez disponga de estas informaciones, el DMAC se ocupa totalmente de la gestión de la transferencia; por cada información transferida, añade uno a la dirección de almacenamiento y resta uno al número de informaciones por transferir. Al final de la transferencia, avisará a la Unidad de Control de que ha terminado.

En la [Figura 1.17](#) se muestra el esquema de funcionamiento de una de estas unidades especiales de E/S, el de un Controlador de Acceso Directo a Memoria (DMAC). En el esquema se muestra tanto la conexión con los buses de comunicación con la CPU y la Memoria Principal, como la conexión con los dispositivos periféricos (en el exterior).

Entre la Unidad Central de Proceso y las Unidades de E/S con capacidad de acceso directo a memoria, debe existir una coordinación para que en cada momento sólo pueda hacer uso de los buses de comunicación con el sistema de memoria sólo uno de ellos (CPU o un DMAC).



**Figura 1.17.** Operaciones de E/S con un DMAC.

## UNIDADES PERIFÉRICAS

Dentro de los dispositivos denominados Unidades Periféricas haremos distinción entre los dos tipos siguientes:

- Las *Memorias Auxiliares*, que sirven de soporte de almacenamiento de gran capacidad (discos, cintas magnéticas, etc.).
- Las *Unidades de Comunicación entre el Sistema y el Exterior* (impresoras, teclados, ratones, pantallas de visualización, etc.).

La mayoría de estas unidades constan de dos partes: una parte electrónica, llamada *Controlador* o *Unidad de Control del Periférico*; y, una *Unidad Electromecánica* que, gobernada por la electrónica, lee o escribe informaciones.

### INTERRUPCIONES

Las interrupciones son órdenes que vienen del medio exterior, tramitadas a través de las Unidades de E/S, que piden al computador ejecutar un programa que está asociado a dicha orden; este programa recibe el nombre de *Rutina de Tratamiento de la Interrupción*. El programa en curso se ve interrumpido para permitir la ejecución del programa solicitado por la interrupción, considerado ahora como prioritario. Acabado este último, se reanuda la ejecución del programa interrumpido. Es gracias a las interrupciones, por poner un ejemplo, como los DMAC's avisan a la Unidad de Control que una operación de E/S ha llegado a su fin.

#### 1.2.2. Desarrollo de una instrucción de procesamiento

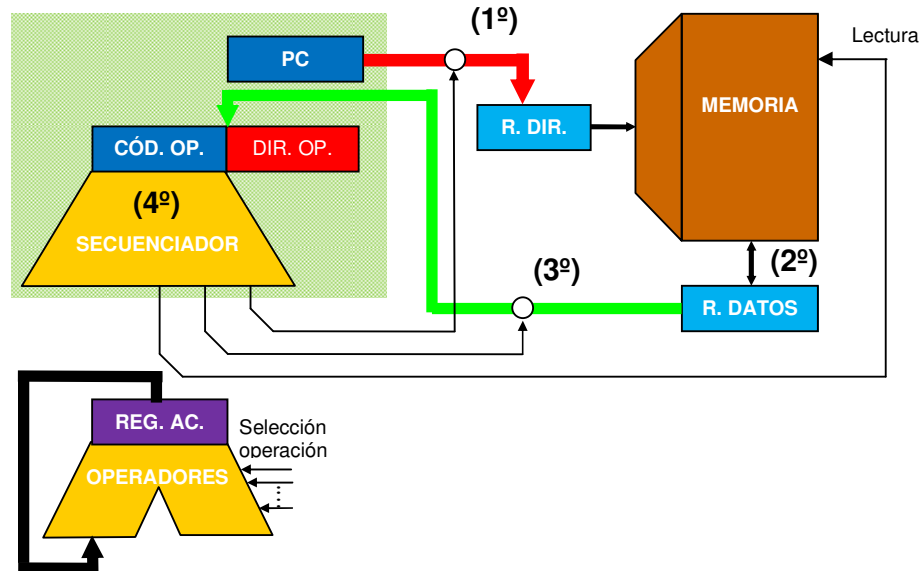
El desarrollo de una instrucción de procesamiento en un computador de una dirección puede describirse según las tres fases siguientes:

- 1) Búsqueda y Análisis de la instrucción.
- 2) a) Búsqueda y Procesamiento del operando  
o  
b) Almacenamiento del operando.
- 3) Preparación para la siguiente instrucción.

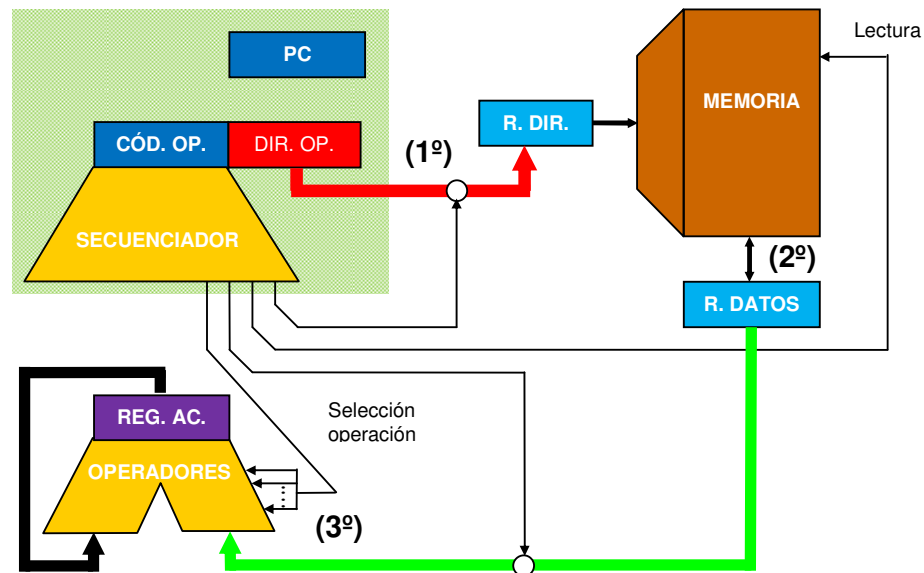
#### FASE DE BÚSQUEDA Y ANÁLISIS DE LA INSTRUCCIÓN ([Figura 1.18](#))

La Unidad de Control ordena la transferencia del contenido del Contador de Programa (es decir, la dirección de la instrucción por ejecutar) al Registro de Dirección de Memoria (**1º**), y envía a la memoria una orden de lectura. Una vez terminada esta operación de lectura en la memoria, la instrucción queda disponible en el Registro de Datos de Memoria (**2º**). Es entonces cuando la Unidad de Control ordena la transferencia

del contenido de este registro al Registro de Instrucción (3º). A partir de aquí los circuitos especializados de la Unidad de Control pueden ya analizar (interpretar, decodificar) el CÓD. OP. de la instrucción (4º). Esta primera fase es común a todas las instrucciones.



**Figura 1.18.** Fase de búsqueda de la instrucción.



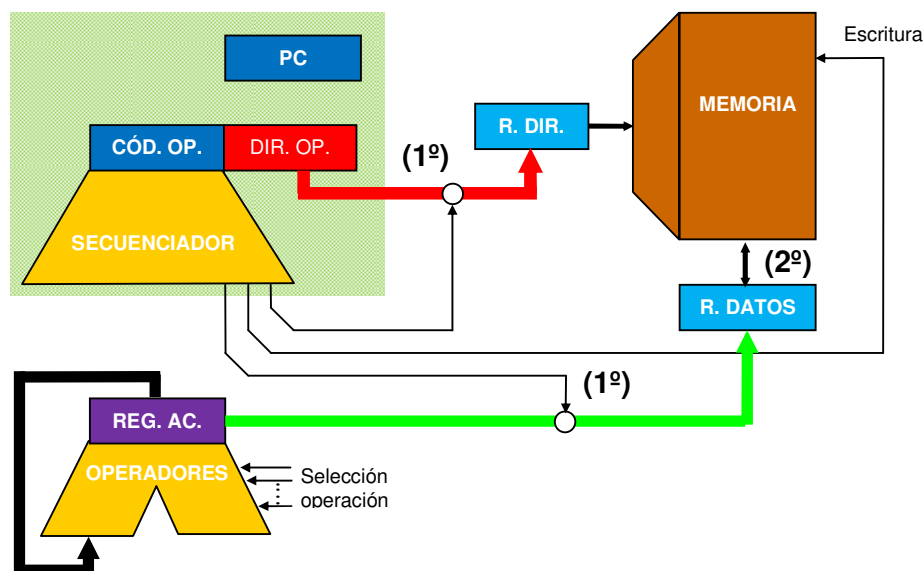
**Figura 1.19.** Fase de búsqueda y procesamiento del operando.

BÚSQUEDA DEL OPERANDO, SEGUIDA DE PROCESAMIENTO DEL MISMO  
([Figura 1.19](#))

Una vez analizado el CÓD. OP. de la instrucción, y si se trata de una instrucción de procesamiento con búsqueda previa de operando, la Unidad de Control transfiere el contenido del campo DIR. OP. al Registro de Dirección de Memoria (1º); y ordena a continuación la operación de lectura en memoria. Al finalizar dicha operación, el operando buscado queda disponible en el Registro de Datos de Memoria (2º), la Unidad de Control posiciona los circuitos de la Unidad Operativa para realizar el procesamiento solicitado por el CÓD. OP. y ordena la transferencia del operando a la Unidad Operativa (3º). El resultado del procesamiento del operando queda almacenado en el acumulador. Obsérvese que un posible procesamiento pudiera ser simplemente una transferencia del operando al acumulador (Instrucción de carga en el acumulador del contenido de una dirección de memoria).

### ALMACENAMIENTO DEL OPERANDO (Figura 1.20)

La dirección de almacenamiento del operando se encuentra en el campo de dirección del Registro de Instrucción; la Unidad de Control ordena su transferencia al Registro de Dirección de Memoria. El operando por almacenar está en el Registro Acumulador; la Unidad de Control ordena su transferencia al Registro de Datos de Memoria (1º). Y, por último, se completa la ejecución ordenando a la memoria una operación de escritura (2º).



**Figura 1.20.** Fase de almacenamiento del operando en una instrucción de almacenamiento del acumulador en una dirección de memoria.

### PREPARACIÓN DE LA SIGUIENTE INSTRUCCIÓN



Consiste esta fase en aumentar el contenido del Registro Contador de Programa (con la descripción que estamos realizando, supone aumentarlo en una unidad ya que, la instrucción la obtenemos con un sólo acceso a memoria). Esta fase, de forma general se realiza de forma solapada con la fase de búsqueda de la instrucción.

### Instrucción de ruptura de secuencia

Este tipo de instrucción, también llamada *Instrucción de Bifurcación o de Salto*, permite modificar el desarrollo secuencial del programa, haciendo que la instrucción de salto sea seguida, no por la instrucción almacenada en la siguiente dirección de memoria, sino por una instrucción cuya dirección viene suministrada por la propia instrucción salto. El salto puede ser condicional; ésta no tendrá efecto más que si se satisface una determinada condición, normalmente relacionada con el contenido del acumulador; si no, el programa continuará en secuencia. El propio CÓD. OP. puede definir la condición y el campo DIR. indica el emplazamiento de la próxima instrucción por ejecutar en el caso de satisfacer la condición. Si la respuesta de la Unidad Operativa es que la condición se ve satisfecha, la Unidad de Control ordena la transferencia de la dirección al Registro Contador de Programa (1º) e inhibe el incremento del mismo; en caso contrario, ordena el incremento del Registro Contador de Programa.

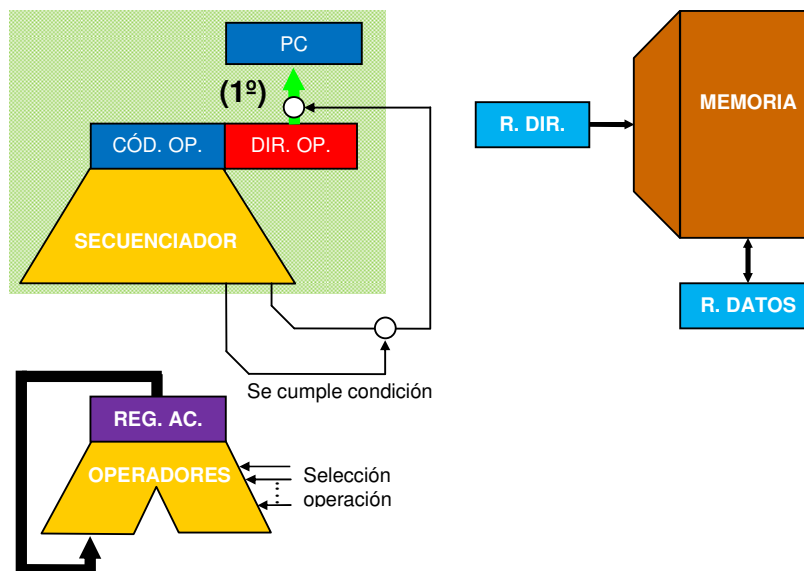


Figura 1.21. Instrucción de salto condicional cuando se cumple la condición de salto.

### 1.3. NIVELES DE ESTUDIO DEL COMPUTADOR

La complejidad de los computadores hace que puedan estudiarse a distintos niveles de detalle, existiendo diversas ópticas en cuanto a su subdivisión en niveles de estudio. En este apartado se hace referencia al estudio del computador desde la óptica de Bell y Newell, de Levy y de Blaauw.

#### NIVELES ESTRUCTURALES (BELL Y NEWELL)

Estos dos autores realizan una *división estructural*, en la que las primitivas o bloques constructivos de cada nivel son los sistemas o conjuntos construidos en el nivel anterior.

El estudio del computador, con esta división, puede abordarse desde cinco niveles; distinguiéndose cada nivel por:

- Un *lenguaje* distinto para describir su estructura y funcionamiento.
- Unas *leyes de comportamiento*.
- Unas *reglas o métodos de diseño* propios.

El primer nivel es el *Nivel de Componentes*. Las leyes aplicadas son las de la electrónica-física, los componentes son semiconductores de tipo n y p, metales, etc.. Los sistemas construidos en este nivel son diodos, transistores, resistencias, condensadores, etc.

El segundo nivel es el *Nivel de Circuito Electrónico*. El comportamiento del circuito se realiza en términos de corriente, tensión, flujo, etc. Los sistemas construidos son las puertas lógicas, los biestables, los osciladores, etc. En este nivel, las leyes utilizadas son las leyes de la electricidad, aunque los circuitos o sistemas construidos son generalmente de tipo digital.

El tercer nivel es el *Nivel de los Circuitos Digitales*. Este nivel puede ser dividido en dos subniveles: el *Subnivel de Circuitos Digitales Combinacionales* y el *Subnivel de Circuitos Digitales Secuenciales*. Las leyes que rigen el comportamiento en este nivel son las del Álgebra de Boole. En el subnivel combinacional se emplean como primitivas las puertas y se obtienen circuitos o sistemas como multiplexores, decodificadores, sumadores, etc., esto es, circuitos sin memoria, cuya salida, en todo momento, es función exclusiva de su entrada en ese momento. En el subnivel secuencial se encuentran los sistemas con memoria; su salida depende de la entrada y de la historia anterior (memoria) del circuito. En estos circuitos hay que tener en cuenta, por tanto, la variable tiempo.

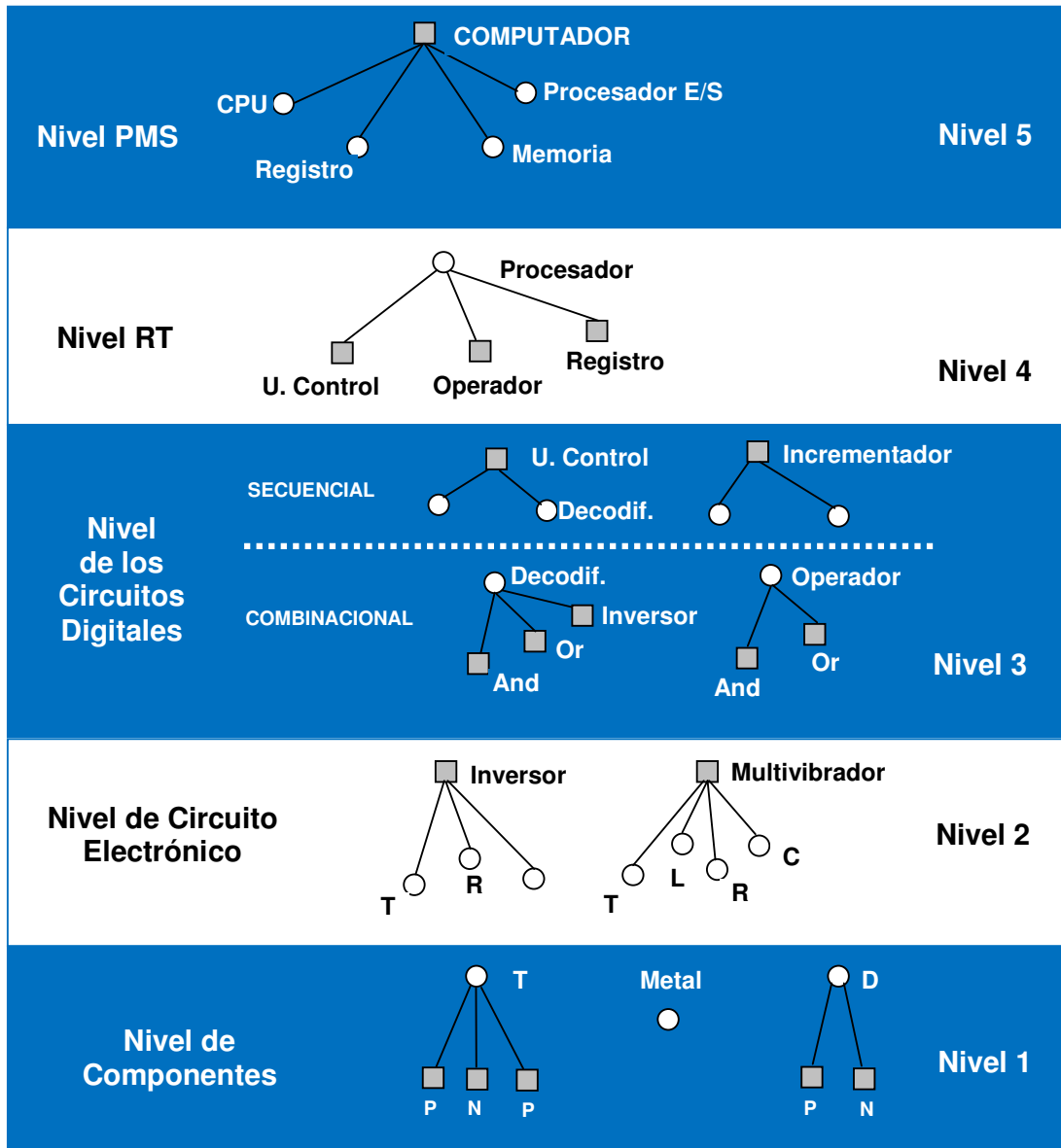


Figura 1.22. Niveles estructurales del computador según Bell y Newell.

El cuarto nivel de representación o estudio constituye el *Nivel de Transferencia entre Registros (Nivel RT)*. En este caso, se estudia el computador como un elemento de flujo de información que se envía de un registro a otro, pasando, en su caso, por el correspondiente circuito combinacional que lo encamina o lo transforma. Las transferencias se representarán mediante flechas, de la forma siguiente:

$A \leftarrow B$

$A \leftarrow B + A$

A, B y C representan registros o posiciones de memoria

$B \leftarrow B \text{ and } C$

Desde el punto de vista de la organización y del funcionamiento del computador, el Nivel RT es uno de los más interesantes y es el que más emplearemos en este curso.

Los elementos constructivos de este nivel son los buses, los registros, los bloques combinacionales, las memorias, etc. Los sistemas formados serán los bloques constructivos del computador (Unidad de Control, Unidad Operativa, etc.) o éste mismo.

Finalmente, el quinto nivel es el denominado *Nivel de Processor Memory Switch* (*Nivel PMS*) es el más alto nivel de representación y estudio. Su utilidad está en representar arquitecturas complejas con varias CPU's, varias memorias, procesadores auxiliares de E/S, etc. Sirve para representar computadores.

### NIVELES DE INTERPRETACIÓN DE LEVY

Esta clasificación en niveles se hace desde un *punto de vista funcional* y, de acuerdo con ella, se considera al computador como una cebolla con varias capas o niveles. Cada nivel se considera como un intérprete que recibe unas instrucciones de un cierto tipo y actúa de acuerdo a ellas.

El nivel más interno o primer nivel es el *Nivel de Proceso de Microórdenes*; en él la máquina realiza la transferencia entre registros, interpretando las órdenes de transferencia entre ellos. Sirve para describir los pasos de ejecución de una instrucción máquina (nivel correspondiente al estudio de la Unidad de Control). Este nivel queda perfectamente formalizado en las máquinas con control microprogramado, mientras que en las de control cableado está implícito en dicho control.

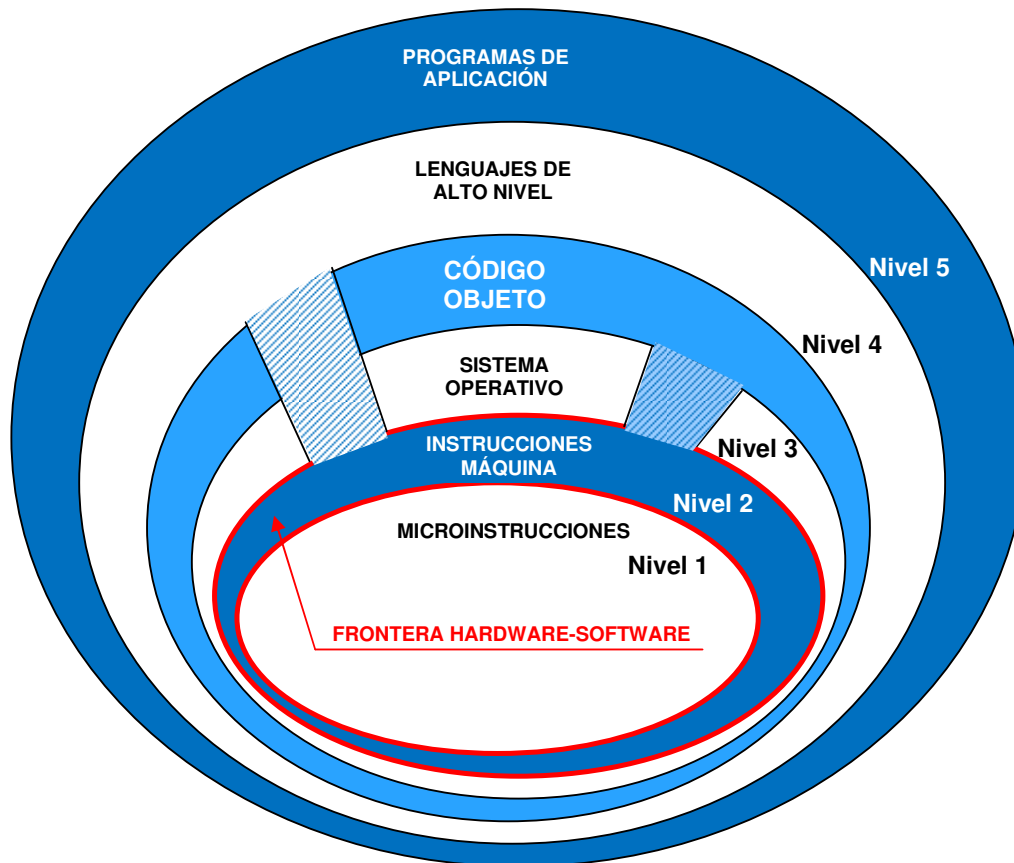
El segundo nivel es el correspondiente a la Unidad Central de Proceso, el *Nivel de las Instrucciones de Máquina*, por lo que la interpretación es la de dichas instrucciones. Este es el nivel de interpretación que ve el programador de ensamblador y se suele tomar como la "frontera entre el hardware y el software".

El tercer nivel es el *Nivel del Sistema Operativo*. El *Sistema Operativo* no es más que un conjunto de programas que ayudan al usuario en la explotación del sistema computador. Es, por tanto, una capa software con la que se rodea al hardware para simplificar su utilización. Las funciones que realiza esta capa van desde la asignación de recursos a los distintos procesos que se desea ejecutar, hasta el control del almacenamiento auxiliar del computador.

El cuarto nivel es el *Nivel de los Lenguajes de Alto Nivel* tipo Fortran, Pascal, Cobol, C, etc. Sin embargo, estos lenguajes no suelen interpretarse sino compilarse, por lo que aparece un nivel intermedio entre el Sistema Operativo y los lenguajes de alto nivel, que es el *Nivel de Código Objeto*, producto de las compilaciones y manejo (generalmente) por

el Sistema Operativo. Adicionalmente, el código objeto e incluso el código fuente en lenguaje de alto nivel, pueden traspasar el nivel de Sistema Operativo e interactuar directamente con el segundo nivel (Nivel de Instrucciones de Máquina).

Finalmente cabría plantear un quinto nivel que es el *Nivel de las Aplicaciones* o los Paquetes de Programas de Aplicación.



**Figura 1.23.** Niveles funcionales o de interpretación de Levy.

### NIVELES CONCEPTUALES DE BLAAUW

Blaauw establece tres *niveles conceptuales* en el computador:

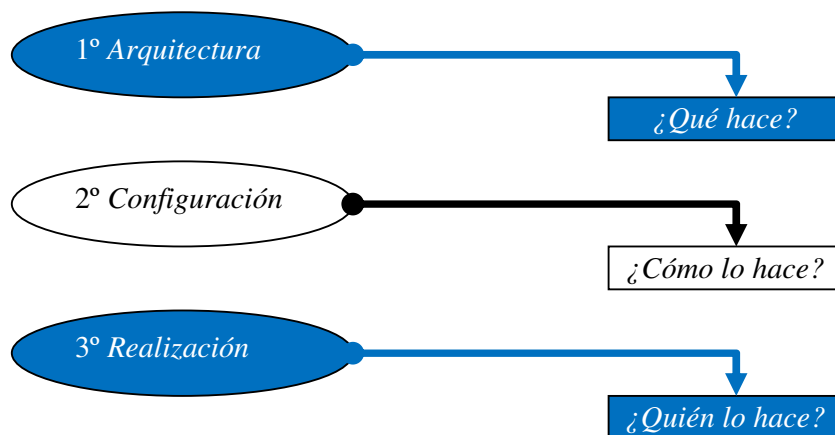
- *Arquitectura.*
- *Configuración.*
- *Realización.*

La *Arquitectura* define el *comportamiento funcional* del computador tal y como aparece para el programador de lenguaje máquina. Este nivel define las representaciones utilizadas por el computador para manejar y almacenar información así como su conjunto de instrucciones de máquina. La Arquitectura define, por tanto, *qué* hace el computador.

La *Configuración* define la *organización interna* del computador, generalmente a nivel de transferencia entre registros y de flujo de información. La Configuración debe diseñarse de acuerdo a la Arquitectura deseada y debe tenerse en cuenta que muchas configuraciones diferentes pueden corresponder a una misma Arquitectura. Sin embargo unas serán más apropiadas para obtener mayores velocidades de ejecución que otras, aunque normalmente con unos costes de realización mayores. La Configuración define, por tanto, *cómo* realiza un computador sus funciones.

Finalmente, la *Realización* se refiere a la forma en que la Configuración es plasmada por *elementos físicos* concretos. El tipo de circuitos lógicos empleados, su nivel de integración, el empaquetamiento, la interconexión de módulos, las ayudas para el mantenimiento, etc. son elementos de este nivel. Es de destacar que una misma Configuración permite múltiples realizaciones, como pueden ser versiones de distinta velocidad de ejecución o resistentes o no a ambientes agresivos. La Realización da forma material a la estructura y define *quién* realiza las operaciones del computador. Este nivel se centra en temas de diseño digital, diseño mecánico, fabricación, etc.

Es interesante destacar que estos niveles no son independientes entre sí. Así, las Configuraciones han de tener en cuenta lo que es posible realizar y construir materialmente, y lo que cuesta construirlo. Por otro lado, la definición de una Arquitectura tendrá que tener en cuenta las posibilidades de Realización actuales y futuras.



**Figura 1.24.** Niveles conceptuales de Blaauw.