

Estructura clásica del computador

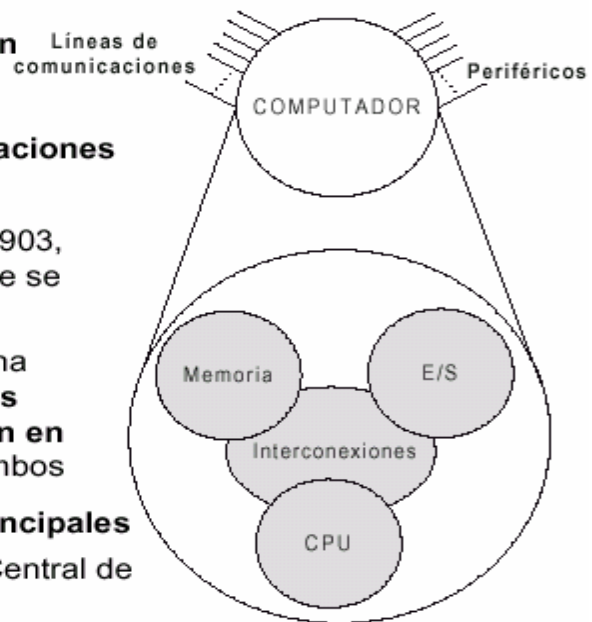
Un computador **intercambia información** con el exterior **a través de periféricos** (dispositivos conectados directamente al computador) **o de líneas de comunicaciones** (dispositivos conectados remotamente).

En **1945**, John Von Neumann (Hungría 1903, EEUU 1957) propone la estructura en que se basan la mayoría de los computadores

Concepto clave: el programa que ordena qué va a hacer el computador **y los datos** sobre los que va a trabajar **se almacenan en memoria**, sin hacerse distinción entre ambos

Cuatro componentes estructurales principales

Memoria, Entrada/Salida (E/S), Unidad Central de Proceso (CPU) e interconexiones



9

Se puede definir la **arquitectura de computadores** como el estudio de la estructura, funcionamiento y diseño de computadores. Esto incluye, sobre todo a aspectos de hardware, pero también afecta a cuestiones de software de bajo nivel.

El primer elemento electrónico usado para calcular fue la válvula de vacío y, probablemente, el primer computador electrónico de uso general fue el E.N.I.A.C. (Electronic Numerical Integrator Calculator) construido en Universidad de Pennsylvania (1943-46). El primer computador de programa almacenado fue el E.D.V.A.C. (Electronic Discrete Variable Computer, 1945-51) basado en la idea arriba expuesta de **John Von Neumann**, que también participó en el proyecto E.N.I.A.C. de que el programa debe almacenarse en la misma memoria que los datos.

En los primeros tiempos de los ordenadores, con sistemas de numeración decimal, una electrónica sumamente complicada muy susceptible a fallos y un sistema de programación cableado o mediante fichas, Von Newman propuso dos conceptos básicos que revolucionarían la incipiente informática:

- a) La utilización del sistema de numeración binario. Simplificaba enormemente los problemas que la implementación electrónica de las operaciones y funciones lógicas planteaban, a la vez proporcionaba una mayor inmunidad a los fallos (electrónica digital).
- b) Almacenamiento de la secuencia de instrucciones de que consta el programa en una memoria interna, fácilmente accesible, junto con los datos que referencia. De esta forma la velocidad de proceso experimenta un considerable incremento; anteriormente una instrucción o un dato estaban codificados en una ficha en el mejor de los casos.

El sistema de memoria

MEMORIA: es la parte del computador que se usa para **almacenar** y después recuperar **datos e instrucciones**. Las operaciones básicas sobre ella se denominan **lectura (R)** y **escritura (W)**.

Fundamentos de la memoria:

- medio o soporte: **almacén físico** de valores binarios 0L y 1L
- operador de escritura: encargado de **generar** físicamente un 0L y 1L
- operador de lectura: encargado de **reconocer** físicamente un 0L y 1L
- gestión de direcciones: encargado de que la escritura y lectura se hagan **en el lugar y en el momento requerido**

Localización de la memoria:

- en la **CPU** (registros): datos temporales
- **interna** (principal): programas y datos actuales
- **externa** (secundaria): resto de programas y datos vía E/S y periféricos

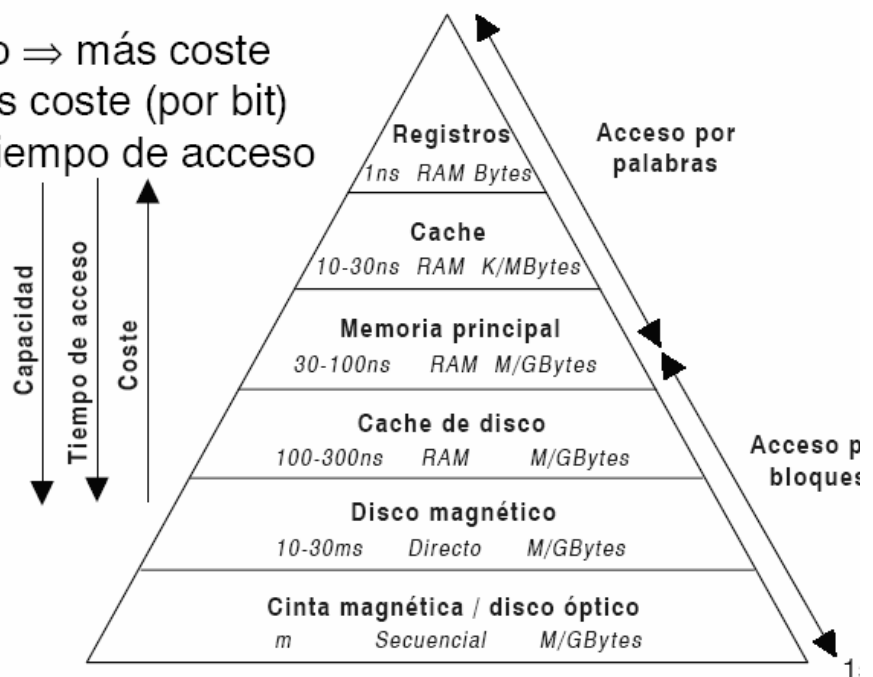
Jerarquía de memoria:

Las restricciones en los sistemas de memoria se ven influidas por la capacidad, el tiempo de acceso y el coste económico

Relaciones de compromiso:

- menos tiempo de acceso \Rightarrow más coste
- más capacidad \Rightarrow menos coste (por bit)
- más capacidad \Rightarrow más tiempo de acceso

Objetivo: la mayor capacidad posible, lo más rápida posible y al menor coste posible \Rightarrow no utilizar una sola memoria sino una **jerarquía de memoria**: la CPU ve su memoria interna y la cache, ésta ve a la memoria principal, y así sucesivamente.

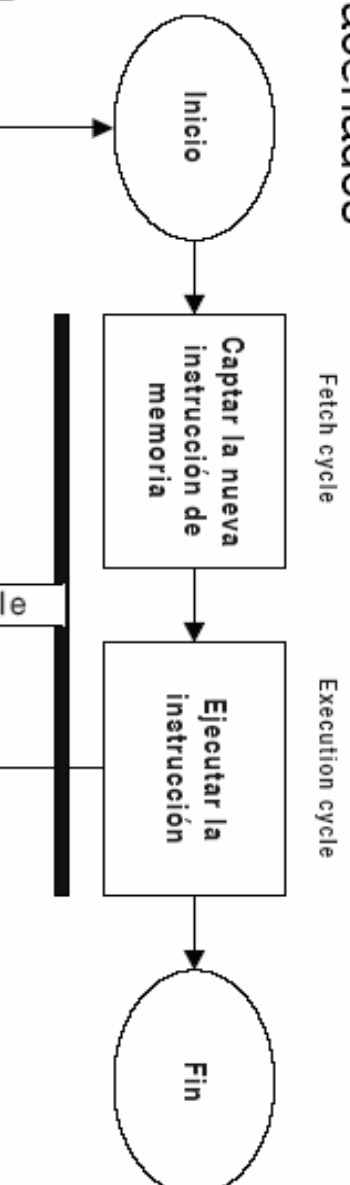


La unidad central de proceso

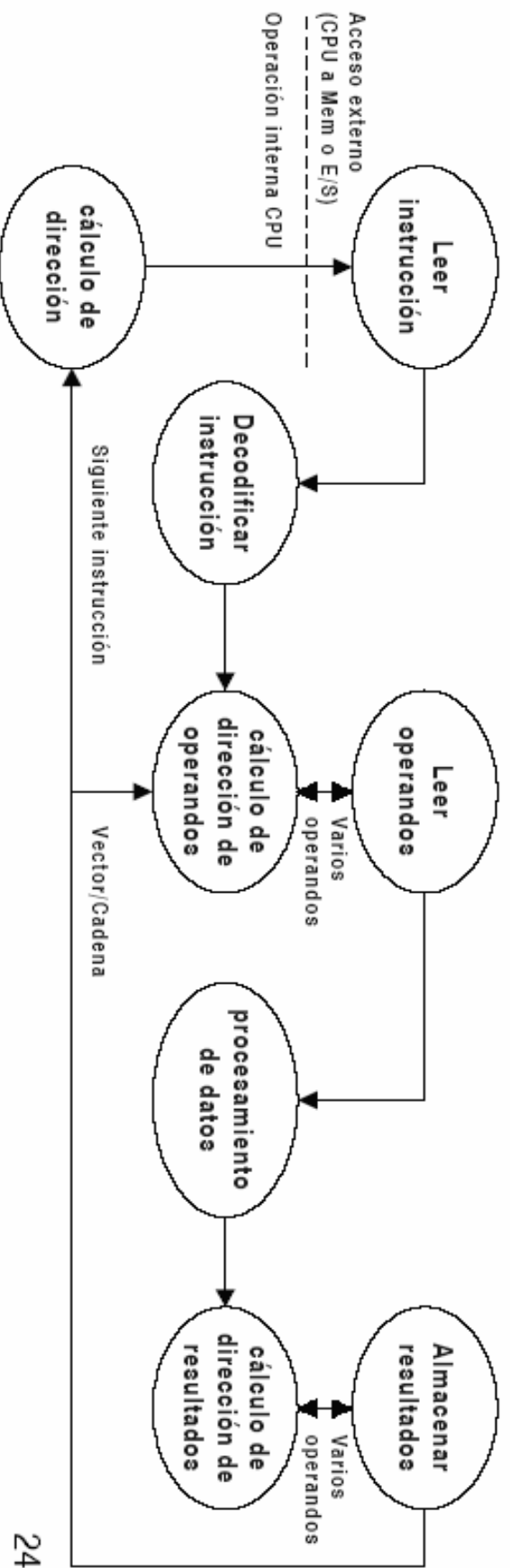
3

UNIDAD CENTRAL DE PROCESO: ejecuta los programas (conjuntos de instrucciones) almacenados en la memoria

Ciclo clásico de instrucción:
el procesamiento de una instrucción de un programa



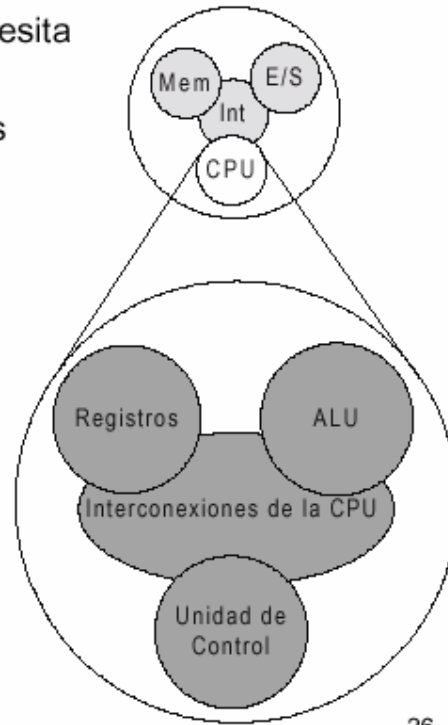
Detalle



La unidad central de proceso

Para poder realizar todas esas tareas la CPU necesita distintos elementos físicos que se encarguen de:

- ➔ **Almacenar temporalmente** los datos, las instrucciones, las direcciones y el estado de operación que se requieren en las distintas fases: **REGISTROS**
- ➔ **Realizar las operaciones** aritméticas y lógicas de procesamiento de datos: **UNIDAD ARITMÉTICO-LÓGICA**
- ➔ **Coordinar las operaciones** de todos los elementos para que puedan realizar sus funciones de manera eficiente y en el instante de tiempo adecuado: **UNIDAD DE CONTROL**
- ➔ **Conectar los distintos elementos** entre sí: **INTERCONEXIONES DE LA CPU**



26

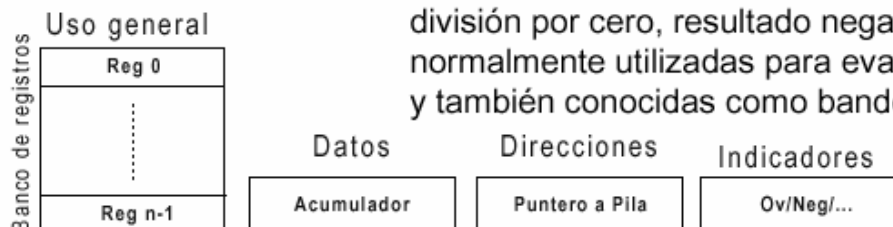
REGISTROS: los registros se encuentran en el nivel más alto de la jerarquía de memoria siendo el tipo de **memoria más rápida, escasa y cara** (por bit)

Tipos de registros de la CPU:

- ➔ de usuario: **visibles y utilizables** directamente **por el programador** de bajo nivel

Tipos de registros de usuario:

- ➔ **uso general:** a discreción del programador
- ➔ **datos:** fuente/destino de operaciones, acumulador, etc.
- ➔ **direcciones:** punteros, índices, direcciones, etc.
- ➔ **indicadores:** condiciones de la UAL como desbordamiento, división por cero, resultado negativo, etc. normalmente utilizadas para evaluar condiciones y también conocidas como banderas (**flags**)



28

Tipos de registros de la CPU:

- ➡ de control y estado: **controlar y almacenar el estado de la CPU**, no accesibles para el programador

Registros de control y estado típicos:

PC	➡ contador de programa : la dirección de la instrucción a captar program counter (PC)
IR	➡ de instrucción : la instrucción que se está ejecutando instruction register (IR)
MAR	➡ de dirección de memoria : la dirección de una posición de memoria memory address register (MAR)
MBR	➡ intermedio de memoria : el dato leído o escrito de la memoria memory buffer register (MBR)
PSW	➡ estado del programa : O, Z, C, S, interrupciones, supervisor, ... program status word (PSW)

29

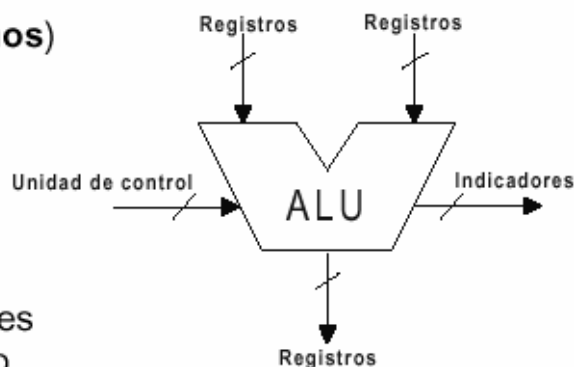
UNIDAD ARITMÉTICO-LÓGICA: encargada de **realizar físicamente las operaciones aritméticas y lógicas** sobre datos, **enteros o en punto flotante**, normalmente mediante unidades digitales sencillas que se utilizan repetida y rápidamente ejecutando un programa (**algoritmo**)

Interfaz y modo de operación:

La unidad de **control** pide a la ALU que realice una operación y le suministra datos a través de los registros de la CPU (**fuentes**) y la ALU efectúa la operación proporcionando los resultados sobre registros de la CPU (**destinos**) e indicando el **estado** de las operaciones (O, Z, C, ...)

En su diseño influyen:

- ➡ el tipo de datos
- ➡ su representación
- ➡ el repertorio de operaciones
- ➡ las restricciones de diseño ...



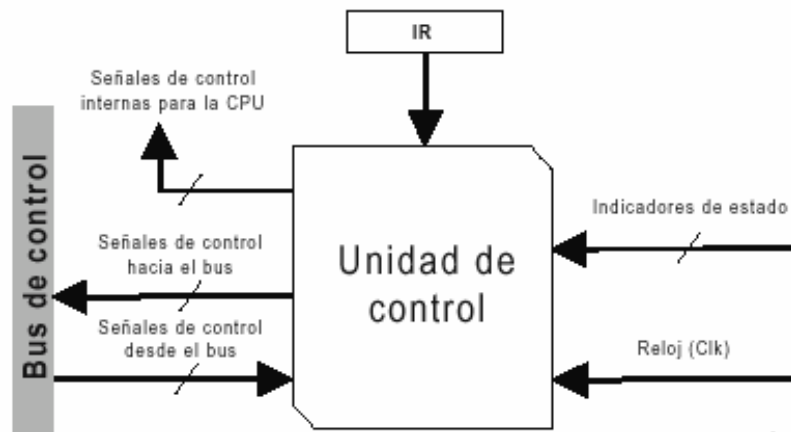
31

UNIDAD DE CONTROL: encargada de indicar **qué, cómo y cuándo** hacen sus funciones el resto de unidades de la CPU

Un computador ejecuta programas, es decir conjuntos de instrucciones. Cada instrucción se ejecuta en un **ciclo de instrucción**. Cada ciclo de instrucción contiene **distintas fases**, cada una de las cuales implica la ejecución de distintas tareas **a nivel de transferencia** (movimiento de datos) **entre registros de la CPU**. A estas transferencias se las denomina **microinstrucciones**, ya que realizan tareas muy simples.

Las microinstrucciones son las **operaciones atómicas** o fundamentales de una CPU y la unidad de control se encarga de decidir qué, cómo y cuándo.

Interfaz y señales:



Tipos de unidades de control:

- ➡ microprogramada: la **notación simbólica** que describía el conjunto de microinstrucciones (microprograma o **firmware**) que formaban parte de la ejecución de la instrucción ADD se define como lenguaje de **microprogramación**

Esto significa que el firmware es más difícil que la programación de un lenguaje de alto y bajo nivel, pero más sencillo que diseñar la secuencia de control hardware equivalente. Por tanto, el firmware está a caballo entre Hw y Sw (**Hardware ⇒ Firmware ⇒ Software**)

Para cada microinstrucción, la unidad de control sólo tiene que **generar los valores concretos (0L ó 1L) para todas las señales de control** existentes, influyan o no en los módulos que ejecutan dicha microinstrucción

Dicho de otro modo, se encarga de poner valores lógicos 0 y 1 en todas las señales (**cables eléctricos**) que indican **qué operación** debe hacer cada módulo y hacerlo **en el instante correcto**

EL SISTEMA DE INTERCONEXIONES está formado por todas las señales que conectan los distintos componentes estructurales situados en el mismo nivel de la jerarquía

Las señales de interconexión son líneas eléctricas que pueden ser:

- ➡ de **uso exclusivo**: dedicadas a una única tarea
- ➡ de **uso compartido**: entre varias unidades o módulos

Cuando son de uso compartido se establecen mecanismos para regular su uso principalmente mediante **métodos de arbitraje** bien **síncronos o asíncronos** (gobernados o no por una señal de reloj). Las conexiones se establecen o no utilizando dispositivos denominados **multiplexor** y **buffer tri-estado** (registro de tres estados).

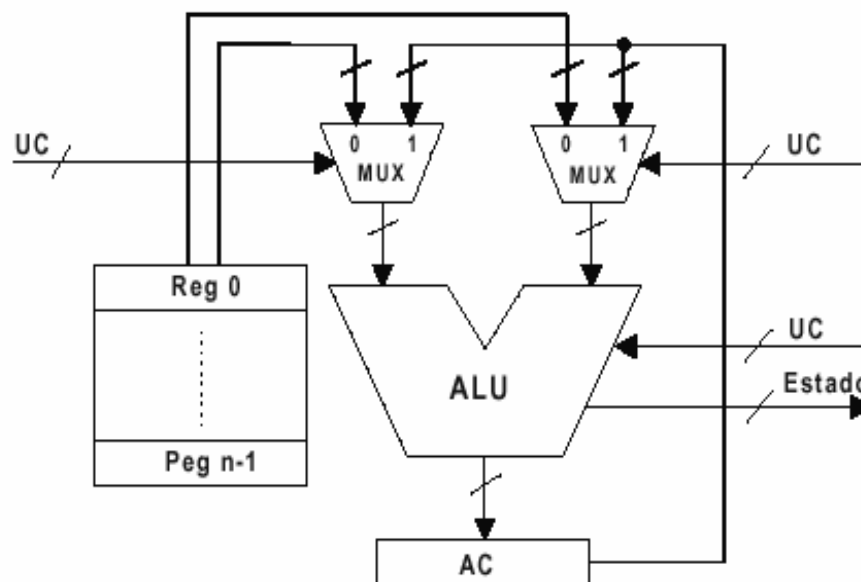
Métodos de arbitraje:

- ➡ centralizado: **un módulo decide** qué, quién, cómo y cuándo realiza una transferencia. Puede ser la CPU o un controlador dedicado
- ➡ distribuido: **cualquier módulo** puede tomar el control del bus

Ambos métodos tienen configuración Maestro-Eslavos

42

EL SISTEMA DE INTERCONEXIONES DENTRO DE LA CPU está formado por todas las señales que habilitan caminos de comunicación entre los registros, la unidad aritmético-lógica y la unidad de control



El sistema de interconexiones: fuera de la CPU

Las interconexiones se realizan a través de **buses** que comunican de forma selectiva los componentes **cumpliendo ciertas normas**, teniendo cada línea un significado particular, y clasificándose en:

- ➡ bus de direcciones: **direcciones de los datos fuente o destino** que circulan por el bus de datos, sean estos datos de la memoria o de los módulos de E/S
Su ancho (número de bits) **determina el tamaño máximo de memoria** a la que se puede acceder
- ➡ bus de datos: por donde **circulan los datos** que se transmiten
Su anchura determina **las prestaciones** del computador
Ejemplo, si el bus de datos es de 16 bits, y las instrucciones son de 32 bits la CPU debe acceder 2 veces a memoria para obtener 1 instrucción
- ➡ bus de control: controla **el acceso y el uso de los buses de datos y direcciones** ya que son líneas compartidas a través de órdenes e información de temporización (validez de datos y direcciones)

46

El sistema de interconexiones que une CPU, memorias y módulos de E/S se suele conocer como **BUS DEL SISTEMA** consta de las entre 50 y 150 líneas individuales que forman los tres sub-buses de direcciones, datos y control

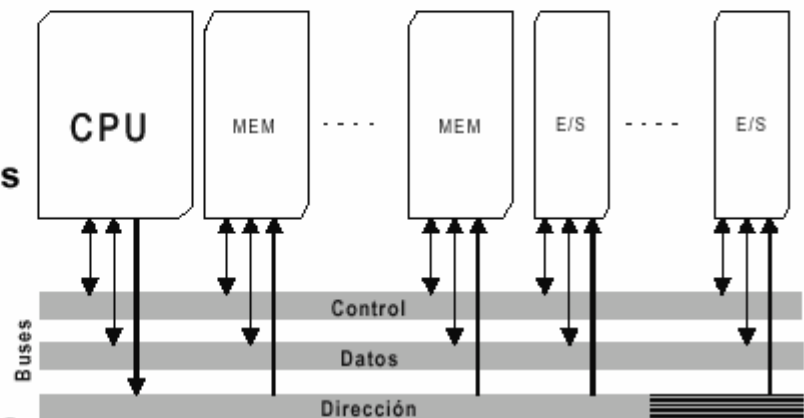
Funcionamiento:

Si un módulo quiere enviar un dato a otro:

- ➡ obtiene el **control del bus**
- ➡ **transfiere** los datos por el bus de datos

Si un módulo quiere pedir un dato a otro:

- ➡ obtiene el **control del bus**
- ➡ transfiere la **petición** por los buses de control y direcciones, **y espera** la respuesta por el bus de datos



Para ello se utilizan banderas o señales de: Vuelta al estado inicial (**Reset**), Petición (**Request**), Cesión (**Grant**), Reconocimiento (**Acknowledge**), etc. 47