



APUNTE N°1: Introducción a los Lenguajes de Programación (LDP)

Prof. Dr. Samuel Sepúlveda
DCI-CEIS-UFRO

2016



CONTENIDOS

1. Definiciones
2. Máquina de Von Neumann
3. Principales componentes del computador
4. Programas y Hardware
5. Actividades a desarrollar

1. DEFINICIONES

1.1 ¿Qué es un LDP?

Lenguaje: Conjunto de caracteres, símbolos, representaciones y reglas que sirven para transmitir uno o más mensajes (ideas) entre dos entidades diferentes.

Programación: Construcción de programas informáticos que permiten introducir y tratar la información en un computador.



Conjunto de elementos válidos (en algún contexto), que nos permiten construir programas informáticos.

1.2 ¿Por qué estos conceptos?

Existen más de 1000 LDP y normalmente uno usa alrededor de 4 sino es 1 ó 2. Entonces el ir aprendiendo nuevos lenguajes puede permitir al programador:

- Mejorar la habilidad para desarrollar algoritmos eficaces,
- Mejorar el uso del lenguaje de programación disponible,
- Acrecentar el propio vocabulario con construcciones útiles sobre programación,
- Hacer posible una mejor elección del lenguaje de programación,
- Facilitar el aprendizaje de un nuevo lenguaje y
- Facilitar el diseño de un nuevo lenguaje.

1.3 Entonces los LDP...

- Proporcionan estructuras básicas (Constructores), que permiten organizar los cálculos y demás tareas.
- Deben ayudar al programador a realizar programas:
 - Fácil de leer
 - Fácil de entender
 - Fácil de modificar

Crear “buenos”
programas

1.4 Características de un Buen LDP

- Claridad, sencillez y unidad:
 - Mínimo conjunto de conceptos que tienen reglas de combinación sencillas y regulares.
- Cercano al dominio del problema:
 - Atributo que permita que la estructura del programa refleje la estructura lógica subyacente del algoritmo.

- Apoyo a la Abstracción:
 - Proyectar e implementar abstracciones adecuadas al problema, basados en elementos más simples del lenguaje.
- Entorno de programación:
 - Compiladores confiables, editores, depuradores, documentación.
- Portabilidad:
 - Programas independientes de la máquina en la que fueron construidos.
- Costo de uso:
 - Costo de ejecución del programa (tiempo de respuesta), costo de traducción (tiempo de compilación), costo de desarrollo (del diseño a la codificación) y costo de mantenimiento.

Obs: Los temas anteriores se detallan en: *Estudio de los lenguajes de Programación, Capítulo I, Lenguajes de Programación, T. Pratt.*

1.5 ¿Cómo los LDP se relacionan con el Hardware?

- Los programas que desarrollamos se basan el uso de LDP que permiten dar instrucciones al HW.
- Luego, nuestros programas se deben ejecutar en un modelo computacional que los soporte.
- Dicho modelo computacional debe “convertir” nuestros programas a instrucciones que el HW “entiende” y ejecute.

2. Modelo de Von Neumann



2.1 Características del modelo

- Seguimiento secuencial de instrucciones (fetch)

- Antecedentes y resultados de operaciones usan acumulador
- Existen instrucciones para intercambio entre memoria y acumulador

2.2 Ventajas del modelo

- El tiempo de acceso a las instrucciones puede superponerse con el de los datos
- Luego se alcanza una mayor velocidad de operación (máquinas más rápidas).

3. Principales Componentes del Computador

Para comprender de mejor forma el modelo de Von Neumann y su relación con los LDP, es necesario conocer algo más de la composición interna de un computador convencional.

Para más detalles se recomienda revisar: *Cap. 1 y 2 eBook de Programación.*

3.1 Componentes

Tomando como modelo las máquinas que incorporan las características de Von Neumann, éstas se pueden considerar compuestas por las siguientes partes:

- Unidad Central de Proceso.
- Memoria Interna.
- Dispositivos de Entrada y Salida, E/S.
- Memoria masiva Externa.

La arquitectura de Von Neumann que, si bien no es la primera en aparecer, sí que lo hizo prácticamente desde el comienzo de los computadores y se sigue desarrollando actualmente. Esta arquitectura hoy en día está siendo desplazada por otras que permite mayores velocidades de procesamiento, como por ejemplo la arquitectura de Harvard.

3.2 La CPU (Central Process Unit)

Se denomina Unidad Central de Proceso al:

- Conjunto de elementos electrónicos, en general de carácter digital,
- Capaz de interpretar y ejecutar el juego de operaciones elementales,
- Así como de gobernar el resto de los componentes que constituyen el sistema de hardware.

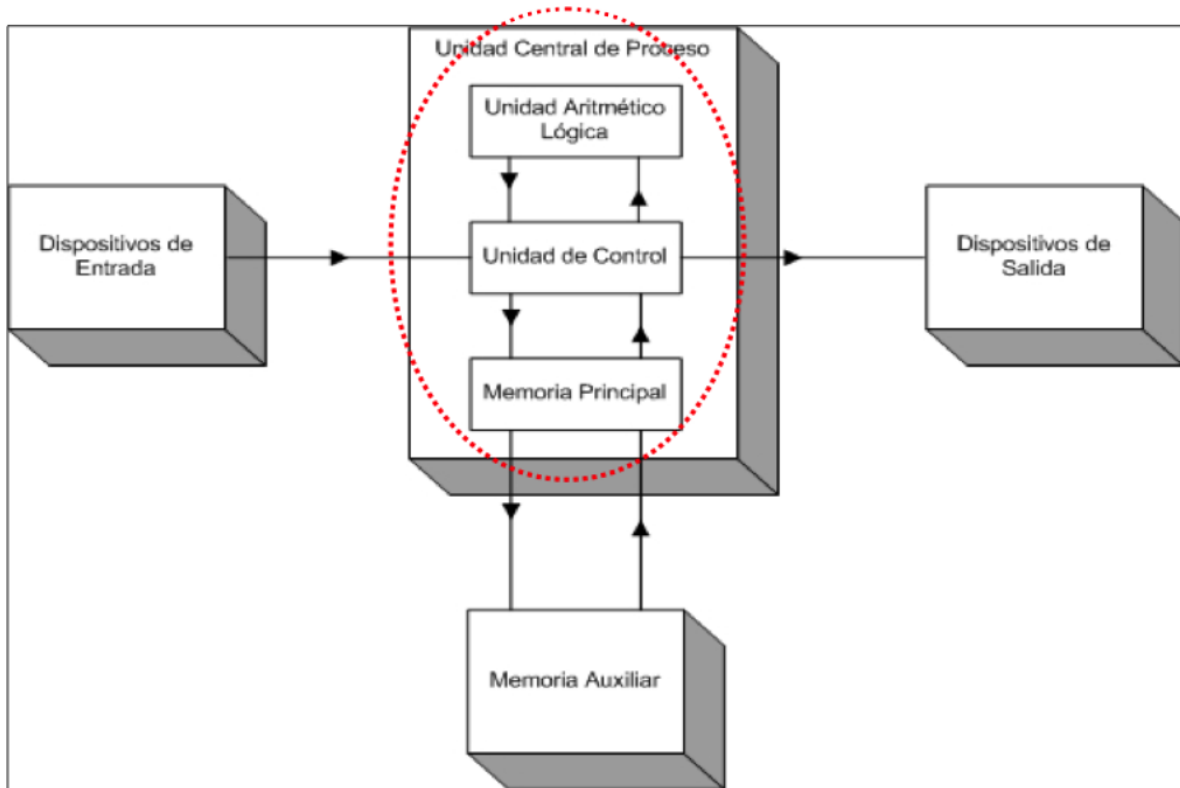
Para más detalles sobre la CPU, su funcionamiento se recomienda revisar *Microprocesadores, Fundamento, Diseño y Aplicaciones en la Industria y en los Microcomputadores 4ª Edición. José Mª. Angulo.*

3.3 La CPU y sus estructura interna

Es posible reconocer 3 zonas básicas estructura interna en la CPU:

- Memoria Principal.

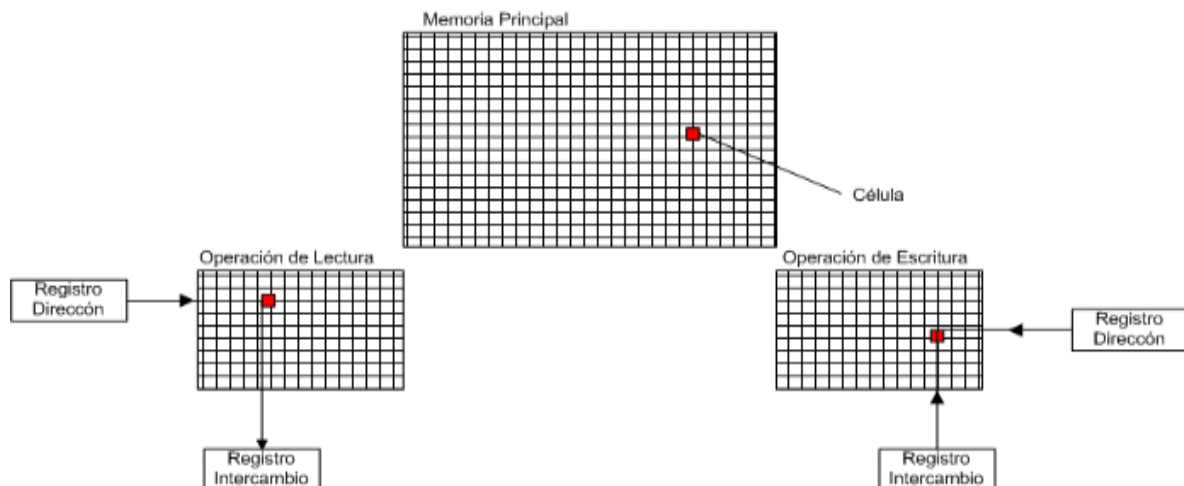
- Unidad de Control.
- Unidad Aritmético Lógica.



Considerar que al interior de la memoria principal se alojan:

- Programa(s) o conjunto de instrucciones a ejecutar
- Datos que manejarán dichas instrucciones

3.4 CPU - Memoria Principal



3.5 CPU – Unidad Aritmético Lógica (ALU, Arithmetic Logic Unit)

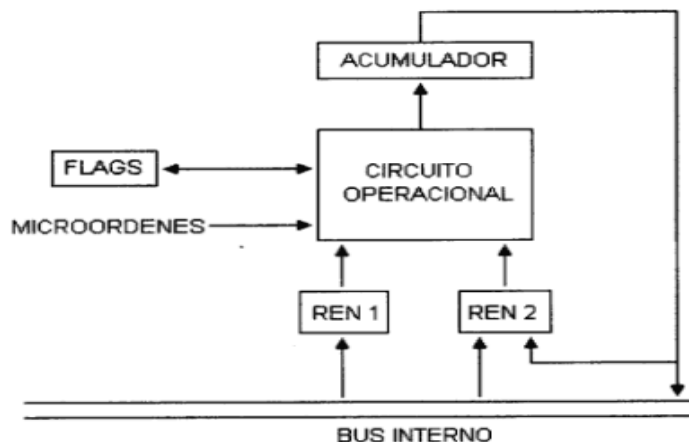
Encargada de realizar las operaciones elementales de tipo:

- aritmético (sumas, restas, productos, divisiones) y
- de tipo lógico (comparaciones).

A través de un bus interno se comunica con la unidad de control la cual le envía los datos y le indica la operación a realizar.

3.6 Registro acumulador

- Almacena los resultados de las operaciones llevadas a cabo por el circuito operacional.
- Está conectado con los registros de entrada para realimentación en el caso de operaciones encadenadas.
- Además tiene una conexión directa al bus de datos para el envío de los resultados a la memoria central o a la unidad de control.



Para que la ALU sea capaz de realizar una operación aritmética, se le deben proporcionar los siguientes datos:

- Código de operación, que indique la operación a efectuar.
- Dirección de las celdas de memoria en la que se encuentran los operandos.
- Dirección de la celda en la que se almacenará el resultado.

4. Programas y Hardware

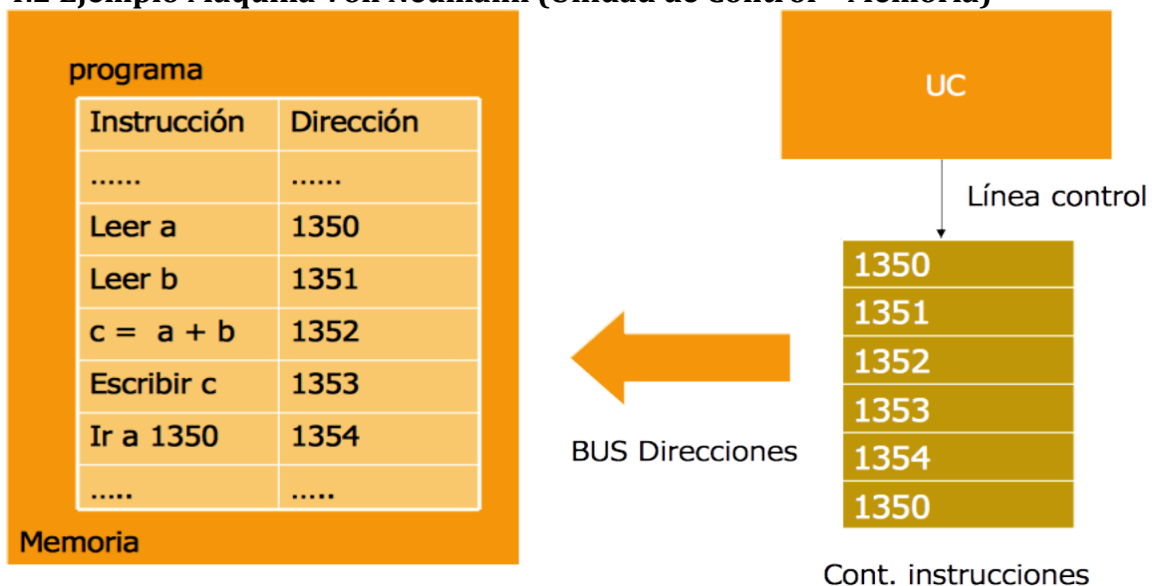
A continuación se ilustra mediante dos ejemplos como el hardware revisado interactúa con un programa de software cuando es ejecutado por una CPU.

4.1 Ejemplo Máquina Von Neumann (ALU - Memoria)



OBS: Debe notar que en paralelo va “trabajando” el bus de direcciones y el bus de datos para llevar los datos e instrucciones a la CPU.

4.2 Ejemplo Máquina Von Neumann (Unidad de Control - Memoria)



5. Actividades a desarrollar

1. ¿Qué otras razones podría argumentar para la necesidad de aprender nuevos LDP?
2. ¿Quién fue Von Neumann? ¿Cuál fue su aporte al campo de la computación e informática?
3. ¿Qué otras arquitecturas de hardware podría mencionar? ¿Qué ventajas presentan sobre la arquitectura de Von Neumann?
4. Explique con sus palabras lo que sucede entre el HW y un programa en Java que captura 2 números enteros por teclado y luego muestra el mayor de ellos.
5. A partir de lo anterior use esquemas para ilustrar lo que sucede a nivel de:
 - memoria y la UAL
 - memoria y UC

Resumiendo

- Características de los LDP
- Arquitectura y máquina Von Neumann
- Estructura interna de un computador
- Componentes de la CPU
- Ejemplos máquina de Von Neumann
- Relación e interacción entre hardware y software