

SISTEMAS DE COMPUTACIÓN I

MÓDULO I

COMPRENDER CÓMO SE CODIFICAN LOS DATOS EN EL INTERIOR DE UN COMPUTADOR Y LOS PRINCIPIOS BÁSICOS DE SU PROCESAMIENTO POR PARTE DE LA UC CONFORME LO ORDENA EL SOFTWARE.

UNIDAD 1:

PROCESOS DE DATOS: HISTORIA Y AUTOMATIZACIÓN

UNIDAD 2:

CODIFICACIÓN Y OPERACIONES CON DATOS

Profesor Titular:

Ing. Mario Ginzburg

Autor de Contenidos:

Ing. Mario Ginzburg

Profesor Tutor:

Ing. Juan Carlos Romero

Diseño Gráfico:

Lic. Paula Bruzzese

PRESENTACIÓN

En esta unidad le proponemos indagar acerca de la forma en que se codifican y operan los datos en el interior de un computador. Verá que es más simple de lo que imagina. Lo importante es comprender que se trata de un lenguaje particular, producto de una construcción arbitraria del hombre, como lo es también nuestro idioma.

No obstante usted deberá, en primer lugar, interiorizarse en el conocimiento de los sistemas numéricos: decimal, hexadecimal y binario. Conocer las características y propiedades de estos sistemas de numeración le facilitará la tarea de comprender y poder efectuar las operaciones que realiza el computador con los datos durante cada uno de los subprocesos que estudiaremos: entrada, memorización, procesamiento y salida.

Esta es la razón por la cual comenzaremos abordando el estudio de los sistemas de numeración. Posteriormente, profundizaremos el análisis de las diferentes codificaciones (código ASCII, lenguaje de máquina, entre otros.) y las operaciones con números binarios, tal como las lleva a cabo la Unidad Aritmética Lógica de un procesador.

Paralelamente, se le brindarán todas las herramientas necesarias para que pueda comprender y utilizar el Programa DEBUG. A través de él podrá vislumbrar, paso a paso, el proceso que lleva a cabo el computador en su interior y verificar, experimentalmente, la resolución de la ejercitación que se le irá presentando en el desarrollo de la unidad.

Le aconsejamos realizar las propuestas de ejercitación en el orden en que vayan apareciendo, pues su concreción impactará significativamente en la comprensión de los contenidos de esta unidad.

Le solicitamos que administre sus tiempos, que realice un control periódico de sus avances cotejando sus aprendizajes con las metas que le proponemos en cada unidad, que consulte a su tutor/a sobre aquellas dudas que están esperando una respuesta y que revise periódicamente el Cronograma de la Asignatura para poder cumplir con los tiempos previstos.

Esperamos que usted, a través del estudio de esta segunda unidad, adquiera capacidad para:

- Comprender la utilidad de los diferentes sistemas numéricos, utilizados en el interior un computador.
- Entender cómo el sistema binario sirve para codificar datos, instrucciones, letras y números, direcciones de memoria; y cómo el sistema hexadecimal permite leer y escribir el sistema binario en forma compacta.

- Conocer los valores máximos y mínimos que se pueden representar en el interior del computador.
- Realizar operaciones de suma y resta en el sistema binario como lo hace la UAL de un procesador.
- Codificar en memoria, en códigos de máquina de Intel o AMD, instrucciones básicas y datos correspondientes a variables tipos magnitudes, enteros y reales como lo haría un programa traductor en una PC, notebook, tablet u otro computador.
- Ejecutar en una PC o notebook, mediante el programa Debug, instrucciones codificadas en memoria, para verificar si ocurren los resultados esperados para la ejecución de cada instrucción.

A continuación, le presentamos un detalle de los contenidos y actividades que integran esta unidad. Usted deberá ir avanzando en el estudio y profundización de los diferentes temas, realizando las lecturas requeridas y elaborando las actividades propuestas, algunas de desarrollo individual y otras para resolver en colaboración con otros estudiantes y con su profesor tutor.

CONTENIDOS Y ACTIVIDADES

1. ¿De qué forma se codifican y operan los datos Dentro de un computador?



TRABAJO COLABORATIVO / FORO

- **Foro: Sistemas Numéricos**



LECTURA REQUERIDA

- Ginzburg, M.; Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pag. 97 a 103. Estructura y operación de un modelo didáctico de computador binario. Pág 5 a 15.



TRABAJO PRÁCTICO SUGERIDO

- Trabajo Práctico N° 2: Sistemas de Numeración: práctica con ejercicios de codificación, conversión y operaciones de suma y resta con binarios naturales

1.2. Codificación ASCII, UNICODE y BCD



LECTURA REQUERIDA

- Ginzburg, M.; Codificación ASCII de caracteres alfanuméricos y unicode. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. p.107-108.



TRABAJO PRÁCTICO SUGERIDO

- Trabajo Práctico Nº3: Código ASCII

1.3. Operaciones de la Unidad Aritmético Lógica (UAL)



LECTURA REQUERIDA

- Ginzburg, M.; Sumas y restas en la UAL. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag 109-112



TRABAJO PRÁCTICO SUGERIDO

- Trabajo Práctico Nº 4: Práctica con operaciones de sumas y restas como las hace la UAL de un computador.



LECTURA REQUERIDA

- Ginzburg, M.; 1.3 Ejecución paso a paso de instrucciones en la PC usando el Debug. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013, pags. 35 a 40.



MATERIAL MULTIMEDIA REQUERIDO

- En el link correspondiente del aula virtual encontrará la presentación denominada **Instructivo del Programa DEBUG**.



TRABAJO PRÁCTICO REQUERIDO

- Trabajo Práctico Nº 5.A Integrador sobre datos numéricos que son magnitudes
- Trabajo Práctico Nº 5.B: Integrador sobre datos numéricos que son enteros



LECTURA REQUERIDA

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 150 a 164.



TRABAJO PRÁCTICO SUGERIDO

- Trabajo Práctico Nº 5.C: Integrador sobre datos numéricos que son reales

**EVALUACIÓN PARCIAL**

Propuesta para la Integración del Módulo I

Para el estudio de estos contenidos usted deberá consultar la bibliografía que aquí se menciona:

BIBLIOGRAFÍA OBLIGATORIA

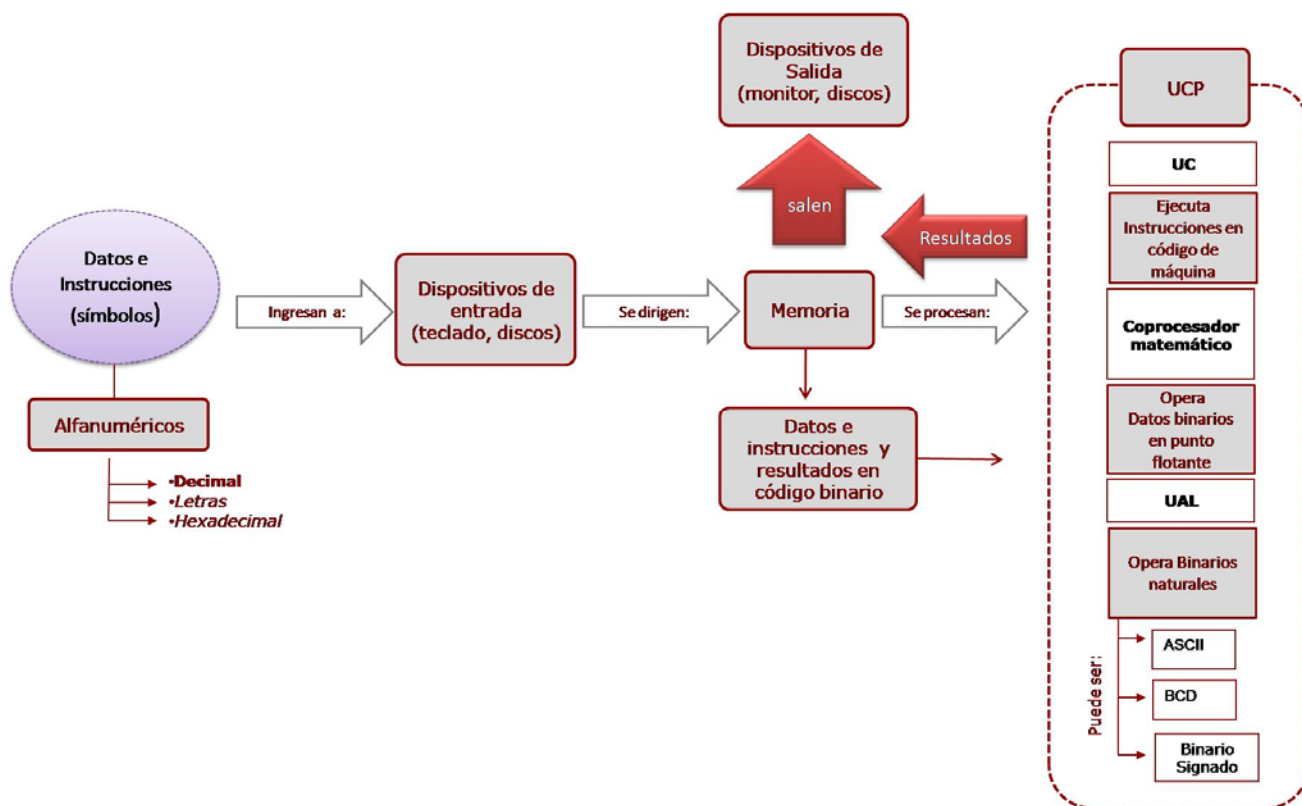
- Ginzburg, Mario. **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013.

Bibliografía Ampliatoria

- García Valle. **Matemáticas Especiales para Computación**. Madrid: Mc Graw Hill, 1988.
- Gutierrez Levine; **Introducción a la Computación**, 2ª Ed. Méjico: Mc Graw Hill, 1992.

ORGANIZADOR GRÁFICO

El siguiente esquema le permitirá visualizar la interrelación entre los conceptos que a continuación abordaremos.



Lo/a invitamos ahora a comenzar con el estudio de los contenidos que conforman esta segunda unidad.

1. ¿DE QUÉ FORMA SE CODIFICAN Y OPERAN LOS DATOS DENTRO DE UN COMPUTADOR?

Autores del presente contenido: Ing. Adrián Rivanera e Ing. Noelia Lloret

Para comprender el lenguaje computacional usaremos el recurso de la analogía, a través del siguiente ejemplo: ¿cómo nos comunicamos los seres humanos?

En principio a través del lenguaje, es decir, de un procedimiento que combina distintos elementos básicos en sonidos, señas o grafos que simbolizan letras, signos de puntuación y caracteres especiales. Las combinaciones a las que aludimos son las “*palabras*”, a las cuales las convenciones sociales les han dado un significado particular. De esta manera, con ciertas combinaciones podemos representar todo

aquello que nos rodea, un sentimiento, una situación, etc. en la medida en que estemos interesados en comunicar.

No todas las combinaciones posibles son utilizadas. Las reglas de representación (sintaxis, ortografía) imponen limitaciones que por el momento no son el objetivo de nuestro análisis. Resumiendo, las palabras gozan del privilegio de poseer un *significado*.

Ahora bien, ¿cuántos elementos básicos tienen un lenguaje como el nuestro? Tenemos: letras (a, b, c, d,), signos de puntuación (, . ; : &) y caracteres especiales (+, -, /, *, <, \$,) pudiendo entre todos sumar unos 200 si se consideran las mayúsculas, las vocales acentuadas y los espacios vacíos.

Pensando sólo en las letras del abecedario contamos con 24 elementos básicos. ¿Podrían ser menos? En realidad sí, pero perderíamos la riqueza lingüística que nos brinda la mayor variedad.

Por otra parte, ¿podrían ser más? En este punto es importante considerar el principio de funcionamiento (factores tecnológicos) de los sistemas transmisor (Tx) y receptor (Rx) del ser humano. Simplificando, las cuerdas vocales (Tx) generan sonidos distintos según las elonguemos y el oído (Rx) recepciona esos sonidos a través del aire, que actúa como medio de transmisión.

Como cada letra puede asociarse a un sonido (fonema) distinto, es posible sonorizar los grafos que las representan y formar secuencias de palabras “habladas” que nos permiten una comunicación constante y fluida. Todo esto resulta sencillo, pero debemos considerar que los artefactos reales poseen limitaciones, es decir, condiciones de funcionamiento anómalo que empeoran su performance.

En nuestro caso, la cantidad de sonidos distintos y distinguibles está limitada tanto por las características constructivas del emisor como por las del receptor. No cualquier sonido puede ser emitido por nuestras cuerdas vocales ni cualquiera recepcionado unívocamente por nuestro oído. En otras condiciones, se obtendría como resultado la *confusión*.

A partir de estas circunstancias se comprende, entonces, la relación entre lenguaje, elementos constituyentes, seguridad interpretativa y tecnologías involucradas.

En ciertas especies los alcances son superiores a los de otra y es así, por ejemplo, que los perros perciben una gama de sonidos más amplia que las personas. En otras, las tecnologías involucradas responden a otros principios (peces, insectos) y por lo tanto cambian los formatos y modalidades de la comunicación.

Supongamos ahora la siguiente situación. Dos personas separadas por una distancia de unos 100 mts. en un espacio abierto, necesitan comunicarse entre sí mediante unos cuantos mensajes comunes. Por

razones obvias, utilizar el sonido como técnica no resulta conveniente. Si es de noche o la visual se ve dificultada, un lenguaje de señas tampoco sería adecuado. Por supuesto, descartamos la posibilidad de comunicarse por los celulares.

Una opción es recurrir al **fenómeno lumínico**: al encender o apagar una linterna (dos estados) puede establecerse un canal de comunicación donde el ojo actúa como Rx y nuestra mano con la linterna como Tx. En este caso, las tecnologías involucradas reducen a dos los elementos básicos: *encendido* y *apagado*. Para formar palabras deberemos combinar estos dos símbolos de manera de obtener secuencias diferentes que nos permitan asignar los significados de cada una.

A continuación, se muestran distintas posibilidades que confirman la necesidad de aumentar la longitud de cada término para obtener mayor cantidad de palabras:

1 símbolo: 2 palabra = 2^1

E
A

2 símbolos: 4
palabras = 2^2

EE
EA
AE
AA

3 símbolos: 8
palabras = 2^3

EEE
EEA
EAE
EAA
AEE
AEA
AAE
AAA

En general, tratándose de lenguajes de dos elementos (lenguajes binarios), combinando n símbolos por palabra obtendremos 2^n palabras distintas (combinaciones). En este punto es importante recalcar dos aspectos:

- Un “encendido” y un “apagado” son eventos fácilmente distinguibles, prácticamente sin posibilidad de error. Claro que una tercera opción sería la de incluir un símbolo adicional representado por un encendido “atenuado”, lo cual nos daría posibilidades de mayor cantidad de combinaciones. Pero, ¿cómo podríamos asegurar que no se trata de un encendido pleno en presencia de niebla o con la batería agotada? El resultado: perderíamos seguridad en el significado del mensaje.
- Por otra parte, la inclusión de otros símbolos necesitaría de mecanismos de generación y recepción más complejos y por lo tanto, el agregado de costos no siempre justificados por los resultados.

Volvamos ahora a nuestro objetivo: la computadora.

Su funcionamiento está basado en los principios de las tecnologías eléctrica y electrónica, donde la formulación de señales binarias conlleva a la utilización de unidades simples, económicas y de pequeñas dimensiones. Si a esto le sumamos la seguridad en la interpretación y transformación de la información, concluimos en el porqué de la adopción de un sistema de las características vistas, donde ahora los símbolos pasarán a denominarse ON-OFF, SI-NO, etc.

Nos falta un paso más. ¿Cómo representamos con estos dos estados todas las palabras, órdenes, datos, etc. que permanentemente recorren los circuitos de una PC?

Como lo mencionamos anteriormente, el lenguaje común (o coloquial) presenta una serie de problemas de interpretación: cada palabra puede tener más de un significado de acuerdo con la frase donde está insertada, con la intencionalidad del emisor, con los usos y costumbres, etc. ¿Cómo haríamos para que la computadora pudiera rescatar el significado correcto en cada caso? Este escollo no sería fácilmente salvable y, en todo caso, agregaría complejidades no deseadas.

La alternativa elegida fue otra: adoptar un código de carácter universal, preciso por definición y sujeto a reglas fáciles de interpretar, transmitir y operar por la máquina.

1.1 SISTEMAS DE NUMERACIÓN

Autor de los contenidos siguientes del módulo: Ing. M Ginzburg

Un **sistema de numeración** es una forma arbitraria de fraccionar la realidad, un conjunto de símbolos y reglas de generación que permiten construir todos los números válidos en un sistema. Consiste en dividir un conjunto de elementos que se quiere simbolizar, en grupos.

Cada uno de éstos tiene el tamaño, tantas veces mayor que el anterior, como símbolos tiene el sistema.

Es una totalidad divisible en subconjuntos, de acuerdo con una determinada convención. Nuestra cultura está acostumbrada a dividir la realidad en subconjuntos, en que cada uno de ellos es diez veces mayor que el siguiente (sistema decimal).

No obstante, existen otros sistemas de numeración tan válidos y útiles como el decimal, entre ellos: los sistemas binarios (base 2), el sistema octal (base 8) y el sistema hexadecimal (base 16).

El hombre, en la antigüedad, comenzó a contar objetos utilizando sus diez dedos, realizando marcas en bastones, nudos en las cuerdas, para ir pasando de un número al siguiente. Algunos pueblos

utilizaron un solo tipo de símbolo (sistemas unitarios) y para representar cada uno de los elementos realizaban tantas marcas como números de objetos.

Sin embargo, a medida que la cantidad de símbolos se fue acrecentando, se hizo necesario un sistema de representación más práctico. Para disminuir la cantidad, se establecieron operaciones (de sumas y multiplicación) implícitas entre los símbolos.

Antes de profundizar sobre el desarrollo de los sistemas numéricos posicionales, le proponemos realizar la siguiente actividad con el fin de investigar y compartir con sus colegas la forma en que cada cultura ha desarrollado diferentes sistemas para representar una misma cantidad de elementos.



Trabajo Colaborativo

Foro: Sistemas Numéricos

- Lo invitamos a navegar en la web e investigar acerca de otros sistemas numéricos que fueron desarrollados por diferentes pueblos, a lo largo de la historia de la humanidad.
- Realice una breve descripción de los sistemas numéricos romano y egipcio, e identifique las semejanzas y diferencias entre ellos.
- Comparta sus conclusiones en el foro.

Los **sistemas numéricos posicionales** (decimal, binario, hexadecimal, octal) se basan en un conjunto limitado y constante de símbolos. Su particularidad es que cada símbolo, además del número de elementos que representa, tiene significado o peso distinto según la posición que ocupa en el grupo de caracteres del que forma parte.

Se trata de un sistema con sumas y multiplicaciones implícitas. Con sólo visualizar un número, se puede apreciar sin pensar su **magnitud**. Es decir, la cantidad de elementos que el número simboliza.

Por ejemplo, al observar el número 382 podemos dar cuenta de su magnitud, sin tener que descomponerlo en centenas, decenas y unidades. Recuerde que la magnitud, se determina sumando los productos que se obtienen al multiplicar el valor de cada dígito, por el “peso” de la posición que ocupa. Si continuamos con nuestro ejemplo, $382 = 3 \times 100 + 8 \times 10 + 2 \times 1$

Luego de esta introducción que trata de aportar la idea generatriz de los sistemas del procesamiento digital y los sistemas numéricos, lo/a remitimos a realizar la siguiente lectura, focalizando su atención en los apartados que especifica la guía y utilizando papel y lápiz, pues la comprensión del tema exige la realización de los ejercicios propuestos en los ejemplos.

Los elementos de los circuitos de un computador sólo pueden mostrar dos estados físicos. Por ejemplo cada uno de los millones de transistores de un chip está en un estado de alta resistencia, o bien en otro de resistencia baja, lo cual hace que también los conductores presenten dos voltajes (2 volts ó 0 volts) que se simbolizan con “1” y “0” respectivamente.

Si en un instante dado, un cierto grupo de 4 conductores, tomados en un cierto orden, están respectivamente en 2 volts, 0 volts, 2 volts y 2 volts, implica que están transportando el número binario 1011.

Como se verificará, con los números binarios se pueden codificar:

- a) **instrucciones** (de un programa),
- b) **datos** (números naturales, enteros o reales a operar, que a su vez representan magnitudes, letras, imágenes, etc.),
- c) **resultados** parciales o totales de un proceso de datos realizado (números nuevos que antes no existían),
- d) **direcciones** identificatorias de lugares (memorias) donde están almacenados números de los tipos a), b), o c)



Lectura requerida

- Ginzburg, M.; Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag. 97 a 112.

Guía para la lectura

- ¿Qué es un sistema numérico posicional? ¿Qué ventajas tiene el sistema numérico decimal en relación al sistema de numeración egipcio?

- ¿Qué tienen en común los sistemas posicionales?
- Características y propiedades de los sistemas numéricos binario y hexadecimal.
- Conversión entre bases.

Le proponemos realizar la siguiente actividad; su elaboración le ayudará a apropiarse los contenidos desarrollados hasta el momento.



Trabajo práctico sugerido

Trabajo práctico Nº 2:

Sistemas de Numeración: práctica con ejercicios de codificación, conversión y operaciones de suma y resta con binarios naturales.

Consulte la consigna de este trabajo en el link correspondiente del aula virtual.
Comuníquese con su tutor si se le presentan dudas en la resolución de las actividades.



Tenga en cuenta que los trabajos que produzca durante los procesos de estudio son insumos muy valiosos y de preparación para la Evaluaciones Parciales. Por lo tanto, guarde sus notas, apuntes y gráficos, le serán de utilidad.

Por otro lado considere que las dudas pueden ser consultadas en los Foros o en el espacio de Tutoría.

El **sistema binario**, al ser base 2, representa cualquier conjunto de elementos utilizando únicamente los símbolos 1 y 0.

Es el único sistema numérico que puede representar el estado interior del hardware de un computador, ya que cada transitor o conductor que compone dicho hardware sólo puede encontrarse

en dos estados físicos: conducir o no electricidad. Uno de dichos estados se corresponde con 0 y el otro con 1. Se puede simbolizar con una llave de luz: encendida (1) y apagada (0).

Sin embargo, en el uso práctico, las representaciones de cada símbolo pueden ser largas cadenas de bits (0 y 1) y confundir fácilmente al usuario al leerlas o escribirlas.

Razón por la cual, se utiliza el sistema numérico hexadecimal para evitar representar, en el papel o en la pantalla del monitor, largas secuencias de unos y ceros. La conversión del sistema binario al hexadecimal es rápida y sencilla, casi instantánea: cada cuarteto binario corresponde unívocamente con un símbolo hexadecimal (La PC por Dentro, 6ª ed. pag 106).

Le proponemos realizar la siguiente actividad de lectura que le permitirá profundizar lo estudiado acerca de la relación del hardware involucrado en los diferentes subprocesos (desarrollados en la primera unidad) y la utilización del sistema binario.



Lectura requerida

- Ginzburg, M.; Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag. 97 a 112.

Guía para la lectura

¿Cuál es la ventaja de operar en el interior de un computador con dos estados eléctricos correspondientes al 0 y 1 binario?

La siguiente actividad podría favorecer la comprensión de lo trabajado hasta el momento y ser puntapié inicial para facilitar la asimilación de los contenidos que se presentarán a continuación; téngala en cuenta.



Actividades para la facilitación de los aprendizajes

- Lo invitamos a tomar un lápiz y escribir los números binarios del 0000 al 1111, uno debajo de otro en orden creciente, utilizando el método de las pesas. Una vez construida la tabla podrá advertir que con 2 bits puede formar 4 números binarios distintos; con 3 bits, 8 números, etc. y, de este modo, generalizar que para n bits puede formar 2^n números o combinaciones binarias, como por ejemplo con 4 bit (2^4) se forman 16 combinaciones, con 5 bit (2^5) 32 combinaciones, etc.
- ¿Qué relación se da entre incrementar en uno el exponente n , y el resultado de 2^n ?



Lectura requerida

- Ginzburg, M.; Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag. 97 a 112.

Guía para la lectura

Le proponemos que retome lo estudiado acerca de las semejanzas entre el computador y la fábrica. Es importante que tenga presente estos conceptos para seguir avanzando en el estudio de la unidad.

¿Qué es lo básico que se necesita conocer acerca de bit, bytes y de la equivalencia entre el sistema binario y el hexadecimal?



Tenga en cuenta que los trabajos que produzca durante los procesos de estudio son insumos muy valiosos y de preparación para la Evaluaciones Parciales. Por lo tanto, guarde sus notas, apuntes y gráficos, le serán de utilidad.

Por otro lado considere que las dudas pueden ser consultadas en los Foros o en el espacio de Tutoría

1.2. CODIFICACIÓN ASCII, UNICODE Y BCD

Considerando lo desarrollado hasta el momento, usted debería estar en condiciones de justificar la necesidad de utilizar un sistema de comunicación informático que conste de dos símbolos 0 y 1 bits; y deducir en cualquier sistema numérico cuántos dígitos hacen falta para formar un número de combinaciones distintas.

Luego de realizar las lecturas habrá advertido que en el sistema decimal para formar 100 números (100 combinaciones distintas) necesita únicamente dos dígitos (del 0 al 99) = 10^2 ; para formar mil combinaciones, tres dígitos (del 0 al 999) = 10^3 y así sucesivamente.

Por otro lado, en el sistema binario, al ser de base dos, para formar 16 combinaciones (16 números) necesitará 4 números binarios (00, 01, 10, 11) = 2^4 , para formar 8 combinaciones, 3 dígitos (2^3), etc.

A partir de los ejemplos mencionados, le proponemos realizar la siguiente actividad para comenzar a reflexionar sobre las diferentes convenciones en código binario utilizadas en el interior del computador.



Actividades para la facilitación de los aprendizajes

- Teniendo en cuenta que la cantidad de los caracteres del teclado son 128, ¿cuántos *bits* hacen falta para representar todos los símbolos? ¿Y para 256 caracteres?
- ¿Cuántos bits hacen falta para formar 67 combinaciones binarias?
- ¿Qué utilidad tienen las potencias 2^{10} y 10^3 ?
- ¿Cuántos dígitos binarios hacen falta para representar el número 256.325.448?

La siguiente actividad podría favorecer la comprensión de lo enunciado hasta el momento.



Actividades para la facilitación de los aprendizajes

Teniendo en cuenta la combinación alfanumérica de la placa patente del automóvil de la fotografía, lo invitamos a analizar la cantidad de combinaciones posibles que se pueden obtener a partir de esa convención.

¿Se le ocurre alguna otra forma para optimizar la cantidad de combinaciones?



Pensando en la relación usuario-computadora, ¿cómo hacemos, entonces, para introducir mediante el teclado letras y caracteres que luego se deben reproducir en la impresora o en la pantalla del monitor?

La forma más simple es la de asignar a cada caracter una combinación numérica binaria, fija y única; de manera tal que al pulsar la tecla "A" en el teclado llegue a la memoria de la computadora el número 01000001. Este procedimiento se denomina **codificación** y su asignación es universal, de forma de asegurar que todas las PCs manejen el mismo código.

El código más empleado en nuestros días, es el **A.S.C.I.I.** (se lee "asqui") y significa *American Standard Code for Information Interchange*. Es un sistema que asigna una representación a cada una de las letras del alfabeto, los dígitos, los signos de puntuación y otros símbolos.

A través de él se pueden representar hasta 256 caracteres utilizando una secuencia de 8 bits ($2^8=256$) que recibe el nombre de *byte*. Recuerden que 1 byte puede representar a un símbolo del lenguaje: letra, número o caracter.

Así, cuando se pulsa en el teclado la letra “A”, le corresponde una combinación de unos y ceros, que en código ASCII es 01000001.

Este código se utiliza en diferentes contextos, por ejemplo, cuando se escribe utilizando el programa Word, directamente los caracteres codificados en ASCII integran textos que son guardados en archivos constituidos por ese mismo código.

Por otro lado, cuando un programador edita un programa que ha desarrollado y lo tipea, queda como en el Word codificado en binario ASCII. Sin embargo, este código no es comprendido por la UC del procesador por lo que se debe llamar a un programa traductor, denominado Compilador, para que pase de binario ASCII a código binario denominado “código de máquina”, único código que “entiende” la UC.

También deben utilizarse programas de traducción para que los datos a procesar, que fueron tipeados y por lo tanto quedaron en memoria en código binario ASCII, sean consultados en números binarios naturales que puede operar la Unidad Aritmética Lógica.

Asimismo, si se pide visualizar mediante la impresora o el monitor algún dato, el proceso de traducción de códigos es inverso: los datos que se encuentran en binario natural o código de máquina son transformados en código binario ASCII por medio de un programa traductor (diferente al compilador) para que esos símbolos sean más familiares a nuestro entendimiento, al verlos en la pantalla del monitor.

En la actualidad existen otros intentos de codificación tratando de ampliar las características del código Binario ASCII. Entre ellos, se destaca el código **UNICODE**, cuya descripción se retomará en la bibliografía. De igual manera, el código **BCD** (decimal codificado en binario) es un intento para simplificar la conversión de sistemas binarios al decimal y viceversa.

Lo invitamos a seguir ahondando sobre las características y propiedades de los sistemas de codificación ASCII, UNICODE y BCD, a partir de la lectura de la bibliografía. No deje de realizar la ejercitación, le será útil para la resolución de los Trabajos Prácticos posteriores.



Lectura requerida

- Ginzburg, M.; Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag. 107-108.

Guía para la lectura

- ¿Para qué sirve el código ASCII?
- ¿Cómo es que un programa puede reconocer indistintamente que el nombre o sitio se escriba con mayúscula o minúscula?

Le proponemos realizar la siguiente actividad pues su elaboración le permitirá experimentar la tarea que realiza el compilador en el interior de la PC.

1.3. OPERACIONES DE LA UNIDAD ARITMÉTICO LÓGICA (UAL)

Recuerde que la Unidad Aritmético Lógica (UAL) **no ejecuta instrucciones**; simplemente, realiza las operaciones aritméticas o lógicas que le ordena la Unidad de Control (UC).

Repararemos ahora, únicamente, en las operaciones aritméticas de sumas y restas, focalizando en las particularidades de estas últimas. La idea es lograr que, por medio de la ejercitación, usted pueda comprender cómo la UAL realiza las operaciones con los datos.

Lo invitamos a realizar la lectura de la bibliografía para profundizar la comprensión de las operaciones de suma y resta con números binarios naturales.



Lectura requerida

- Ginzburg, M.; Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013. Pag. 97 a 112.

Guía para la lectura

- ¿De qué forma la UAL suma dos números?
- ¿Cómo efectúa una resta sin pedir prestado, mediante una suma?

Le proponemos realizar la siguiente actividad sugerida para ejercitar las operaciones aritméticas que realiza la UAL.



Trabajo práctico sugerido

Trabajo práctico N° 3: **Operaciones de la UAL**

Consulte la consigna de este trabajo en el link correspondiente del aula virtual.
Comuníquese con su tutor si se le presentan dudas en la resolución de la actividad.

1.4. CODIFICACIÓN DE INSTRUCCIONES Y DATOS EN MEMORIA, Y OPERACIONES ARITMÉTICAS DE LA UAL CON DATOS BINARIOS PROCESADOS EN UN MODELO DIDÁCTICO BÁSICO DE PROCESADOR.

En Ginzburg, M.; Cap. 1.2. *Estructura y operación de un modelo didáctico de computador binario*. En su: **La PC por Dentro**. 6ª ed.: Biblioteca Técnica Superior, 2013. Pags. 5 a 26 se plantea el funcionamiento de un modelo básico *representativo de cualquier procesador* que realiza automáticamente procesos de datos. Esto es, a partir de materia prima que son datos (números), mediante la ejecución de un plan lógico (programa) se obtienen, sin ingerencia humana, resultados (otros números) que constituyen información, como se planteó en la unidad 1.

La comprensión de este modelo **es fundamental** como base **para entender la arquitectura y el funcionamiento de cualquier sistema de procesamiento de datos** (sean mainframes, servidores, PCs, notebooks, tablets . . .), así como para abordar detalles de codificaciones y operaciones en los subsistemas que lo constituyen.

A medida que va analizando y verificando el modelo básico responda al siguiente cuestionario vinculado a la verificación de la asimilación de conocimientos y a su examen parcial:

¿Por qué en el interior de un computador sólo puede usarse el sistema binario?

¿Qué son las direcciones y los contenidos de la memoria principal?

¿Cómo se prepara el proceso de datos en el computador antes definido y cómo se le ordena a éste qué debe hacer?

¿Qué sería “alto nivel” y “bajo nivel” en la codificación realizada?

¿Qué 3 cosas tienen que ocurrir antes de ejecutar las instrucciones de un programa?

¿Podría ser que el cod-op, la dirección del dato y el valor de éste sean el mismo número? Justificar la respuesta explicando en general si son todos números cómo la UCP no confunde un cod-op con direcciones y con datos

¿Cuáles son las funciones de la UCP, la UC, la UAL y de los registros RDI, A, RI y PI ?

¿Cómo está constituida la MP, qué significa direccionar una celda, y cuántos bits guarda en un computador real?

¿Qué diferencias existen entre un calculador común y un computador y cómo se dan en ellos los procesos de datos?

¿Qué funcionamiento común tienen una calculadora y su visor con la UAL y el registro A?

¿Qué ocurre en los lugares de origen y destino cuando un número viaja de un lugar a otro de un computador?

¿Cuáles son las limitaciones de la UC en cuanto a la localización de números y a conocer el valor de los mismos?

¿Qué aspectos comunes tienen una caja musical con un computador?

¿Por qué se le debe dar al PI un valor inicial? ¿Cuál es éste valor, y qué sucede si no se le da el valor indicado?

¿Por qué las instrucciones de una secuencia deben estar en posiciones consecutivas y los datos no?

¿Cuánto debe aumentar el PI luego de ejecutar una instrucción?

¿Qué pasa si las instrucciones de la figura 1.4.a no están en posiciones consecutivas?

¿Qué sucede si debajo del código de operación se escribe, en lugar de la dirección del dato, el valor del dato?

¿Puede existir la instrucción 0010 1011? Justifique su respuesta.

¿Cuál es contenido informativo de una instrucción?

¿Cómo se expresa con palabras lo que ordenan cada una de las instrucciones de la figura 1.4.a ?

¿Qué pasa en la ejecución de I_1 , I_2 e I_3 si en MP se han omitido los valores de los datos ?

¿Qué significados puede tener un número binario que está en memoria. ?

¿Para qué sirven los pulsos reloj, y qué significa en relación con ellos que una UCP es de 900 MHz ?

¿Qué ciclo se realiza continuamente en el interior de un procesador y qué sucede en cada una de sus fases, y qué diferencia existe entre éste ciclo y los ciclos reloj ?

¿Para qué sirven las LC, de dónde salen y hacia donde van ?

¿Cómo las LC manejan la memoria, y qué diferencia existe entre la LC de L/E y la de 1/2. ?

¿Qué es el cod-op de una instrucción ? Explicar a partir del mismo su relación con las LC en la decodificación, y el concepto de que las instrucciones gobiernan y organizan lo que debe hacer el hardware.

¿Cuáles son los contenidos de la **RC**, qué es un μ código, cómo se localiza, y en qué instantes cambian de valor las LC ?

¿Por qué puede decirse que la **RC** transforma los bits de cada cod-op en una combinación binaria sobre las LC ?

¿Cómo se relacionan los MHz con los ciclos/seg y con los pulsos/seg. ?

¿En qué instante de cada ciclo los registros retienen nueva información. ?

1.5. TRES (3) TRABAJOS PRÁCTICOS CON CODIFICACIÓN DE INSTRUCCIONES EN CÓDIGOS DE MÁQUINA DE INTEL Ó AMD Y DATOS BINARIOS (NATURALES, ENTEROS Y REALES) ESCRITOS EN LA MEMORIA REAL DE CUALQUIER PC O NOTEBOOK, Y EJECUCIÓN DE DICHAS INSTRUCCIONES MEDIANTE EL DEBUG.

A continuación, explicaremos el funcionamiento del **Programa DEBUG** que permite realizar estos procedimientos a través del teclado. Este programa nos permitirá escribir, leer la memoria, visualizar los registros del procesador y ejecutar, paso por paso, cada instrucción; es decir, sacarle una “radiografía” a los puntos nodales del procesamiento.

Para lograrlo es preciso que comprenda previamente cómo debe suministrar las órdenes o instrucciones al procesador y de qué forma éste las guarda en la memoria. Por ello, lo invitamos a

realizar ahora la siguiente lectura que le ayudará a comprender el proceso de codificación y operación con datos que realiza una PC.

Tenga en cuenta que los ejercicios que van a plantearse de aquí en adelante requerirán el conocimiento del uso del Debug, que por cierto es sencillo.



Lectura requerida

- Ginzburg, M.; 1.3 Ejecución paso a paso de instrucciones en la PC usando el Debug. En su: **La PC por Dentro**. 6ª ed. Buenos Aires: Biblioteca Técnica Superior, 2013, pags. 35 a 40.

Guía para la lectura

- ¿Cómo se usa el DEBUG para escribir datos e instrucciones en la memoria?
- ¿Cómo encuentra la UC en memoria la primera instrucción y las siguientes de un programa a ejecutar, mediante el registro IP?
- ¿Quién se encarga de proporcionar la dirección de la primera instrucción de cada programa a ejecutar?
- ¿Cómo se cambia la dirección de instrucción que indica el IP?
- ¿Cómo puede visualizarse en el DEBUG la forma que se van procesando los datos, al ejecutarse las instrucciones de una PC?
- ¿Cómo ordenar que los códigos de máquina del proceso anterior sean ejecutados una tras otro automáticamente, conforme sucede realmente?

Una vez realizada la lectura que anteriormente le sugerimos, lo invitamos a visualizar la siguiente animación cuyo propósito principal es que usted pueda sistematizar las instrucciones para la utilización del programa DEBUG.



Material Multimedia Requerido

En el link correspondiente del aula virtual encontrará la presentación denominada **Instructivo del Programa DEBUG**.

Ahora si, lo invitamos a realizar esta actividad de facilitación que le permitirá comprobar la comprensión que usted ha alcanzado del funcionamiento del programa DEBUG. Recuerde que es muy importante que dedique tiempo a la realización de estas propuestas en tanto constituyen instancias propedéuticas o de preparación para poder abordar luego los trabajos prácticos.



Actividades para la facilitación de los aprendizajes

Le solicitamos que responda los siguientes interrogantes:

- ¿Qué debe estar en memoria antes de ejecutar un programa y cuál debe ser el valor del IP?
 - ¿Por qué las instrucciones de cada secuencia deben estar en posiciones consecutivas en memoria?
 - ¿Por que las variables no deben necesariamente estar una debajo de otra?
 - Durante la ejecución de un programa, ¿cómo se localizan las instrucciones mediante el IP si ocupan distinta cantidad de celdas en memoria? y ¿cómo se localiza el dato, que cada instrucción opera?
 - ¿Hay algún problema si la dirección presente en una instrucción coincide con su cod-op o con el cod-op de otra instrucción?
 - ¿Qué bits de la instrucción indican que el resultado debe ir a AX
-



Trabajos prácticos requeridos

Es tiempo de articular todo lo aprendido a través de la práctica y por ello lo invitamos a realizar los 3 (tres) Trabajos Prácticos planteados a continuación.

Como verificará, una vez que concretó el primero de ellos, los otros dos son muy similares de realizar en poco tiempo.

1.5.1 REALIZACIÓN EN LA PC DEL TRABAJO PRÁCTICO INTEGRADOR SOBRE DATOS QUE SON NÚMEROS NATURALES (MAGNITUDES) USANDO EL PRIGRANA DEBUG

Este trabajo práctico se basa en los números binarios naturales que ha estudiado y se divide en tres partes. Encontrará las consignas relativas a esta primera parte en el link correspondiente del aula virtual. Recuerde que estos trabajos son de entrega obligatoria. Consulte las fechas previstas en el Cronograma de la asignatura y comuníquese con su tutor si se le presentan dudas en su resolución.



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 113 a 117.

Hasta el momento, usted ha operado con variables que en lenguajes de alto nivel son “magnitudes”, o sea números naturales, que a nivel de máquina se representan en la memoria como binarios naturales.

Como en el interior de un computador sólo existen unos y ceros, no se pueden representar números enteros, que en el papel tienen signo menos, ni los que llevan coma. Para ellos existen convenciones para representarlos y operarlos.

Los números enteros en base diez, como usted sabe, son números negativos o positivos. Su magnitud es un número natural y se simbolizan en la pantalla o en el papel con los signos (+) y (-). En la memoria de la máquina se representan como binarios con un bit de signo como bit extremo izquierdo. Este bit es 0 para los positivos ó 1 para los negativos, pero para estos últimos no se trata simplemente de colocar un uno a la izquierda.

Lo invitamos ahora a realizar la siguiente lectura que le permitirá profundizar sobre la representación de números enteros. Es importante que, además, realice los ejercicios que allí se proponen.



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 119 a 149.

Guía para la lectura

- ¿Quién opera los números naturales y los enteros?
- ¿Dónde se operan los números en punto flotante?
- ¿Cómo un computador compara dos números?
- ¿Cuáles son los valores máximos y mínimos representables para n bits:
 - a) para naturales,
 - b) para enteros.

Le proponemos que realice ahora la siguiente actividad. Recuerde que es muy importante que dedique tiempo a la realización de estas propuestas en tanto constituyen instancias propedéuticas o de preparación para poder abordar luego la resolución de los trabajos prácticos.



Actividades para la facilitación de los aprendizajes

Realice los ejercicios planteados en el apartado N.4 del Complemento de la Unidad 1 del texto de M. Ginzburg, que figura en la página 185, (CU1-15 y CU1-16).

(Preste particular atención a los ejercicios del 17 al 20, por su importancia cognitiva, los cuales además se utilizarán más adelante)



Tenga en cuenta que los trabajos que produzca durante los procesos de estudio son insumos muy valiosos y de preparación para la Evaluaciones Parciales. Por lo tanto, guarde sus notas, apuntes y resoluciones; le serán de mucha utilidad. Por otro lado considere que las dudas pueden ser consultadas a través de los medios disponibles para realizar las Tutorías.

También para que la UC pase de la ejecución de una secuencia en función de un resultado anterior es importante conocer si el mismo fue positivo, cero o negativo, si su valor excede la capacidad del procesador o si se debe tener en cuenta el “acarreo” de un “1” fuera del formato del número. Estas indicaciones se conocen con el nombre de **flags** y son generados por la Unidad Aritmético Lógica del procesador.

La lectura que le proponemos realizar aquí le brindará información para comprender y profundizar el estudio acerca de la forma en que la UAL realiza las operaciones lógicas.



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 150 a 164.

Guía para la lectura

- ¿Quién genera los Flags y dónde se guardan sus valores?
- Ejemplificar para qué sirven los Flags en una instrucción de salto, cuando la UC debe tomar una decisión.
- Indicar para una suma y una resta las conclusiones que resultan de los valores de SZVC, para enteros y naturales.

Lo invitamos a realizar la siguiente actividad de facilitación que le brinda la oportunidad de poner en práctica los contenidos estudiados hasta aquí. Recuerde que es muy importante que dedique tiempo a la realización de estas propuestas en tanto constituyen instancias de preparación para poder abordar luego los planteos realizados en los trabajos prácticos.



Actividades para la facilitación de los aprendizajes

Realice los ejercicios planteados en el apartado N5.1 del Complemento de la Unidad 1 del texto de M. Ginzburg, CU1-23 y CU1-24.



Tenga en cuenta que los trabajos que produzca durante los procesos de estudio son insumos muy valiosos y de preparación para la Evaluaciones Parciales. Por lo tanto, guarde sus notas, apuntes y la resolución de los ejercicios; le serán de utilidad.

Por otro lado considere que las dudas pueden ser consultadas en los Foros o en el espacio de Tutoría.

Llegados a este momento del estudio le solicitamos la realización de la segunda parte del Trabajo Práctico requerido. Recuerde que este material tiene fecha de entrega y su presentación en tiempo y forma son consideradas en la evaluación de la tarea realizada. Consulte el Cronograma de la Asignatura.



Trabajo práctico requerido

Trabajo práctico Nº 5.B: Integrador sobre datos numéricos que son enteros

Con esta entrega finaliza la resolución de trabajo práctico requerido. Encontrará las consignas en el link correspondiente del aula virtual. Consulte las fechas previstas en el Cronograma de la asignatura y comuníquese con su tutor si se le presentan dudas en su resolución.



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 165 a 170.

Del mismo modo que los números Enteros se representan sin utilizar los signos (+) y (-), los números Reales (enteros positivos y negativos, fraccionarios e irracionales) se representan en la memoria con la convención de “punto flotante” (coma flotante).

Punto flotante es una convención de que permite operar con números reales dentro de un amplio rango de valores, tengan o no coma, sean positivos o negativos. Se expresa mediante una notación científica a fin de representar con pocos bytes números grandes y pequeños e magnitud y cantidad de cifras

Lo invitamos a realizar la siguiente lectura para profundizar su conocimiento acerca de la forma en que opera la UAL con números reales.



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 180 a 186.



Trabajo práctico requerido

Trabajo práctico N° 5.C: **Integrador sobre datos numéricos que son reales**



Lectura requerida

- Ginzburg, M. En su: **La PC por Dentro**. 6ª ed. Buenos Aires; Biblioteca Técnica Superior, 2013. Cap. 1.9. Codificaciones empleadas en Computadoras y Periféricos. Pags. 113 a 117.

CIERRE DE LA UNIDAD

Usted está en condiciones de realizar el problema inverso que ha efectuado en el Trabajo Práctico N° 5A y N° 5B.

1) Le proponemos deducir qué números en base diez originaron la siguiente operación en números binarios y qué resultado se obtiene haciendo la operación correspondiente.

a) suponiendo que son naturales.

b) suponiendo que son enteros.

$$\begin{array}{r}
 1001 \\
 + \quad 0111 \\
 \hline
 1 \\
 \hline
 10001
 \end{array}$$

1 2) Una vez realizadas las operaciones solicitadas en el punto 1, determine los flags que generaría la Unidad Aritmético Lógica. En el caso que no se conocieran los números determinados en 1, establecer en función de los flags si el minuendo es $>$, $<$ o $=$ que el sustraendo, y si el resultado es correct



Evaluación Parcial

Propuesta para la Integración del Módulo I

En esta primera Evaluación Parcial de la asignatura usted deberá desarrollar consignas de trabajo que apuntan a la comprensión de los principios básicos del procesamiento de datos.

Durante el desarrollo de la propuesta ponga en juego sus conocimientos, revise el material y no olvide revisar los criterios que se tomarán en cuenta para su evaluación, tratando de ajustar su producción a ellos.

Este trabajo tiene fecha de comienzo y finalización. Consulte el Cronograma de actividades de la Asignatura para realizar su entrega en tiempo y forma, ya que es de carácter obligatorio.

Fin del Módulo I

1. Teniendo presente la sucesión de números hexadecimales, indicar cuál es el siguiente de los siguientes números:
FFF, 2ABF, 3B99, 1FF, ABCDE, AFF9, C0D0, A0F, 999
2. Dado el número 10 en el sistema en base doce, indicar qué número es en base diez.
Generalizar para el número 10 en base r , qué número es en base diez.
3. Dado el número binario 1111111, hallar una forma rápida de pasarlo a decimal, sin tener que hallar el peso decimal de cada bit y luego sumar los pesos. Generalizar el procedimiento hallado.
4. Es válida la siguiente regla de conversión rápida de binario a decimal: "hallar qué número decimal son los cuatro últimos bits de la derecha del número, y a partir del quinto bit (peso 16) sumarle al número hallado los pesos decimales donde el número binario presente unos"
6. Convertir a hexadecimal el número 11000010010001_2 . Luego convertirlo a octal.
9. Dado el número 10000000000 en base dos, expresarlo como 10^n en esa base, determinando n .
El mismo número en base dieciseis, expresarlo como 10^n en esa base, determinando n .
Idem en base diez, expresarlo como 10^n en esa base, determinando n .
10. ¿ Cuántos bits hacen falta para representar números decimales entre 0 y 10^6 ?
11. El número binario 10000000000, expresarlo como 2^k en base diez,
12. Siendo que un número binario negativo con bit de signo (complemento al módulo de otro número binario N de n dígitos) es $10111001 = -N + 10^n$ hallar N realizando la operación de resta $10^n - 10111001$ por el método convencional de "pedir prestado". Luego determinar N por el método de

invertir los bits del número dado y sumar uno. Justificar este método para el presente caso. Explicar por qué no se usa restar uno y luego invertir los bits.

13. ¿Cuántos bytes hacen falta para representar en binario números decimales entre -32768 y 32767 ?
14. Para un formato de 4 bytes: ¿cuál es la máxima magnitud que puede representarse, y cuál el mayor número positivo entero? Idem el número negativo de mayor magnitud.
15. Representar el número binario negativo 10110010 en formato 16.
16. Realizar en módulo 10^5 las siguientes operaciones en base diez con números enteros usando dígito de signo con la convención del complemento al módulo. Verificar que los resultados con dígito de signo coincidan con los que se obtendrían con el símbolo de signo corriente.
a) $96 + 180$; b) $96 + (-180)$ c) $-96 + 180$; d) $-96 + (-180)$
17. Para el ejercicio b) anterior justificar por qué al realizar una suma de naturales también se puede efectuar una suma de enteros.
18. Efectuar las siguientes sumas algebraicas dadas en base diez en base dos para números con bit de signo, en el menor formato posible, indicando SZVC. Verificar si el resultado binario convertido a decimal concuerda con el esperado haciendo las operaciones en base diez.
a) $18 + 102$ b) $18 + (-102)$ c) $-18 + 102$ d) $-18 + (-102)$
19. Idem anterior para las sumas $30 + 102$ y $-30 + (-102)$
Si no resultara la verificación, explicar por qué no concuerda, y determinar en qué formato mínimo es posible realizar correctamente cada operación. Verificar que es posible realizarla en formato 16.
20. En función del problema anterior explicar por qué se afirma que el overflow es un error por truncamiento que por definición sólo puede darse en números signados.
21. Efectuar las siguientes restas algebraicas dadas en base diez en base dos, en el menor formato posible, indicando SZVC. No es factible hacer "menos por menos = mas", etc. Verificar si el resultado binario convertido a decimal concuerda con el esperado haciendo las operaciones en base diez.
a) $18 - 102$, b) $18 - (-102)$ c) $-18 - 102$ d) $-18 - (-102)$

22. Efectuar en formato 8 la resta entre números decimales $150 - 37$ representado los mismos como binarios **naturales** (magnitudes), indicando SZVC. Verificar si el resultado binario convertido a decimal concuerda con el esperado haciendo las operaciones en base diez.

Si con los mismos números se intentara realizar $37 - 150$ (operación prohibida entre naturales), indicar una forma de conocer la magnitud de la diferencia.

¿ Se podría haber hecho las operaciones en formato 8 si los números hubiesen sido signados?

24. Para el problema 20, justificar para el caso d) por qué los binarios con bit de signo se pueden sumar como si fueran naturales (magnitudes)

25. Explique la diferencia entre los indicadores **V** y **C**. Ejemplificar numéricamente para bases diez y dos. Indicar para qué tipos de números se usa cada uno

- Comprender, sin conocimientos de electrónica o electricidad, sobre la base de un modelo general didáctico de un computador, los principios de su funcionamiento, sea PC, Servidor, Notebook, Tablet u otro, así como la relación inseparable entre hardware y software. Para ello se deben codificar en binario instrucciones básicas y los datos que ellas procesarán en direcciones de memoria, y ejecutar dichas instrucciones mediante el programa Debug, para verificar que ocurren los resultados esperados de la ejecución de cada una.