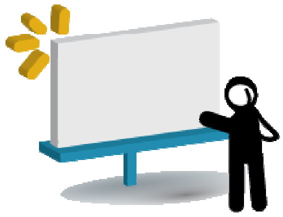
FUNDAMENTOS DE LA PROGRAMACIÓN

Centro de e-Learning SCEU UTN - BA.
Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148
www.sceu.frba.utn.edu.ar/e-learning

MÓDULO 1 - UNIDAD 2

Lógica Computacional



Presentación:

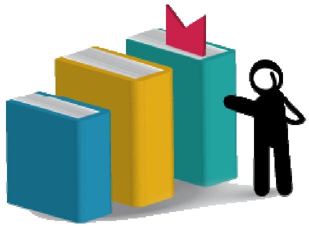
En esta Unidad nos centraremos a algunos aspectos que resultan claves para explicar algo de los visto en la Unidad anterior y mucho de lo que veremos las próximas Unidades. Vamos a empezar a hablar sobre la "forma de pensar" del programador. Esta es una idea sobre aquellas habilidades, conocimientos y capacidades que un programador debe tener, y que conjuga conceptos de lógica, álgebra y otras ramas de la matemática. Temas que podrán parecer desconectados y abstractos se irán conectando e interrelacionando en las próximas Unidades a partir de lo que veamos en esta Unidad.

Porque todo tiene un porqué y punto de partida, lo visto en esta Unidad será el punto de partida para aprender a programar, para comenzar a "pensar como programador".

Objetivos:

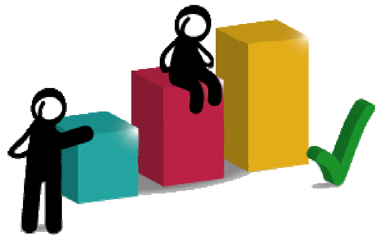
Que los participantes:

- Incorporen los aspectos principales de la lógica matemática
- Comprendan la aplicación y uso de las proposiciones y las operaciones lógicas
- Incorporen los aspectos principales del álgebra y su aplicación en las expresiones
- Comprendan los ejemplos resueltos que se presentan
- Incorporen conceptos que serán profundizados en las próximas unidades, como las matrices



Bloques temáticos:

1. Lógica Matemática
2. Proposiciones
3. Operaciones lógicas
4. Algebra
    - 4.1. Expresiones y variables
    - 4.2. Expresiones con dos variables
    - 4.3. Ejemplos
5. Reglas de Inferencia
    - 5.1. Modus ponendo ponens
6. Sistemas de numeración
    - 6.1. Sistemas de numeración posicional
    - 6.2. Relaciones entre conjuntos de tipos numéricos
7. Representaciones de números en distintas bases
    - 7.1. Conversión de cualquier base a base 10
    - 7.2. Conversión de decimal a otras bases
    - 7.3. Casos particulares
8. Matrices



Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

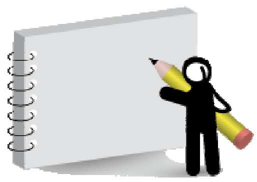
- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

* El MEC es el modelo de E-learning colaborativo de nuestro Centro.



Tomen nota

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

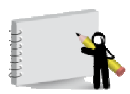
1. Lógica Matemática

La lógica matemática es una variedad de la lógica filosófica, permite el análisis del razonamiento, trata el estudio y análisis de los métodos de razonamiento o argumentación. La lógica centra su análisis en la relación entre los enunciados, pero no en el contenido de los mismos, es decir no nos permite determinar si los enunciados son verdaderos o falsos, pero partiendo de su vedad o falsedad nos permite determinar si las conclusiones son verdaderas o falsas.

Ejemplo:

- **Enunciado 1**: Todos los alumnos de Matemática recuerdan los conceptos adquiridos en la secundaria.
- **Enunciado 2**: En la secundaria se adquirió el concepto de conjunto.
- **Por lo tanto**: Todos los alumnos de Matemática comprenden el concepto de conjunto.

Si consideramos que los enunciados son verdaderos, la conclusión es verdadera.



La lógica no se centra en el análisis de los enunciados (no sabe si son verdaderos o falsos) pero partiendo de su afirmación o negación podemos decir si la conclusión es correcta o incorrecta (verdadera o falsa).

Los objetivos principales de la lógica son esencialmente:

- Eliminar las ambigüedades propias del lenguaje ordinario.
- Dar rigor a aquello que se está estudiando.

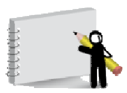
¿Por qué es esto importante en la programación? Porque las operaciones lógicas son parte fundamental de todo desarrollo de software, ya que constantemente tenemos que decidir qué hacer ante diferentes estímulos que recibe nuestro programa. Para esto será importante comprender como plantearnos preguntas en forma abstracta para luego llevarlas a un lenguaje de programación para crear nuestro software.

2. Proposiciones

Como se vio en el ejemplo del punto anterior, el razonamiento parte de lo que llamamos "enunciados". Estos enunciados son la base de la lógica y reciben el nombre de **proposiciones**.

Por lo general son afirmaciones y no instrucciones o preguntas (que es justamente lo que hacemos cuando programamos: **indicar acciones y hacernos preguntas**), y pueden ser verdaderas o falsas pero no tener una ambigüedad.

Proposición:



Se entiende como proposición o enunciado a una oración declarativa sin ambigüedad, que es verdadera (V) o falsa (F), pero nunca las dos cosas simultáneamente.

La veracidad o falsedad de un enunciado se llama su "Valor de Verdad". Los términos verdadero o falso se consideran como atributos de una proposición.

Por lo general se utilizan letras minúsculas para representar las preposiciones (Ej.: p, q, r, s, t, etc.).

Consideramos proposiciones validas aquellas que carecen de ambigüedad (son verdaderas o falsas, pero no ambas).

Vemos algunos ejemplos de proposiciones válidas y no válidas, y se explica por qué algunos enunciados no son proposiciones. Las proposiciones se indican por medio de una letra minúscula, dos puntos y la proposición propiamente dicha.

Ejemplos:

p: La luna es un planeta.

q: $(17)^3 = 21$

r: $x = y+12$

w: Hola ¿cómo estás?

t: Camina dos pasos hacia la derecha.

- **p** y **q** sabemos que pueden tomar un valor de falso o verdadero, pero no
- ambos, por lo tanto son proposiciones válidas.
- **r** también es una proposición valida, aunque el valor de falso o verdadero depende del valor asignado a las variables **x** e **y** en determinado momento.
- **t** y **w** no son válidos, ya que no pueden tomar un valor de falso o verdadero

Como ya dijimos la lógica centra su atención en la relación entre los enunciados (proposiciones) y no en el contenido de los mismos.

Las proposiciones resultado de las relaciones entre proposiciones simples, se denominan **proposiciones compuestas**.

3. Operaciones lógicas

Disyunción: Dadas dos proposiciones **p** y **q**,

Se llama disyunción de **p** y **q**, y se escribe:

p ˅ q

Y se la lee como "**p** o **q**".

Definición: p ˅ q es verdadera cuando al menos una de ellas lo es. En otras palabras, solamente será falsa cuando ambas proposiciones sean falsas.

Ejemplo:

- **p**: Tengo un piloto para la lluvia
- **q**: El día esta radiante
- **p ˅ q** : “Salgo si” tengo un piloto para la lluvia o el día esta radiante

Conjunción: Dadas dos proposiciones **p** y **q**,

Se llama conjunción de **p** y **q**, y se escribe:

p ˄ q

Y se la lee como "**p** y **q**".

Definición: p ˄ q es verdadera cuando p y q son verdaderas simultáneamente.

Ejemplo:

- **n**: Tengo un piloto para la lluvia
- **q**: El día esta radiante
- **p ∧ q**: “Salgo si” tengo un piloto para la lluvia y el día esta radiante

<u>Negación</u>: Dada una proposición p, se llama negación de p, y se escribe ¬p a la afirmación, y se dice "no p".

Definición: ¬p es verdadera cuando p es falsa (y viceversa)

Ejemplo:

- **p**: Tengo un piloto para la lluvia
- **¬p**: “No” tengo un piloto para la lluvia

- **q**: El día esta radiante
- **¬q**: El día “No” esta radiante

ACTIVIDAD 1:

Tema de reflexión e investigación: ¿Porqué las OPERACIONES LÓGICAS que se destacan son estas tres: "y", "o" y "no"?

Comparta sus respuestas en el foro de actividades de la Unidad.

4. Algebra

El álgebra tiene probablemente su origen más bien "formal" en el libro "Compendio de cálculo por compleción y comparación", del matemático del siglo 8 o 9 llamado Al-Juarismi, que vivió en una región cercana a la actual Bagdad. Si bien existen registros del año 2000 AC sobre trabajos en álgebra en Babilonia, se considera al documento de Al-Juarismi como la base del álgebra moderna y a su autor como el "padre del álgebra", aunque no es posible dejar de lado otras grandes contribuciones de matemáticos importantes, como Diofanto de Alejandría o Brahmagupta de India.

Podemos definir el Algebra como "recuperación / compleción", literalmente. El concepto de "compleción" se refiere al sentido de completo o completitud que pudiera tener algo.

El álgebra se basa en el planteo de soluciones para determinas incógnitas. En este sentido, las incógnitas suelen representarse como letras, como por ejemplo:

$x + 4 = 5$

En un sentido práctico, el uso de letras se trata de una convención, ya que la misma expresión tendría le mismo significado que las siguientes:

__ + 4 = 5

? + 4 = 5

Siendo estrictos, la forma de trabajo sería la misma en cualquiera de los casos, aunque, nuevamente, por convención (además de claridad, facilidad de comprensión, entre otros factores) se asume el uso de letras como estándar.

4.1. Expresiones y variables

Las **variables son valores que pueden cambiar de un momento a otro**, ya que valor depende de factores externos o internos que no pueden ser determinados exactamente en forma predeterminada.

Para ejemplificarlo, supongamos el caso de un mozo, al que se la paga por día una cantidad fija como sueldo y a lo que se suman las propinas que recibe.

Veamos el flujo de caja de este mozo de la semana pasada (trabaja de domingo a viernes):

Día	Sueldo	Propinas
DOM	100	450
LUN	100	50
MAR	100	100
MIE	100	100
JUE	100	150
VIE	100	250

Podemos ver que, siendo el domingo el día en el que atiende a más comensales es cuando recibió la mayor cantidad de dinero en propinas. El lunes, día en que se acercan pocos clientes es cuando menos recibe, pero va a aumentando en la semana a medida que se aproxima el fin de semana.

Pero existen casos particulares, por ejemplo, los domingos del Día de la Madre y el Día del Padre, cuando se duplica la cantidad de clientes y sus propinas aumentan. Y, por el contrario, están los días festivos y feriados, en los cuales el restaurante cierra sus puertas, por lo que recibe su sueldo, pero no obtiene propinas.

En este ejemplo podemos identificar que el **sueldo es una constante**, ya que a través del período de tiempo analizado su valor no cambia. Por el contrario, las propinas que obtiene fluctúan en el tiempo debido al día de la semana, si es un día conmemorativo o festivo, entre otros factores. **Por esto es que decimos que las propinas son variables**, dado que ***NO*** se pueden determinar por adelantando qué valor tendrán.

Estas variables son las que en álgebra se representan con letras, como vimos antes.

En la programación, el concepto de "variable" y "constante" es muy importante, por lo que resulta interesante que se comprenda el origen del dicho término y su aplicación práctica en álgebra, ya que la programación lo toma de allí.

Siguiendo con el ejemplo, podemos representar la situación económica del mozo a través de la siguiente expresión:

s = 100 + p

Donde s es el sueldo que recibe y p la cantidad de propinas en dinero. Esta expresión representa los ingresos diarios del mozo, y aplica para cualquier día de la semana.

Evaluemos a continuación el siguiente caso:

“Una sociedad de fomento está vendiendo un colecta de fondos a través de la venta de "rifas". El costo de cada número está dado por la expresión 2 + (3*x), donde x es la cantidad de números”.

Evaluemos ahora el costo de adquirir 4 rifas o 6 rifas. Para esto, deberemos sustituir la cantidad de números (x) por estas cantidades, comenzando por el primer caso.

x = 4

s = 2 + (3 * x)

s = 2 + (3 * 4)

s = 2 + (12)

s = 14

El segundo caso, comprando 6 rifas sería así:

x = 6

s = 2 + (3 * x)

s = 2 + (3 * 6)

s = 2 + (18)

s = 20

¡¡RECUERDEN!!

Es común encontrar que el signo "*" (multiplicación) no se agrega en forma explícita en la expresiones, por lo que:

4 * x

Es equivalente a expresar:

4x

Supongamos un caso más práctico, en el cual estamos leyendo un libro de origen norteamericano, en el cual nos encontramos con una situación en el que los protagonistas se encuentran escalando una montaña y, en el texto, nos dice que la temperatura es de 41 grados Fahrenheit. Para el que no tienen en claro la conversión podría resultar frustrante no saber cuánto representan los grados Fahrenheit (f) en los grados Celsius (c) que nos resultan más naturales.

Esto se podría solucionar sabiendo que la expresión para convertir grados f a grados c es la siguiente:

$$c = (f - 32) * 5/9$$

Aplicamos la sustitución de variables en la expresión dada y hacemos las cuentas:

$$c = (41 - 32) * 5/9$$

$$c = (9) * 5/9$$

$$c = 5$$

Finalmente, llegamos a la conclusión que los protagonistas se encontraban en la montaña a 5^0 Celcius. Hacía bastante frío después de todo… ☺

4.2. Expresiones con dos variables

Supongamos que nos encontramos con la siguiente expresión:

a + b

Y que tenemos que evaluar dicha expresión para los valores

a = 2

b= 3

Lo que hacemos a continuación, es sustituir los valores y realizar los cálculos:

a + b

2 + 3

= 5

De esta forma estamos resolviendo una expresión con dos variables.

Supongamos otro ejemplo:

5x + 3y

Si evaluamos la expresión para los valores:

x = 3

y = -2

Reemplazamos y realizamos los cálculos:

5(3) + 3(-2)

(Recuerden que esta expresión es equivalente a 5*(3) + 3*(-2))

15 + (-6)

= 9

4.3. Ejemplos

a) Evaluar la expresión $a^2 + 5b - 6$ con a = 2 y b = -3
b) Evaluar la expresión $3n^1 - 4n^0$ con n=5
c) Evalúa la expresión 6a + 3b - 5 con a = 7y b = 8
d) La suma de -4 y la cantidad de 3 veces x
e) Toma la cantidad -5 veces x, y entonces suma uno
f) -2 más el producto de -2 y x

a)

$= 2^2 + 5(-3) - 6$

$= 4 + (-15) - 6$

$= 4 - 21$

$= -17$

b)

$= 3n^1 – 4n^0$

$= 3(5^1) – 4(5^0)$

$= 3(5) – 4(1)$

$= 15 – 4$

$= 11$

c)

$= 6a + 3b - 5$

$= 6(7) + 3(8) - 5$

$= 42 + 24-5$

$= 61$

d)

$= -4 + 3x$

e)

$= -5x + 1$

f)

$= -2 + (-2x)$

$= -2 -2x$

5. Reglas de Inferencia

Las reglas de inferencia son muy simples. Para comprenderlas más fácilmente sirve una analogía: se las puede comparar con el esfuerzo equivalente a aprender las reglas de un juego de mesa. En este caso, el juego se compone de proposiciones o fórmulas lógicas, nombre que tienen las *proposiciones simbolizadas*.

El punto inicial se basa en conjuntos de fórmulas que se denominan *premisas*. El objetivo del juego será utilizar las reglas de inferencia de forma que estas conduzcan a otras fórmulas que se denominan *conclusiones*. La conexión entre premisas y conclusiones será una *deducción*.

premisa → deducción → conclusión

La conclusión que se obtiene es una consecuencia lógica de las premisas, si cada paso que se da para llegar a la conclusión está permitido por una regla. El concepto de inferencia se puede resumir de la siguiente forma: **de premisas verdaderas se obtienen sólo conclusiones que son verdaderas**. Dicho de otra forma, si las premisas son verdaderas, entonces las conclusiones que se derivan de ellas lógicamente, serán verdaderas.

Veamos algunos ejemplos de inferencia (dado que un método efectivo para aprender un juego es a través de ejemplos de jugadas): es de suponer que se tienen dos premisas, la fórmula $P \rightarrow Q$ (que se lee "p entonces q") y la fórmula P. Estas premisas están dadas; es decir, se empieza diciendo que se ha dado P y que se ha dado $P \rightarrow Q$. ¿Se puede sacar una conclusión de estas dos proposiciones? Es decir, ¿se puede idear otra proposición que haya de ser cierta si las premisas son ciertas? La conclusión es clara si se leen las premisas en la forma:

Si P entonces Q, y P.

La primera proposición expresa que si se verifica P, entonces se verifica Q, y la segunda dice que se verifica P. La conclusión es que se verifica Q. La proposición Q es consecuencia lógica de las premisas, P y P → Q.

Veamos otro ejemplo, con una inferencia de la misma forma, pero cuyo contenido se ha suplido por lenguaje corriente. La primera premisa es:

Si está soleado, entonces el cielo debe estar despejado.

La segunda premisa es:

Está soleado.

¿Qué conclusión se puede sacar de las dos premisas?

La respuesta es la conclusión "El cielo está despejado". Esta conclusión se puede inferir lógicamente de las premisas dadas.

A continuación veremos una regla particular de inferencia que permite deducir esta conclusión de las premisas.

5.1. Modus ponendo ponens

La regla de inferencia que analizaremos tiene un nombre latino: modus ponendo ponens.

Veamos algunos ejemplos del uso de esta regla en la deducción de conclusiones a partir de premisas.

Premisa 1. Si él está en clase, entonces él está en la escuela.

Premisa 2. Él está en clase.

Conclusión. Él está en la escuela.

Otro ejemplo del uso del *modus ponendo ponens* es el siguiente:

Premisa 1. Si hace frío, entonces el agua se congelará.

Premisa 2. Hace frío.

Conclusión. El agua se congelará.

Simbólicamente, el primer ejemplo se expresa así:

Sea:

P= "Él está en clase"

Q= "Él está en la escuela"

Entonces

Premisa 1. P → Q

Premisa 2. P

Conclusión Q

La regla de inferencia llamada ***modus ponendo ponens*** permite demostrar Q a partir de $P \rightarrow Q$ y P.

El segundo ejemplo se simboliza de la manera siguiente, donde P es la proposición “Hace frío” y Q es la proposición “El agua se congelará”.

$\neg P \rightarrow \neg Q$

$\neg P$

$\neg Q$

En todos los ejemplos vistos, la regla *modus ponendo ponens* **permite pasar de dos premisas a una conclusión**. Afirmar que la conclusión es consecuencia lógica de las premisas, es decir, que siempre que las premisas son verdaderas, la conclusión es también verdadera. **La regla de inferencia aprendida dice que si se tienen dos proposiciones de la forma P → Q y P, se puede deducir la conclusión Q.**

La segunda premisa afirma el antecedente, que es ¬P. Por tanto, el consecuente, que es ¬Q, se sigue de la regla *modus ponendo ponens*. En todos los ejemplos que se dan a continuación se aplica el *modus ponendo ponens*.

a)

$R \rightarrow S$

R

S

b)

P

$P \rightarrow \neg Q$

$\neg Q$

c)

P & Q → R

P & Q

R

d)

¬P → Q

¬P

Q

e)

P → Q & R

P

Q & R

Observen que en el segundo ejemplo, que la condicional figura en segundo lugar, y P, que es precisamente el antecedente, está situado primero. Cuando el *modus ponendo ponens* o cualquiera de las otras reglas se aplica para sacar una conclusión de dos o más proposiciones, el orden de aquellas proposiciones es indiferente.

Recuerden que una condicional (una sentencia con forma de cuestionamiento o pregunta) se puede escribir (P) → (Q). Con los paréntesis, el *modus ponendo ponens* es:

(P) → (Q)

(P)

(Q)

Si es una ayuda, se pueden usar paréntesis cuando el antecedente o el consecuente son proposiciones moleculares, como en los tres últimos ejemplos anteriores o en el siguiente:

¬P v R → S & ¬Q	**(¬P v R) → (S & ¬Q)**
¬P v R	**(¬P v R)**
S & ¬Q	**(S & ¬Q)**

El nombre *modus ponendo ponens* se puede explicar de la siguiente manera: Esta regla de inferencia es el método (*modus*), que afirma (*ponens*) el consecuente, afirmando (*ponendo*) el antecedente.

Tener en cuenta esta clase de planteo lógicos facilita enormemente las tareas de programación. Pónganse por un momento en el papel de programador, y supongamos que están trabajando en un banco y reciben una solicitud ("requerimiento funcional") para modificar un sistema de dicho banco, en el cual, a partir de determinada fecha por Resolución del Banco Central sólo se podrá hacer movimientos entre cuentas propias si monto supera los $100.

Planteemos esta situación según el modus ponendo ponens en dos casos:

Caso 1: Permitir transacción

P → Q. Si el cliente tiene al menos $100 de saldo, entonces podrá hacer la transacción.

P. El cliente tiene $143 de saldo.

Q. El cliente puede hacer la transacción.

Caso 2: Rechazar transacción

P → Q. Si el cliente tiene al menos $100 de saldo, entonces podrá hacer la transacción.

¬ P. El cliente tiene $97 de saldo.

¬ Q. El cliente no puede hacer la transacción.

Como podemos ver en ambos casos, el hecho de plantear una situación abstracta o un problema en forma ordenada, siguiendo ciertas "reglas del juego", ayuda a comprender mejor la situación o problema a resolver. No significa que los programadores estén constantemente haciendo esta clase de planteos ni pensando "acá conviene aplicar el modus ponendo ponens porque nos va facilitar la...". No, eso ocurre en la práctica, porque los programadores realizan esta clase de planteos muchas veces sin saberlo o sin ser conscientes de lo que están haciendo, ya que lo tienen interiorizado. Pero interiorizarlo en poco tiempo, en un curso breve como este, requiere comprender cuáles son los mecanismos y cuáles son las "fórmulas mentales" que usan los programadores en su día a día y que, probablemente, les haya costado mucho tiempo incorporar. Simplemente porque no se detuvieron a **pensar cómo deben pensar**.

Esto, de nuevo, es parte de lo que llamamos en este curso "la forma de pensar" del programador.

ACTIVIDAD 2:

¿Qué otros ejemplos de proposiciones utilizando el *modus ponendo pones* se les vienen a la mente?

- Posteen al menos 1 y cómo máximo 3 ejemplos.
- Cada uno debe abrir un tema nuevo en el foro.
- En ese tema envíen todos los ejemplos en el mismo mensaje (NO hacer un posteo por cada ejemplo).
- Los que se animen, propongan mejoras o correcciones a las proposiciones publicadas de sus compañeros.
- Respeten la nomenclatura y formas vistas en el material.

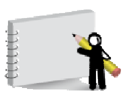
Comparta sus respuestas en el foro de actividades de la Unidad.

6. Sistemas de numeración

La forma de escribir los números usada en la actualidad procede de sistemas de numeración de la India. Estos sistemas ofrecían dos ventajas sustanciales respecto a los usados en Europa:

1. El concepto del **número 0**, que, aunque probablemente fue importado de las culturas mesopotámicas, se integró por primera vez en un sistema decimal junto con las otras nueve cifras del sistema.
2. **Valor posicional** a cada cifra, de manera que un mismo símbolo tiene un valor diferente según su posición en la expresión numérica (compárenlo con el sistema de numeración Romano)

Este sistema fue adoptado por los árabes antes del siglo IX, y popularizado en el primer manual de aritmética inspirado en el sistema decimal posicional.



Sistemas de numeración:

Un sistema de numeración puede definirse como un conjunto de signos, relaciones, convenios y normas destinados a expresar de modo gráfico y verbal el valor de los números y las cantidades numéricas.

6.1. Sistemas de numeración posicional

En él se tienen en cuenta los siguientes aspectos:

- **La base del sistema** es la cantidad de símbolos con que se cuenta. Por ejemplo, la base 8 (octal) agrupa ocho símbolos, mientras que la base 10 (decimal) agrupa diez símbolos.

- **Los símbolos del sistema**, o cifras elementales que se utilizan. En el sistema decimal, se usan los símbolos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. En cambio, por ejemplo, en el sistema hexadecimal se emplean el 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F.
- **Las normas de combinación** de los símbolos para formar los números. Según ello, a cada cifra 0 símbolo se le asocian dos propiedades:
  - Su valor absoluto intrínseco y
  - Su valor posicional o relativo, que depende de la posición que ocupa
  - En la cantidad numérica.

Dado un numero **n** escrito como la sucesión de numerales

$$a_n , a_{n-1} , \dots , a_2 , a_1 , a_0 \text{ en la base } b$$

Puede descomponerse en forma polinómica del modo siguiente:

$$n_b = a_n , a_{n-1} , \dots , a_2 , a_1 , a_0$$

$$n_b = a_0 * b^0 + a_1 * b^1 + a_2 * b^2 + \dots + a_{n-1} * b^{n-1} + a_n * b^n$$

En esta forma polinómica se ve bien el carácter posicional, cada símbolo (a_0, a_1,...,a_n) esta multiplicado por un valor que depende de la posición en el número, este número es la base elevada a la posición:

Ejemplo:

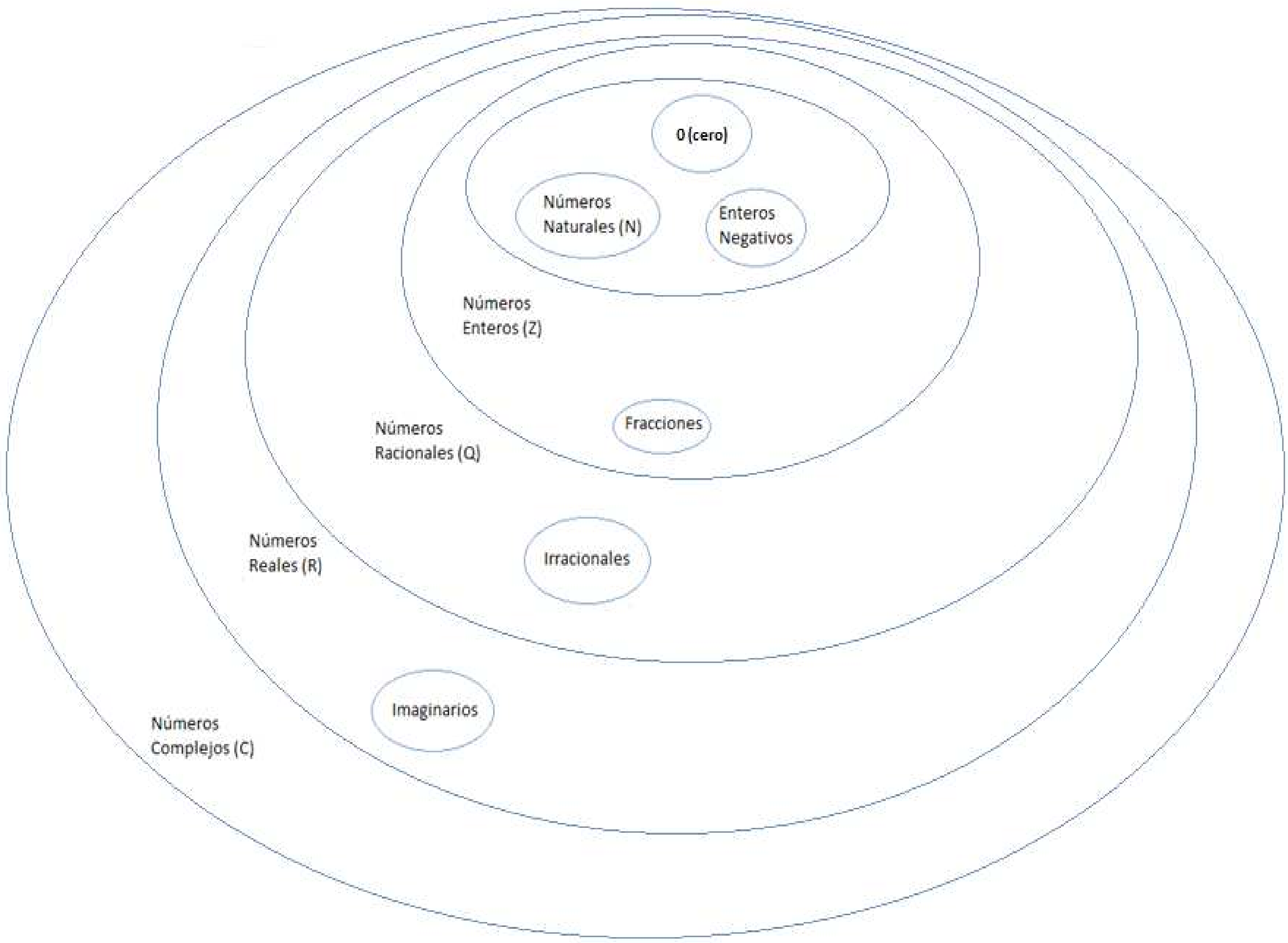
A. $n = 134_8$

$n = 1 * 8^2 + 3 * 8^1 + 4 * 8^0 \quad = 92_{10} \quad (92_{10} = 134_8)$

B. $n = 134_{10}$

$n = 1 * 10^2 + 3 * 10^1 + 4 * 10^0 \quad = 100 + 30 + 4 \quad = 134$

6.2. Relaciones entre conjuntos de tipos numéricos



Este concepto de tipo de numéricos será importante que lo tengan presente para las próximas Unidades del curso donde será utilizados en conjunto con otros conceptos vistos en esta Unidad, aunque de momento es meramente informativo.

7. Representaciones de números en distintas bases

Como hemos visto, los números pueden representarse en sistemas de distintas bases, los cuales tienen una cantidad de símbolos igual al número de la base y el peso de la posición de esos símbolos depende también de la base del mismo.

La elección de la base puede ser cualquiera y podemos convertir la expresión de una cantidad de una base a otra.

En particular nos centraremos en algunas bases que son las más usadas:

- Base 10 (Decimal), cuyos símbolos son 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9
- Base 2 (Binario), cuyos símbolos son 0 y 1
- Base 16 (Hexadecimal), cuyos símbolos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F

Veamos cómo es la notación del "treinta y cinco" en estos sistemas:

$$\mathbf{35_{10} \equiv 100011_{2} \equiv 23_{16}}$$

El símbolo "≡" significa que dos valores son equivalente, por lo que la primera expresión sería: "el número treinta y cinco en base decimal es equivalente al número uno, cero, cero, uno, uno en base binaria".

Veamos cómo se llega desde cada uno haciendo operaciones en base 10, pasa pasarlos a base decimal:

$\mathbf{35_{10}} = 3 * 10^1 + 5 * 10^0$

$= 3 * 10 + 5 * 1 = \mathbf{35_{10}}$

NOTA: las potencias a las que se eleva cada término surgen de la posición en la que se encuentran dentro de la expresión, comenzando a contar desde 0 de derecha a izquierda. Por lo que en el número 35, el 5 correspondería a la posición 0, y cálculo sería "5 * 100". El 3 estaría en la posición 1, por lo que se "3 * 101" y así sucesivamente.

$\mathbf{100011_2} = 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 =$

$= 1 * 32 + 0 * 16 + 0 * 8 + 0 * 4 + 1 * 2 + 1 * 1$

$= 32 + 2 + 1 = \mathbf{35_{10}}$

$\mathbf{23_{16}} = 2 * 16^1 + 3 * 16^0 = 2 * 16 + 3 * 1$

$= 32 + 3 = \mathbf{35_{10}}$

Como vemos hemos utilizado la definición polinómica de un número.

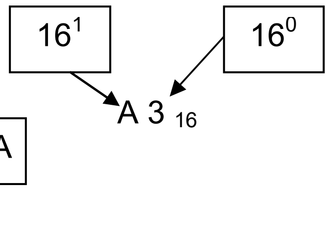
$$\mathbf{n_b = a_n , a_{n-1} , \dots , a_2 , a_1 , a_0 = a_0 * b^0 + a_1 * b^1 + a_2 * b^2 + \dots + a_{n-1} * b^{n-1} + a_n * b^n}$$

La conversión de la expresión de un número de cualquier base a base 10 se realiza teniendo en cuenta este polinomio y haciendo las operaciones en base 10.

7.1. Conversión de cualquier base a base 10

<u>**Ejemplos**</u>:

Hexadecimal → Decimal

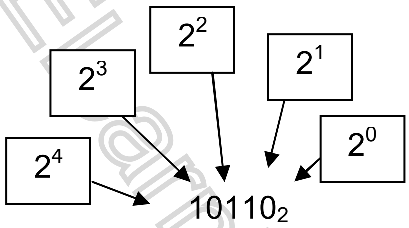


$A3_{16} = A * 16^1 + 3 * 16^0 = 10 * 16 + 3 * 1 =$

$= 160 + 3 =$

$= 163_{10}$

Binario → Decimal



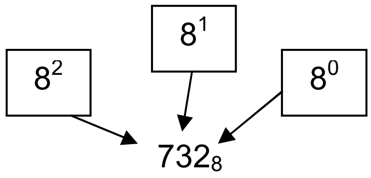
$10110_2 = 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 =$

$= 1 * 16 + 0 * 8 + 1 * 4 + 1 * 2 + 0 * 1 =$

$= 16 + 0 + 4 + 2 + 0 =$

$= 22_{10}$

Octal →Decimal



$732_8 = 7 * 8^2 + 3 * 8^1 + 2 * 8^0$

$= 7 * 64 + 3 * 8 + 2 * 1 =$

$= 448 + 24 + 2 =$

$= 474_{10}$

7.2. Conversión de decimal a otras bases

La conversión de un decimal a otras bases se realiza dividiendo el numero decimal por la base a la que se quiere convertir, el resto de esa división será el digito menos significativo del número en la nueva base. El resultado se vuelve a dividir por la base, el resto es el digito siguiente en significatividad en la nueva base. Se continúa de esta manera hasta que el resultado es menor que la base, este resultado es el digito más significativo en el número convertido. Veamos el mecanismo en práctica.

Ejemplos:

Decimal → Binario

$35_{10} = 100011_2$

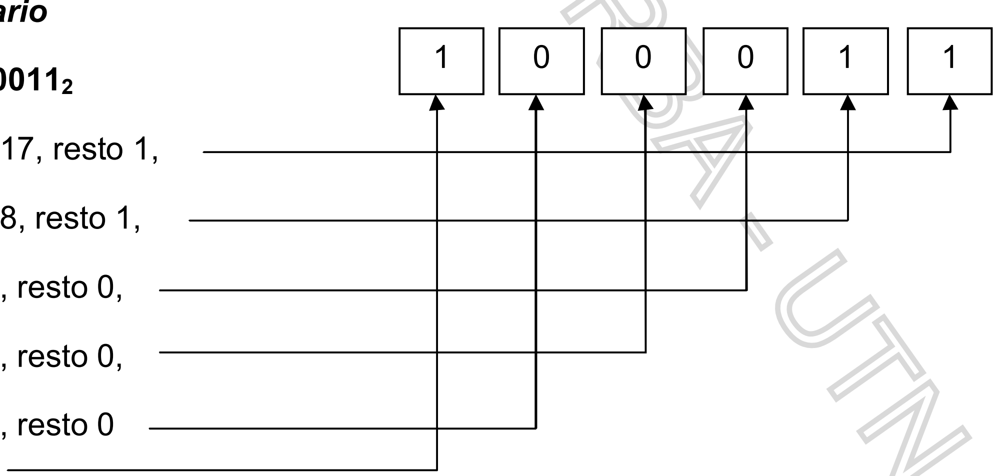
35/2→resultado 17, resto 1,

17/2→resultado 8, resto 1,

8/2→resultado 4, resto 0,

4/2→resultado 2, resto 0,

2/2→resultado 1, resto 0



Decimal→Hexadecimal

$35_{10} = 23_{16}$

35/16→resultado 2, resto 3

2	3

Decimal→ Octal

$35_{10} = 43_{8}$

35/8 →resultado 4, resto 3

4	3

7.3. Casos particulares

<u>Conversión de octal a binario</u>

Cada digito se convierte separadamente a su equivalente en binario, expresado en 3 dígitos.

<u>Ejemplo:</u>

532_8 a binario

5 = 101

3 = 011

2 = 010

$532_8 = 101011010_2$

Conversión de binario a octal

Se separa el número binario en grupos de a tres dígitos de derecha a izquierda, cada uno de estos grupos se convertirá en su expresión en octal. Si el conjunto binario de la extrema izquierda no tiene 3 dígitos, agregar ceros hasta completarlo.

Ejemplo:

110111000100_2 a octal

$$\underbrace{110}_{\mathbf{6}}\ \underbrace{111}_{\mathbf{7}}\ \underbrace{000}_{\mathbf{0}}\ \underbrace{100_2}_{\mathbf{4_8}}$$

Conversión de hexadecimal a binario

Cada dígito se convierte separadamente a su equivalente en binario, expresado en 4 dígitos.

Ejemplo:

$BC5_{16}$ a binario

B(11) = 1011

C(12) = 1100

5 = 0101

$BC5_{16}$ = 1011110001012

Conversión de binario a hexadecimal

Se separa el número binario en grupos de a cuatro dígitos de derecha a izquierda, cada uno de estos grupos se convertirá en su expresión en hexadecimal. Si el conjunto binario de la extrema izquierda no tiene cuatro dígitos, agregar ceros hasta completarlo.

Ejemplo:

$1111\ 1110\ 0010\ 1001\ 1110_2$ a hexadecimal

$1111\ 1110\ 0010\ 1001\ 1110_2$

F E 2 9 E_{16}

¿Por qué son importantes estas representaciones?

- **Base 10: Tomamos como punto de partir la base decimal, que es la que nos resulta más natural de entender a la gran mayoría.**
- **Base 2: A partir de ahí, vemos como se llega a convertir un número decimal a binario. Esto es relevante, ya que esta clase operaciones son las que hace y procesa, justamente, un procesador de una computadora. Pero no solamente un número es susceptible de ser convertido a decimal, sino también letras y símbolos que, en conjunto y agrupados, representan todo aquello que realizamos en una computadora.**
- **Base 16: Normalmente las direcciones de memoria RAM de una computadora (que tiene muchísimas combinaciones posibles), se representan con números hexadecimales ya que se logra una importante abreviación en cuanto a la cantidad de cifras utilizadas con respecto a números decimales o, más aún, a los binarios.**

Como conclusión, podemos ver que estas simples operaciones de transformación son TODO lo que ocurre dentro de una computadora, y esa es la misión y función del procesador: transmitir impulsos eléctricos que se expresan, en la teoría, como 1s y 0s, que es la base binaria y es la base a que se transforman todos los datos que maneja una computadora.

8. Matrices

Es posible que les resulte familiar las estructuras formadas por datos en filas y columnas como forma de almacenar información de manera ordenada. Un ejemplo de esto podría ser una planilla de cálculo, como las de Microsoft Excel o las de Google Spreadsheets.

Un ejemplo de datos almacenables en este tipo de estructuras son, por ejemplo, las posiciones de los clubes del futbol en un campeonato, de modo que las filas dan información por cada equipo y las columnas lo hacen sobre los partidos jugados, ganados, empatados, perdidos, goles a favor, goles en contra y total de puntos. Como vemos a continuación en esta matriz compuesta de 21 filas y 9 columnas:

2014/2015

POS	EQUIPO	PJ	PG	PE	PP	GF	GC	PTS
1	Juventus	27	19	7	1	54	14	64
2	Roma	27	13	11	3	38	21	50
3	Lazio	27	15	4	8	49	27	49
4	Napoli	27	13	7	7	46	35	46
5	Fiorentina	27	12	9	6	39	29	45
6	Sampdoria	27	11	12	4	36	28	45
7	Internazionale	27	9	10	8	42	35	37
8	Genoa	26	9	10	7	37	32	37
9	Torino	27	9	9	9	30	30	36
10	Milan	27	8	11	8	38	34	35
11	Palermo	27	8	11	8	38	40	35
12	Udinese	26	8	8	10	29	34	32
13	Sassuolo	27	7	11	9	33	40	32
14	Hellas Verona	27	8	8	11	33	46	32
15	Empoli	27	5	15	7	27	29	30
16	Chievo	27	7	8	12	20	30	29
17	Atalanta	27	5	10	12	22	37	25
18	Cagliari	27	4	9	14	33	50	21
19	Cesena	27	4	9	14	25	48	21
20	Parma	25	3	3	19	21	51	9

ACTIVIDAD 3:

¿Qué otros ejemplos de la vida cotidiana se representan en estructuras del tipo MATRICES?

A tener en cuenta:

- Posteen al menos 1 y cómo máximo 3 ejemplos.
- Cada uno debe abrir un tema nuevo en el foro.
- En ese tema envíen todos los ejemplos en el mismo mensaje (NO hacer un posteo por cada ejemplo).
- Los que se animen, propongan mejoras o correcciones a las proposiciones publicadas de sus compañeros.

Comparta sus respuestas en el foro de actividades de la Unidad.

Una definición formal de matriz es la siguiente:

Definición: *Sean m y n enteros positivos. Un arreglo rectangular en m filas y n columnas de números reales:*

$$A = \begin{pmatrix} a_{11} a_{12} \dots a_{1n} \\ a_{21}\ a_{22} \dots a_{2n} \\ \vdots \quad \vdots \qquad \vdots \\ a_{m1}\ a_{m2} \dots a_{mn} \end{pmatrix}$$

Se llama matriz de orden (o tamaño) m $\times$ n sobre IR.

***IR* es una notación que significa "números reales" (ver el punto 6.2 de esta Unidad)**

En forma abreviada se escribe $A = (a_{ij})_{m \times n}$ donde a_{ij} denota la entrada de A en la fila i y la columna j.

El conjunto de las matrices de orden m$\times$n con entradas reales se denota con M(m, n, IR). En el caso de matrices cuadradas, es decir con igual número de filas y columnas, se usa la notación M(n, IR) y se dice que son de orden n.

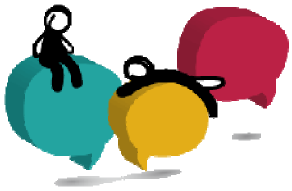
Ejemplo:

Si A $\in$ M(2, 3, IR) es la matriz de dos filas y tres columnas con entradas reales:

$$A = \begin{pmatrix} A\ B\ C \\ D\ E\ F \end{pmatrix}$$

Entonces $a_{12} = B$ y $a_{23} = F$

De momento es importante y suficiente que les quede este concepto sobre las matrices



Actividades de la Unidad 2:

- Participar de los foros con las consignas mencionadas en esta Unidad

Lo que vimos

- Lógica Matemática
- Proposiciones y Operaciones lógicas
- Algebra
- Reglas de Inferencia y el Modus ponendo ponens
- Sistemas de numeración
- Representaciones de números en distintas bases
- Matrices

Lo que viene:

- Principales aspectos necesarios relacionados a la "forma de pensar" del programador
- Prácticas de razonamiento y resolución de problema necesarias para desarrollar algoritmos
- Proceso para el planteo de soluciones a un problema dado
- Tipos de algoritmos y la forma de representarlos