Técnico en

<DESARROLLO DE SOFTWARE>

Introducción a la Programación de Computadoras

Introducción a la Programación de Computadoras

Unidad I: Introducción a la programación

En esta lección es una breve introducción a los temas que necesita entender previo a empezar a programar. En los próximos cursos estará entrando en más detalle de los siguientes temas y realizando mayor práctica de los mismos. Por el momento es importante que entienda y pueda aplicar los conceptos básicos mediante una introducción a la programación.

1. Sistema binario

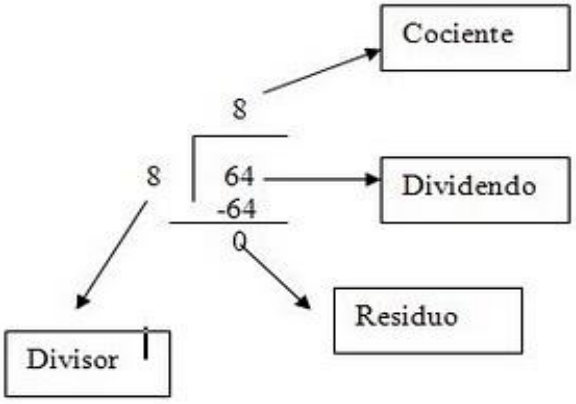
Para poder empezar a programar, es necesario entender que las computadoras utilizan el sistema binario para procesar las instrucciones que ingresamos en código que nosotros entendemos.

El sistema binario es un sistema numérico representado por dos dígitos, conocidos como bits: 0 y 1. Brevemente aprenderemos cómo convertir números decimales a binario. Lo único que necesitamos saber de forma clara es el método de la división.

Existen varias técnicas que puede utilizar para traducir un número de decimal (recuerde que un número decimal se encuentra en base 10, es decir que tiene 10 dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) a un número binario (dos dígitos: 0, 1). A continuación se muestran dos de ellas:

Divisiones, utilizando el residuo

Esta técnica lo que hace es que iremos utilizando el cociente para generar el próximo dividendo y el residuo para obtener el nuevo número binario. En nuestro caso, el divisor será 2 y el dividendo original será el número decimal.



Paso 1

Definir nuestra división. Con fines de ejemplo, utilizaremos el decimal original a trasladar a binario: 179.

$$2 \overline{)\,179}$$

Paso 2

Operar la división hasta llegar a tener un residuo igual a 0 o 1.

$$\begin{array}{r} 89 \\ 2 \overline{)\,179} \\ \underline{16} \\ 19 \\ \underline{18} \\ 1 \end{array}$$

Para esta división, **el residuo es 1**. Por el momento guarde el residuo.

Paso 3

Realice una nueva división, pero ahora utilice el cociente como el dividendo. Repita este paso hasta que el cociente sea 0 o 1.

```
   44
2 |89
   8
  09
   8
   1
```

Paso 4

Repetir el paso 3 hasta que se tenga un cociente igual a 0.

Para la división anterior, **el residuo es 1**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

```
   22
2 |44
   4
  04
   4
   0
```

Para esta división, **el residuo es 0**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

$$\begin{array}{r|l} & 11 \\ 2 & \overline{22} \\ & 2 \\ \hline & 02 \\ & 2 \\ \hline & 0 \end{array}$$

Para esta división, **el residuo es 0**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

$$\begin{array}{r|l} & 5 \\ 2 & \overline{11} \\ & 10 \\ \hline & 1 \end{array}$$

Para esta división, **el residuo es 1**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

$$\begin{array}{r|l} & 2 \\ 2 & \overline{5} \\ & 4 \\ \hline & 1 \end{array}$$

Para esta división, **el residuo es 1**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

$$\begin{array}{r|l} & 1 \\ 2 & \overline{2} \\ & 2 \\ \hline & 0 \end{array}$$

Para esta división, **el residuo es 0**. Repetiremos el paso 3 de nuevo, pero ahora utilizando como dividendo el cociente de la división anterior:

$$\begin{array}{r|l} & 0 \\ 2 & 1 \\ & 0 \\ \hline & 1 \end{array}$$

Ahora que se tiene un cociente igual a 0 o simplemente ya no se pueden realizar más divisiones con el último cociente, entonces se debe proceder a construir el nuevo número binario, iniciando con el residuo de la última división, hasta llegar al primer residuo. En este caso el binario del decimal 179 es **10110011**. También se puede utilizar la siguiente notación:

$$\mathbf{179_{10} = 10110011_2}$$

Potencias de dos

Para este método, es necesario crear una tabla, creando todas las potencias de 2 que se consideren necesarias, de derecha a izquierda, como la siguiente:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
2x2x2x2x2x2x2	2x2x2x2x2x2	2x2x2x2x2	2x2x2x2	2x2x2	2x2	2	1
128	64	32	16	8	4	2	1

De nuevo se estará utilizando el número decimal 179 para corroborar que el cálculo del método anterior sea exitoso.

Paso 1

Crear la tabla con la mayor cantidad de potencias de dos posible, tomando en cuenta que la mayor potencia, en decimal, no debe sobrepasar al número decimal a trasladar a binario. Esto quiere decir que para el ejemplo ya no es necesario calcular 2^8, ya que este

valor es igual a 256, y 256 es mayor a 179, por lo que para este ejemplo basta con llegar a 2^7.

Paso 2

Restar la mayor potencia que cabe en 179.

179 - 128 = 51

Y se le coloca un 1 a la casilla de 2^7.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
1							

Paso 3

Al resultado de la resta del paso anterior, volver a restarle la mayor potencia que quepa en ella. Repetir este paso hasta que el resultado sea 0.

51 – 32 = 19

Y se le coloca un 1 a la casilla de 2^5.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
1		**1**					

Se repite el proceso con el resultado:

19 – 16 = 3

Y se le coloca un 1 a la casilla de 2^4.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1		1	1				

Se repite el proceso con el resultado:

3 – 2 = 1

Y se le coloca un 1 a la casilla de 2^1.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1		1	1			1	

Se repite el proceso con el resultado:

1 – 1 = 0

Y se le coloca un 1 a la casilla de 2^0.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1		1	1			1	1

Paso 4

Se rellena con 0 las casillas vacías, ya que no se utilizó esta potencia para las restas.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	1	0	0	1	1

Como puede notar, el resultado de trasladar 179 en base 10 a binario es igual a **10110011**. Siendo éste el mismo resultado del método anterior.

$$179_{10} = 10110011_2$$

2. Lógica

Ésta nos brinda los lineamientos y herramientas que se necesitan para determinar si un razonamiento es válido o no es válido y dependiendo de su ambiente, a esta validez le podemos llamar de distinta forma: 0 o 1, verdadero o falso, entre otros.

En los próximos cursos entrará en más detalle de la lógica matemática. Por el momento es necesario que entienda que una proposición o argumento o razonamiento puede ser verdadera o falsa.

Una proposición lógica es un enunciado al que se le puede asignar un valor de verdad: verdadero (correcto) o falso (incorrecto).

Ejemplos de proposiciones lógicas

- El cielo es color azul.

- El mar es color amarillo.
- Las mesas vuelan.
- Los osos son morados.
- Todas las sillas son blancas.
- Su catedrática se llama Ana Isabel.

Algunas de las proposiciones son siempre verdaderas, otras son verdaderas o falsas si se cumple una condición extra, mientras que otras son siempre falsas.

Por ejemplo: El cielo es color azul siempre y cuando sea de día y el clima esté soleado, de lo contrario el cielo será de otro color. El mar no es color amarillo. Las mesas puede que vuelen, siempre y cuando no haya gravedad en el entorno donde se encuentren. Los osos que tienen vida no son morados, pero si son de peluche puede darse el caso en que si sean morados. No todas las sillas son blancas. Y por último, su catedrática si se llama Ana Isabel.

Es importante aprender estos conceptos, ya que son necesarios para aplicar acciones dependiendo de nuestros argumentos; por lo que se necesita corroborar si éstas son verdaderas o falsas y a la vez saber qué acciones aplicar dependiendo del tipo de respuesta que se tenga.

Conjunción / Y / AND / && / ^

Dependiendo del lenguaje o si está trabajando únicamente en pseudocódigo o con operaciones matemáticas, este símbolo va a cambiar, pero en el lenguaje natural, esto no cambia mucho con la programación.

Se utiliza para conectar dos proposiciones lógicas que deben ser verdaderas para poder obtener un resultado verdadero.

Ejemplo

- Proposición 1: El teléfono es iPhone.
- Proposición 2: El sistema operativo del teléfono es iOS.

Si se quieren conectar las dos proposiciones anteriores:

El teléfono es iPhone **y** el sistema operativo del teléfono es iOS, podemos concluir que nuestra nueva proposición es verdadera, ya que ambas se cumplen.

Nota: Si se diera el caso en que alguna de las dos proposiciones no se cumple, entonces el resultado siempre será falso.

Disyunción / ó / OR / || / V

Se utiliza para conectar dos proposiciones lógicas, donde cualquiera de las dos debe ser verdadera para obtener un resultado verdadero.

Ejemplo

- Proposición 1: El sistema operativo del teléfono es Android.
- Proposición 2: El sistema operativo del teléfono es iOS.

Si se quieren conectar las dos proposiciones anteriores:

El sistema operativo del teléfono es Android **o** el sistema operativo del teléfono es iOS.

Negación/ ¬ / ! / NO / NOT

Se utiliza para negar una proposición lógica. Si la proposición es verdadera, ésta se vuelve falsa y si la proposición es falsa, ésta se vuelve verdadera.

Ejemplo

- Proposición: La pared es blanca.
- Proposición negada: La pared **no** es blanca.

Existen más operaciones, pero las aprenderá, al igual que las anteriores, con mayor detalle en su curso de matemática y las aplicará en los próximos cursos.

3. Algoritmo

Es un conjunto de pasos finitos que se deben seguir para resolver un problema. El conjunto de pasos debe tener un orden lógico y pueden ser intercambiables siempre y cuando no alteren el resultado del problema.

Ejemplo

Algoritmo para planchar una camisa

1. Colocar la camisa en el planchador o en una base estable y plana.
2. Conectar la plancha.
3. Encender la plancha y esperar a que se encuentre en la temperatura correcta.
4. Colocar la plancha sobre el cuello de la camisa, moviéndola constantemente hasta que ya no esté arrugado.

5. Colocar la plancha sobre los hombros de la camisa, moviéndola constantemente hasta que ya no esté arrugado.
6. Colocar la plancha sobre las mangas y brazos de la camisa, moviéndola constantemente hasta que se encuentren lisos.
7. Cambiar de posición la camisa, de tal forma que se pueda planchar la parte frontal, iniciando por la sección de los botones. Mover la plancha constantemente hasta que ya no esté arrugado.
8. Darle vuelta a la camisa y colocar la plancha sobre la espalda de la camisa, moviendo la plancha constantemente hasta que ya no esté arrugado.
9. Darle la vuelta a la camisa y colocar la plancha sobre la sección donde se encuentran los ojales de la camisa, moviendo la plancha sobre esta sección de forma constante hasta que esta sección ya no se encuentre arrugada.

4. Programación

La programación en el área de informática define la acción de llevar a cabo una serie de pasos con el fin de generar código fuente de un programa con especificaciones previamente definidas con la ayuda de un lenguaje definido.

El primer paso para programar es tener un problema a resolver. Al momento en que se tenga un problema, es necesario tener claro el problema previo a iniciar a programar y por lo mismo es necesario diseñar el algoritmo o los pasos ordenados que se llevarán a cabo para resolver el problema.

Es de gran utilidad crear pruebas manuales y automáticas, existen distintos programas que se pueden utilizar para aplicar las pruebas automáticas; todo depende del lenguaje que se esté utilizando y si son para probar el código o la parte visual del programa.

Se recomienda que las pruebas sean diseñadas y se escriban previo a empezar a escribir una solución en cualquier lenguaje de programación.

Las mejoras se van implementando en el proceso, pero de preferencia se espera que el diseño de su programa sea revisado previo a iniciar a programar para evitar que le tome demasiado tiempo realizar las modificaciones o que incluso el programa final cambie de dirección en el camino y el resultado sea totalmente distinto a lo esperado.

Lenguaje de programación

Es la herramienta o software utilizado para crear programas. Un lenguaje de programación provee las abstracciones, principios de organización y las estructuras de control para escribir programas

Solo hay un tipo de lenguaje de programación que las computadoras pueden entender e interpretar: el código binario, también llamado código máquina. Este es el nivel más bajo de los lenguajes en los que se puede escribir un programa. Todos los demás lenguajes se pueden distinguir como de alto nivel o bajo nivel de acuerdo a que tanto se asemejan al código máquina.

Programa

Es la implementación de un algoritmo de tal forma que la computadora entienda. Para la implementación de los algoritmos se utilizan lenguajes de programación.

5. Editores de texto y entornos de desarrollo

Antes de dar inicio a las primeras instrucciones de código del presente curso, y en general del Técnico, es necesario generar un entorno de trabajo en el que se sientan cómodos y que les sea posible trabajar sin dificultades.

Para ello necesitamos una computadora con un sistema operativo que nos permita trabajar en nuestros programas. Ustedes pueden trabajar en la versión que mejor les convenga, pero recomiendo que utilicen versiones actualizadas, ya sea de Windows, Linux o macOS. En lo personal, recomiendo que utilicen una de las distribuciones de Linux o macOS.

Los editores de texto que recomiendo su uso, por experiencia propia, son los siguientes:

Editores de texto con entorno visual

- Atom: https://atom.io/ (Windows, Linux, macOS)
- Sublime: https://www.sublimetext.com/ (Windows, Linux, macOS)
- Notepad++: https://notepad-plus-plus.org/downloads/ (Windows)
- Visual Studio Code: https://code.visualstudio.com/ (Windows, Linux, macOS)

Editores de texto para utilizar en la terminal/consola

- Emacs: sudo apt-get install emacs-nox
- Nano
- Vim
- IDE (Integrated Development Environment)

Como sus siglas lo dicen, es un ambiente de desarrollo que proporciona servicios integrados para facilitar el desarrollo, maximizando la productividad del programador.

A continuación les comparto un listado de IDEs que pueden utilizar:

- Visual Studio IDE: https://visualstudio.com/ (Windows, Linux, macOS)
- Eclipse IDE: https://www.eclipse.org/downloads/
- Netbeans: https://netbeans.apache.org/download/index.html
- IntelliJ IDEA: https://www.jetbrains.com/es-es/idea/

No se limite al listado anterior. Si conoce otro editor o IDE distinto al listado anterior y se le facilita su uso, entonces puede utilizarlo para el desarrollo de sus tareas y proyectos.