

Principios de informática

Braulio José Solano Rojas

15 de abril de 2010

Capítulo 1

Introducción

1.1. Historia

¿Qué es la computación y de dónde viene? Aunque parezca increíble, podemos buscar a la computación en el origen de nuestra especie, de la técnica y de la escritura, mucho antes de la computación electrónica.

Hubo desde los tiempos de los primeros homo sapiens sapiens una gran preocupación por registrar y calcular. En las cuevas con pinturas rupestres de nuestra prehistoria podemos ver como se graficaba la cantidad de animales que habían cerca del coto de caza y los guerreros de otros grupos. Los primeros vestigios de la incipiente necesidad de cálculo realizados por medios mecánicos artesanales los podemos encontrar 3000 años antes de Cristo (AC). En la Antigua Mesopotamia, Babilonia, los habitantes realizaban sus “cuentas” en pequeñas bolas de semillas o piedras que agrupaban en carriles de caña. Además, es absolutamente necesario decir que el algoritmo (un serie de pasos ordenados y finitos para llevar a cabo un cálculo), que es una de las bases de la computación actual, fue inventado por un matemático babilónico aproximadamente 1800 AC. Estos conocimientos se exportaron con los intercambios comerciales y las guerras.

Existió otro instrumento que da fe del ímpetu humano por hacer cálculos: el ábaco. Su origen se pierde en la historia de la humanidad. Fue perfeccionado por las culturas orientales y aún hoy en día es utilizado.

También en el nuevo mundo tenemos testimonios de la necesidad de llevar registros y de hacer cálculos. El Quipu, instrumento utilizado por el imperio Inca, cuyos ejemplares más antiguos se encontraron en la ciudad más antigua de América, Caral, fue utilizado para contabilidad y registro (censos y cosechas). Además, actualmente se investiga su uso como sistema de representación lingüística y de memoria (historia, canciones y poemas).

Así, no es de extrañar que los primeros computadores fueron personas. ¡Sí! Las computadoras electrónicas y sus versiones mecánicas más tempranas fueron llamadas así porque realizaban el trabajo que fue anteriormente realizado por seres humanos (predominantemente mujeres). El trabajo de computadora consistía en llevar a cabo cálculos repetitivos para computar cosas tales como tablas de navegación y mareas, y posiciones planetarias para almanaques astronómicos. Tal trabajo tan difícil llevó a los inventores a buscar su automatización, lo que llevaría en el futuro al nacimiento de la computación por medio de máquinas tal como la conocemos hoy.

En la historia de las máquinas, podríamos hablar de dos partes: mecánicas y digitales. Con respecto de las máquinas mecánicas podemos mencionar personajes notorios como John Napier (cuyo aporte llevó a la regla de cálculo), Leonardo Da Vinci (realizó planos de una máquina mecánica de cálculo), Wilhelm Schickard (construyó la primer máquina mecánica de cálculo), Blaise Pascal (niño prodigio y matemático francés que inventó la Pascaline, calculadora sumadora), Gottfried Wilhelm Leibniz (inventor de una calculadora con las cuatro operaciones aritméticas a diferencia de la Pascaline), Joseph Marie Jacquard (inventó las tarjetas perforadas para poder programar su telar mecánico y así automatizar el diseño de las telas) y Charles Babbage (propuso dos máquinas: la máquina de diferencias -cerca de 1822- y la máquina analítica, la última con equivalentes a la unidad de memoria y la unidad procesadora central modernas).

Charles Babbage fue un visionario, ideó el diseño de la máquina analítica, e hizo la conexión entre las tarjetas perforadas del telar mecánico y su máquina. Las tarjetas perforadas servirían como medio de almacenamiento. La máquina de Charles no fue construida debido a dificultades técnicas. Sin embargo, su invento nos lleva a Ada Byron. Ada era amiga de Charles e hija del famoso poeta Lord Byron (su nombre de casada será Condesa Lady Lovelace). A pesar de tener 19 años se encontraba fascinada por la máquina analítica y aprendió mucho acerca de ella por medio de la correspondencia y de los encuentros con Charles. Aunque Charles no publicó su conocimiento durante mucho tiempo, Ada escribió una serie de notas donde detalló secuencias de instrucciones que ella había preparado para la máquina analítica. Ada inventó las subrutinas y fue la primera en reconocer la importancia de los ciclos en los programas. Así la primer persona programadora de la historia fue una mujer.

1.2. Por qué programar

1.3. Hardware (mecamática)

1.4. Software (programática)

Capítulo 2

Java

2.1. Tipos de datos

2.2. Entrada y salida

La entrada y salida es importante dado que cómo se vió anteriormente la computadora tiene periféricos o dispositivos de entrada y salida que van a permitir la interacción entre humanos y ella misma. Es decir, la comunicación entre el usuario y la computadora se hace por medio de la entrada y la salida de información.

2.2.1. Salida en consola

En Java podemos hacer salida en consola por medio de `System.out.println` o `System.out.print`¹. La primera opción realiza un cambio de línea al final de la salida, mientras que la segunda no.

Ejemplo:

```
System.out.println ("¡Qué_solazo_hace_hoy!");
```

```
System.out.println (variable);
```

2.2.2. Entrada en consola

Para hacer entrada en Java por medio de la consola se requiere de las clases `BufferedReader`² e `InputStreamReader`³. Es necesario incluir al inicio del programa la biblioteca de clases `java.io`:

```
import java.io.*;
```

La clase `BufferedReader` se utiliza por razones de eficiencia. Sirve para envolver a `InputStreamReader`.

Ejemplo:

¹<http://java.sun.com/javase/6/docs/api/java/lang/System.html>

²<http://java.sun.com/javase/6/docs/api/java/io/BufferedReader.html>

³<http://java.sun.com/javase/6/docs/api/java/io/InputStreamReader.html>

```
1 BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
```

2.2.2.1. Ejemplo: cómo leer un número en consola

Asumiendo que creamos el objeto entrada del ejemplo anterior, entonces, para leer un número flotante grande, se haría lo siguiente:

```
2 double numeroReal = Double.parseDouble(entrada.readLine());
```

También podríamos hacerlo “más orientado a objetos” utilizando la clase Double⁴:

```
2 Double numeroReal = new Double(entrada.readLine());
```

2.2.2.2. Alternativa a BufferedReader e InputStreamReader

También es posible utilizar la clase Scanner⁵ dado que las clases BufferedReader e InputStreamReader juntas imponen una carga mnemónica al programador que no es tan simple como la entrada usual en otros lenguajes.

Ejemplo:

```
1 Scanner rastreador = new Scanner(System.in);  
2 Double numeroReal = new Double(rastreador.nextDouble());
```

Para utilizar la clase Scanner es necesario incluir la biblioteca de clase java.Util.Scanner:

```
import java.Util.Scanner;
```

2.2.3. Entrada gráfica

2.3. Variables

2.4. Expresiones y operadores

2.5. Estructuras de control

Las estructuras de control nos servirán para controlar el flujo del programa. Donde por flujo entenderemos los pasos a seguir para llegar a una solución, o sea, el procedimiento.

2.5.1. Instrucciones de selección

2.5.1.1. Instrucción if simple

La instrucción if (si condicional) nos permite probar una condición y luego ejecutar instrucciones basándose en si la condición era verdadera o falsa. Java posee dos formas de instrucciones if, la if simple y la if/else.

⁴<http://java.sun.com/javase/6/docs/api/java/lang/Double.html>

⁵<http://java.sun.com/javase/6/docs/api/java/util/Scanner.html>

La forma más simple de la instrucción if tiene la siguiente forma general:

```

1 if (condición)
2     instrucción 1;

```

Si la condición es verdadera ejecutamos la instrucción 1. Si la condición es falsa omitimos la instrucción 1. La instrucción dentro de la declaración if está indentada para que sea más fácil visualizar la estructura del programa dentro de la declaración.

Si deseamos ejecutar varias instrucciones cuando la condición es verdadera se utiliza un bloque que está compuesto de un conjunto de instrucciones dentro de llaves. Por ejemplo:

```

1 if (condición) {
2     instrucción 1;
3     instrucción 2;
4     ...
5     instrucción n;
6 }

```

2.5.1.2. Instrucción if/else

Una instrucción if/else nos permite ejecutar un conjunto de instrucciones si una condición es verdadera y un conjunto de instrucciones diferente si la condición es falsa. La forma más simple de instrucción if/else es la siguiente:

```

1 if (condición)
2     instrucción 1;
3 else
4     instrucción 2;

```

Las instrucciones 1 y 2 puede ser reemplazadas por instrucciones compuestas (bloques). Las instrucciones 1 y 2 pueden ser también instrucciones vacías (sólo un punto y coma). Si la instrucción 2 es una instrucción vacía, la declaración if/else debería ser probablemente planteada como un if simple.

2.5.1.3. Condiciones

Una condición es toda aquella expresión que podamos evaluar a falso o verdadero. Las expresiones que podemos utilizar para una condición pueden ser comparaciones, expresiones lógicas o incluso variables.

Operador	Ejemplo	Significado
==	x == y	$\text{¿}x = y\text{?}$
!=	x != y	$\text{¿}x \neq y\text{?}$
>	x > y	$\text{¿}x > y\text{?}$
<	x < y	$\text{¿}x < y\text{?}$
>=	x >= y	$\text{¿}x \leq y\text{?}$
<=	x <= y	$\text{¿}x \geq y\text{?}$

2.5.2. Ciclos

Los ciclos