

Control de Flujo

Solución de Problemas con Programación (TC1017)

M.C. Xavier Sánchez Díaz
mail@tec.mx



Outline

- 1 Control de Flujo
- 2 Representaciones del control de flujo

¿Qué es control de flujo?

Control de Flujo

Un lenguaje de programación nos permite expresar la **manera en la que los componentes de ejecución se entrelazan** para realizar un cálculo.

De este modo, tenemos distintas herramientas a nuestra disposición para poder ejecutar comandos y obtener los resultados que queremos de distintas maneras:

- 1 Usando **secuencias de instrucciones**
- 2 **Seleccionando** entre distintas opciones
- 3 **Repitiendo** procesos
- 4 Haciendo múltiples procesos a la vez

Ejemplos

Control de Flujo

- ¿Me traes una coca? **Si** hay sin azúcar, mejor. **Si no**, la que sea.
- Tengo que ir al cajero, **luego** a la lavandería y **luego** pasar por algo de comer.
- Necesito que me ayudes a calificar **cada uno** de estos ejercicios.
- Ya hablaremos de eso **cuando** seas más grande.
- Vas a tener que buscar en **cada** rincón de la casa para encontrarlo.

Ejemplos

Control de Flujo

Podemos identificar con claridad los tres tipos de *maneras* en que podemos entrelazar los componentes de las oraciones:

- Instrucciones **secuenciales** (Primero voy acá, luego allá. . .)
- Instrucciones **condicionales** (Si esto, entonces aquello, si no. . .)
- Instrucciones **cíclicas** (Para cada uno de estos, haz tal y tal. . .)

Estos son las tres **estructuras de control de flujo** que revisaremos en este curso.

Secuencias

Representaciones del control de flujo

Para comprender qué es una **secuencia**, hay que estudiar primero qué es un **conjunto**.

Conjunto

*A **set** is a finite or infinite **collection** of objects in which order has no significance, and multiplicity is ignored. Members of a set are referred to as **elements** and the notation $a \in A$ is used to denote that a is an element of a set A .*

Paráfrasis de Stover, C. y Weisstein, E., de Set en MathWorld:
<http://mathworld.wolfram.com/Set.html>

Secuencias

Representaciones del control de flujo

Ejemplos de conjuntos

- El conjunto de números naturales $\mathbb{N} = \{1, 2, 3, \dots\}$
- El conjunto de funciones que tiene el MATLAB
- El conjunto de estudiantes del campus
- El conjunto de estudiantes en la clase
- El conjunto de estudiantes en el salón

Secuencias

Representaciones del control de flujo

Una **secuencia** es un **conjunto** el cual está **ordenado**.

Ejemplo

Los números naturales son un conjunto ordenado:

$$\mathbb{N} = \langle 1, 2, 3, 4, \dots \rangle$$

Una receta de cocina es una **secuencia de instrucciones** porque el orden es importante.

Una **guión** es también una secuencia de diálogos, pues una obra de teatro o una película tienen una naturaleza secuencial. Este **guión** se llama en inglés **script**.

Secuencias

Representaciones del control de flujo

Los **scripts** son naturalmente secuenciales:

```
1 % Hypotenuse of a famous triangle
2
3 a = 3;
4 b = 4;
5 c = sqrt(a^2 + b^2);
6
7 message = 'If your triangle has sides of %d and %d units, then the
            hypotenuse is %f units long';
8 printthis = sprintf(message, a, b, c);
9 disp(printthis)
```

Condicionales

Representaciones del control de flujo

Los estatutos condicionales *también* tienen una base matemática:

Implicación lógica

*"Implies" is the connective in **propositional calculus** which has the meaning that "if A is true, then B is also true". In formal terminology, the term conditional is often used to refer to this connective. The symbol used to denote the implication is*

$$A \implies B$$

Paráfrasis de Weisstein, E., de *Implies* en MathWorld:
<http://mathworld.wolfram.com/Implies.html>

Condicionales

Representaciones del control de flujo

Los condicionales son de naturaleza **disyuntiva**:

```
1  % Positive or not
2
3  function positiveornot(x)
4      if x > 0
5          disp('This is a positive number.')
6      elseif x < 0
7          disp('This is a negative number.')
8      else
9          disp('The number is 0, which is not negative or positive.')
10     end
11 end
```

Ciclos

Representaciones del control de flujo

Los **ciclos**, también conocidos como **bucles**, son estructuras de flujo cuya representación matemática es más bien implícita.

¿Cuál es la suma de todos los números que están en un reloj análogo?
Pensemos un poco...

Ciclos

Representaciones del control de flujo

$$\begin{aligned} R &= 12 + 11 + 10 \\ &+ 9 + 8 + 7 \\ &+ 6 + 5 + 4 \\ &+ 3 + 2 + 1 \\ &= 78 \end{aligned}$$

Hay que sumar el 12, más el 11, más el 10, más el... hay un ciclo que podemos identificar, donde el **valor base** va cambiando de uno en uno... A este valor base le llamamos **iterador**.

Y esta suma, que se lee como *la suma de i , desde 1 hasta 12*, se representa de la siguiente manera:

$$\sum_{i=1}^{i=12} i = 78$$

Ciclos

Representaciones del control de flujo

El **iterador** va de uno en uno en los elementos de una **secuencia**, que en el caso anterior era la secuencia de los números del 1 al 12.

Estos ciclos en los cuales sabemos específicamente **cuántas veces** hay que repetir el proceso se conocen como **ciclo PARA**—o en inglés, *for*— pues es *para cada uno...*

```
1 % Display all odd numbers from 1 to 15
2
3 for i = [1 3 5 7 9 11 13 15]
4     disp(i)
5 end
```

Ciclos

Representaciones del control de flujo

Otro ejemplo del ciclo `for...`

```
1 % Display all my grades
2
3 mygrades = [100 90 95 89 88 92]
4
5 for grade = mygrades
6     msg = sprintf('I got a %d!', grade);
7     disp(msg)
8 end
```

Ciclos

Representaciones del control de flujo

Como hemos visto, el **for** necesita de una **secuencia** o **lista** para poder **iterar** en ella. Si el patrón de esta secuencia es muy claro, podemos expresarla por *comprensión* como si fuera un conjunto:

$$X = \{x \mid 0 < x < 15, x \text{ es impar}\}$$

Esto lo podemos expresar en el MATLAB de una manera similar:

$\langle start \rangle : \langle increment \rangle : \langle stop \rangle$

```
1 % Display all odd numbers from 1 to 15
2
3 for i = 1:2:15
4     disp(i)
5 end
```
