# Capítulo 4

# Repaso

## Objetivos

Hasta ahora se ha definido la manera de escribir programas utilizando el lenguaje del robot. También se ha presentado la sintaxis utilizada que permite trasladar al robot, recoger y/o depositar flores y papeles y saber si hay o no flores en la esquina o en la bolsa.

Por otro lado se han analizado diferentes situaciones que requieren la posibilidad de representar información específica del problema.

El objetivo de este capítulo es presentar, analizar y resolver diferentes ejemplos que permitirán la ejercitación de los temas vistos en los capítulos anteriores.

**Temas a tratar**

- ✓ Presentación, análisis y resolución de ejemplos.
- ✓ Conclusiones..

## 4.1 Repaso de variables

En los capítulos anteriores se ha definido la sintaxis de las acciones u órdenes que el robot puede llevar a cabo y se ha indicado como se representará y trabajará con la información relevante que presenta el problema a resolver utilizando el lenguaje DaVinci Concurrente.

Como ya hemos visto:

En general, durante la ejecución de un programa es necesario manipular información que puede cambiar continuamente. Por este motivo, es necesario contar con un recurso que permita variar la información que se maneja en cada momento. Este recurso es lo que se conoce como variable.

Además, sabemos que dentro de un mismo programa pueden utilizarse tantas variables como sean necesarias para representar adecuadamente todos los datos presentes en el problema.

Sin embargo, de todos los ejemplos vistos en los capítulos 2 y 3 podríamos pensar que cada vez que un enunciado requiere informar una cantidad, es necesario recurrir a una variable. A través de un ejemplo, podemos observar que esto no siempre es así.

Analicemos el siguiente ejemplo:

**Ejemplo 4.1:** Programe al robot para que recorra la calle 4 deteniéndose cuando encuentre una esquina que no tiene flores, sabiendo que esa esquina seguro existe. Al terminar debe informar la cantidad de pasos dados.

Este problema admite dos soluciones. Una de ellas utiliza una variable para representar la cantidad de pasos que da el robot y la otra no.

```
programa Cap4Ejemplo1a
comenzar
  iniciar
  mientras HayFlorEnLaEsquina
    mover
  Informar(PosAv-1)
fin
```

```
programa Cap4Ejemplo1an
variables
  pasos:numero
comenzar
  pasos := 0
  iniciar
  mientras HayFlorEnLaEsquina
    comenzar
    pasos := pasos + 1
    mover
    fin
  Informar(pasos)
fin
```

El ejemplo 4.1 demuestra que antes de decidir representar nueva información dentro del programa, es importante analizar si el robot no cuenta con la posibilidad de manejar los datos pedidos y de este modo evitar la declaración de un dato.

- ¿Cuál de las formas de resolver el problema le parece más adecuada? Justificar la respuesta.

## 4.2 Repaso de expresiones lógicas

Recordemos que las expresiones lógicas pueden formarse con variables y expresiones relacionales utilizando los operadores lógicos (conjunción, disyunción y/o negación)

Analicemos los siguientes ejemplos para ejercitar la resolución de expresiones lógicas que combinan varias proposiciones:

**Ejemplo 4.2:** Programe al robot para que informe si en la esquina (7,4) hay solo flor o solo papel (pero no ambos).

```
programa Cap4Ejemplo2
comenzar
  iniciar
  Pos(7,4)
  si (HayFlorEnLaEsquina &  !HayPapelEnLaEsquina) |        (1)
     (!HayFlorEnLaEsquina & HayPapelEnLaEsquina)           (2)
```

```
    Informar( V )
  sino
    Informar( F )
fin
```

Como puede verse en este ejemplo, que en la selección se ha utilizado una disyunción de conjunciones. Es decir que basta con que una de las dos conjunciones sea verdadera para que toda la proposición lo sea.

Cada una de las conjunciones requiere que haya uno solo de los dos elementos: la primera pide que haya flor y no papel (1) y la segunda que haya papel y no flor (2). Obviamente, no pueden ser verdaderas al mismo tiempo. Pero basta con que solo una de ellas lo sea para que se informe V.

**Ejemplo 4.3:** Programe al robot para que recorra la avenida 6 buscando una flor que puede no existir. Al finalizar informar donde está (si la encontró) o F (falso) en caso contrario.

```
programa Cap4Ejemplo3
comenzar
{ posicionarse al comienzo de la Av.6}
{ avanzar sobre la av. hasta encontrar la flor o hasta terminar la avenida}
{ el robot está detenido pero la flor no necesariamente tiene que estar }
{ Informar según corresponda}
fin
```

En el lenguaje del robot esto se escribe de la siguiente manera:

```
programa Cap4Ejemplo3
comenzar
  iniciar
  Pos(6,1)
  {recorre la Av.6 buscando la flor }
  mientras !HayFlorEnLaEsquina & (PosCa < 10)      (1)
    mover
    {ver si encontró la flor o no }
  si HayFlorEnLaEsquina                             (2)
    Informar  ( PosCa )
  sino
    Informar( F )
fin
```

Como podemos observar, en la línea (1) se utiliza una proposición molecular para controlar la iteración. Ahora no alcanza con verificar solamente que la flor no exista (!HayFlorEnLaEsquina) sino que además es necesario tener en cuenta que no se termine la avenida (PosCa < 10).

Dado que ambas condiciones deben cumplirse simultáneamente, se las ha unido por una conjunción. Esta proposición molecular será verdadera cuando ambas proposiciones lo sean. La iteración puede leerse como: “mientras no encuentre la flor y a la vez, el robot no llegue a la calle 10, debe seguir avanzando".

La iteración termina cuando la conjunción es falsa. Esto ocurre por tres motivos:

1. Encontró la flor durante el recorrido de la avenida. Es decir que la condición (PosCa<10) es verdadera pero la proposición (!HayFlorEnLaEsquina ) es falsa.

2. No encontró la flor pero llegó a la calle 10. Es decir que (!HayFlorEnLaEsquina ) es verdadera y (PosCa < 10) es falsa.

3. Encontró la flor sobre la calle 10. En este caso ambas condiciones son falsas.

Por lo tanto, la iteración no necesariamente termina cuando la flor ha sido hallada y para poder informar lo solicitado en el enunciado del problema, será necesario distinguir lo que pasó. Esa es la función de la selección que aparece en la línea (2).

- Se logra el mismo resultado reemplazando la condición que aparece en la línea (2) por PosCa=10 ? Justificar la respuesta.

- Pensar en otra proposición que permita dar a la selección de (2) el mismo funcionamiento.

## 4.3 Ejemplos

Habiendo repasado los aspectos más importantes para la ejercitación propuesta para este capítulo, a continuación se presentan diferentes ejemplos que combinan los temas vistos hasta aquí. Es recomendable que prestemos especial atención a la definición y evaluación de proposiciones.

**Ejemplo 4.4**: Programe al robot para que recorra la calle 2 hasta encontrar una esquina vacía que puede no existir. En caso de encontrarla depositar en ella una flor. Si no pudo depositar (porque no tenía) informar F (falso).

El siguiente programa resuelve este problema:

```
programa Cap4Ejemplo4
comenzar
  iniciar
  {ubicar el robot al comienzo de la calle 2}
  Pos(1,2)
  derecha
  {recorrer la calle hasta encontrar una esquina vacía o hasta terminar}
  mientras (HayFlorEnLaEsquina | HayPapelEnLaEsquina) & (PosAv < 10)
    mover
  {si la encontró depositar en ella una flor}
  si !HayFlorEnLaEsquina &  !HayPapelEnLaEsquina
    si HayFlorEnLaBolsa
      depositarFlor
    sino
      Informar(F)
fin
```

- Cuales son los casos en los que la evaluación de la proposición molecular que maneja la iteración da como resultado falso ?

- ¿Puede reemplazarse la selección anterior por la siguiente?:

```
si !HayFlorEnLaEsquina & !HayPapelEnLaEsquina & HayFlorEnLaBolsa
  depositarFlor
sino
  Informar(F)
```

**Ejemplo 4.5:** Programe al robot para que informe la cantidad de papeles que hay en la esquina(6,2) SIN modificar el contenido de la esquina.

Este problema es una variante del ejemplo 3.5, donde no se pide que se recojan los papeles sino sólo que informe la cantidad. Sabemos que para poder resolver esto será necesario juntar los papeles contando y luego depositar exactamente la cantidad de papeles recogidos. Notemos que no es lo mismo vaciar los papeles de la bolsa porque ella podría contener papeles ANTES de comenzar a recoger. El programa es el siguiente:

```
programa Cap4Ejemplo5
variables
  cantP : numero
comenzar
  iniciar
  Pos(6,2)

  {Indicar que aun no se ha recogido nada}
  cantP := 0
  mientras HayPapelEnLaEsquina
    comenzar
    tomarPapel
    cantP := cantP + 1
    fin
  {Ahora la esquina ya no tiene papeles}
  Informar (cantP)

  { Volver a poner los papeles en la esquina}
  repetir cantP
    depositarPapel
fin
```

Se ha utilizado una repetición para volver a poner los papeles en la esquina porque, luego de haberlos recogido, se conoce exactamente la cantidad de papeles que se quiere depositar. Además, para depositar no es necesario preguntar si hay papeles en la bolsa para satisfacer esta demanda porque se ha recogido la misma cantidad de papeles a través de la iteración Es más, suponiendo que originalmente no hubiera habido papeles en (6,2), *cantP* valdrá cero en cuyo caso el repetir no ejecutará ninguna instrucción.

- Propone otra forma de escribir el segmento de código que vuelve a poner los papeles en la esquina (último repetir). Que habría que tener en cuenta en este caso?

**Ejemplo 4.6**: Programe al robot para que informe la cantidad de flores que hay en cada una de las esquinas de la avenida 1.

Para resolver este problema alcanzará con una única variable que represente la cantidad de flores de la esquina actual. Cada vez que llega a una esquina, el robot inicializará la variable en cero, anotará en ella cada vez que logre recoger una flor y finalmente informará su valor. Esto se debe repetir para cada esquina de la avenida 1. El programa será el siguiente:

```
programa Cap5Ejemplo6
variables
  flores : numero
comenzar
  iniciar
  {Se recorrerán las primeras 9 esquinas}
  repetir 9
    comenzar
    {Indicar que aun no se ha recogido nada en esta esquina}
    flores := 0                                              (1)
    mientras HayFlorEnLaEsquina
      comenzar
      tomarFlor
      flores := flores + 1
      fin
    {Ahora la esquina ya no tiene flores}
    Informar(flores)
    {Pasar a la esquina siguiente}
    mover
    fin
  {Falta la esquina (1,10)}                                 (2)
  {Indicar que aun no se ha recogido nada en esta esquina}
  flores := 0
  mientras HayFlorEnLaEsquina
    comenzar
    tomarFlor
    flores := flores + 1
    fin
  {Ahora la esquina ya no tiene flores}
  Informar(flores)
fin
```

- ¿Qué ocurriría si la línea (1) fuera trasladada *antes* del repetir, es decir antes de comenzar la repetición? ¿Qué valores informaría?

- ¿Por qué es necesario procesar por separado la esquina (1,10)? Vea que aparece fuera de la repetición en la línea (2).

- ¿Cómo modificaría el programa anterior para que el robot también pueda informar para cada esquina, el número de calle y la cantidad de flores que contiene.

## 4.4 Conclusiones

Se han presentado varios ejemplos que muestran el uso de los dos tipos de datos que puede manejar el robot: valores numéricos y valores booleanos. A través de ellos se ha mostrado la forma de mejorar la potencia de las soluciones ofrecidas, permitiendo que el robot registre valores para un procesamiento posterior.

También se ha definido y ejemplificado el uso de los conectivos lógicos permitiendo manejar las estructuras de control selección e iteración a través de proposiciones moleculares.

**Nota: para realizar este apunte se utilizó ( con el permiso correspondiente) el material del curso de Ingreso a la Facultad de Informática de la UNLP. En el mismo se han realizado algunos agregados y modificaciones.**