



# UNIVERSIDAD NACIONAL DE HURLINGHAM

*Instituto de Tecnología e Ingeniería / Programación Estructurada*

## **PROGRAMAS Y CONTRATOS**

### CONSEJOS:

- Leer el enunciado en su totalidad y pensar en la forma de resolverlo ANTES de empezar a escribir código.
- Si un ejercicio no sale, se puede dejar para después y continuar con los ejercicios que siguen.
- Los ejercicios están pensados para ser hechos después de haber mirado la teórica correspondiente.
- Algunos de los ejercicios están tomados de las guías prácticas utilizadas en la materia de Introducción a la Programación de la Universidad Nacional de Quilmes por Pablo Ernesto "Fidel" Martínez López y su equipo. También Federico Aloí y Miguel Miloro, a su vez basada en las guías Ejercicios de Introducción a la Programación del CIU General Belgrano, elaboradas por Carlos Lombardi y Alfredo Sanzo, y Fundamentos de la Programación del Proyecto Mumuki. Agradecemos a todos los que nos ayudaron con su inspiración.
- Realizar en papel los ejercicios que así lo indiquen.
- Sí un ejercicio indica BIBLIOTECA significa que podrá ser utilizado en el parcial sin definirlo. Es útil mantener registro de dichos procedimientos en su carpeta.

## PRIMEROS PROGRAMAS

**ATENCIÓN:** No es necesario que escriba contratos en estos ejercicios. El objetivo es ir entendiendo una idea de los mismos. Se le indicará a partir de qué punto es necesario que comience a escribirlos.

### 1. Reemplazando bolitas

Escribir un programa que reemplace una bolita de color Roja con otra de color Verde en la celda actual. Pruebe su programa en la computadora, modificando el tablero inicial de forma que el programa funcione satisfactoriamente.

### 2. Moviendo bolitas

Escribir un programa que mueva una bolita de color Negro de la celda actual a la celda vecina al Este, dejando el cabezal en la celda lindante al Este.

### 3. Poniendo en vecinas

Escribir un programa que ponga una bolita de color Azul en la celda vecina al Norte de la actual.

### 4. ¿Qué pide el enunciado?

**EN PAPEL.** Los siguientes programas resuelven incorrectamente el ejercicio anterior de esta práctica. Para cada uno de ellos, explicar por qué no son correctos. Comparar con la solución dada para el ejercicio anterior.

a)

```
program {
  Mover(Norte)
  Poner(Azul)
}
```

b)

```
program {
  Poner(Azul)
  Mover(Este)
  Mover(Oeste)
}
```

c)

```
program {
  Mover(Norte)
  Poner(Verde)
  Mover(Sur)
}
```

d)

```
program {
  Poner(Azul)
  Mover(Norte)
  Mover(Sur)
}
```

### 5. Analizando propósitos

**EN PAPEL.** Dado el siguiente programa:

```
program {
  Poner(Verde)
  Sacar(Verde)
  Poner(Azul)
  Poner(Rojo)
}
```

Diversos estudiantes realizaron propuestas para redactar su propósito, y también un profesor realizó explicaciones sobre cada una de estas propuestas. Asociar cada propuesta de propósito para el mismo (indicadas con las letras A, B, etc.) con la explicación que resulta correcta para dicha propuesta (indicadas con los números 1, 2, etc.)

(A) Poner una bolita azul y luego una roja en la celda actual.	(1) No es un propósito, sino una descripción del funcionamiento. Para ser un propósito no debe preocuparse de los estados intermedios, solamente de la transformación final.
(B) Agregar una bolita azul y una roja.	(2) Es un propósito incompleto ya que no establece los colores de las bolitas que se agregan. El propósito debe establecer con precisión la transformación esperada.
(C) Pone una bolita verde y luego la saca, para a continuación poner una bolita azul y una roja.	(3) Es una enunciación correcta del propósito. El orden en que se agregan las bolitas es irrelevante, siempre que la celda actual finalice con una más de cada uno de los colores indicados.
(D) Agregar una bolita roja y una bolita azul en la celda actual.	(4) Es un propósito incompleto, ya que no establece dónde se agregan las bolitas en cuestión. El propósito debe establecer con precisión la transformación esperada.
(E) Agregar dos bolitas en la celda actual.	(5) Es una forma incorrecta de indicar la transformación esperada. Utiliza un lenguaje que sugiere un pensamiento operacional (o sea, centrado en las acciones individuales antes que en la transformación esperada).

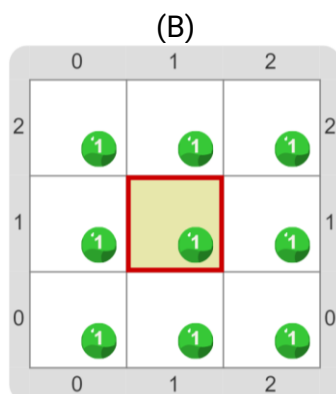
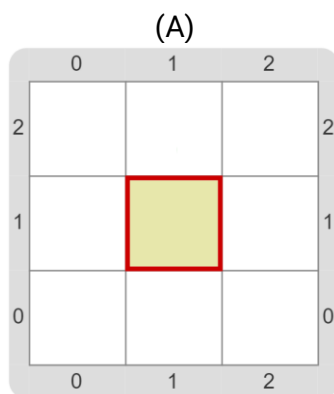
## PROGRAMAS QUE PUEDEN FALLAR

ATENCIÓN: A partir de este momento se pide que escriba los propósitos de su código, aunque no así las precondiciones de los mismos, cosa que iremos practicando en los siguientes ejercicios.

### 6. Cuadrados verdes

Escribir los siguientes programas:

- Uno que ponga un cuadrado de tamaño 3 con bolitas de color verde, con centro en la celda inicial (dejando el cabezal en dicha celda al finalizar)
  - ¿Qué ocurre si el tablero ya tenía bolitas verdes? ¿Es esto un problema, o el efecto obtenido es acorde al propósito?
  - ¿Qué ocurriría si desde la celda inicial no hay espacio para colocar las bolitas necesarias para que se cumpla el propósito?
- Uno que saque un cuadrado de tamaño 3 con bolitas de color verde (saca una bolita de cada celda), siendo la celda inicial el centro del cuadrado, dejando el cabezal en dicha celda al finalizar.
  - ¿Qué sucede si no hay al menos una bolita verde en cada una de las celdas necesarias? ¿Y si el tablero no tiene el tamaño adecuado?



Efectos de poner y sacar un cuadrado verde. Empezando con el Tablero (A), el programa a) termina en el Tablero (B), mientras que empezando con el Tablero (B) se obtiene el Tablero (A) al ejecutar el programa b).

### 7. ¿Y cuándo funciona?

EN PAPEL. Discutir en clase cuáles serían las precondiciones para programas cuyo propósito sea:

- Poner 1000 bolitas color Azul en la celda actual.

- b) Poner una bolita color Rojo en la celda lindante al Este de la celda actual y sacar una bolita Azul de la celda lindante al Oeste de la celda actual.
- c) Poner un rectángulo de bolitas Negras cuyo tamaño sea 3 filas y 5 columnas, centrado en la celda actual. Importante: no hay que escribir ningún programa, simplemente basta con discutir cuál es la precondition que esperarían de tales programas.

## 8. Analizando preconditiones

**EN PAPEL.** La solución propuesta por diversos estudiantes para el punto c, del ejercicio anterior, fue corregida por los docentes. Sin embargo, las correcciones se mezclaron, y hay que juntar cada corrección con su precondition correspondiente.

(A) Debe haber al menos una celda al Norte y otra al Sur, y dos celdas al Este y otras dos al Oeste de la celda actual.	(1) Es una precondition incorrecta, pues al no establecer dónde debe ubicarse el cabezal hay tableros de ese tamaño que no sirven.
(B) El rectángulo va a representar a una ventana en el dibujo de una casa.	(2) Es una precondition correcta, pues establece las condiciones mínimas necesarias para que el programa funcione, y define con precisión el conjunto de tableros iniciales que sirven.
(C) El tablero debe tener 3 o más filas, y 5 o más columnas, y el cabezal debe estar al menos a una celda de distancia de los bordes Norte y Sur, y a dos celdas de distancia de los bordes Este y Oeste.	(3) Es otra forma de enunciar correctamente la precondition: todos los tableros del conjunto que la satisfacen sirven, y todos los que no la satisfacen no sirven. Sin embargo, especificar las dimensiones del tablero es innecesario, pues son las distancias a los bordes lo relevante para que el programa no falle
(D) Debe haber lugar suficiente en las direcciones adecuadas.	(4) Es una precondition correcta pero insuficiente, pues todos los tableros que la satisfacen sirven, pero hay muchos tableros que sirven que no la satisfacen.
(E) Se van a colocar 35 bolitas de color Negro.	(5) Es una precondition insuficiente, pues es demasiado vaga y no permite determinar cuáles son los tableros que sirven.
(F) El tablero debe tener 3 filas y 5 columnas y el cabezal estar ubicado en el centro del mismo.	(6) No es una precondition, sino una aclaración. Una precondition debe hablar sobre los tableros iniciales y no sobre el dominio del problema.
(G) El tablero debe ser de 3 filas por 5 columnas.	(7) No es una precondition, sino una explicación de la tarea. Una precondition debe hablar sobre los tableros iniciales y no sobre las características de lo que se va a realizar.

## 9. Parece pero no es

A continuación hay una serie de preconditiones que un alumno escribió en un examen. Las mismas son todas incorrectas por el mismo motivo. ¿Puede adivinar por qué son incorrectas? Justifique claramente su respuesta.

- a) Se está escribiendo código Gobstones.
- b) El cabezal se encuentra en la celda actual.
- c) Existe una celda en el tablero.
- d) Existe un tablero.
- e) El cabezal existe y está en la celda actual.

## PROGRAMAS UN POCO MÁS COMPLEJOS

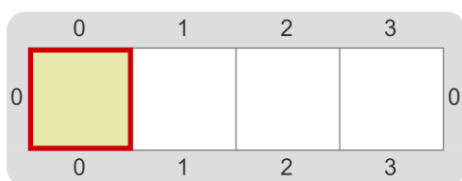
**ATENCIÓN:** A partir de este momento se pide que escriba los contratos completos, es decir, con propósito y preconditiones (y observaciones si las hubiera).

## 10. Arcoiris

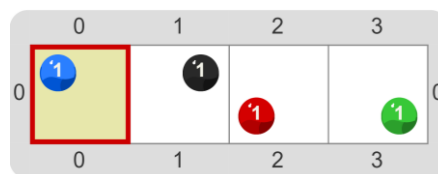
**EN PAPEL.** Escribir un programa que ponga un “arco iris”, poniendo una bolita Azul en la celda actual, una Negra en la celda siguiente al Este, una Roja en la siguiente al Este, y una Verde en la siguiente al Este, dejando el cabezal en la celda inicial.

Podemos ver un ejemplo en la figura, partiendo del tablero (a) se obtendría el (b):

Tablero inicial (a)



Tablero final (b)



Luego contestar las siguientes preguntas:

- ¿Qué sucedería en otros tableros iniciales?
- ¿Cuál es la precondition de este programa?
- ¿Puede darse otro programa diferente que cumpla el mismo contrato? PISTA: en lugar de poner las bolitas en la ida, considere si no lo puede hacer a la vuelta.
- Pruebe ahora sus programas en la máquina para verificar que funcionan, detectar y errores y corregir.

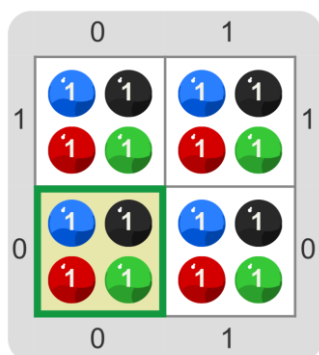
## 11. Sacando un cuadrado

Escriba un programa que saque del tablero un cuadrado multicolor de dos celdas de lado, donde la celda actual representa el vértice inferior izquierdo del mismo. Recuerde escribir primero el contrato del programa, y luego el código.

Considere las siguientes preguntas como guía para escribir su programa:

- ¿Que hace el programa? (Determina el propósito del programa)
- ¿Cuándo funciona tal cual se espera? (Determina la precondition del programa)
- ¿Cómo lo hace? (Determina el código del programa)

Tablero inicial



Tablero final

