



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Instituto de Tecnología e Ingeniería / Programación Estructurada

FUNCIONES CON PROCESAMIENTO Y VARIABLES

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.*

FUNCIONES CON PROCESAMIENTO

1. Mirando la celda vecina

Escribir la función **hayBolitas_EnCeldaAl_**, que, suponiendo que existe una celda lindante en la dirección dada, indica si la misma tiene o no bolitas del color indicado. Si no hay una celda lindante, hace BOOM.

2. Mirando la celda vecina, incluso si no hay vecina

BIBLIOTECA. Escribir la función **hayBolitas_AI_**, que indica si hay una celda lindante en la dirección indicada y la misma tiene bolitas del color dado. Si no hay celda lindante describe Falso.

3. Mirando en la celda al borde

Escribir la función **hayBolitas_EnElBorde_**, que indica si en la celda que se encuentra en el borde dado por la dirección, hay bolitas del color indicado.

4. Mirando en la fila o columna

Escribir la función **hayBolitas_Hacia_** que indica si en alguna de las celdas hacia la dirección dada (sin incluir la celda actual) hay bolitas del color dado.

5. Y volviendo a mirar en la fila o columna

Escribir la función **hayCeldaVacíaHacia_**, que indica si en alguna de las celdas hacia la dirección dada (sin incluir la celda actual) hay una que esté vacía.

6. Y si miramos el tablero

Escribir la función **hayAlgunaBolita_**, que indica si en alguna de las celdas del tablero existe una bolita del color dado.

7. Y volvemos a mirar el tablero

Escribir la función **hayAlgunaCeldaVacía**, que indica si alguna de las celdas del tablero está vacía.

VARIABLES

8. Copiamos una celda

Escribir el procedimiento **CopiarCeldaAl_**, que copia los contenidos de la celda actual a la celda lindante en la dirección dada. Note que se deben eliminar los contenidos originales que hubieran en la celda de destino.

9. Copiamos las esquinas /

Escribir el procedimiento **CopiarOrigenEnEsquinas** que copia en cada esquina los contenidos que hay en la celda actual (las 4 esquinas deben terminar con exactamente las mismas bolitas de cada color que había en la celda donde estaba originalmente el cabezal en el tablero inicial. La posición final del cabezal no es relevante).

10. ¡Y Dale!, ¡Y dale!, ¡Y dale Nova dale!!

La revisión de código sigue mal para Nova. Esta vez se trata de unos procedimientos que además de no tener contratos ni buenos nombres, algunos no andan y otros usan las variables de formas inadecuadas. Se pide entonces, encontrar los errores en los procedimientos que escribió Nova y justificar por qué son errores. Luego,

renombrar los procedimientos, variables y parámetros (y sus usos) de manera adecuada, para que el programa resulte legible, y escribir los contratos.

```
procedure P(p)
{ p := p + 1; Poner__Veces(Azul, p) }
procedure Q(p) {
  if(p /= 3) {v := 3}
  Poner__Veces(Azul, v)
}
procedure R()
{ Poner__Veces(Azul, v) Mover__Veces(Este, v) }
procedure S(p) {
  v := Este
  Mover__Veces(v, 5)
  v := 5
  Mover__Veces(Este, v)
}
```

¡Alguien debe enseñarle pronto a Nova lo difícil que es entender y corregir el código si no se escriben buenos nombres y buenos contratos!