



# Programación

Clase 8

Repetición simple

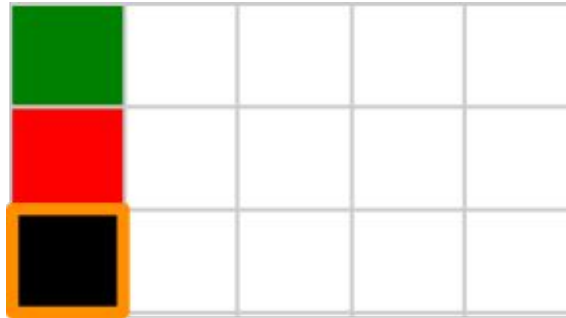
Universidad Nacional de Quilmes

Ejercitamos un poco



# Ejercicio

Realice el siguiente dibujo en QDraw donde el cabezal comienza en la esquina inferior izquierda.



# Solución

```
programa {
```

```
  /*
```

PROPÓSITO: Pinta la celda actual de negro, la superior de rojo y la de dos unidades más arriba de verde.

PRECONDICIÓN: Debe haber al menos dos celdas hacia arriba de la celda actual.

```
  */
```

```
  ▨ ↑ ▤ ↑ ▥ ↓ ↓
```

```
}
```



¿Y los procedimientos?



# Solución

```
procedimiento ColumnaMulticolor () {
```

```
    /*
```

PROPÓSITO: Pinta la celda actual de negro, la superior de rojo y la de dos unidades más arriba de verde.

PRECONDICIÓN: Debe haber al menos dos celdas hacia arriba de la celda actual.

```
    */
```

▨ ↑ ▨ ↑ ▨ ↓ ↓

```
}
```

```
programa {
```

```
    ColumnaMulticolor()
```

```
}
```



# Ejercicio

Realice ahora el siguiente dibujo.



# Solución

```
programa {  
    ColumnaMulticolor()  
    →  
    ColumnaMulticolor()  
    →  
    ColumnaMulticolor()  
    →  
    ColumnaMulticolor()  
    →  
    ColumnaMulticolor()  
}
```



# Ejercicio

Ahora hagamos este...



¿Dónde está el problema?



NO ES EFICIENTE REPETIR CÓDIGO...

entonces qué hacemos ??



# Repeticiones



# Repetición

La **repetición simple** es un **bloque de código** que se ejecutará de forma consecutiva un determinada cantidad de veces, es decir, se **repetirá un número fijo de veces**.

La sintaxis que utilizaremos serán las palabras “**repetir N veces**”, donde N es un **número natural** cualquiera, seguido de un **bloque de código** que se repetirá N veces.



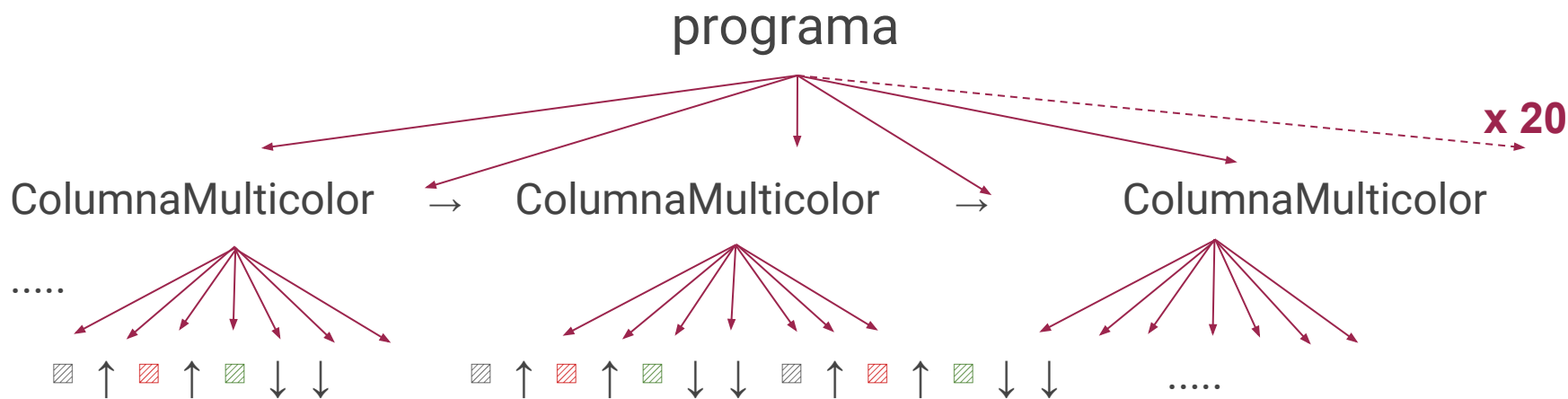
# Repetición: Ejemplo

```
programa {  
    repetir 20 veces {  
        ColumnaMulticolor()  
    }  
}
```

→



# Repetición: Árbol de ejecución

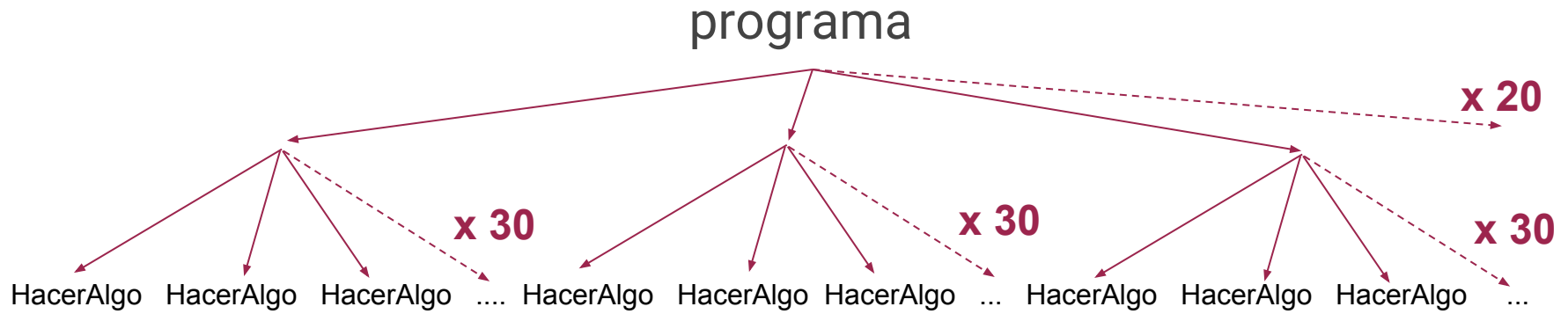


# Anidación de repeticiones

```
programa {  
    repetir 20 veces {  
        repetir 30 veces {  
            HacerAlgo()  
        }  
    }  
}
```



# Repetición: Árbol de ejecución



Recordemos entonces....

¡NO ANIDAR REPETICIONES!



# Qué sucede si anidamos repeticiones...

- Anidar repeticiones suele ser **causa de errores graves** en el código.
- Anidar repeticiones **dificulta leer el código y complica la comunicación.**
- Anidar repeticiones dificulta pensar en el árbol de ejecución.



# Cúal es la otra alternativa para no anidar ?

- La solución a la anidación consiste en utilizar correctamente los procedimientos.
- De esta forma es mucho más fácil razonar el problema, transmitir la idea, y visualizar el árbol de ejecución.



# Forma correcta

```
procedimiento HacerAlgo () {  
    repetir 30 veces {  
        HacerAlgo()  
    }  
}  
programa {  
    repetir 20 veces {  
        HacerAlgo()  
    }  
}
```



# Repeticiones y equivalencias



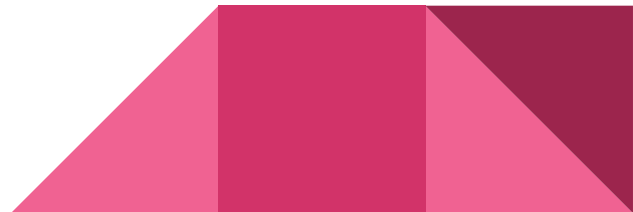
# Ambas son equivalentes

```
programa {  
    repetir 3 veces {  
        HacerAlgo()  
    }  
}
```



!!!Esta forma es más adecuada!!!!

```
programa {  
    HacerAlgo()  
    HacerAlgo()  
    HacerAlgo()  
}
```



# Ambas son equivalentes

```
programa {  
  repetir 1 veces {  
    HacerAlgo()  
  }  
}
```

```
programa {  
  HacerAlgo()  
}
```



!!!LA FORMA CORRECTA ES LA SIGUIENTE !!!

# Ejercitamos



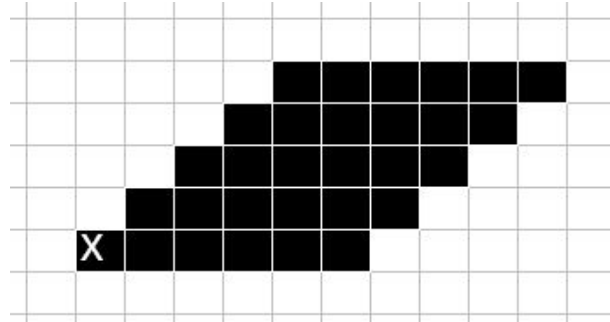
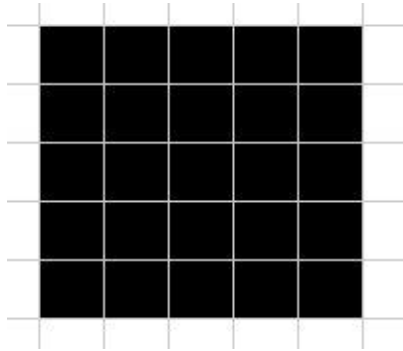
Recuerde:

El truco está en encontrar un patrón que se repita.



# Actividad 1

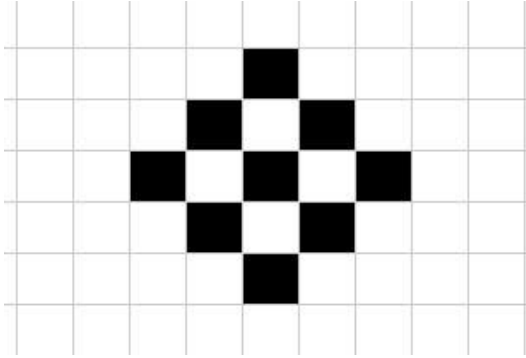
Implemente un procedimiento **A ) DibujarCuadrado** **2 ) Dibujarparalelogramo**



**Tips:** Siempre ingrese el Propósito y Precondición en TODOS sus programas y procedimientos

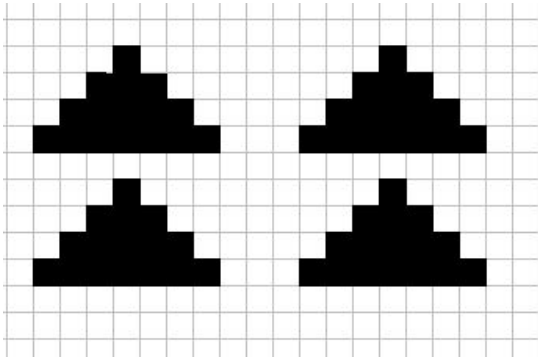
## Actividad 2

Implemente un procedimiento **DibujarRombo**. El cabezal comienza en el extremo inferior **derecho**.



## Actividad 3

Implemente un procedimiento [DibujarPiramides](#) que realice el dibujo a continuación. El cabezal comienza en la celda de la esquina inferior izquierda.



Ya puede realizar hasta el  
ejercicio 23 de la Guía!





# Programación

Clase 8

Universidad Nacional de Quilmes