



Programación

Repetición simple

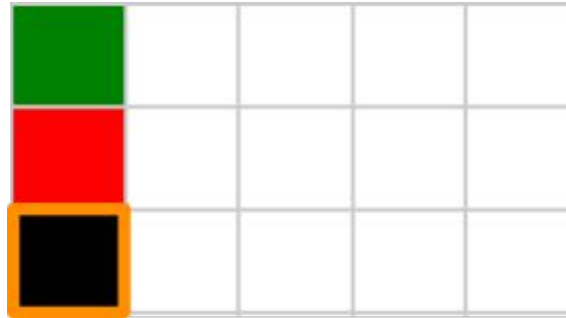
Universidad Nacional de Quilmes

Precalentando motores



Enunciado

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



Solución

```
programa {
```

```
/* PROPÓSITO: Dibujar una columna de tres colores: negro en la base, rojo y verde en el tope.
```

```
El cabezal inicia y finaliza en la base de la columna*/
```

```
    PintarNegro
```

```
    MoverArriba
```

```
    PintarRojo
```

```
    MoverArriba
```

```
    PintarVerde
```

```
    MoverAbajo
```

```
    MoverAbajo
```

```
}
```

¡No olvidar
documentar las
precondiciones!
En este ejemplo,
¿tiene? ¿cuáles son?

¿Y los procedimientos?



Solución utilizando procedimientos

```
programa {
```

```
  /* PROPÓSITO: Dibujar una columna multicolor. El cabezal inicia y finaliza en la base de la columna*/
```

```
    ColumnaMulticolor()
```

```
}
```

```
procedimiento ColumnaMulticolor () {
```

```
  /* PROPÓSITO: Dibujar una columna de tres colores: negro en la base, rojo y verde en el tope. El cabezal inicia y finaliza en la base de la columna*/
```

```
    PintarNegro
```

```
    MoverArriba
```

```
    PintarRojo
```

```
    MoverArriba
```

```
    PintarVerde
```

```
    MoverAbajo
```

```
    MoverAbajo
```

```
}
```



Extendemos el dibujo

Ahora realizar el siguiente dibujo:



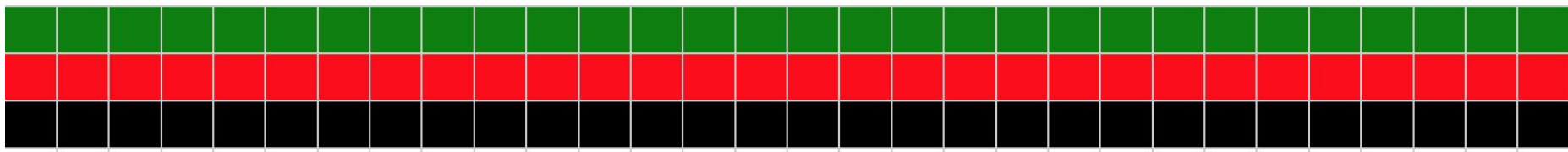
Acá debe quedar el
cabecal al terminar
el programa

Extendemos la solución del algoritmo anterior

```
programa {  
    /* PROPÓSITO: Dibujar un rectángulo multicolor. El cabezal inicia en la esquina  
    inferior izquierda y finaliza en la esquina inferior derecha del rectángulo */  
    ColumnaMulticolor()  
    MoverDerecha  
    ColumnaMulticolor()  
    MoverDerecha  
    ColumnaMulticolor()  
    MoverDerecha  
    ColumnaMulticolor()  
    MoverDerecha  
    ColumnaMulticolor()  
}
```


Continuamos avanzando en el dibujo

Ahora qué pasa si necesitamos extenderlo aún más de esta manera

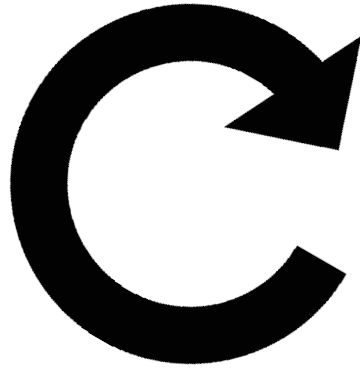


YA NO ES LEGIBLE REPETIR
TANTO CÓDIGO... ¿VERDAD?

Entonces ¿qué hacemos ?



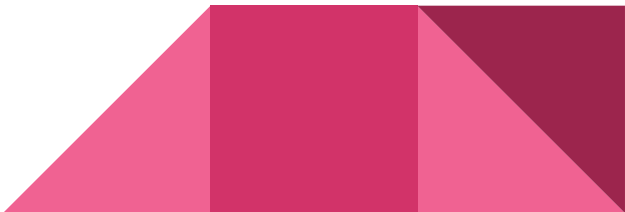
Repetición simple



Definición

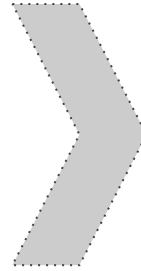
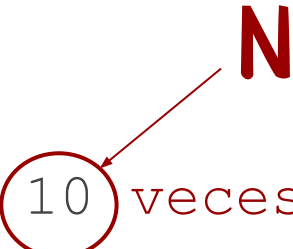
La **repetición simple** es un **bloque de código** que se ejecutará de forma consecutiva una determinada cantidad de veces, es decir, se **repetirá un número fijo de veces**.

La sintaxis que utilizaremos se compone de las siguientes palabras: “**repetir N veces**”, donde N es un **número natural**, seguido de un **bloque de código** entre llaves.



Ejemplo

```
programa {  
  repetir 10 veces {  
    PintarNegro  
    MoverDerecha  
  }  
}
```



Bloque de código
que se repite 10
veces

Pero ¿qué hace este programa?

Solución final: usamos repeticiones

¡Justo lo que necesitábamos!

Aplicamos esta nueva herramienta como solución del ejercicio anterior:



```
programa {
```

```
/* PROPÓSITO: Dibujar una guarda multicolor. El cabezal inicia en la esquina inferior izquierda  
y finaliza en la esquina inferior derecha de la guarda
```

```
PRECONDICIÓN: debe haber al menos 2 celdas hacia arriba y 29 celdas hacia la derecha  
desde el inicio del cabezal.*/
```

```
    repetir 29 veces {  
        ColumnaMulticolor()  
        MoverDerecha  
    }  
    ColumnaMulticolor()  
}
```

¡Analizar cuál es el patrón que se repite!
Dicho patrón es el bloque de código que conformará la repetición

Responsabilidad profesional

Dado que se va ampliando nuestro repertorio de **estructuras de control**, es decir, aquellas que permiten modificar y controlar la secuencia/flujo de las instrucciones planteadas en el **algoritmo**, debemos ser responsables en la manera en que las utilizamos. Como programadores/as debemos **avaluar la calidad** de nuestros desarrollos. Es por ello, que se cuenta con **buenas y malas prácticas de programación**.

Siendo que acabamos de agregar la instrucción **repetir**, analicemos las prácticas que debemos tener en cuenta al momento de utilizarla. **Veamos un ejemplo:**

```
programa{  
  /* Propósito: ....  
  Precondición: ..... */  
  repetir 2 veces {  
    repetir 2 veces {  
      PintaVerde  
      MoverArriba  
    }  
  }  
}
```

- 1) ¿Qué notas raro?
- 2) ¿Te resulta fácil entender cuál es el propósito?
- 3) ¿Te parece una buena o mala práctica? ¿Por qué?
- 4) ¿Cómo lo solucionarías?

Errores comunes: analicemos el código de ejemplo

Respondamos las preguntas que quedaron pendientes:

- 1) ¿Qué notas raro?
- 2) ¿Te resulta fácil entender cuál es el propósito?
- 3) ¿Te parece una buena o mala práctica? ¿Por qué?
- 4) ¿Cómo lo solucionarías?

2) No. No sabemos cuál es. Y el código quizá no expresa lo que se necesita.



```
programa{  
  /* Propósito: ....  
  Precondición: ..... */
```

```
    repetir 2 veces {  
      repetir 2 veces {  
        PintarVerde  
        MoverArriba  
      }  
    }  
}
```

**1) Resaltado
en rojo**



3) MALA PRÁCTICA.
Se llama **ANIDAR REPETICIONES**. No debemos anidar las instrucciones



4) SOLUCIÓN

1. Primero identificar cuál es el propósito.
2. Luego, buscar el patrón que se repite en el código
3. Reescribir el código utilizando buenas prácticas que reflejen el patrón encontrado y su propósito.

Errores comunes: soluciones aplicando buenas prácticas (Ej1)

Veamos ejemplos de dibujos, con sus respectivas soluciones, que sí aplican buenas prácticas de programación.

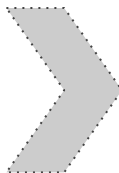
Propósito: Lo que se necesita es dibujar una sólo línea vertical de color verde de 4 de alto



```
programa{  
    repetir 2 veces {  
        repetir 2 veces {  
            PintarVerde  
            MoverArriba  
        }  
    }  
}
```



Unificar la cantidad de repeticiones en una sola.
No pintar de a 2 celdas, sino las 4 celdas en una única repetición.



El patrón: **PintarVerde y MoverArriba** se repite 3 veces. El último PintarVerde completa la 4ta celda.
Se mueve 3 veces y pinta 4 veces.

```
programa{  
    /*PROPÓSITO: dibujar una línea vertical verde de 4 de alto. El cabezal inicia en el extremo inferior y termina en el extremo superior.  
    PRECONDICIÓN: Debe haber 3 celdas hacia arriba desde el inicio del cabezal. */  
  
    repetir 3 veces {  
        PintarVerde  
        MoverArriba  
    }  
    PintarVerde  
}
```



Errores comunes: soluciones aplicando buenas prácticas (Ej2)



Escribir 2 veces consecutivas la instrucción **repetir** en el mismo bloque de código

```
procedimiento VolverAlInicio() {  
/*...*/  
    repetir 2 veces {  
        MoverAbajo  
    }  
    repetir 2 veces {  
        MoverIzquierda  
    }  
}
```



MALA PRÁCTICA:
así como anidar,
tampoco debemos
colocar **2 repetir
consecutivos** en el
mismo bloque de
código



Solución: nuevamente,
Descomponer en
procedimientos

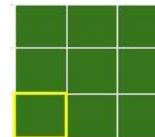
```
procedimiento VolverAlInicio(){  
/*...*/  
    Bajar2veces ()  
    Volver2veces ()  
}
```



```
procedimiento Bajar2veces{  
/*...*/  
    repetir 2 veces {  
        MoverAbajo  
    }  
}  
procedimiento Volver2veces{  
/*...*/  
    repetir 2 veces {  
        MoverIzquierda  
    }  
}
```

Errores comunes: soluciones aplicando buenas prácticas (Ej3)

Ejemplo que contiene las 2 situaciones de malas prácticas que vimos: **anidaciones** y **repetir consecutivos**



```
programa {
```

*/*PROPÓSITO: dibujar un cuadrado verde de 3x3. El cabezal inicia en la esquina inferior izquierda y finaliza en la esquina superior derecha.*

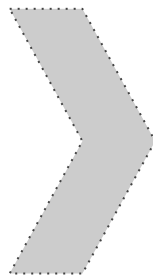
PRECONDICIÓN: debe haber 2 celdas hacia la derecha y 2 hacia arriba desde el inicio del cabezal/*

```
    repetir 2 veces {  
        repetir 2 veces {  
            PintarVerde  
            MoverIzquierda  
        }  
        repetir 2 veces {  
            MoverDerecha  
        }  
        MoverArriba  
    }  
    repetir 2 veces {  
        PintarVerde  
        MoverIzquierda  
    }
```

¡Código poco legible y difícil de continuar!



Nuevamente... sí, procedimientos



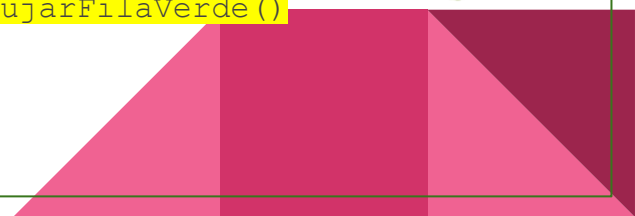
(Revisar los colores para comprender la mejora)

```
programa {
```

*/*PROPÓSITO: dibujar un cuadrado verde de 3x3. El cabezal inicia en la esquina inferior izquierda y finaliza en la esquina superior derecha.*

PRECONDICIÓN: debe haber 2 celdas hacia la derecha y 2 hacia arriba desde el inicio del cabezal/*

```
    repetir 2 veces {  
        DibujarFilaVerde()  
        VolverAlInicioDeFila()  
        MoverArriba  
    }  
    DibujarFilaVerde()  
}
```



¡Repasemos un punto importante!



Retomemos el programa de ejemplo del slide 17 para notar la siguiente diferencia:

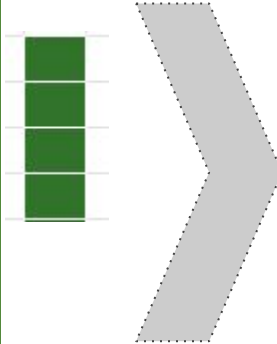
¡En el propósito menciona **4** celdas pero en la instrucción "repetir", el "N" es **3**! (y por lo tanto también la precondition.)
¿Por qué?

```
programa{
```

```
/*PROPÓSITO: dibujar una línea vertical  
verde de 4 celdas de alto. El cabezal inicia en  
el extremo inferior y termina en el extremo  
superior de la línea.
```

```
PRECONDICIÓN: Debe haber 3 celdas  
hacia arriba del inicio del cabezal. */
```

```
    repetir 3 veces {  
        PintarVerde  
        MoverArriba  
    }  
    PintarVerde  
}
```



Prestar atención a los patrones que se repiten!

No confundir la cantidad de celdas que se quieren pintar con la cantidad de veces que se repite el patrón (bloque de código interno del repetir)



Resumiendo ...

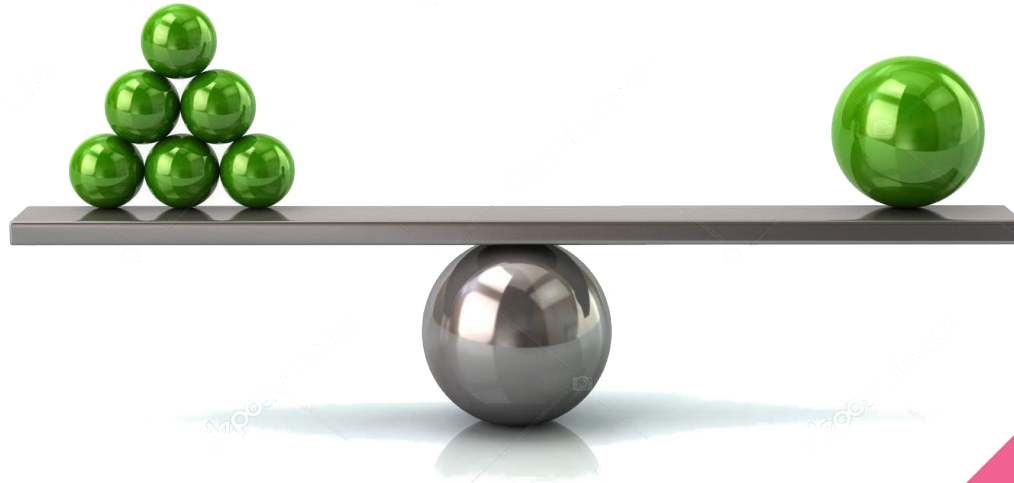
Tanto anidar, como escribir código repetido de manera consecutiva, genera los siguientes problemas:

- Es propenso a cometer errores graves
- Dificultad para leer el código, complicando así comunicación. Código poco legible
- Dificultad para entender si el código cumple con el propósito
- Código más largo y poco eficiente



Y por último: ¡Prestar atención a los límites del dibujo!

Equivalencias



Comparamos códigos equivalentes

A continuación se muestran ejemplos de programas que cumplen con el mismo propósito, es decir que su código es equivalente, pero una versión refleja la manera inadecuada y la otra, utiliza las buenas prácticas.

Ejemplo 1:

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



**Ya contamos con una
instrucción que realiza
esta acción.
¡Aprovecharla entonces!**



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



Comparamos códigos equivalentes

Ejemplo 2:

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



**No hay repetición si se
ejecuta sólo una vez**



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



Ejercicio para precalentar



Recordar que....



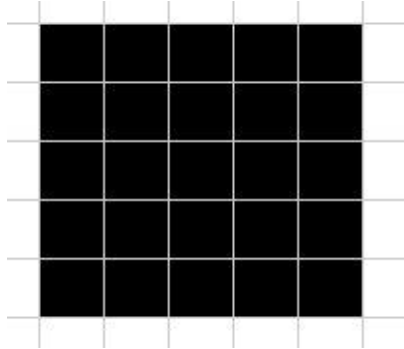
El truco está en encontrar el
patrón adecuado



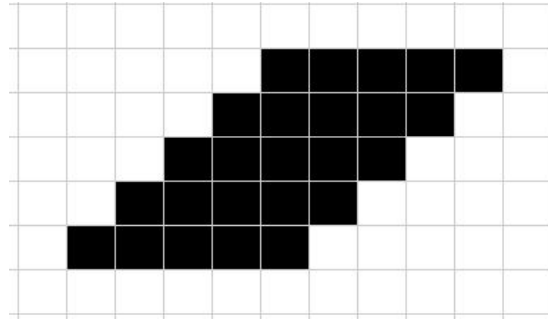
Actividad

Definir los siguientes procedimientos (con su respectiva documentación), que dibuje las siguientes figuras:

DibujarCuadrado

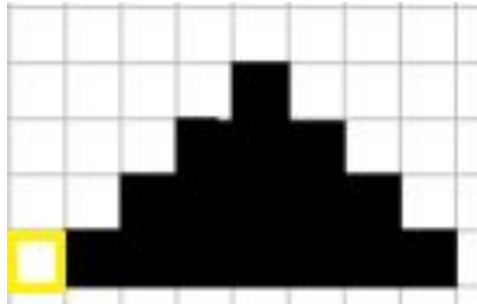


DibujarParalelogramo



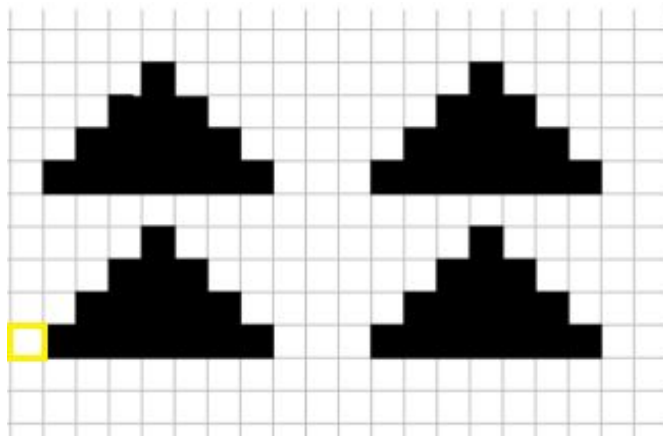
Actividad 2

Definir el procedimiento [DibujarPiramide](#) que realice el dibujo a continuación. El cabezal comienza a la izquierda de la esquina inferior izquierda de la pirámide.



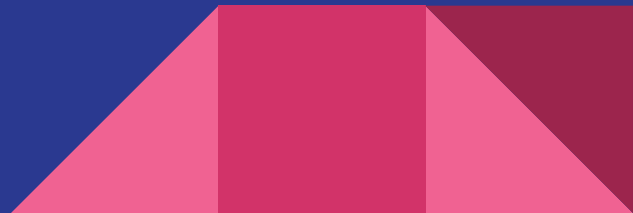
Actividad 3

Definir el procedimiento [DibujarCuatroPiramides](#) que realice el dibujo a continuación. El cabezal comienza en la posición pintada de amarillo que se muestra en la imagen.



Para reflexionar...

"Si todo te da igual,
estás haciendo mal las
cuentas"





Programación

Clase 4

Repetición simple

Universidad Nacional de Quilmes