

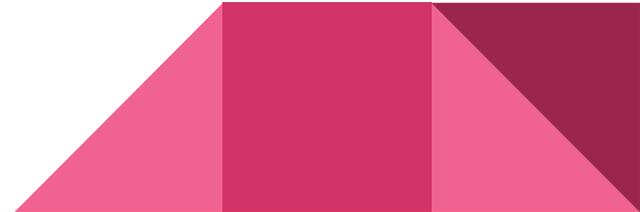
Programación

Clase 11

Parámetros

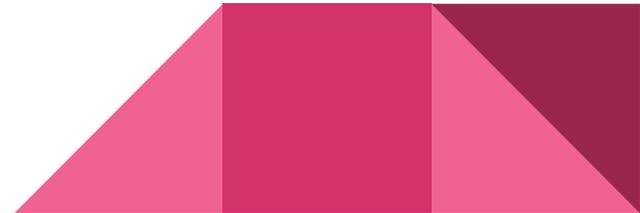
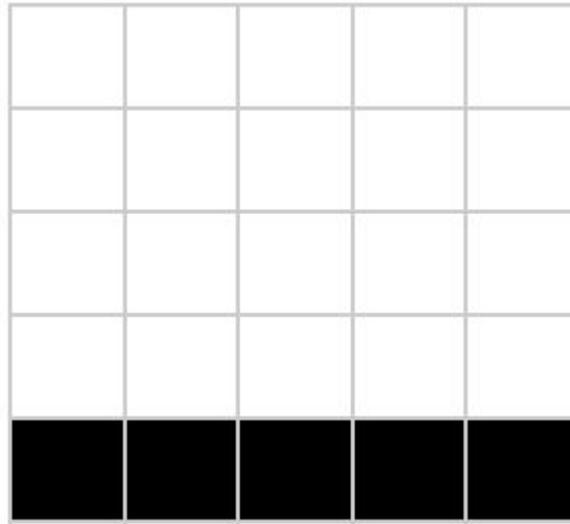
Universidad Nacional de Quilmes

Ejercitamos un poco



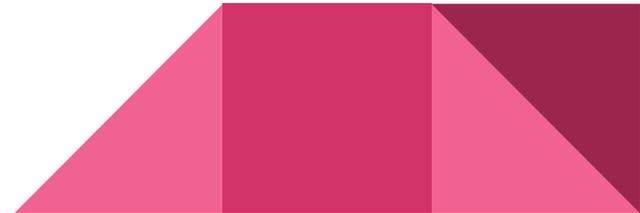
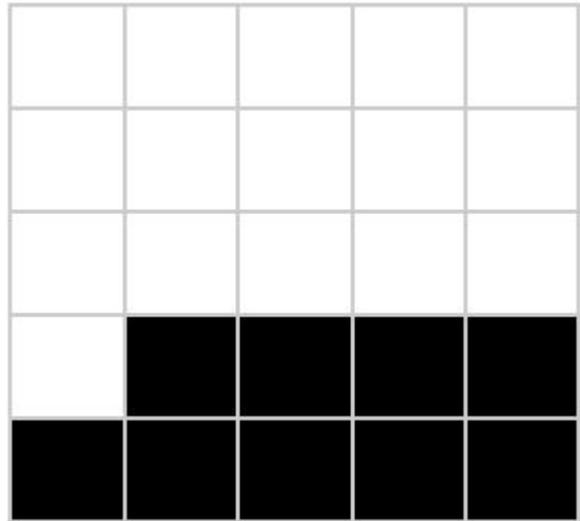
Ejercicio

Comenzando con el cabezal en la esquina inferior izquierda sobre un tablero vacío, realicemos el siguiente dibujo:



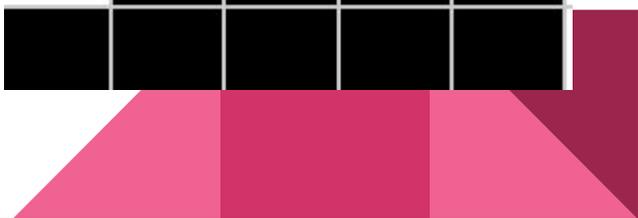
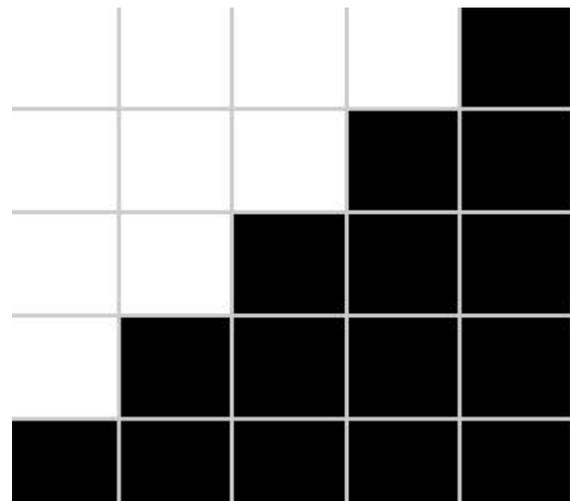
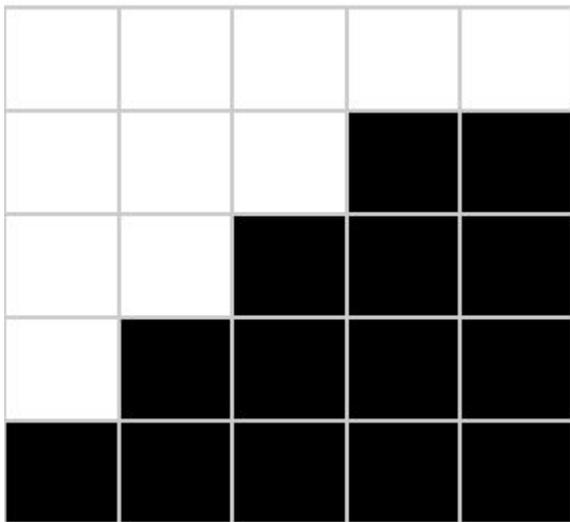
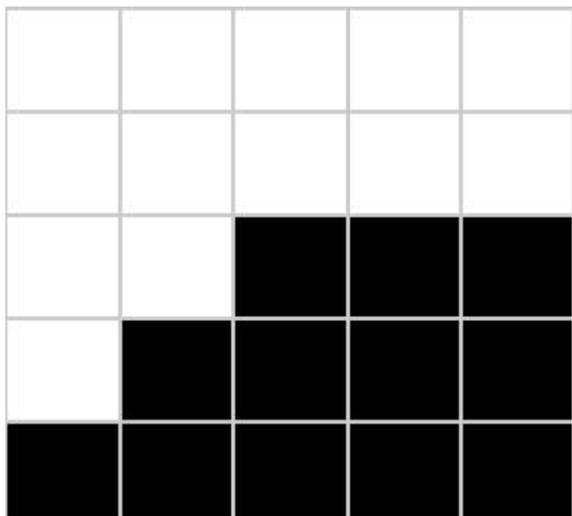
Ejercicio

Idem, pero ahora para este dibujo:



Ejercicio

Por último, hagámoslo para estos tres:



Solución

```
programa {
```

```
/* Propósito: Dibujar una escalera como la de la figura
```

```
Precondición: Tenemos disponibles al menos 6 celdas hacia arriba y al menos 6  
hacia la derecha.*/
```

```
    DibujarEscalonDeLargo5YSubir()
```

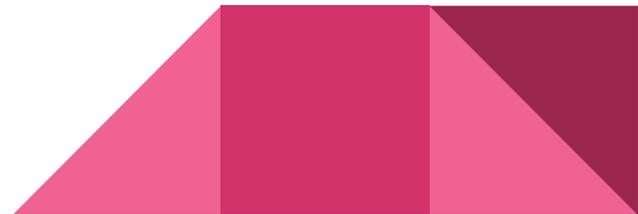
```
    DibujarEscalonDeLargo4YSubir()
```

```
    DibujarEscalonDeLargo3YSubir()
```

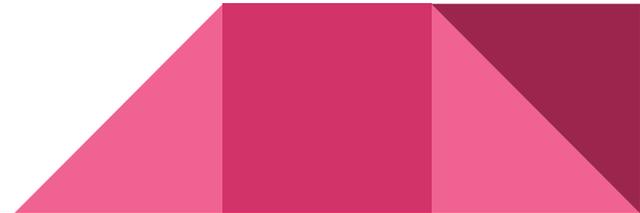
```
    DibujarEscalonDeLargo2YSubir()
```

```
    ▨
```

```
}
```



¿Qué encontramos en los
procedimientos?



Solución

```
procedimiento DibujarEscalonDeLargo5YSubir () {  
  repetir 4 veces {  
    ▨→  
  }  
  ▨  
  repetir 4 veces {  
    ←  
  }  
  →↑  
}
```

```
procedimiento DibujarEscalonDeLargo4YSubir () {  
  repetir 3 veces {  
    ▨→  
  }  
  ▨  
  repetir 3 veces {  
    ←  
  }  
  →↑  
}
```



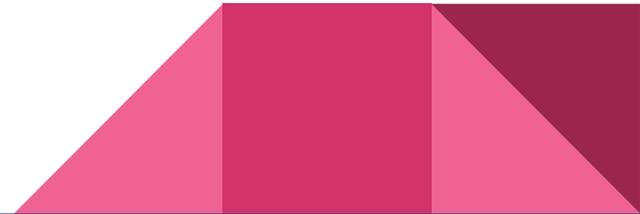
Solución

```
procedimiento DibujarEscalonDeLargo3YSubir () {  
  repetir 2 veces {  
    ▨→  
  }  
  ▨  
  repetir 2 veces {  
    ←  
  }  
  →↑  
}
```

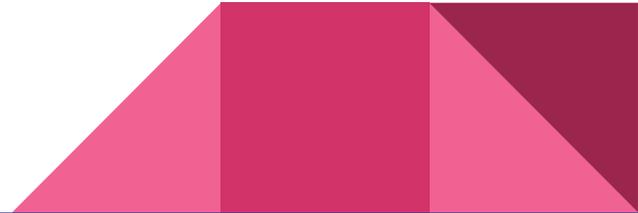
```
procedimiento DibujarEscalonDeLargo2YSubir () {  
  repetir 1 veces {  
    ▨→  
  }  
  ▨  
  repetir 1 veces {  
    ←  
  }  
  →↑  
}
```



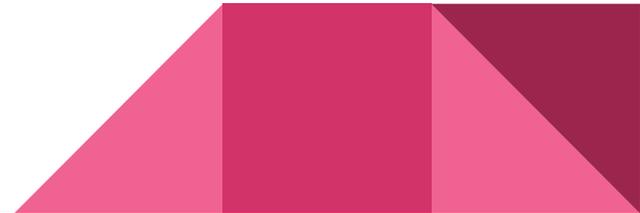
Son como muy parecidos, pero algo
cambia....



¿ Y si pudiéramos usar algo genérico,
que sirva siempre?



Parámetros



Parámetros

Un **parámetro** es un **nombre** al cual se asociará, al **momento de ejecución**, un **valor** dado, por aquel que **llama a un procedimiento**.

Nos permite pasar a un procedimiento un valor (un número) específico, y reutilizar un mismo procedimiento pasando distintos valores en cada llamada.



Parámetros

El **nombre** de un parámetro es una **palabra sin espacios** y que se anteceden del símbolo **\$**. El mismo se **declara** dentro de los **paréntesis** que siguen al nombre de un **procedimiento en su definición**.

El nombre debería ser lo más significativo posible con respecto al valor que intenta representar.

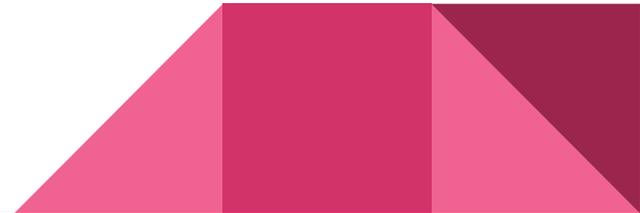
**PARA USAR UN PARÁMETRO EN UN PROCEDIMIENTO,
EL MISMO DEBE ESTAR DECLARADO EN SU
DEFINICIÓN.**



Parámetros

Una vez declarado un parámetro, éste se puede usar en un procedimiento como si fuera un valor más.

En QDraw todos los parámetros son números, por tanto se pueden usar como número para una repetición.



Parámetros: Ejemplo

```
procedimiento DibujarEscalonYSubir ($largoEscalon) {  
  repetir $largoEscalon veces {  
    ▨ →  
  }  
  repetir $largoEscalon veces {  
    ←  
  }  
  →↑  
}
```

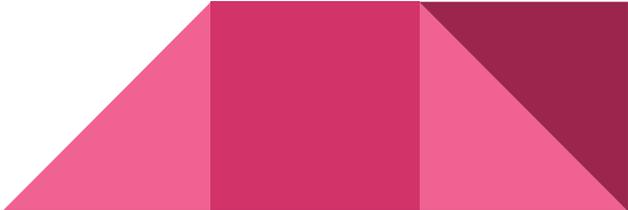


Argumento

Si un procedimiento declara un parámetro, en su **llamada** hay que **indicarle qué valor** va a tomar dicho parámetro. Ese valor se conoce como **argumento**.

Para esto, al momento de realizar la llamada, tras el nombre del procedimiento a llamar y **dentro de los paréntesis** que siguen, se debe colocar el **nombre del parámetro** seguido de un **signo igual** y el **valor** a que tomará el mismo.

SI UN PROCEDIMIENTO DECLARA UN PARÁMETRO SIEMPRE HAY QUE DARLE UN ARGUMENTO.



Argumentos: Ejemplo

```
programa {
```

```
    DibujarEscalonYSubir($largoEscalon = 4)
```

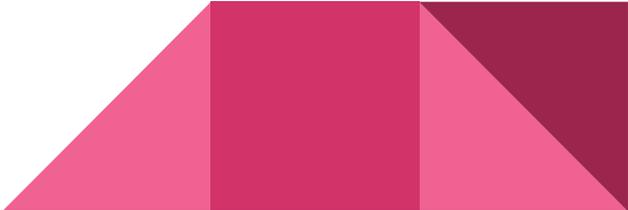
```
    DibujarEscalonYSubir($largoEscalon = 3)
```

```
    DibujarEscalonYSubir($largoEscalon = 2)
```

```
    DibujarEscalonYSubir($largoEscalon = 1)
```

```
    ▨
```

```
}
```



Árbol de ejecución con parámetros



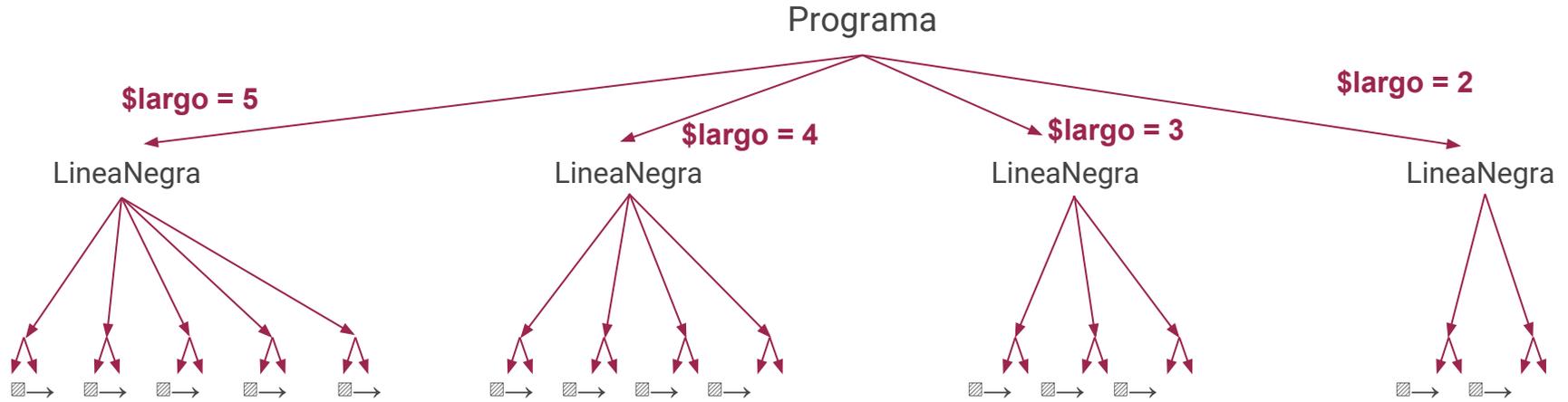
Ejemplo más sencillo

```
procedimiento LineaNegra ($largo) {  
  repetir $largo veces {  
    ▨ →  
  }  
}
```

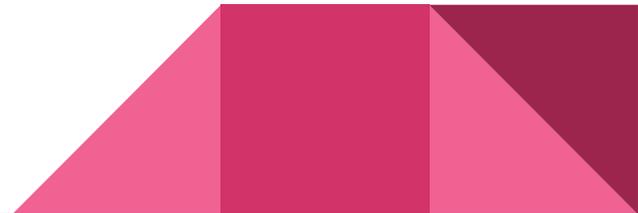
```
programa {  
  LineaNegra($largo = 5)  
  LineaNegra($largo = 4)  
  LineaNegra($largo = 3)  
  LineaNegra($largo = 2)  
}
```



Parámetros: Árbol de ejecución



Alcance



Alcance

El **alcance** de un parámetro comprende el **lugar** en donde podemos **utilizar** efectivamente el parámetro.

Un parámetro solo puede utilizarse dentro del procedimiento que lo declara, y en ningún otro procedimiento.

Dos parámetros en distintos procedimientos pueden tener el mismo nombre, pero no son el mismo parámetro.



Alcance: Ejemplo distintos nombres

```
procedimiento LineaNegraDerecha ($largo1) {
```

```
  repetir $largo1 veces {
```

```
    ▣→
```

```
  }
```

```
  repetir $largo1 veces {
```

```
    ←
```

```
  }
```

```
}
```

```
procedimiento LineaNegraArriba ($largo2) {
```

```
  repetir $largo2 veces {
```

```
    ▣↑
```

```
  }
```

```
  ▣
```

```
  repetir $largo2 veces {
```

```
    ↓
```

```
  }
```

```
}
```



Alcance: Ejemplo mismo nombre

```
procedimiento LineaNegraDerecha ($largo) {
```

```
  repetir $largo veces {
```

```
    ▣→
```

```
  }
```

```
  repetir $largo veces {
```

```
    ←
```

```
  }
```

```
}
```

```
procedimiento LineaNegraArriba ($largo) {
```

```
  repetir $largo veces {
```

```
    ▣↑
```

```
  }
```

```
  repetir $largo veces {
```

```
    ↓
```

```
  }
```

```
}
```



Parámetros como argumentos

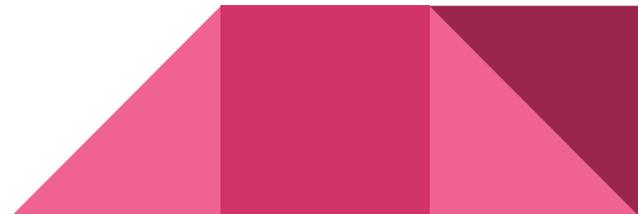


Parámetros como argumentos

Dentro de un procedimiento, **un parámetro actúa como un número.**

Por tanto, puede colocarse como el **valor en un repetición**

Como es un número, **puede usarse como argumento de un parámetro,**
como cualquier otro número.



Parámetros como argumentos: Ejemplo

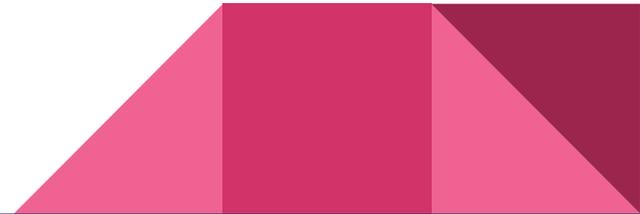
```
procedimiento DibujarCruzNegra ($lado) {  
  → LineaNegraDerecha($largo = $lado) ←  
  ↑ LineaNegraArriba($largo = $lado) ↓  
  ←LineaNegraIzquierda($largo = $lado) →  
  ↓ LineaNegraAbajo($largo = $lado) ↑  
  ▨  
}
```

```
programa {  
  DibujarCruzNegra ($lado = 3)  
}
```



¿Cuál es la precondición de este programa?

Múltiples parámetros

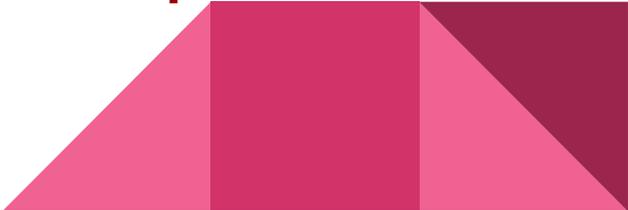


Múltiples parámetros

Un procedimiento puede tener **tantos parámetros como se desee**, no solo uno.

Para definir más de una parámetro **basta separar con coma** cada uno de ellos dentro de los paréntesis en la definición.

Al momento de realizar la llamada al procedimiento cada uno de los parámetros definidos debe tener un valor asociado. **No puede haber parámetros a los que no se les dé valor.**



Múltiples parámetros: Ejemplo

```
procedimiento LNegra ($largo, $salto) {
```

```
  repetir $salto veces {
```

```
    ▨ ↓
```

```
  }
```

```
  repetir $largo veces {
```

```
    ▨ →
```

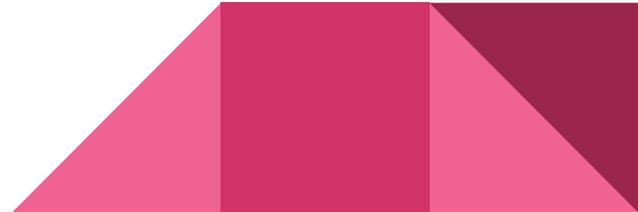
```
  }
```

```
}
```

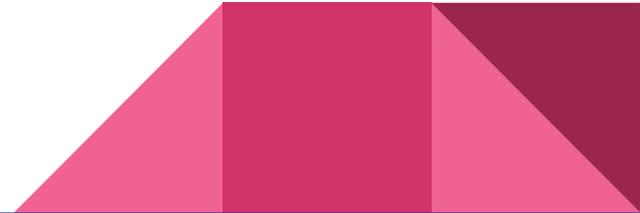
```
programa {
```

```
  LNegra($largo = 3, $salto = 5)
```

```
}
```



Errores comunes



No se define el parámetro en el procedimiento

procedimiento LineaNegra () {
 repetir **\$largo** veces {
 ▨ →
 }
}

programa {
 LineaNegra(**\$largo = 5**)
}



Se define un parámetro pero no se usa

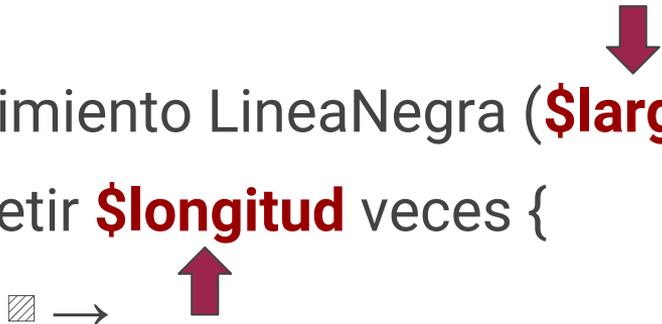
```
procedimiento LineaNegra ($largo) {  
  repetir 7 veces {  
      
  }  
}
```

```
programa {  
  LineaNegra($largo = 5)  
}
```



El nombre definido y el nombre usado no coinciden

```
procedimiento LineaNegra ($largo) {  
  repetir $longitud veces {  
    ▨ →  
  }  
}
```



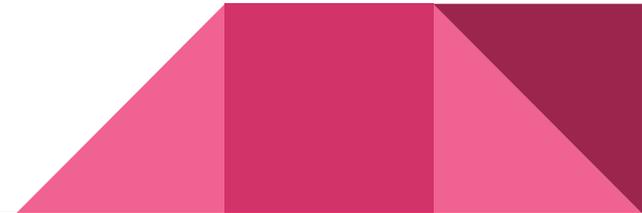
```
programa {  
  LineaNegra($largo = 5)  
}
```



El nombre definido y el nombre en la llamada no coinciden

```
procedimiento LineaNegra ($largo) {  
  repetir $largo veces {  
    ▣ →  
  }  
}
```

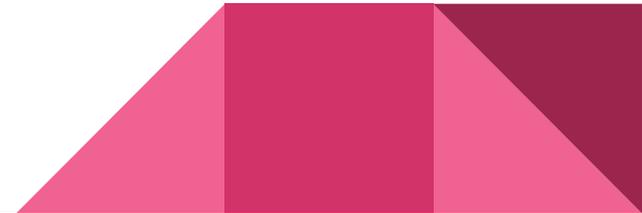
```
programa {  
  LineaNegra($long = 5)  
}
```



Hay un parámetro definido, pero en el llamado no se le da valor

```
procedimiento LineaNegra ($largo) {  
  repetir $largo veces {  
    ▨ →  
  }  
}
```

```
programa {  
  LineaNegra()  
}
```



El procedimiento no define parámetro, pero en la llamada se le asigna valor a uno.

```
procedimiento LineaNegraDe5 () {  
  repetir 5 veces {  
    ▣ →  
  }  
}
```

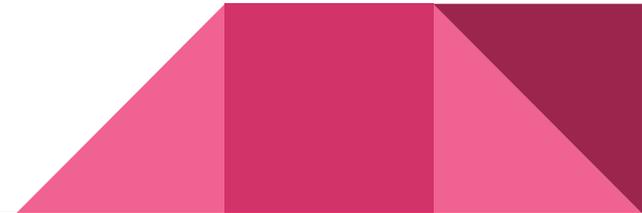
```
programa {  
  LineaNegra($largo = 5)  
}
```



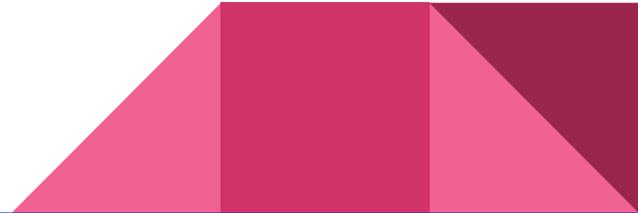
Se define más de un parámetro, pero solo se les dá valor a algunos de ellos

```
procedimiento LNegra ($largo, $salto) {  
  repetir $salto veces {  
    ▨ ↓  
  }  
  repetir $largo veces {  
    ▨ →  
  }  
}
```

```
programa {  
  LNegra($salto = 3)  
}
```

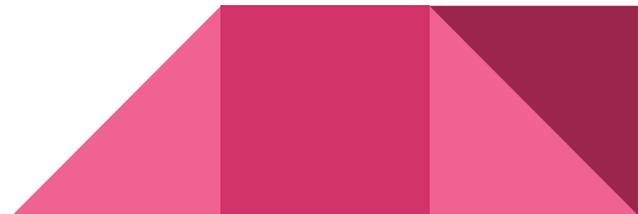
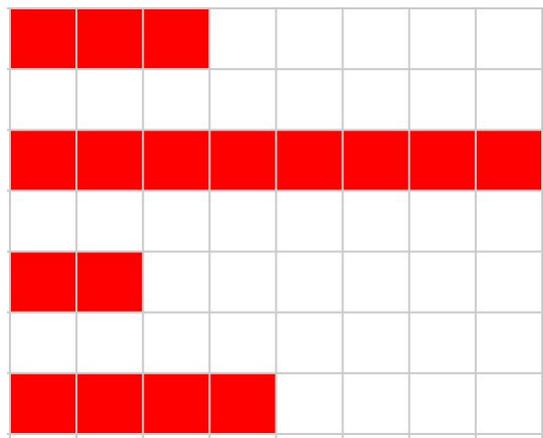


Ejercitamos



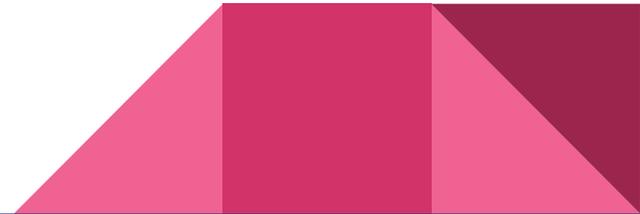
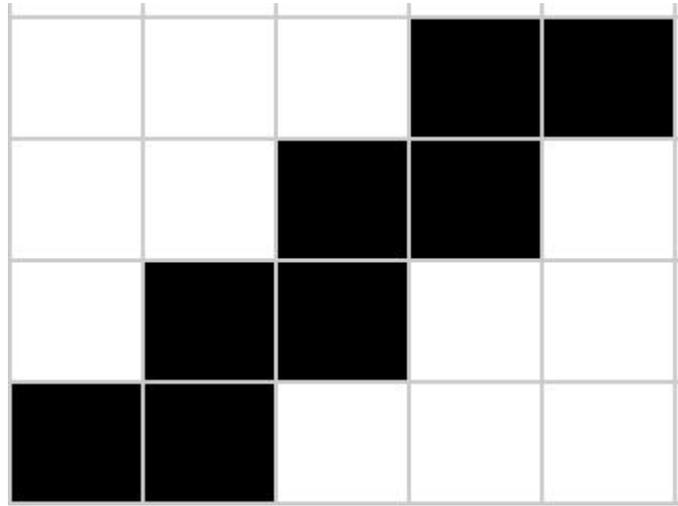
Ejercicio

Utilizando parámetros, escriba un procedimiento capaz de realizar las líneas de la figura. Escriba un programa en donde se utilice dicho procedimiento para generar el dibujo mostrado, considerando el cabezal en la esquina inferior izquierda y el tablero vacío al inicio del programa.



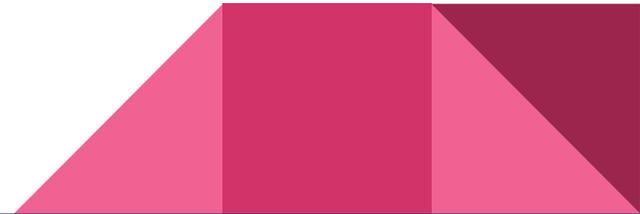
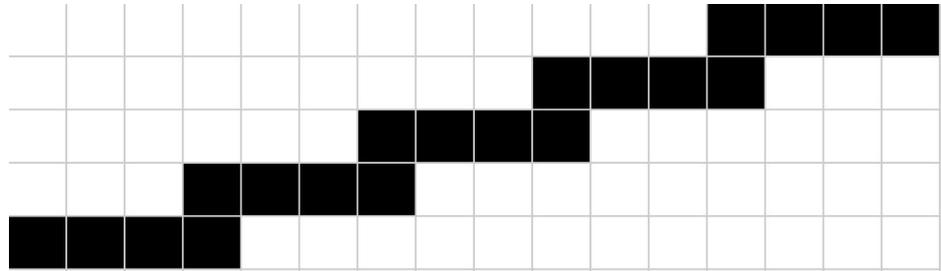
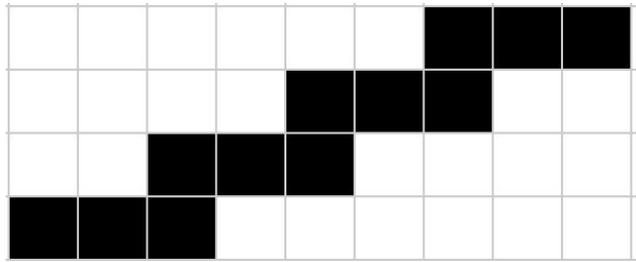
Ejercicio

Realice un procedimiento DibujarEscalera capaz de dibujar la siguiente escalera.



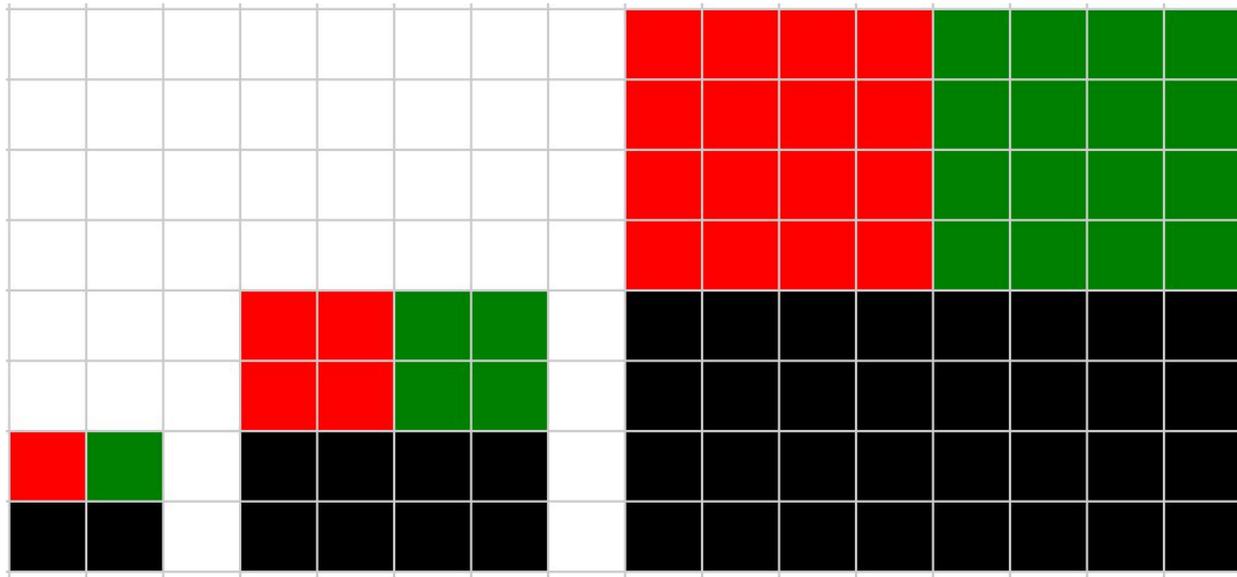
Ejercicio

Agregue parámetros a los procedimientos realizados en el ejercicio anterior de forma de poder dibujar las siguientes escaleras usando los mismos procedimientos.



Ejercicio

Realice utilizando parámetros un procedimiento capaz de dibujar las siguientes figuras.



Programación

Clase 6

Universidad Nacional de Quilmes