

# Programación

Clase 2

Introducción a la sintaxis estricta

Universidad Nacional de Quilmes

# Introducción a QDraw



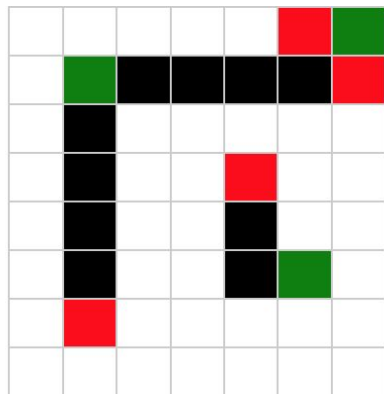
# Lenguaje QDraw

Utilizaremos este pequeño lenguaje para realizar programas. Nuestro autómeta será un cabezal que se mueve sobre un tablero (una hoja cuadriculada) pintando los casilleros cuando se le indica.



# QDraw: Tablero

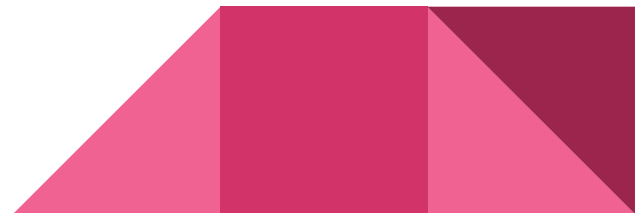
El tablero consiste en una superficie de tamaño variable (ancho y alto) con celdas. Cada celda puede estar en blanco o pintada de algún color (sólo se permiten los colores Negro, Rojo y Verde).



**Nota.** El tamaño del tablero se expresa en:  
**Columnas x Filas**  
(Ancho x Alto)

Ejemplo:

Un tablero de 7x8 con algunas celdas pintadas de distintos colores.



## QDraw: Cabezal

Podemos pensar en el cabezal como una lapicera con puntas de múltiples colores que se posiciona sobre una y solo una celda del tablero al mismo tiempo. El cabezal puede pintar dicha celda, o despintarla. También puede moverse hacia arriba, abajo, izquierda o derecha para desplazarse a una nueva celda.



# QDraw

## Set limitado de instrucciones:

- MoverDerecha      Mueve el cabezal una celda a la derecha.
- MoverArriba      Mueve el cabezal una celda hacia arriba.
- MoverIzquierda      Mueve el cabezal una celda a la izquierda.
- MoverAbajo      Mueve el cabezal una celda hacia abajo.
- Limpiar      Despinta la celda actual sin importar el color que tenga.
- PintarNegro      Pinta la celda actual de color negro.
- PintarRojo      Pinta la celda actual de color rojo.
- PintarVerde      Pinta la celda actual de color verde.



## QDraw: Puede fallar

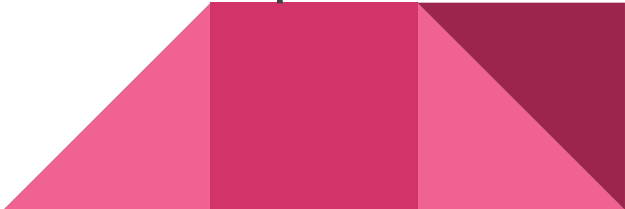
- ¿Qué pasa si le decimos al cabezal que se mueva a la derecha cuando ya no quedan celdas a la derecha?
- ¿Qué pasa si le decimos que despinte una celda que no está pintada de ningún color?



## QDraw: “BOOM”

En el caso de Lightbot al caminar de más, no pasa nada, el robot queda en el lugar.

En el caso de QDraw, si se realiza alguna acción inválida el programa falla, el cabezal explota por los aires y programador muere en el acto... bueno, no tan así, pero no queremos instrucciones inválidas.





# Sintaxis estricta

La mayoría de los lenguajes generales no usan un entorno gráfico como Lightbot, sino que se basan en un lenguaje con una sintaxis estricta que debe seguirse para que el autómata entienda las instrucciones que queremos darle.

QDraw es un lenguaje con sintaxis estricta.



# QDraw: Programa

Todo programa QDraw inicia con la palabra **programa**, seguida de un bloque de código con las instrucciones a ejecutar.



# QDraw: Bloque

Un **bloque** es un conjunto de instrucciones que se agrupan de forma secuencial, colocándolas entre llaves.

```
programa {  
    bloque  
}
```

# QDraw: Primer programa

Nuestro primer programa simplemente pinta la celda que se encuentra a dos celdas arriba de la celda actual.

```
programa {  
    MoverArriba  
    MoverArriba  
    PintarNegro  
}
```

**Nota:** Por convención decimos que el Cabezal siempre arranca en la posición Inferior Izquierda, salvo que se indique lo contrario.



# QDraw: Ejecución primer programa

Inicio



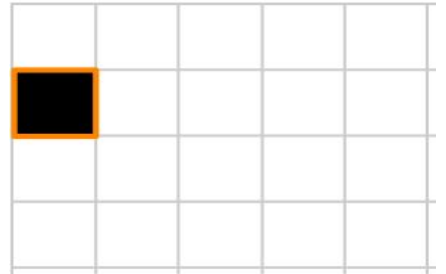
MoverArriba



MoverArriba

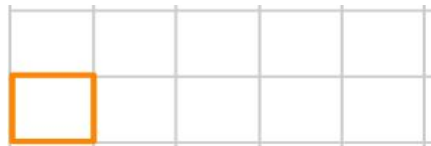


PintarNegro

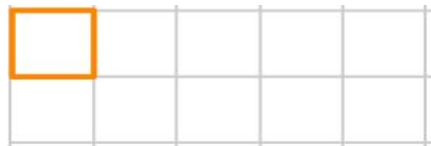


# QDraw: Ejemplo sobre tablero inválido

Inicio



MoverArriba



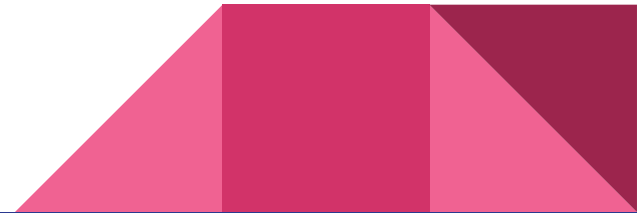
MoverArriba



PintarNegro

Jamás  
llega a  
ejecutarse


# Errores



# Errores sintácticos

La sintaxis del lenguaje es estricta, si no respetamos la sintaxis existe un error sintáctico y el código es inválido. Esto es lo mismo que decir que el código ni siquiera comienza a ejecutarse, pues está mal.

Ejemplos:

- `{ } programa` (el orden de la palabra programa y el bloque están mal)
  - `prog {}` (la palabra prog no es la esperada, sino programa)
  - `prog ()` (se espera que se coloquen llaves, no paréntesis)
- 



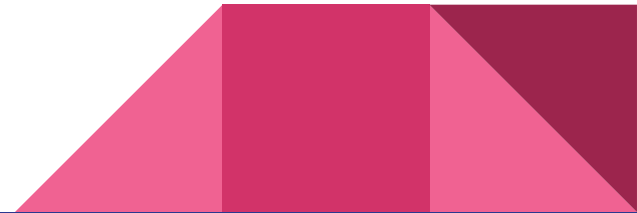
# Errores lógicos

Son aquellos que se dan como resultado de ejecutar un programa sobre un tablero inválido, como cuando el cabezal intenta moverse a una celda inexistente , o intenta despintar una celda que no está pintada.

El código visto anteriormente produce un error lógico al ejecutarse sobre un tablero de menos de 3 celdas de alto



# Comentarios



# QDraw: Comentarios

Los comentarios consisten en texto que el programador puede agregar en su código, útiles para determinar qué hace el programa.

Sirven para comunicar, no al autómeta sino a otros desarrolladores y a uno mismo sobre la estructura del código o las intenciones del mismo.

Los comentarios forman parte de la documentación de un programa y es preciso que sean claros y concisos.

Los comentarios se escriben en español, inglés, etc.



# QDraw: Comentarios

Los comentarios en QDraw están delimitados por los siguientes caracteres:

`/*` (para iniciar un comentario)

`*/` (para terminar un comentario)


Todo lo que esté entre esos dos símbolos es ignorado por el cabezal (autómata). Ej:

```
programa {  
    /* Esto es un comentario y será ignorado */  
    MoverArriba  
    MoverArriba  
    PintarNegro  
}
```

# QDraw: Comentarios

Los comentarios pueden ocupar múltiples líneas, y puede haber más de uno en un mismo programa.

```
programa {  
    /* Esto es un comentario y será ignorado, salvo por otros programadores.  
       Puede ocupar más de una línea.    */  
    MoverArriba  
    MoverArriba  
    PintarNegro  
    /* Acá hay otro comentario. */  
}
```



# Propósito



# Propósito

El **propósito** es el estado final en el cual se pretende que quede el tablero al finalizar el programa.

**Es aquello que el programa hace, independientemente de cómo lo hace.**



# Ejemplo

Los siguientes programas tienen el mismo propósito, aunque su código es distinto.

```
programa {  
    MoverArriba  
    MoverArriba  
    PintarNegro  
}
```

```
programa {  
    MoverArriba  
    MoverArriba  
    MoverArriba  
    MoverAbajo  
    PintarNegro  
}
```

**Pinta de negro la celda dos lugares más arriba de la celda actual.  
El cabezal queda en la celda pintada.**






# Escribiendo propósitos

En general definimos los propósitos en base al **estado inicial**, es decir, el estado en el que se encuentra el tablero al momento de iniciar el programa.

Vamos a definir la **celda actual** como la celda en la cual se encuentra el cabezal al iniciar el programa.




# Escribiendo propósitos

- **Nos va a importar mencionar en donde termina el cabezal luego de ejecutar, si no se menciona, asumimos que el cabezal termina en el mismo lugar donde comenzó.**
  - **Nos va a importar mencionar cuál es el estado final del programa.**
  - No nos importa qué pasos realiza el cabezal para lograr el estado final.
  - No nos importan que cosas se necesitan para que el programa termine de forma exitosa.
- 

# Ejemplos de propósitos

- Pinta la celda dos unidades hacia arriba de la celda actual, el cabezal termina en la celda pintada.
- Pinta las celdas inmediatamente hacia arriba y abajo de la celda actual.
- Dibuja un cuadrado de tres unidades de lado, donde la celda actual corresponde a la esquina inferior izquierda del cuadrado.
- Dibuja un perro donde la nariz está centrada en la celda actual.

Más ejemplos:

- Determina si la nave espacial puede despegar con el combustible que tiene en el tanque de reserva.
  - Calcula cuántos días se necesitan para viajar en barco desde Bolivia a Madagascar.
- 

# Precondición



# Precondición

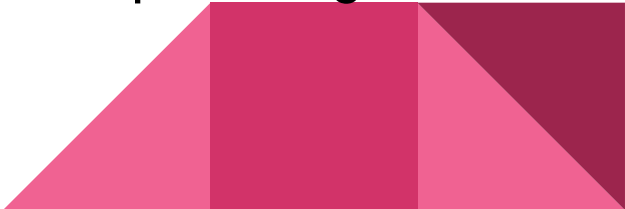
La **precondición** de un programa es aquello que el programa necesita del estado inicial para poder llegar al estado final indicado en el propósito.

**Es aquello que se necesita para que el programa termine de forma satisfactoria.**



# Escribiendo precondiciones


Usamos las mismas convenciones que para los propósitos, pero:

- **Nos importan que cosas se necesitan para que el programa termine de forma exitosa.**
  - No nos importa qué pasos realiza el cabezal para lograr el estado final.
  - No nos importa el estado final en el que termina el programa.
  - No nos importa qué pasos realiza el cabezal para llegar al estado final.
- 

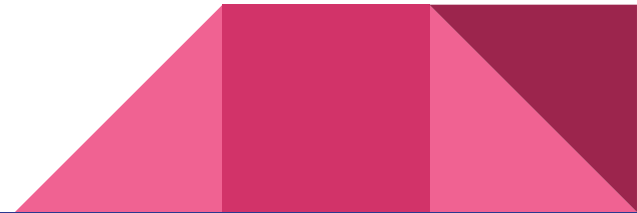
# Ejemplos de precondiciones

- Deben haber al menos dos celdas hacia arriba de la celda actual.
- Deben haber al menos 3 celdas hacia arriba y 3 hacia la derecha de la celda actual.
- Todas las celdas de la fila actual deben estar pintadas.

Más ejemplos:

- La nave espacial debe tener un depósito de combustible de reserva.
  - Existe una ruta marítima de Bolivia a Madagascar.
- 

Tanto propósito como precondiciones se colocan como comentarios en el programa.





# Ejemplo

```
programa {
```

```
  /*
```

```
  PROPÓSITO:   Pinta la celda dos lugares más arriba de la actual de color negro.  
               El cabezal termina en la celda pintada.
```

```
  PRECONDICIÓN: Existe una celda dos lugares más arriba de la actual.
```

```
  */
```

```
  MoverArriba
```

```
  MoverArriba
```

```
  PintarNegro
```

```
}
```

**Nota.** Las precondiciones son necesarias cuando un comando no es total. ¿Cuándo un comando no es total? Cuándo puede fallar y hacer BOOM.

Los comandos que no son totales y pueden fallar son los que se desplazan es decir, MoverDerecha, MoverIzquierda, MoverArriba, MoverAbajo, y el LIMPIAR

# Ejercitamos





# Programación

Clase 2

Introducción a la sintaxis estricta

Propósito

Universidad Nacional de Quilmes