

# Especificaciones de QDraw

<b>Primitivas</b>	<b>Parcial/Total</b>	<b>Efecto</b>
<b>PintarNegro</b>	<b>Total</b>	Pinta la celda actual de color negro(si había otro color en esta misma, ese color se pierde).
<b>PintarRojo</b>	<b>Total</b>	Pinta la celda actual de color rojo(si había otro color en esta misma, ese color se pierde).
<b>PintarVerde</b>	<b>Total</b>	Pinta la celda actual de color verde(si había otro color en esta misma, ese color se pierde).
<b>Limpiar</b>	<b>Parcial</b>	Remueve el color de la celda en la que se encuentra actualmente el cabezal. Esto puede fallar si no hay un color en la celda actual.
<b>MoverArriba</b>	<b>Parcial</b>	Mueve el autómeta/cabezal a la celda que está ubicada arriba de la celda actual(puede fallar si las dimensiones del tablero están acotadas).
<b>MoverAbajo</b>	<b>Parcial</b>	Mueve el autómeta/cabezal a la celda que está ubicada abajo de la celda actual(puede fallar si las dimensiones del tablero están acotadas).
<b>MoverDerecha</b>	<b>Parcial</b>	Mueve el autómeta/cabezal a la celda que está ubicada a la derecha de la celda actual(puede fallar si las dimensiones del tablero están acotadas).
<b>MoverIzquierda</b>	<b>Parcial</b>	Mueve el autómeta/cabezal a la celda que está ubicada a la izquierda de la celda actual(puede fallar si las dimensiones del tablero están acotadas).
<b>estaPintadaDeNegro?</b>	<b>Total</b>	Retorna verdadero si la celda actual está pintada de negro, si no retorna falso.
<b>estaPintadaDeVerde?</b>	<b>Total</b>	Retorna verdadero si la celda actual está pintada de verde, si no retorna falso.
<b>estaPintadaDeRojo?</b>	<b>Total</b>	Retorna verdadero si la celda actual está pintada de Rojo, si no retorna falso.
<b>estaVacía?</b>	<b>Total</b>	Retorna verdadero si la celda actual no está pintada, en caso contrario retorna falso.

## CONECTORES LOGICOS

AND	V	F
V	V	F
F	F	F

OR	V	F
V	V	V
F	V	F

NOT	
V	F
F	V

# Especificaciones de QDraw

## Estructuras de código válidas

```
programa {  
  ProcedimientoA()  
  ProcedimientoR()  
  ProcedimientoC()  
  ProcedimientoD()  
}
```

---

```
procedimiento ProcedimientoA(){  
  /*  
  *   Documentación  
  */  
  
  BLOQUE DE CÓDIGO  
}
```

---

```
procedimiento ProcedimientoR(){  
  /*  
  *   Documentación  
  */  
  
  repetir <nro> veces {  
    BLOQUE DE CÓDIGO  
  }  
}
```

---

```
procedimiento ProcedimientoC(){  
  /*  
  *   Documentación  
  */  
  si (<condición>) entonces {  
    BLOQUE DE CÓDIGO  
  }  
  sino {  
    BLOQUE DE CÓDIGO  
  }  
}
```

---

```
procedimiento ProcedimientoD(){  
  /*  
  *   Documentación  
  */  
  si (<condición>) entonces {  
    BLOQUE DE CÓDIGO  
  }  
}
```

## Estructuras de código inválidas

---

```
programa {  
  procedimiento ProcedimientoA(){  
    BLOQUE DE CÓDIGO  
  }  
}
```

---

```
procedimiento ProcedimientoR(){  
  /*  
  *   Documentación  
  */  
  repetir <nro> veces {  
    repetir <nro> veces {  
      BLOQUE DE CÓDIGO  
    }  
  }  
}
```

---

```
procedimiento ProcedimientoC(){  
  /*  
  *   Documentación  
  */  
  si (<condición>) entonces {  
    BLOQUE DE CÓDIGO  
  }  
  sino {  
    si (<condición>) entonces {  
      BLOQUE DE CÓDIGO  
    }  
  }  
}
```

---

```
procedimiento ProcedimientoD(){  
  /*  
  *   Documentación  
  */  
  repetir <nro> veces {  
    BLOQUE DE CÓDIGO  
  }  
  repetir <nro> veces {  
    BLOQUE DE CÓDIGO  
  }  
}
```