

**Aclaración importante:** todos los ejercicios tienen preguntas teóricas. Es importante responderlas para practicar la manera de redactar y autoevaluar los conceptos teóricos. Las evaluaciones pueden contener preguntas teóricas, por lo cual será necesario su práctica.

### Ejercicio 1: Top-Down - Castillo

- Responder:** ¿Para qué sirve la técnica TOP-DOWN?
- A partir del siguiente dibujo, se solicita realizar el gráfico **TOP-DOWN** que permita visualizar la descomposición de las partes que lo representan:



### Ejercicio 2: Tito el Robotito

Resolver los problemas de cada enunciado utilizando el lenguaje QDraw, y las buenas prácticas de programación. Para ello contamos con Tito el robotito, nuestro robot que se encarga de encender luces.

Se incorporaron a QDraw el siguiente set de primitivas:

- **EncenderLuz:** enciende la luz de la baldosa actual. Debe haber una lamparita apagada en la baldosa actual.
- **ApagarLuz:** apaga la luz de la baldosa actual. Debe haber una lamparita encendida en la baldosa actual.
- **hayLamparita?:** denota VERDADERO si en la baldosa actual hay una lamparita. Falso en caso contrario
- **estaEncendida?** Denota VERDADERO si la lamparita de la baldosa actual está encendida. Falso en caso contrario.

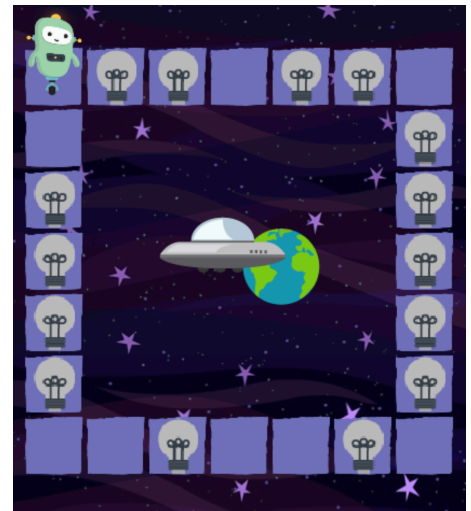
**a. Enunciado: Encender luces diagonales**

Tito deberá encender todas las luces de la habitación. Las luces **siempre están en la misma ubicación**, formando diagonales, y están todas apagadas. Tito inicia en la esquina inferior izquierda de la habitación (donde hay una lamparita). Y por cada diagonal debe iniciar en la lamparita de su extremo inferior. A continuación se muestra un mapa de la habitación con la distribución de sus lamparitas:



**b. Enunciado: Encender luces cuadrado**

Ahora Tito se encuentra en una habitación cuyas luces se distribuyen formando un cuadrado. Nuevamente deberá encender las luces que encuentre, pero en este caso no se sabe en qué espacio (metro) se encuentran las lamparitas ni qué estado tienen, por lo que deberá encender sólo aquellas que se encuentren apagadas. Tito inicia en la esquina superior izquierda de la habitación. Observar la imagen que se muestra con el mapa de la habitación y su estructura real, pero con un **ejemplo** de la distribución de sus lamparitas.



**c. Responder:** ¿Cuándo se necesita utilizar un condicional y cuándo no?

#### Ejercicio 3: Ana

##### a. Enunciado: Ana y sus caminos

Resolver el problema que respete el siguiente enunciado, utilizando el lenguaje QDraw y las buenas prácticas de programación:

##### Enunciado (problema):

Ana está asustada porque se apagaron todas las luces de su habitación. Para encender las lámparas, deberá llegar a los interruptores que se encuentran en la otra punta de la habitación. Pero como está a oscuras decide utilizar su pequeña linterna para seguir el camino hasta los interruptores, según el mapa que se muestra en la siguiente imagen.

El problema es que las baldosas pueden tener objetos tirados, ya sea un juguete o un lápiz, por lo que Ana deberá despejar el camino, y así poder continuar. Una vez que llegue al interruptor, que se encuentra una baldosa más arriba del camino, deberá encender la luz, pero sólo en caso que éste funcione correctamente. Se sabe que al menos uno de los interruptores funciona correctamente.

Ana inicia una baldosa a la izquierda del primer camino, tal como se muestra en la imagen.

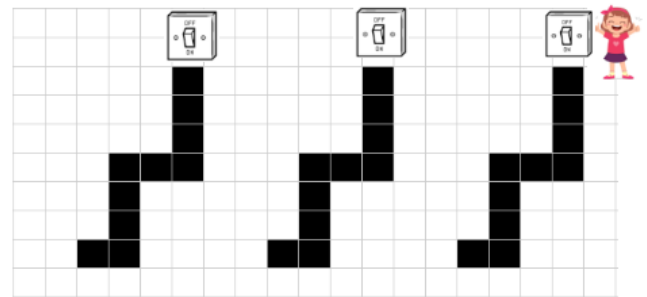
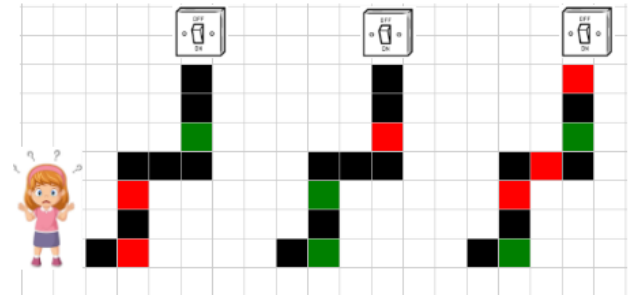
Para esta tarea será necesario conocer cómo se representa la información:

- **Baldosa con juguete:** se representa con una celda roja
- **Baldosa con lápiz:** se representa con una celda verde
- **Baldosa despejada:** se representa con una celda negra
- **Baldosa fuera del camino:** se representa con una celda blanca (vacía)

Utilizando QDraw, definir el procedimiento **EncenderLucesDeHabitacion()** para que Ana logre encender al menos una de las luces de la habitación.

Para su resolución se incorporaron a QDraw las siguientes primitivas:

- **interruptorFunciona?:** denota VERDADERO si el interruptor funciona correctamente. FALSO en caso contrario.
- **EncenderLuz:** presiona el interruptor para encender la luz correspondiente. Debe haber un interruptor funcionando.



#### b. Enunciado: Ana y sus caminos rectos

Ana nuevamente deberá llegar hasta los interruptores de su habitación para encender las luces, pero en esta ocasión, la habitación tiene un total de 8 caminos (la imagen muestra 4 pero es a modo de ejemplo) y con una nueva forma. Las baldosas nuevamente pueden tener objetos tirados, ya sea un juguete o un lápiz, que también deberá despejar para llegar hasta los interruptores. La diferencia es que en esta oportunidad, la manera de despejar la baldosa será corriendo el obstáculo de la siguiente manera:

- **Juguete:** deberá mover el juguete hacia la izquierda
- **Lápiz:** deberá mover los lápices hacia la derecha

La representación de la información es la misma que en el ejercicio 3) y es la siguiente:

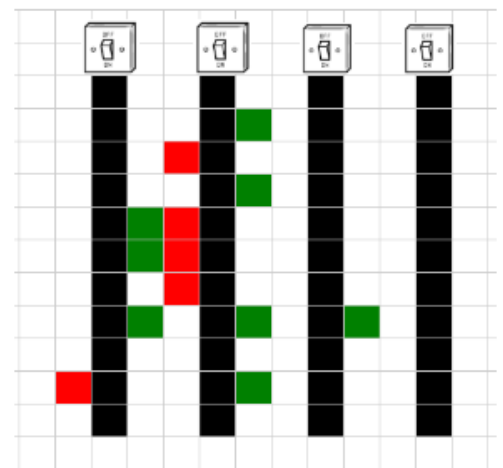
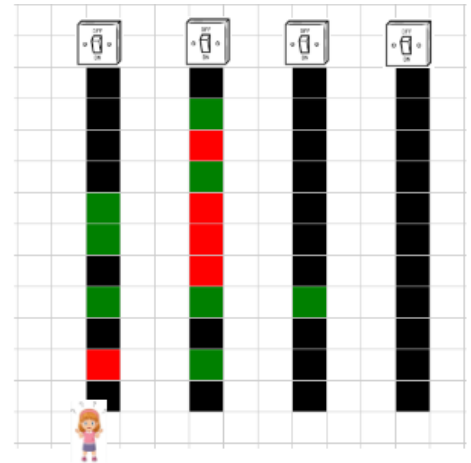
- **Baldosa con juguete:** se representa con una celda roja
- **Baldosa con lápiz:** se representa con una celda verde
- **Baldosa despejada:** se representa con una celda negra
- **Baldosa fuera del camino:** se representa con una celda blanca (vacía)

Se sabe que al menos uno de los interruptores funciona correctamente.

Ana inicia sobre la baldosa del extremo inferior del primer camino, tal como se muestra en la imagen.

Utilizando QDraw, definir el procedimiento

**EncenderLucesDeCaminosRectos()** para que Ana logre encender al menos una de las luces de la habitación.



Para su resolución se incorporaron a QDraw las siguientes primitivas:

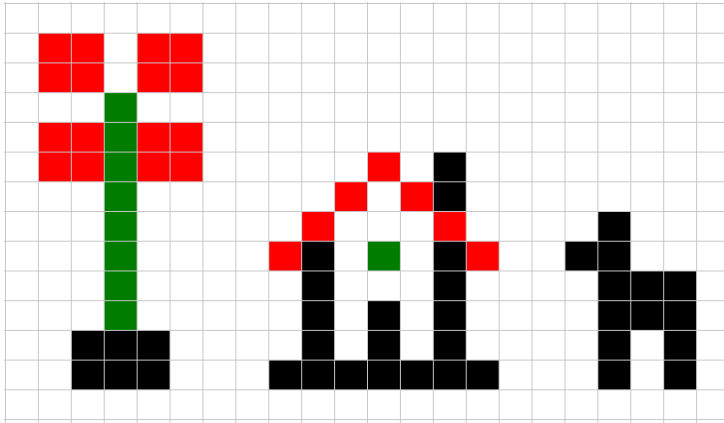
- **interruptorFunciona?:** denota VERDADERO si el interruptor funciona correctamente. FALSO en caso contrario.
- **EncenderLuz:** presiona el interruptor para encender la luz correspondiente. Debe haber un interruptor funcionando.

- c. **Responder:** ¿Por qué en la documentación de este tipo de problemas no es correcto mencionar el tablero y las celdas que lo componen? ¿Qué concepto esencial se pone en práctica ante esta situación?

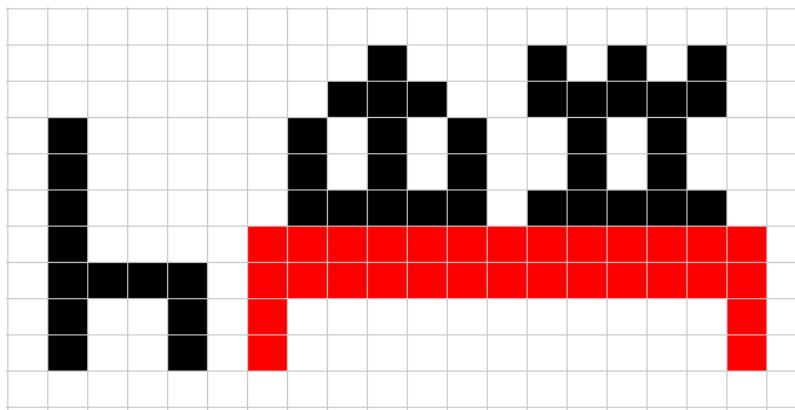


### Ejercicio 4: Dibujos

- a. Armar el TOP-DOWN del siguiente dibujo y definir un programa que genere dicho dibujo. Documentar el programa (el inicio y fin del cabezal quedan a criterio de quien programa).



- b. Enumerar los beneficios que se obtiene al reusar código
- c. Ahora se cuenta con el siguiente dibujo. Armar su TOP-DOWN y el programa correspondiente:



***Tener en cuenta que el tamaño de cada dibujo se encuentra fuera de escala con respecto al resto.***

- d. **Responder:** ¿Qué se debe hacer cuando no contamos con código para reusar? ¿Se define todo de nuevo? ¿Por qué?

### Ejercicio 5: Sapo Pepe

- a. Resolver el problema que respete el siguiente enunciado, utilizando el lenguaje QDraw y las buenas prácticas de programación:

**Enunciado (problema):**

El jardín donde habita el sapo Pepe tiene una superficie de 11 x 5 metros de parcelas. En dicho lugar podemos encontrar varias mariposas, las cuales deberán ser atrapadas por Pepe.

Para ello, Pepe dividió su jardín en 5 filas y 11 columnas, determinando así pequeñas parcelas de 1 metro c/u. En cada parcela Pepe podrá encontrar **Tierra, Pasto o bien arena**, y en el caso de la 3era fila, una pequeña Tarima que le

servirá para subirse y, de ser necesario, atrapar a la mariposa. Además Pepe decide recorrer las columnas de una por vez.

Para avanzar en cada columna, tendrá que tener en cuenta que:

- Si en la primera fila, comenzando desde la primera parcela, encuentra pasto o tierra, entonces puede avanzar a la siguiente parcela sobre su columna.
- Si en la segunda fila, comenzando desde primera parcela, encuentra pasto o tierra, entonces puede avanzar a la siguiente parcela sobre su columna.
- Y si llega a la última parcela de la columna (3era fila) y encuentra una tarima, entonces podrá subir a la misma y visualizar si hay mariposas para atraparlas.

Se solicita definir el procedimiento **AtraparLasMariposas()** que permita a Pepe atrapar las mariposas que encuentre en el jardín respetando las condiciones dadas. Pepe inicia en la esquina inferior izquierda del jardín. La imagen muestra la estructura real del jardín, pero el estado de sus parcelas sólo son un ejemplo ilustrativo.

Para ello, se incorporaron a QDraw las siguientes primitivas:

- **hayTierra?**: denota VERDADERO si en la parcela actual hay Tierra. FALSO en caso contrario.
- **hayPasto?**: denota VERDADERO si en la parcela actual hay Pasto. FALSO en caso contrario.
- **hayTarima?**: denota VERDADERO si en la parcela actual hay una Tarima a la cual subirse. FALSO en caso contrario.
- **hayMariposa?**: denota VERDADERO si ve una mariposa desde la parcela actual. FALSO en caso contrario. Pepe debe estar sobre una tarima en la parcela actual
- **SubirALaTarima**: sube a Pepe a la tarima. Debe haber una tarima en la parcela actual.
- **SaltarHastaLaMariposa**: Pepe salta hasta la posición donde se encuentra la mariposa.
- **AtraparMariposa**: Pepe atrapa a la mariposa de la posición actual y cae sobre la tarima. Debe haber una mariposa y una tarima.



- b. **Responder**: para cada conectiva provista por QDraw, ¿en qué casos se deben/pueden utilizar en un condicional?

#### Ejercicio 6: Sembrando el campo

- a. Resolver el problema que respete el siguiente enunciado utilizando QDraw:

##### Enunciado (problema):

Clara es dueña de un campo en Ranchos y nos contrató para que la ayudemos a optimizar el tiempo de siembra. El campo mide 1000 x 1000 m<sup>2</sup>. Para esta tarea nos provee un mapa del campo dividido en parcelas de 10 x 10 m<sup>2</sup> cada una. Clara especificó que cada parcela antes de sembrarse debe estar preparada.

##### Para ello se incorporaron a QDraw las siguientes primitivas:

- **hayTrigoSembrado?**: denota VERDADERO si la parcela actual cuenta con trigo previamente sembrado. FALSO en caso contrario
- **estaFertilizada?**: denota VERDADERO si la parcela actual ha sido fertilizada y se encuentra lista para sembrar trigo. FALSO en caso contrario
- **estaRegada?**: denota VERDADERO si la parcela actual se encuentra correctamente regada. FALSO en caso contrario
- **FertilizarParcela**: fertiliza la parcela actual dejándola lista para sembrar. La parcela no debe estar fertilizada.
- **PlantarTrigo**: planta las semillas de trigo en la parcela actual. La parcela no debe tener trigo sembrado.
- **PlantarBanderin**: coloca un banderín en la parcela actual marcando que la misma no se encuentra regada. La parcela no debe estar regada.

##### Para cumplir con la tarea solicitada, se deben tener en cuenta los procedimientos cuya información se encuentra a continuación:

- **SembrarParcela()**: planta las semillas de trigo en la parcela actual sólo si no hay trigo sembrado previamente. En caso contrario, y si no se encuentra regada, deberá colocar un banderín, marcando así la parcela.
- **SembarSeccionAlNorte()**: siembra todas las parcelas hacia el norte, sólo en caso que se encuentren preparadas.
- **PrepararParcela()**: fertiliza la parcela actual sólo si ya no se encuentra fertilizada

##### Se solicita:

1. Definir un **programa** que **siembre el campo completo**, invocando en el orden que corresponde, a los procedimientos dados (asumir que ya están definidos).
2. Definir **todos los procedimientos** en el orden invocado (con su respectiva documentación) en base a la información dada.

**Se aconseja realizar un dibujo del campo con sus dimensiones para visualizar la solución del problema.**

- b. **Responder**: ¿Qué diferencias y similitudes existen entre un programa y un procedimiento?



#### Ejercicio 7: RoboMac

- Responder:** ¿Cuál es la primera etapa de programación que se debe realizar? ¿Para qué sirve?
- Armar un Top-Down, diagrama, gráfico o algún esquema de tipo visual que resuma el enunciado (problema) a resolver. Se pueden mezclar distintos tipos de gráficos.
- Resolver el siguiente problema utilizando las prácticas de programación correspondientes según el enunciado a continuación:

#### Enunciado (problema):

La empresa de comida rápida MacHamburger, solicita definir un **programa** para que su robot “RoboMac” deje la cocina abastecida con todos los productos necesarios para los combos, y luego prepare 5 pedidos de combos.

Para la primera tarea será necesario definir el procedimiento **AbastecerCocina()**. Éste consiste en abastecer la cocina con todos los productos (pan y carne) para el armado de las hamburguesas. La cocina tiene dos hileras, una para el pan y otra para la carne. La primera se encuentra dividida en 26 casilleros, mientras que la segunda se divide en 16 casilleros. En ambos casos, deberá completar con el producto correspondiente aquellos casilleros que se encuentren vacíos.

En cuanto a los pedidos, se deberá definir el procedimiento **PrepararPedidos()** donde RoboMac deberá preparar los 5 pedidos de combos que se juntaron en la recepción, para luego ir a las freidoras y completar con papas aquellas que se encuentren vacías. Para armar los pedidos considerar que todos los combos se componen de una hamburguesa o una ensalada, una gaseosa y unas papas. RoboMac deberá ir a buscar cada componente y cargarlos en la bandeja. Una vez completado el combo, deberá dejar la bandeja en la recepción de pedidos.

Para ambas tareas tener en cuenta que el local cuenta con la siguiente estructura: la recepción de pedidos, el stand de gaseosas, la heladera de ensaladas, las freidoras, el stand de hamburguesas, los depósitos (pan y carne) y la cocina. RoboMac inicia en la puerta del local, pero debe terminar en la recepción.

#### Además RoboMac cuenta con las siguientes primitivas:

- **IrAFreidoras:** Ubica al robot en la sección de freidoras del local.
- **IrHeladeraEnsaladas:** Deja ubicado al robot en la heladera de ensaladas.
- **IrARecepcion:** Ubica al robot en la sección de recepción de pedidos.
- **IrASandDeHamburguesas:** Deja ubicado al robot en el stand de hamburguesas.
- **IrASandDeGaseosas:** Deja ubicado al robot en el stand de gaseosas.
- **IrADepositoDePan:** El robot va al depósito de pan, agarra el pan y vuelve al último casillero de pan donde se encontraba.
- **IrADepositoDeCarne:** El robot va al depósito de carne, agarra la carne y vuelve al último casillero de carne donde se encontraba.





- **UbicarCasillerosPan:** Deja ubicado al robot a la altura de los casilleros de pan. Debe estar ubicado en la cocina.
- **UbicarCasillerosCarne:** Deja ubicado al robot a la altura de los casilleros de carne. Debe estar ubicado en la cocina.
- **IrASigCasilleroDePan:** Posiciona al robot en el siguiente casillero de pan. Debe estar ubicado en los casilleros de pan.
- **IrASigCasilleroDeCarne:** Posiciona al robot en el siguiente casillero de carne. Debe estar ubicado en los casilleros de carne.
- **IrACocina:** Deja ubicado al robot en la cocina del local.
- **DejarPedido:** Lleva a RoboMac hasta la recepción de pedidos y deja la bandeja con el combo. Debe estar la bandeja cargada con el combo.
- **PonerPapasEnFreidora:** El robot coloca las papas freezadas que saca del freezer en la freidora que detectó vacía.
- **AgarrarBandeja:** El robot agarra una bandeja vacía en la recepción de pedidos. Debe estar en la recepción de pedidos.
- **AgarrarGaseosa** El robot agarra una gaseosa del stand y la coloca en la bandeja. Debe estar en el stand de gaseosas y tener la bandeja.
- **AgarrarHamburguesa** El robot agarra una hamburguesa del stand y la coloca en la bandeja. Debe estar en el stand de hamburguesas y tener la bandeja.
- **AgarrarEnsalada:** El robot agarra una ensalada de la heladera y la coloca en la bandeja. Debe estar en las heladeras de ensaladas y tener la bandeja.
- **AgarrarPapas:** El robot agarra un paquete de papas y lo coloca en la bandeja. Debe estar en la sección de freidoras y tener la bandeja.
- **CompletarCasilleroDePan:** El robot coloca el pan en un casillero vacío. Debe estar en un casillero de pan vacío.
- **CompletarCasilleroDeCarne:** El robot coloca la carne en un casillero vacío. Debe estar en un casillero de carne vacío.
- **pedidoConHamburguesa?:** Denota VERDADERO si el pedido del combo contiene una hamburguesa. FALSO caso contrario.
- **hayFreidoraVacía? :** Denota VERDADERO cuando al menos una de las freidoras está vacía. FALSO caso contrario
- **casilleroPanLLeno? :** Denota VERDADERO cuando el casillero de pan está lleno. FALSO caso contrario.
- **casilleroCarneLLeno?:** Denota VERDADERO cuando el casillero de carne está lleno. FALSO caso contrario.

#### Ejercicio 8.A: Repartidora de diarios

##### Enunciado (problema):

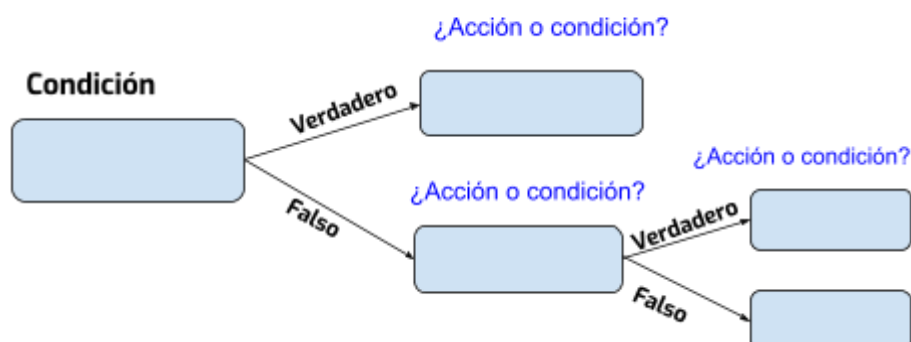
Juana, la repartidora de diarios de una empresa familiar deberá repartir, según corresponda, diarios o panfletos en las casas del barrio. Juana deberá dejar un diario en aquellas casas que sean clientes y requieran un diario, y dejar un panfleto en aquellas que no sean clientes; mientras que en las casas clientes que no requieran un diario no deberá realizar ninguna tarea. Por otro lado, como Juana recorre el barrio en moto, en caso de encontrar una estación de servicio y de no contar con nafta, deberá cargar nafta allí. De lo contrario, en caso de que la moto tenga nafta, deberá cargar en la estación de servicio, un bidón con nafta extra a modo de reserva. Las estaciones de servicio se encuentran en las esquinas del barrio, pero no todas las esquinas tienen una, ni se conoce dónde se encuentran. Por lo que, en aquellas esquinas donde no haya una estación de servicio, Juana no deberá realizar nada.

Juana inicia en la esquina inferior derecha del barrio y debe finalizar en el mismo lugar (ver imagen con el mapa del barrio en el ítem e).

**Las primeras consignas a continuación requieren de los conocimientos aprendidos en la unidad de Lógica. De esta manera, se solicita:**

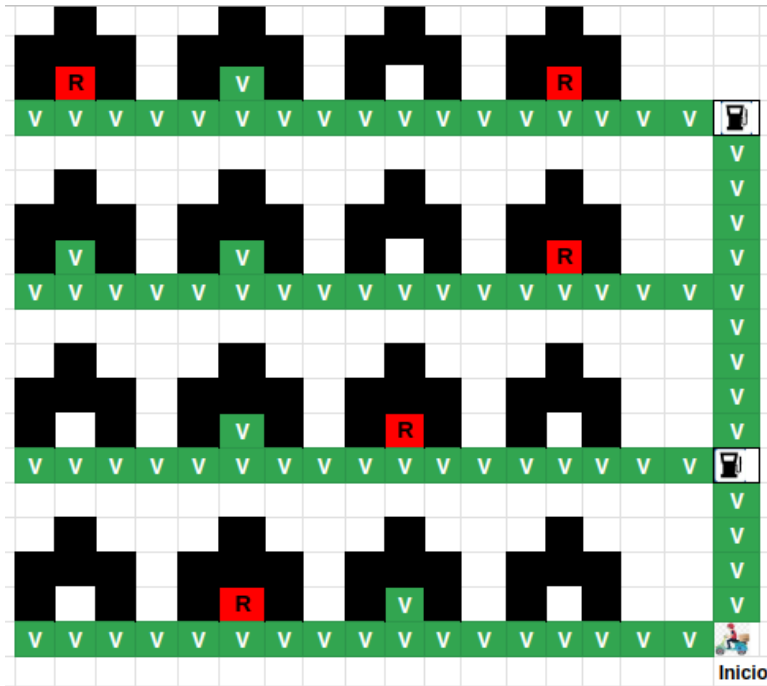
- Analizar la tarea a realizar en la estación de servicio y escribir la condición principal, **teniendo en cuenta que es compuesta (con sus conectivas correspondientes)**, utilizando las primitivas agregadas a QDraw que se detallan en el último ítem.
- Armar un diccionario en el lenguaje de la lógica proposicional con las proposiciones que componen la condición del punto anterior. Luego escribir la fórmula de la proposición compuesta que, luego, representará la condición a evaluar en el código.
- Armar una tabla de verdad con la proposición compuesta (fórmula) de la condición. Luego comparar las valuaciones de la tabla, el diccionario definido y el diagrama del ítem a), para analizar y validar las acciones tomadas. Por último responder: ¿cuándo es necesario crear una tabla de verdad para programar? ¿cómo se la puede aprovechar en este caso?
- En base a la condición compuesta definida en el ítem a) y al análisis de la tabla de verdad, completar el siguiente diagrama que representa la condición a evaluar en el código, junto con sus respectivas acciones por verdadero y por falso. Utilizar las primitivas agregadas a QDraw que se detallan en el último ítem.

#### Bloque de código





- e. Definir el **procedimiento RepartirMaterialesEnBarrio()**, que le permita a Juana repartir en las casas del barrio los materiales que correspondan. Para ello es importante conocer el mapa del barrio:



Este mapa muestra la **estructura real y fija del barrio**, pero no así el estado de las casas ni de las estaciones de servicio; dicha información sólo es un ejemplo ilustrativo a modo de referencia.

**Se cuenta además con la información representada de la siguiente manera:**

- Las cuadras del barrio se representan con celdas **verdes**
- Las casas del barrio se representan con celdas **negras**
- La puerta de cada casa (celda central) indica si la misma es cliente, y si necesita o no un diario:
  - Puerta de color **blanco**: es un cliente que necesita un diario
  - Puerta de color **verde**: es un cliente que no necesita un diario
  - Puerta de color **rojo**: no es un cliente
- De haber estaciones de servicio, se encuentran en la esquina de la cuadra
- Cada celda del mapa del barrio representa 1 metro



Para todo esto, se incorporaron a QDraw las siguientes primitivas:

*Nota: hay primitivas que sólo serán necesarias en el ítem 8.b)*

- **DejarDiario:** deja un diario en una casa. Juana debe estar en la puerta de una casa cliente que necesite un diario
- **DejarPanfleto:** deja un panfleto en una casa. Juana debe estar en la puerta de una casa que no sea cliente
- **CargarNafta:** carga nafta en su moto. Juana debe estar en una estación de servicio y tener la moto sin nafta
- **CargarGas:** carga gas en su moto. Juana debe estar en una estación de servicio y tener la moto sin gas
- **CargarNaftaExtra:** carga nafta extra en el bidón de Juana. Juana debe estar en una estación de servicio y su moto debe tener tanto nafta como gas.
- **hayEstacion?:** denota VERDADERO cuando en la esquina hay una estación de servicio. FALSO en caso contrario. Juana debe estar en una esquina.
- **tieneNafta?:** denota VERDADERO cuando la moto de Juana tiene nafta. FALSO en caso contrario
- **tieneGas?:** denota VERDADERO cuando la moto de Juana tiene gas. FALSO en caso contrario.

### Ejercicio 8.B: Repartidora reloaded

En esta oportunidad Juana cuenta con una moto que permite cargar tanto nafta como gas, similar a un auto. Por lo que la tarea a realizar en las estaciones de servicio se complejiza de la siguiente manera:

“Ahora en caso de encontrar una estación de servicio y de no contar con algún combustible, deberá cargar nafta, gas o ambos, según necesite. En caso contrario, es decir que ya tenga ambos combustibles, sólo deberá cargar en la estación de servicio, un bidón con nafta extra a modo de reserva.”

El resto del enunciado se mantiene como el original.

**De esta manera se solicita:**

- a. Escribir la **proposición compuesta** que refleje la nueva condición (utilizar las primitivas incorporadas a QDraw). Luego actualizar el diccionario del punto anterior y armar la fórmula correspondiente
- b. Armar otra tabla de verdad con la nueva proposición compuesta y analizar su información.
- c. **Responder:** ¿Qué procedimiento del enunciado original se deberá redefinir para contemplar esta nueva situación? ¿Qué sucede con el resto de los procedimientos, se pueden reutilizar? Definir sólo el procedimiento que resuelva este nuevo problema. ¿Qué nombre es más representativo para esta situación?

**ANEXO: EJERCICIOS DE ALGORITMIA COMPLEMENTARIOS**

**Ejercicio 9: Plaza**

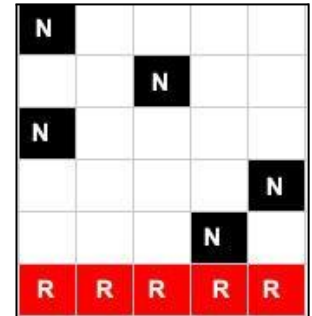
Juan es contratado por el municipio de Berazategui para limpiar la plaza “San Martín”, que luego de un recital quedó llena de papeles. Juan deberá barrer los papeles para depositarlos en el contenedor de residuos.

Nos solicitan ayudar a Juan automatizando su trabajo.

La tarea de **barrer** no implica levantar los papeles y llevarlos hacia cada contenedor, sino simular el efecto de usar una escoba.

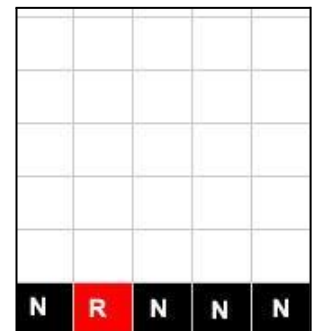
Los contenedores se encuentran distribuidos a lo largo de toda la plaza de manera horizontal e inicialmente se encuentran vacíos.

A continuación vemos un mapa de la plaza con sus dimensiones reales pero con un ejemplo de su estado, es decir que la distribución de los papeles puede variar. Juan inicia en la esquina superior izquierda de la plaza.



Para ello es importante conocer cómo se representa la información:

- **Contenedores de residuos:** se representan con celdas rojas
- **Papeles:** se representan con celdas negras
- **Baldosas sin papeles:** se representan con celdas blancas (vacías)



Mediante el lenguaje QDraw, se solicita definir el procedimiento

**LimpiarPlaza()** que ayude a Juan a cumplir con su tarea.

**Ejercicio 10: Piano**

Se solicita realizar la afinación de un piano. Para esto, el afinador posee un dispositivo que escucha la nota de cada tecla, la compara con una nota esperada, e indica si es correcta o no. La nota escuchada en el piano será correcta si la celda que la representa tiene el mismo color que la celda que representa la nota esperada. Hay dos tipos de teclas, blancas y negras, por lo que hay dos formas de representar la nota, con una celda blanca (vacía) o negra. En el caso contrario, el dispositivo indicará que la nota del piano debe afinarse y esto se representará marcando la tecla mediante una celda de color Rojo. La siguiente imagen muestra un **ejemplo** antes y después de marcar las teclas, teniendo en cuenta que:

- Cada columna representa una tecla del piano con sus 2 notas a analizar
- Sólo se representan las primeras 12 teclas del piano (a modo de ejemplo)
- La primera tecla (de izquierda a derecha) está afinada pues ambas notas son del mismo color (negro)
- La cuarta tecla también está afinada, pues ambas son del mismo color (blanco)
- La segunda tecla está desafinada, pues ambas notas son de diferentes colores. La nota del piano escuchada es de color negro, mientras que la esperada es de color blanco. De la misma manera la tercera tecla también está desafinada, pero en orden inverso.

