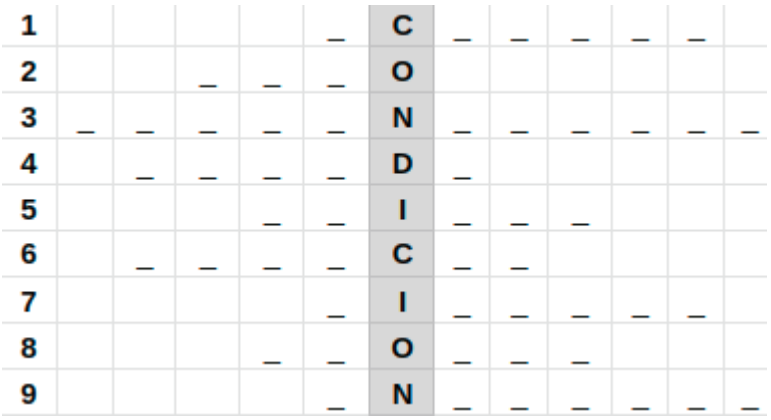
Universidad Nacional de Quilmes

Elementos de Programación y Lógica

Práctica 2.5 - Alternativa Condicional

Ejercicio 1

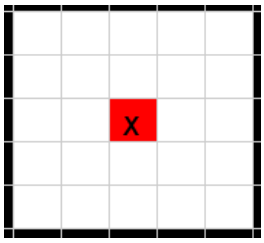
Resolver el siguiente crucigrama:

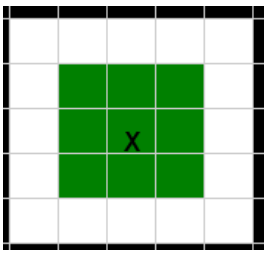


1. Es una forma, que nos permite aplicar una buena práctica al momento de usar una alternativa condicional
2. Primitiva utilizada en la alternativa condicional para escribir el bloque de código de la alternativa por falso
3. Caminos a ejecutar en base a una condición
4. Situación que se da en el tablero en un momento dado
5. Acción de incluir una alternativa condicional dentro de otra, considerada como mala práctica
6. Tipo de operadores que se utilizan en las condiciones de la alternativa condicional
7. Tipo de condición, cuya respuesta sólo puede tomar dos posibles valores
8. Conjunto de instrucciones que se ejecuta de forma consecutiva y ordenada
9. Primitiva utilizada en la alternativa condicional para escribir el bloque de código de la alternativa por verdadero

Ejercicio 2

Utilizando QDraw, definir un procedimiento que dibuje un cuadrado o una cruz dependiendo del color del centro de la figura. En caso que esté coloreado de Rojo debe dibujar un Cuadrado de 3x3 de color Verde centrado, caso contrario debe dibujar una Cruz de 3x3 de color negro también centrada. Para ello se cuenta con un cuadrado de 5x5.
El cabezal inicia en la celda central del cuadrado.
A continuación un ejemplo:





Ejemplo antes de ejecutarse el procedimiento **Después de ejecutarse el procedimiento**

Ejercicio 3

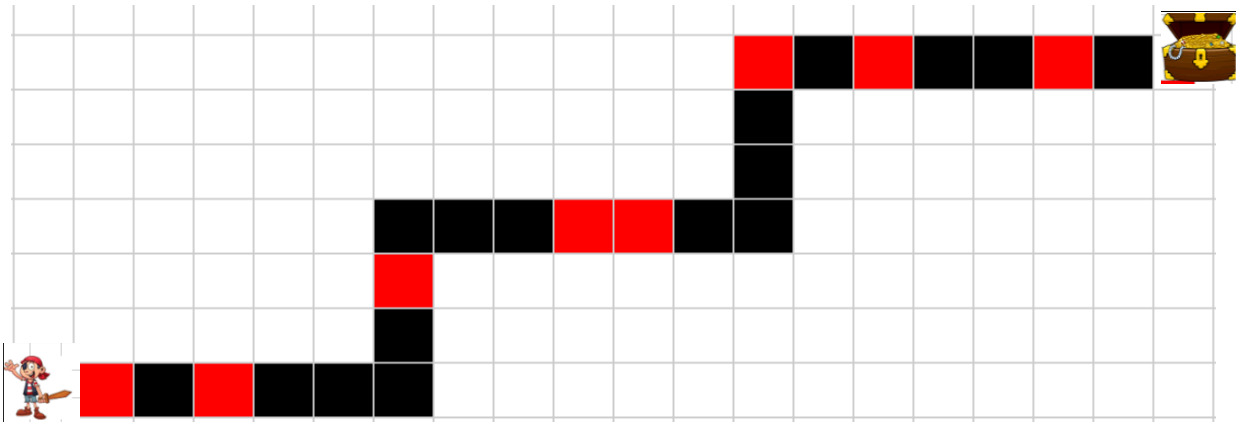
El pirata debe recorrer el camino marcado en el mapa hasta llegar al tesoro. Puede suceder que a medida que avanza se tope con una muralla; en dicho caso, deberá derribarla, quedando el paso libre, para así poder continuar con su camino.

Para esto será necesario conocer cómo se representa la información:

- **Muralla**: se representa con una celda roja
- **Paso libre (de muralla)**: se representa con una celda negra

Uilizando QDraw, se solicita definir el procedimiento **EncontrarTesoro()** que resuelva la situación mencionada. Sabemos que el camino tiene una forma fija. El pirata se encuentra ubicado un paso previo al inicio del camino (extremo sureste del mismo).

A continuación se muestra la estructura que tiene el camino, y un ejemplo de distribución de las murallas (no necesariamente están ubicadas en esas posiciones, es sólo un ejemplo):



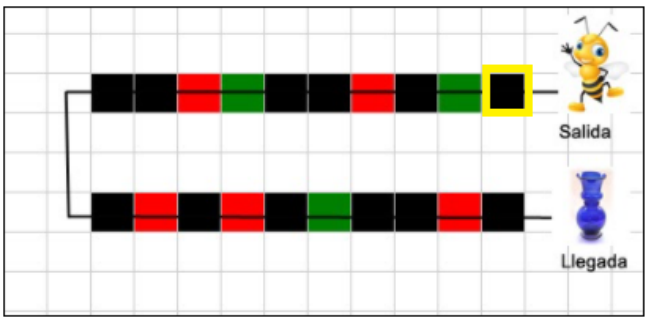
Universidad Nacional de Quilmes

Elementos de Programación y Lógica

Práctica 2.5 - Alternativa Condicional

Ejercicio 4

La abeja Susy necesita polinizar las flores de su jardín, el cual se encuentra representado en el siguiente mapa.



Se solicita definir el procedimiento **PolinizarJardin()** que ayude a Susy a polinizar todas las flores del jardín que se encuentren maduras o crecidas.
La figura muestra la estructura real del camino, pero no así la distribución de las flores, las cuales son sólo un ejemplo ilustrativo.
Susi inicia y finaliza en la ubicación mostrada en la imagen

Para ello será necesario conocer la representación de la información:

- **Flor madura**: se representa con una celda **roja**
- **Flor crecida**: se representa con una celda **verde**
- **Flor polinizada:** se representa con una celda blanca (vacía)
- **Espacio sin flor**: se representa con una celda **negra**

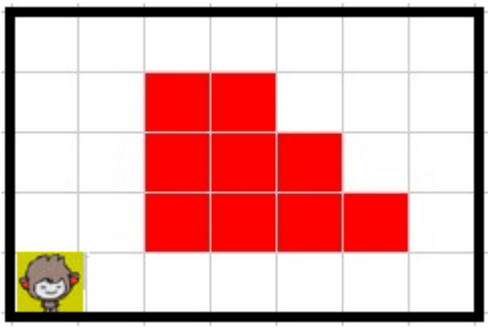
Ejercicio 5

La casa donde vive el mono Jorgito mide 7x5. La madre trabaja todo el día y le dejó suficientes bananas para que pueda alimentarse. Sabemos de antemano que las bananas **pueden estar distribuidas por toda la casa** pero Jorgito no sabe dónde se encuentran (no necesariamente como en el ejemplo). Jorgito necesita comer todas las bananas de la casa para quedar satisfecho.

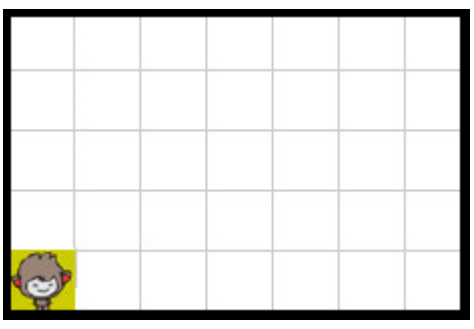
Para ello será necesario conocer cómo se representa la información:

- **Baldosa con banana:** se representa con una celda roja
- **Baldosa sin banana**: se representa con una celda blanca (vacía)

Se solicita definir el procedimiento **ComerTodasLasBananas()** que a partir del mapa dado ayude a Jorgito a comer todas las bananas de la casa**.**
Jorgito inicia en la esquina inferior izquierda de la casa.



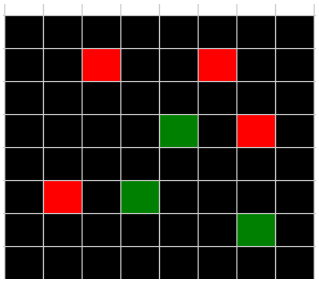
Antes de ejecutarse el procedimiento



Después de ejecutarse el procedimiento

Ejercicio 6

Una empresa dedicada a realizar videojuegos comienza a desarrollar la batalla naval, para ello define un tablero de 8 x 8 representado en el mapa a continuación:



La siguiente figura es un ejemplo de un tablero del videojuego con 4 barcos grandes y 3 barcos chicos.

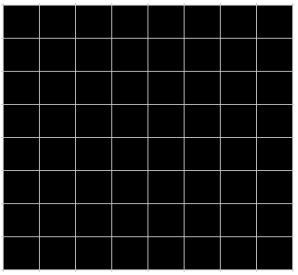
Tablero inicial (A)

Para entender el mapa será necesario conocer cómo se representa la información:

- **Porción de agua**: se representa con una celda **Negra**
- **Barco chico**: se representa con una celda **Verde**
- **Barco grande**: se representa con una celda **roja**

Se solicita definir en QDraw los siguientes procedimientos:

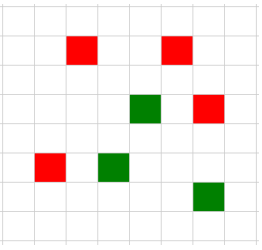
1. **EliminarBarcosDelOceano():** retira del juego tanto barcos chicos como grandes, colocando porciones de agua en su lugar.



Tablero final (B)

2. **MostrarSoloBarcos():** elimina todas las porciones de agua, mostrando sólo los barcos.

Tablero final (a partir de tablero final B)



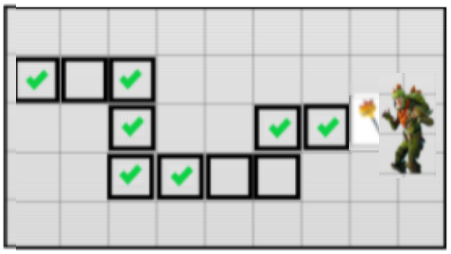
Tablero final (a partir de tablero A)

Ejercicio 7

REX el temerario combatiente de Fornite quien “lleva la competencia hacia la extinción”, necesita llegar hasta la otra punta de un túnel para recobrar un “Pico” (arma) que ha perdido en medio de una batalla. El túnel **tiene la forma que se representa en la imagen** y REX se topará con bombas y trampas durante su recorrido, las cuales deberá desactivar para continuar avanzado. Tanto las bombas como las trampas pueden estar distribuidas en el túnel en cualquier posición. Desactivar tanto bombas como trampas implica marcar dicho paso como liberado. Rex inicia fuera del túnel.
Utilizando las siguientes **primitivas** definir el procedimiento **RecuperarArma()**, que ayude al combatiente REX a recuperar su arma. Una vez que la recupere sacar a Rex del túnel:

- **IrAlComienzo**: coloca a REX al comienzo del túnel (en el primer paso)
- **MoverRex:** mueve a REX un paso hacia adelante. Debe haber camino
- **GirarRex**: gira a REX 90° en sentido de las agujas del reloj.
- **TomarArma:** toma el arma. Debe haber un arma.
- **DesactivarBomba**: desactiva la bomba marcando el paso como liberado. Debe haber bomba.
- **DesactivarTrampa**: desactiva la trampa marcando el paso como liberado. Debe haber trampa.
- **hayBomba?:** denota VERDADERO cuando hay Bomba. FALSO caso contrario
- **hayTrampa?:** denota VERDADERO cuando hay Trampa. FALSO caso contrario

Antes de ejecutarse RecuperarArma()



Después de ejecutarse RecuperarArma()

Ejercicio 7.2

Suponer ahora que se desconoce la forma del túnel, pero se sabe que tiene un largo de exactamente 10 pasos, y al salir del mismo se encuentra el arma. Teniendo en cuenta el algoritmo del ejercicio anterior, analizar qué se podría reutlizar o modificar para este nuevo algoritmo, y definir el procedimiento **RecuperarArmaEn10Pasos().** Reutilizar, donde corresponda, lo ya definido.

Para ello se agregaron las siguientes primitivas:

- **PosicionarRex:** ubica a Rex en la dirección correspondiente para poder avanzar un paso.
- **hayCamino?**: denota VERDADERO si el camino aún tiene pasos. FALSO en caso contrario.

Universidad Nacional de Quilmes

Elementos de Programación y Lógica

Práctica 2.5 - Alternativa Condicional

Ejercicio 8

La fábrica de alfajores **Capitán del Espacio** nos ha pedido que programemos a su robot "Capi" para utilizarlo en su línea de producción. La fábrica cuenta con 500 pallets, donde cada pallet almacena 75 cajas de una docena de alfajores. Cada caja contiene alfajores de un único sabor: blanco, fruta o bien chocolate. Pero para el caso del chocolate, las cajas tienen 24 alfajores y puede haber algunas que sean mixtas, conteniendo alfajores simples y triples.

Se solicita definir el procedimiento **CompletarStock()** para que Capi complete las cajas con sus alfajores según corresponda de todos los pallets. Cada pallet tiene una etiqueta del sabor pedido por el cliente, y sus cajas inicialmente se encuentran vacías y cerradas. Por lo que Capi primero deberá abrir la caja, completarla según corresponda a la etiqueta y luego cerrarla.
Para el caso de los alfajores de fruta, antes de completar la caja deberá limpiarla, dado que no son los más vendidos y suelen estar guardadas por más tiempo.
Para el caso de las cajas de chocolate que sean mixtas, luego de completarla, Capi deberá agregar una descripción y guardar en su memoria interna dicha información.
Capi se encuentra en la entrada de la fábrica listo para iniciar con su tarea. Al finalizar deberá volver allí.

Para esta tarea Capi cuenta con el siguiente set de instrucciones primitivas:

- **IrAEntrada:** ubica a Capi en la entrada, dentro de la fábrica.
- **IrAProxPallet**: ubica a Capi frente a la primera caja del pallet más cercano. Debe estar dentro de la fábrica.
- **IrAProxCaja**: ubica a Capi frente a la caja más cercana. Debe estar en un pallet.
- **IrAProxEspacio**: ubica a Capi en el espacio más cercano. Debe estar frente a una caja.
- **AbrirCaja**: abre la caja de alfajores. Debe estar frente a una caja.
- **CerrarCaja**: cierra la caja de alfajores. Debe estar frente a una caja.
- **LeerEtiqueta**: lee, mediante sus sensores, la etiqueta de la caja. Debe estar frente a una caja.
- **LimpiarCaja**: limpia la caja. Debe estar frente a una caja de alfajores de fruta.
- **ColocarSaborFruta**: coloca un alfajor de fruta en la caja. Debe estar frente a una caja de alfajores de fruta.
- **ColocarSaborBlanco**: coloca un alfajor de fruta en la caja. Debe estar frente a una caja de alfajores blancos.
- **ColocarSaborChocolate**: coloca un alfajor de chocolate en la caja. Debe estar frente a una caja de alfajores de chocolate.
- **AgregarDescripcion**: agrega una descripción a la caja. Debe estar frente a una caja de chocolate mixta.
- **GuardarInfo**: guarda en su memoria interna la información de la caja. Debe estar frente a una caja de chocolate mixta.
- **esFruta?**: Denota VERDADERO cuando la etiqueta de la caja dice sabor Fruta. FALSO en caso contrario. Debe estar frente a una caja y haber leído la etiqueta.
- **esBlanco?**: Denota VERDADERO cuando la etiqueta de la caja dice sabor Blanco. FALSO en caso contrario. Debe estar frente a una caja y haber leído la etiqueta.
- **esMixta?**: Denota VERDADERO cuando la etiqueta de la caja indica que puede almacenar alfajores de sabor chocolate simples y triples. FALSO en caso contrario. Debe estar frente a una caja de chocolate.