Trabajo práctico Nº 1

“TP 1”

Algoritmos y Estructura de Datos

Docentes:

- Ing. Pablo Damián Mendez

Alumnos:

Nombre: Ignacio Manuel Camporro
Nro de legajo: 2034610
Correo Institucional: icamporro@frba.utn.edu.ar
Usuario gitHub: IgnacioCamporroUTN
Link al repositorio: https://github.com/IgnacioCamporroUTN/TP-1

Fecha de entrega:

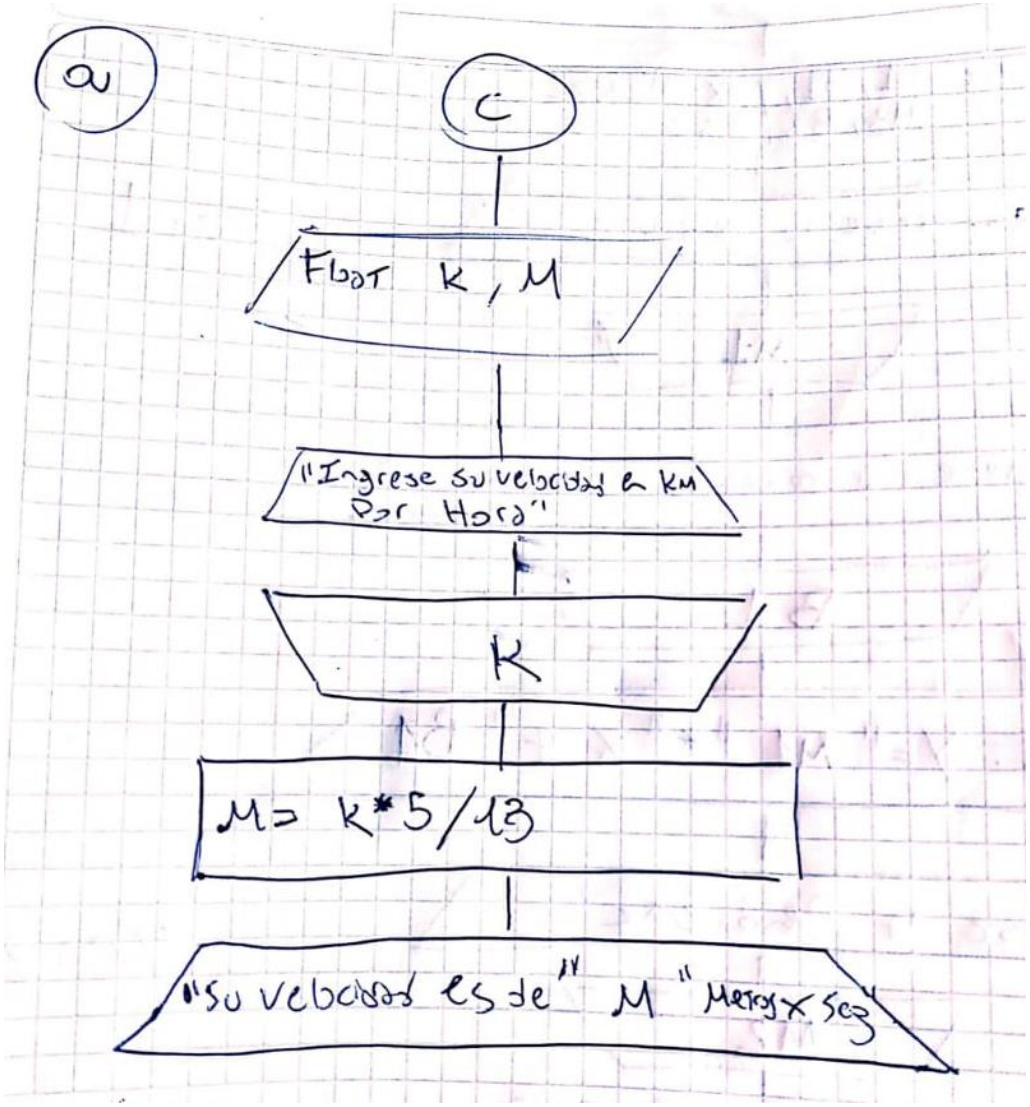
- 13 de Mayo de 2021

Ejercicio a.

El objetivo del programa es que un usuario ingrese un valor en km/hr y transformarlo a mts/seg. La conversión es realizada multiplicando el número ingresado por 5/18.

5/18 es por lo siguiente:
Para pasar de km/h a m/s hay que multiplicar por 1000 (los metros que tiene un km) y dividirlo por 3600 (los segundos que tiene una hora).

1000/3600 = 10/36 = 5/18 sería la simplificación.



Ejercicio b.

El objetivo del programa es obtener la intersección entre dos rectas ingresadas por el usuario. Se logra la intersección al igualar ambas rectas.

Y1= M1.X1 + B1
Y2= M2.X2 + B2

Igualo ambas rectas Y1=Y2 ,

Despeje:

M1.X1 + B1 = M2.X2 + B2
M1.X1 - M2.X2 = B2 - B1

Utilizo nuevas variables para almacenar los datos de M,B y X

(M3.X3 Representaría la diferencia entre M1.X1 y M2.X2)
M3.X3=B3

(Paso M3 dividiendo a B3)
X3= B3/M3.

Luego se remplaza la X en la ecuación de la recta para obtener el punto Y.

Y=M.X+B

Los puntos de Intersección son (X,Y).

En código:

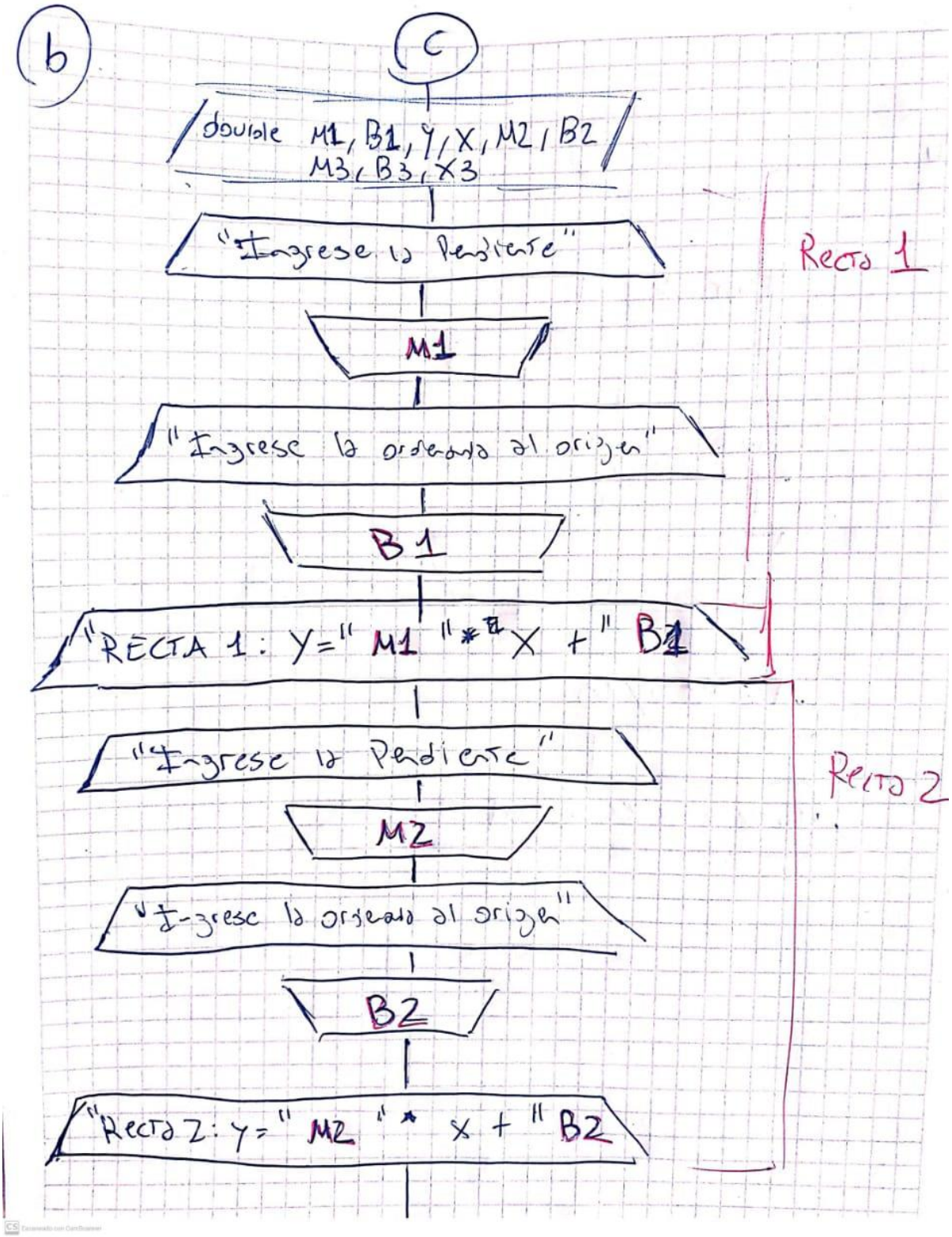
```
m3=m1-m2;
b3=b2-b1;
x=b3/m3;
y = m1 * x + b1;
```

Situaciones a tener en cuenta:

- Si las rectas son Coincidentes.
  Cuando ambas rectas son iguales no existe un único punto de intersección ya que estas se cruzan infinitas veces.
  Es por eso que establecí la condición de que si M3 es igual a 0 (si la resta entre ambas pendientes da 0 es porque ambas son iguales.) y si b3 es igual a 0 (si la resta entre ambas ordenadas da 0 es porque ambas son iguales).

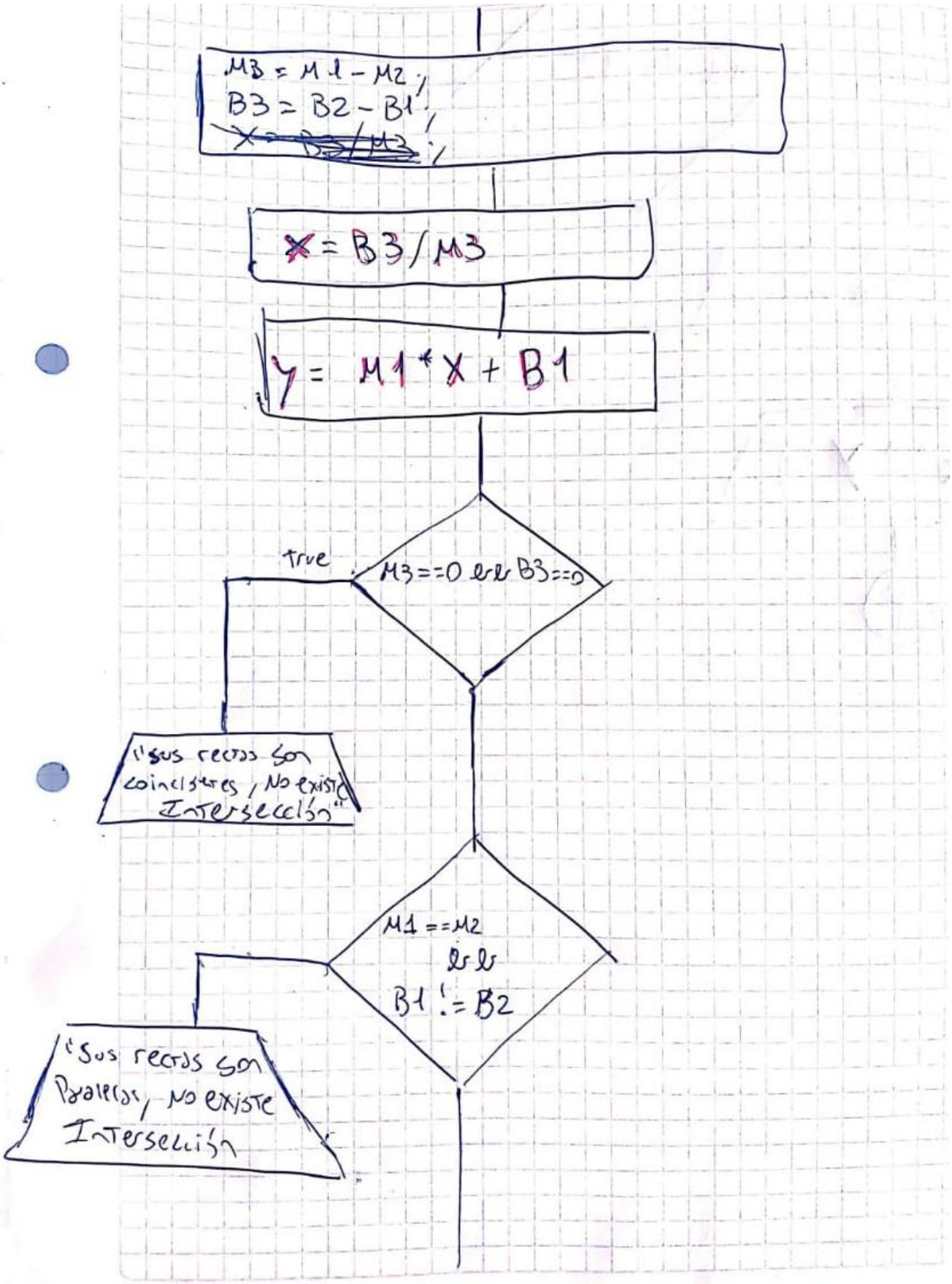
- Si las rectas son Paralelas.
  Cuando ambas rectas son paralelas no existe un punto de intersección jamás. Esto se da cuando las rectas tienen pendientes iguales. Para evitar que el programa confunda esta condición con la anterior detallada tuve que aclarar que las pendientes deben ser distintas para esta condición.

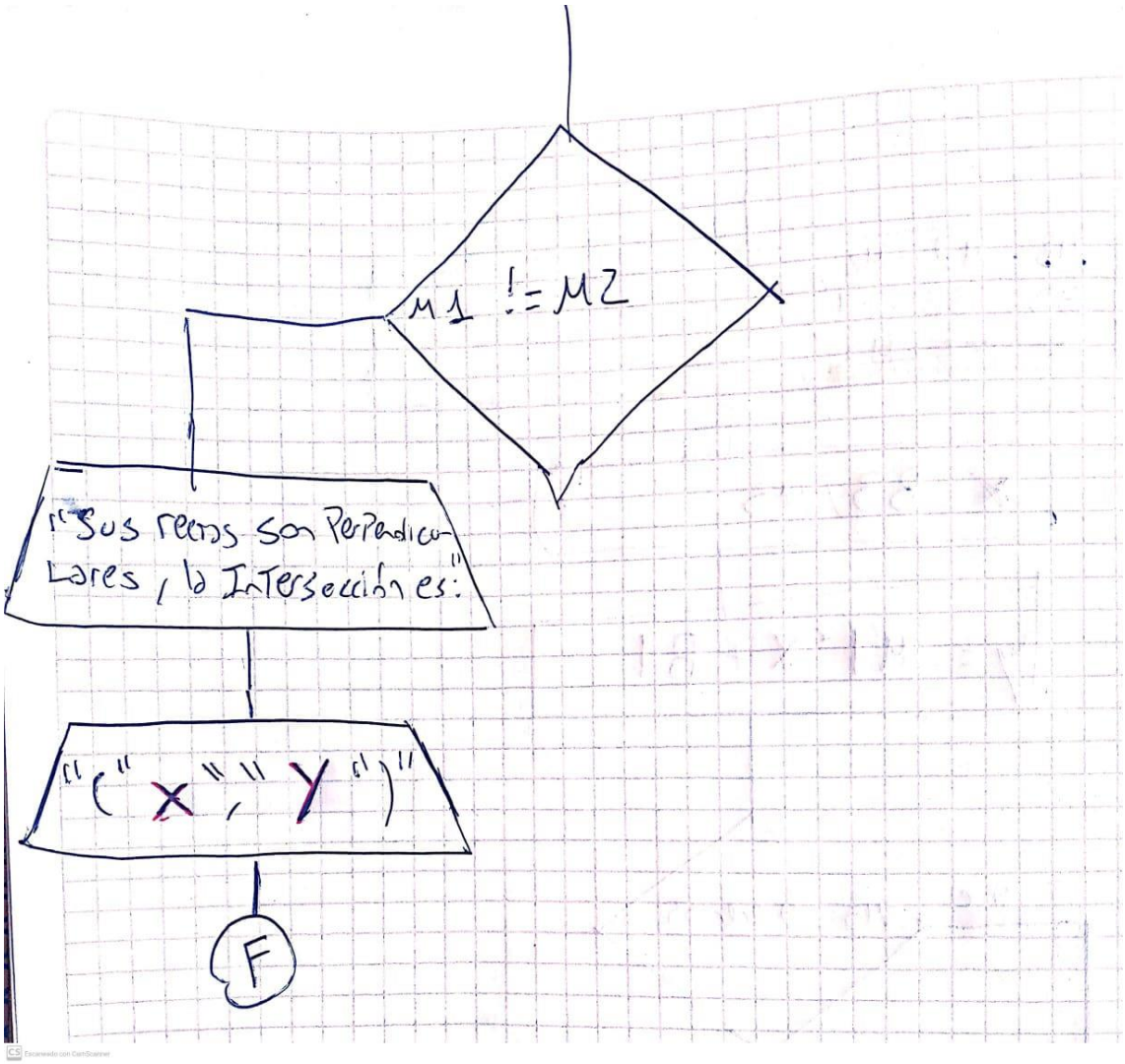
A partir de la siguiente página se puede observar el Diagrama de Lindsay de este ejercicio.


b
c
double M1, B1, Y, X, M2, B2
M3, B3, X3
"Ingrese la Pendiente"
M1
"Ingrese la ordenada al origen"
B1
"RECTA 1: Y=" M1 "*" X + " B1
"Ingrese la Pendiente"
M2
"Ingrese la ordenada al origen"
B2
"Recta 2: Y=" M2 "*" X + " B2
Recta 1
Recta 2

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Algoritmos y Estructura de Datos
Trabajo Práctico Nº 1





4)

Declaración de variables en Javascript.

Aunque no es obligatorio, se recomienda declarar siempre las variables antes de usarlas en JavaScript. Como sabemos esta declaración se hace escribiendo la palabra clave var seguida del nombre de variable. También se pueden declarar múltiples variables en una sola línea escribiendo var y a continuación los nombres de las variables separados por comas.

Por ejemplo podemos declarar:

var coches;
var motos;
var trenes;

O bien hacerlo en una sola línea de esta manera: *var coches, motos, trenes;*

También se permite la inicialización de variables en línea. Por ejemplo podríamos escribir:

var coches = 32, motos = 9, trenes = 12;

De este modo quedan las variables inicializadas en el mismo momento de declaración.

Sentencia "if" en Javascript:

La instrucción if … else funciona de forma análoga a como lo hace en otros lenguajes de programación. Permite controlar qué procesos tienen lugar, típicamente en función del valor de una o varias variables, de un valor de cálculo o booleano, o de las decisiones del usuario. La sintaxis a emplear es:

```
/* Ejemplo Estructura IF - aprenderaprogramar.com */
if (condición) {
        instrucciones
} else {
        instrucciones
}
```

La cláusula else (no obligatoria) sirve para indicar instrucciones a realizar en caso de no cumplirse la condición. JavaScript admite escribir un else y dejarlo vacío: else { }. El else vacío se interpreta como que contemplamos el caso pero no hacemos nada en respuesta a él. Un else vacío no tiene ningún efecto y en principio carece de utilidad, no obstante a veces es usado para remarcar que no se ejecuta ninguna acción cuando se alcanza esa situación.

Cuando se quieren evaluar distintas condiciones una detrás de otra, se usa la expresión else if { }. En este caso no se admite elseif todo junto como en otros lenguajes. De este modo, la evaluación que se produce es: si se cumple la primera condición, se ejecutan ciertas instrucciones; si no se cumple, comprobamos la segunda, tercera, cuarta… n

condición. Si no se cumple ninguna de las condiciones, se ejecuta el else final en caso de existir.

```
//if con else if y cláusula final else

if (DesplazamientoX == 0 && DesplazamientoY == 1) {
    alert ("Se procede a bajar el personaje 1 posición");
}
 else if (DesplazamientoX == 1 && DesplazamientoY == 0) {
    alert ("Se procede a mover el personaje 1 posición a la derecha"); }
else if (DesplazamientoX == -1 && DesplazamientoY == 0) {
    alert ("Se procede a mover el personaje 1 posición a la izquierda");
}
 else {
        alert ("Los valores no son válidos");
    }
```

Diferencias con C++:

En Javascript no es necesario declarar que tipo de variables son a diferencia de C++ donde hay que siempre delcararlas indicando su tipo, por ejemplo (Int, float, double, char). La sentencia if se maneja de igual manera dentro del lenguaje C++ tanto como en Javascript.

Fuentes:

https://bit.ly/3eLAuMK
https://bit.ly/3hvu7P4