# Algoritmos y estructuras de datos

### Trabajo Practico 1

Legajo: 1682258

Nombre y Apellido: Carlos Diego Simon Hurtado

Correo: [csimon@frba.utn.edu.ar](mailto:csimon@frba.utn.edu.ar)

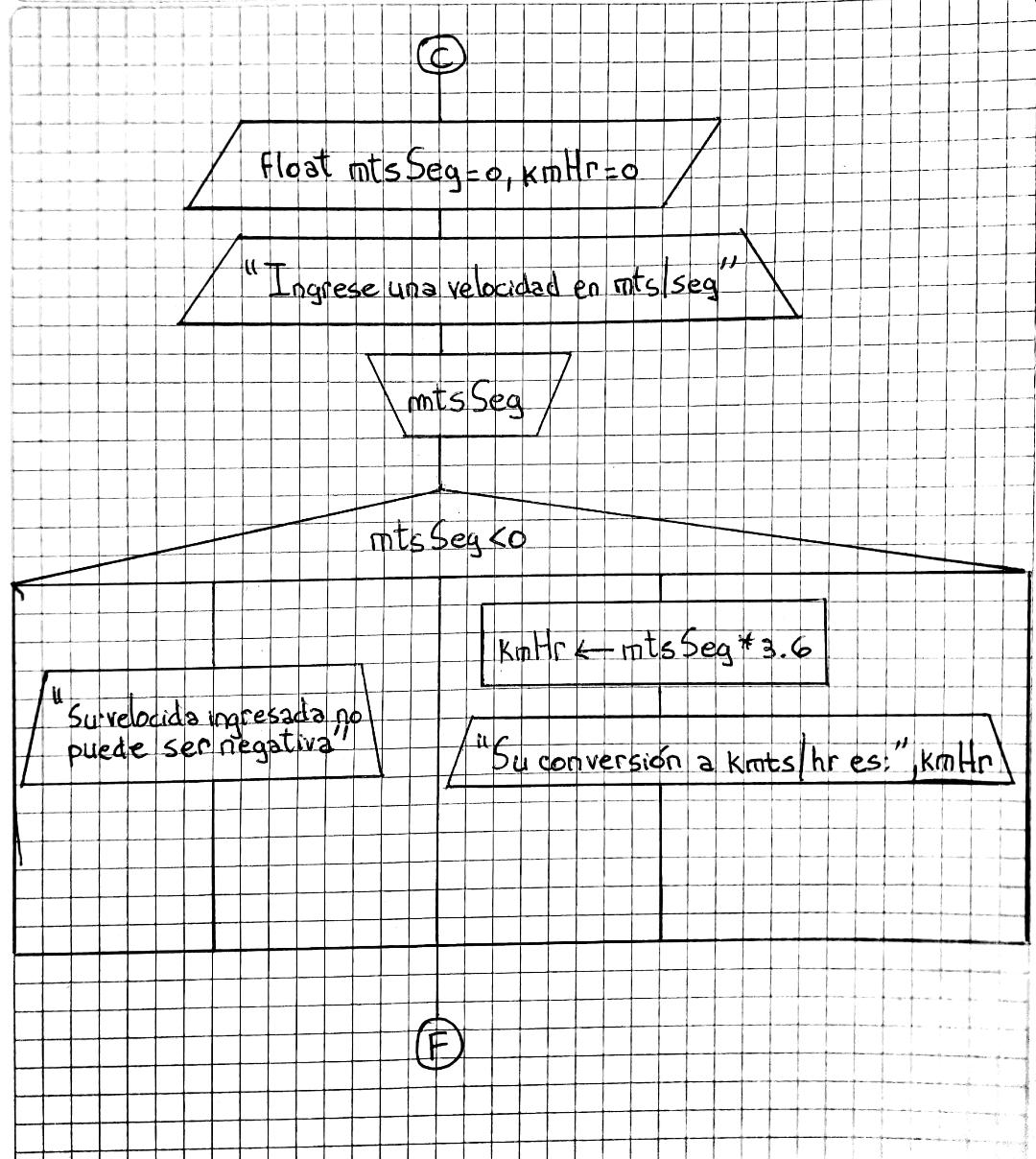
Usuario de GitHub: Tiempo Dulce

Enlace al repositorio: <https://github.com/TiempoDulce/RepositorioPrivado1>

Ejercicio 3 A

Análisis del problema:

Para el problema se pensó aplicar una regla de 3 simple para convertir de metros por segundos a kilómetros por hora, la misma corresponde a que si 1 metro por segundo son 3,6 kilómetros por hora entonces se multiplica a 3,6 por el valor ingresado por el usuario y luego se lo divide por 1 lo que daría el mismo resultado. Aparte se tiene en cuenta si el usuario ingresa valores negativos, ya que la velocidad no puede tener valores negativos. Luego se utilizaron variables del tipo “float” debido a que podía ser decimal la entrada de los metros por segundo y también podría ser decimal la salida de este numero convertido a kilómetros por ahora.



Ejercicio 3 B

Análisis del problema:

Para el problema primero se piden ingresar los valores de las variables de a, b, c e y, esta última debido a que si su valor es diferente de cero implica que habría que restarla a la ecuación de segundo grado y hacer que el valor de la “y” sea cero, ya que para que se saquen las raíces su valor debe ser cero.

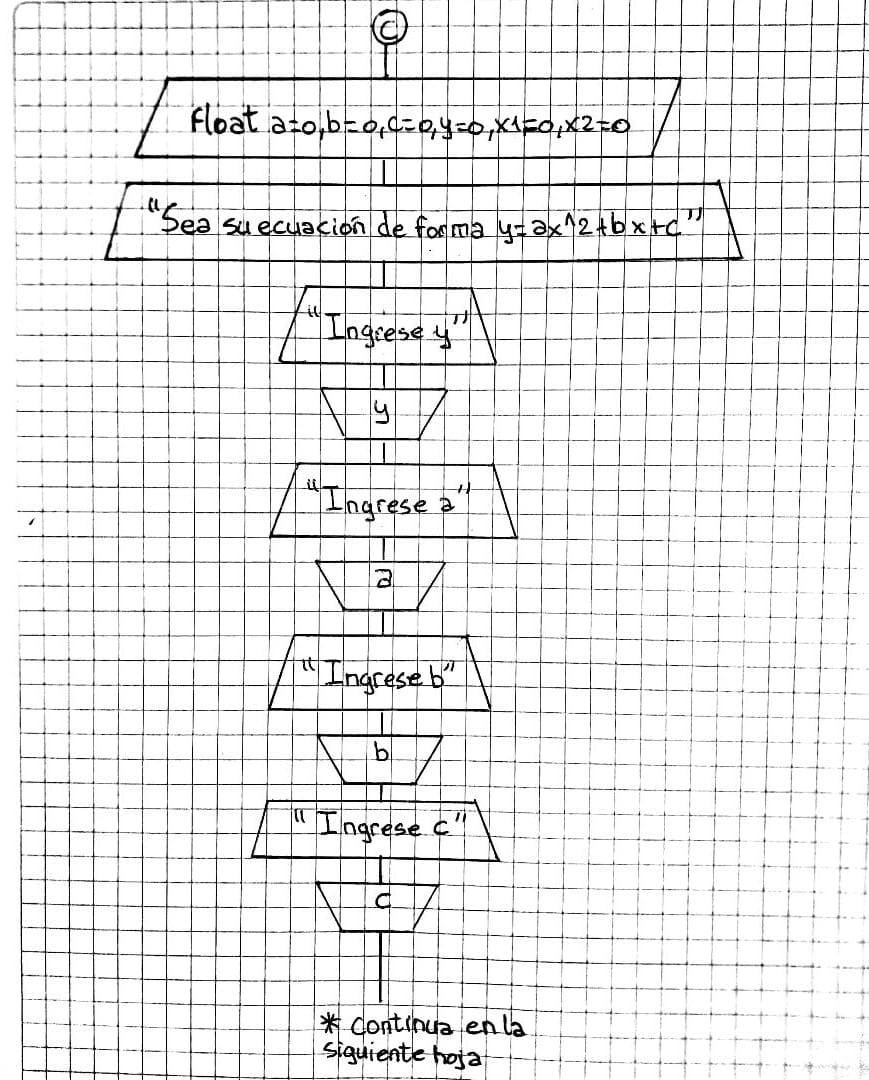
Una vez que se ingresaron todos los valores, se compara a la variable “a” con cero, ya que si este es cero entonces no sería una función cuadrática, sería una ecuación lineal.

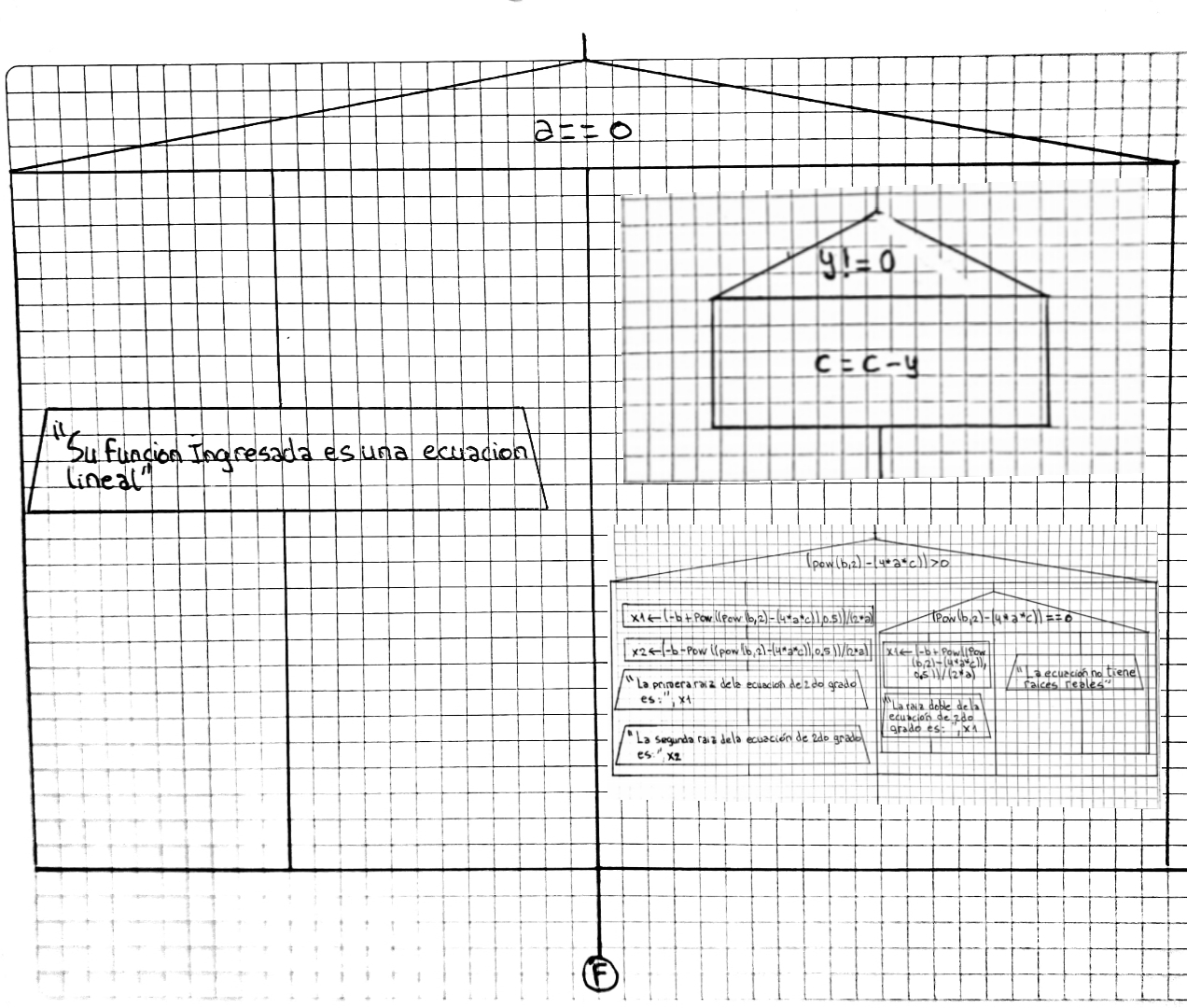
Consecuentemente si la variable “a” es distinto de cero, una sentencia condicional compara a “y” con cero y en caso de que sea igual se lo resta a la variable “c”, esto porque si es negativo el número entonces “pasaría” sumando y si es positivo entonces restaría a la variable “c”.

A partir de ahí ya tenemos una ecuación de segundo grado con “y” igual a cero, por ende, se pueden calcular las raíces de esta. Primero se tiene en cuenta el discriminante, el numero dentro de la raíz, si este es mayor que cero entonces tiene 2 raíces diferentes, si es igual a cero entonces tiene una raíz doble y si es menor a cero entonces no tiene raíces reales debido a que ese discriminante queda dentro de una raíz y no existiría solución alguna.

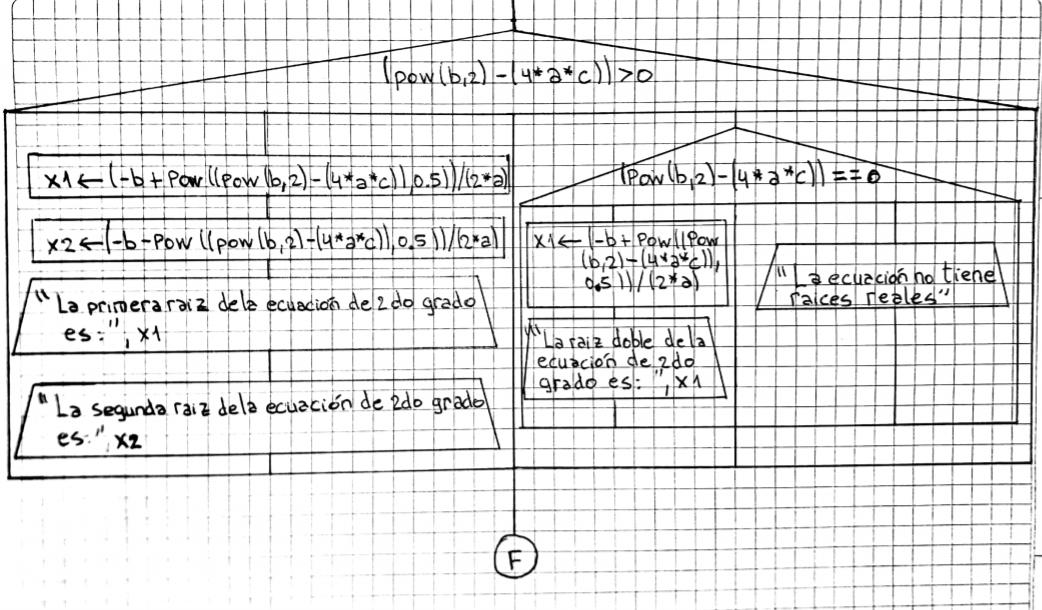
Todas estas posibilidades se contemplan con una sentencia selectiva del tipo if-else donde se compara primero si el discriminante es mayor a cero y en caso de que no entonces hay un if-else anidado donde se compara con cero y si es falso se muestra por pantalla que la ecuación no posee raíces reales.

Aclaración: para la 2da hoja lo hice de forma horizontal para que quede más legible.





\*Aclaración, no me entraba todo en la hoja, le adjunto el diagrama de la sentencia if anidada que casi no se ve.



Ejercicio 4

Variables:

|  |  |
| --- | --- |
| C++ | Javascript |
| Reglas estrictas al momento de declarar variables. | Permisividad para declarar variables. |
| Lenguaje fuertemente tipado, las variables son declaradas especificando su tipo, por ejemplo: int, float, double, char, bool, string. | Las únicas palabras claves son var, let, const y sirven para declarar cualquier tipo de variables que se utilicen. |
| Una vez que se declara el tipo de variable este no puede cambiar. | No es necesario especificar el tipo de variable, solo con la palabra var y la asignación de un valor ya se define la variable. Se puede usar la palabra clave “let” para declarar una constante la cual no puede variar su valor a lo largo de la ejecución de un programa |
| A menos que se declaren de manera global son locales al bloque donde se estén declarando. | Con la palabra clave “let” se afecta el ámbito de una variable, ya que solo es local al bloque donde se esta declarando. |

Sentencia if:

|  |  |
| --- | --- |
| C++ | Javascript |
| La sentencia se escribe con la palabra clave “if”, consecuentemente entre paréntesis la condición selectiva y entre llaves las sentencias de código a ejecutar. | La misma forma para declarar la sentencia selectiva que en C++ |