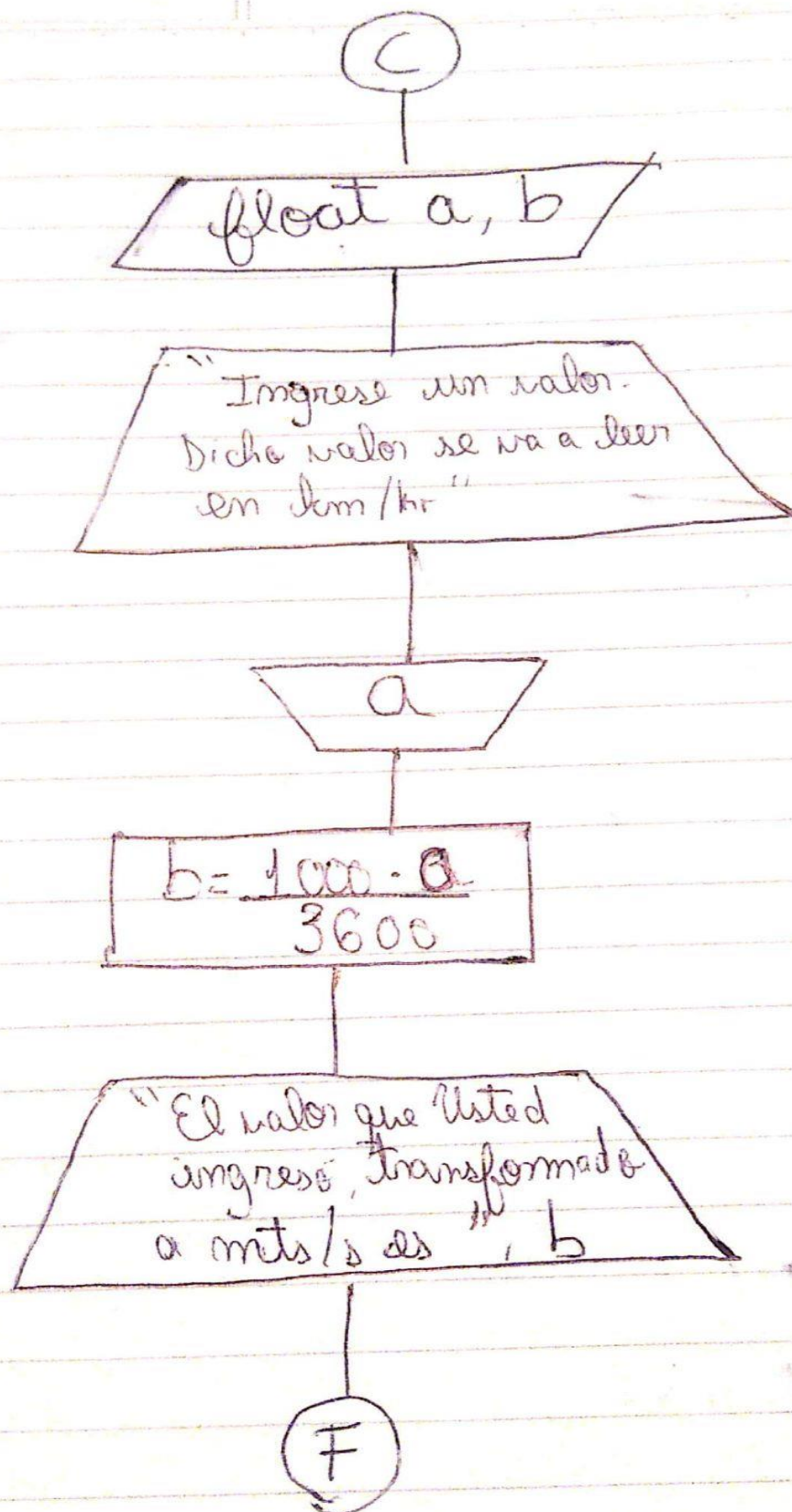


Trabajo Práctico 1

Algoritmos y estructuras de datos.

- Legajo: 2034456
- Alumno: Gabriel Borré
- Correo institucional: gborre@frba.utn.edu.ar
- Usuario Github: GabrielBorre
- Link al repositorio: <https://github.com/GabrielBorre/RepositorioPrivado1>
- Curso: K1024

Ejercicio 1
Diagrama de Lindsay



El primer paso del algoritmo consiste en declarar dos variables como números flotantes.

El segundo paso radica en solicitar al usuario que ingrese un valor numérico.

Además, se aclara que este valor será leído en km/hr

El tercer paso del algoritmo es asignarle una variable al valor introducido por el usuario.

El cuarto paso consiste en utilizar la otra variable declarada y asignarle el valor introducido por el usuario (a), multiplicado por 1000 y dividido por 3600. La razón de esta operación aritmética se fundamenta en que hay 1000 m por cada km, y existen 3600 segundos en una hora. Por lo tanto, si por ejemplo el usuario ingresa el valor 100(km/h), la conversión de este valor a mts/s es $100000\text{m}/3600\text{s}=27,7777\text{m/s}$ aproximadamente.

Finalmente, mostramos en pantalla “El valor que Usted ingresó, transformado a mts/s es” y el valor de la variable “b”.

El programa finaliza.

Ejercicio 2.

Diagrama de Lindsay

C

float m_1, b_1, m_2, b_2, X, Y

"Ingresar el valor de la pendiente de la primera recta"

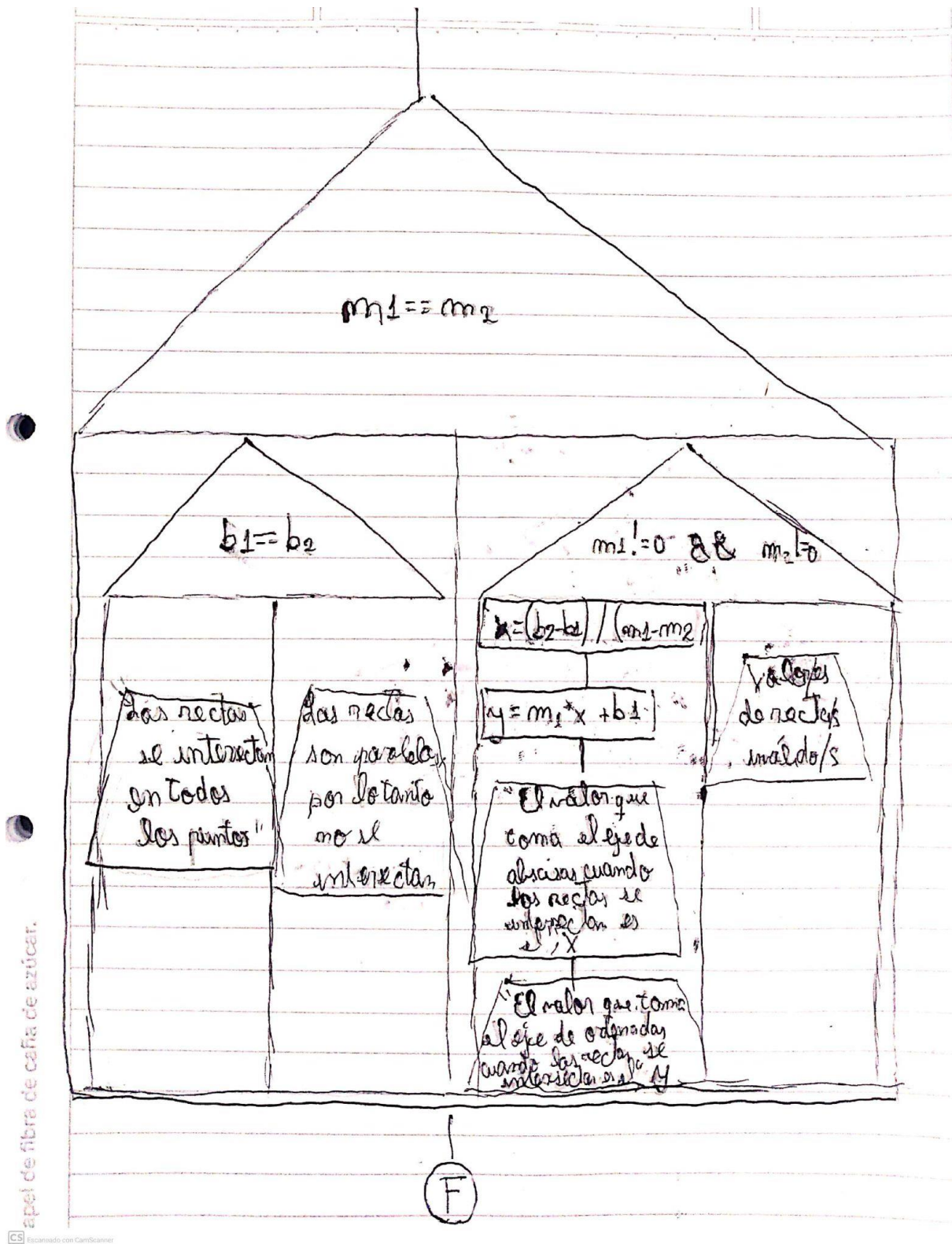
m_1

"Ingresar la ordenada al origen de la primera recta"

b_1

"Ingresar el valor de la pendiente de la segunda recta"

m_2



El razonamiento del ejercicio comienza a partir de la conceptualización de los distintos valores que pueden tomar la pendiente y la ordenada al origen de cada una de una de las dos rectas. La primera posibilidad es que los valores de las

pendientes de cada recta coincidan. Si esto sucede, caben dos posibilidades: las rectas son iguales o son paralelas. Para que el primer caso se cumpla, el valor de la ordenada al origen de las mismas debe coincidir. Por lo tanto, si son iguales, los puntos de intersección entre ellas son todos los comprendidos en las rectas. En caso de que sean paralelas, la ordenada al origen de cada una debe ser distinta. Si esto es efectivamente así, las rectas no se intersectan en ningún punto.

La segunda posibilidad es que los valores de las pendientes difieran. De pasar lo mencionado, y si los valores de ambas pendientes son distintos de cero, habrá que calcular la intersección entre las rectas.

El paso que sigue es pensar cómo se halla la intersección de dos rectas. Primero, se deben plantear las ecuaciones de ambas rectas:

$$y_1 = m_1 \cdot x + b_1$$

$$y_2 = m_2 \cdot x + b_2$$

Luego, se procede a igualar ambas ecuaciones: $m_1 \cdot x + b_1 = m_2 \cdot x + b_2$

$m_1 \cdot x - m_2 \cdot x = b_2 - b_1$. Se realiza factor común x y nos queda: $x \cdot (m_1 - m_2) = b_2 - b_1$

Por lo tanto, el valor de " x " cuando las rectas se intersectan es $x = (b_2 - b_1) / (m_1 - m_2)$.

El algoritmo prosigue, asignándole a la variable " x " la operación $(b_2 - b_1) / (m_1 - m_2)$

Posteriormente, se le asigna a la variable " y " la operación $m_1 \cdot x + b_1$.

De esta manera, en la variable " x " se albergará el valor que toma el eje de abscisas cuando las rectas se intersectan. Asimismo, en el eje " y " se almacena el valor del eje de ordenadas cuando las rectas se intersectan.

El programa debe mostrar en pantalla el valor que toma el eje " x " y el valor que toma " y ", cuando las rectas se intersectan.

Finalmente, en caso de que una de las pendientes equivalga a cero se informa por pantalla que el valor que el usuario ingresó es inválido.

4) Existen dos formas de declarar una variable global en Java:

La primera de ellas consiste en introducir la palabra "`var`", sucedida por una palabra. Por ejemplo, `var x=23`.

La segunda de ellas consiste en introducir la palabra clave "`const`" o "`let`", seguida por el nombre de la variable creada. Por ejemplo, `const nombre="Matías"`.

La declaración de variables en C++ es distinta ya que cuando creamos la variable, debemos primero especificar a qué tipo de dato corresponde la misma. Por ejemplo, si la variable corresponde a un número entero, debemos declararla con la palabra clave "`int`" y el nombre de la variable.

En lo correspondiente a la utilización de la estructura `If`, el lenguaje Javascript guarda una apreciable semejanza con el C++.

En primer lugar, ambos lenguajes de programación permiten utilizar la estructura `if-else` y, además, el `if` anidado.

En segundo lugar, tanto el C++ como el Javascript incluyen llaves luego de la declaración del operador lógico `IF-Else`. Sin embargo, cuando empleamos solamente el `IF` estas son optativas.

En C++ se declara de esta manera:

`if (condición)`

```

{
    //Acción a ejecutar a partir de la condición
}
else
{

}

```

En javascript se declara de esta manera:

```

if (condición) {
    //Acción a ejecutar a partir de la condición
}else{
}

```

Se observa que si bien se usan los mismos símbolos, el orden de ellos en las líneas de código no es el mismo. Vale también aclarar que ambos programas utilizan distintos operadores para mostrar un texto por pantalla: el C++ utiliza “cout”; el Javascript emplea “document.write”.

Ilustrado con un ejemplo, esto sería:

Para C++:

```

if (x>10)
{
    cout<<"El número ingresado es mayor a 10"<<endl;
}
else
{
    cout<<"El número ingresado es menor o igual a 10"<<endl;
}

```

Para Javascript:

```

if (x<10){
    document.write("El número ingresado es mayor a 10)
}else{
    document.write("El número ingresado es menor o igual a 10")
}

```

Si bien es verdad que ambos lenguajes presentan rasgos distintivos, se observa que comparten algunas características en común, como las recientemente expuestas.