

# **Trabajo Práctico 1:**

## ***Sentencias Selectivas y Sentencias de Asignación***

### **Algoritmos y Estructura de Datos**

**Ing. Pablo Damián Méndez**

**Curso: K1024 Año: 2021**

**Alumno: Marcos Fernando Gurruchaga**

**Legajo: 2037129**

**Mail Institucional: mgurruchaga@frba.utn.edu.ar**

**Usuario de Github: MarcosGurruchaga (Secundario  
MarcosGurru)**

**Link al repositorio: <https://github.com/MarcosGurruchaga/TP-1-AyED>**

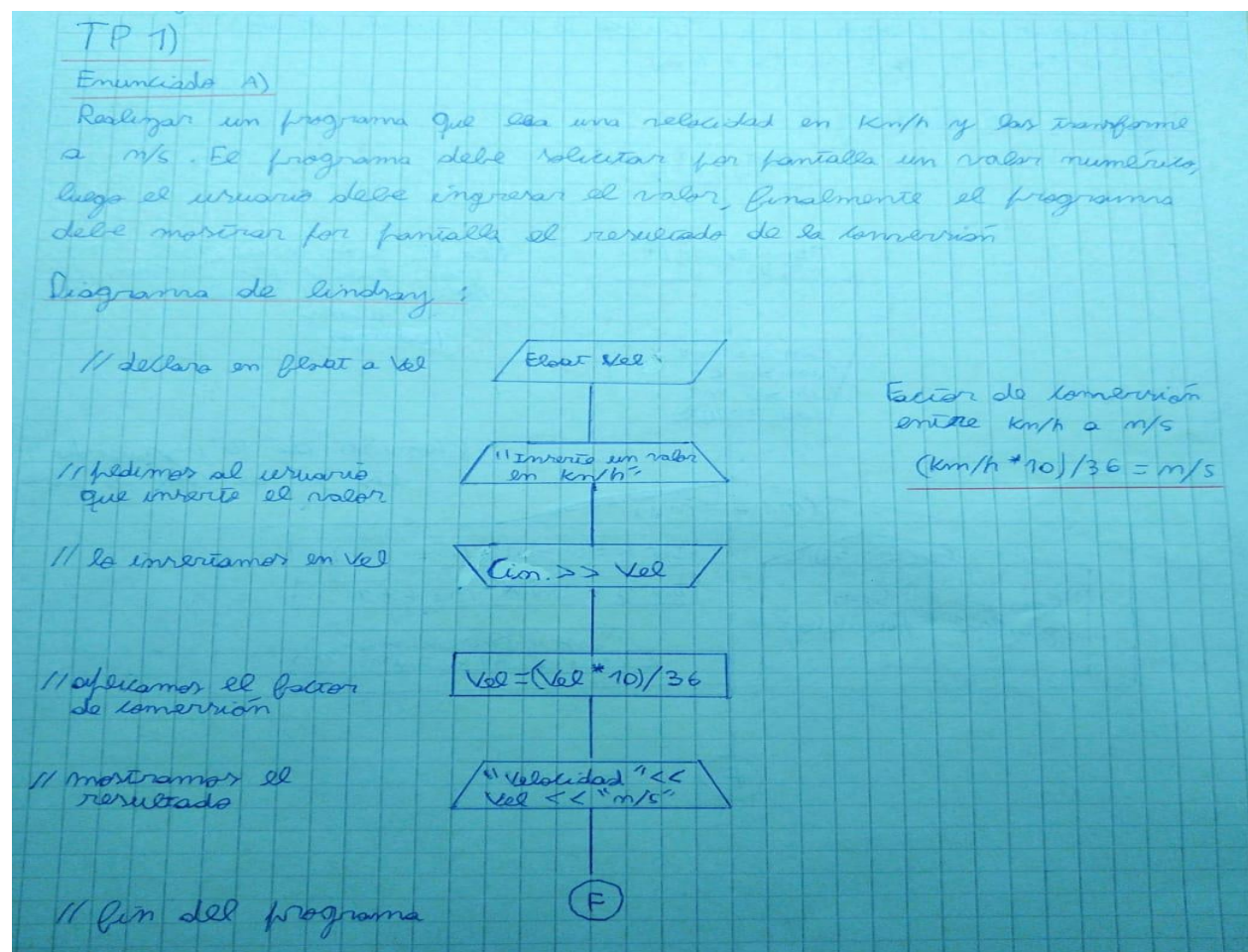
## Enunciado:

1. Crear un usuario en GitHub <https://github.com/> con el correo institucional frba. Crear un repositorio. Dentro del repositorio deberá subir todos los archivos que compongan la entrega de este trabajo dentro de una carpeta llamada "TP 1". Para desarrollar este punto, ver el apéndice de este documento.
2. Descargar alguna aplicación para escaneo rápido con dispositivo móvil (tablet o celular), como Scannable o CamScanner.
3. Resolver los siguientes ejercicios en diagrama manuscrito y en código C/C++ con los conceptos vistos en clase.
  - a. Realizar un programa que lea una velocidad en km/hr y la transforme a mts/seg.. El programa debe solicitar por pantalla un valor numérico, el luego el usuario debe ingresar un valor, finalmente el programa debe mostrar por pantalla el resultado de la conversión.

## Análisis del problema:

Después de que el usuario inserte un valor en la variable declarada en float, le aplicamos el factor de conversión y lo mostramos por consola.

## Diagrama de Lindsay:



b. Confeccionar un programa que calcule la intersección de dos rectas:

$$y1 = m1 * x + b1$$

$$y2 = m2 * x + b2$$

Realizar un adecuado análisis del problema, entendiendo los posibles resultados que se puedan generar. Tener en cuenta en ambos casos seleccionar los tipos de datos más adecuados.

#### **Análisis del problema:**

Antes de comenzar a hacer cuentas, le pedimos al usuario que ingrese en las cuatro variables declaradas (en float para considerar a todos los números reales) la ordenada al origen y pendiente de ambas rectas. Se tratan de m1, b1, m2 y b2.

Cuando tenemos dos rectas y debemos sacar su intersección las debemos igualar. Al realizar esto, debemos pasar los dos términos independientes para un lado y los dos términos dependientes (que multiplican a la x) hacia el otro. Luego de realizar estas operaciones debemos dividir en el lado donde pasamos a las variables independientes por el término dependiente que nos resultó, y así obtendremos la intersección en abscisas. Una vez que la tenemos, reemplazamos en cualquiera de las ecuaciones originales y así obtenemos el valor de Y correspondiente a la intersección de ambas rectas.

Sin embargo, antes de poder hacer los cálculos debemos descartar los dos casos en los que las rectas no se cruzan en solo punto con la sentencia "if". Esto se puede dar cuando las rectas son paralelas o cuando son la misma recta. En el caso de que sean paralelas, la pendiente es igual ( $m1 == m2$ ) y cuando son la misma recta además de cumplirse lo dicho anteriormente se cumple que la ordenada al origen es idéntica ( $b1 == b2 \ \&\& \ m1 == m2$ ). Considerado esto, solo quedaría mostrar por consola los resultados acordes.

Con cálculos intermedios, por ejemplo, usando A y B como variables intermedias los cálculos son los siguientes:

$$A = m1 - m2$$

$$B = b2 - b1$$

$$Intx = B / A$$

Esto no es necesario, ya que se puede resumir en una sola línea de código sin usar variables extra.

$$Intx = (b2 - b1) / (m1 - m2)$$

Finalmente, encontramos la intersección.

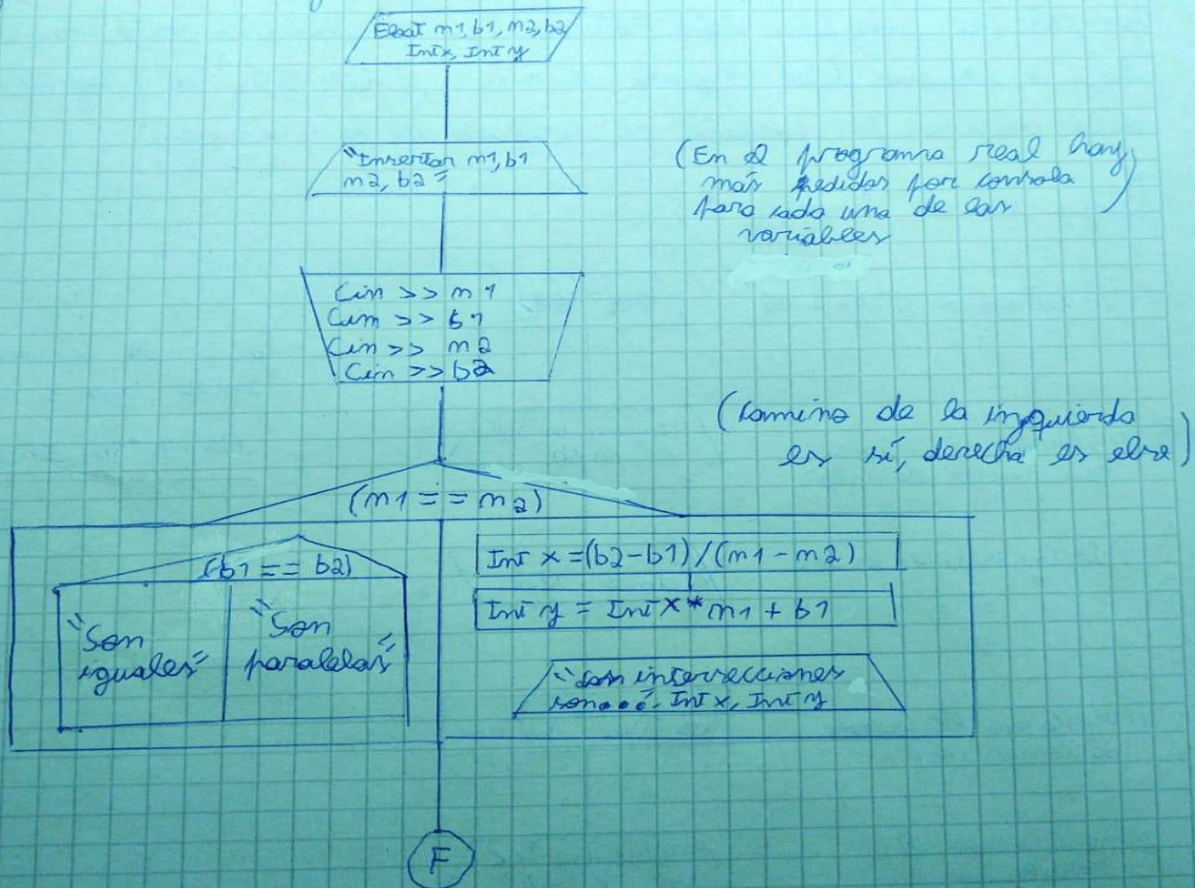
$$Inty = Intx * m1 + b1$$

#### **Diagrama de Lindsay:**

### Enunciado B)

Confeccionar un programa que calcule la intersección de dos rectas  
( $y = m_1 * x + b_1$ ,  $y = m_2 * x + b_2$ )

### Diagrama de flujo



#### 4. Investigue cómo se declaran variables y la sentencia "if" en lenguaje Javascript. Describa diferencias y similitudes.

La manera de declarar variables en Javascript difiere de la manera en la que se hace en C++. En lugar de escribir primero si la variable es un carácter o un entero u otro, se escribe si la variable es local (let), global (var) o constante (const).

Un ejemplo podría ser: **let n1 = 6; (alcance o tipo de variable-nombre)**

Esto quiere decir que no hace falta especificar que tipo de valores va a tener la variable a futuro, se adaptan al valor que reciben. Además, el uso del ; al final de cada sentencia no es estricto como lo es en C++ y esto permite que tanto let n1 como let n1; puedan ser leídos por el programa intérprete.

En cambio, un ejemplo en C++ podría ser: **Int n1 = 6; (tipo de valor va a contener la variable-nombre)**

La sentencia "If" en lenguaje JavaScript se escribe de exactamente a como se hace en C++, haciendo uso de los paréntesis para enmarcar a la condición que se va a evaluar y a continuación el corchete para delimitar hasta que línea se va a ejecutar el código si la condición es verdadera. La forma de usar la sentencia "Else" tampoco cambia.

Algo que también los hace diferir es que C++ es un lenguaje que debe ser compilado para poder ser ejecutado (Lenguaje de programación), mientras que el código de JavaScript necesita un programa "intérprete" para poder ser ejecutado y no necesita ser transformado previamente a binario (Lenguaje de Scripting). Esto explica el porqué de la forma en la que se declaran las variables en ambos lenguajes.