

```

#include <iostream>
int main () {
    float x, y ;
    char opt;
    std::cin >> x >> opt >> y;

    switch (opt) {
        case '*':
            std::cout << x * y << std::endl;
            break;

        case '/':
            if(y==0) {
                std::cout << "error";
                break;
            }
            std::cout << x / y << std::endl;
            break;
        default:
            std::cout << "sintaxis error" << std::endl;
    }
}

```

Figure 2: codigo

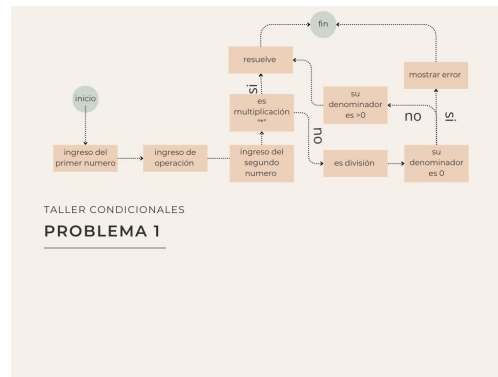


Figure 3: codigo

## 9 Preguntas del 2b:

¿Qué validaciones son necesarias?

- Valide fechas de nacimiento ingresadas por el usuario.
- Verifique si el mes de cumpleaños del usuario coincide con el mes actual.
- Otorgue o deniegue el descuento en función de esa verificación.

## 10 Preguntas del 2.1b:

¿Qué validaciones son necesarias?

- Validar que la fecha sea real y exista en el calendario:
- No puede haber un 30 de febrero.
- No puede haber un 32 de enero.
- Solo hay 29 de febrero en años bisiestos.
- Verificar que el mes de la fecha ingresada coincida con el mes actual.

## 11 Preguntas del 2.2b:

¿Qué validaciones son necesarias?

Validar que la fecha sea real y exista en el calendario:

No puede haber un 30 de febrero.

No puede haber un 32 de enero.

Solo hay 29 de febrero en años bisiestos.

Verificar que el mes de la fecha ingresada coincida con el mes actual.

## 12 Preguntas del 2.3b:

¿Qué casos especiales debes considerar?

29 de febrero: Solo es válido en años bisiestos (por ejemplo, 2020, 2024, 2028...).

Fechas con días fuera de rango (como 31 de abril, 32 de enero, etc.).

Fechas en formatos incorrectos o con letras.

Fechas en el futuro (opcional, dependiendo de si se aceptan recién nacidos).

## 13 Preguntas del 2.4b:

Qué entrada necesita el programa del usuario?

El programa necesita que el usuario ingrese su fecha de nacimiento completa, por ejemplo:

DD/MM/AAAA

Ejemplos válidos:

15/08/1990

29/02/2020

## 14 Preguntas del 2.5b:

¿Qué debe mostrar como resultado?

El programa debe mostrar uno de los siguientes resultados:

“¡Felicidades! Este mes es tu cumpleaños. Tienes un descuento.”

“Tu cumpleaños no es este mes. No aplicas para el descuento.”

“Fecha inválida. Intenta nuevamente.”

```
include <iostream>
int main () {
    int fecha, x;

    std::cout << "dia de cumpleaños" << std::endl;
    std::cin >> x;

    if (x<29) {
        std::cout << "mes de cumpleaños" << std::endl;
        std::cout << "expreselo en forma numerica (1-12)" << std::endl;
        std::cin >> fecha;
    }else{
        return 0;
    }

    if (fecha==02){
        std::cout << "descuento redimible" << std::endl;
    }else {
        std::cout << "descuento no redimible" << std::endl;
    }
}
```

Figure 4: codigo

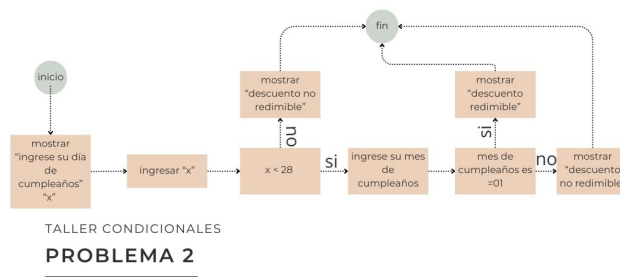


Figure 5: código

## 15 Preguntas del 3b:

¿Qué hace el programa?

Es un programa que aplica descuentos según el tipo de cliente. Si eres VIP, tienes un (20) de descuento. Si eres cliente regular y compras más de 3 productos, te dan un (10) de descuento. Si compras 3 o menos, no hay descuento.

## 16 Preguntas del 3.1b:

¿Qué situaciones especiales pueden pasar?

Que el usuario escriba una opción inválida (como '3' o una letra).

Que ponga montos negativos o cero.

Que diga que es VIP sin serlo (el programa lo acepta igual porque no hay validación real).

## 17 Preguntas del 3.2b:

¿Qué cosas hay que revisar (validar)?

Que el usuario elija bien el tipo de cliente (solo '1' o '2').

Que los precios y cantidades que ingrese sean números y no negativos.

Que no deje campos vacíos o escriba cosas raras (como letras en lugar de números)

## 18 Preguntas del 3.2b:

¿Qué le pide al usuario?

Que diga si es cliente VIP o regular.

Que escriba cuánto cuesta su compra.

Y si es cliente regular, también cuántos productos compró.

## 19 Preguntas del 3.3b:

¿Qué muestra al final?