

Si tiene descuento, le dice cuánto es.
 Si no tiene descuento, le muestra el total normal.
 Si escribió algo mal, le avisa que la opción es inválida.

```
#include <iostream>
int principal ( ) {
    carácter x ;
    flotador y , s , r ;

    std :: cout << " ¿que tipo de cliente es? " << std :: endl ;
    std :: cout << " 1. cliente vip " << std :: endl ;
    std :: cout << " 2. cliente regular " << std :: endl ;

    std :: cin >> x ;
    cambiar ( x ) {
        caso '1' :
            std :: cout << " usted tiene el 20% " << std :: endl ;
            std :: cout << " ingrese el costo de sus productos " << std :: endl ;
            std :: cin >> y ;
            std :: cout << " su total fue de " << std :: endl ;
            std :: cout << y - ( y * 0.20 ) << std :: endl ;
            romper ;

        caso '2' :
            std :: cout << " ¿cual es el total de su compra? " << std :: endl ;
            std :: cin >> s ;
            std :: cout << " ¿cuantos productos comprar? " << std :: endl ;
            std :: cin >> r ;
            si ( r > 3 ) {
                std :: cout << " usted tiene el 10% " << std :: endl ;
                std :: cout << " su total es de " << std :: endl ;
                std :: cout << s - ( s * 0.10 ) << std :: endl ;
                romper ;
            }
            demás {
                std :: cout << " usted no cuenta con descuentos " << std :: endl ;
                std :: cout << " su total es de " << std :: endl ;
                estándar :: cout << s ;
                romper ;
            }
        }
    }
    por defecto :
    std :: cout << " opción invalida " << std :: endl ;
    **** romper ; }
}
```

Figure 6: codigo

20 Preguntas del 4c:

¿Qué tipo de programa necesitas crear?

Un programa que simule un cajero automático o un menú bancario simple, donde el usuario pueda: • Ingresar un PIN (validación de acceso). • Retirar dinero (con verificación de saldo). • Consultar el saldo disponible. • Salir del menú.

21 Preguntas del 4.1c:

¿Qué validaciones son necesarias? 1. PIN correcto/incorrecto → si el PIN no es válido, debe salir. 2. Monto a retirar: • No puede ser negativo ni cero. • No puede ser mayor al saldo actual. 3. Entrada del menú → si el usuario digita una opción inválida, mostrar mensaje de error. 4. Control de tipos de datos → evitar que el programa falle si se ingresa un dato que no es número.

22 Preguntas del 4.2c:

¿Qué casos especiales debes considerar?

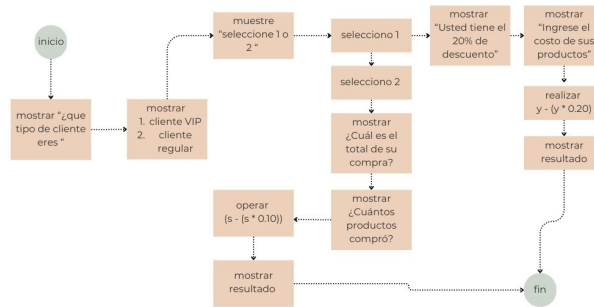
Que el usuario ingrese mal el PIN.

Que intente retirar más dinero del que tiene.

Que ingrese letras o símbolos en vez de números (esto normalmente se maneja con validaciones extra).

Que retire exactamente todo el saldo (saldo queda en 0).

Que intente retirar un valor negativo o cero.



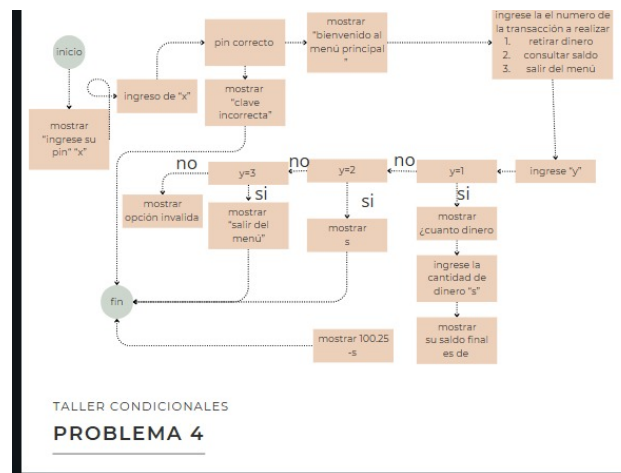
TALLER CONDICIONALES
PROBLEMA 3

Figure 7: código

23 Preguntas del 4.3c:

¿Qué entrada necesita el programa del usuario?

1. El PIN (número).
2. La opción del menú (1, 2 o 3).
3. En caso de retiro: el monto a retirar (número).



TALLER CONDICIONALES
PROBLEMA 4

Figure 8: código

24 Preguntas del 5d:

¿Qué tipo de programa necesitas crear?

Un programa en C++ que reciba la fecha de nacimiento de un usuario (día y mes) y determine su signo zodiacal occidental según la astrología.