

**Universidad Mariano Gálvez de Guatemala**

**Facultad de Ingeniería en Sistemas de la Información y la Computación**

**Campus Villa Nueva, Guatemala**

**Ingeniería en Sistemas -5090**

**Curso: PROGRAMACIÓN I**

**Licenciado/Ingeniero titular: Ing. Carlos Alejandro Arias**

## **Creación de clases en C++ Lab#6**

**ESTUDIANTES:** Carlos Eduardo García Cortez

**CARNET:** 5090-24-14824

**FECHA:** 10/03/2025

# Parte 1: Realización del código

Objetivo: Desarrollar una clase en C++ llamada CuentaBancaria, aplicando condicionales (if-else, switch-case) para simular operaciones bancarias. Se busca que los estudiantes integren al menos 5 condiciones en la lógica de la clase sin el uso de arreglos, vectores o punteros.

## Instrucciones

1. Creación de la clase CuentaBancaria con los siguientes atributos:
  - o string titular
  - o double saldo
  - o bool activa
  - o int intentosFallidos
2. Implementar los siguientes métodos:
  - o depositar(double monto): Aumenta el saldo si la cuenta está activa.
  - o retirar(double monto): Disminuye el saldo si hay fondos suficientes.
  - o consultarSaldo(): Muestra el estado de la cuenta.
3. Condiciones a incluir (Elijan 5 mínimo de las siguientes o creen sus propias reglas):
  - o No se puede retirar más dinero del saldo disponible.
  - o Si el saldo cae por debajo de 100, mostrar una advertencia.
  - o Si hay 3 intentos fallidos de retiro, la cuenta se bloquea.
  - o No se pueden hacer depósitos si la cuenta está bloqueada.
  - o No se pueden hacer retiros si la cuenta está bloqueada.
4. Estructura del programa:
  - o Implementar la clase CuentaBancaria.
  - o Crear un menú interactivo con switch-case que permita al usuario:
    - i. Depositar
    - ii. Retirar
    - iii. Consultar saldo
    - iv. Salir o Probar diferentes escenarios para asegurarse de que las condiciones funcionan correctamente.
5. Manejo del programa
  - o Crear una función para manejar punteros en el programa, usando gotoxy.
  - o Crear una función para limpiar el buffer de pantalla una vez se utilice una función del programa.

## Ejemplo de uso del programa

```
--- Menú ---  
1. Depositar  
2. Retirar  
3. Consultar Saldo  
4. Salir  
Seleccione una opción: 2  
Ingrese monto a retirar: 600  
Saldo insuficiente. Intento fallido #1  
Seleccione una opción: 2  
Ingrese monto a retirar: 600  
Saldo insuficiente. Intento fallido #2  
Seleccione una opción: 2  
Ingrese monto a retirar: 600  
Saldo insuficiente. Intento fallido #3  
La cuenta ha sido bloqueada por intentos fallidos.  
Seleccione una opción: 3  
Titular: Juan Pérez  
Saldo: Q500  
Estado: Bloqueada
```

Entrega: Sube tu código en un archivo .cpp en GitHub y elabora con comentarios explicativos y un resumen de lo aprendido. Recuerda que este laboratorio debe tener comentarios personales para la elaboración.

Evaluación: Estructura de la clase y uso de condicionales (40%), Implementación correcta de al menos 5 condiciones (30%), Uso adecuado de switch-case para el menú (20%) y Pruebas y funcionamiento del programa (10%).

## Parte 2: Reflexión

Al ver el código en C++ dentro del repositorio de Github, he llegado a la conclusión de que este laboratorio es el que ha sido más desafiante en el curso de Programación, ya que al usar funciones específicas para cada uso de variables (activar, desactivar, desbloquear, ver saldo, depositar y retirar 30 cuentas), incluyendo el uso de punteros como gotoxy y un limpiador de pantalla como un system("cls") y un void cleanScreen() { dentro de un código de C++ es un poco tedioso por los errores que pueden generar, pero que el resultado es satisfactorio una vez se ejecuta correctamente y cumple con las 5 condiciones que se eligieron en el laboratorio para este trabajo.

El trabajo es publicado dentro del repositorio de Github, donde estará este informe y el código para que pueda ser ejecutado y analizado para ver que cada condición esté ejecutándose correctamente.

<https://github.com/Xplod883/Laboratorio-7.git>