

# Proyecto de Aplicación No. 1 2024s2 Sistema de Gestión de Préstamos Bancarios

#### **Enunciado**

El objetivo de este proyecto es desarrollar un sistema de gestión de préstamos bancarios utilizando los principios avanzados de programación orientada a objetos (POO) en C++. Los estudiantes deberán aplicar conceptos de polimorfismo, herencia, abstracción y encapsulamiento; así como utilizar métodos, arreglos, métodos virtuales y destructores.

El banco se encarga de generar prestamos, los cuales tiene propiedades como monto, tasa de interés y plazo para el pago de estos. Para cada préstamo se tiene la opción del cálculo de los intereses a la fecha y el de mostrar la información del préstamo.

Existen 3 tipos de préstamos, cada uno con sus reglas específicas para el cálculo de intereses:

Los préstamos personales (interés simple – 7.85%)

I = (P \* i \* n) / 360

I = Interés a pagar.

P = Saldo de Principal a la fecha de pago.

i = Tasa de interés anual expresada en porcentaje.

n = Número de días transcurridos entre la última fecha de pago de intereses y la fecha actual de pago.

Los préstamos hipotecarios (compuesto – banco 6% y FHA 1.26%)

Intereses = Capital pendiente x (Euríbor + Diferencial) / 12

En esta fórmula, "Capital pendiente" es la cantidad de dinero que queda por pagar en la hipoteca, "Euríbor" es el valor del índice de referencia en el momento de la revisión, y "Diferencial" es el margen que el banco cobra sobre el índice de referencia.

Los préstamos de automóviles (interés anual 9.75%)

Cuota Mensual =  $(P * i) / (1 - (1 + i) ^ -n)$ 

P = Monto de Préstamo

i = Tasa de Interés Mensual

n = Plazo en Meses



## Requerimientos de implementación

- Los atributos deben ser accesibles solo a través de métodos `get` y `set` adecuados.
- Utilizar polimorfismo para manejar un arreglo de objetos de tipo `Prestamo` que puedan ser instancias de cualquiera de las clases derivadas.
- Implementar un método para mostrar la información de todos los préstamos utilizando polimorfismo.
- Crear una clase `Banco` que administre una lista (arreglo dinámico) de objetos de tipo `Prestamo`.
- Implementar métodos en la clase `Banco` para agregar, eliminar y mostrar préstamos.
- Asegurarse que los destructores se llamen correctamente para liberar memoria cuando los objetos se eliminen.
- Crear un menú en modo consola que permita al usuario interactuar con el sistema, permitiendo acceder a las siguientes opciones:
  - Agregar un nuevo préstamo.
  - Eliminar un préstamo existente.
  - Mostrar todos los préstamos.
  - Calcular y mostrar el interés de cada préstamo.
- El proyecto debe ser modular, dividiendo las clases en archivos `.h` y `.cpp` separados.
- Se debe utilizar manejo de excepciones para gestionar posibles errores, como entradas no válidas.

### Criterios de evaluación

- Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
- Correcta implementación de la herencia, polimorfismo, abstracción y encapsulación.
- Uso adecuado de métodos virtuales y destructores.
- Gestión eficiente del arreglo de préstamos en la clase `Banco`.
- Claridad y modularidad del código.



- Funcionalidad: DEBE cumplir a cabalidad con todos los requerimientos.
- Manejo adecuado de errores y excepciones.

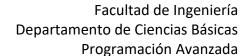
Codificación y corrida  Total	65 pts.  100 pts.
Manual de usuario	05 pts.
Documentación: carátula, introducción, conclusiones, recomendaciones	10 pts.
Diseño: diagrama de flujo.	15 pts.
Análisis: entradas, salidas, procesos, restricciones	05 pts.

## **Entregables**

- Se realizarán 2 entregables de la siguiente manera:
  - Fase1: entrega el 21 de agosto antes de las 23:59 horas.
    - Análisis y Diseño
  - Fase 2: entrega el 9 de septiembre antes de las 07:00 horas.
    - Carpeta de solución del proyecto (código fuente con documentación interna)
    - Programa ejecutable
    - Documentación externa con las correcciones y mejoras realizadas al análisis y diseño
    - Manual de usuario
- Fecha de presentación del proyecto: 9 de septiembre en periodo de Laboratorio

### **Consideraciones**

- Este proyecto se trabajará en grupos de 2 integrantes.
- Durante la calificación del proyecto es requerido que el estudiante demuestre el domino completo de lo implementado para que se le asigne la nota obtenida.
- Toda solución presentada debe compilar correctamente para poder tener derecho a revisión.





- El uso de código de terceros debe estar documentado, referenciado y justificado su uso, debe demostrarse el dominio completo de lo implementado, IMPORTANTE: No puede usar librerías de terceros para implementar estructuras lineales.
- Cada entrega deberá hacerse en el espacio habilitado en el portal. Para la calificación se descargará el proyecto entregado por esta vía. No se aceptan entregas vía correo electrónico u otro medio.
- Se podrá demandar que en la calificación presencial del proyecto se realicen cambios de funcionalidad.
- COPIA PARCIAL O TOTAL DEL PROYECTO TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA COORDINACIÓN DEL ÁREA PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES A LOS ALUMNOS DEL EQUIPO DE TRABAJO.