



Actividad | 3 | Programa 2 (Parte 2)

Desarrollo de Aplicaciones Móviles IV

Ingeniería en Desarrollo de Software



academi**ag**lobal

TUTOR: Marco Rodríguez Tapia

ALUMNO: Carlos Fco Estrada Salazar

FECHA: 07/Agosto/2025

ÍNDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	
Codificación	6
Prueba del programa	7
CONCLUSIÓN	8

GitHub Link:

GitHub Code Link: <https://github.com/Calinny17/Desarrollo-de-aplicaciones-m-viles-IV/blob/a6d796961019a2f647b1dad6f5124c125d47a094/bancaEnLineaCompleto>

INTRODUCCIÓN

En la actualidad, el desarrollo de soluciones digitales en el ámbito financiero representa una necesidad indispensable para ofrecer a los usuarios herramientas accesibles, seguras y eficientes. Las aplicaciones de banca en línea permiten a los clientes realizar operaciones desde cualquier lugar y en cualquier momento, facilitando el control y la gestión de sus finanzas. En este contexto, se presenta la segunda parte del desarrollo de un programa en el lenguaje Swift para el Banco Mexicano. Esta aplicación simula un sistema bancario básico con funcionalidades interactivas que permiten al usuario realizar depósitos, retiros, consultar su saldo y cerrar sesión.

Durante la primera parte de este proyecto, se implementaron con éxito las funciones correspondientes a las opciones de **Depósito** y **Retiro**, incluyendo validaciones y controles de flujo para garantizar una experiencia fluida. En esta nueva etapa, se completará la funcionalidad del menú principal con las opciones faltantes: **Saldo** y **Salir**. La opción **Saldo** permitirá al usuario visualizar el monto disponible en su cuenta y decidir si desea continuar operando. Por otro lado, la opción **Salir** brindará al usuario la posibilidad de finalizar la sesión de forma clara y segura.

La implementación de estas funciones busca no solo completar el programa bancario, sino también reforzar el uso de estructuras de control, lectura de datos y salida de información en Swift, promoviendo buenas prácticas de programación. Esta actividad representa un paso importante para la comprensión y diseño de sistemas interactivos orientados al usuario, que pueden escalar a aplicaciones reales en el mundo financiero.

DESCRIPCIÓN

El escenario presentado en esta actividad corresponde a la segunda etapa del desarrollo de un sistema de banca en línea para el Banco Mexicano, utilizando el lenguaje de programación Swift. En la primera parte de la actividad se implementaron las funciones de **depósito** y **retiro**, permitiendo al usuario realizar transacciones básicas sobre su cuenta de manera interactiva. Ahora, el objetivo es **completar el programa** añadiendo la funcionalidad a las dos opciones restantes del menú principal: **Saldo** y **Salir**.

La opción **Saldo** debe permitir al usuario visualizar en pantalla el monto actual disponible en su cuenta, lo cual es una característica fundamental en cualquier plataforma bancaria. Además, debe mantener la lógica interactiva del sistema preguntando si se desea realizar otra operación, tal como se ha hecho en las funciones anteriores. Por su parte, la opción **Salir** cumple la función de cerrar la sesión del usuario de forma clara y directa, finalizando así la ejecución del programa de forma segura.

Este tipo de implementación no solo enseña conceptos de programación como estructuras condicionales, control de flujo y manejo de entradas, sino que también refleja situaciones reales en el diseño de software orientado al cliente, fomentando el desarrollo de soluciones digitales útiles y prácticas para la vida diaria.

JUSTIFICACIÓN

La implementación de un sistema bancario básico mediante programación en Swift, como el que se desarrolla en esta actividad, representa una solución ideal para abordar las necesidades funcionales y pedagógicas del escenario planteado. Por un lado, desde el punto de vista funcional, este tipo de solución permite simular de manera clara y estructurada las operaciones esenciales de una banca en línea: depósito, retiro, consulta de saldo y cierre de sesión. Cada una de estas acciones refleja los servicios más utilizados por los clientes en plataformas digitales reales, por lo que su implementación resulta altamente representativa de un entorno financiero moderno.

Desde el enfoque educativo, esta solución ofrece una excelente oportunidad para que los estudiantes refuercen sus conocimientos en programación estructurada utilizando Swift, un lenguaje ampliamente utilizado en el desarrollo de aplicaciones para iOS. Al desarrollar cada opción del menú, los alumnos aprenden a manejar estructuras de control como condicionales, bucles, validación de entradas, y funciones. Esto les permite no solo comprender cómo funciona un flujo lógico en una aplicación, sino también desarrollar habilidades prácticas que pueden aplicar en proyectos más complejos en el futuro.

El enfoque modular y organizado del programa facilita su mantenimiento, escalabilidad y comprensión. Por estas razones, esta solución es adecuada y valiosa tanto para cumplir con los objetivos de la actividad académica como para preparar a los estudiantes en el diseño de aplicaciones útiles, funcionales y orientadas a la experiencia del usuario.

DESARROLLO

Codificación

```

83
84     preguntarOtraOperacion()
85 }
86
87 func consultarSaldo() {
88     print("\nTu saldo actual es: ${saldo}")
89     preguntarOtraOperacion()
90 }
91
92 func preguntarOtraOperacion() {
93     print("\nDeseas realizar otra operación (Sí/No): ", terminator:"")
94     if let respuesta = readLine()?.lowercased(), respuesta != "sí" && respuesta != "si" {
95         operacionActiva = false
96         print("\nGracias por usar Banco Mexicano. ¡Hasta pronto!")
97     }
98 }
99 }
100

```

```

func consultarSaldo() {
    print("\nTu saldo actual es: ${saldo}")
    preguntarOtraOperacion()
}

```

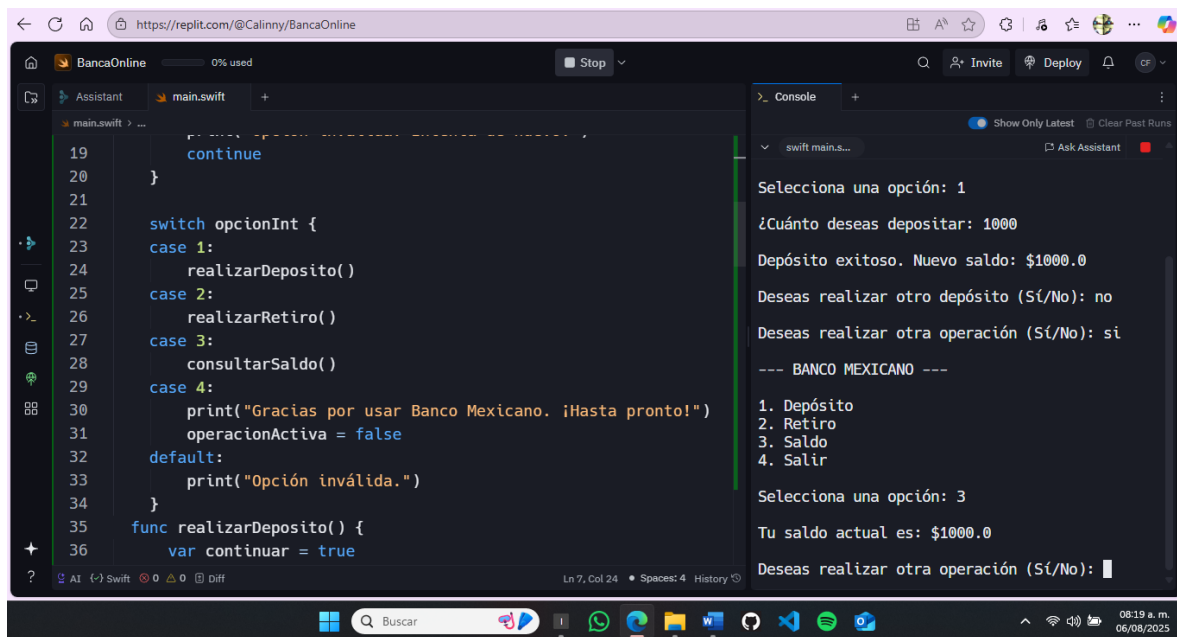
La función `consultarSaldo()` mostrará el saldo actual del usuario, además de preguntar si desea realizar otra operación. Se tubo que modificar el `switch` en el `while` principal para llamar a la función correctamente.

```

21
22     switch opcionInt {
23     case 1:
24         realizarDeposito()
25     case 2:
26         realizarRetiro()
27     case 3:
28         consultarSaldo()
29     case 4:
30         print("Gracias por usar Banco Mexicano. ¡Hasta pronto!")
31         operacionActiva = false
32     default:
33         print("Opción inválida.")
34     }

```

Prueba del programa



```
19     continue
20 }
21
22 switch opcionInt {
23 case 1:
24     realizarDeposito()
25 case 2:
26     realizarRetiro()
27 case 3:
28     consultarSaldo()
29 case 4:
30     print("Gracias por usar Banco Mexicano. ¡Hasta pronto!")
31     operacionActiva = false
32 default:
33     print("Opción inválida.")
34 }
35 func realizarDeposito() {
36     var continuar = true
```

Selecciona una opción: 1
¿Cuánto deseas depositar: 1000
Depósito exitoso. Nuevo saldo: \$1000.0
Deseas realizar otro depósito (Sí/No): no
Deseas realizar otra operación (Sí/No): si
--- BANCO MEXICANO ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
Selecciona una opción: 3
Tu saldo actual es: \$1000.0
Deseas realizar otra operación (Sí/No):

La opción "Saldo" mostrará el saldo actual del usuario, además de preguntar si desea realizar otra operación.

CONCLUSIÓN

La actividad realizada es un claro ejemplo de cómo la programación puede aplicarse para resolver necesidades reales en el ámbito financiero y en la vida cotidiana. Desarrollar un programa de banca en línea con funciones como depósito, retiro, consulta de saldo y cierre de sesión no solo permite practicar la lógica de programación, sino también comprender la importancia de crear sistemas que sean claros, seguros y fáciles de usar para los clientes.

En el campo laboral, este tipo de ejercicios ayuda a reforzar competencias esenciales para el desarrollo de software orientado a servicios financieros, un sector en el que la precisión, la confiabilidad y la experiencia del usuario son factores críticos. Implementar funciones como la consulta de saldo y la opción de salir del sistema son pasos importantes para brindar al usuario control sobre sus operaciones y garantizar que la interacción con el sistema sea completa y satisfactoria.

En la vida cotidiana, comprender cómo funcionan estos procesos nos permite valorar el trabajo detrás de las aplicaciones que utilizamos a diario, además de desarrollar una mentalidad lógica y orientada a la solución de problemas. También fomenta la capacidad de planificar y estructurar proyectos, cualidad esencial tanto en el ámbito profesional como personal. Este tipo de actividades no solo fortalecen habilidades técnicas, sino también la capacidad de pensar en las necesidades reales de las personas y cómo la tecnología puede satisfacerlas de manera eficiente.