

## **Actividad |2| Programa 2 (parte 1).**

### **Desarrollo de Aplicaciones Móviles IV.**

Ingeniería en Desarrollo de Software.



TUTOR: Marco Alonso Rodriguez Tapia.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 27/11/2025.

|                             |           |
|-----------------------------|-----------|
| <b>Introducción.</b>        | <b>3</b>  |
| <b>Descripción.</b>         | <b>3</b>  |
| <b>Justificación.</b>       | <b>4</b>  |
| <b>Desarrollo.</b>          | <b>4</b>  |
| <b>Codificación.</b>        | <b>5</b>  |
| <b>Prueba del Programa.</b> | <b>8</b>  |
| <b>Conclusion.</b>          | <b>14</b> |
| <b>Referencias.</b>         | <b>15</b> |

## Introducción.

En esta segunda actividad de la materia Desarrollo de Aplicaciones Móviles IV, nos enfocaremos en la realización de una del Banco Mexicano en el cual nos embarcamos en un proyecto de modernización digital que requiere el desarrollo de una aplicación de banca en línea robusta y fácil de usar. Este sistema, diseñado en el lenguaje Swift, es fundamental para ofrecer a los clientes la capacidad de gestionar sus finanzas con total seguridad y eficiencia. La aplicación se estructurará alrededor de un menú claro y conciso, presentando cuatro opciones principales: Depósito, Retiro, Saldo y Salir.

Nuestro enfoque inicial se centra en implementar la estructura básica y dos funcionalidades críticas. La opción de Depósito no solo permitirá al usuario ingresar una cantidad, sino que también maneja un flujo interactivo para facilitar múltiples depósitos antes de volver al menú principal. Además, la opción Salir garantizará un cierre de sesión seguro. Una vez que estas funciones de control y flujo estén consolidadas, el proyecto avanzará hacia la programación detallada de las opciones de Retiro y Consulta de Saldo, completando así el core de la plataforma de banca digital.

## Descripción.

En esta segunda actividad de la materia Desarrollo de Aplicaciones Móviles IV, continuamos con el ambicioso proyecto de modernización digital del Banco Mexicano, cuyo propósito es desarrollar una aplicación de banca en línea segura y eficiente utilizando el lenguaje Swift. En la fase previa, se implementó la estructura fundamental del menú transaccional (Depósito, Retiro, Saldo, Salir) y se puso en funcionamiento la funcionalidad de Depósito, diseñada para manejar flujos interactivos de múltiples transacciones, junto con la opción de cierre de sesión. Esta etapa se concentra en la programación detallada de las funciones restantes para completar el núcleo operacional de la plataforma. Específicamente, se abordará la funcionalidad de Retiro, la cual deberá incorporar la validación de fondos suficientes para evitar sobregiros, y la opción de Consulta de Saldo, que permitirá a los usuarios acceder a su información financiera en tiempo real. La finalización de estas tareas consolida la capacidad

transaccional básica, proporcionando una herramienta integral para la gestión de cuentas.

## **Justificación.**

La implementación de un programa de banca en línea para el Banco Mexicano se justifica por la imperante necesidad de alinearse con las expectativas del mercado digital, ofreciendo a los clientes una herramienta segura y accesible para la gestión autónoma de sus cuentas. La elección del lenguaje Swift es estratégica, ya que garantiza un desarrollo robusto, de alto rendimiento y nativo, crucial para una aplicación financiera donde la estabilidad y la rapidez son prioritarias.

La estructura de menú definida (Depósito, Retiro, Saldo, Salir) constituye el conjunto de funcionalidades esenciales para un Producto Mínimo Viable (MVP). La capacidad de manejar depósitos de manera interactiva optimiza el flujo transaccional, mientras que la futura incorporación de validaciones de fondos para el Retiro y la consulta de Saldo en tiempo real aseguran la integridad de las operaciones. Este desarrollo no solo mejora la experiencia del usuario, sino que también reduce la carga operativa del banco al automatizar las consultas y transacciones más comunes, reafirmando el compromiso con la eficiencia y la seguridad transaccional.

## **Desarrollo.**

En esta segunda parte de la materia Desarrollo de Aplicaciones Móviles IV, realizaremos el programa solicitado. Documentaremos el desarrollo de la aplicación, describiendo cada uno de sus requisitos. La aplicación debe incluir un menú de cuatro opciones de la aplicación de banco de México las cuales son (Depósito, Retiro, Saldo, Salir), así como un menú interactivo. Cada una de las funciones de la aplicación será la evidencia que se adjuntará a la documentación, e incluirá una breve explicación de su propósito.

**Link: GitHub.**

## Codificación.

En el código Swift se implementa una simulación simple de un cajero de banco mexicano en línea. Su función principal es realizar depósitos y salir de la aplicación se dará funcionalidad a dos opciones del menú como la parte 1 de la app del banco de México.

```

>_ Console      ×  main.swift  ×  +
main.swift > f preguntarSiContinuar() > ...

1  import Foundation
2  // CLASE de nombre CuentaBancaria
3  /// Representa una cuenta bancaria con funcionalidades básicas.
4  class CuentaBancaria {
5      // Variable para almacenar el saldo actual de la cuenta, inicializado en 0.
6      private var saldo: Double = 0.0
7      // Función para obtener el saldo actual (para futuros usos, como la opción 3)
8      func obtenerSaldo() -> Double {
9          return saldo
10     }
11     // Inicialización
12     init(saldoInicial: Double = 0.0) {
13         self.saldo = saldoInicial
14     }
15     // Operaciones
16     /// Realiza un depósito en la cuenta.
17     /// Parameter cantidad: La cantidad a añadir al saldo.
18     func depositar(cantidad: Double) {
19         if cantidad > 0 {
20             saldo += cantidad
21             print("\n✅ ¡Depósito exitoso! Se han depositado $(String(format: "%.2f", cantidad))")
22             print("Nuevo saldo: $(String(format: "%.2f", saldo))")
23         } else {
24             print("\n❌ Error: La cantidad a depositar debe ser positiva.")
25         }
26     }
27     // Función de Retiro (Implementación básica para completar la estructura)
28     func retirar(cantidad: Double) -> Bool {
29         if cantidad > 0 && saldo >= cantidad {
30             saldo -= cantidad
31             print("\n✅ ¡Retiro exitoso! Se han retirado $(String(format: "%.2f", cantidad))")
32             print("Nuevo saldo: $(String(format: "%.2f", saldo))")
33             return true
34         } else if cantidad <= 0 {
35             print("\n❌ Error: La cantidad a retirar debe ser positiva.")
36             return false
37         } else {
38             print("\n❌ Error: Saldo insuficiente. Saldo actual: $(String(format: "%.2f", saldo))")
39             return false
40         }
41     }
42 }
43

```

En esta parte del código define un objeto (class) llamado CuentaBancaria en Swift, que encapsula la información (saldo) y las acciones (depositar/retirar) relacionadas con una cuenta.

```

>_ Console x main.swift x +
main.swift > f preguntarSiContinuar() > ...

44 // Ejecuta la aplicación de banca en línea con el menú de opciones.
45 func ejecutarBancaEnLinea() {
46     // Se inicializa la cuenta bancaria con un saldo inicial ficticio para el ejemplo.
47     let cuenta = CuentaBancaria(saldoInicial: 1500.50)
48     var sesionActiva = true
49
50     print("=====")
51     print("          BIENVENIDO AL BANCO MEXICO          ")
52     print("=====")
53
54     // Bucle principal del menú
55     while sesionActiva {
56         mostrarMenuPrincipal()
57         // Capturar la opción del usuario
58         guard let entrada = readLine(), let opcion = Int(entrada) else {
59             print("\n❌ Entrada inválida. Por favor, ingrese un número del 1 al 4.")
60             continue // Vuelve al inicio del bucle
61         }
62
63         switch opcion {
64             case 1: // Depósito
65                 manejarDeposito(cuenta: cuenta)
66             case 2, 3: // Retiro o Saldo - Ambas opciones ahora muestran un mensaje de construcción
67                 print("\n🚧 Opción en Construcción. Por favor, intente con otra operación.")
68             case 4: // Salir
69                 print("\n👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.")
70                 sesionActiva = false // Termina el bucle y la aplicación
71             default:
72                 print("\n❌ Opción no reconocida. Por favor, seleccione una opción válida (1-4).")
73         }
74         // Si la sesión sigue activa y no es la opción Salir, preguntar por otra operación
75         if sesionActiva {
76             if !preguntarSiContinuar() {
77                 sesionActiva = false
78                 print("\n👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.")
79             }
80         }
81     }
82 }

```

La porción del código que has compartido define la función central que ejecuta y controla el flujo de toda tu aplicación de banca: el menú principal y la interacción con el usuario.

Esencialmente, esta función `ejecutarBancaEnLinea()` es el motor del programa.

- Inicialización de la Cuenta.
- Bucle principal del menú.
- Captura y validación de la entrada.
- Selección de operación.
- Continuidad de la sesión.

```

>_ Console x main.swift x +
main.swift > f preguntarSiContinuar() > ...

83 /// Muestra las opciones del menú principal.
84 func mostrarMenuPrincipal() {
85     print("\n--- MENÚ DE OPERACIONES ---")
86     print("1. Depósito")
87     print("2. Retiro")
88     print("3. Saldo")
89     print("4. Salir")
90     print("-----")
91     print("Ingrese su opción (1-4): ", terminator: "")
92 }
93 /// Maneja el flujo de depósitos, incluyendo depósitos múltiples.
94 func manejarDeposito(cuenta: CuentaBancaria) {
95     var deseaOtroDeposito = true
96     while deseaOtroDeposito {
97         print("\n--- OPERACIÓN DE DEPÓSITO ---")
98         print("Ingrese la cantidad a depositar: $", terminator: "")
99         // Función auxiliar para preguntar si desea otro depósito
100        func preguntarPorOtroDeposito() -> Bool {
101            var respuestaValida = false
102            var deseaContinuar = false
103
104            while !respuestaValida {
105                print("¿Desea realizar otro depósito? (Sí/No): ", terminator: "")
106                let respuesta = readLine()?.lowercased() ?? ""
107
108                if respuesta == "si" {
109                    deseaContinuar = true
110                    respuestaValida = true
111                } else if respuesta == "no" {
112                    deseaContinuar = false
113                    respuestaValida = true
114                } else {
115                    print("\n⚠ Respuesta no válida. Por favor, escriba 'Sí' o 'No'.")
116                }
117            }
118            return deseaContinuar
119        }
120        // Capturar y validar la cantidad
121        guard let entradaCantidad = readLine(), let cantidad = Double(entradaCantidad), cantidad > 0 else {
122            print("\n✗ Error: Cantidad inválida. Debe ser un número positivo.")
123            // Preguntar si desea intentar otro depósito y validar la respuesta
124            deseaOtroDeposito = preguntarPorOtroDeposito()
125        }

```

Esta parte del código Swift contiene dos funciones clave para la interacción con el usuario en tu aplicación bancaria: una para mostrar el menú y otra para gestionar la operación de depósito con validación de entradas.

```

>_ Console x main.swift x +
main.swift > f preguntarSiContinuar() > ...
126     continue // Vuelve al inicio del bucle 'while deseaOtroDeposito'
127 }
128 // Llama al método depositar de la clase CuentaBancaria.
129 cuenta.depositar(cantidad: cantidad)
130 // Preguntar si desea realizar otro depósito y validar la respuesta
131 deseaOtroDeposito = preguntarPorOtroDeposito()
132 }
133 }
134 /// Preguntar si el usuario desea realizar otra operación y devuelve un booleano.
135 func preguntarSiContinuar() -> Bool {
136     while true { // El bucle se repite hasta que haya una respuesta válida
137         print("\n-----")
138         print("¿Desea realizar otra operación? (Sí/No): ", terminator: "")
139
140         guard let respuesta = readLine()?.lowercased().trimmingCharacters(in: .whitespacesAndNewlines) else {
141             continue // Si no hay respuesta, repite el bucle
142         }
143         if respuesta == "si" || respuesta == "sí" {
144             return true // Responde "Sí"
145         } else if respuesta == "no" {
146             return false // Responde "No"
147         } else {
148             print("❌ Respuesta inválida. Por favor, ingrese 'Sí' o 'No'.") // Mensaje de error y repite el bucle
149         }
150     }
151 }
152
153 // Iniciar la aplicación
154 ejecutarBancaEnLinea()

```

La parte del código que has compartido cubre el manejo central de la operación de depósito, incluyendo la validación de la cantidad de dinero, la ejecución del depósito y la función auxiliar para controlar la continuidad de la sesión.

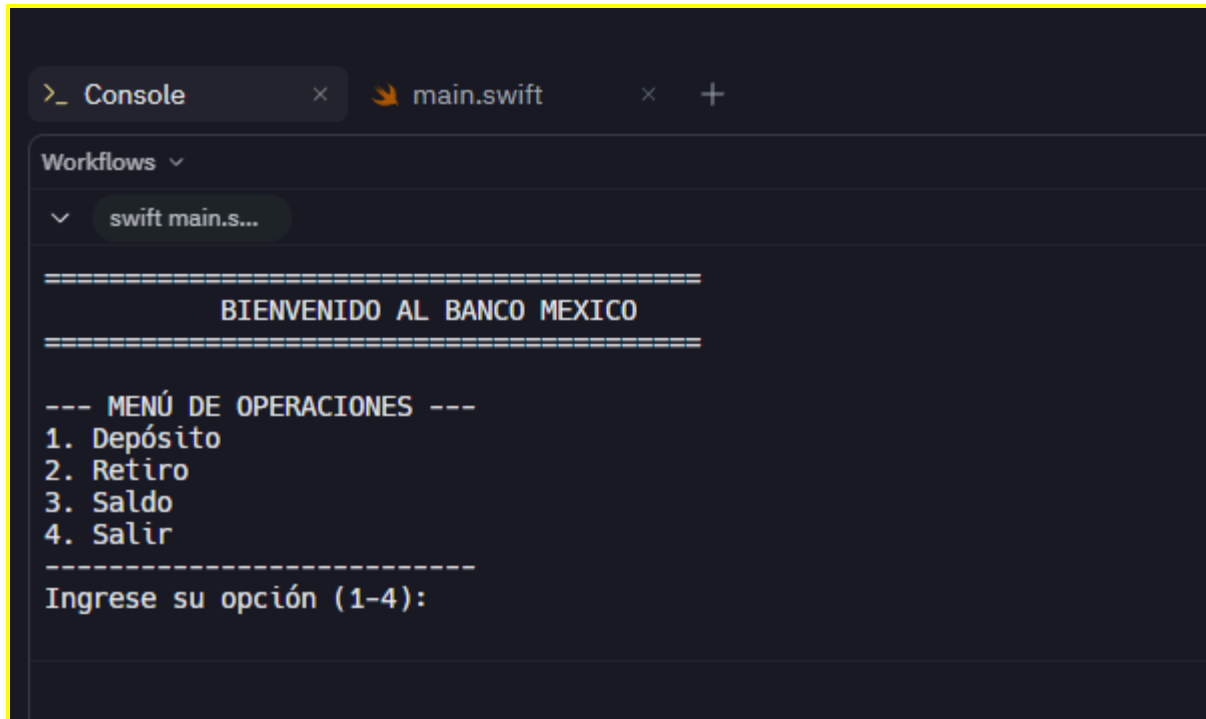
- Captura y validación de la cantidad.
- Ejecución del depósito y continuidad.
- Función auxiliar.

## Prueba del Programa.

En esta etapa realizaremos la prueba del funcionamiento de la ejecución del código Swift anterior donde anexaremos los screenshot de evidencia del funcionamiento correcto de aplicación del banco de

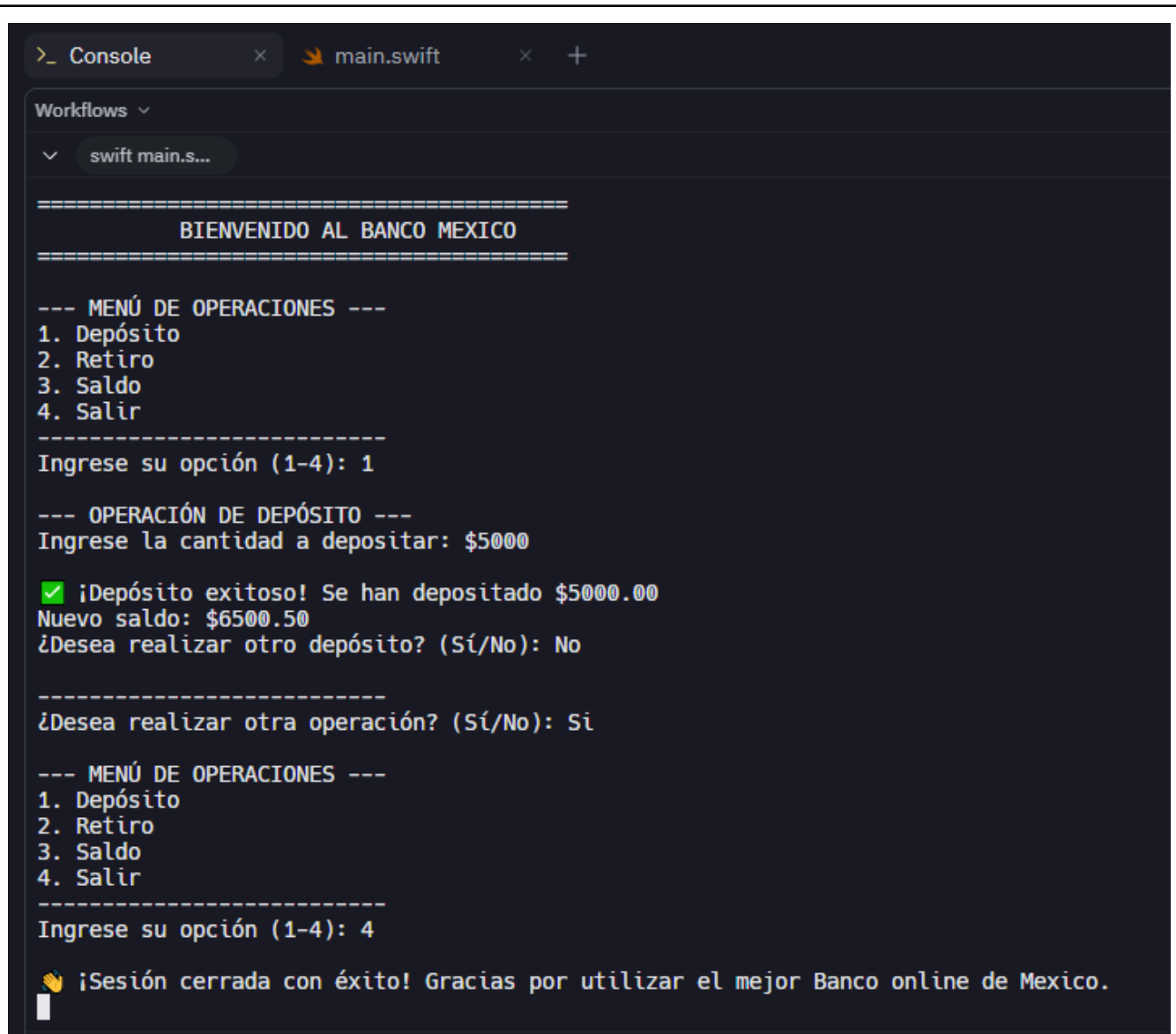


méxico en línea.



```
>_ Console × main.swift × +  
Workflows ▾  
  ▾ swift main.s...  
=====   
                BIENVENIDO AL BANCO MEXICO  
=====   
  
--- MENÚ DE OPERACIONES ---  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
-----  
Ingrese su opción (1-4):
```

En esta parte de la ejecución nos muestra la estructura del menú principal del banco de México en línea.



The screenshot shows a terminal window with two tabs: 'Console' and 'main.swift'. The 'main.swift' tab is active, displaying the output of a Swift program. The program simulates a bank interface for 'BANCO MEXICO'. It starts with a welcome message, followed by a menu of operations: 1. Depósito, 2. Retiro, 3. Saldo, and 4. Salir. The user selects option 1 (Depósito). The program prompts for the amount to deposit, and the user enters '\$5000'. A confirmation message shows a green checkmark and states that \$5000.00 has been deposited successfully, resulting in a new balance of \$6500.50. It then asks if the user wants to perform another deposit, and the user responds 'No'. Next, it asks if the user wants to perform another operation, and the user responds 'Si'. The menu is shown again, and the user selects option 4 (Salir). The program concludes with a success message: '¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.'

```
>_ Console × main.swift × +
Workflows ▾
  ▾ swift main.s...
=====
                BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 1

--- OPERACIÓN DE DEPÓSITO ---
Ingrese la cantidad a depositar: $5000

✅ ¡Depósito exitoso! Se han depositado $5000.00
Nuevo saldo: $6500.50
¿Desea realizar otro depósito? (Sí/No): No

-----
¿Desea realizar otra operación? (Sí/No): Si

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 4

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.
```

En esta parte de la ejecución se muestra el flujo de la primera opción que es depósito.

```

Workflows ▾
  ▾ swift main.s...

=====
                BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 2

🚧 Opción en Construcción. Por favor, intente con otra operación.

-----
¿Desea realizar otra operación? (Sí/No): No

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.

```

```

Workflows ▾
  ▾ swift main.s...

=====
                BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 3

🚧 Opción en Construcción. Por favor, intente con otra operación.

-----
¿Desea realizar otra operación? (Sí/No): No

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.

```

A continuación en estas pantallas anteriores entramos en las opciones 2 y 3 que no son parte de la primera actividad y nos manda un mensaje de que estamos en construcción.

```
>_ Console x main.swift x +
Workflows v
  v swift main.s...

=====
                BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 2

🚧 Opción en Construcción. Por favor, intente con otra operación.

-----
¿Desea realizar otra operación? (Sí/No): a
🚫 Respuesta inválida. Por favor, ingrese 'Sí' o 'No'.

-----
¿Desea realizar otra operación? (Sí/No): 1
🚫 Respuesta inválida. Por favor, ingrese 'Sí' o 'No'.

-----
¿Desea realizar otra operación? (Sí/No): No

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.
█
```

```
>_ Console x main.swift x +
Workflows v
v swift main.s...

=====
BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 3

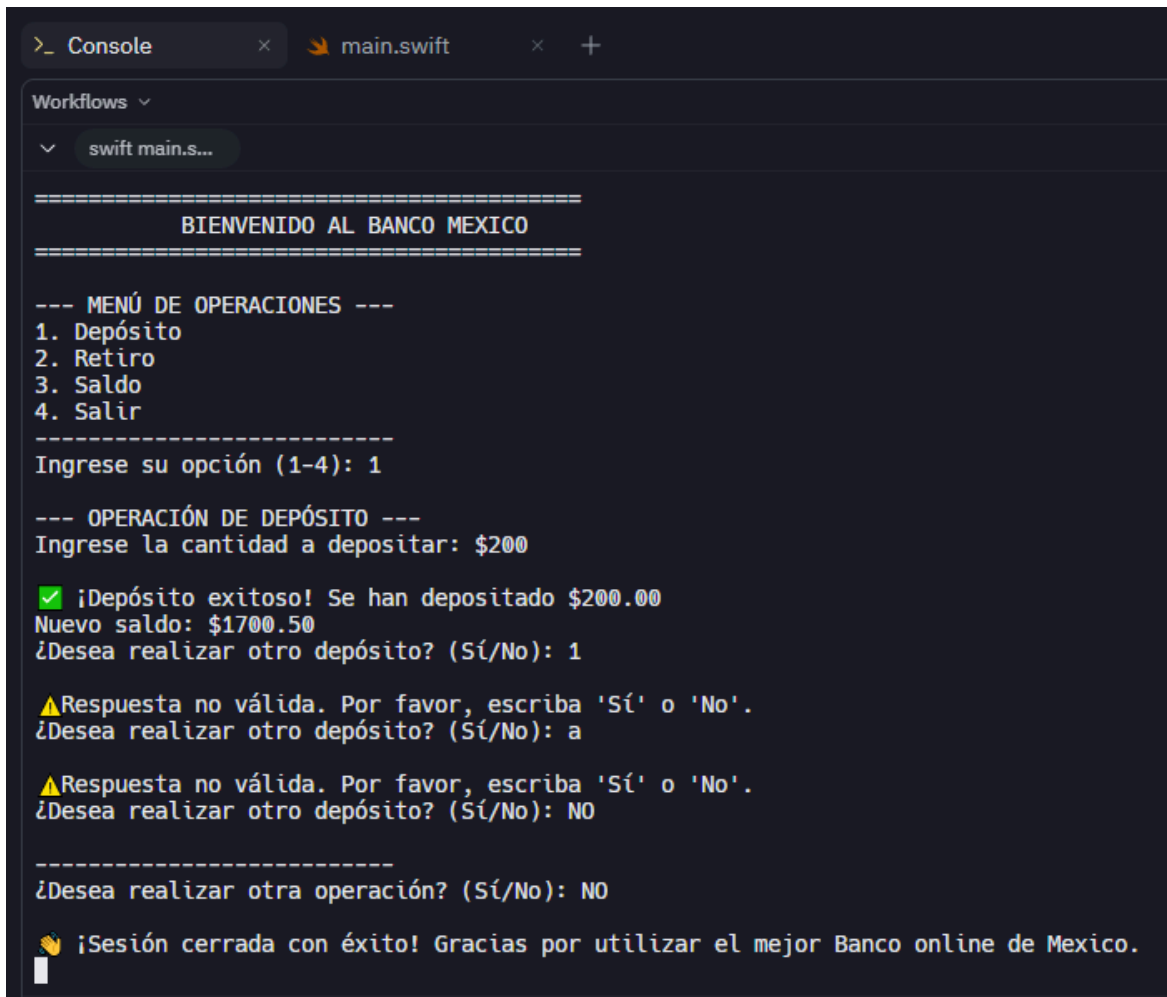
🚧 Opción en Construcción. Por favor, intente con otra operación.

-----
¿Desea realizar otra operación? (Sí/No): 1
❌ Respuesta inválida. Por favor, ingrese 'Sí' o 'No'.

-----
¿Desea realizar otra operación? (Sí/No): a
❌ Respuesta inválida. Por favor, ingrese 'Sí' o 'No'.

-----
¿Desea realizar otra operación? (Sí/No): NO

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.
█
```



```

>_ Console x main.swift x +
Workflows v
v swift main.s...

=====
BIENVENIDO AL BANCO MEXICO
=====

--- MENÚ DE OPERACIONES ---
1. Depósito
2. Retiro
3. Saldo
4. Salir
-----
Ingrese su opción (1-4): 1

--- OPERACIÓN DE DEPÓSITO ---
Ingrese la cantidad a depositar: $200

✅ ¡Depósito exitoso! Se han depositado $200.00
Nuevo saldo: $1700.50
¿Desea realizar otro depósito? (Sí/No): 1

⚠️ Respuesta no válida. Por favor, escriba 'Sí' o 'No'.
¿Desea realizar otro depósito? (Sí/No): a

⚠️ Respuesta no válida. Por favor, escriba 'Sí' o 'No'.
¿Desea realizar otro depósito? (Sí/No): NO

-----
¿Desea realizar otra operación? (Sí/No): NO

👋 ¡Sesión cerrada con éxito! Gracias por utilizar el mejor Banco online de Mexico.

```

En estas pantalla de evidencia se muestran los mensajes de error al no ingresar (Si/No).

## Conclusion.

En conclusión: El desarrollo de esta aplicación de control de inventario en Swift representa un ejercicio crucial en la creación de herramientas prácticas que impactan directamente en la eficiencia operativa. La capacidad de registrar, listar y consultar productos de manera ágil es fundamental para cualquier negocio, grande o pequeño. La relevancia de esta actividad va más allá de la simple codificación. A nivel laboral, demuestra la habilidad de traducir una necesidad real de un cliente para optimizar la gestión de stock en una solución tecnológica funcional y escalable. En la vida cotidiana, la lógica detrás de esta aplicación es aplicable a la organización personal, desde la gestión de una despensa

hasta el control de activos del hogar. Comprender cómo un sistema puede automatizar y simplificar tareas manuales es una habilidad invaluable que mejora la productividad y minimiza errores, sentando las bases para proyectos de desarrollo de aplicaciones móviles más complejos y robustos en el futuro. Como dato personal y adicional, se me hizo muy fácil el poder realizar esta actividad debido a que la consola de programación es muy intuitiva y aporta mucho apoyo al desarrollador con el autoayuda o autocompletar al estar desarrollando alguna aplicación.

## Referencias.

*Google.* (n.d.). Gemini. Retrieved November 27, 2025, from  
<https://gemini.google.com/>