

Actividad | 2 | Programa 2 (parte 1).

Desarrollo de Aplicaciones Móviles IV.

Ingeniería en Desarrollo de
Software.



TUTOR: Marco Alonso Rodríguez Tapia.

ALUMNO: Carlos Ariel Nicolini

FECHA: 25/11/2025

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
• Codificación	6
• Prueba del programa	9
Conclusión.....	12
Referencias.....	13

Introducción

Swift es un lenguaje de programación de alto nivel diseñado para ser seguro, rápido y fácil de usar. Fue creado por Apple en 2014 y se ha convertido en el lenguaje de elección para desarrollar aplicaciones para los dispositivos y plataformas de Apple, Incluyendo iPhone, iPad, Mac y más. Swift reemplazo gradualmente a Objective-C como el principal lenguaje de desarrollo de aplicaciones IOS y macOS debido a sus ventajas en términos de rendimiento, legibilidad de código y seguridad.

Swift ha ganado popularidad gracias a una serie de características clave que lo hacen atractivo para los desarrolladores.

- **Rendimiento:** A sido diseñado para ser rápido. Gracias a su compilación altamente optimizada, las aplicaciones escritas en Swift tienden a ejecutarse más rápido que las escritas en Objective-C.
- **Legibilidad:** Swift enfatiza la legibilidad del código. Su sintaxis es mas concisa y clara que la de Objective-C, lo que facilita la lectura y el mantenimiento del código.
- **Seguridad:** Swift incluye características de seguridad, como la prevención de desbordamiento de enteros y el manejo automático de la memoria, en ayudar a reducir errores comunes de programación y vulnerabilidades de seguridad.
- **Interoperabilidad:** Swift se integra fácilmente con el código Objective-C existente, lo que permite a los desarrolladores migrar gradualmente a Swift sin tener que reescribir por completo sus aplicaciones.
- **Comunidad activa:** Swift cuenta con una comunidad de desarrolladores activa y un amplio conjunto de bibliotecas y marcos de trabajo que facilitan el desarrollo de aplicaciones.

Descripción

Contextualización:

El banco mexicano necesita un programa para su banca en línea. En este se podrán realizar diversas acciones, como depositar, retirar, consultar saldo o salir. La app deberá estar creada con el lenguaje Swift.

Actividad:

Crear un programa en lenguaje Swift que funcione para la banca en línea. El menú deberá contar con las siguientes opciones:

1. Deposito.
2. Retiro.
3. Saldo.
4. Salir.

En caso de que el usuario ingrese la opción Deposito, el programa deberá ser capaz de capturar por teclado los siguientes datos:

- Cantidad a depositar.
- Preguntar si desea realizar otro deposito; si la respuesta es “No”, deberá preguntar si desea realizar otra opción.

En caso de que el usuario ingrese la opción Salir, el programa deberá mostrar en pantalla que se ha cerrado la sesión.

Nota: En la siguiente actividad, programar “Retiro” y “Saldo”.

Justificación

En esta segunda actividad realizaremos un sistema del Banco Mexicano, donde utilizaremos nuevamente la herramienta en línea de Replit, y el lenguaje de programación Swift. Esta nueva oportunidad nos servirá para no solo seguir afinando nuestro conocimiento en programación en general, sino también con el lenguaje y con la herramienta.

En conclusión, este ejercicio es una oportunidad para seguir aprendiendo sobre este lenguaje Swift el cual es muy sencillo y amigable, utilizaremos la herramienta en línea Replit la cual es una muy buena opción para realizar trabajos simples o realizar pruebas pequeñas sobre partes de código, es una alternativa muy eficaz y gratis. Esta vez realizaremos un sistema para un banco donde tendremos varias opciones tales como realizar deposito, realizar un retiro, consultar el saldo y salir, pero únicamente estarán disponibles la de depositar dinero y salir. Las otras dos opciones nos marcaran un mensaje de que en este momento no se encuentran funcionales.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-IV>

Desarrollo

Codificación

En el código realizado para este trabajo se crea una variable principal llamada `opcionIngresada`, donde el usuario escribe la opción y otra variable `saldoTotal` donde se almacena el saldo acumulado de la cuenta.

Se crea la función de `pedirSioNo()` que recibe un mensaje, que se lo muestra al usuario y solo acepta como ingresos validos si, s, no y n. Si el usuario escribe algo diferente, se lo vuelve a preguntar. Esto nos va a servir para controlar errores de escritura.

Se crea la función `deposito()`, donde se le solicita al cliente ingresar la cantidad a depositar. Se le asigna un ciclo `while` para:

Si el usuario escribe un numero valido mayor a cero, se suma al saldo y se sale del ciclo.

Si el usuario escribe un dato invalido, se le avisa con un mensaje de Cantidad invalida y se le vuelve a pedir.

Se aplica que después del depósito le pregunte si desea realizar otro deposito, de ser si, se le solicita nuevamente que ingrese la cantidad, de ser no, se sale del sistema con un mensaje de despedida.

Si no desea realizar otro deposito se le pregunta si desea realizar otra operación, de ser si, lo envía al menú, de ser no, se sale del sistema con un mensaje de despedida.

Al seleccionar en el menú la opción Retirar (2) o Consultar saldo (3), se muestra un mensaje de que el modulo no esta disponible en este momento y se envía nuevamente al menú principal.

Al seleccionar en el menú la opción salir, sale del sistema con un mensaje de despedida.

Si se selecciona cualquier valor que no sea del 1 al 4, marca opción invalida.

A continuación, se presenta captura del código:

```

1  var opcionIngresada = ""
2  var saldoTotal: Double = 0.0
3
4  func pedirSiONo(_ mensaje: String) -> Bool {
5      while true {
6          print(mensaje)
7          let respuesta = (readLine() ?? "").lowercased()
8
9          if ["si", "s"].contains(respuesta) { return true }
10         if ["no", "n"].contains(respuesta) { return false }
11
12         print("\nRespuesta inválida, escribe SI o NO.")
13     }
14 }
15
16 func deposito() {
17     while true {
18         print("\nIngresa la cantidad a depositar:")
19
20         if let entrada = readLine(), let cantidad = Double(entrada), cantidad > 0 {
21             saldoTotal += cantidad
22             print("\nSe depositaron \$(cantidad) pesos")
23             break
24         } else {
25             print("\nCantidad inválida. Intenta de nuevo.")
26         }
27     }
28 }
29
30 while true {
31     print("""
32
33     -----Banco Mexicano-----
34
35     1.- Depositar dinero
36     2.- Retirar dinero
37     3.- Consultar saldo
38     4.- Salir
39
40     Ingrese el número de una opción:
41     """)
42
43     opcionIngresada = readLine() ?? ""
44
45     if opcionIngresada == "4" {
46         print("\nGracias por usar el Banco Mexicano. ¡Hasta luego!")
47         break
48     }
49 }

```

```
50     else if opcionIngresada == "1" {
51
52         repeat {
53             deposito()
54         } while pedirSiNo("\n¿Deseas realizar otro depósito? (si/no)")
55
56         if !pedirSiNo("\n¿Deseas realizar otra operación? (si/no)") {
57             print("\nGracias por usar el Banco Mexicano. ¡Hasta luego!")
58             break
59         }
60     }
61     else if opcionIngresada == "2" || opcionIngresada == "3" {
62         print("\nMódulo no disponible en este momento. Intenta más tarde.")
63     }
64     else {
65         print("\nOpción inválida")
66     }
67 }
```


Prueba del programa

Ejecutamos el sistema y pasaremos a explicar su funcionalidad.

Al ingresar al sistema y elegir la opción Depositar dinero (1), nos solicita la cantidad a depositar (1000). Al ingresarlos nos indica la cantidad que se depositó en pesos y nos pregunta si deseamos realizar otro deposito, al responder si, nos solicita nuevamente la cantidad.

Si seleccionamos que no, nos preguntara si deseamos realizar otra operación, si respondemos que sí, nos enviara al menú. Si respondemos no, nos enviará un mensaje de despedida y se saldrá del sistema.

```
-----Banco Mexicano-----
1.- Depositar dinero
2.- Retirar dinero
3.- Consultar saldo
4.- Salir

Ingrese el número de una opción:
1

Ingrese la cantidad a depositar:
1000

Se depositaron 1000.0 pesos

¿Deseas realizar otro depósito? (si/no)
si

Ingrese la cantidad a depositar:
500

Se depositaron 500.0 pesos

¿Deseas realizar otro depósito? (si/no)
no

¿Deseas realizar otra operación? (si/no)
si

-----Banco Mexicano-----
1.- Depositar dinero
2.- Retirar dinero
3.- Consultar saldo
4.- Salir

Ingrese el número de una opción:
█
```

```
-----Banco Mexicano-----
1.- Depositar dinero
2.- Retirar dinero
3.- Consultar saldo
4.- Salir

Ingrese el número de una opción:
1

Ingrese la cantidad a depositar:
1000

Se depositaron 1000.0 pesos

¿Deseas realizar otro depósito? (si/no)
si

Ingrese la cantidad a depositar:
500

Se depositaron 500.0 pesos

¿Deseas realizar otro depósito? (si/no)
no

¿Deseas realizar otra operación? (si/no)
no

Gracias por usar el Banco Mexicano. ¡Hasta luego!
█
```

Si en el menú principal elegimos la opción de Retirar dinero, nos enviara un mensaje de Modulo no disponible en este momento, intenta mas tarde. Lo mismo sucederá si seleccionamos la opción Consultar saldo (3).

Si seleccionamos la opción Salir (4), se saldrá del sistema con un mensaje Gracias por usar el Banco Mexicano, Hasta luego.

```
-----Banco Mexicano-----  
1.- Depositar dinero  
2.- Retirar dinero  
3.- Consultar saldo  
4.- Salir  
  
Ingrese el número de una opción:  
2  
  
Módulo no disponible en este momento. Intenta más tarde.  
  
-----Banco Mexicano-----  
1.- Depositar dinero  
2.- Retirar dinero  
3.- Consultar saldo  
4.- Salir  
  
Ingrese el número de una opción:  
3  
  
Módulo no disponible en este momento. Intenta más tarde.  
  
-----Banco Mexicano-----  
1.- Depositar dinero  
2.- Retirar dinero  
3.- Consultar saldo  
4.- Salir  
  
Ingrese el número de una opción:  
4  
  
Gracias por usar el Banco Mexicano. ¡Hasta luego!  
█
```

Si seleccionamos cualquier opción en el menú principal que no sean los números del 1 al 4, enviará mensaje de opción inválida y volverá al menú principal.

```

-----Banco Mexicano-----
1.- Depositar dinero
2.- Retirar dinero
3.- Consultar saldo
4.- Salir

Ingrese el número de una opción:
5

Opción inválida

-----Banco Mexicano-----
1.- Depositar dinero
2.- Retirar dinero
3.- Consultar saldo
4.- Salir

Ingrese el número de una opción:
█

```

En la opción de Deposito, si ingresamos letras o números negativos nos indicara que es una cantidad invalida y nos pedirá ingresar nuevamente la cantidad a depositar.

<pre> -----Banco Mexicano----- 1.- Depositar dinero 2.- Retirar dinero 3.- Consultar saldo 4.- Salir Ingrese el número de una opción: 1 Ingrese la cantidad a depositar: sd232 Cantidad inválida. Intenta de nuevo. Ingrese la cantidad a depositar: █ </pre>	<pre> -----Banco Mexicano----- 1.- Depositar dinero 2.- Retirar dinero 3.- Consultar saldo 4.- Salir Ingrese el número de una opción: 1 Ingrese la cantidad a depositar: sd232 Cantidad inválida. Intenta de nuevo. Ingrese la cantidad a depositar: 0 Cantidad inválida. Intenta de nuevo. Ingrese la cantidad a depositar: -2546 Cantidad inválida. Intenta de nuevo. Ingrese la cantidad a depositar: █ </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Conclusión

En este ejercicio utilizando nuevamente la herramienta Replit y utilizando el lenguaje Swift realizaremos un sistema, en este caso un sistema para un banco. Este ejercicio es de muchísima utilidad tanto en los conocimientos técnicos que nos enseña sino también en la práctica del desarrollo del código, además esta herramienta con su IA incorporada nos permite optimizar el código, en mi caso utilice el código original de la clase y lo optimice sin perder o cambiar demasiado su arquitectura. Estos conocimientos más la utilización de medios IA son herramientas que nos van a servir muchísimo en nuestra vida laboral. Hace unos días tuvimos una charla en el trabajo sobre el uso de la IA en el desarrollo, la cual debemos aprender a utilizar y sacarle todo el provecho posible, y que, al utilizarla, no dejemos de pensar, sino en confrontarla, en saber que preguntarle, como y además indagar y cuestionar sus respuestas o consejos, por que el uso de esta tecnología es lo que viene fuerte y es necesario aprender a comprenderla.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-IV>

Referencias

El lenguaje de programación Swift. (2023, September 14). *Cev.com*.
<https://www.cev.com/blog/el-lenguaje-de-programacion-swift>

Villa, A. A. (2023, July 18). *Descubriendo Swift: Características, sintaxis y usos principales*. Profile Software Services. <https://profile.es/blog/swift/>

(N.d.). Medium.com. Retrieved November 29, 2025, from
<https://medium.com/@tahirbalarabe2/what-is-replit-ai-the-ai-powered-development-platform-46997a5124e5>

Actividad 2 y 3. (n.d.). Replit. Retrieved November 30, 2025, from
<https://replit.com/@elfoludo/Actividad-2-y-3>