



Actividad [#2] - [Programa 2 (Parte 1)]

[Desarrollo de Aplicaciones Móviles IV]

Ingeniería en Desarrollo de Software

Tutor: Marco Alonso Rodríguez Tapia

Alumno: Ricardo Rivas Rocha

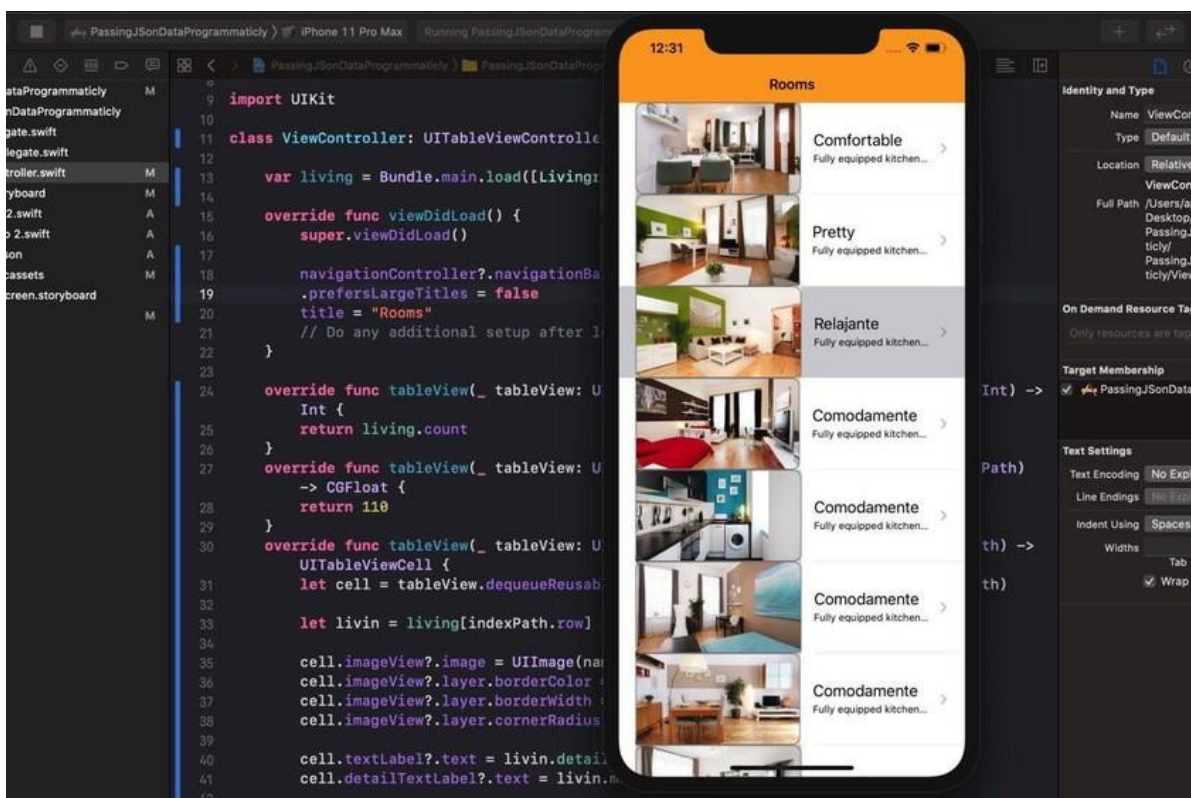
Fecha: 3-Agosto-2025

Índice

Portada	Página 1
Índice	Página 2
Introducción	Página 3
Descripción	Página 4
Justificación	Página 5
Investigación	Página 6 a 8
Desarrollo	Página 9 a 13
▪ Codificación	
▪ Prueba de la Aplicación	
Conclusión	Página 14
Referencias	Página 15

Introducción

En este segundo trabajo de esta materia vamos a ocupar la misma herramienta que nos ayudara a crear un entorno de programación en el mismo lenguaje que trabajamos en la anterior actividad a diferencia de la actividad anterior nos enfocaremos en un programa que se usa cotidianamente en entornos financieros que conocemos hoy en día, vamos a aprender mas funciones que nos maneja este lenguaje de programación ya que es esencial saber construir este mismo ya que en la actualidad muchas empresas necesitan de muchos requerimientos, adaptarse de igual manera al cambio es una gran ventaja para estas y el uso de estas tecnologías o lenguajes de programación nos puede ayudar a entender mas las estructuras de estos mismos programas adaptándonos a diferentes lenguajes de programación, ocuparemos diferentes valores constantes, constructores de objetos que nos ayudaran para la lógica de esta actividad reforzando el aprendizaje en este tipo de entornos, ya que requerimos de algunas funciones de guardado, revisión, sumas, restas y algunas otras operaciones para que nuestra app este lista para funcionar en este escenario elaboraremos la primera parte de esta misma con algunas funciones limitadas pero que nos ayuden a entender como funciona este lenguaje de programación en sí.



Descripción

En esta segunda actividad lo que realizaremos es una aplicación de un menú de banca en línea que se llamara Banco Mexicano en esta aplicación se podrán hacer diferentes acciones como lo son:

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

En este escenario lo que vamos a ocupar es la herramienta que manejamos que es replit y trabajaremos con el lenguaje Swift en donde elaboraremos el menú de Deposito y Salir esto es muy importante para que la aplicación pueda ser funcional y cumpla con los requisitos de esta actividad, empezamos con la opción de **depósito**, en cuanto el usuario ingrese a esta opción tendrá que mostrar un mensaje en pantalla con lo siguiente:

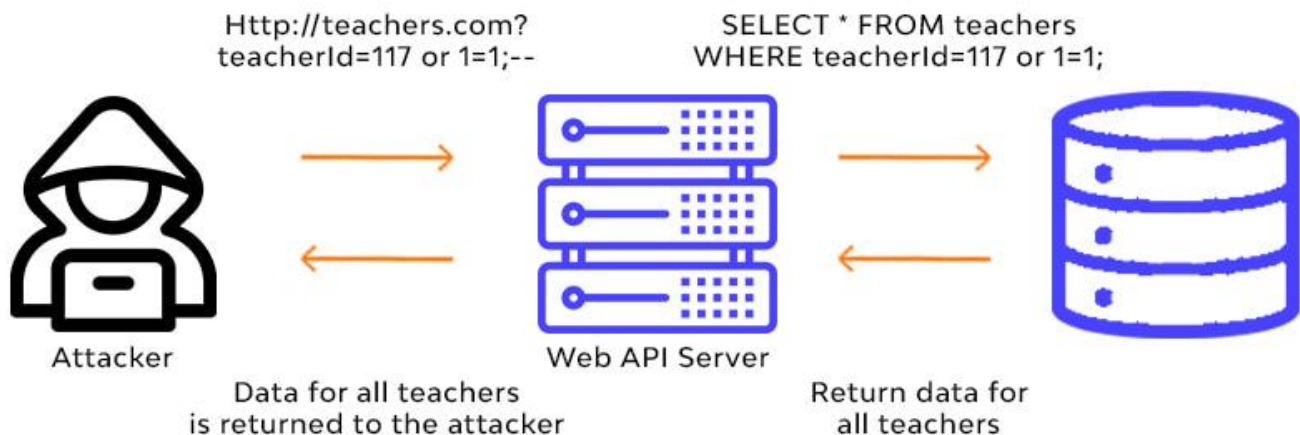
- Cantidad a depositar
- Preguntar si desea realizar otro deposito: si la respuesta es no, deberá preguntar si desea realizar otra operación.

La siguiente opción a programar va a hacer la opción de **salir** ya que es importante realizarla ya que no tiene sentido que nuestro programa no se finalice correctamente y esto va a ayudar a que finalice sin ningún detalle.

Justificación

Este tipo de aprendizaje nos va ayudar para nuestra vida cotidiana como desarrollador de software, porque este lenguaje de programación se programa en dispositivos IOS algo que nunca escuche hablar en mi escuela preparatoria hace tiempo yo creaba aplicaciones en diferentes lenguajes como lo son java, c#, c++, pero nunca supe de este lenguaje de programación ya que para nosotros saber mas lenguajes y practicarlos nos abre mas las puertas a mejores ofertas laborales en este caso de igual manera se ofrece conocimiento de las variables y estructuras de este mismo lenguaje entendiendo como es que funcionan estos mismos adaptándonos a diferentes formas de realizar aplicaciones móviles que es el caso ya que en la actualidad muchas empresas quieren su propia aplicación ya sea para reducir procesos dentro de los CEDIS de sus Empresas o como para realizar ventas en línea de igual manera generando sus sistemas para tiendas de autoservicio que pueden ser mas eficaces con la ayuda de estos programas, también algo a tener en cuenta es estar en constante cambio ya sea diseño, consultas o interfaces porque he visto que desde hace años las empresas usan sistemas muy poco confiables y algo desactualizados en lo que desencadenaría un hackeo a esas bases de datos por medio de inyección SQL, si se ve alguna vulnerabilidad es por ello que se tienen que actualizar estos sistemas hoy en día mandando versiones del sistema periódicamente.

SQL Injection



Investigación

• Cuadro comparativo de ventajas y desventajas del lenguaje Swift

En este apartado elaborare un cuadro comparativo de las ventajas y desventajas del uso de este lenguaje de programación.

VENTAJAS	DESVENTAJAS
1. Es un lenguaje de programación moderno y fácil de usar, con una sintaxis limpia y clara. Esto hace que sea ideal para principiantes y programadores experimentados.	1. Es un lenguaje de programación relativamente nuevo, por lo que hay una menor cantidad de recursos disponibles para los programadores. Esto significa que los programadores pueden tener dificultades para encontrar la información que necesitan para desarrollar aplicaciones.
2. Es un lenguaje de programación seguro, con una gran cantidad de características de seguridad incorporadas. Esto significa que los errores de seguridad son menos probables.	2. Es un lenguaje de programación de alto nivel, lo que significa que los programadores tienen que escribir más código para lograr los mismos resultados que con otros lenguajes de programación. Esto significa que los programadores tienen que dedicar más tiempo a escribir código.
3. Es un lenguaje de programación rápido, con una ejecución más rápida que otros lenguajes de programación. Esto significa que los programas escritos en Swift se ejecutan más rápido.	3. Es un lenguaje de programación de propósito general, lo que significa que no está diseñado para ninguna tarea específica. Esto significa que los programadores tienen que escribir código adicional para lograr los resultados deseados.
4. Es un lenguaje de programación compatible con muchos sistemas operativos, incluyendo iOS, macOS, watchOS y tvOS. Esto significa que los programas escritos en Swift se pueden ejecutar en muchos dispositivos diferentes.	
5. Es un lenguaje de programación compatible con muchas herramientas de desarrollo, incluyendo Xcode, AppCode y Swift Playgrounds. Esto significa que los programadores pueden usar herramientas de desarrollo modernas para crear aplicaciones para iOS, macOS, watchOS y tvOS.	

Investigación

- ¿Qué es Xcode?

Xcode es un entorno de desarrollo integrado (IDE) para macOS creado por Apple. Se utiliza para desarrollar software para Mac, iPhones, iPads, Apple Watch y Apple TV. Xcode proporciona a los desarrolladores un conjunto de herramientas para diseñar, desarrollar, analizar, crear, probar y desplegar sus aplicaciones. El IDE Xcode incluye un código editor, constructor de interfaces visuales, herramientas de depuración y análisis, y mucho más.

Origen de Xcode

El origen de Xcode se remonta a los primeros días de **macOS X**. En aquel entonces, Apple estaba desarrollando un sistema operativo basado en Unix y necesitaba una herramienta para facilitar el desarrollo de aplicaciones para el sistema operativo. Así fue como nació **Project Builder**, una herramienta de desarrollo integrada que se incluía con el sistema operativo macOS X.

Con el tiempo, Project Builder evolucionó y se convirtió en **Xcode**. La primera versión de Xcode se lanzó en 2003 junto con Mac OS X 10.3 «Panther». Desde entonces, Xcode ha sido la herramienta principal para el desarrollo de aplicaciones para Mac y dispositivos iOS. Ha pasado por varias actualizaciones y mejoras a lo largo de los años y sigue siendo una de las herramientas de desarrollo más populares y completas disponibles para el desarrollo de aplicaciones para plataformas de Apple.

Características de Xcode

Algunas de las principales características de Xcode son:

- **Editor de Código Avanzado:** Xcode ofrece un editor de código sofisticado con resaltado de sintaxis, autocompletado inteligente, fragmentos de código y navegación rápida, lo que facilita la escritura y gestión del código.
- **Constructor de Interfaces Visuales:** Incluye Interface Builder, una herramienta visual que permite a los desarrolladores diseñar interfaces de usuario de manera intuitiva, arrastrando y soltando elementos y definiendo interacciones sin necesidad de escribir código.
- **Herramientas de Depuración y Análisis:** Xcode proporciona potentes herramientas para depurar y analizar el código, incluyendo un depurador gráfico, análisis de rendimiento y capacidades de perfilado para optimizar el uso de recursos de las aplicaciones.
- **Compatibilidad con Múltiples Lenguajes y Frameworks:** Soporta una amplia gama de lenguajes de programación, como Swift, Objective-C, C++, Java y Python, así como frameworks populares de Apple como Cocoa y Cocoa Touch.

Investigación

- **¿Qué es Xcode?**
- **Simulador de iOS:** Permite probar aplicaciones en un entorno simulado que emula dispositivos iOS, facilitando la verificación de la funcionalidad y el diseño antes de la implementación en dispositivos reales.
- **Integración Continua y Pruebas Automatizadas:** Xcode incluye herramientas para configurar flujos de integración continua y pruebas automatizadas, asegurando que las aplicaciones mantengan alta calidad a lo largo del ciclo de desarrollo.

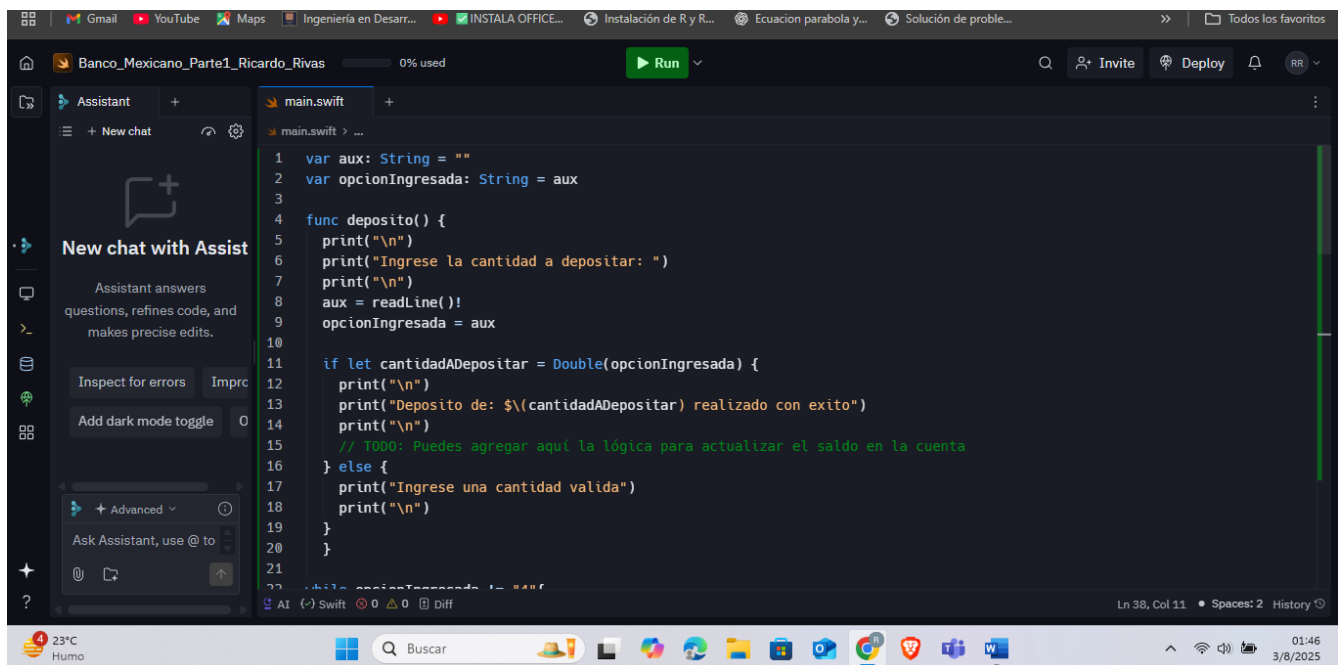
Desarrollo:

- **Codificación**

En esta parte les mostrare el proceso de codificación de cada función del programa desde la elaboración del menú de retiros de Banco Mexicano como es la opción de Deposito y Salir.

En esta siguiente captura de pantalla vamos a crear las variable auxiliar y la opción ingresada que nos permitirá guardar los datos que tecleamos como un usuario, declaramos la función de deposito con la salida de pantalla ingrese la cantidad a depositar con sus saltos de línea para después almacenar esto en la variable auxiliar para que esta accione al momento que el usuario tecle una opción válida.

Hacemos una constante if que se llamara cantidad a depositar de tipo decimal que es con un double anexamos la opción ingresada y dentro de esta imprimiremos en un mensaje de salida de pantalla deposito de concatenamos con cantidad a depositar definida por el usuario y después sigue el mensaje realizado correctamente, si el usuario quiere agregar otra cantidad que no sea doble mostrara un else que diga en salida de pantalla que la opción no es válida.



The screenshot shows a Swift code editor with a dark theme. The code is for a deposit function. It starts with two variable declarations: `var aux: String = ""` and `var opcionIngresada: String = aux`. Then, a `func deposito()` function is defined. Inside, it prints a prompt to enter the amount, reads the input into `aux`, and assigns it to `opcionIngresada`. A conditional statement `if let cantidadADepositar = Double(opcionIngresada)` checks if the input is a valid double. If true, it prints the deposit amount and a success message. If false, it prints a message asking for a valid amount. The code is currently at line 38, column 11.

```
1 var aux: String = ""
2 var opcionIngresada: String = aux
3
4 func deposito() {
5     print("\n")
6     print("Ingrese la cantidad a depositar: ")
7     print("\n")
8     aux = readLine()!
9     opcionIngresada = aux
10
11     if let cantidadADepositar = Double(opcionIngresada) {
12         print("\n")
13         print("Deposito de: $(cantidadADepositar) realizado con exito")
14         print("\n")
15         // TODO: Puedes agregar aqui la lógica para actualizar el saldo en la cuenta
16     } else {
17         print("Ingrese una cantidad valida")
18         print("\n")
19     }
20 }
21
22 // ... opcionIngresada = ""
```

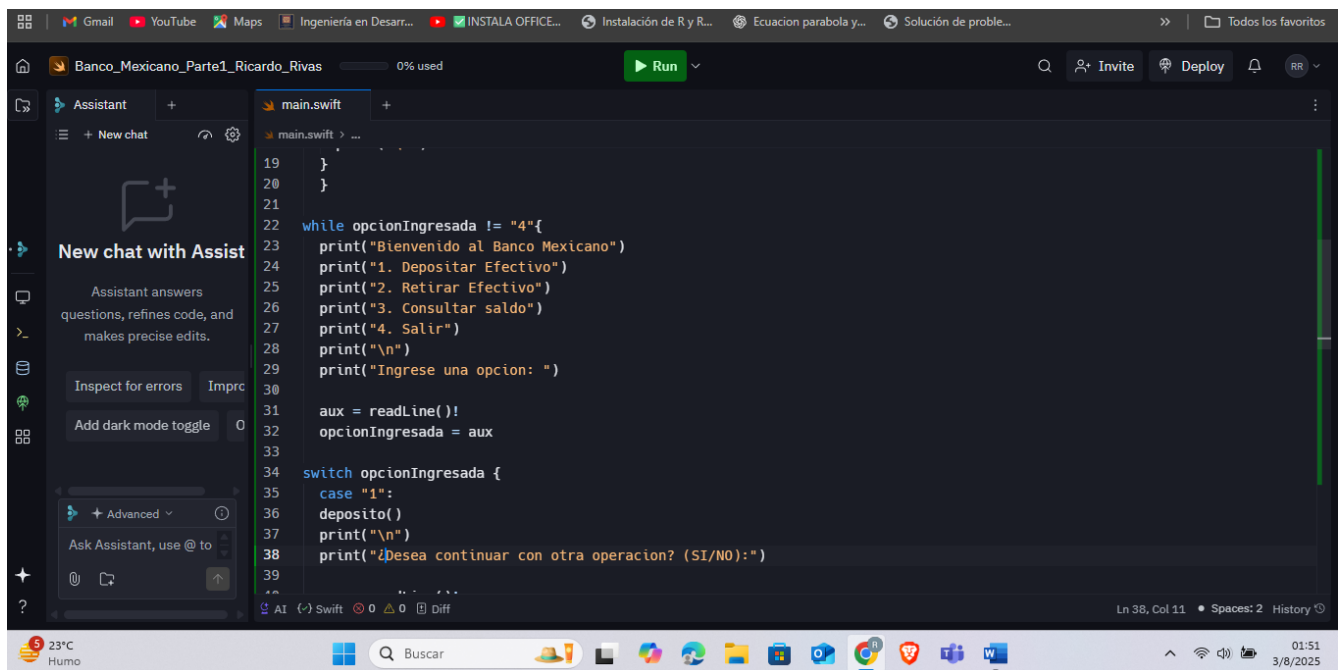
Desarrollo:

- **Codificación**

Enseguida elaboraremos un ciclo while de la variable opción ingresada en donde almacenaremos las salidas de pantalla del menú que es este:

- 1.-Bienvenido al banco mexicano
- 2.-Depositar efectivo
- 3.-Retirar efectivo
- 4.-Consultar saldo
- 5.-salir

Guardamos todo esto en nuestra variable auxiliar junto con la variable opción ingresada.



The screenshot shows a Swift code editor with a dark theme. The code is in a file named 'main.swift'. It features a 'while' loop that continues as long as 'opcionIngresada' is not equal to '4'. Inside the loop, it prints a welcome message and a menu with five options: '1. Depositar Efectivo', '2. Retirar Efectivo', '3. Consultar saldo', '4. Salir', and a blank line. It then prompts the user to 'Ingrese una opcion: '. After reading the input into 'aux' and assigning it to 'opcionIngresada', it uses a 'switch' statement to handle the selected option. For '1', it calls 'deposito()'. For '2', it prints a blank line. For '3', it prints a prompt for another operation ('¿Desea continuar con otra operacion? (SI/NO):'). The code is currently at line 38, column 11. The status bar at the bottom shows 'Ln 38, Col 11', 'Spaces: 2', and 'History'. The system tray at the very bottom indicates a temperature of 23°C and the date 3/8/2025.

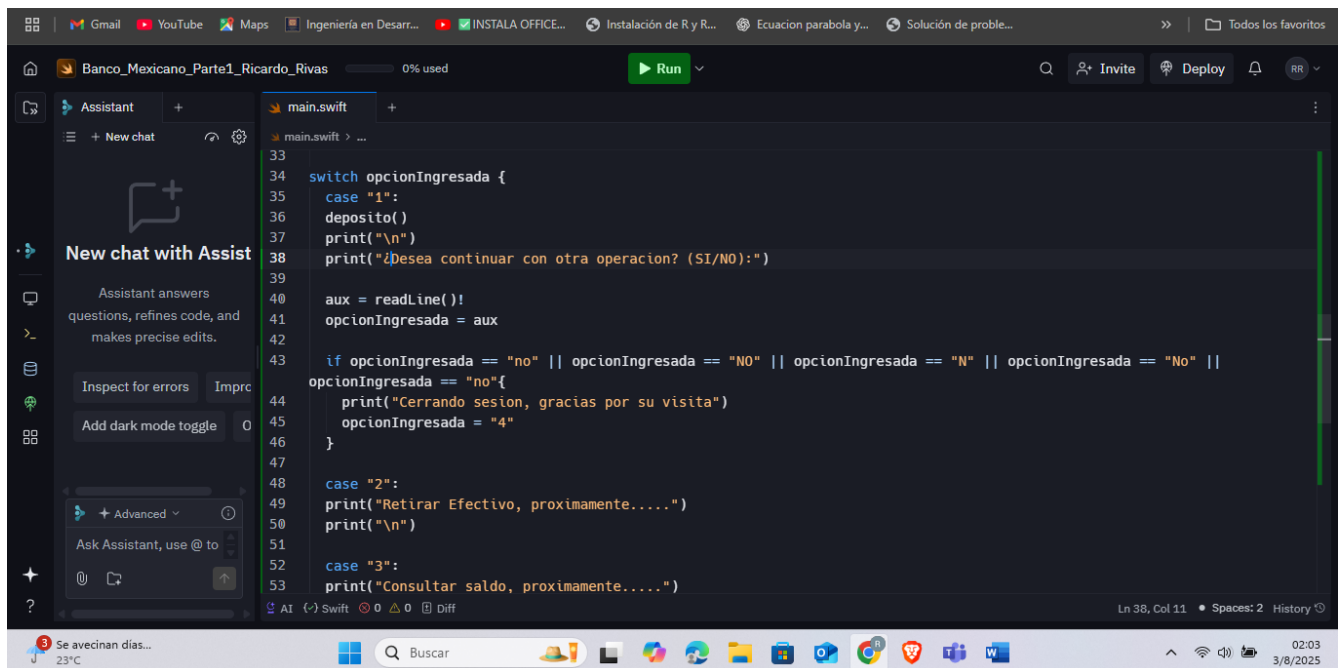
```
19 }
20 }
21
22 while opcionIngresada != "4"{
23     print("Bienvenido al Banco Mexicano")
24     print("1. Depositar Efectivo")
25     print("2. Retirar Efectivo")
26     print("3. Consultar saldo")
27     print("4. Salir")
28     print("\n")
29     print("Ingrese una opcion: ")
30
31     aux = readLine()!
32     opcionIngresada = aux
33
34     switch opcionIngresada {
35     case "1":
36         deposito()
37         print("\n")
38         print("¿Desea continuar con otra operacion? (SI/NO):")
39     }
```

Desarrollo:

- **Codificación**

Enseguida hacemos un switch en donde pondremos los casos de las opciones ingresadas en este caso elaboraremos la primera opción de nuestro menú del banco mexicano que será la de depósito la definiremos y imprimimos en pantalla desea continuar con otra operación ahí el usuario elegirá si si o no se continua con otra operación de igual forma con sus salidas de pantalla de saltos de línea para más estética, después se guardara esto en la variable auxiliar y la de opción ingresada.

Hacemos un if de opción ingresada por si el usuario dice que no desea continuar con otra operación con las respuestas variadas de no, entre las llaves de if mandaremos mensaje de salida de cerrando sesión gracias por su visita mientras la opción ingresada sea igual a 4 que es la que elaboramos.



The screenshot shows a Swift IDE window titled 'Banco_Mexicano_Parte1_Ricardo_Rivas'. The code in 'main.swift' is as follows:

```
33
34 switch opcionIngresada {
35     case "1":
36         deposito()
37         print("\n")
38         print("¿Desea continuar con otra operacion? (SI/NO):")
39
40         aux = readLine()!
41         opcionIngresada = aux
42
43         if opcionIngresada == "no" || opcionIngresada == "NO" || opcionIngresada == "N" || opcionIngresada == "No" ||
44            opcionIngresada == "no"{
45             print("Cerrando sesion, gracias por su visita")
46             opcionIngresada = "4"
47         }
48
49         case "2":
50             print("Retirar Efectivo, proxicamente.....")
51             print("\n")
52
53         case "3":
54             print("Consultar saldo, proxicamente.....")
```

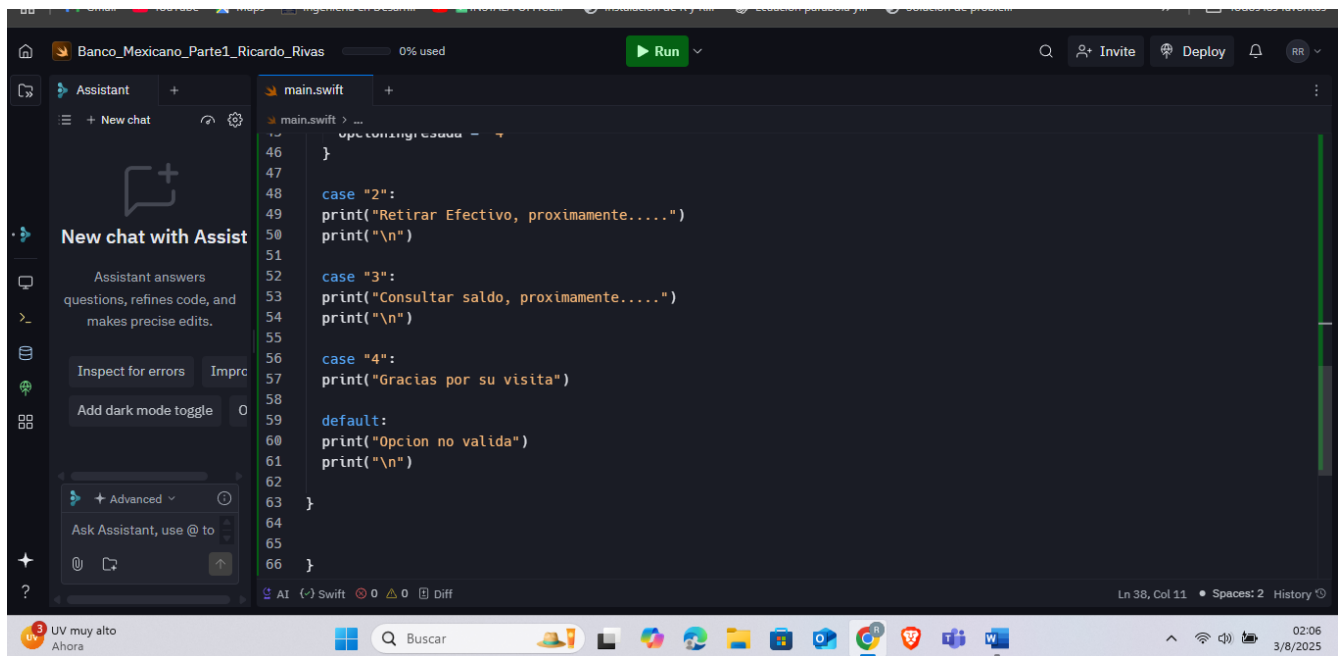
The IDE interface includes a sidebar with 'New chat with Assist', a 'Run' button, and a status bar at the bottom showing 'Ln 38, Col 11 • Spaces: 2 History'.

Desarrollo:

- **Codificación**

En este caso realizaremos los demás casos que son las demás opciones y agregaremos el default del método que estamos usando que es el switch ya que si no lo agregamos nos da errores de sintaxis nuestro código.

Estas opciones las agregamos de una vez para facilitarnos la elaboración de las otras opciones de la siguiente actividad.



The screenshot shows a code editor with a Swift file named `main.swift`. The code implements a switch statement for a banking application. The switch statement has four cases and a default case. The cases are: "2" for withdrawing cash, "3" for checking balance, "4" for saying thank you, and a default case for invalid options. The code is as follows:

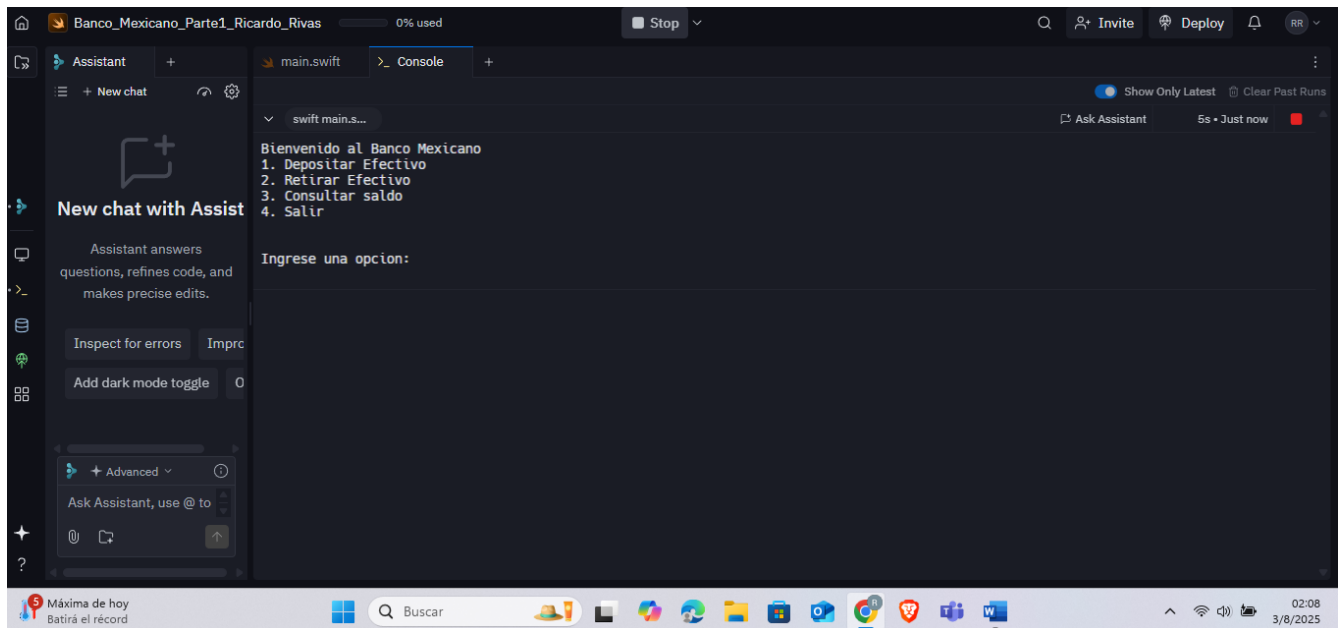
```
46 }
47
48 case "2":
49     print("Retirar Efectivo, proxicamente.....")
50     print("\n")
51
52 case "3":
53     print("Consultar saldo, proxicamente.....")
54     print("\n")
55
56 case "4":
57     print("Gracias por su visita")
58
59 default:
60     print("Opcion no valida")
61     print("\n")
62
63 }
64
65
66 }
```

The editor interface includes a sidebar with an Assistant chat, a top bar with a Run button, and a bottom status bar showing the current line and column (Ln 38, Col 11) and the number of spaces (2).

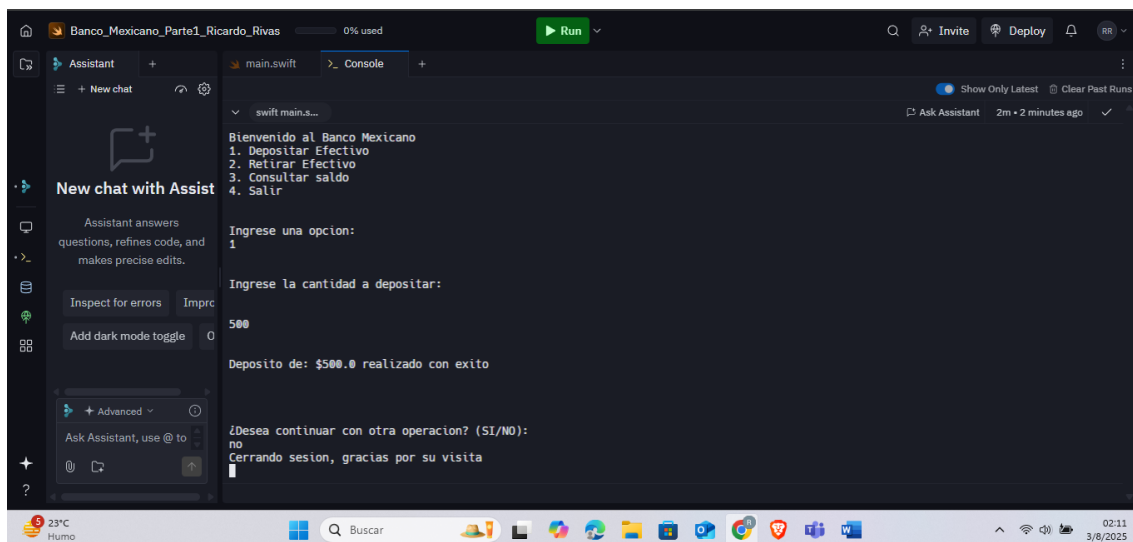
Desarrollo:

- Prueba de la Aplicación

En este escenario se muestra el menú que corresponde a la aplicación de Banco Mexicano en función.

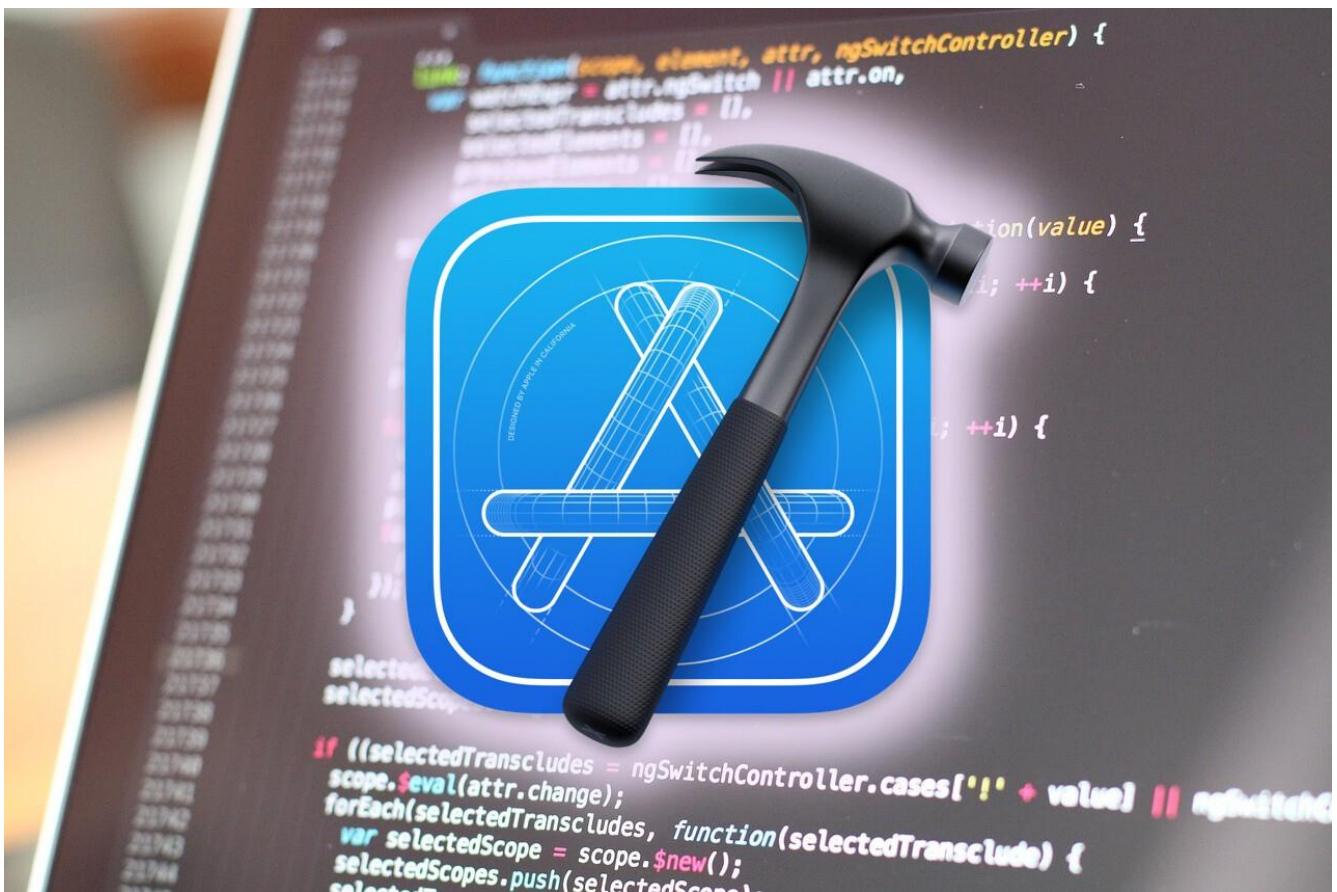


Seleccionamos la opción número 1 que es depositar y le digo que quiero depositar \$500 y me dice el programa que mi depósito de \$500 se realizó correctamente, después me muestra una opción que dice que si requiero continuar con otra operación le pongo que no y finaliza el programa.



Conclusión

En este segundo trabajo aprendimos nuevos métodos que se ocupan en el ámbito de la programación para llevar una lógica adecuada de estos mismos ya que son muy usados en la actualidad aunque hay más métodos que nos permiten optimizar el desarrollo de estos códigos en si pero como apenas nos adentramos a este lenguaje observamos como es que funcionan estas funciones y variables, es muy importante de igual manera saber para que funcionan porque unas si pueden cambiar mientras que otras no en este pequeño avance observamos como es que funciona y se elabora una aplicación en Swift aunque me gustaría hacer la aplicación con el entorno en si como es Android Studio pero para IOS para poder explorar que opciones tiene este lenguaje y trabajar con una interfaz intuitiva para el usuario final así como también guardar los valores ingresados en la cuenta del cliente para que no se nos reinicie el programa como en el entorno de replit ya que igual es de mucha ayuda poder realizar este tipo de prácticas en XCode.



Referencias

Jperez. (2023, 28 julio). *Que es Swift y cuales son sus ventajas y desventajas*. SiteCloudy.

<https://sitecloudy.com/que-es-swift-y-cuales-son-sus-ventajas-y-desventajas/>

Que es XCode. (2025, 3 agosto). Arimetrics. Recuperado 3 de agosto de 2025, de

<https://www.arimetrics.com/glosario-digital/xcode>

Link GitHub: <https://github.com/Ricardorivas94cr/Desarrollo-de-Aplicaciones-M-viles-4>