



Actividad 2 Programa 2 (parte 1)

Desarrollo de Aplicaciones Móviles IV

Ingeniería en Desarrollo de Software



TUTOR: Marco Alonso Rodríguez Tapia

ALUMNO: Darío Ismael Núñez Manriquez

Desarrollo de Aplicaciones Móviles IV

Nombre del Autor
Darío Ismael Núñez Manriquez

Telefono/celular
6981176617

correo
Darione48@gmail.com

Actividad
Programa 2 (parte 1)

Unidad
2

Fecha de entrega
25/11/2023

INDICE

INDICE	1
Introduccion	2
Descripcion	3
Justificacion	4
1. Desarrollo	5
Codificación	5
2 Prueba del Programa	8
Conclusion	10
Link de GitHub	11
LINK de replit.com	11
LINK De Google Drive	11

Introduccion

En la actualidad, la evolución tecnológica ha transformado significativamente la forma en que interactuamos con los servicios financieros. Los bancos en línea se han convertido en una parte integral de nuestra vida cotidiana, brindando la conveniencia de administrar nuestras finanzas desde cualquier lugar y en cualquier momento. En este contexto, la actividad que se abordará se centra en el desarrollo de un programa utilizando el lenguaje Swift para la banca en línea de un Banco Mexicano.

Esta actividad se enfoca en la creación de un programa de banca en línea que permita a los usuarios realizar operaciones como depósitos, retiros, consulta de saldos y finalización de sesiones. La implementación de este programa en Swift no solo busca simular un entorno bancario virtual, sino también ofrecer una experiencia interactiva y funcional a los usuarios.

La importancia de esta actividad radica en la comprensión y aplicación práctica de los principios de programación para desarrollar soluciones funcionales en el ámbito financiero. Los estudiantes aprenderán a manejar flujos de control, lógica condicional y entrada/salida de datos para crear un sistema interactivo que simule las operaciones bancarias básicas.

Descripcion

La actividad propuesta busca proporcionar una experiencia práctica en el desarrollo de aplicaciones para servicios financieros. Se pretende que los estudiantes comprendan y apliquen conceptos fundamentales de la programación, como la gestión de datos, la interacción con el usuario y el control de flujos. Específicamente, se emplearán herramientas y técnicas de Swift para construir un sistema interactivo que simule las operaciones bancarias habituales.

Esta actividad tiene un propósito educativo significativo. Ayuda a los estudiantes a comprender cómo se construyen y operan las aplicaciones de banca en línea, permitiéndoles aprender sobre la seguridad, la lógica de transacciones financieras y la experiencia del usuario. Además, les brinda la oportunidad de aplicar los conocimientos adquiridos en programación para desarrollar soluciones funcionales en un contexto financiero simulado.

En resumen, la actividad busca proporcionar una base práctica sólida en la programación Swift, al mismo tiempo que permite a los estudiantes comprender la importancia y complejidad de desarrollar aplicaciones seguras y funcionales para servicios financieros en línea.

Justificacion

La elección de emplear una solución que implique el desarrollo de un programa de banca en línea utilizando el lenguaje de programación Swift se fundamenta en varios aspectos cruciales.

En primer lugar, esta actividad proporciona una plataforma de aprendizaje ideal para los estudiantes, ya que les ofrece la oportunidad de aplicar conceptos de programación en un contexto práctico y relevante. La implementación de un sistema bancario simulado permite comprender los fundamentos de Swift y las estructuras de datos en un entorno realista, lo que mejora la comprensión y la aplicación de los conocimientos teóricos adquiridos en programación.

Además, el sector financiero es una de las áreas más sensibles en términos de seguridad y precisión. Al desarrollar una solución de banca en línea, los estudiantes aprenden sobre la importancia de la seguridad de los datos, la gestión adecuada de transacciones financieras y la validación de la entrada de datos. Esto les brinda una visión integral de cómo abordar y mitigar posibles riesgos y errores en sistemas financieros reales.

Asimismo, esta actividad fomenta el desarrollo de habilidades de resolución de problemas y pensamiento lógico. Los estudiantes deben enfrentarse a desafíos como el diseño de algoritmos para manejar transacciones, controlar flujos de información y proporcionar una interfaz de usuario intuitiva.

1. Desarrollo

Codificación

```
var saldo: Double = 0.0
var cuentaInicial: Bool = true
var operacion = true

func hacerDeposito() {
    print("Ingrese la cantidad a depositar:")
    if let deposito = readLine(), let cantidad = Double(deposito) {
        saldo += cantidad
        print("Depósito exitoso. Saldo actual: \(saldo)")

        var otroDeposito = true
        while otroDeposito {
            print("¿Desea realizar otro depósito? (s/S para sí, n/N para no)")
            if let opcion = readLine() {
                switch opcion.lowercased() {
                    case "s":
                        otroDeposito = false
                        hacerDeposito()
                    case "n":
                        otroDeposito = false
                        otraOperacion()
                    default:
                        print("Opción no válida.")
                }
            }
        }
    }

    // Una vez hecho el primer depósito, cambiar el estado de cuentaInicial
    cuentaInicial = false
} else {
    print("Entrada inválida. Por favor, ingrese un número válido.")
}

func hacerRetiro() {
    if cuentaInicial {
        print("No cuentas con saldo.")
        var otraOperacion = true
        while otraOperacion {
            print("¿Desea realizar otra operación? (s/S para sí, n/N para no)")
            if let opcion = readLine() {
                switch opcion.lowercased() {
                    case "s":
```

```

        otraOperacion = false
        mostrarMenu()
    case "n":
        otraOperacion = false
        print("Cerrando sesión de cuenta. ¡Vuelva pronto!")
        operacion = false
    default:
        print("Opción no válida.")
    }
}
}
} else {
    print("Ingrese la cantidad a retirar:")
    if let retiro = readLine(), let cantidad = Double(retiro) {
        if saldo > 0 && cantidad <= saldo {
            saldo -= cantidad
            print("Retiro exitoso. Saldo actual: \(saldo)")

            var otroRetiro = true
            while otroRetiro {
                print("¿Desea realizar otro retiro? (s/S para sí, n/N para no)")
                if let opcion = readLine() {
                    switch opcion.lowercased() {
                        case "s":
                            otroRetiro = false
                            hacerRetiro()
                        case "n":
                            otroRetiro = false
                            otraOperacion()
                        default:
                            print("Opción no válida.")
                    }
                }
            }
        }
    } else {
        print("Monto de retiro inválido o saldo insuficiente.")
        otraOperacion()
    }
} else {
    print("Entrada inválida. Por favor, ingrese un número válido.")
}
}
}

func consultarSaldo() {
    print("Su saldo actual es: \(saldo)")
    otraOperacion()
}

func otraOperacion() {

```



```

var continuarOperando = true
while continuarOperando {
    print("¿Desea continuar con otra operación? (s/S para sí, n/N para no)")
    if let opcion = readLine() {
        switch opcion.lowercased() {
            case "s":
                continuarOperando = false
                mostrarMenu()
            case "n":
                continuarOperando = false
                print("Cerrando sesión de cuenta. ¡Vuelva pronto!")
                operacion = false
            default:
                print("Opción no válida.")
        }
    }
}

func mostrarMenu() {
    print("\n¿Qué operación desea realizar?")
    print("1. Depósito")
    print("2. Retiro")
    print("3. Saldo")
    print("4. Salir")

    if let opcion = readLine(), let eleccion = Int(opcion) {
        switch eleccion {
            case 1:
                hacerDeposito()
            case 2:
                hacerRetiro()
            case 3:
                consultarSaldo()
            case 4:
                operacion = false
                print("Gracias por utilizar nuestros servicios.")
            default:
                print("Opción no válida.")
        }
    } else {
        print("Por favor, ingrese un número correspondiente a la opción deseada.")
    }
}

print("¡Bienvenido al Banco Mexicano!")
mostrarMenu()

```

2 Prueba del Programa

Primera captura

En la siguiente captura se muestra el primer paso que se quiere realizar

```
¡Bienvenido al Banco Mexicano!  
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
█
```

Segunda captura

En esta captura se muestra la realizacion de un depositop

```
¡Bienvenido al Banco Mexicano!  
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
1  
Ingrese la cantidad a depositar:  
1250  
Depósito exitoso. Saldo actual: 1250.0  
¿Desea realizar otro depósito? (s/S para sí, n/N para no)  
s  
Ingrese la cantidad a depositar:  
1  
Depósito exitoso. Saldo actual: 1251.0  
¿Desea realizar otro depósito? (s/S para sí, n/N para no)  
n  
¿Desea continuar con otra operación? (s/S para sí, n/N para no)  
s  
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
█
```

3

Captura 3

En la siguiente captura se muestra una consulta de saldo

```
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
3  
Su saldo actual es: 1251.0  
¿Desea continuar con otra operación? (s/S para sí, n/N para no)  
□
```

4ta captura se mostraria los pasos para realizar un retiro pero no pude encontrar la falla por el cual siempre me decia que no contaba con saldo

```
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
3  
Su saldo actual es: 1251.0  
¿Desea continuar con otra operación? (s/S para sí, n/N para no)  
s
```

```
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
2  
No cuentas con saldo.  
¿Desea realizar otra operación? (s/S para sí, n/N para no)  
s
```

```
¿Qué operación desea realizar?  
1. Depósito  
2. Retiro  
3. Saldo  
4. Salir  
2  
No cuentas con saldo.  
¿Desea realizar otra operación? (s/S para sí, n/N para no)  
□
```

Conclusion

La realización de la actividad enfocada en el desarrollo de un programa de banca en línea utilizando Swift tiene una relevancia significativa tanto en el ámbito laboral como en la vida cotidiana de los individuos.

Desde una perspectiva laboral, la comprensión y aplicación de los conocimientos adquiridos en esta actividad son altamente valorados en el campo de la programación y el desarrollo de software. Las habilidades desarrolladas al diseñar y crear una aplicación simulada de banca en línea son directamente transferibles a entornos profesionales.

Los desarrolladores de software que dominan el lenguaje Swift y comprenden la lógica detrás de las aplicaciones financieras tienen una ventaja competitiva en la industria. La capacidad para diseñar soluciones seguras, precisas e intuitivas para servicios financieros en línea es altamente demandada en el mundo laboral actual.

Link de GitHub

<https://github.com/dario1156/Desarrollo-de-Aplicaciones-M-viles-IV>

LINK de replit.com

<https://replit.com/@darionunez1156/Banco-MexicanoDeposito-y-Retiro>

LINK De Google Drive

https://drive.google.com/file/d/1SedN1-kZluiovsalEcnLF_S4pol13qEy/view?usp=sharing