

Actividad [#1]

[Programa 2 (parte 1) ]

[“Desarrollo de Aplicaciones Móviles IV”]

Ingeniería en Desarrollo de Software

**Tutor: Marco Alonso Rodríguez Tapia**

**Alumno: Manuel Enrique Ramirez Lopez**

**Fecha: 13/03/2023**

***Indice***

***[Introduccion](#_Toc26329)* [1](#_Toc26329)**

***[Descripción](#_Toc3264)* [2](#_Toc3264)**

***[Justificación](#_Toc27720)* [3](#_Toc27720)**

***[Investigación](#_Toc26285)* [4](#_Toc26285)**

***[Codificación](#_Toc18037)* [5](#_Toc18037)**

***[Prueba del programa](#_Toc11368)* [7](#_Toc11368)**

***[Conclusion](#_Toc21533)* [9](#_Toc21533)**

***[Referencias](#_Toc21511)* [10](#_Toc21511)**

# *Introduccion*

La programación en Swift para cajeros automáticos es compleja, pero muy interesante. Como programador, me apasiona crear soluciones innovadoras que puedan mejorar la vida de las personas y este proyecto me brinda la oportunidad de hacer precisamente eso en el sector financiero. La creación de una aplicación de cajero automático funcional y fácil de usar requiere mucho código y lógica, pero el desafío es emocionante y motivador. Mi objetivo es diseñar aplicaciones que sean seguras, eficientes y fáciles de usar. Esto requiere un conocimiento detallado de la programación de Swift y una buena comprensión de la banca. Espero trabajar en este proyecto y ver cómo mi trabajo puede marcar una diferencia positiva en la vida de las personas.

# *Descripción*

**Contextualización:** Coppel 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 programada con el lenguaje Swift.

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

Depósito

Retiro

Saldo

Salir

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

● Cantidad a depositar

● Preguntar si desea realizar otro depósito; si la respuesta es “No”, deberá preguntar si desea realizar otra operación.

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

● Si el cliente ingresa por primera vez, deberá mostrar un mensaje donde diga que no cuenta con saldo.

● Si el cliente ingresa por segunda vez, y ya se tiene dinero en la cuenta, deberá mostrar lo siguiente:

● Cantidad a retirar. El sistema deberá validar si cuenta con el saldo a retirar; y si no, mostrar un mensaje diciendo que no cuenta con el saldo tecleado.

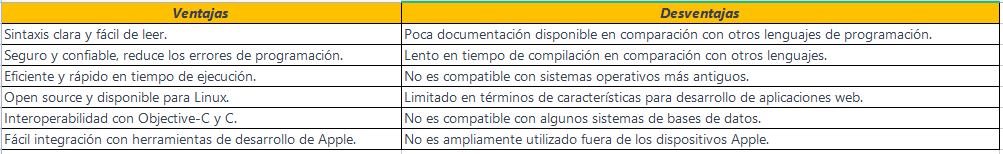
● Preguntar si desea realizar otro retiro; si la respuesta es no, deberá preguntar si desea realizar otra operación.

Nota. En la siguiente actividad programarás las opciones “Saldo” y “Salir”.

# *Justificación*

La programación en Swift es una habilidad invaluable en tecnología, ya que es el lenguaje de programación utilizado por Apple para desarrollar aplicaciones móviles y de escritorio en sus dispositivos y sistemas operativos. Por lo tanto, crear una aplicación en Swift como un cajero automatico es una tarea muy importante y útil porque los usuarios pueden acceder a los servicios financieros en cualquier momento y en cualquier lugar. Además, la programación de cajeros automáticos en Swift es un trabajo desafiante pero gratificante, ya que requiere no solo habilidades lógicas y técnicas, sino también una sólida comprensión de cómo funcionan los sistemas bancarios. Así que trabajar en este proyecto no es solo una oportunidad para mejorar mis habilidades de programación, sino también una oportunidad para crear soluciones útiles y valiosas para las personas. En resumen, la actividad de programación de cajeros automáticos en Swift se justifica por su importancia en el ámbito técnico y su capacidad para mejorar la vida de las personas a través del acceso seguro y eficiente a los servicios financieros.

# *Investigación*

***Cuadro comparativo***

***¿Qué es XCode?***

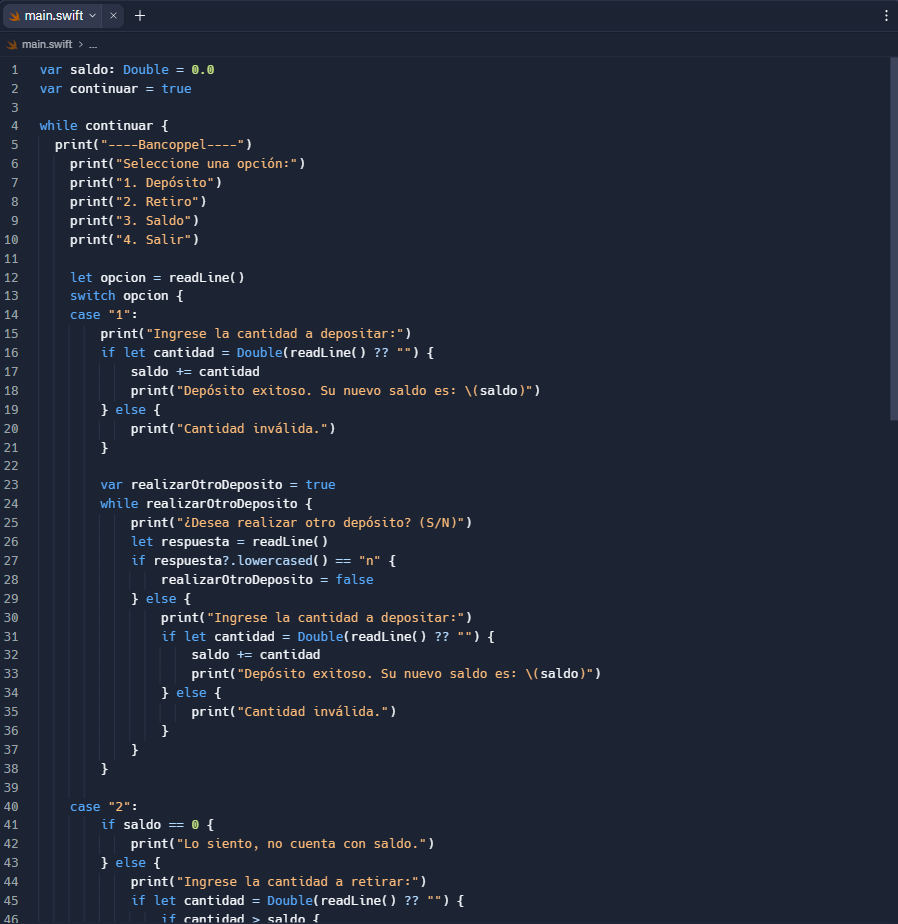
Xcode es un entorno de desarrollo integrado (IDE) creado por Apple para desarrollar aplicaciones para sistemas operativos como macOS, iOS, watchOS y tvOS. Es una herramienta poderosa y completa que los desarrolladores de software pueden usar para crear aplicaciones utilizando lenguajes de programación como Swift, Objective-C y C.

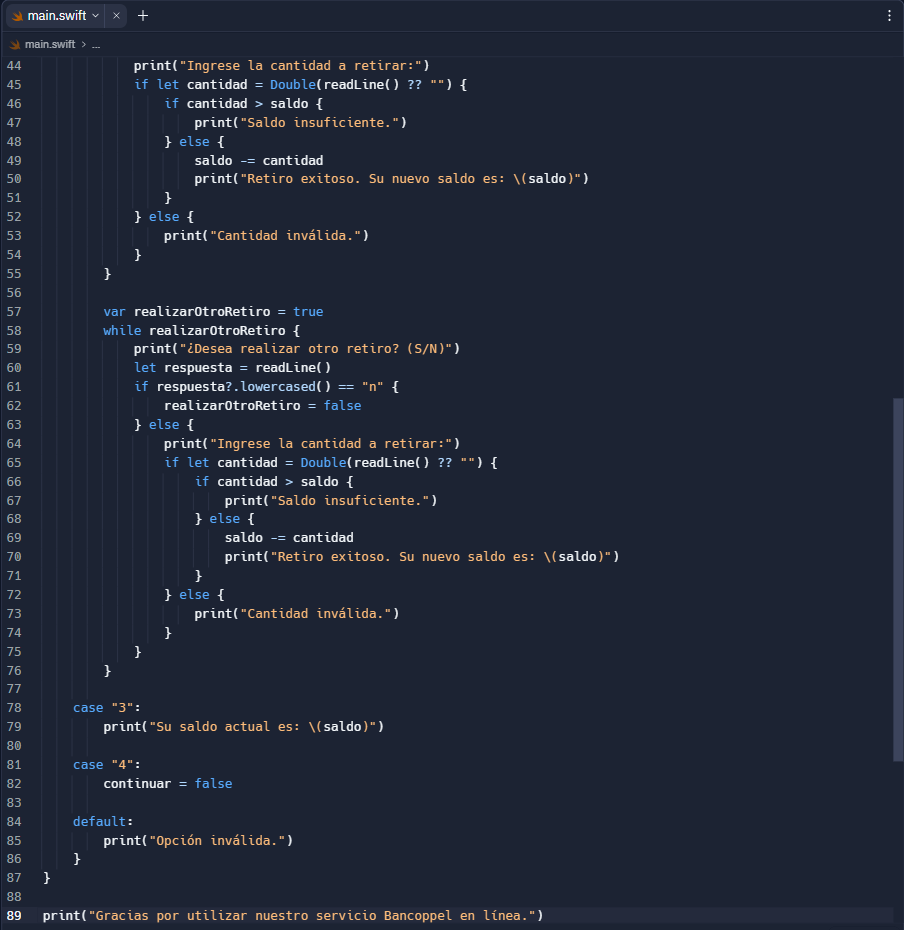
Xcode proporciona una interfaz gráfica de usuario (GUI) para crear aplicaciones. Esto incluye herramientas para la edición de código, la depuración, la creación de interfaces de usuario, el control de versiones y la gestión de proyectos. También incluye una extensa biblioteca de clases y marcos para simplificar el desarrollo de aplicaciones.

Además, Xcode incluye un simulador de iOS que permite a los desarrolladores probar sus aplicaciones en una variedad de dispositivos móviles sin tener que poseer físicamente cada dispositivo móvil. También admite la integración de herramientas y complementos de terceros, lo que permite a los desarrolladores personalizar y ampliar aún más la funcionalidad del IDE.

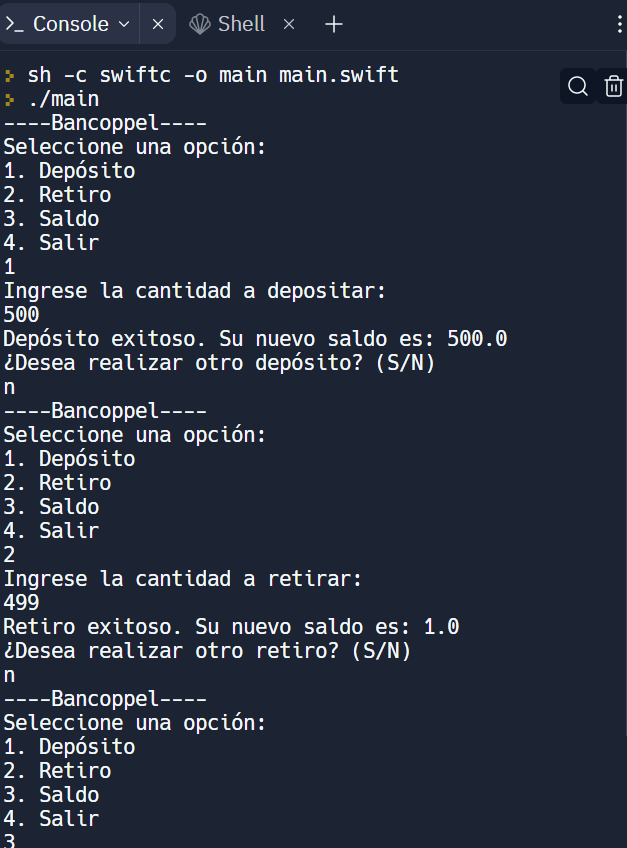
En resumen, Xcode es un IDE completo y potente que se utiliza para el desarrollo de aplicaciones para los sistemas operativos de Apple, que proporciona diversas herramientas y funciones que facilitan el proceso de desarrollo y mejoran la calidad de las aplicaciones creadas.

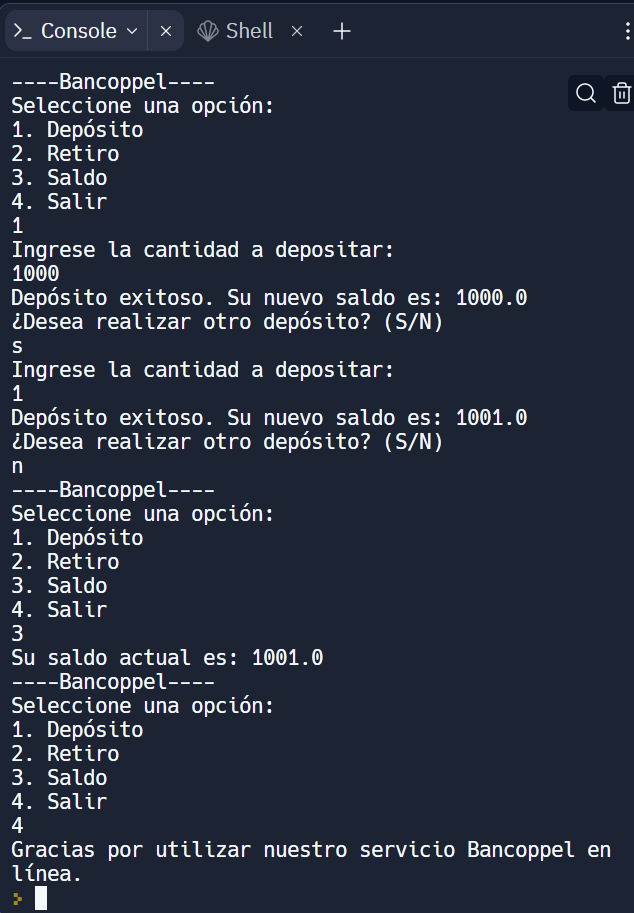
# *Codificación*





# *Prueba del programa*





# *Conclusion*

En conclusión, el desarrollo de un programa en lenguaje Swift para realizar funciones bancarias básicas como depósitos y retiros demuestra la capacidad de este lenguaje de programación para crear aplicaciones cotidianas que son muy útiles para los usuarios. La facilidad de lectura y escritura de código, la velocidad de ejecución y la capacidad de integración con otras herramientas como Xcode hacen de Swift una opción atractiva para desarrollar aplicaciones móviles y de escritorio. Además, la inclusión de funciones de seguridad de la información y protección de datos en el diseño del programa es necesaria para garantizar la privacidad y confidencialidad de los usuarios en la gestión de asuntos financieros.

# *Referencias*

School, T. (2022, 29 julio). *Lenguaje de programación Swift: ventajas y desventajas*. Tokio School. <https://www.tokioschool.com/noticias/lenguaje-programacion-swift/>

Data, S. B. (2021, 21 marzo). *Swift: Lo bueno y lo malo*. sitiobigdata.com. <https://sitiobigdata.com/2019/01/31/lenguaje-swift-bueno-y-malo/>

Cristancho, F. (2022, 6 abril). *Características del lenguaje de programación Swift - Talently*. Talently Blog. <https://talently.tech/blog/caracteristicas-del-lenguaje-de-programacion-swift/>

Urrutia, D. (2023, 3 enero). *Qué es Xcode | Definición, origen y características*. Arimetrics. <https://www.arimetrics.com/glosario-digital/xcode>

Urrutia, D. (2023, 3 enero). *Qué es Xcode | Definición, origen y características*. Arimetrics. <https://www.arimetrics.com/glosario-digital/xcode>

colaboradores de Wikipedia. (2022, 30 mayo). *Xcode*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Xcode>