

Actividad [#1] – [nombre act] [“materia”]

Ingeniería en Desarrollo de Software

**Tutor: Marco Alonso Rodríguez Tapia**

**Alumno: Manuel Enrique Ramirez Lopez**

**Fecha: 14/03/2023**

***Indice***

*[Introduccion](#_Toc6523)* [1](#_Toc6523)

*[Descripción](#_Toc317)* [2](#_Toc317)

*[Justificación](#_Toc20408)* [3](#_Toc20408)

*[Investigación](#_Toc1904)* [4](#_Toc1904)

*[Codificación](#_Toc6525)* [6](#_Toc6525)

*[Prueba del programa](#_Toc4006)* [8](#_Toc4006)

*[Conclusion](#_Toc24684)* [9](#_Toc24684)

*[Referencias](#_Toc27595)* [10](#_Toc27595)

# *Introduccion*

Programar Swift para cajeros automáticos implica crear un sistema que permita a los usuarios realizar transacciones financieras de forma segura y eficiente. Este tipo de sistema debe poder aceptar depósitos, realizar retiros, proporcionar información de cuenta y procesar transacciones con tarjetas de crédito y débito.

Lograr esto requiere una combinación de habilidades de software y conocimiento de los sistemas financieros. La seguridad es fundamental para este tipo de sistemas, por lo que es importante que los desarrolladores tengan un conocimiento sólido de la criptografía y la seguridad de la red.

Además, se debe prestar atención a la facilidad de uso y accesibilidad del sistema, ya que los usuarios esperan una experiencia de transacción rápida y sin esfuerzo. La interfaz de usuario debe ser intuitiva y fácil de usar, y el sistema debe estar diseñado para manejar múltiples solicitudes de transacciones simultáneamente.

En conclusión, la programación de Swift para cajeros automáticos es una tarea compleja que requiere una combinación de habilidades técnicas y conocimientos financieros. Establecer un sistema de cajeros automáticos eficiente y seguro es fundamental para garantizar la satisfacción del usuario y la confianza en el sistema financiero.

# *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:** En la actividad anterior creaste la interfaz del menú principal del programa de banca en línea. Además, programaste las opciones *“Depósito” y “Retiro”*. Ahora deberás terminar de darle funcionalidad a las opciones faltantes: *“Saldo” y “Salir”.*

En caso de que el usuario ingrese la opción Saldo, el programa deberá capturar por teclado los siguientes datos: ● El saldo del usuario ● Preguntar si desea realizar otra operación.

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

# *Justificación*

La creación de un sistema de cajero automático en Swift es una actividad importante por varios motivos:

- Ayuda a desarrollar habilidades de programación: la creación de un sistema de cajero automático en Swift implica la aplicación práctica de habilidades de programación y resolución de problemas complejos, lo que puede ayudar a mejorar las habilidades de los programadores y las habilidades de resolución de problemas. capacidad

-Ofrece la oportunidad de aprender sobre los sistemas financieros: la creación de un sistema de cajero automático implica el manejo de datos financieros confidenciales, lo que brinda la oportunidad de aprender sobre los sistemas financieros, el cifrado y la seguridad de la red.

-Facilita el desarrollo de tecnología segura y accesible: un sistema de cajero automático seguro y accesible puede ayudar a aumentar la confianza en el sistema financiero y mejorar la experiencia del usuario. Al desarrollar un sistema que cumpla con estos requisitos, puede promover una tecnología más segura y accesible.

-Tiene aplicaciones prácticas del mundo real: la creación de un sistema de cajero automático en Swift puede tener aplicaciones prácticas del mundo real, ya que los cajeros automáticos son una parte integral del sistema financiero y son utilizados por millones de personas en todo el mundo.

En conclusión, construir un sistema de cajero automático en Swift es una actividad importante que ayuda a desarrollar habilidades de programación, brinda la oportunidad de aprender sobre los sistemas financieros y contribuye al desarrollo de una tecnología más segura y accesible. Además, tiene aplicaciones prácticas del mundo real, lo que lo convierte en un esfuerzo que vale la pena para cualquier desarrollador que quiera mejorar sus habilidades y hacer una contribución positiva al mundo.

# *Investigación*

Importancia del lenguaje Swift en la industria

Swift es un lenguaje de programación de alto nivel desarrollado por Apple para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS. Desde su lanzamiento en 201

, Swift se ha convertido en un lenguaje de programación muy popular y ha tenido un gran impacto en la industria de la tecnología por las siguientes razones:

Facilidad de uso: Swift está diseñado para ser fácil de aprender y usar, lo que lo hace atractivo para los desarrolladores. . . . en todos los niveles de habilidad. El lenguaje es conciso y fácil de leer, lo que reduce el tiempo y los costos de desarrollo.

Compatibilidad con Objective-C: Swift es compatible con Objective-C, lo que significa que los desarrolladores pueden usar el código existente de Objective-C en sus aplicaciones Swift. Esto permite una transición más fácil de Objective-C a Swift y permite a los desarrolladores usar las bibliotecas y herramientas de Objective-C existentes.

Mejora del rendimiento: Swift es un lenguaje de programación más rápido que Objective-C. Esto se debe en parte a que Swift usa un compilador optimizado y es un lenguaje más moderno y eficiente en términos de memoria.

Popularidad en la industria: Swift es un lenguaje de programación muy popular en la industria de la tecnología y es ampliamente utilizado por empresas como Uber, Airbnb, LinkedIn y muchas otras.

Desarrollo de aplicaciones multiplataforma: Swift se utiliza para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS, lo que permite a los desarrolladores crear aplicaciones multiplataforma de manera eficiente.

En conclusión, Swift es un lenguaje de programación importante en la industria de la tecnología debido a su facilidad de uso, interoperabilidad con Objective-C, rendimiento mejorado, popularidad industrial y la capacidad de desarrollar aplicaciones para múltiples plataformas. Como resultado, se espera que la demanda de desarrolladores de Swift siga creciendo en los próximos años y Swift seguirá siendo un lenguaje de programación importante en la industria.

## Aplicación del lenguaje Swift en la industria

Swift se usa ampliamente en la industria de la tecnología para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS. Algunas de las aplicaciones más populares creadas con Swift incluyen:

-Instagram, una popular aplicación de redes sociales utilizada para compartir fotos y videos, ha sido reconstruida en Swift para un mejor rendimiento y capacidad de respuesta.

-Lyft: la aplicación de transporte utiliza Swift para brindar una experiencia de usuario rápida y eficiente.

-LinkedIn: la red social empresarial utiliza Swift para mejorar la velocidad y el rendimiento de la aplicación.

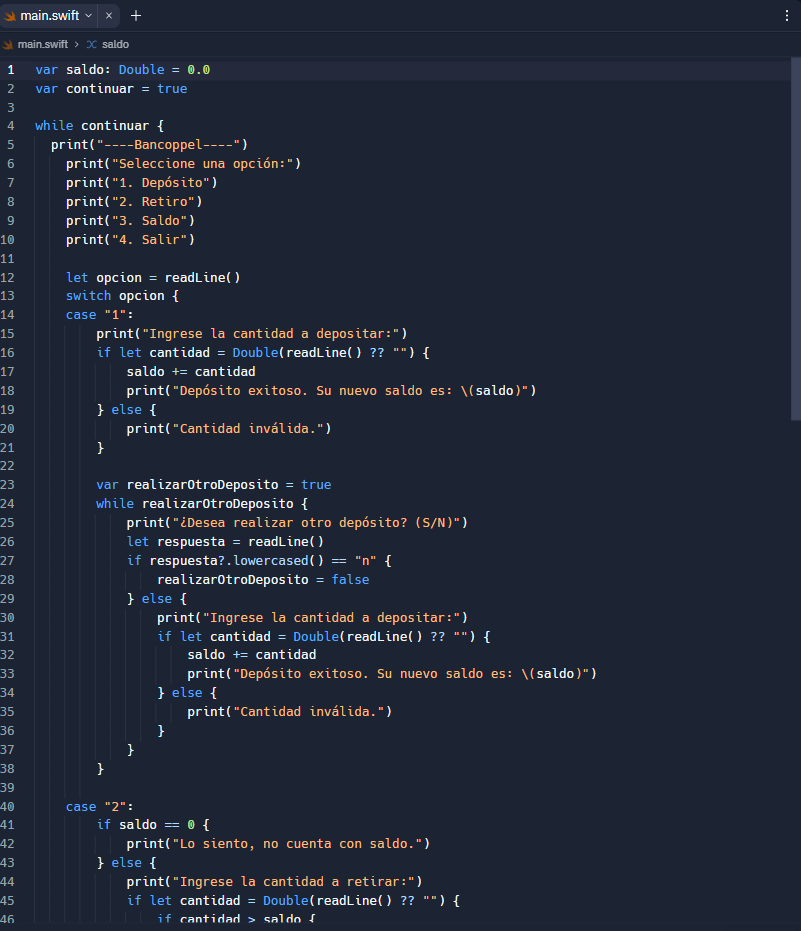
-Airbnb: la popular plataforma de alojamiento utiliza Swift para brindar una experiencia de usuario rápida y confiable.

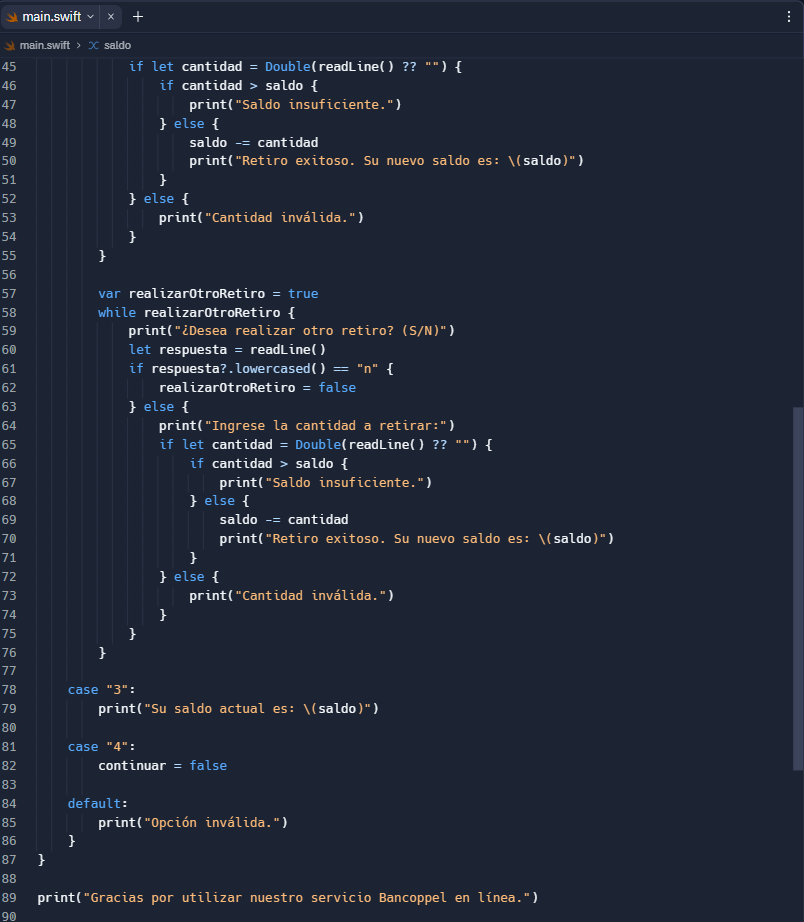
-Slack: la plataforma de comunicación empresarial utiliza Swift para mejorar el rendimiento de la aplicación.

Además de estas aplicaciones, Swift también se utiliza en el desarrollo de juegos iOS, aplicaciones de realidad virtual y aumentada y programas de automatización de procesos comerciales.

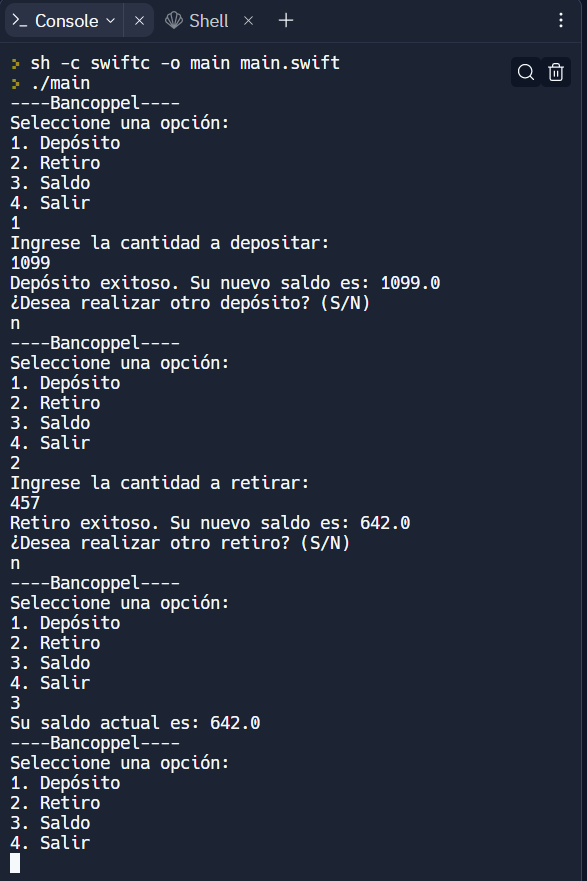
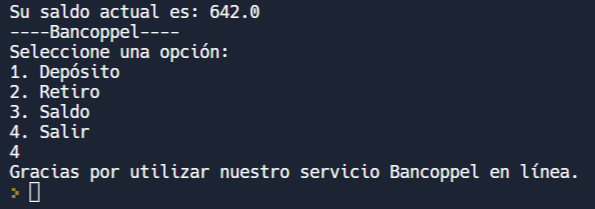
En la industria de la tecnología, Swift se ha convertido en un lenguaje de programación importante debido a su facilidad de uso, interoperabilidad con Objective-C, rendimiento mejorado y la capacidad de desarrollar aplicaciones para múltiples plataformas. Como resultado, los desarrolladores que dominan Swift tienen una demanda creciente en la industria y pueden trabajar en múltiples aplicaciones y proyectos en todas las industrias.

# *Codificación*





# *Prueba del programa*



# *Conclusion*

En conclusión, la programación de Swift para cajeros automáticos es una tarea compleja pero importante que requiere una combinación de habilidades técnicas y conocimientos financieros. Al crear un sistema de cajeros automáticos seguro y accesible, se puede mejorar la experiencia del usuario y aumentar la confianza en el sistema financiero, lo que puede tener un impacto positivo en la sociedad. Además, la creación de un sistema de cajero automático en Swift puede ayudar a desarrollar habilidades de programación, brindar la oportunidad de aprender sobre los sistemas financieros y contribuir al desarrollo de una tecnología más segura y accesible. Por lo tanto, es una actividad rentable para cualquier desarrollador que quiera mejorar sus habilidades y hacer una contribución positiva al mundo.

# *Referencias*

Team, K. (2022, 4 octubre). *Importancia del lenguaje Swift para el desarrollo de aplicaciones móviles*. KeepCoding Tech School. <https://keepcoding.io/blog/importancia-swift-desarrollo-de-aplicaciones/>

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

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

Apple. (s. f.). *Swift - Apple (MX)*. <https://www.apple.com/mx/swift/>

Team, K. (2022b, octubre 4). *Importancia del lenguaje Swift para el desarrollo de aplicaciones móviles*. KeepCoding Tech School. <https://keepcoding.io/blog/importancia-swift-desarrollo-de-aplicaciones/>

Tena, V. (2020, 24 febrero). *El lenguaje de programación Swift de Apple triunfa en Silicon Valley*. Business Insider España. <https://www.businessinsider.es/lenguaje-programacion-swift-apple-triunfa-silicon-valley-579491>