



Actividad | 1 | Instalación XCode/Aplicación 1

Desarrollo de Aplicaciones Móviles III

Ingeniería en Desarrollo de Software



academiaglobal

TUTOR: Sandra Luz Lara Dévora

ALUMNO: Carlos Fco Estrada Salazar

FECHA: 31/Marzo/2025

INDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
Codificación	6
Prueba de aplicación	9
CONCLUSIÓN	11
REFERENCIAS	12

GitHub Link:

INTRODUCCIÓN

El desarrollo de aplicaciones móviles ha tomado un papel fundamental en el mundo digital actual. Con la creciente demanda de aplicaciones funcionales y seguras, es crucial dominar lenguajes de programación eficientes como Swift. Este lenguaje, desarrollado por Apple, ofrece una sintaxis moderna, mayor seguridad y un rendimiento óptimo, lo que lo convierte en una excelente opción para la creación de aplicaciones en el ecosistema de iOS.

En esta actividad, nos enfocaremos en la instalación de XCode, el entorno de desarrollo integrado (IDE) utilizado para programar en Swift y desarrollar aplicaciones para dispositivos Apple. Sin embargo, en caso de preferir una alternativa, también es posible emplear compiladores en línea, como Replit, para ejecutar y probar el código sin necesidad de instalar software adicional en el equipo. Esto es particularmente útil para aquellos que buscan flexibilidad en el desarrollo y acceso inmediato a un entorno funcional.

Como parte del ejercicio práctico, desarrollaremos una aplicación sencilla que permita ingresar un número y determinar si es par o impar. Este tipo de programas son fundamentales en la programación ya que permiten aplicar conceptos básicos de estructuras de control, lógica condicional y manejo de entrada y salida de datos. Además, esta actividad nos ayudará a familiarizarnos con el entorno de desarrollo y con la sintaxis de Swift, estableciendo las bases para futuros proyectos más complejos en el desarrollo de aplicaciones móviles.

A través de este ejercicio, se refuerza la importancia del pensamiento lógico y la resolución de problemas en la programación. La comprensión de cómo determinar si un número es par o impar puede parecer simple, pero representa un concepto clave en la programación que se aplica en numerosos escenarios del desarrollo de software. Al completar esta actividad, los desarrolladores principiantes ganarán confianza en su habilidad para escribir y ejecutar código en Swift, allanando el camino para proyectos más avanzados en el futuro.

DESCRIPCIÓN

El contexto de esta actividad gira en torno a la importancia del lenguaje de programación Swift en el desarrollo de aplicaciones móviles y el uso de herramientas adecuadas para su implementación. Swift es un lenguaje moderno que ofrece diversas ventajas, como seguridad mejorada, facilidad de aprendizaje y eficiencia en la ejecución del código. Para trabajar con este lenguaje, es fundamental contar con un entorno de desarrollo adecuado, siendo XCode la opción oficial proporcionada por Apple. Sin embargo, también existen alternativas en línea, como Replit, que permiten la ejecución de código Swift sin necesidad de instalar software adicional.

El objetivo de la actividad es desarrollar una aplicación sencilla que determine si un número ingresado por el usuario es par o impar. Para lograr esto, se utilizará una estructura de control condicional que evalúe si el residuo de la división del número entre dos es cero. Este tipo de ejercicios son esenciales para fortalecer la comprensión de conceptos básicos de programación, como el uso de operadores matemáticos y estructuras de decisión.

La elección de Replit como entorno de desarrollo permite un acceso más rápido y sencillo para la ejecución de la aplicación, evitando las posibles complicaciones derivadas de la instalación y configuración de XCode. Esto resulta especialmente beneficioso para aquellos que buscan una solución práctica y eficiente para desarrollar y probar su código en Swift.

Esta actividad no solo permite reforzar los conocimientos en Swift, sino que también introduce el uso de herramientas adecuadas para su programación. La aplicación desarrollada es un ejemplo práctico de cómo utilizar estructuras de control en Swift para resolver problemas comunes en la programación, estableciendo una base sólida para el aprendizaje y la práctica en el desarrollo de aplicaciones móviles.

JUSTIFICACIÓN

La solución planteada en esta actividad, que consiste en el desarrollo de una aplicación en Swift para determinar si un número es par o impar, es una excelente estrategia para reforzar el aprendizaje de conceptos básicos de programación y estructuras de control. Swift es un lenguaje ampliamente utilizado en el desarrollo de aplicaciones para dispositivos Apple, por lo que su dominio es fundamental para cualquier programador que desee incursionar en este ecosistema.

El uso de Replit como entorno de desarrollo es una elección acertada debido a su facilidad de acceso y configuración. A diferencia de XCode, que requiere una instalación en dispositivos Mac y puede tener una curva de aprendizaje más pronunciada, Replit permite ejecutar y probar código de manera inmediata sin necesidad de configuraciones complejas. Esto es particularmente útil para estudiantes y desarrolladores principiantes que buscan enfocarse en el aprendizaje de la sintaxis y la lógica del lenguaje sin preocuparse por detalles técnicos de la configuración del entorno.

El desarrollo de una aplicación sencilla como la propuesta permite aplicar conceptos fundamentales de la programación estructurada, como el uso de condicionales y operadores matemáticos. Estas habilidades son esenciales en el desarrollo de software, ya que constituyen la base para la creación de algoritmos más complejos en el futuro.

La actividad fomenta la resolución de problemas mediante la implementación de soluciones prácticas en un lenguaje de programación moderno. Aprender a escribir y ejecutar código en Swift desde el inicio permite a los estudiantes adquirir confianza en sus habilidades de programación y comprender mejor los principios del desarrollo de software.

La implementación de esta solución no solo facilita el aprendizaje de Swift, sino que también proporciona una base sólida para futuras aplicaciones móviles. La combinación de una tarea sencilla, pero significativa, con una herramienta accesible como Replit, permite maximizar el aprendizaje y la práctica efectiva en el desarrollo de aplicaciones móviles.

DESARROLLO

Codificación

import Foundation

Importa la biblioteca *Foundation*, que proporciona funcionalidades básicas como entrada/salida y manipulación de cadenas y números.

// Función para determinar si un número es par o impar

func verificarParImpar() {

Declara una función llamada *verificarParImpar()*, que contendrá la lógica para verificar si un número es par o impar.

print("Ingresa un número:", terminator: " ")

Imprime el mensaje *"Ingresa un número:"* en la consola y usa *terminator: " "* para que la entrada del usuario aparezca en la misma línea.

if let input = readLine(), let numero = Int(input) {

readLine() captura la entrada del usuario como una cadena (*String*).

if let input = readLine() verifica si se ha recibido un valor válido.

let numero = Int(input) intenta convertir la entrada de *String* a *Int*.

Si la conversión es exitosa, el número se almacena en la variable *numero*.

if numero % 2 == 0 {

Comprueba si el número es par.

numero % 2 == 0 significa que el residuo de la división entre 2 es 0, lo que indica que el número es par.

print("El número \(numero) es par.")

Usa interpolación de cadenas (*\(numero)*) para incluir el valor del número en el mensaje.

```

} else {
    print("El número \((numero) es impar.")
}

```

Si el número no es par (*else*), entonces es impar y se muestra el mensaje correspondiente.

```

} else {
    print("Entrada no válida. Por favor, ingresa un número entero.")
}

```

Si la conversión de *input* a *Int* falla, significa que el usuario ingresó un valor no numérico entero.

En este caso, se muestra un mensaje de error.

```

// Ejecutar la función
verificarParImpar()

```

Llama a la función *verificarParImpar()* para ejecutar el programa.

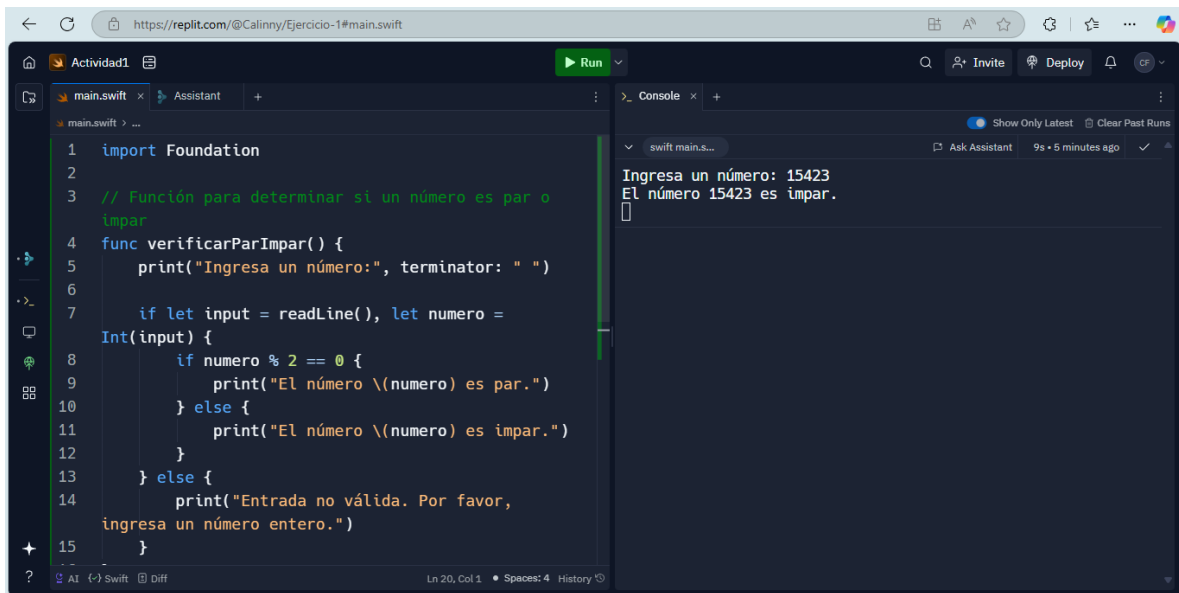
Código completo:

```
import Foundation

// Función para determinar si un número es par o impar
func verificarParImpar() {
    print("Ingresa un número:", terminator: " ")
    if let input = readLine(), let numero = Int(input) {
        if numero % 2 == 0 {
            print("El número \(numero) es par.")
        } else {
            print("El número \(numero) es impar.")
        }
    } else {
        print("Entrada no válida. Por favor, ingresa un número entero.")
    }
}

// Ejecutar la función
verificarParImpar()
```


Prueba de la aplicación



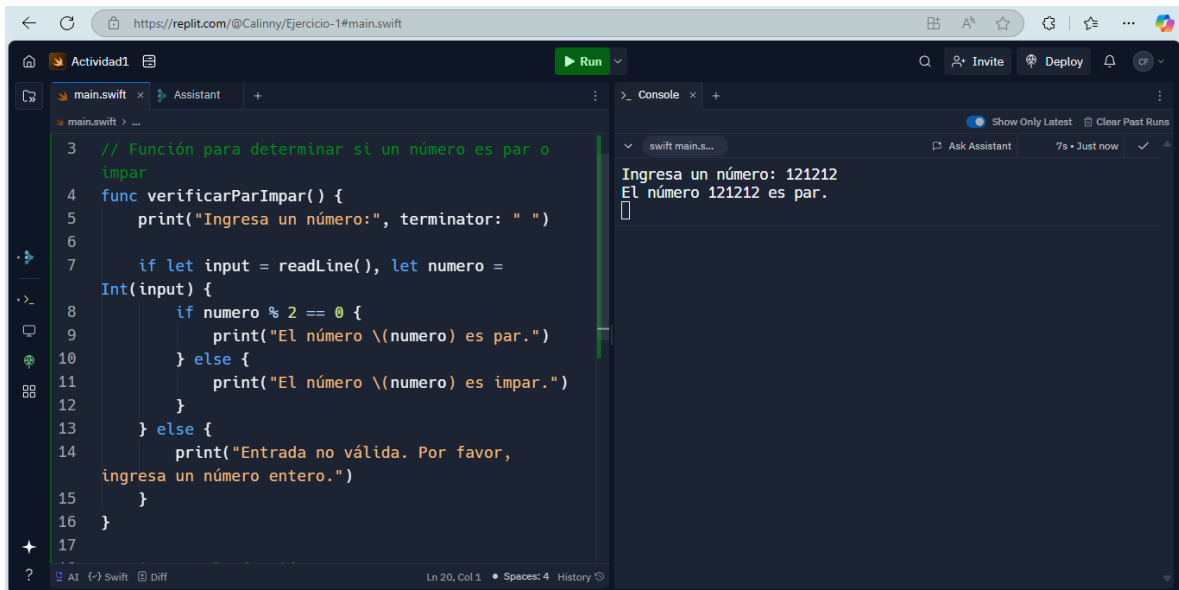
The screenshot shows a Swift REPL interface with a file named 'main.swift'. The code defines a function 'verificarParImpar()' that prompts the user to enter a number. It then checks if the number is even or odd using a modulo operation. The console output shows the user entering '15423' and the program responding that it is an odd number.

```
1 import Foundation
2
3 // Función para determinar si un número es par o
  impar
4 func verificarParImpar() {
5     print("Ingresa un número:", terminator: " ")
6
7     if let input = readLine(), let numero =
      Int(input) {
8         if numero % 2 == 0 {
9             print("El número \(numero) es par.")
10        } else {
11            print("El número \(numero) es impar.")
12        }
13    } else {
14        print("Entrada no válida. Por favor,
      ingresa un número entero.")
15    }
16 }
```

Console output:

```
swift main.s...
Ingresa un número: 15423
El número 15423 es impar.
```

Comprobando un número IMPAR



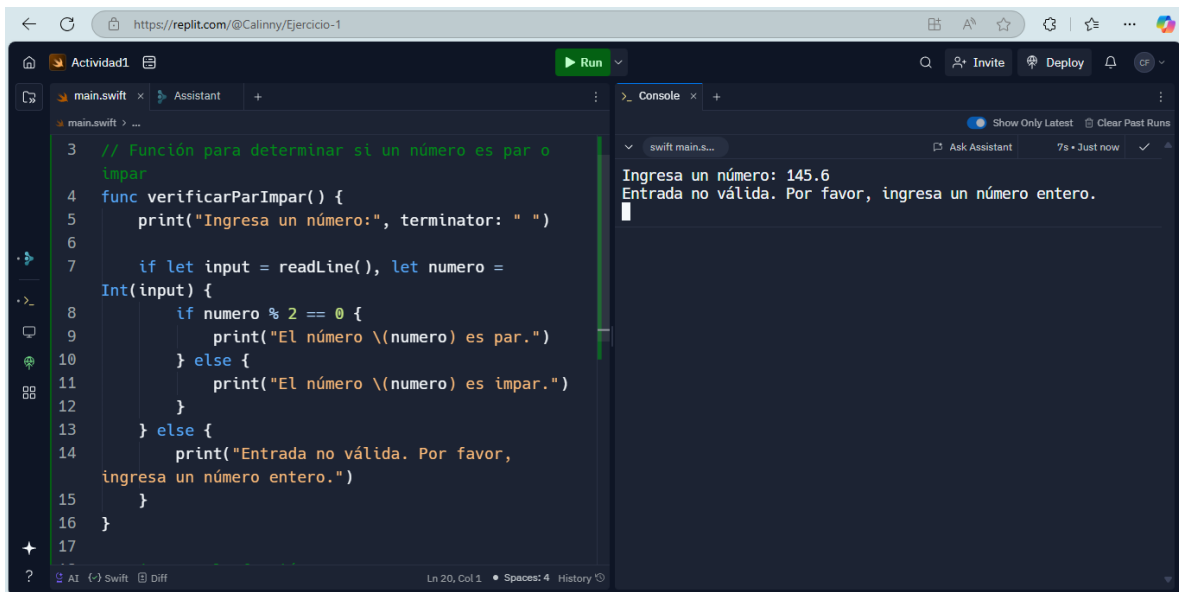
The screenshot shows the same Swift REPL interface as the previous one, but with the user entering '121212'. The program correctly identifies it as an even number.

```
3 // Función para determinar si un número es par o
  impar
4 func verificarParImpar() {
5     print("Ingresa un número:", terminator: " ")
6
7     if let input = readLine(), let numero =
      Int(input) {
8         if numero % 2 == 0 {
9             print("El número \(numero) es par.")
10        } else {
11            print("El número \(numero) es impar.")
12        }
13    } else {
14        print("Entrada no válida. Por favor,
      ingresa un número entero.")
15    }
16 }
```

Console output:

```
swift main.s...
Ingresa un número: 121212
El número 121212 es par.
```

Comprobando un número PAR



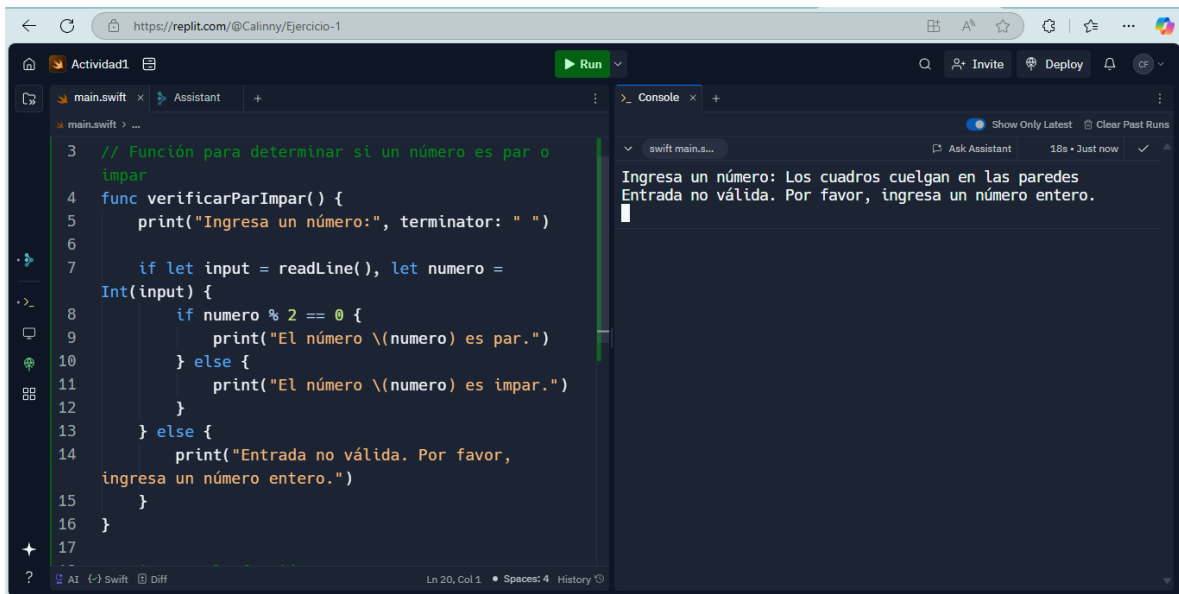
The screenshot shows a Swift REPL interface with a code editor on the left and a console on the right. The code in the editor defines a function `verificarParImpar()` that prompts the user for a number and checks if it is even or odd. The console shows the program's output after the user entered `145.6`.

```
3 // Función para determinar si un número es par o
4 impar
5 func verificarParImpar() {
6     print("Ingresa un número:", terminator: " ")
7
8     if let input = readLine(), let numero =
9     Int(input) {
10         if numero % 2 == 0 {
11             print("El número \(numero) es par.")
12         } else {
13             print("El número \(numero) es impar.")
14         }
15     } else {
16         print("Entrada no válida. Por favor,
17         ingresa un número entero.")
18     }
19 }
```

Console output:

```
Ingresa un número: 145.6
Entrada no válida. Por favor, ingresa un número entero.
```

Prueba de errores, ingresando un numero DECIMAL



The screenshot shows the same Swift REPL interface as the previous one. The code is identical. The console shows the program's output after the user entered a string of text.

```
3 // Función para determinar si un número es par o
4 impar
5 func verificarParImpar() {
6     print("Ingresa un número:", terminator: " ")
7
8     if let input = readLine(), let numero =
9     Int(input) {
10         if numero % 2 == 0 {
11             print("El número \(numero) es par.")
12         } else {
13             print("El número \(numero) es impar.")
14         }
15     } else {
16         print("Entrada no válida. Por favor,
17         ingresa un número entero.")
18     }
19 }
```

Console output:

```
Ingresa un número: Los cuadros cuelgan en las paredes
Entrada no válida. Por favor, ingresa un número entero.
```

Prueba de errores, ingresando una CADENA DE TEXTO

CONCLUSIÓN

La realización de esta actividad ha permitido reforzar conocimientos fundamentales en el desarrollo de aplicaciones móviles utilizando Swift. Este lenguaje, diseñado por Apple, no solo es potente y eficiente, sino que también proporciona una sintaxis intuitiva y segura para los desarrolladores. A través de la implementación de un programa que determina si un número es par o impar, se ha logrado comprender la estructura básica del lenguaje y la forma en que se manejan las entradas de usuario, las conversiones de datos y la toma de decisiones condicionales.

En el ámbito laboral, este tipo de ejercicios resulta sumamente valioso, ya que representa la base para desarrollar aplicaciones más complejas en iOS. La lógica de programación aplicada en esta actividad es la misma que se usa en escenarios más avanzados, como la validación de datos en formularios, la implementación de algoritmos de búsqueda y filtrado, o la toma de decisiones dentro de un sistema automatizado. Por ello, dominar estos conceptos es crucial para cualquier desarrollador de aplicaciones móviles.

Desde una perspectiva cotidiana, la capacidad de programar soluciones automatizadas para problemas comunes es una ventaja significativa. Aplicaciones como calculadoras, conversores de unidades y gestores de tareas utilizan estructuras similares a la de este ejercicio. Comprender estos principios permite no solo consumir tecnología, sino también crear herramientas útiles para mejorar la eficiencia en diversas tareas diarias.

Esta actividad no solo ha facilitado el aprendizaje de Swift, sino que también ha brindado una visión clara de cómo la lógica de programación impacta tanto en el ámbito profesional como en la vida cotidiana, fortaleciendo habilidades esenciales para el desarrollo de software.

REFERENCIAS

- MoureDev by Brais Moure. (2022, 14 enero). *¿CÓMO ser APPLE DEVELOPER? GUÍA de ESTUDIO Desde Cero [2022]* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=-w3R4DEwd2g>
- José Dimas Luján Castillo. (2023, 27 enero). *Clase 1 - Presentación del curso* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Hxmc9L30-3g>