



**Actividad [#1] - [Instalación
XCode/Aplicación 1]
[Desarrollo de Aplicaciones Móviles III]
Ingeniería en Desarrollo de Software**

Tutor: Sandra Luz Lara Dévora

Alumno: Alan David López Rojas

Fecha: 30/06/2023

Índice

Introducción.....	pág. 3
Descripción.....	pág. 4
Justificación.....	pág. 5
Desarrollo.....	pág. 6
Codificación.....	pág. 6
Prueba de aplicación.....	pág. 7
Conclusión.....	pág. 9
Bibliografía.....	pág. 10

Introducción

En el mundo de la programación, el lenguaje Swift ha ganado popularidad debido a su modernidad, seguridad y su enfoque intuitivo. Con Swift, los desarrolladores tienen la capacidad de crear una amplia variedad de aplicaciones. En esta ocasión, nos enfrentamos a la tarea de desarrollar una aplicación que permita al usuario ingresar un número y determinar si es par o impar.

Antes de realizar el desarrollo de la aplicación, es importante contar con las herramientas adecuadas. Una de las opciones más comunes y poderosas es Xcode, el entorno de desarrollo integrado (IDE) de Apple. Xcode proporciona un conjunto de herramientas completo para el desarrollo de aplicaciones iOS, macOS, watchOS y tvOS. Siendo la opción preferida por muchos desarrolladores, la instalación de Xcode es esencial para seguir adelante con esta actividad.

La instalación de Xcode se puede realizar siguiendo los pasos proporcionados por Apple en su sitio web oficial. Una vez instalado, se obtiene acceso a un amplio conjunto de características y funcionalidades que facilitan el desarrollo de aplicaciones en Swift.

Sin embargo, también existe la opción de utilizar compiladores online que brindan un entorno de desarrollo Swift en la nube. Estos compiladores permiten escribir, compilar y ejecutar código Swift sin la necesidad de instalar ninguna herramienta en local. Esto puede ser una alternativa conveniente para aquellos que deseen evitar la instalación de software adicional en sus dispositivos.

Descripción

En esta actividad, se nos presenta la tarea de crear una aplicación utilizando el lenguaje de programación Swift. La aplicación en cuestión debe permitir al usuario ingresar un número y determinar si es par o impar.

Swift es un lenguaje moderno y seguro que ha ganado popularidad en el desarrollo de aplicaciones para dispositivos Apple. Su sintaxis intuitiva facilita la escritura de código claro y legible. En este caso, se deben aprovechar las capacidades de Swift para diseñar una interfaz de usuario sencilla que permita al usuario introducir un número y recibir una respuesta visual que indique si es par o impar.

Para llevar a cabo esta actividad, se mencionan dos opciones: la instalación de Xcode, el entorno de desarrollo integrado proporcionado por Apple, o el uso de compiladores online que ofrecen un entorno Swift en la nube. Ambas opciones tienen sus ventajas, y la elección depende de las preferencias y las necesidades del desarrollador.

Una vez que tenemos el entorno de desarrollo preparado, podemos comenzar a crear la aplicación. La tarea principal es diseñar una interfaz de usuario amigable que permita al usuario ingresar un número y, mediante la implementación de una lógica de programación adecuada, determinar si es par o impar.

La aplicación debe ser capaz de analizar el número ingresado por el usuario y realizar una operación matemática para verificar si es divisible por 2. Si el resultado es un número entero, entonces es par; de lo contrario, es impar. El resultado debe ser comunicado de manera clara al usuario, utilizando elementos visuales o mensajes de texto.

Justificación

La elección de emplear una solución basada en el lenguaje de programación Swift para la actividad presentada, es una opción justificada por varias razones.

En primer lugar, Swift es un lenguaje diseñado por Apple para el desarrollo de aplicaciones en sus dispositivos. Al utilizar Swift, nos aseguramos de estar trabajando con un lenguaje de programación sólido y confiable, respaldado por una gran comunidad de desarrolladores y con el soporte activo de Apple.

Además, Swift ofrece una sintaxis intuitiva y legible, lo que facilita la comprensión y el mantenimiento del código. Esto es especialmente importante en aplicaciones sencillas como la que se requiere en esta actividad, donde se busca determinar si un número es par o impar. La claridad en el código facilita la identificación de errores y agiliza el proceso de desarrollo.

Otro aspecto a tener en cuenta es la versatilidad de Swift. Si bien la actividad plantea la creación de una aplicación simple, Swift nos proporciona las herramientas y características necesarias para futuras expansiones y mejoras. En caso de que se desee agregar más funcionalidades a la aplicación en el futuro, como por ejemplo permitir al usuario realizar operaciones matemáticas básicas, Swift nos brinda la flexibilidad para implementar esas funcionalidades de manera eficiente y escalable.

Además, al emplear Swift, se aprovechan las ventajas del ecosistema de desarrollo de Apple. Swift está estrechamente integrado con las tecnologías y frameworks de Apple, lo que facilita la creación de aplicaciones nativas para dispositivos iOS, macOS, watchOS y tvOS. Esto implica que, si se desea expandir la aplicación en un futuro, se podrá aprovechar la amplia gama de herramientas y recursos disponibles en el ecosistema de Apple.

Desarrollo

❖ Codificación

```
import Foundation

func numEnt (numero: Int) -> String {
    if numero % 2 == 0 {
        return "El número es par."
    } else {
        return "El número es impar."
    }
}

print("Ingrese un número:")
if let input = readLine(), let numero = Int(input) {
    let resultado = numEnt(numero: numero)
    print(resultado)
} else {
    print("Entrada inválida. Asegúrese de ingresar un número entero válido.")
}
```

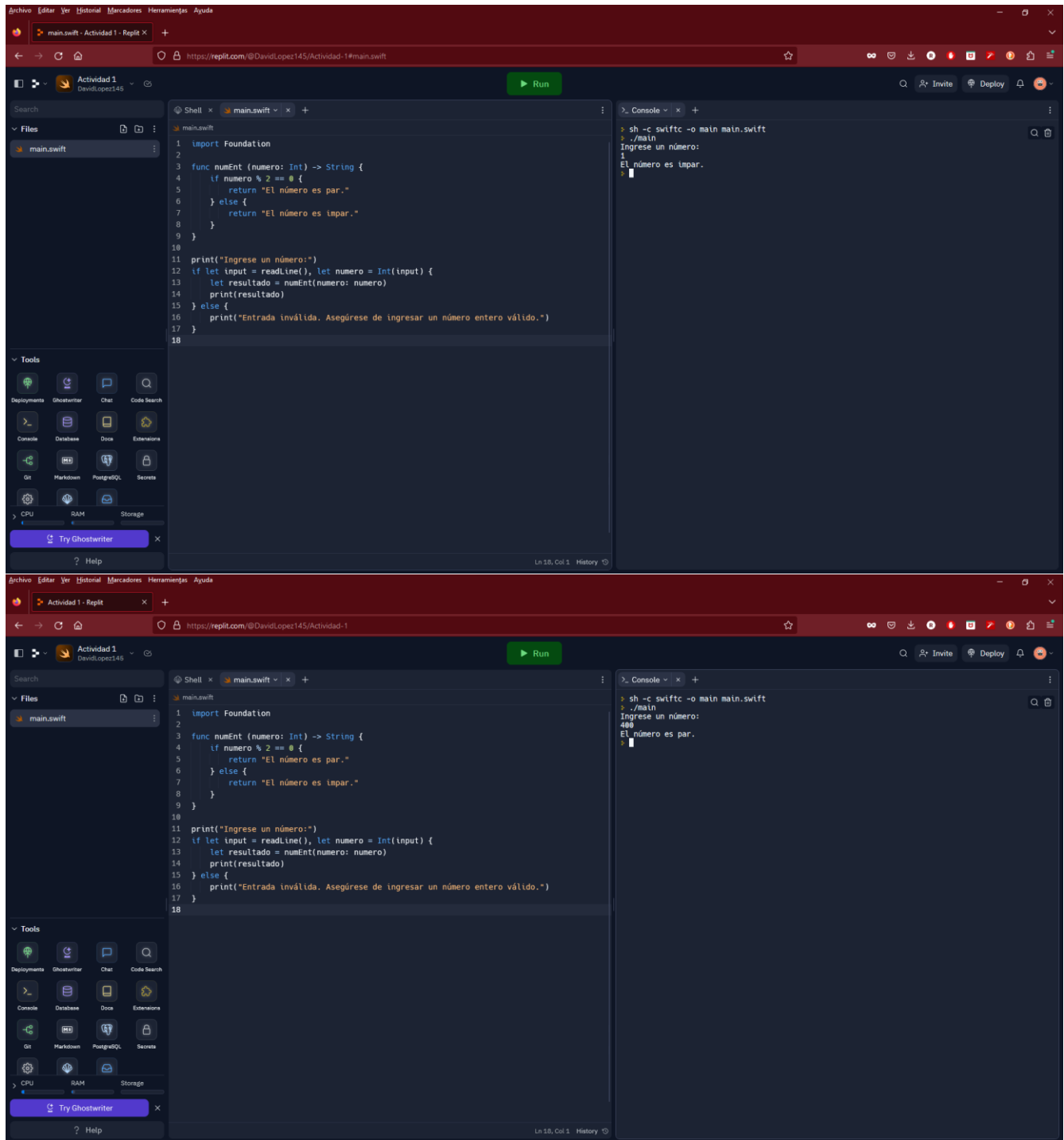
Este código en Swift define una función llamada " numEnt " que toma un número entero como argumento y devuelve una cadena que indica si el número es par o impar.

La función verifica si el número proporcionado es divisible por 2 utilizando el operador de módulo (%). Si el resultado de la división es igual a 0, significa que el número es par y se devuelve la cadena "El número es par." De lo contrario, si el resultado es distinto de 0, se considera que el número es impar y se devuelve la cadena "El número es impar."

Luego, el código solicita al usuario que ingrese un número mediante la función "readLine()", que permite leer una línea de entrada desde la consola. Si el usuario ingresa un número válido, se convierte en un entero utilizando "Int(input)" y se llama a la función " numEnt " para obtener el resultado. Finalmente, se imprime el resultado en la consola.

Si el usuario no ingresa un número válido, se imprimirá el mensaje "Entrada inválida. Asegúrese de ingresar un número entero válido." Esto sucede cuando el usuario ingresa caracteres no numéricos o no ingresa nada en absoluto.

❖ Prueba de la aplicación



The image displays two screenshots of a Replit environment, showing a Swift application being tested. The top screenshot shows the code editor with the following Swift code:

```
1 import Foundation
2
3 func numEnt (numero: Int) -> String {
4     if numero % 2 == 0 {
5         return "El número es par."
6     } else {
7         return "El número es impar."
8     }
9 }
10
11 print("Ingrese un número:")
12 if let input = readLine(), let numero = Int(input) {
13     let resultado = numEnt(numero: numero)
14     print(resultado)
15 } else {
16     print("Entrada inválida. Asegúrese de ingresar un número entero válido.")
17 }
18
```

The bottom screenshot shows the same code editor, but the console output is visible on the right side, indicating the program has been executed. The output shows the prompt "Ingrese un número:" followed by the input "499" and the result "El número es par.".

Link programa en replit

<https://replit.com/join/rjdipngnen-davidlopez145>

https://drive.google.com/file/d/1cPCoV8866NNvOZiS3fr3T0NKQdJ-TVqD/view?usp=drive_link

<https://github.com/AlanDavidLR/Desarrollomovil3.git>

Conclusión

En conclusión, la actividad de crear una aplicación para determinar si un número es par o impar utilizando Swift tiene una importancia relevante tanto en el campo laboral como en la vida cotidiana.

En el ámbito laboral, la capacidad de desarrollar aplicaciones utilizando Swift es altamente valorada en la industria de la tecnología, especialmente en el desarrollo de aplicaciones para dispositivos Apple.

Swift es un lenguaje de programación seguro que permite a los desarrolladores crear aplicaciones eficientes y robustas. Dominar Swift y poder desarrollar aplicaciones sencillas como la mencionada en la actividad demuestra habilidades técnicas y competencias en el desarrollo de software.

Además, esta actividad es una oportunidad para adquirir experiencia práctica en la creación de interfaces de usuario simples pero funcionales. La capacidad de diseñar interfaces amigables y fáciles de usar es un aspecto esencial en el desarrollo de aplicaciones exitosas. Esta habilidad es transferible a muchos otros campos laborales donde se requiera la creación de interfaces de usuario intuitivas.

En cuanto a la vida cotidiana, el conocimiento de Swift y la capacidad de desarrollar aplicaciones puede tener un impacto significativo. Las aplicaciones móviles son una parte integral de nuestras vidas, y poder crear aplicaciones personalizadas según nuestras necesidades o intereses puede brindarnos una mayor autonomía y satisfacción. Además, este tipo de habilidades puede abrir oportunidades para emprendedores o personas interesadas en crear sus propias aplicaciones para compartir con el mundo.

La importancia de lo realizado en la actividad de desarrollar una aplicación en Swift para determinar números pares o impares radica en la adquisición de habilidades técnicas valiosas, la capacidad de diseñar interfaces de usuario efectivas y la posibilidad de aplicar estos conocimientos tanto en el campo

laboral como en la vida cotidiana. El dominio de Swift y la capacidad de desarrollar aplicaciones son habilidades altamente demandadas en la industria y pueden brindar oportunidades profesionales y personales significativas.

Bibliografía

Stack Overflow Contributors, Aprendizaje Swift Language, 2019, Manual-informática, En línea: <https://manual-informatica.com/download-file.html>