

Actividad |1| Instalación XCode/Aplicación 1.

Desarrollo de Aplicaciones Móviles III.

Ingeniería en Desarrollo de Software.



TUTOR: Sandra Luz Lara Devora.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 29/07/2025.

Introducción.	3
Descripción.	3
Justificación.	4
Desarrollo.	4
Codificación.	5
Prueba de la aplicación.	7
Conclusion.	11
Referencias.	12

Introducción.

En esta primera actividad de la materia de Desarrollo de Aplicaciones Móviles III, donde tenemos que realizar la documentación de la creación de una aplicación que determine si un número ingresado por el usuario es par o impar utilizando la herramienta siguiente: swift que es un lenguaje de programación moderno, seguro e intuitivo, creado por Apple, que ha revolucionado la forma en que se desarrollan aplicaciones para sus plataformas. Su diseño limpio y sintaxis expresiva permiten a los desarrolladores escribir código de manera más rápida y con menos errores, lo que se traduce en aplicaciones más estables y eficientes. La versatilidad de Swift lo hace ideal para una amplia gama de proyectos, desde aplicaciones móviles para iOS y iPadOS hasta programas para macOS, watchOS, y tvOS. Sin embargo, para aprovechar su potencial al máximo, es fundamental comprender a fondo sus conceptos y estructuras básicas. Con un conocimiento sólido, es posible enfrentar cualquier desafío de programación.

Descripción.

En esta primera actividad de la materia de Desarrollo de Aplicaciones Móviles III, nos sumergimos en la creación de una aplicación básica utilizando Swift, el lenguaje de programación moderno y seguro de Apple. Swift, con su sintaxis limpia e intuitiva, se ha consolidado como la herramienta preferida para el desarrollo en plataformas como iOS, iPadOS, macOS y watchOS. Su diseño no solo facilita la escritura de código, sino que también minimiza los errores, resultando en aplicaciones más estables y eficientes.

A través de este ejercicio, exploramos los fundamentos de Swift para construir una aplicación sencilla pero funcional. El objetivo principal es diseñar una aplicación que permita al usuario ingresar un número y, a partir de este, determinar si es par o impar. Esta tarea nos servirá para familiarizarnos con conceptos clave como la entrada de datos por el usuario, el manejo de la lógica condicional y la salida de resultados, sentando las bases para proyectos más complejos en el futuro.

Justificación.

La realización de esta primera actividad se justifica plenamente por su papel fundamental en la adquisición de las habilidades esenciales para el Desarrollo de Aplicaciones Móviles III con Swift. Al construir una aplicación que determina si un número es par o impar, los estudiantes no solo se familiarizan con la sintaxis de Swift, un lenguaje crucial en el ecosistema de Apple, sino que también aplican conceptos de lógica de programación de manera práctica.

Este ejercicio actúa como un pilar educativo, permitiendo la comprensión de procesos clave como la interacción del usuario y la toma de decisiones lógicas (condicionales if-else). La elección de Swift se justifica por su seguridad, legibilidad y eficiencia, que garantizan la creación de aplicaciones robustas. En definitiva, esta actividad sienta las bases sólidas para afrontar desafíos más complejos, construyendo una comprensión práctica y duradera del desarrollo de apps.

Entre otros puntos adicionales a utilizar en la justificación para la realización de la documentación de esta actividad que son los siguientes:

- PDF de esta actividad en el portafolio GitHub.
- Descarga el script genera de swift y subirlo como parte de la actividad a GitHub
- Anexa link de GitHub en documento.
- Utilizar la herramienta del entorno de trabajo online Replit o XCode que se usarán con lenguaje de programación: Swift.

Desarrollo.

En esta actividad final de la materia continuamos con el desarrollo de la aplicación que determina si un número es par o impar, aquí tomaremos en cuenta que utilizaremos la consola de programación online Repl con el cual utilizaremos el lenguaje de programación Swift. Esto será documentado en los puntos siguientes de este documento donde nuestro propósito central es garantizar una interpretación de manera

correcta del desglose y uso del script de la aplicación a crear.

Link: [GitHub](#).

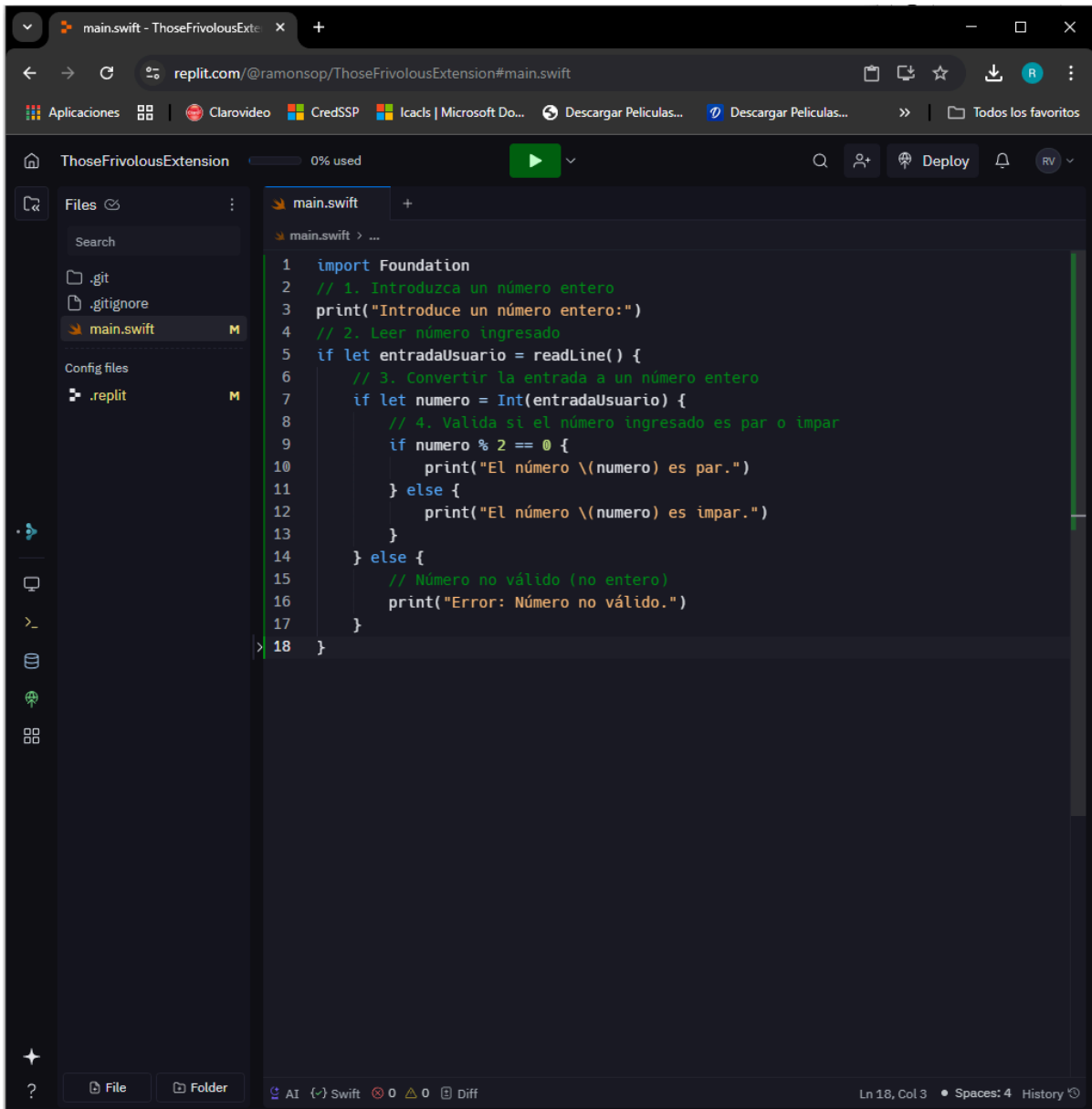
Link: [Script](#).

Codificación.

En este punto de la actividad realizaremos un breve descripción de las líneas del script de la aplicación que determinará si un número es par o impar y se anexará una imagen del código como evidencia adicional.

Codificación de app.	
import Foundation	Es la primera línea y es necesaria para acceder a herramientas básicas de Swift, como la función <code>readLine()</code> . Luego, la línea <code>print("Introduce un número entero:")</code> simplemente muestra un mensaje en la consola para pedirle al usuario que ingrese un número.
if let entradaUsuario = readLine()	Intenta leer la entrada del usuario desde la consola. La variable <code>entradaUsuario</code> guardará lo que escriba el usuario. El <code>if let</code> es un método seguro de Swift para manejar datos opcionales, asegurando que el programa no falle si no se puede leer ninguna entrada.
if let numero = Int(entradaUsuario)	Intenta convertir el texto que ingresó el usuario (que se guardó como un <code>String</code>) a un número entero (<code>Int</code>). Este paso es crucial porque las operaciones matemáticas solo se pueden realizar con números enteros.

if numero % 2 == 0	<p>Es la lógica principal. El operador % se llama módulo y devuelve el residuo de una división. En este caso, si el residuo de dividir número entre 2 es 0, significa que el número es par.</p> <ul style="list-style-type: none">• Si el residuo es 0, se ejecuta el primer print que dice que el número es par.• Si el residuo no es 0, se ejecuta la parte else, que imprime que el número es impar
else y su print("Error: Número no válido.")	<p>El último se encarga de los casos en los que el usuario ingresa algo que no puede ser convertido a un número entero (por ejemplo, la palabra "hola" o algún número con decimal en lugar de un número entero). De esta forma, el programa informa al usuario que hubo un error en lugar de detenerse inesperadamente.</p>



The screenshot shows a web browser window with the URL `replit.com/@ramonsop/ThoseFrvilousExtension#main.swift`. The browser's address bar and tabs are visible at the top. Below the browser, the Replit interface is shown with a dark theme. On the left, a file explorer sidebar displays the project structure: `Files` (containing `.git`, `.gitignore`, and `main.swift`) and `Config files` (containing `.replit`). The main editor area shows the `main.swift` file with the following Swift code:

```
1 import Foundation
2 // 1. Introduzca un número entero
3 print("Introduce un número entero:")
4 // 2. Leer número ingresado
5 if let entradaUsuario = readLine() {
6     // 3. Convertir la entrada a un número entero
7     if let numero = Int(entradaUsuario) {
8         // 4. Valida si el numero ingresado es par o impar
9         if numero % 2 == 0 {
10             print("El número \(numero) es par.")
11         } else {
12             print("El número \(numero) es impar.")
13         }
14     } else {
15         // Número no válido (no entero)
16         print("Error: Número no válido.")
17     }
18 }
```

The status bar at the bottom of the editor indicates the current position: `Ln 18, Col 3`, `Spaces: 4`, and a `History` button.

Prueba de la aplicación.

En este punto realizaremos la ejecución del código de la aplicación que nos determinará si el número ingresado en la aplicación es par, impar o enviará un mensaje de error por realizar el ingreso de datos incorrectos como texto o algún número con decimales y anexamos la evidencia de cada uno de los resultados agregando una screenshot del resultado obtenido.

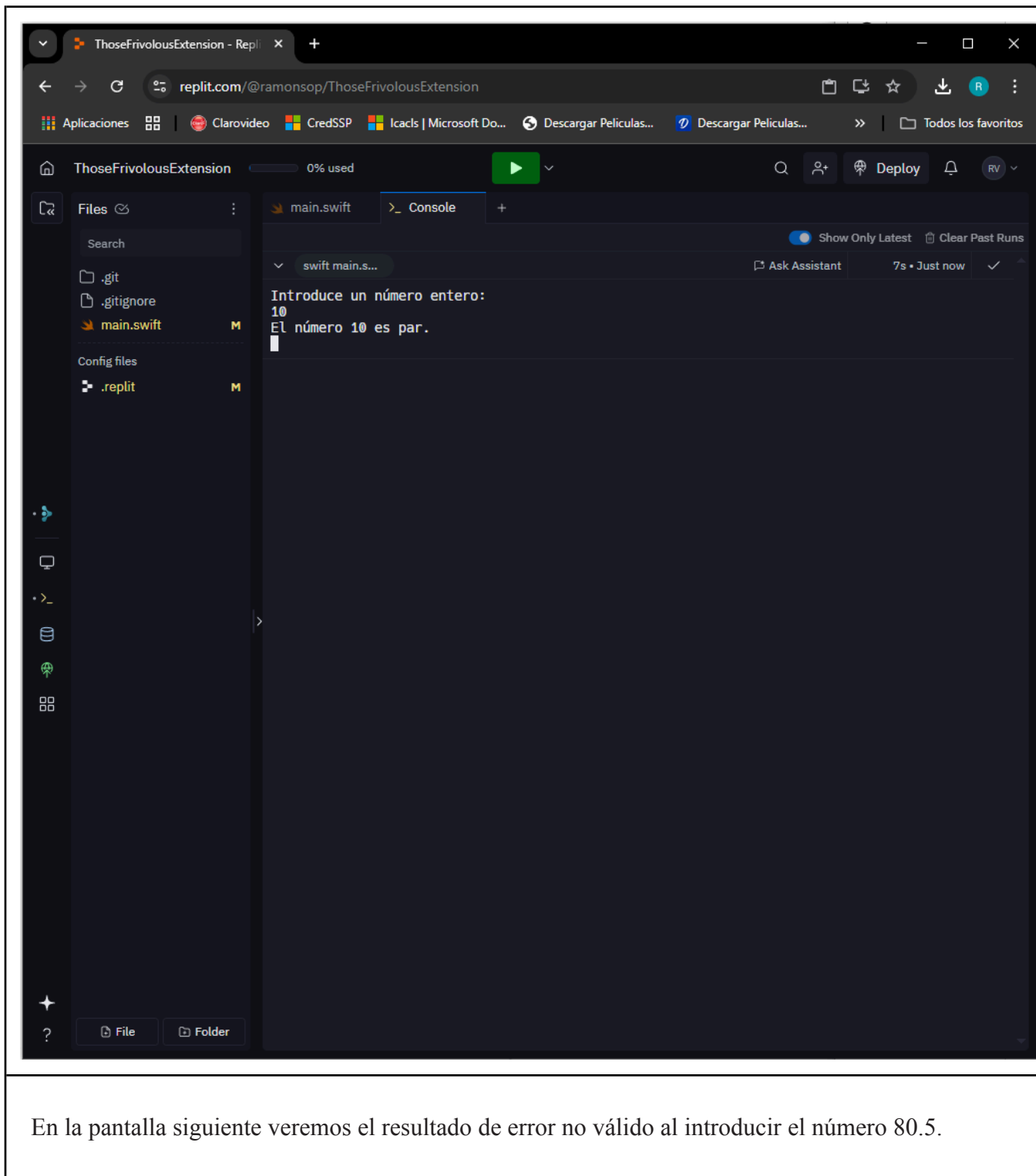
Pruebas de ejecución de la app:

En la pantalla siguiente veremos el resultado de un número impar al introducir el número 9.

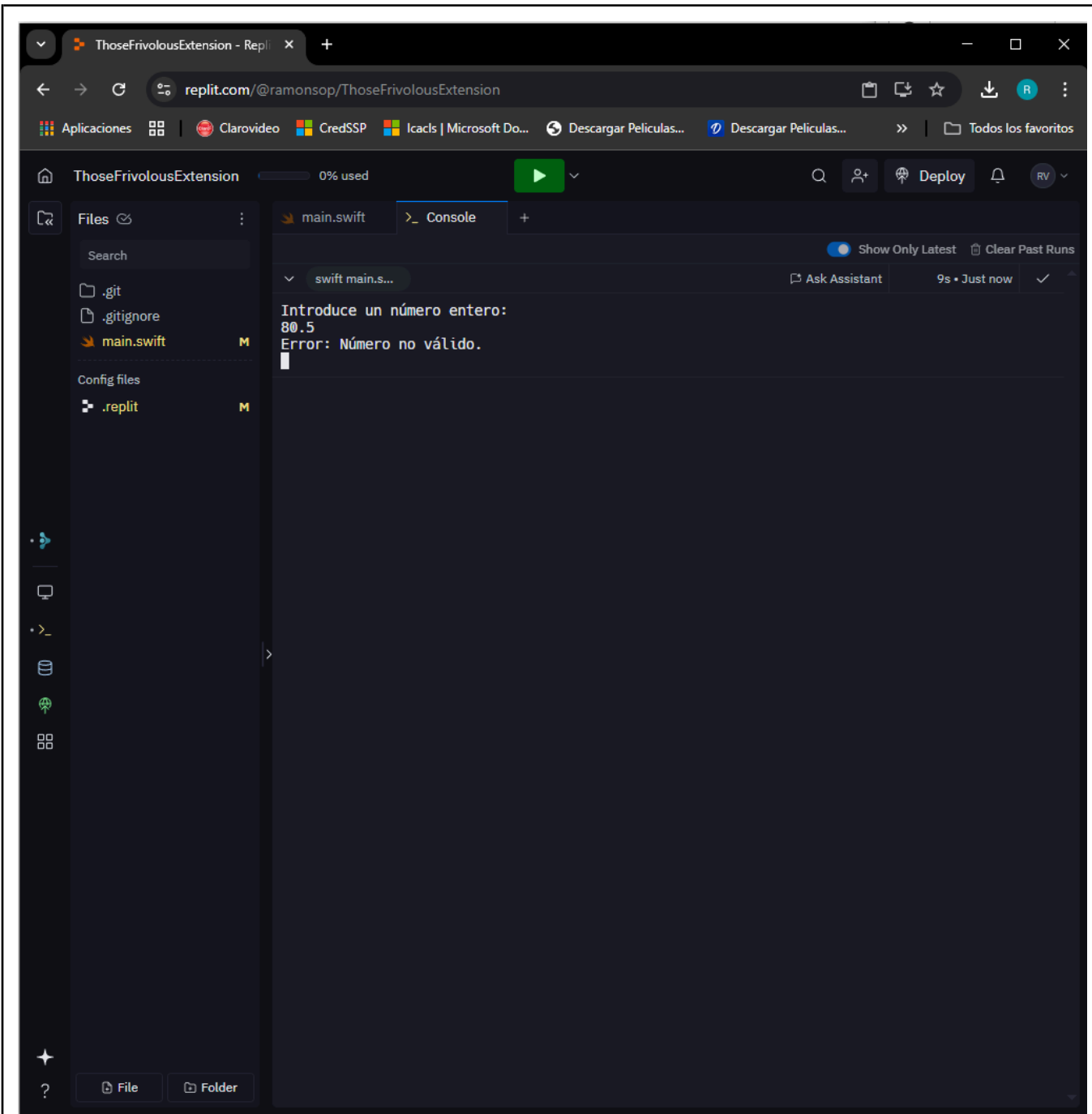
The screenshot shows a web browser window with the URL `replit.com/@ramonsop/ThoseFrivolousExtension`. The page displays a Swift REPL environment. On the left, a file explorer shows a project named `ThoseFrivolousExtension` with files `.git`, `.gitignore`, `main.swift`, and `.replit`. The main area shows the `main.swift` file and a terminal window. The terminal output is as follows:

```
Introduce un número entero:
9
El número 9 es impar.
```

En la pantalla siguiente veremos el resultado de un número par al introducir el número 10.

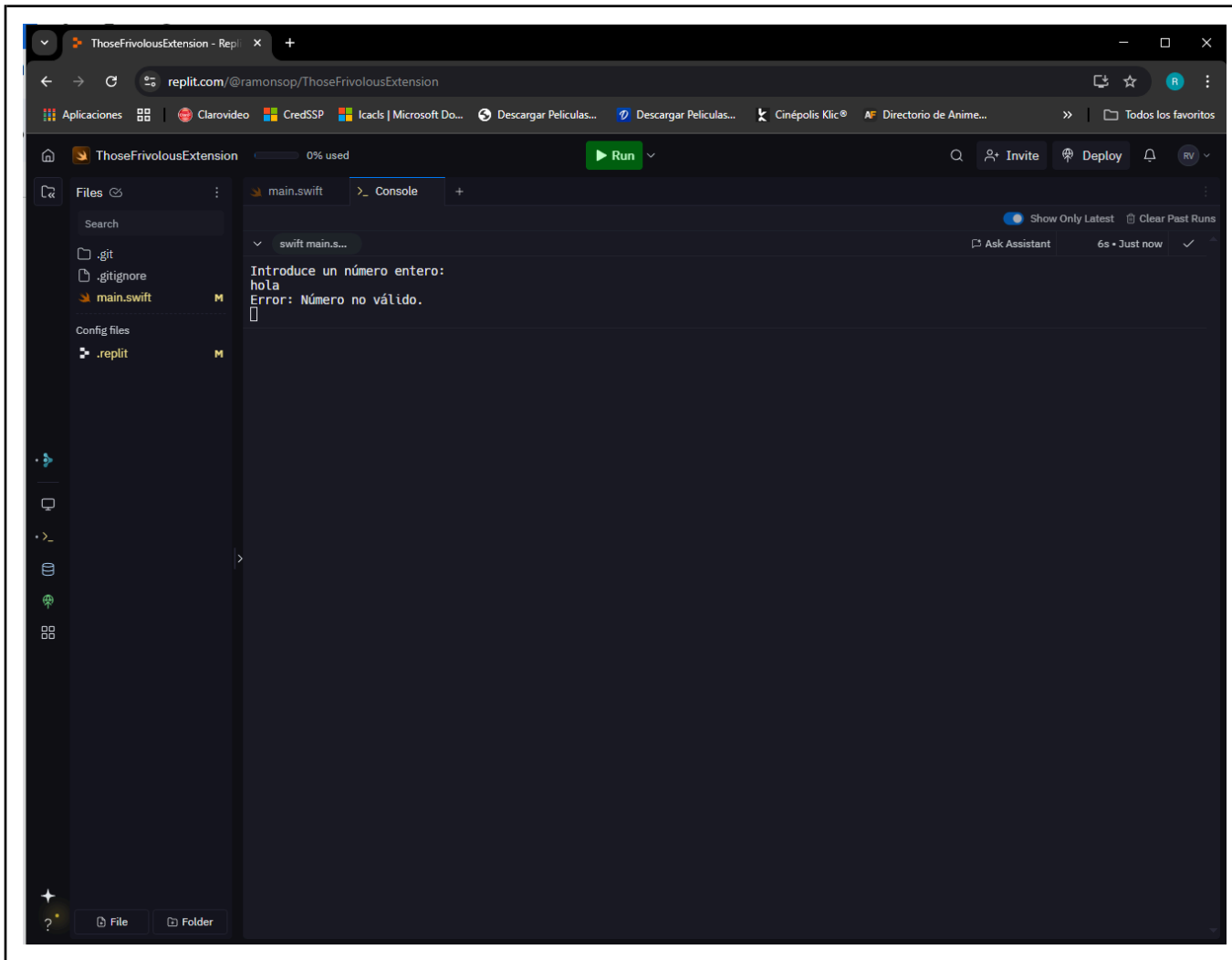


En la pantalla siguiente veremos el resultado de error no válido al introducir el número 80.5.



The screenshot shows a web browser window with the URL `replit.com/@ramonsop/ThoseFrivolousExtension`. The browser's address bar and tabs are visible at the top. Below the browser, the Replit interface is shown. The left sidebar contains a file explorer with a search bar and a list of files: `.git`, `.gitignore`, `main.swift`, and `.replit`. The main area is divided into two tabs: `main.swift` and `_ Console`. The `_ Console` tab is active, displaying the output of a Swift program. The output shows the prompt `Introduce un número entero:`, the user input `80.5`, and the resulting error message `Error: Número no válido.`. The console also shows a status bar at the bottom with `9s • Just now` and a checkmark icon.

En la última pantalla veremos el resultado de error no válido al introducir texto en el cuadro donde se debe ingresar un número entero.



Conclusion.

En conclusión: esta actividad, aparentemente simple, es una piedra angular en la formación de un desarrollador Swift. Al manejar la entrada de usuario, la conversión de datos y la lógica condicional, se establecen las bases para cualquier aplicación interactiva. En el campo laboral, estas habilidades son cruciales para construir interfaces de usuario responsivas, validar datos en formularios o implementar flujos de negocio complejos en apps de banca, comercio o salud.

En la vida cotidiana del desarrollador, este ejercicio refuerza el pensamiento lógico y la resolución de problemas, habilidades transferibles a cualquier desafío. Aprender a manejar errores y a estructurar el código de manera limpia son prácticas esenciales que garantizan la creación de software robusto y eficiente. En definitiva, esta primera inmersión en Swift no solo enseña a programar, sino a pensar

como un desarrollador. Como dato personal y adicional, se me hizo muy fácil el poder realizar esta actividad debido a que la consola de programación es muy intuitiva y aporta mucho apoyo al desarrollador con el autoayuda o autocompletar al estar desarrollando alguna aplicación.

Referencias.

Gemini - chat to supercharge your ideas. (n.d.). Gemini. Retrieved January 9, 2025, from <https://gemini.google.com/>

Ingeniería en desarrollo de software. (n.d.). Edu.Mx. Retrieved January 9, 2025, from <https://umi.edu.mx/coppel/IDS/login/index.php>