Universidad Corporativa | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad | 1 |

Instalación XCode/Aplicación 1

Desarrollo de Aplicaciones Móviles III

Ingeniería en Desarrollo de Software

academia global

TUTOR: Sandra Luz Lara Devora

ALUMNO: Yanira Lizbeth Lopez Navarro

FECHA: 19/03/2024

Índice

Introducción 3
Descripción 4
Justificación 5
Desarrollo: 6
Conclusión 13
Referencias 14

Introducción

En el ámbito del desarrollo de software, contar con las herramientas adecuadas es fundamental para llevar a cabo proyectos de manera eficiente y exitosa. En esta actividad, nos enfocaremos en una solución innovadora para aquellos que desean desarrollar aplicaciones para el ecosistema de Apple, pero que trabajan en un sistema operativo Windows: la instalación de Xcode y la creación de la Aplicación 1 utilizando Replit.

Xcode es una potente suite de desarrollo integrado (IDE) diseñada por Apple, ampliamente utilizada por desarrolladores de todo el mundo para crear aplicaciones para dispositivos iOS, macOS. Sin embargo, su uso tradicionalmente ha estado limitado a los sistemas operativos macOS. Aquí es donde entra en juego Replit, una plataforma en línea que ofrece un entorno de desarrollo en la nube, lo que permite a los usuarios ejecutar y desarrollar aplicaciones utilizando diversos lenguajes de programación, incluido Swift, el lenguaje de programación principal de Xcode.

En esta actividad, exploraremos los pasos necesarios para instalar Xcode en un sistema operativo Windows utilizando Replit como plataforma de desarrollo. Además, nos sumergiremos en la creación de la Aplicación 1, aprovechando las capacidades de Replit para desarrollar y probar nuestro proyecto de manera efectiva.

Descripción

En el contexto proporcionado, se destaca la relevancia de Swift como un lenguaje de programación moderno que no solo ofrece mayor seguridad, sino que también se caracteriza por su intuitividad. Esta combinación de características hace que Swift sea idóneo para el desarrollo de diversas aplicaciones. Se enfatiza la necesidad de adquirir un profundo conocimiento sobre el funcionamiento de Swift para poder diseñar y desarrollar aplicaciones de manera efectiva.

La actividad plantea la tarea de crear una aplicación específica utilizando Swift: se busca desarrollar una aplicación que permita ingresar un número y determine si es par o impar. Esta solicitud demuestra cómo, a través de Swift, es posible abordar problemas prácticos y crear soluciones funcionales de manera eficiente.

La elección entre la instalación de Xcode o el uso de compiladores en línea se presenta como una alternativa, brindando flexibilidad a los desarrolladores según sus preferencias. Este enfoque permite adaptarse a diferentes entornos y facilita el acceso a Swift, incluso para aquellos que no tienen acceso inmediato a un sistema operativo compatible con Xcode.

En resumen, la actividad propone un desafío práctico que combina la comprensión del lenguaje de programación Swift con la creación de una aplicación concreta, reforzando la idea de que la adquisición de conocimientos en Swift capacita a los desarrolladores para abordar diversas problemáticas de manera efectiva y concreta.

Justificación

La elección de emplear Swift y la instalación de Xcode o el uso de compiladores en línea para la actividad propuesta se justifica por varias razones fundamentales que convergen en la eficiencia y accesibilidad del proceso de desarrollo de aplicaciones.

En primer lugar, Swift se destaca como un lenguaje de programación moderno, diseñado específicamente para brindar mayor seguridad y ser intuitivo. Estas características son cruciales, especialmente en proyectos de desarrollo de software, ya que facilitan la comprensión del código, reducen la probabilidad de errores y contribuyen a la creación de aplicaciones más robustas y seguras.

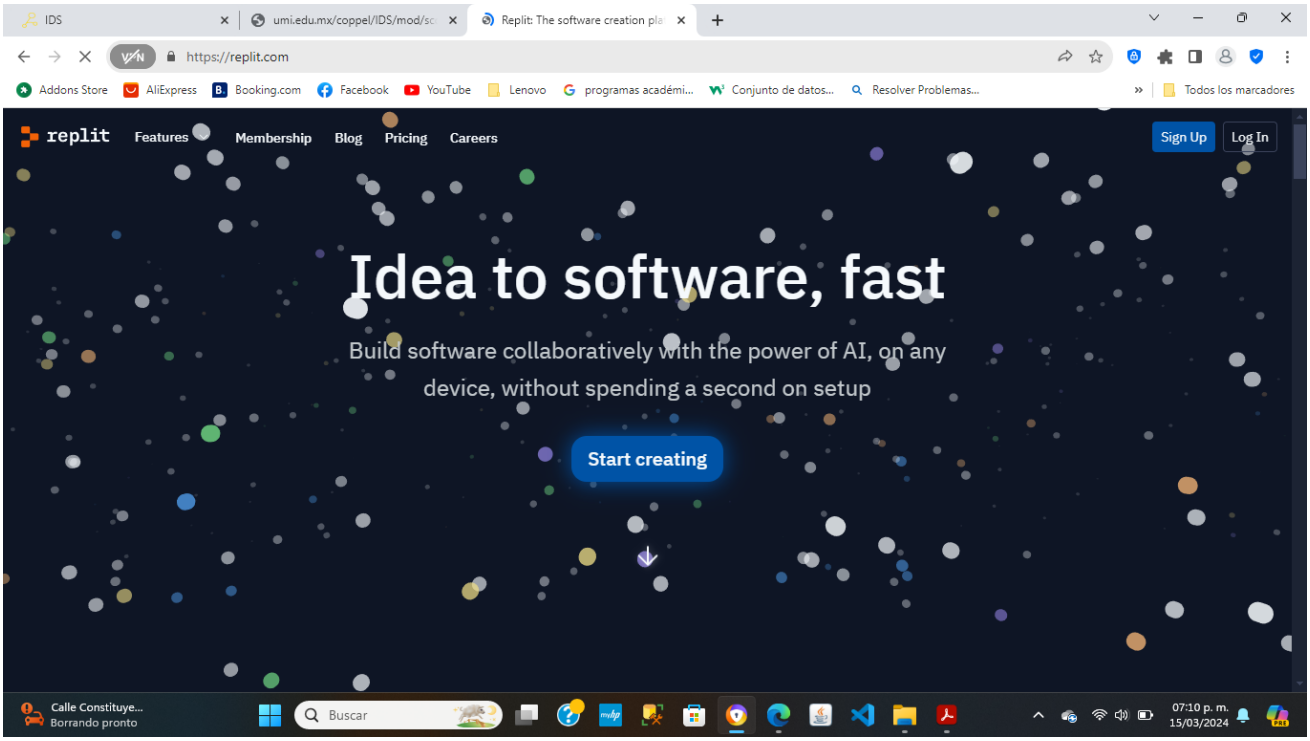
La elección de Swift también se respalda en su versatilidad. Dada su capacidad para desarrollar aplicaciones para diversos dispositivos en el ecosistema Apple, desde iOS hasta macOS, proporciona una solución integral para cubrir una amplia gama de necesidades de desarrollo. En cuanto a la instalación de Xcode o el uso de compiladores en línea, la flexibilidad es clave.

Permitir a los desarrolladores elegir entre estas opciones se adapta a diversos contextos y niveles de experiencia. Xcode, como entorno de desarrollo integrado completo, es ideal para aquellos que buscan un conjunto robusto de herramientas y funciones. Por otro lado, los compiladores en línea ofrecen accesibilidad instantánea a Swift sin requerir instalaciones locales, lo que resulta beneficioso para principiantes y aquellos que no tienen acceso a un entorno de desarrollo específico.

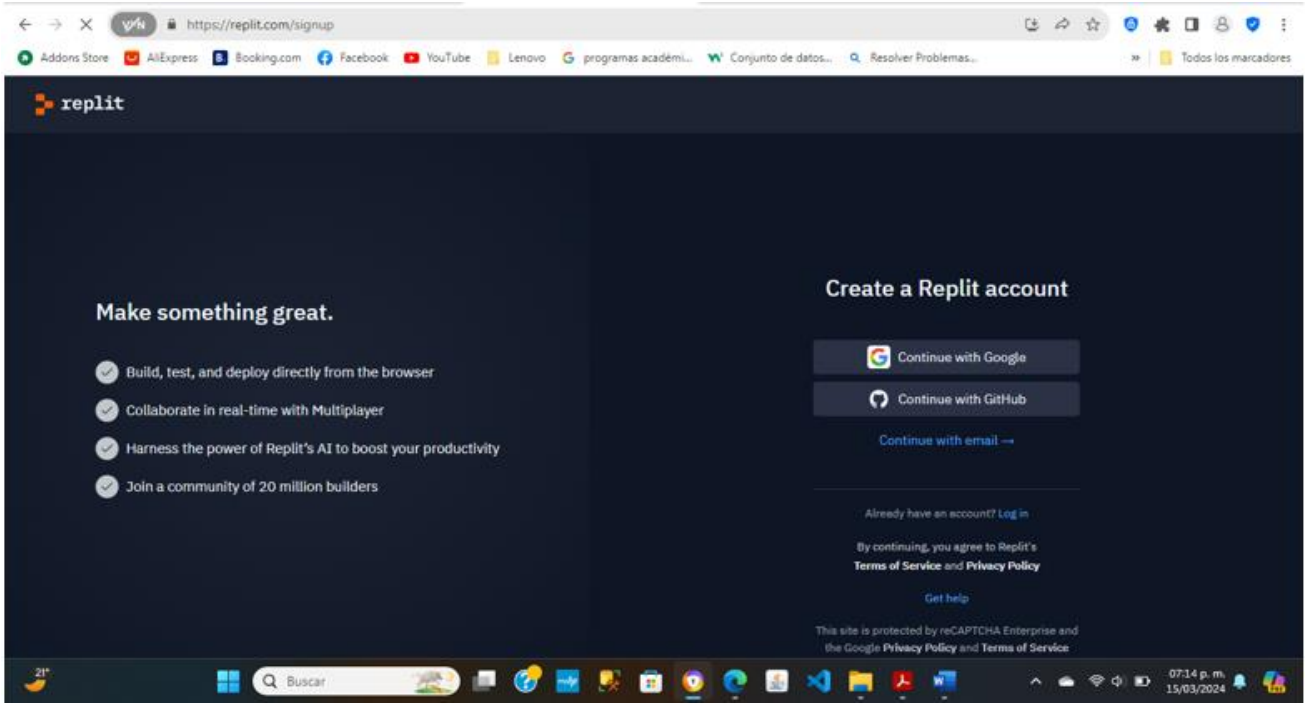
En resumen, la elección de emplear Swift y la instalación de Xcode o compiladores en línea se justifica por su combinación de seguridad, versatilidad y flexibilidad, brindando a los desarrolladores un entorno propicio para abordar la actividad de manera eficiente y adaptada a sus necesidades particulares.

Desarrollo:

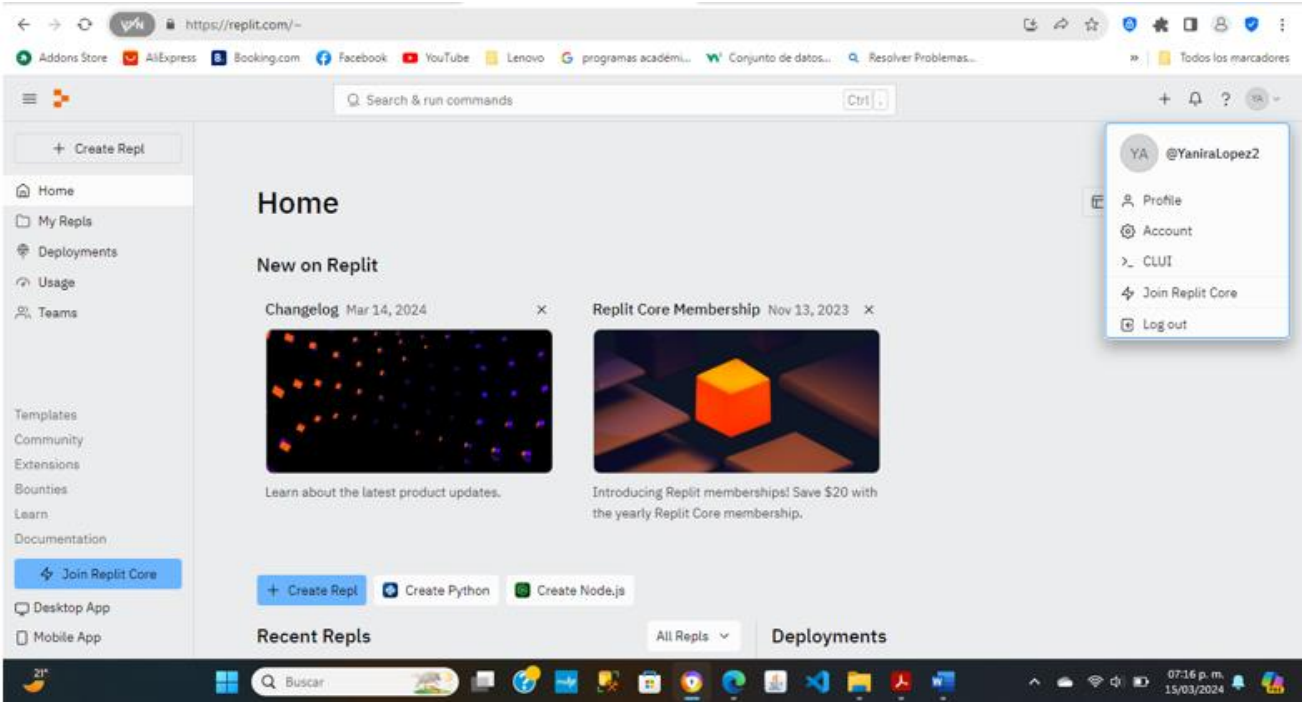
En la siguiente imagen podemos observar el inicio a la página de https://replit.com/ la cual nos permitirá realizar el registró utilizando la opción Sing Up.



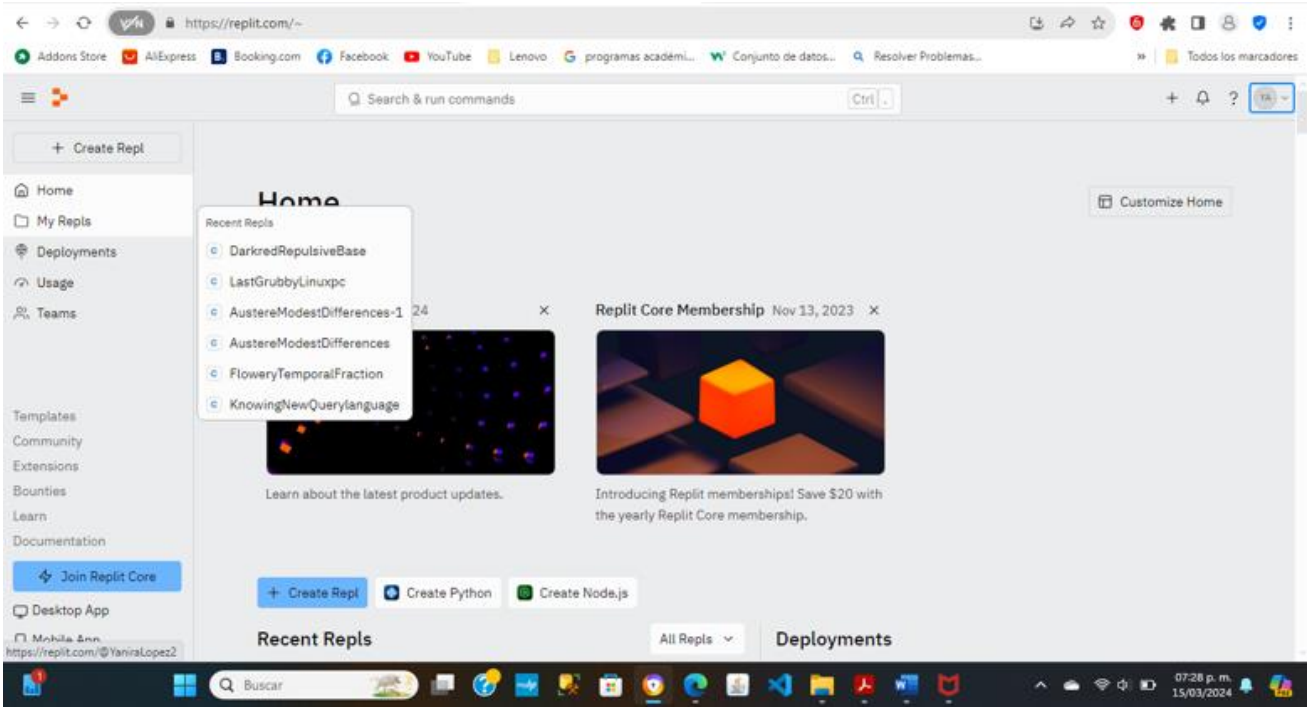
Una vez que estamos en la página de Replit procedemos realizar el registro mediante nuestra cuenta de Google, aunque cuenta con más opciones.



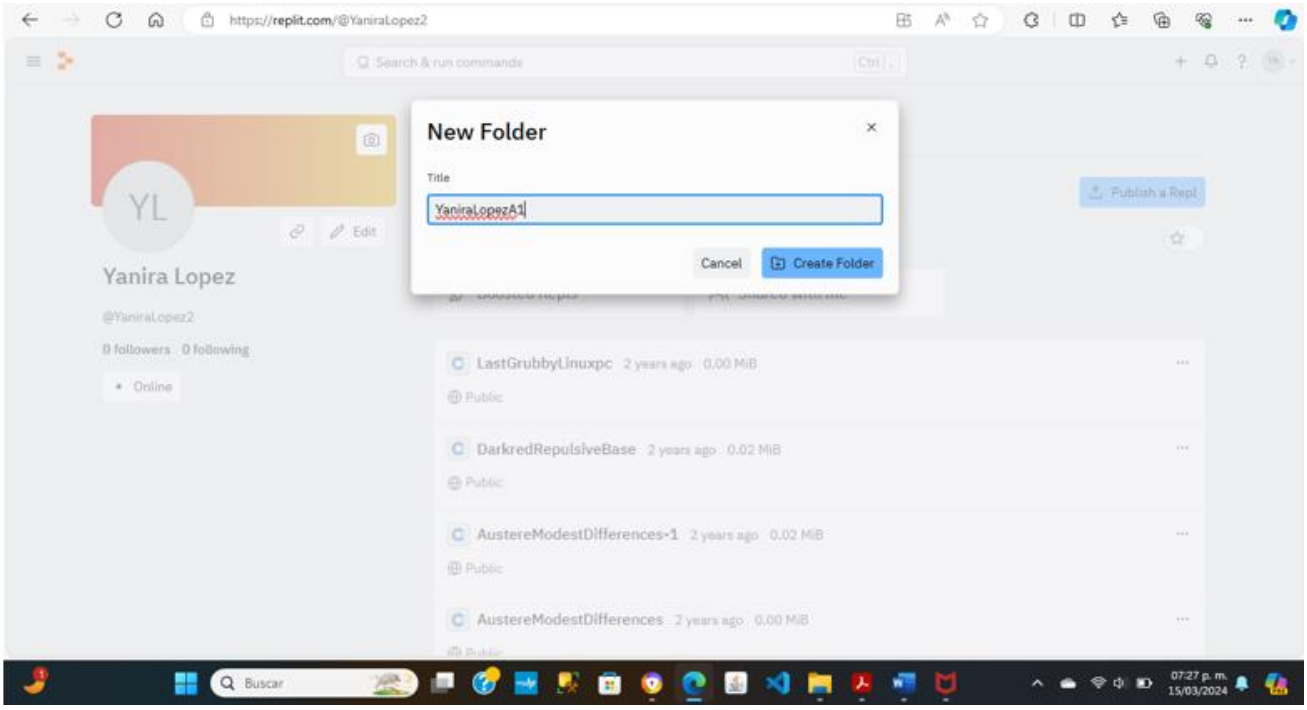
En la siguiente imagen podemos observar que ya se llevó a cabo el registro el cual se confirma mediante mi cuenta @YaniraLopez2.



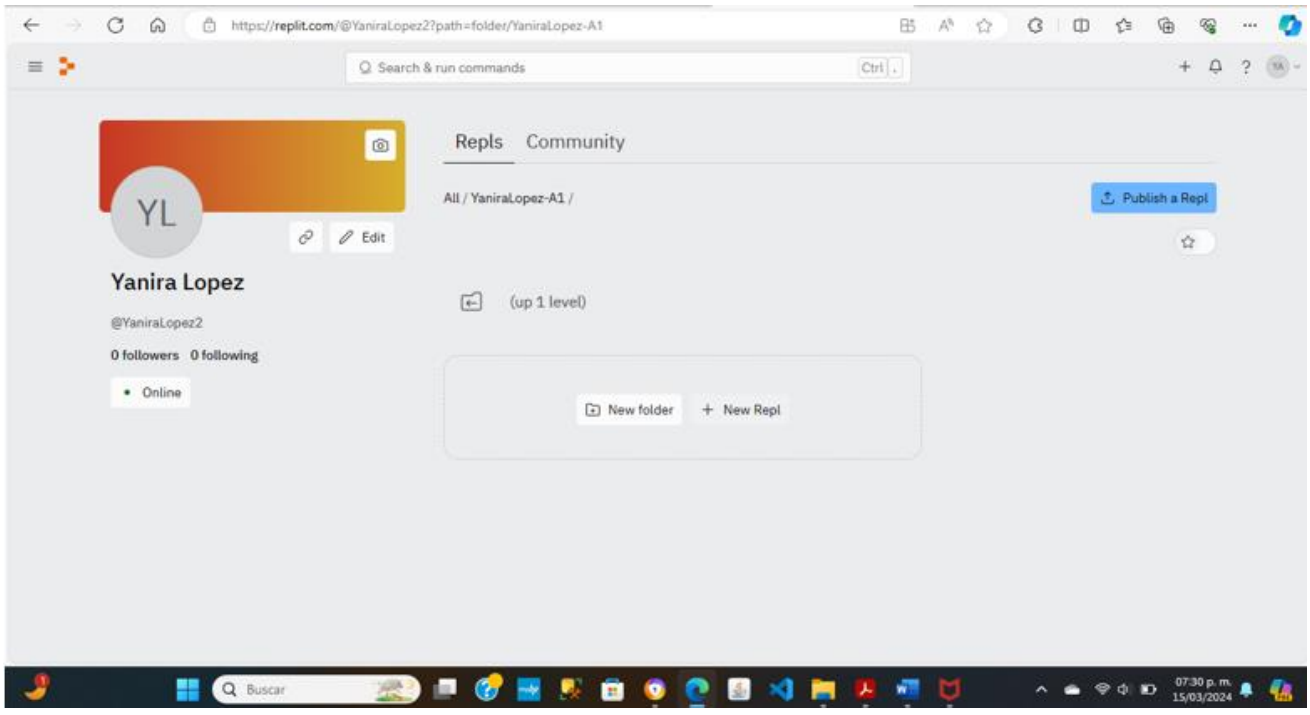
A continuación, podemos observar la pantalla en la cual podemos crear un repositorio.



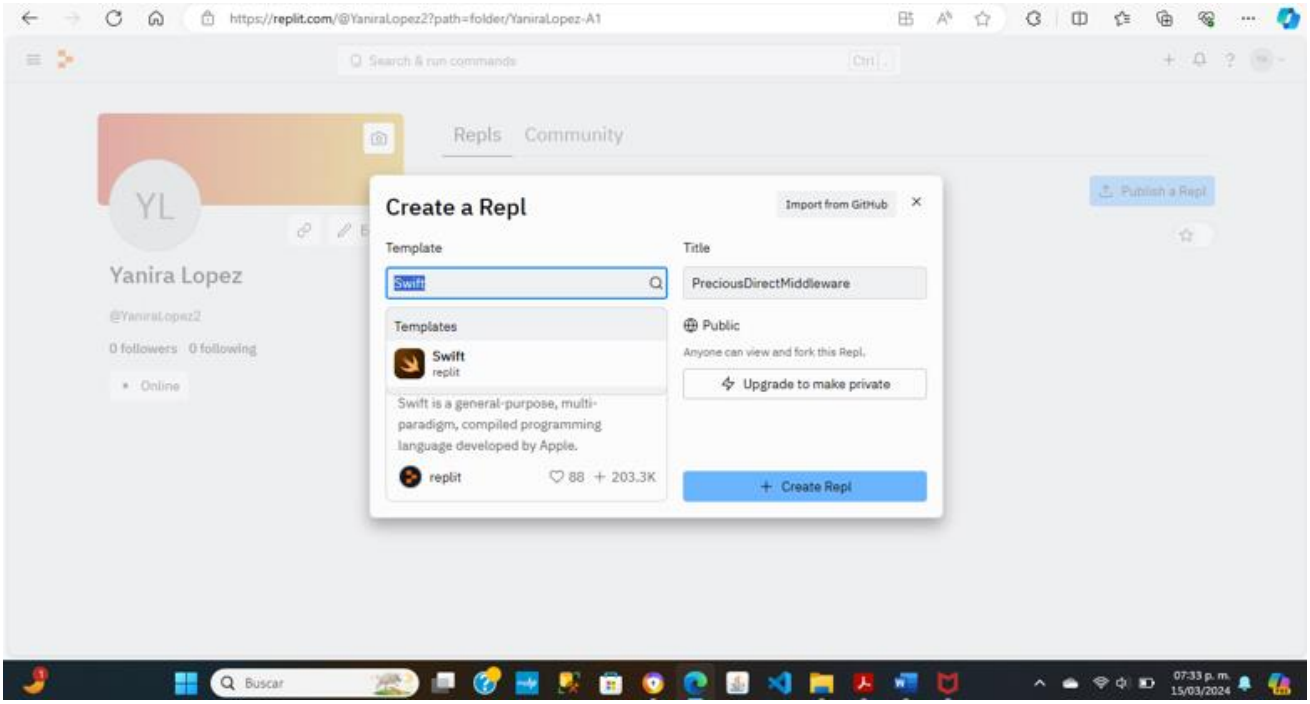
En la siguiente imagen podemos observar cómo se está creando un nuevo folder el cual vamos a nombrar YaniraLopezA1.



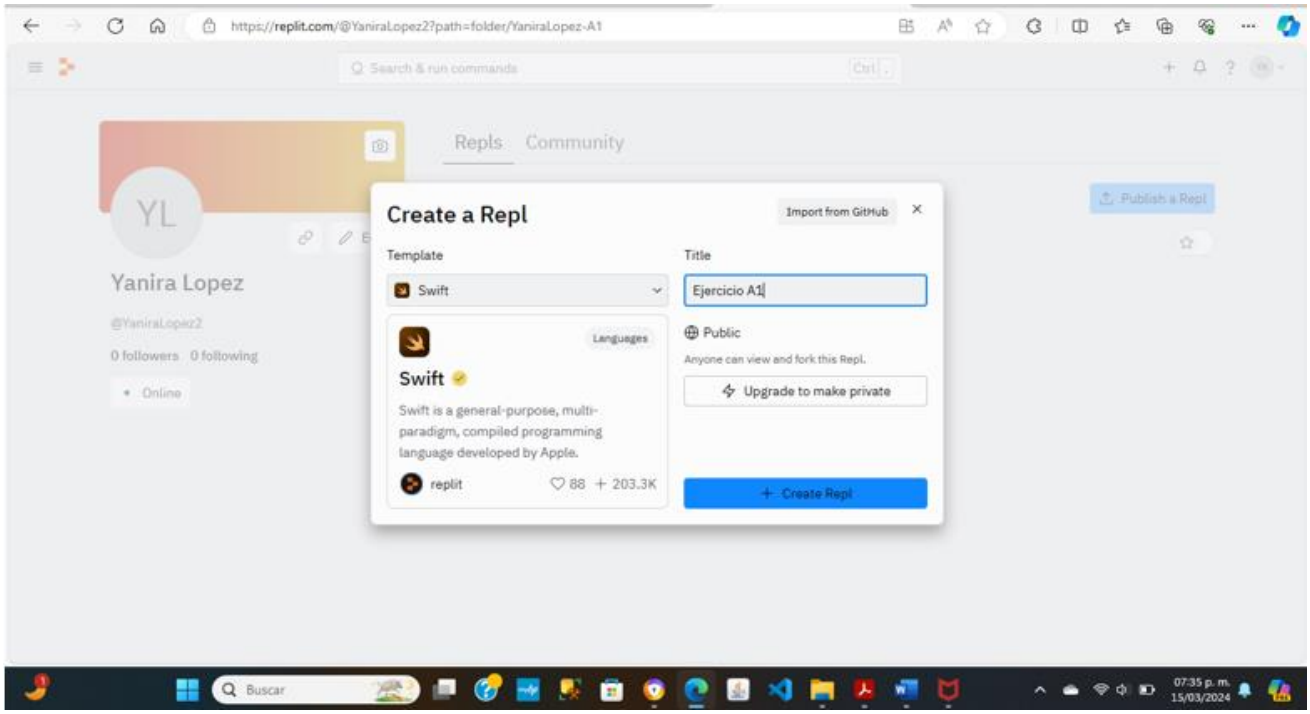
En la siguiente imagen sé que ya se realizó el nuevo folder y en el cual podemos crear una nuevo Replit.



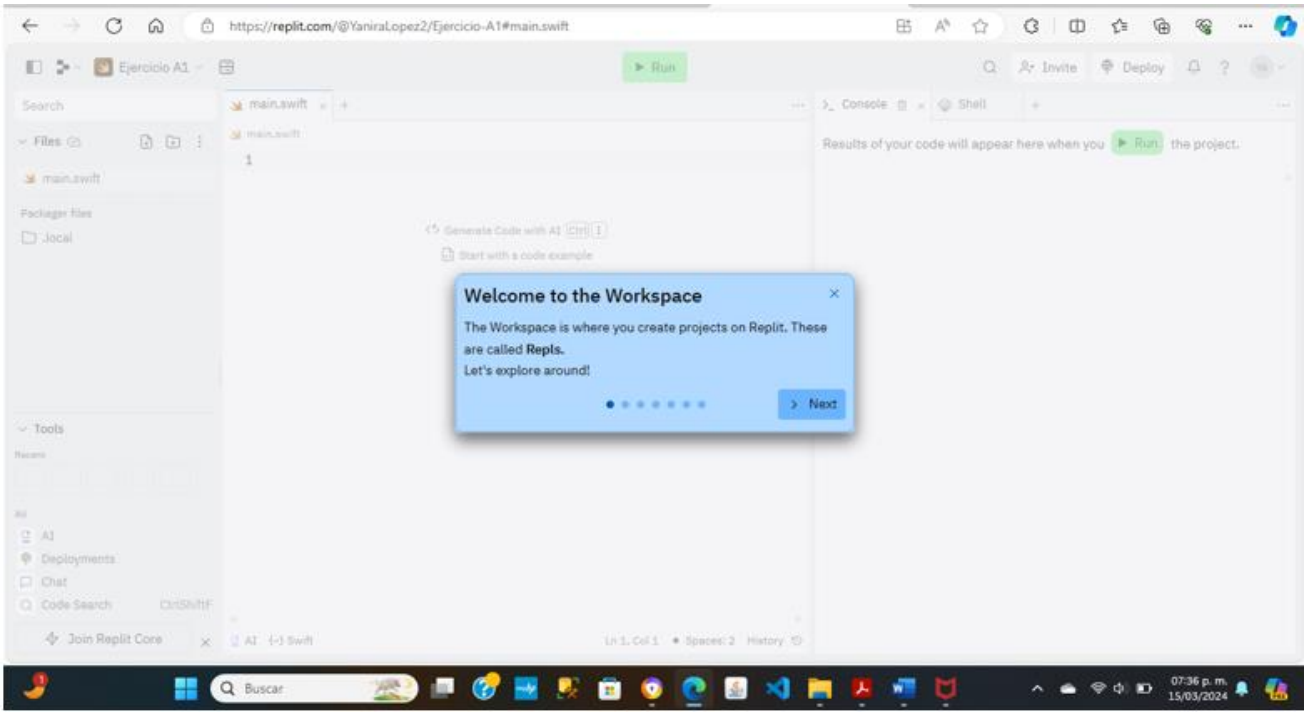
Una vez que se está creando el Repl configuramos el lenguaje Swift con el que vamos a estar trabajando durante el desarrollo de la actividad.



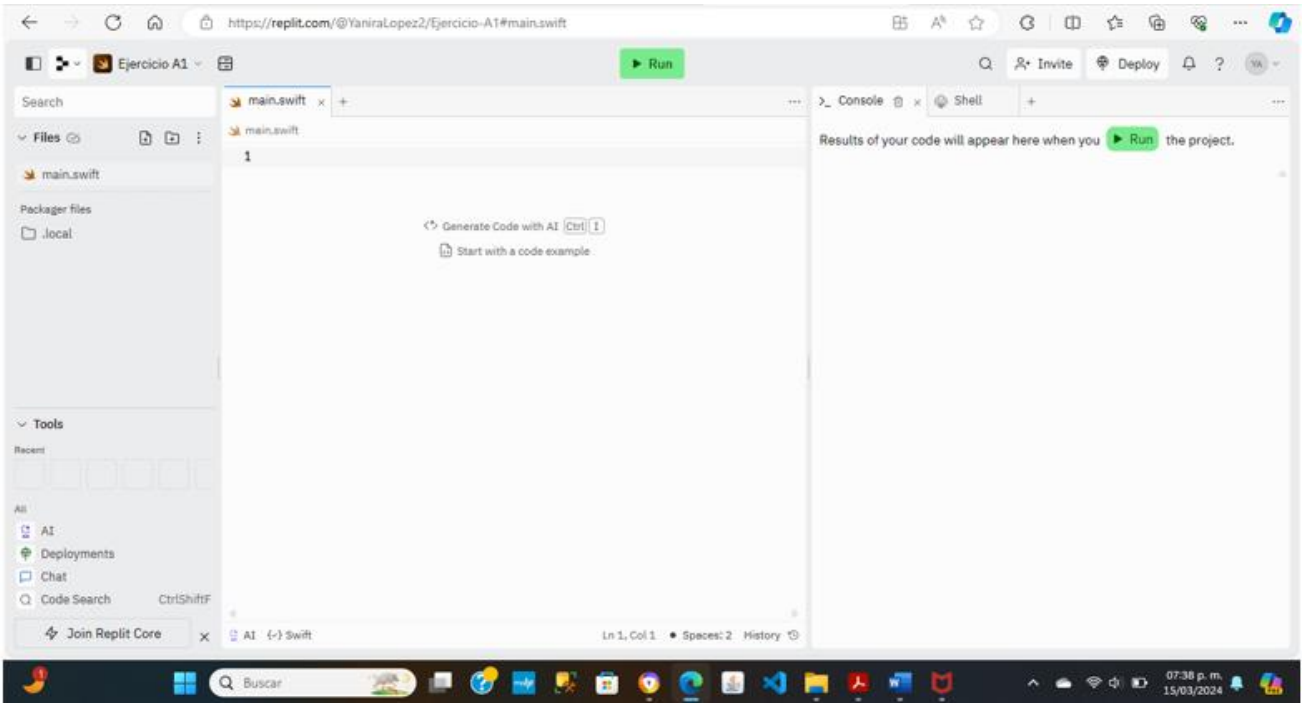
En la siguiente imagen podemos observar el nombre que se le asignar a nuestro Repl el cual se nombrara como Ejercicio A1.



Una vez que se realizaron las configuraciones anteriores nos arroja el siguiente mensaje de confirmación.



En las siguiente imagen se muestra el entorno en el cual procedemos llevar acabo la codificación de lo solicitado en la actividad.



Codificación

Este código solicita al usuario ingresar un número, luego verifica si es par o impar utilizando el operador de módulo. Finalmente, imprime el resultado. **%**

Esta línea importa el framework Foundation de Swift, que proporciona tipos y funciones fundamentales para trabajar con datos, fechas, archivos, y más.

```
import Foundation
```

Aquí se define una función llamada 'verificarParidad' que toma un parámetro 'número' de tipo entero y devuelve una cadena ('String') como resultado.

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

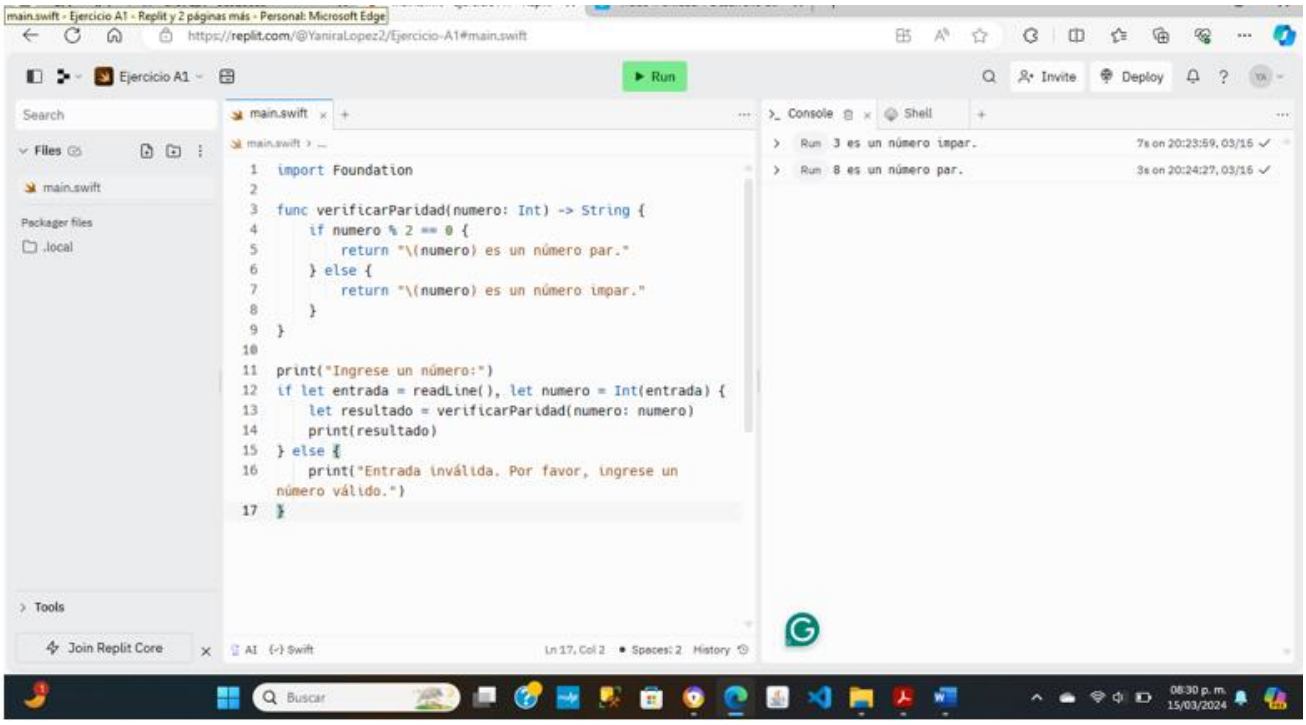
Esta es la lógica principal de la función. Se utiliza el operador de módulo '%' para verificar si el número es divisible por 2. Si el residuo de la división es 0, entonces el número es par; de lo contrario, es impar. Dependiendo del resultado, se construye una cadena que indica si el número es par o impar.

```
print ("Ingrese un número:")
if let entrada = readLine (), let numero = Int(entrada) {
    let resultado = verificarParidad (numero: numero)
    print(resultado)
} else {
    print ("Entrada inválida. Por favor, ingrese un número válido.")
}
```

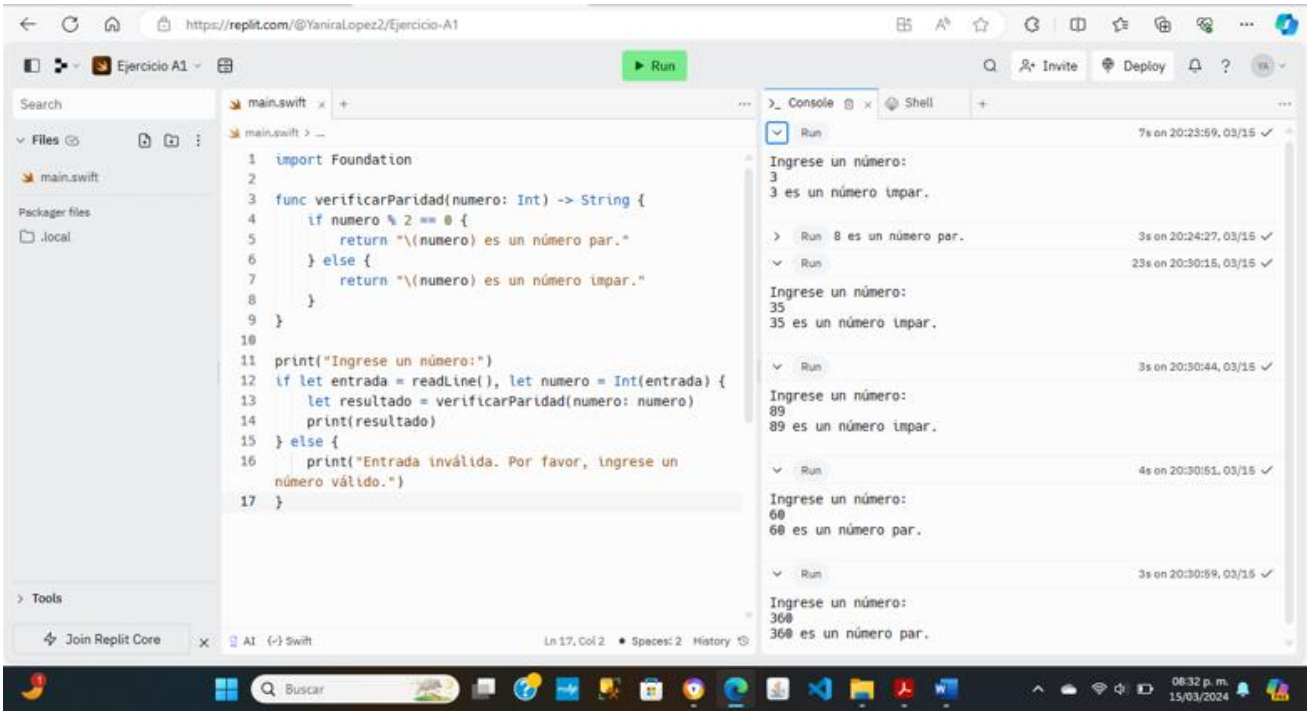
Aquí se encuentra la lógica principal del programa. Imprime un mensaje solicitando al usuario que ingrese un número. Luego, utiliza 'readLine ()' para leer la entrada del usuario como una cadena de texto opcional. Si la entrada es válida y puede ser convertida a un número entero ('Int'), entonces se llama a la función 'verificarParidad' para determinar si el número es par o impar, y se imprime el resultado. En caso de que la entrada no sea válida, se imprime un mensaje indicando que la entrada es inválida y se le pide al usuario que ingrese un número válido.

Prueba de la aplicación

En la siguiente imagen nos nuestro el código que nos permite cumplir con lo solicitado en la actividad.



Una vez que se llevó a cabo la configuración en el apartado de consola podemos observar el resultado obtenido una vez que se llevó a cabo la ejecución del código y en el cual se introdujo algunos números que permitió comprobar que si se cumple con lo solicitado en la actividad.



Conclusión

La conclusión de la actividad, que involucra la creación de una aplicación para determinar si un número es par o impar utilizando Swift y la instalación de Xcode o compiladores en línea, refleja una experiencia significativa con implicaciones tanto en el ámbito laboral como en la vida cotidiana de un desarrollador de software.

Desde una perspectiva laboral, la capacidad de trabajar con Swift y utilizar herramientas como Xcode es esencial para los profesionales que buscan destacarse en el desarrollo de aplicaciones para el ecosistema de Apple. La aplicación práctica de los conocimientos adquiridos en esta actividad permite a los desarrolladores abordar problemas específicos y crear soluciones funcionales, contribuyendo así a su habilidad para desarrollar software robusto y eficiente.

Además, la versatilidad de Swift y el uso de compiladores en línea ofrecen flexibilidad en el entorno de desarrollo, adaptándose a diferentes contextos laborales. Esto se traduce en una mayor capacidad para enfrentar desafíos específicos en el campo laboral, ya que los desarrolladores pueden ajustar su enfoque según las necesidades del proyecto y sus propias preferencias.

En la vida cotidiana, la actividad resalta la aplicabilidad práctica de las habilidades de programación. La capacidad de crear una aplicación que resuelva una tarea específica, como determinar la paridad de un número, demuestra cómo los conocimientos técnicos pueden tener impacto en situaciones diarias. Además, la experiencia adquirida al instalar herramientas de desarrollo y trabajar en proyectos concretos proporciona una sensación de logro y confianza que se traduce en un mayor disfrute y satisfacción personal en el ámbito de la programación.

En conclusión, la actividad no solo fortalece las habilidades técnicas y laborales, sino que también resalta la relevancia y la aplicación práctica de estas habilidades en la vida cotidiana, subrayando la importancia de la programación y el desarrollo de aplicaciones en el mundo actual.

Referencias

Ingeniería en desarrollo de software. Universidad México Internacional. Recuperado el día 10 de marzo de 2024, https://umi.edu.mx/coppel/IDS/mod/scorm/player.php

Video conferencing, web conferencing, webinars, screen sharing. (s. f.). Zoom. https://academiaglobal-mx.zoom.us/rec/play/IVp0pwPkWuHpqrhSr1gaJKufUZuOBjXTXtNWpi4fki40cLhPOC16JdXCwVM13JnSwIjdch907-aGFa2B.b_Dya6X4IzgwjkaL?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2FAJ8WceZxxDx59vIkFrGFwchemU4pY7gSXb4X7zcMWAInWas8y00-m9IhuXLwALGX.mZCT1fbTEoKhl-uQ