

Actividad | 3 |

Menú de Áreas de Figuras Geométricas

Desarrollo de Aplicaciones Móviles III

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Devora

ALUMNO: Yanira Lizbeth Lopez Navarro

FECHA: 05/04/2024

Índice

Introducción 3

Descripción 4

Justificación 5

Desarrollo: 6

Conclusión 15

Referencias 16

Introducción

En el ámbito de la programación de aplicaciones móviles, la creación de soluciones que aborden necesidades específicas y proporcionen utilidad a los usuarios es un desafío constante. Una de las áreas donde estas soluciones pueden ser especialmente útiles es en la realización de cálculos matemáticos, como el cálculo de áreas de figuras geométricas. En este contexto, esta actividad se centra en el desarrollo de una aplicación en lenguaje Swift que permita calcular áreas de diversas figuras geométricas de manera sencilla y eficiente.

La introducción de esta actividad plantea un escenario donde la necesidad de calcular áreas de figuras geométricas es común, tanto en entornos educativos como profesionales. Desde estudiantes que necesitan calcular el área de un triángulo para resolver problemas de geometría hasta profesionales que requieren determinar el área de una figura para realizar cálculos de ingeniería o arquitectura, esta actividad tiene un amplio alcance de aplicación.

El objetivo principal de esta actividad es presentar una solución práctica y funcional que permita a los usuarios calcular áreas de diferentes figuras geométricas de manera rápida y precisa. Para ello, se utilizará el lenguaje de programación Swift, ampliamente utilizado en el desarrollo de aplicaciones para dispositivos iOS.

En esta introducción, se establece el contexto y la relevancia de la actividad, así como el propósito de la solución que se desarrollará. A lo largo del desarrollo de la actividad, se explorarán los diferentes aspectos técnicos y conceptuales involucrados en la creación de la aplicación, así como las consideraciones de diseño y usabilidad que garantizarán una experiencia satisfactoria para el usuario final.

Descripción

El contexto presentado implica el desarrollo de una aplicación utilizando el lenguaje Swift, con el propósito de calcular áreas de diversas figuras geométricas. La tarea consiste en crear un menú de opciones que permita al usuario seleccionar la figura cuya área desea calcular. Las figuras incluidas son el cuadrado, el rectángulo, el triángulo y el círculo.

Para abordar esta actividad, se requerirá un conocimiento sólido de Swift y de los conceptos matemáticos asociados con el cálculo de áreas para cada una de estas figuras. Además, será necesario diseñar una interfaz de usuario intuitiva que presente al usuario las opciones disponibles de manera clara y accesible.

La implementación de la aplicación implicará la creación de funciones específicas para calcular el área de cada figura geométrica. Por ejemplo, para calcular el área del cuadrado, se necesitará una función que tome la longitud del lado como entrada y devuelva el área calculada. De manera similar, para el rectángulo se requerirá una función que tome la longitud y el ancho como parámetros. Además, la aplicación deberá gestionar la entrada del usuario de manera robusta, validando que los datos proporcionados sean adecuados para el cálculo del área de cada figura. Por ejemplo, se deben evitar entradas negativas o nulas que puedan producir resultados incorrectos o errores en el programa.

El desarrollo de esta aplicación requerirá habilidades de programación en Swift, así como un entendimiento profundo de los conceptos matemáticos involucrados en el cálculo de áreas de figuras geométricas. La capacidad para diseñar una interfaz de usuario amigable y la habilidad para manejar adecuadamente la entrada del usuario serán aspectos clave para el éxito de esta tarea.

Justificación

La creación de una aplicación en lenguaje Swift para calcular áreas de figuras geométricas ofrece numerosas ventajas que justifican su empleo en la actividad presentada. Una aplicación de este tipo proporciona una solución rápida y eficiente para calcular áreas de figuras geométricas. Al ofrecer un menú de opciones con las diferentes figuras disponibles, los usuarios pueden seleccionar la que deseen calcular sin necesidad de realizar cálculos manuales tediosos. Esto ahorra tiempo y reduce la posibilidad de errores, especialmente en casos donde las figuras son complejas o sus dimensiones son grandes.

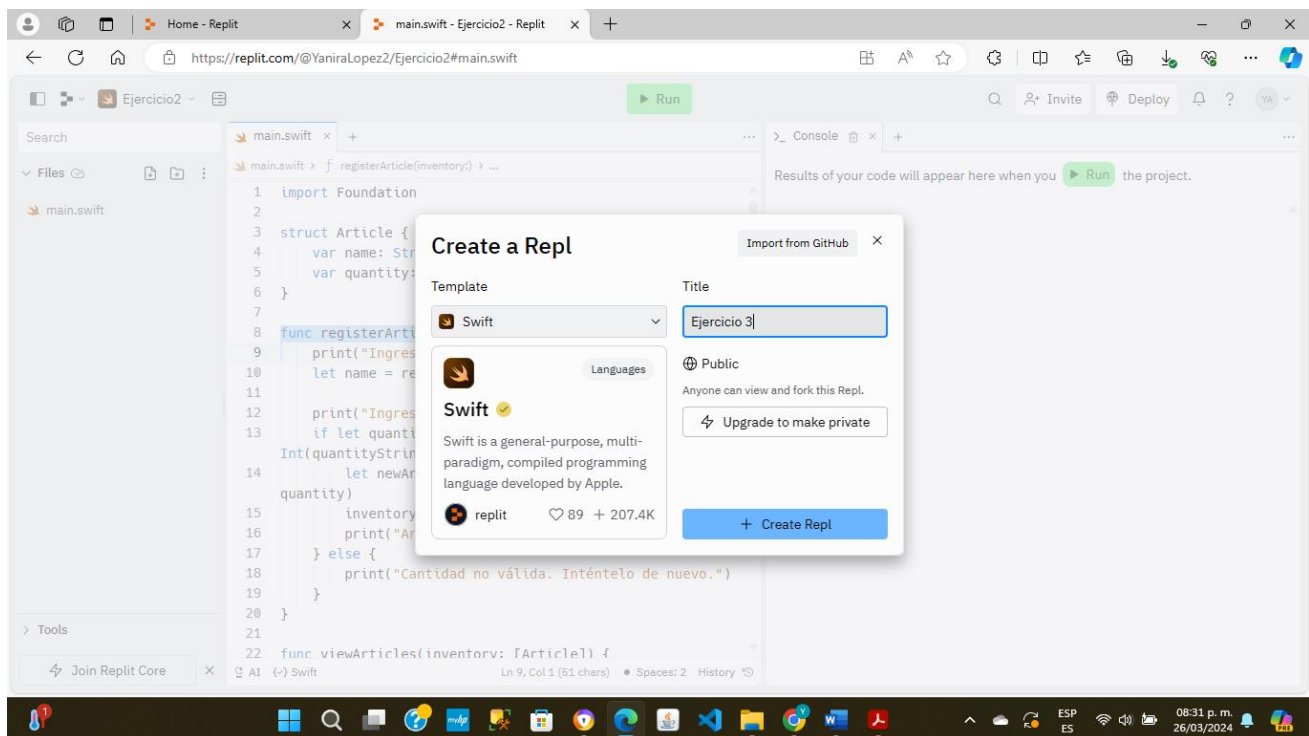
El empleo de una aplicación en lenguaje Swift permite aprovechar las capacidades de los dispositivos iOS, ofreciendo una experiencia de usuario optimizada y adaptada a las características y estándares de estos dispositivos. Esto incluye una interfaz de usuario intuitiva y fácil de usar, que guíe al usuario a través del proceso de selección de la figura y entrada de datos de manera clara y efectiva.

Otra ventaja es la versatilidad que ofrece una aplicación de este tipo. A medida que surjan nuevas necesidades o se deseen añadir funcionalidades adicionales, es relativamente sencillo modificar y actualizar la aplicación para adaptarse a estos cambios. Esto garantiza que la solución sea escalable y pueda seguir siendo útil en el futuro.

Además, el desarrollo de esta aplicación en lenguaje Swift ofrece la posibilidad de aprovechar las características y herramientas propias de este lenguaje y del entorno de desarrollo de iOS, lo que facilita el proceso de desarrollo y optimiza el rendimiento de la aplicación. El empleo de una aplicación en lenguaje Swift para calcular áreas de figuras geométricas proporciona una solución eficiente, versátil y adaptada a las necesidades y estándares de los dispositivos iOS, ofreciendo una experiencia de usuario óptima y simplificando el proceso de cálculo de áreas para los usuarios.

Desarrollo:

Una vez que disponemos de una cuenta en Replit, el siguiente paso consiste en crear un nuevo Repl. Esto lo hacemos configurando el entorno en el lenguaje de programación Swift. De esta manera, tendremos la capacidad de desarrollar nuestro código conforme a los requisitos establecidos en la actividad.



Codificación

Esta función calcula el área de un cuadrado dado el valor de un lado.

```
import Foundation
func calculateSquareArea (side: Double) -> Double {
    return side * side
}
```

Esta función calcula el área de un rectángulo dados el largo y el ancho.

```
func calculateRectangleArea (length: Double, width: Double) -> Double {
    return length * width
}
```

Esta función calcula el área de un triángulo dados la base y la altura.

```
func calculateTriangleArea (base: Double, height: Double) -> Double {
    return 0.5 * base * height
}
```

Esta función calcula el área de un círculo dado el radio.

```
func calculateCircleArea (radius: Double) -> Double {
    return Double.pi * radius * radius
}
```

Esta parte del código define la función **displayMenu ()**, que se encarga de mostrar el menú de opciones al usuario.

```
func displayMenu () {
    print ("Menú de áreas")
    print ("1- Área del cuadrado")
    print ("2- Área del rectángulo")
    print ("3- Área del triángulo")
    print ("4- Área del círculo")
    print ("Por favor introduce una opción (número):", terminator: "")
}
```

La función **getInput ()** se encarga de obtener la entrada del usuario desde la consola y devuelve esta entrada como una cadena de texto (**String**). Utiliza la función **readLine ()** para leer la entrada del usuario, que espera a que el usuario introduzca texto y presione la tecla "Enter".

```
func getInput () -> String {  
    return readLine () ?? ""  
}
```

La función **main ()** controla el flujo principal del programa, mostrando el menú de opciones y procesando la entrada del usuario. Utiliza un bucle **while** para mantenerse en ejecución hasta que el usuario decide salir. Dependiendo de la opción seleccionada, solicita los valores necesarios y calcula el área correspondiente, mostrando el resultado o un mensaje de error en caso de entrada inválida.

```
func main () {  
    var option: Int = 0  
  
    while option != 5 {  
        displayMenu ()  
  
        if let input = Int (getInput ()), (1...5). contains(input) {  
            option = input  
        }  
    }  
}
```

Este bloque **switch** se activa cuando el usuario selecciona la opción 1 en el menú. Primero, solicita al usuario que ingrese el valor del lado del cuadrado. Luego, intenta convertir la entrada del usuario en un número decimal (**Double**). Si la conversión tiene éxito, calcula el área del cuadrado utilizando la función **calculateSquareArea ()** con el lado proporcionado y muestra el resultado en la consola. Si la entrada no es válida (por ejemplo, si el usuario ingresa texto en lugar de un número), imprime un mensaje de "Valor inválido."

```
switch option {  
case 1:  
    print ("Introduce el valor del lado:", terminator: " ")  
    if let side = Double (getInput ()) {  
        let area = calculateSquareArea (side: side)  
        print ("El área del cuadrado es: \ (área)")  
    } else {  
        print ("Valor inválido.")  
    }  
}
```


Este caso se activa cuando el usuario elige la opción 2 en el menú. Pide al usuario que introduzca la longitud del rectángulo, intenta convertirlo a número. Si es válido, solicita el ancho, lo convierte y verifica su validez. Si ambos lados son válidos, calcula y muestra el área del rectángulo. Si la entrada es inválida en cualquier punto, muestra un mensaje de error.

case 2:

```
print ("Introduce el valor del lado 1:", terminator: " ")
if let length = Double(getInput ()) {
    print("Introduce el valor del lado 2:", terminator: " ")
    if let width = Double (getInput ()) {
        let area = calculateRectangleArea (length: length, width: width)
        print ("El área del rectángulo es: \"(area)\")
    } else {
        print ("Valor inválido.")
    }
} else {
    print ("Valor inválido.")
}
```

Cuando el usuario selecciona la opción 3, se le pide ingresar la base del triángulo. Si el valor ingresado es válido, se solicita la altura. Si ambos valores son válidos, se calcula y muestra el área del triángulo. Si la entrada es inválida en algún punto, se muestra un mensaje de error.

case 3:

```
print ("Introduce el valor de la base:", terminator: " ")
if let base = Double (getInput ()) {
    print ("Introduce el valor de la altura:", terminator: " ")
    if let height = Double (getInput ()) {
        let area = calculateTriangleArea (base: base, height: height)
        print ("El área del triángulo es: \"(area)\")
    } else {
        print ("Valor inválido.")
    }
} else {
    print ("Valor inválido.")
}
```

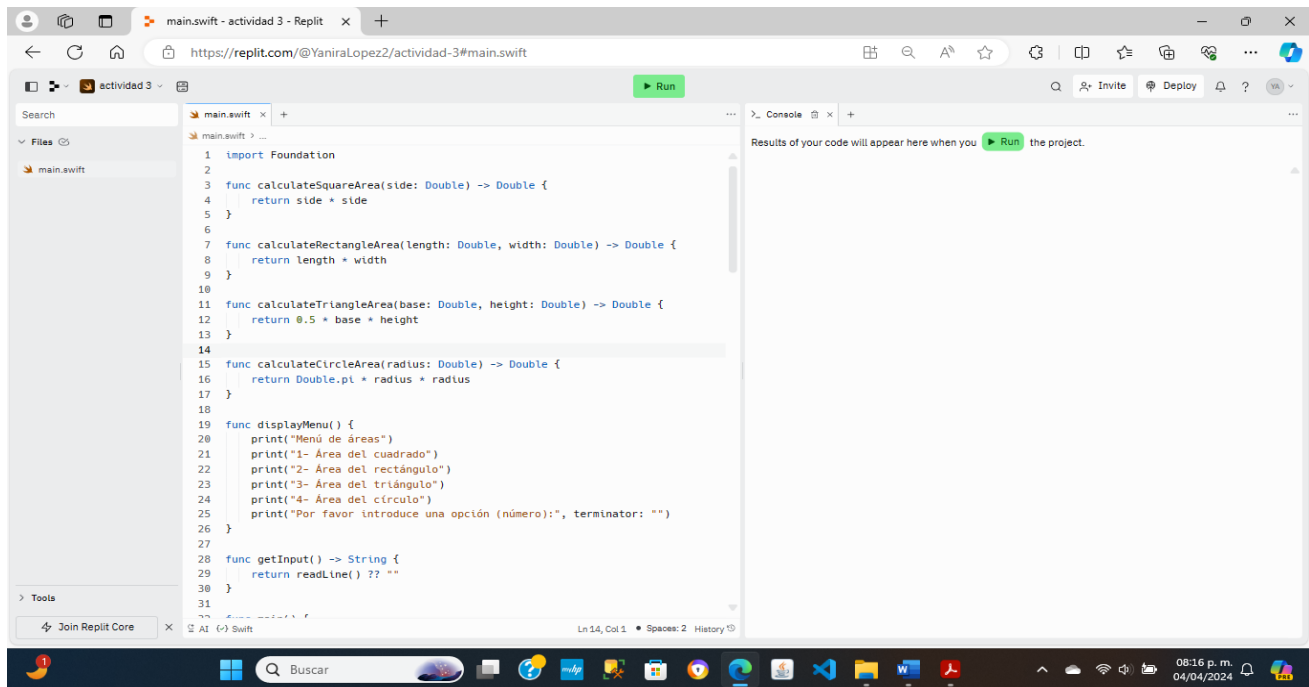
Cuando el usuario elige la opción 4, se le pide ingresar el valor del radio del círculo. Si el valor ingresado es válido, se calcula y muestra el área del círculo. Si la entrada es inválida, se muestra un mensaje de error. Si el usuario selecciona una opción no válida en el menú, se le indica que elija una opción del 1 al 5. El programa se ejecuta hasta que el usuario decide salir.

```
case 4:
    print ("Introduce el valor del radio:", terminator: " ")
    if let radius = Double (getInput ()) {
        let area = calculateCircleArea (radius: radius)
        print ("El área del círculo es: \"(area)\")
    } else {
        print ("Valor inválido.")
    }
default:
    break
}
} else {
    print ("Opción inválida. Por favor, elige una opción del 1 al 5.")
}
}
}

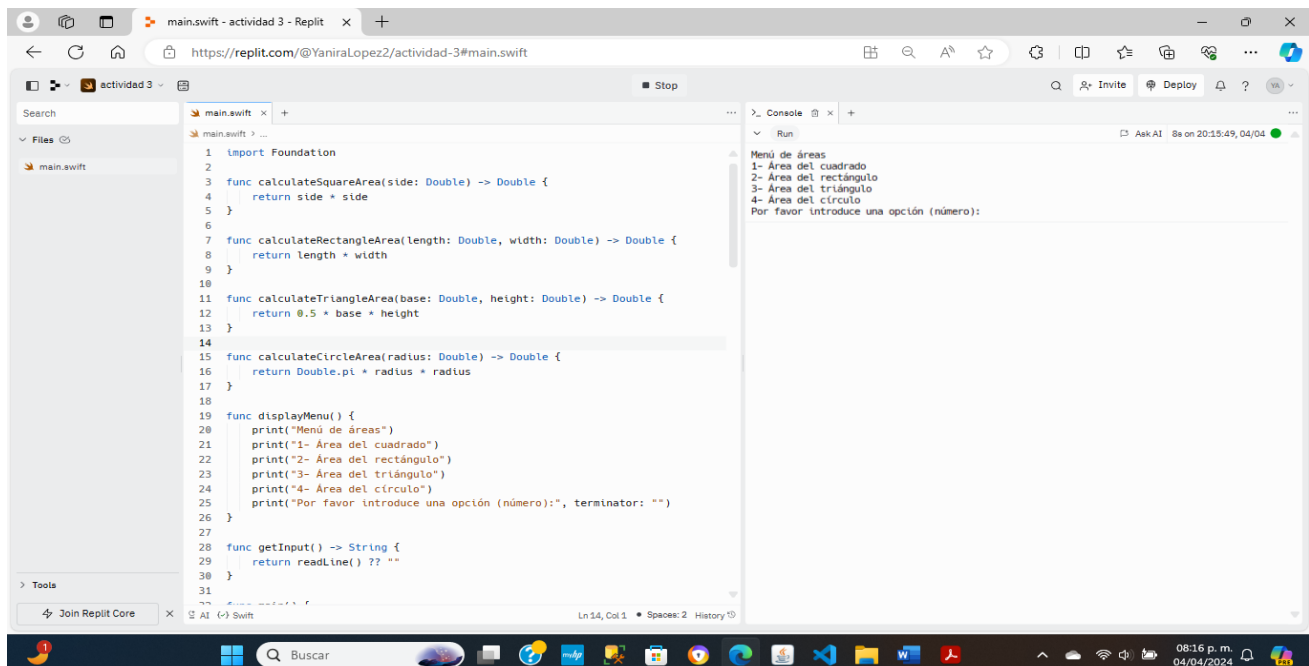
main ()
```

Prueba de la aplicación

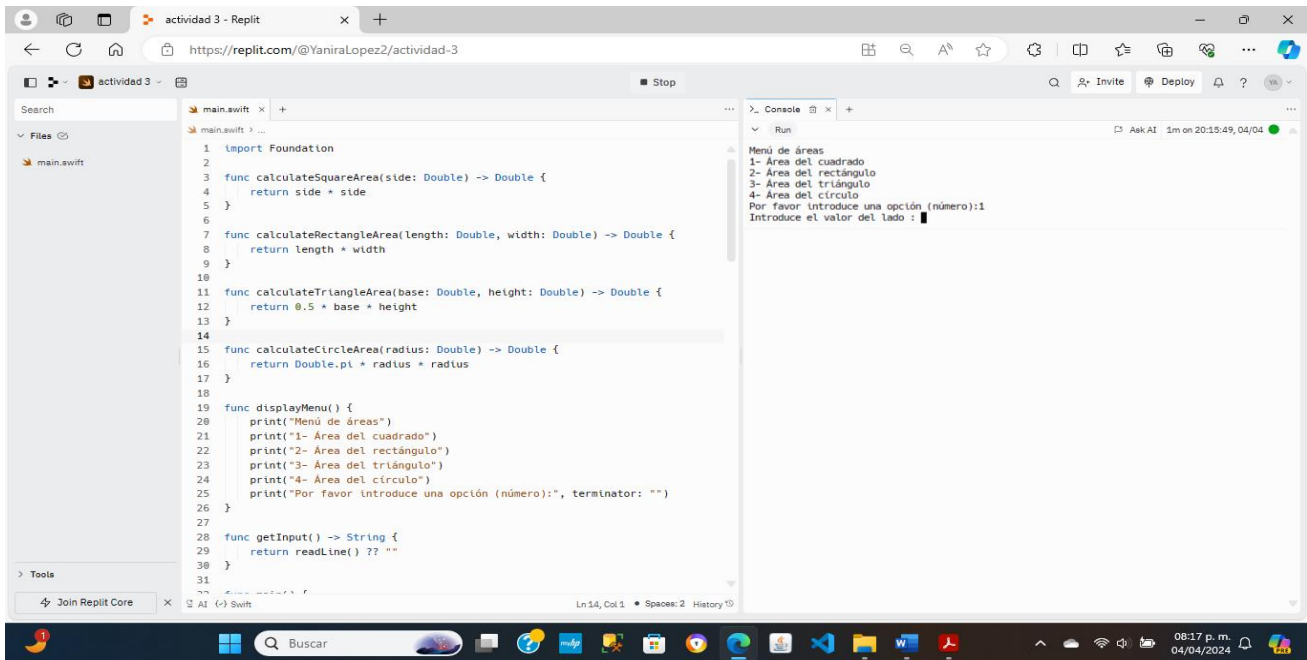
En la siguiente imagen podemos observar que una vez que ejecutamos nuestro código seleccionando la opción "Run", el resultado se muestra en la consola en forma de menú, como se ilustra claramente.



A continuación, en la siguiente imagen se muestra el **diseño** de la interfaz, así como el menú de la aplicación una vez que ya está funcionando correctamente:



En la siguiente imagen podemos observar que cuando se elige la "opción 1", se desencadena automáticamente una solicitud para que se ingrese un valor específico para el lado 1 de figura geométrica. Es decir, al seleccionar cierta opción, el sistema guía al usuario para que ingrese información relevante de manera específica.

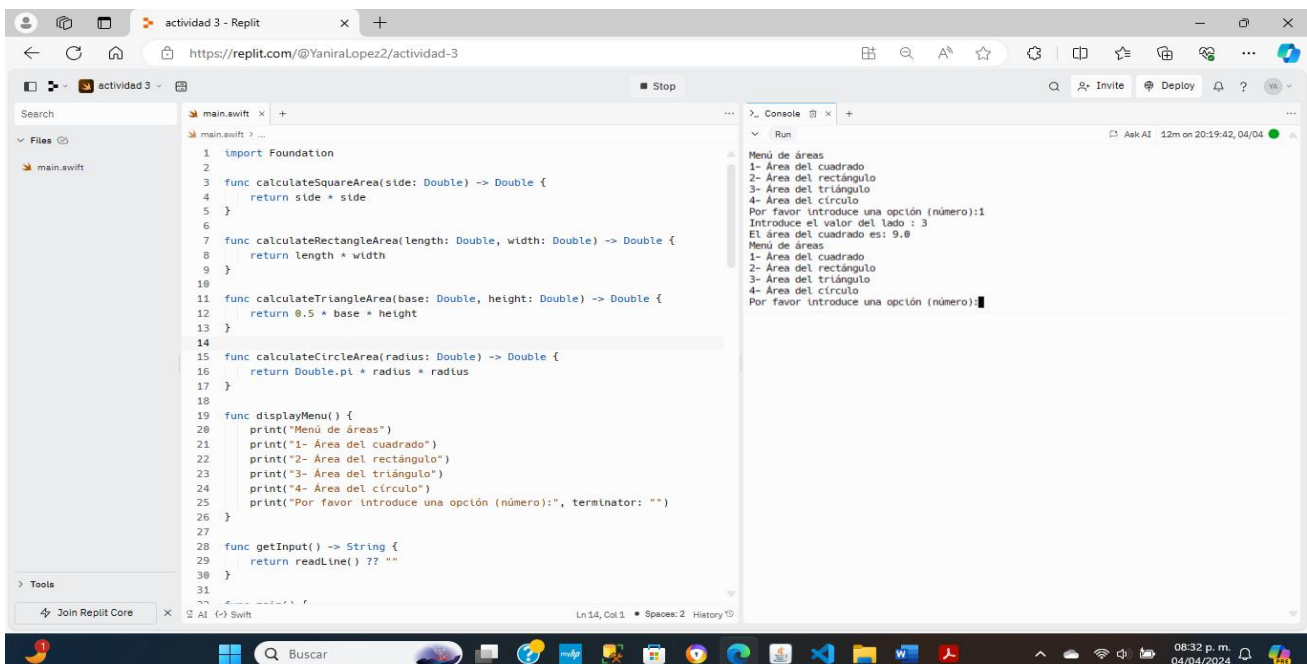


```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console Output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):1
Introduce el valor del lado :
```

Al ingresar un valor para un lado determinado, el sistema automáticamente ejecutará una operación utilizando ese valor y mostrará el resultado correspondiente. Esto sugiere un proceso automatizado en el que el sistema realiza cálculos en función de la entrada del usuario y presenta el resultado de manera inmediata.

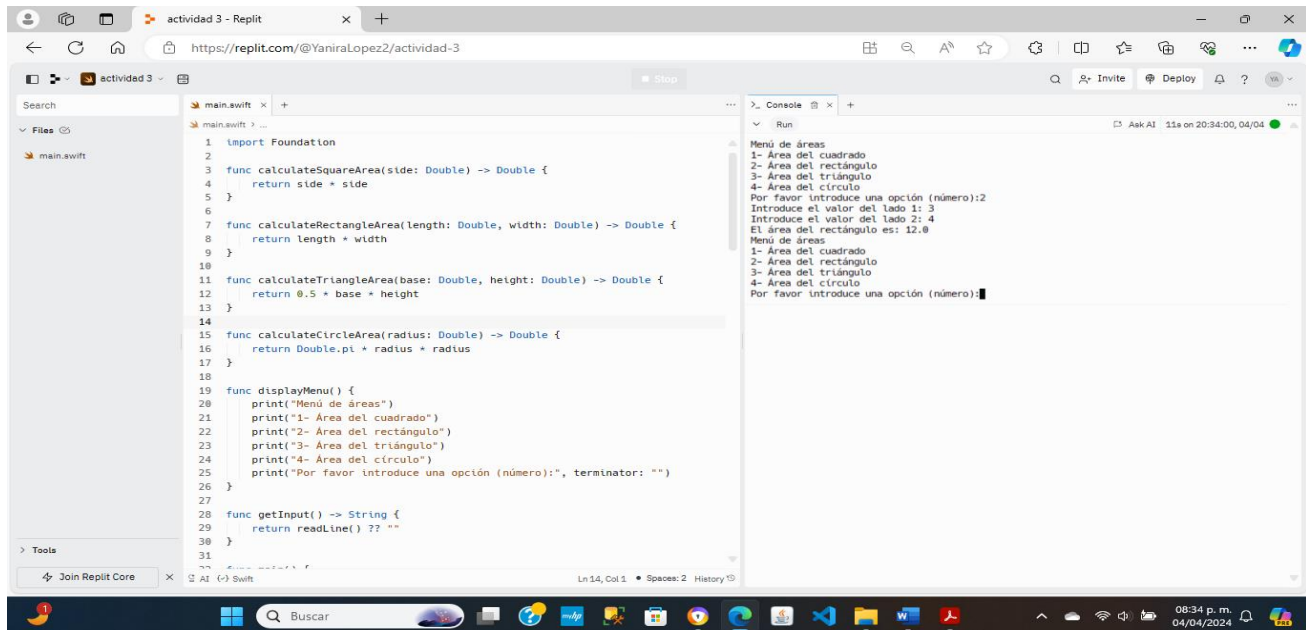


```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console Output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):1
Introduce el valor del lado : 3
El área del cuadrado es: 9.0
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):
```

Al elegir la "opción 2", se activa automáticamente una solicitud para que se ingrese un valor específico para ambos lados, de un rectángulo. Posteriormente, el sistema calcula automáticamente el área del rectángulo utilizando los valores ingresados y proporciona el resultado esperado sin intervención adicional.

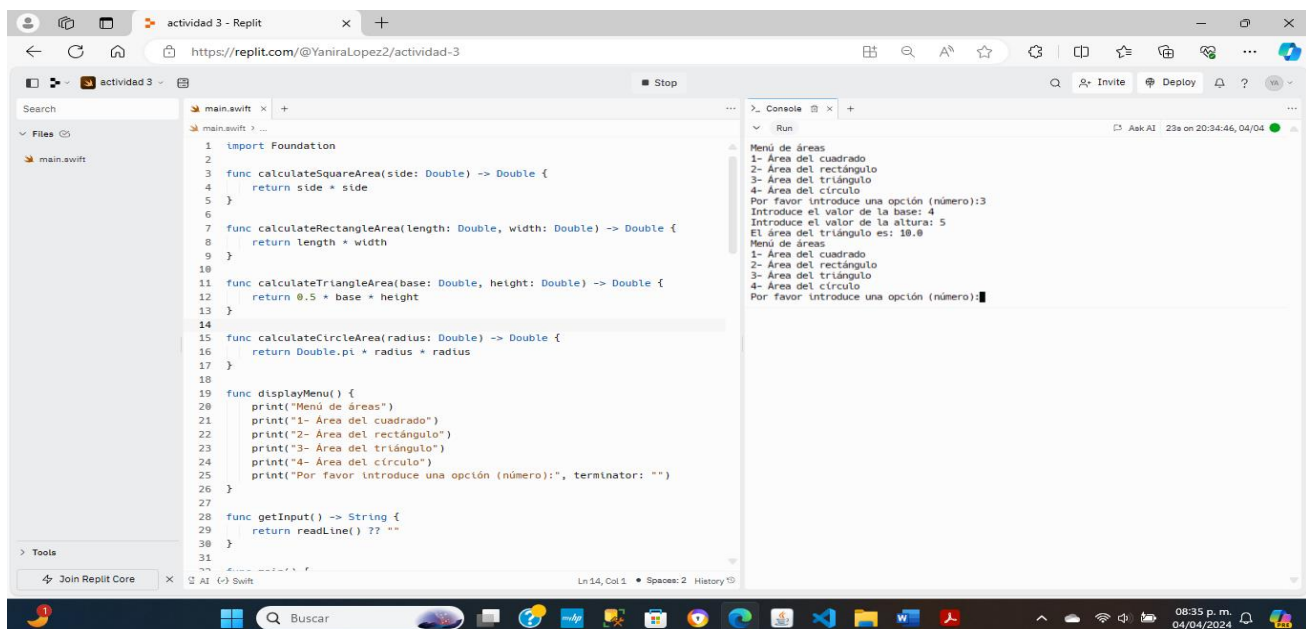


```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):2
Introduce el valor del lado 1: 3
Introduce el valor del lado 2: 4
El área del rectángulo es: 12.0
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):
```

En la siguiente imagen se muestra que al seleccionar la “opción 3”, se desencadena automáticamente una solicitud para ingresar valores específicos correspondientes a la base y altura de un triángulo. Una vez ingresados estos valores, el sistema realiza automáticamente el cálculo del área del triángulo utilizando la fórmula correspondiente y muestra el resultado esperado sin necesidad de intervención adicional por parte del usuario.

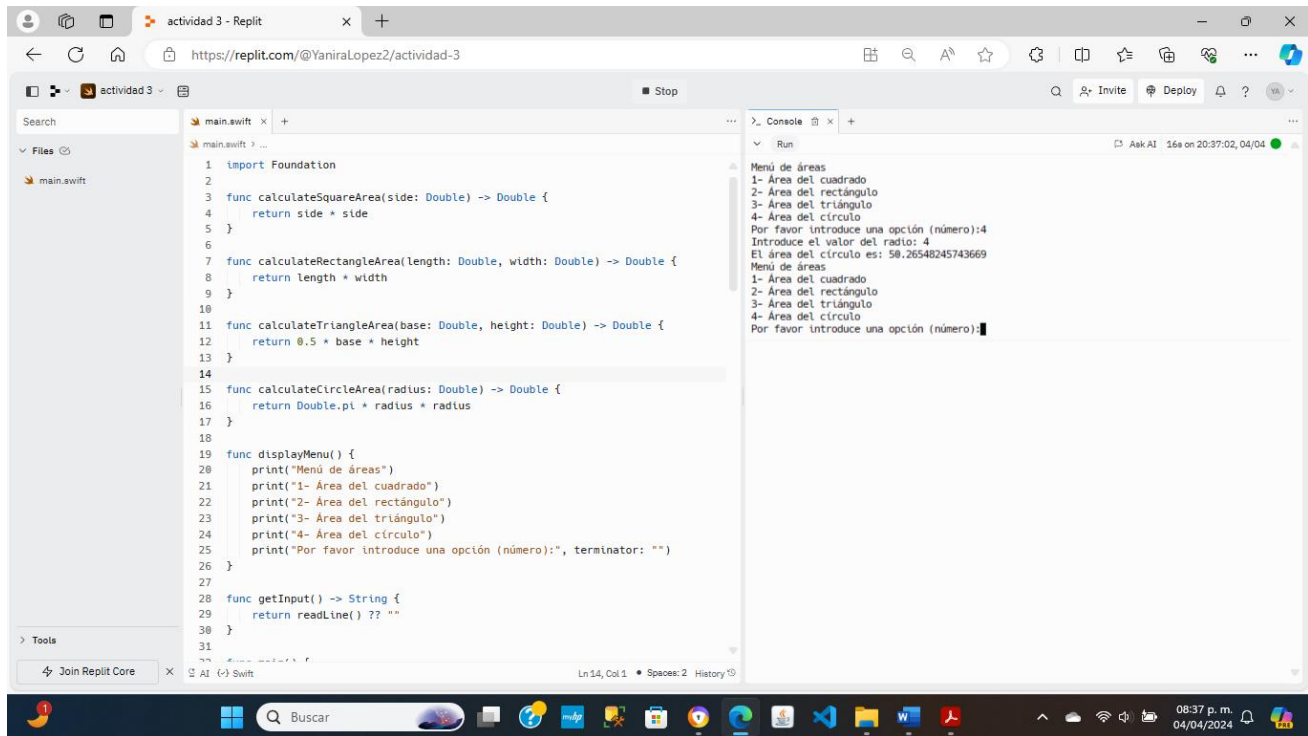


```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):3
Introduce el valor de la base: 4
Introduce el valor de la altura: 5
El área del triángulo es: 10.0
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):
```

A continuación, se muestra que al elegir la "opción 4", se activa automáticamente una solicitud para ingresar el valor del radio de un círculo. Una vez ingresado este valor, el sistema realiza automáticamente el cálculo del área del círculo y muestra el resultado.



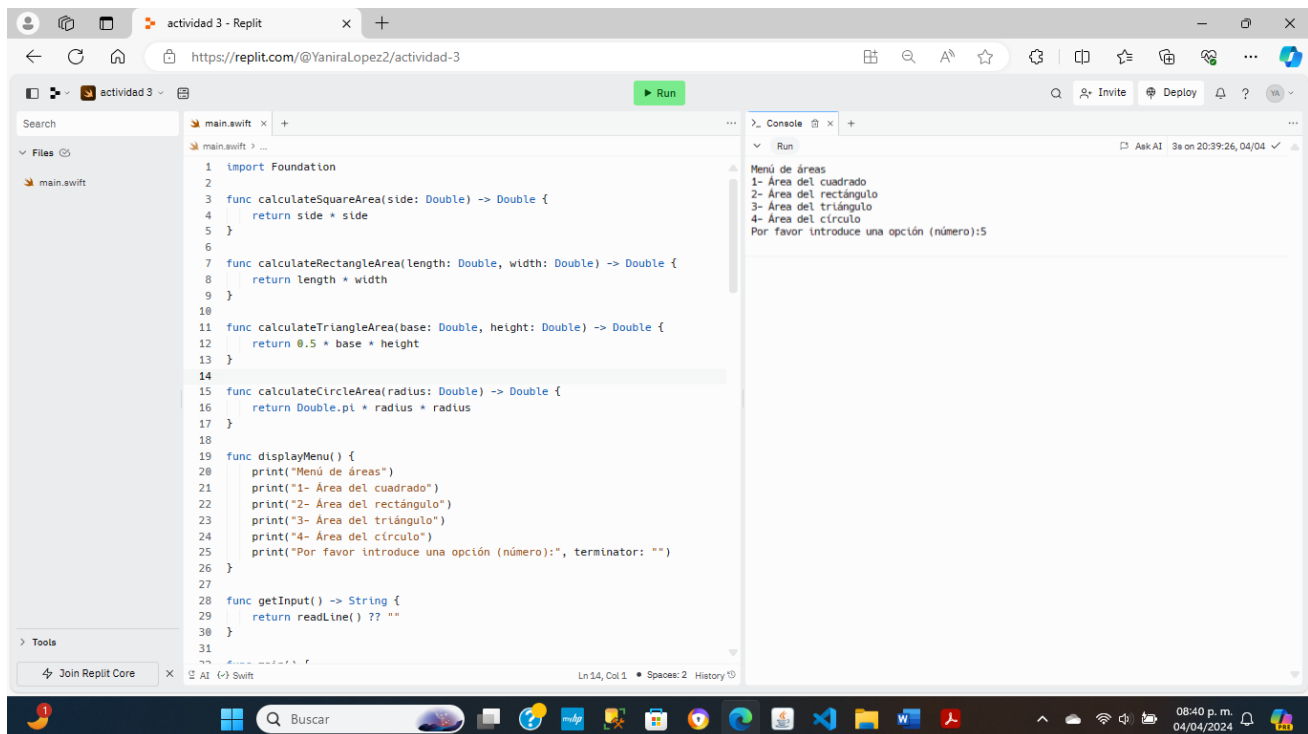
The screenshot shows a Replit environment with a Swift file named `main.swift`. The code defines functions for calculating the area of a square, rectangle, triangle, and circle, along with a menu function and an input function. The console output shows the menu, the selection of option 4, the prompt for the radius, and the calculated area of the circle.

```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):4
Introduce el valor del radio: 4
El área del círculo es: 50.26548245743669
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):
```

En la siguiente imagen podemos observar que al introducir el numero 5 como última opción no nos arroja otro resultado.



The screenshot shows the same Replit environment with the same Swift code. The console output shows the menu, the selection of option 5, and the prompt for the radius, but no further output is shown.

```
1 import Foundation
2
3 func calculateSquareArea(side: Double) -> Double {
4     return side * side
5 }
6
7 func calculateRectangleArea(length: Double, width: Double) -> Double {
8     return length * width
9 }
10
11 func calculateTriangleArea(base: Double, height: Double) -> Double {
12     return 0.5 * base * height
13 }
14
15 func calculateCircleArea(radius: Double) -> Double {
16     return Double.pi * radius * radius
17 }
18
19 func displayMenu() {
20     print("Menú de áreas")
21     print("1- Área del cuadrado")
22     print("2- Área del rectángulo")
23     print("3- Área del triángulo")
24     print("4- Área del círculo")
25     print("Por favor introduce una opción (número):", terminator: "")
26 }
27
28 func getInput() -> String {
29     return readLine() ?? ""
30 }
31
```

Console output:

```
Menú de áreas
1- Área del cuadrado
2- Área del rectángulo
3- Área del triángulo
4- Área del círculo
Por favor introduce una opción (número):5
```

Conclusión

La creación de una aplicación en lenguaje Swift para calcular áreas de figuras geométricas ofrece beneficios significativos que impactan tanto en el campo laboral como en la vida cotidiana de las personas. Esta actividad no solo proporciona una solución práctica para realizar cálculos matemáticos complejos de manera rápida y precisa, sino que también demuestra el potencial transformador de la tecnología en nuestra vida diaria.

En el ámbito laboral, una aplicación de este tipo puede ser invaluable para profesionales de diversas áreas, como arquitectos, ingenieros, diseñadores y educadores. Permitirles calcular áreas de figuras geométricas de forma eficiente puede mejorar la productividad y la precisión en sus tareas diarias. Por ejemplo, un arquitecto podría utilizar la aplicación para determinar rápidamente el área de un terreno antes de diseñar un edificio, mientras que un profesor de matemáticas podría emplearla como herramienta educativa para enseñar conceptos geométricos de manera interactiva.

En la vida cotidiana, esta aplicación también puede ser de gran utilidad. Desde ayudar a estudiantes con sus tareas escolares hasta facilitar tareas domésticas como calcular la cantidad de pintura necesaria para pintar una habitación, la capacidad de realizar cálculos de áreas con facilidad y precisión puede simplificar muchas actividades diarias.

En conclusión, la importancia de esta actividad radica en su capacidad para proporcionar una herramienta práctica y funcional que puede tener un impacto significativo tanto en el ámbito laboral como en la vida cotidiana de las personas. La aplicación desarrollada no solo simplifica el proceso de cálculo de áreas, sino que también resalta el valor de la tecnología como facilitador de tareas y mejorador de la eficiencia en múltiples contextos.

Referencias

Ingeniería en desarrollo de software. Universidad México Internacional. Recuperado el día 28 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.bDya6X4IzgwjkaL?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2FAJ8WceZxxDx59vIkFrGFwchemU4pY7gSXb4X7zcMWAIInWas8y00-m9IhuXLwALGX.mZCT1fbTEoKhl-uQ

Video conferencing, web conferencing, webinars, screen sharing. (s. f.-b). Zoom.

https://academiaglobal-mx.zoom.us/rec/play/Px1ih2hLXYkCMi-uBGgmMhoehG_v6PJvzo7xpbVQbuJcZ4NaB6Kwet5WPiAjlOGelY_69zUHHnrX7gcU.wUW9px8gqL4IKRd?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2FAafO4CnpSamMq-FFtHBOiQofV73hUbrRBPtvcRxgnHfvNjSqkqYTcCongUEN7xP.c-P2Gk3dYBWkWm34

Array | Apple Developer Documentation. (s. f.). Apple Developer Documentation.

<https://developer.apple.com/documentation/swift/array>

Structures and Classes — The Swift Programming Language en Español. (s. f.). The Swift Programming Language En Español. <https://swift-book-es.vercel.app/guia-del-lenguaje/estructuras-y-clases>

Video conferencing, web conferencing, webinars, screen sharing. (s. f.-c). Zoom.

https://academiaglobal-mx.zoom.us/rec/play/chRvnFVi7Elr64cItL79zRUuGqj4xuT9U-3nB5qwm4d4XUpS0Jpz7ZjQp-XjWhH-v9NBydtRBoBLpZkv.AUkKD-A7mh21xLJv?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2F73IqdC-I80FHm7wNCZsNNbIVfFba1S74NLWS7x2-z7Y3nxtBzKifXuRZMbYH99as.WfMhdN3zhNM