

Actividad 1 - Instalación Xcode / Compilador Online / Tienda de ropa

Desarrollo de aplicaciones móviles IV

Ingeniería en Desarrollo de Software

Tutor: Marco Alonso Rodríguez Tapia

Alumno: Fernando Pedraza Garate

Fecha: 27 de noviembre del 2024

Índice

Etapa 1 – Instalación de Xcode / Compilador Online.

○ Introducción.	Pág. 3 - 4
○ Descripción	Pág. 5
○ Justificación	Pág. 6
○ Desarrollo	Pág. 7 - 13
• Codificación	
• Prueba del programa	
○ Conclusión	Pág. 14
○ Referencias bibliográficas.	Pág. 15

Introducción

Xcode es el entorno de desarrollo integrado (IDE) oficial de Apple, una herramienta poderosa que incluye todo lo necesario para programar, diseñar, probar y depurar aplicaciones para sus distintas plataformas, como **iOS**, **macOS**, **watchOS** y **tvOS**, algunas de sus características es que es un editor de código que admite lenguajes como Swift, Objective-C y C++, ofreciendo funciones como autocompletado de código, resaltado de sintaxis y navegación entre archivos, es un diseñador de interfaces gráficas (Interface Builder) que Permite crear interfaces de usuario para aplicaciones mediante un enfoque visual, con soporte para el diseño adaptativo, con simuladores para poder probar aplicaciones en diferentes dispositivos de Apple sin necesidad de usar un hardware físico, con herramientas avanzadas para poder identificar errores en el código y monitorear el rendimiento, facilitando la organización y configuración de proyectos, con opciones para gestionar dependencias y configuraciones de compilación, e incluye herramientas como XCTest para escribir pruebas automatizadas e Instruments para analizar el rendimiento y funciona únicamente en computadoras con macOS.

Si se cuenta con equipo con sistema operativo Windows, se deberá utilizar un compilador online, herramienta basada en la web que permite escribir, compilar y ejecutar código directamente en un navegador web, sin necesidad de instalar un entorno de desarrollo o un compilador en el equipo a utilizar.

Algunos ejemplos de compiladores online populares son:

- **Replit** (replit.com): Soporta múltiples lenguajes y proyectos colaborativos.
- **JDoodle** (jdoodle.com): Ligero y fácil de usar para varios lenguajes.
- **Compiler Explorer** (godbolt.org): Excelente para explorar el funcionamiento interno de compiladores como GCC y Clang.
- **Programiz** (programiz.com): Focalizado en enseñar programación con soporte para Python, C, y más.

Descripción.

La boutique Norma requiere desarrollar una aplicación que funcione como una tienda de ropa en línea, donde los clientes puedan visualizar productos y realizar compras directamente desde la app, este proyecto debe implementarse utilizando el lenguaje de programación Swift.

Se deberá instalar Xcode como entorno de desarrollo o utilizar un compilador en línea compatible con Swift y programar la aplicación siguiendo las especificaciones indicadas, para los requisitos del programa que deberá poder mostrar 4 artículos disponibles, incluyendo su nombre, precio y cantidad en stock e incluir un menú interactivo para que el cliente pueda:

- Seleccionar un artículo para comprar.
- Salir de la aplicación.

Y al finalizar una compra, el programa debe mostrar el nombre del artículo adquirido y el monto total a pagar, con el objetivo de crear una experiencia sencilla y funcional que simule una tienda en línea básica para mejorar la interacción cliente-producto y facilitar la compra directa desde la app.

Justificación.

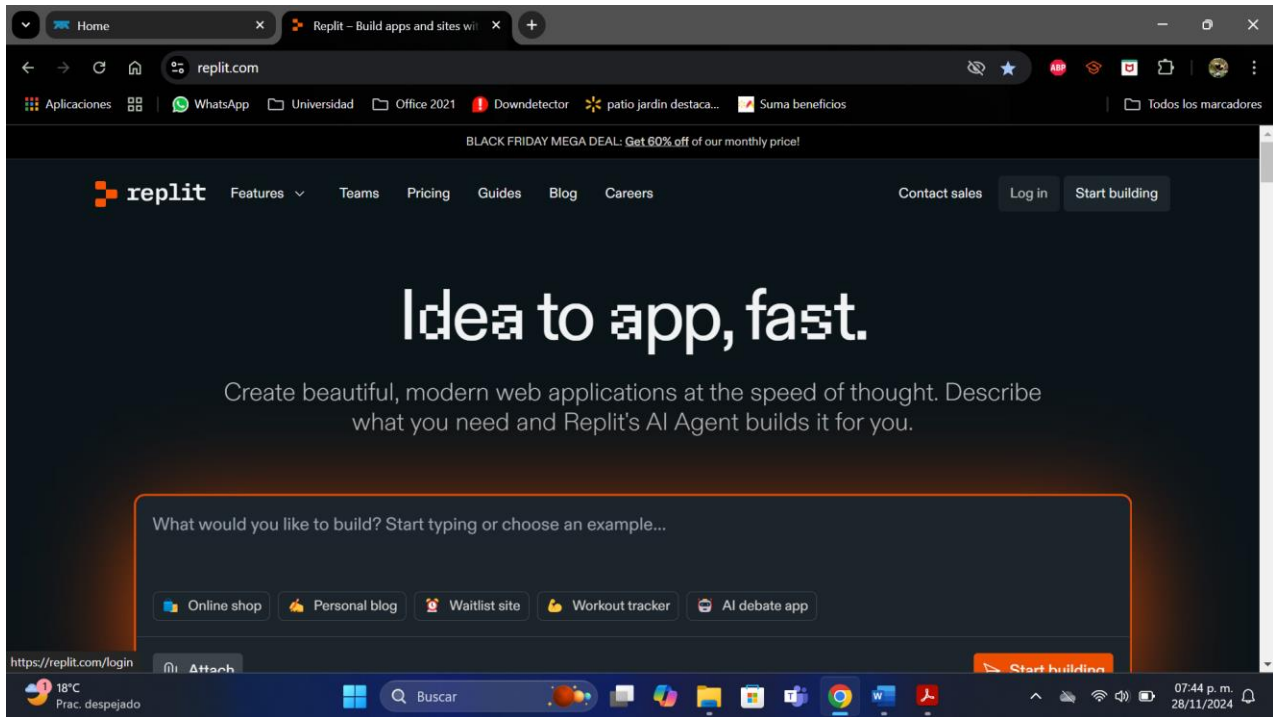
El usar Xcode o un compilador en línea depende de las necesidades y el contexto del proyecto, en el caso de Xcode por ser un entorno integrado profesional y ser el entorno oficial de Apple para desarrollar aplicaciones en Swift, que ofrece herramientas avanzadas como un simulador de dispositivos, autocompletado, depuración y diseño de interfaces gráficas, permitiendo desarrollar, probar y optimizar aplicaciones para iOS, iPadOS, macOS y más, asegurando un rendimiento óptimo en los dispositivos Apple, ayudando a los programadores a familiarizarse con herramientas profesionales utilizadas en la industria, lo que mejora sus habilidades prácticas.

Y el uso de un compilador en línea ofrece acceso inmediato, al no requerir instalación ni configuraciones avanzadas, ideal para usuarios que no disponen de macOS o equipos potentes, es útil para proyectos pequeños o educativos, enfocándose en la lógica y el código sin distracciones de herramientas avanzadas, ideal para aprender Swift sin necesidad de adquirir un dispositivo Apple.

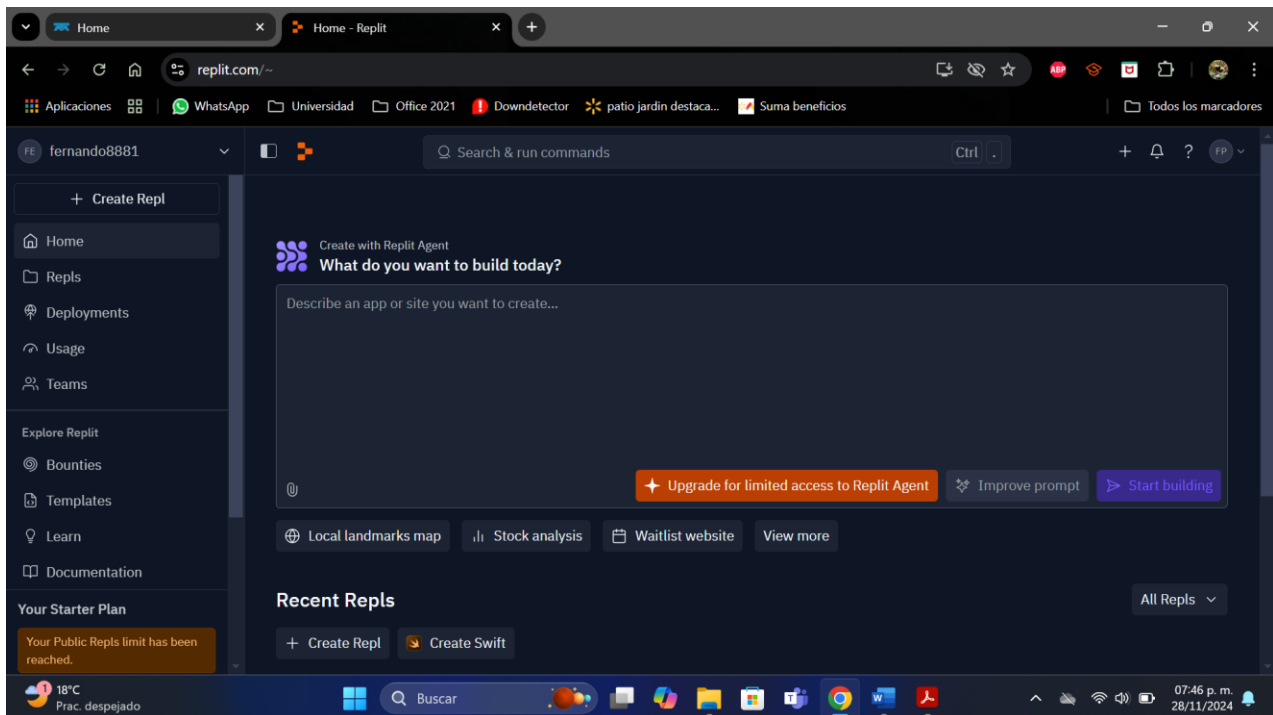
Ambas opciones son válidas según los recursos disponibles y la complejidad del proyecto. Xcode es ideal para producción, mientras que un compilador en línea es más práctico para aprendizaje rápido o demostraciones.

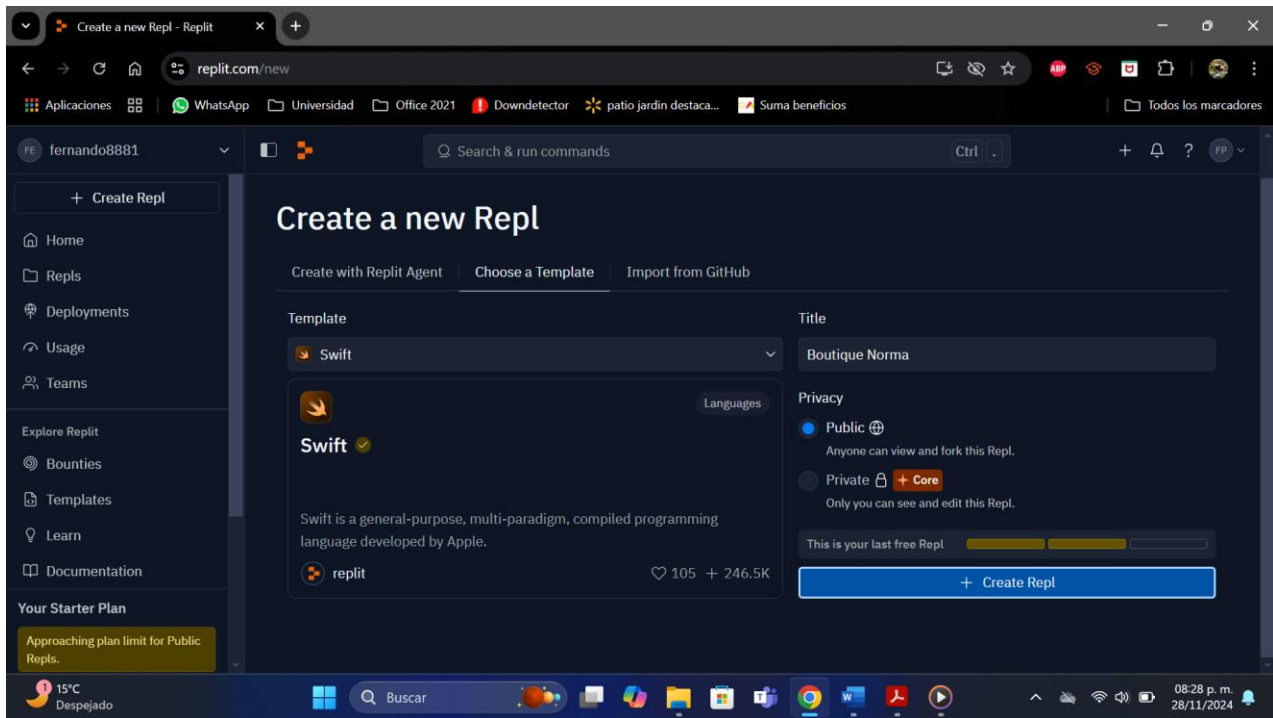
Desarrollo.

Codificación / compilador online

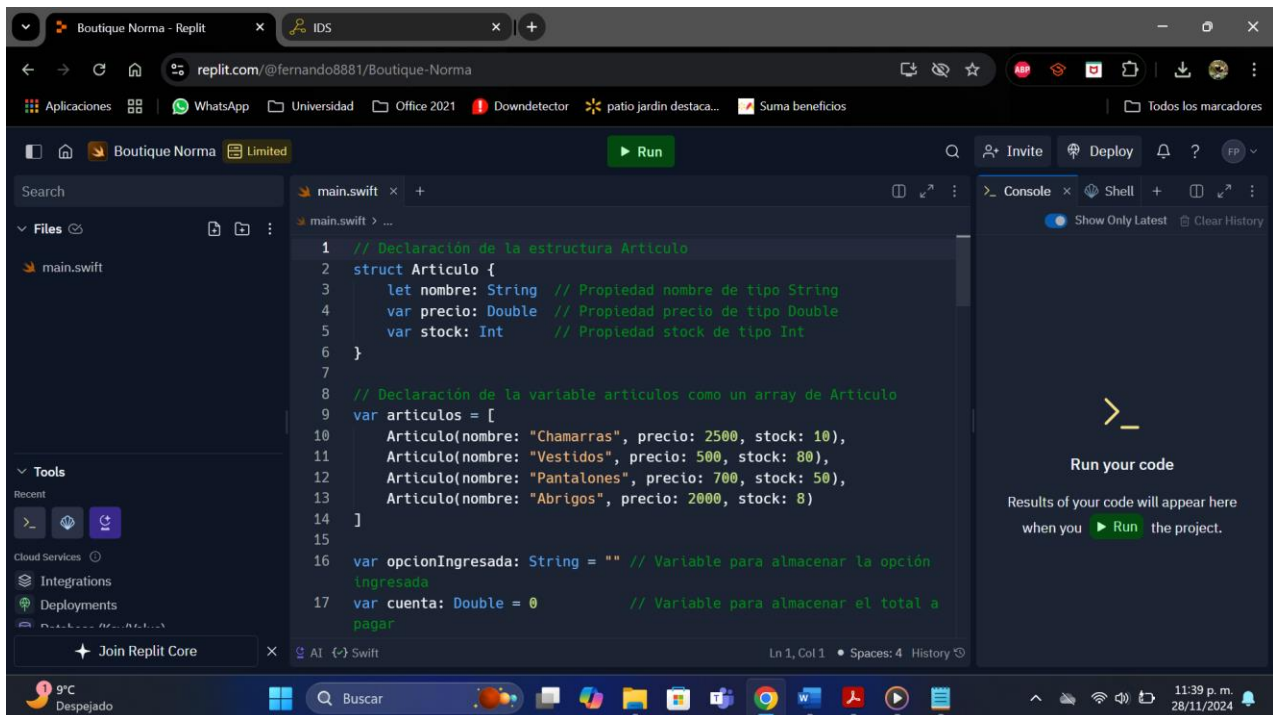


Se ingresa a la página de Replit

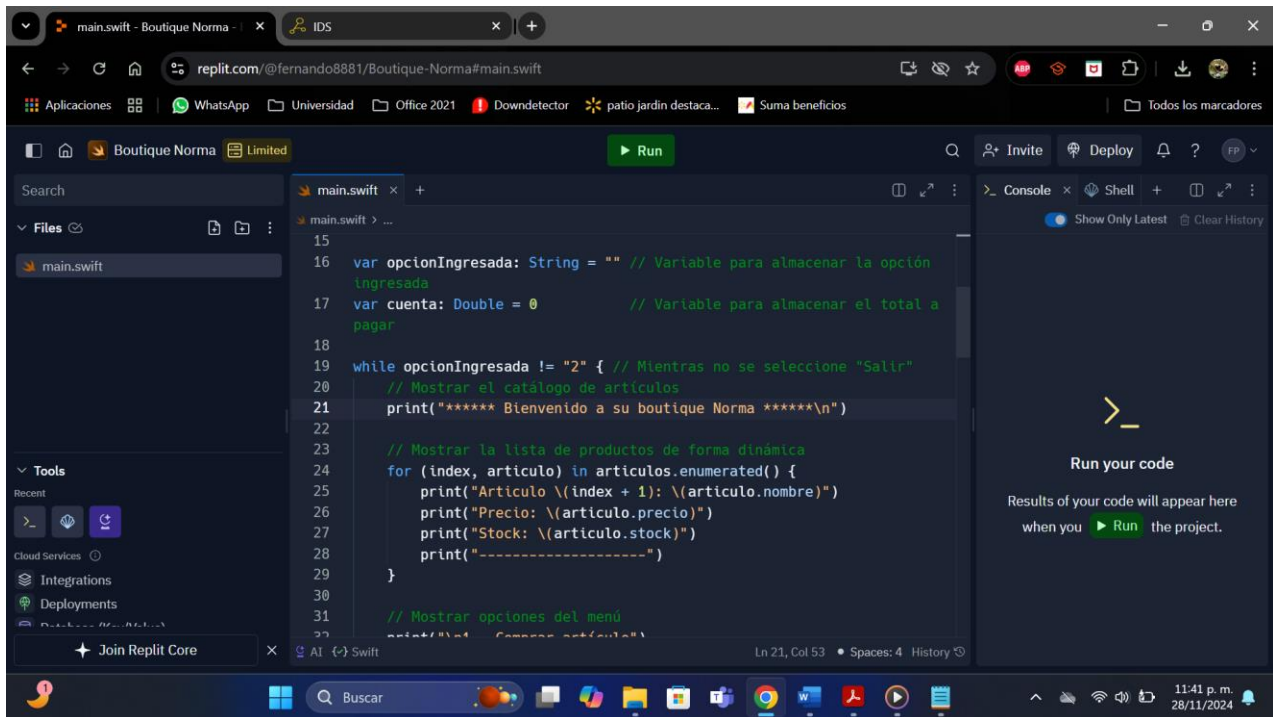




Se crea el nuevo Repl con el nombre de Boutique Norma

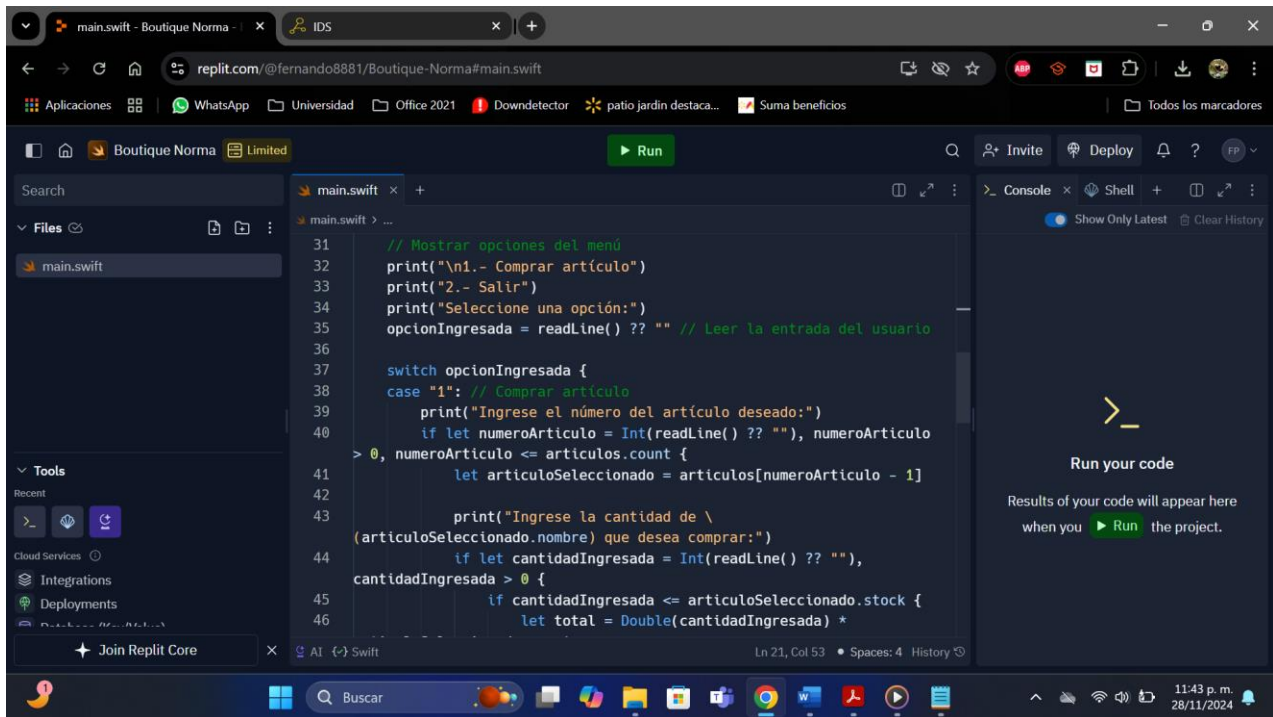


Se declara la estructura de los artículos y las variables como un arreglo



```
15
16 var opcionIngresada: String = "" // Variable para almacenar la opción
17   ingresada
18 var cuenta: Double = 0 // Variable para almacenar el total a
19   pagar
20 while opcionIngresada != "2" { // Mientras no se seleccione "Salir"
21   // Mostrar el catálogo de artículos
22   print("***** Bienvenido a su boutique Norma *****\n")
23
24   // Mostrar la lista de productos de forma dinámica
25   for (index, articulo) in articulos.enumerated() {
26     print("Artículo \(index + 1): \(articulo.nombre)")
27     print("Precio: \(articulo.precio)")
28     print("Stock: \(articulo.stock)")
29     print("-----")
30   }
31
32   // Mostrar opciones del menú
33   print("\n1.- Comprar artículo")
34   print("2.- Salir")
35   print("Seleccione una opción:")
36   opcionIngresada = readLine() ?? "" // Leer la entrada del usuario
37
38   switch opcionIngresada {
39     case "1": // Comprar artículo
40       print("Ingrese el número del artículo deseado:")
41       if let numeroArticulo = Int(readLine() ?? ""), numeroArticulo
42         > 0, numeroArticulo <= articulos.count {
43         let articuloSeleccionado = articulos[numeroArticulo - 1]
44         print("Ingrese la cantidad de \(
45           articuloSeleccionado.nombre) que desea comprar:")
46         if let cantidadIngresada = Int(readLine() ?? ""),
47           cantidadIngresada > 0 {
48           if cantidadIngresada <= articuloSeleccionado.stock {
49             let total = Double(cantidadIngresada) *
```

Se crean las variables para los datos ingresados por el usuario y el ciclo while para continuar dentro del programa mientras no se seleccione la opción salir, mostrando las opciones del menú.



```
31 // Mostrar opciones del menú
32 print("\n1.- Comprar artículo")
33 print("2.- Salir")
34 print("Seleccione una opción:")
35 opcionIngresada = readLine() ?? "" // Leer la entrada del usuario
36
37 switch opcionIngresada {
38 case "1": // Comprar artículo
39   print("Ingrese el número del artículo deseado:")
40   if let numeroArticulo = Int(readLine() ?? ""), numeroArticulo
41     > 0, numeroArticulo <= articulos.count {
42     let articuloSeleccionado = articulos[numeroArticulo - 1]
43     print("Ingrese la cantidad de \(
44       articuloSeleccionado.nombre) que desea comprar:")
45     if let cantidadIngresada = Int(readLine() ?? ""),
46       cantidadIngresada > 0 {
47       if cantidadIngresada <= articuloSeleccionado.stock {
48         let total = Double(cantidadIngresada) *
```

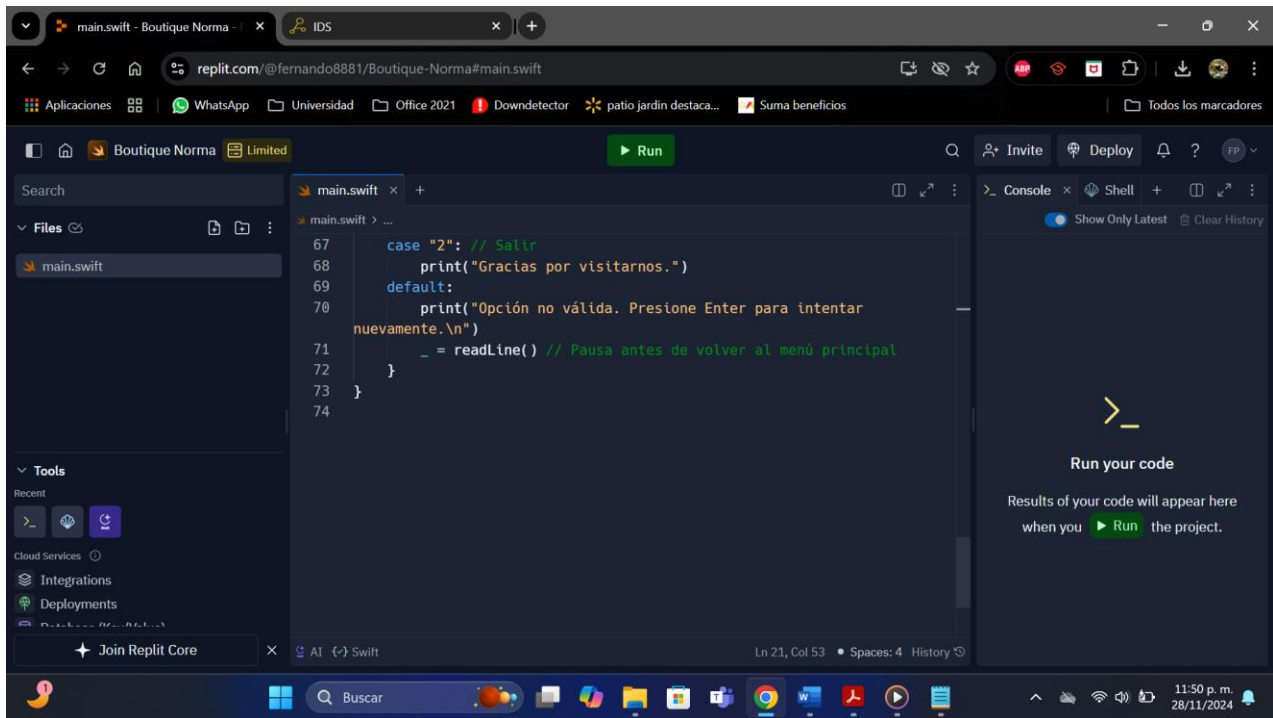
Creando el switch para que el programa ejecute una acción dependiendo el caso establecido.

```
42
43     print("Ingrese la cantidad de \
(articuloSeleccionado.nombre) que desea comprar:")
44     if let cantidadIngresada = Int(readLine() ?? ""),
cantidadIngresada > 0 {
45         if cantidadIngresada <= articuloSeleccionado.stock {
46             let total = Double(cantidadIngresada) *
articuloSeleccionado.precio
47             cuenta += total
48             articulos[numeroArticulo - 1].stock -=
cantidadIngresada
49             print("Compra realizada exitosamente.")
50             print("Total a pagar por esta compra: $\(total)")
51             print("Gracias por su compra. Le esperamos
nuevamente.\n")
52             print("Presione Enter para regresar al menú
principal.")
53             _ = readLine() // Pausa antes de volver al menú
principal
54         } else {
```

Imprimiendo en pantalla el contexto establecido según la actividad presentada, haciendo pausas antes de mostrar el menú principal.

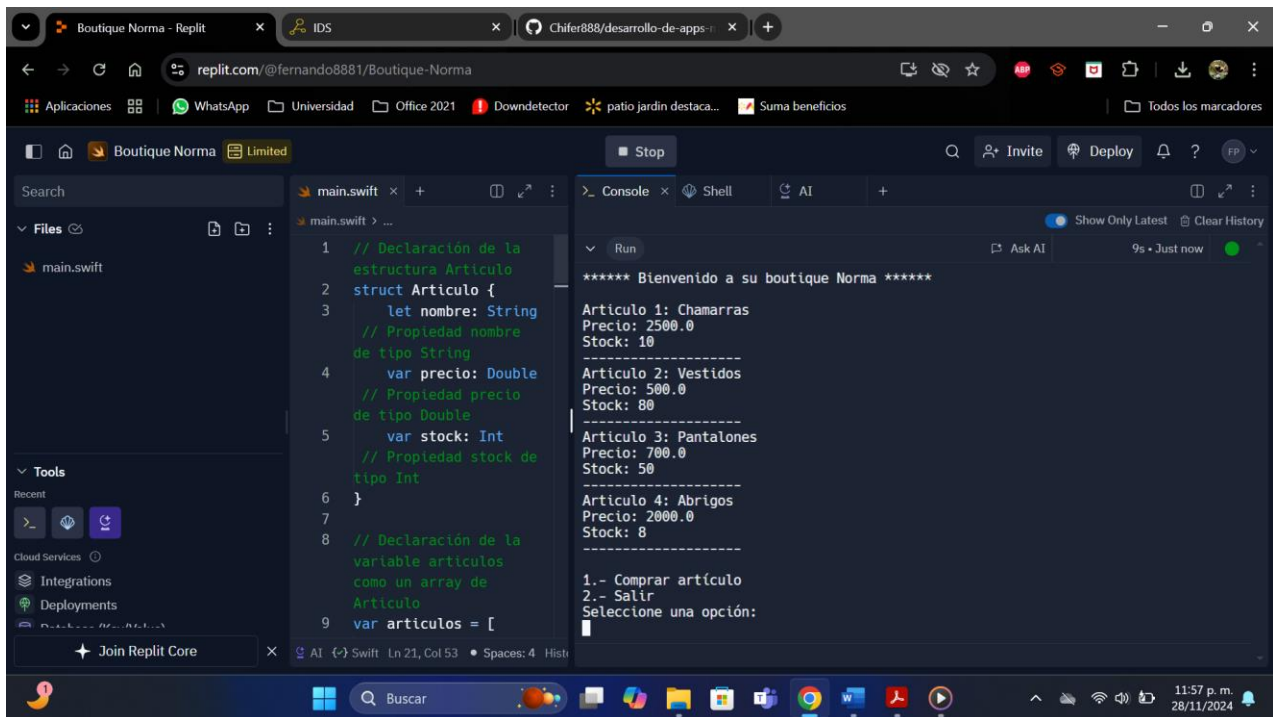
```
54         } else {
55             print("Lo sentimos, no hay suficiente stock para
su pedido.\n")
56             print("Presione Enter para regresar al menú
principal.")
57             _ = readLine() // Pausa antes de volver al menú
principal
58         }
59     } else {
60         print("Cantidad no válida. Presione Enter para
intentar nuevamente.\n")
61         _ = readLine() // Pausa antes de volver al menú
principal
62     }
63     } else {
64         print("Artículo no válido. Presione Enter para intentar
nuevamente.\n")
65         _ = readLine() // Pausa antes de volver al menú principal
66     }
```

Creando un else para que se muestran advertencias en caso de ingresar algún dato erróneo.



Y para que el programa se detenga al seleccionar la opción salir.

Prueba del programa



Programa iniciado

```
1 // Declaración de la
2 struct Artículo {
3     let nombre: String
4     // Propiedad nombre
5     de tipo String
6     var precio: Double
7     // Propiedad precio
8     de tipo Double
9     var stock: Int
10    // Propiedad stock de
11    tipo Int
12 }
13 // Declaración de la
14 variable articulos
15 como un array de
16 Artículo
17 var articulos = [
```

***** Bienvenido a su boutique Norma *****

Artículo 1: Chamarras
Precio: 2500.0
Stock: 10

Artículo 2: Vestidos
Precio: 500.0
Stock: 80

Artículo 3: Pantalones
Precio: 700.0
Stock: 50

Artículo 4: Abrigos
Precio: 2000.0
Stock: 8

1.- Comprar artículo
2.- Salir
Seleccione una opción:
5
Opción no válida. Presione Enter para intentar nuevamente.

Al ingresar erróneamente una opción manda la observación y solicita dar enter para continuar.

```
1 // Declaración de la
2 struct Artículo {
3     let nombre: String
4     // Propiedad nombre
5     de tipo String
6     var precio: Double
7     // Propiedad precio
8     de tipo Double
9     var stock: Int
10    // Propiedad stock de
11    tipo Int
12 }
13 // Declaración de la
14 variable articulos
15 como un array de
16 Artículo
17 var articulos = [
```

Stock: 80

Artículo 3: Pantalones
Precio: 700.0
Stock: 50

Artículo 4: Abrigos
Precio: 2000.0
Stock: 8

1.- Comprar artículo
2.- Salir
Seleccione una opción:
1
Ingrese el número del artículo deseado:
4
Ingrese la cantidad de Abrigos que desea comprar:
5
Compra realizada exitosamente.
Total a pagar por esta compra: \$10000.0
Gracias por su compra. Le esperamos nuevamente.
Presione Enter para regresar al menú principal.

Selección correcta del menú, del articulo y de la cantidad a comprar, mostrando el monto total a pagar y solicitando dar enter para regresar al menú principal.


```
1 // Declaración de la
2 struct Articulo {
3     let nombre: String
4     // Propiedad nombre
5     de tipo String
6     var precio: Double
7     // Propiedad precio
8     de tipo Double
9     var stock: Int
10    // Propiedad stock de
11    tipo Int
12 }
13 // Declaración de la
14 variable articulos
15 como un array de
16 Articulo
17 var articulos = [
```

Run

Artículo 2: Vestidos
Precio: 500.0
Stock: 80

Artículo 3: Pantalones
Precio: 700.0
Stock: 50

Artículo 4: Abrigos
Precio: 2000.0
Stock: 3

1.- Comprar artículo
2.- Salir
Seleccione una opción:
1
Ingrese el número del artículo deseado:
3
Ingrese la cantidad de Pantalones que desea comprar:
900
Lo sentimos, no hay suficiente stock para su pedido.
Presione Enter para regresar al menú principal.

Haciendo lo mismo pero con un stock elevado para que el programa notifique la falta de stock.

```
1 // Declaración de la
2 struct Articulo {
3     let nombre: String
4     // Propiedad nombre
5     de tipo String
6     var precio: Double
7     // Propiedad precio
8     de tipo Double
9     var stock: Int
10    // Propiedad stock de
11    tipo Int
12 }
13 // Declaración de la
14 variable articulos
15 como un array de
16 Articulo
17 var articulos = [
```

Run

***** Bienvenido a su boutique Norma *****

Artículo 1: Chamarras
Precio: 2500.0
Stock: 10

Artículo 2: Vestidos
Precio: 500.0
Stock: 80

Artículo 3: Pantalones
Precio: 700.0
Stock: 50

Artículo 4: Abrigos
Precio: 2000.0
Stock: 3

1.- Comprar artículo
2.- Salir
Seleccione una opción:
2
Gracias por visitarnos.

Y por ultimo se sale del programa y se detiene el programa agradeciendo la visita.

Enlace de GitHub: <https://github.com/Chifer888/desarrollo-de-apps-moviles-4.git>

Conclusión.

Xcode es esencial para desarrolladores que quieran publicar aplicaciones en la App Store, ya que incluye las herramientas necesarias para compilar y firmar los paquetes requeridos, y los compiladores online, son una gran opción para aprendizaje, experimentación y desarrollo rápido, sin sustituir completamente a un entorno de desarrollo local en proyectos más avanzados.

¿Qué aprendo?

Que Xcode desde mi punto de vista tiene similitudes con Android, solo que enfocado a los sistemas operativos mencionados (**iOS**, **macOS**, **watchOS** y **tvOS**), los cuales se utilizan para el desarrollo de aplicaciones, pero con un lenguaje mas sencillo de entender al momento de programar, evitando re trabajo y muy predictivo, en el caso de los compiladores en línea son muy útiles para poder aprender mas acerca de este lenguaje.

Referencias

ChatGPT. (n.d.). <https://chatgpt.com/c/6746b3b6-9754-8003-8af0-ee6595749f25>

Videoconferencias, conferencias web, seminario web, uso compartido de pantalla. (n.d.). Zoom.

https://academiaglobal-mx.zoom.us/rec/play/f5QMWaCWfJB4Uy0VupE-GLUxae7SwX9KqHvg0Ok32_jaS9gdRogobbfXOkqbrZ9FFHCcHAGvHJVM5TyK.Tu62DJVU67cd4Bg_?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2F0xbGbjl0HJL5J43u5Jx0UmUi0y7uH1GWVilRPJrrbLKANDh3tPRw15ubM8mwig18.xdF78u4lqBmTxZRU