



Actividad |1| Instalación XCode/Programa 1

Desarrollo de Aplicaciones Móviles IV

Ingeniería en Desarrollo de Software



TUTOR: Marco Rodríguez Tapia

ALUMNO: Carlos Eco Estrada Salazar

FECHA: 24/Jul/2025

ÍNDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	
Codificación	6
Prueba del programa	8
Versión optimizada	9
CONCLUSIÓN	11
REFERENCIAS	12

GitHub Links:

Document:

Code:

https://github.com/Calinny17/Desarrollo-de-aplicaciones-m-viles-IV/blob/6e3237e49091fbc69a8880169ffe1a2e6a3ec412/BoutiqueNorma_Code

CodeOpti:

https://github.com/Calinny17/Desarrollo-de-aplicaciones-m-viles-IV/blob/6e3237e49091fbc69a8880169ffe1a2e6a3ec412/BoutiqueNormaOptimi_code

INTRODUCCIÓN

En la presente actividad se desarrollará un programa básico utilizando el lenguaje de programación Swift, enfocado en simular el funcionamiento de una tienda de ropa en línea para la boutique "Norma". El objetivo principal de este ejercicio es aplicar conocimientos fundamentales de programación orientada a objetos y estructuras de control en Swift, diseñando una pequeña aplicación interactiva que permita a los clientes visualizar productos, seleccionar artículos disponibles y realizar compras de manera sencilla desde la misma interfaz de la aplicación.

El desarrollo de esta tienda virtual considerará cuatro artículos de ropa, cada uno con su respectivo nombre, precio y disponibilidad en inventario (stock). El programa ofrecerá un menú interactivo mediante el cual el cliente podrá elegir si desea comprar alguno de los artículos o salir de la tienda. En caso de seleccionar un producto, el sistema confirmará la compra mostrando el artículo adquirido junto con el monto total a pagar, calculado en función del precio del artículo y la cantidad solicitada.

Esta actividad no solo fortalece las habilidades de programación en Swift, sino que también representa una introducción al diseño de sistemas interactivos con orientación al comercio electrónico. A través de este ejercicio, se ponen en práctica elementos clave como la entrada y salida de datos, el uso de ciclos y condicionales, así como la gestión de listas o arreglos para el almacenamiento de productos. Es un paso importante en la construcción de aplicaciones funcionales dentro del entorno de desarrollo de Xcode o compiladores en línea compatibles con Swift.

DESCRIPCIÓN

En esta actividad se nos presenta el reto de desarrollar un programa en lenguaje Swift que funcione como una tienda de ropa en línea para la boutique llamada “Norma”. El contexto parte de una necesidad real: facilitar al cliente la visualización y compra de productos directamente desde una aplicación. Esto implica diseñar un sistema básico de ventas donde el cliente pueda ver los productos disponibles, elegir uno, y realizar su compra de forma sencilla y funcional.

La aplicación deberá mostrar cuatro artículos de ropa, especificando el nombre de cada uno, su precio y el stock disponible. Esto permitirá al cliente conocer la disponibilidad del producto antes de comprar. Además, se implementará un menú interactivo para que el cliente tenga la opción de realizar una compra o salir de la tienda. Si decide comprar, el programa deberá procesar esa acción, reducir el stock correspondiente y mostrar un resumen claro de lo adquirido, incluyendo el nombre del producto y el total a pagar.

Esta actividad tiene como propósito poner en práctica los conocimientos adquiridos en programación utilizando Swift, permitiendo al alumno diseñar una solución funcional basada en una situación real del comercio digital. También fomenta el pensamiento lógico y la capacidad para construir aplicaciones orientadas al usuario, habilidades esenciales para el desarrollo de software moderno.

JUSTIFICACIÓN

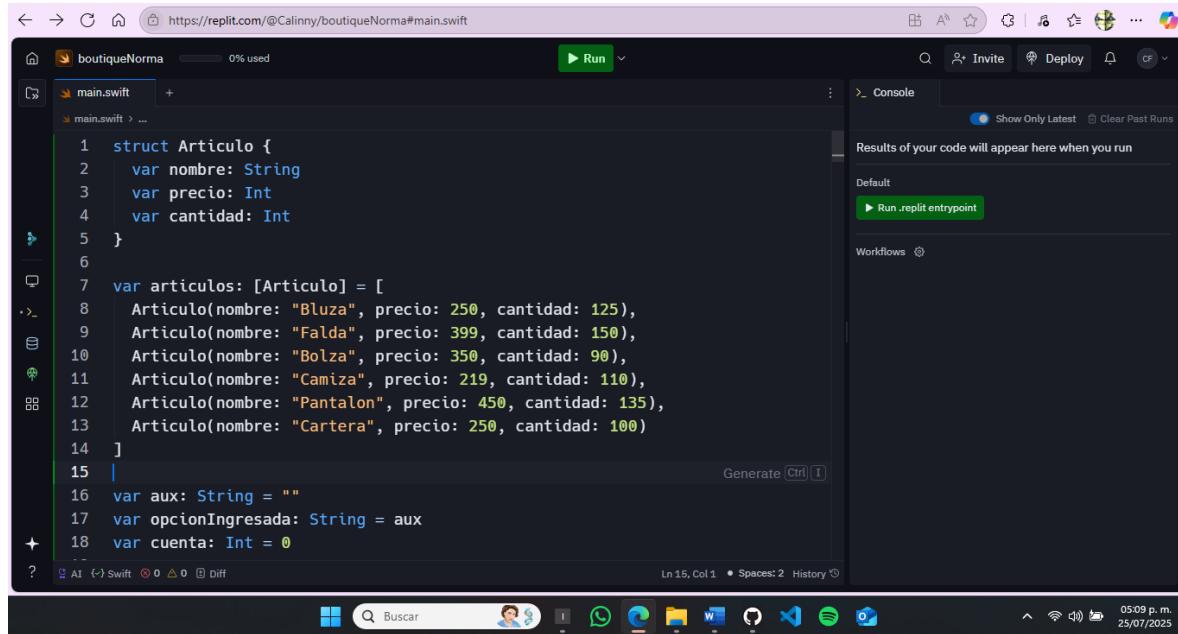
Implementar una solución digital como una tienda de ropa en línea, desarrollada en el lenguaje de programación Swift, representa una estrategia efectiva y moderna para atender las necesidades comerciales de la boutique “Norma”. Actualmente, la digitalización de servicios es una herramienta clave para mejorar la experiencia del cliente, facilitar procesos de compra y ampliar el alcance del negocio. Al desarrollar una aplicación que permita mostrar productos, gestionar inventario y procesar compras, se está optimizando tanto la operación interna del negocio como el acceso del cliente a sus productos preferidos.

Utilizar Swift como lenguaje de programación es ideal debido a su eficiencia, seguridad y compatibilidad con el ecosistema Apple, lo que hace posible crear aplicaciones visualmente atractivas y funcionales para dispositivos iOS. Esto es especialmente importante si se considera que muchos clientes actuales prefieren comprar desde sus teléfonos móviles. La automatización del proceso de compra permite disminuir errores humanos, agilizar las transacciones y ofrecer al usuario una experiencia interactiva más fluida.

Esta solución puede escalarse fácilmente, integrando más funciones como pagos electrónicos, filtros de búsqueda, o recomendaciones personalizadas. Por tanto, el desarrollo de este tipo de sistema no solo atiende una necesidad inmediata, sino que también abre la puerta a nuevas oportunidades de crecimiento para la boutique, posicionándola competitivamente en el mercado actual.

DESARROLLO

Codificación



```

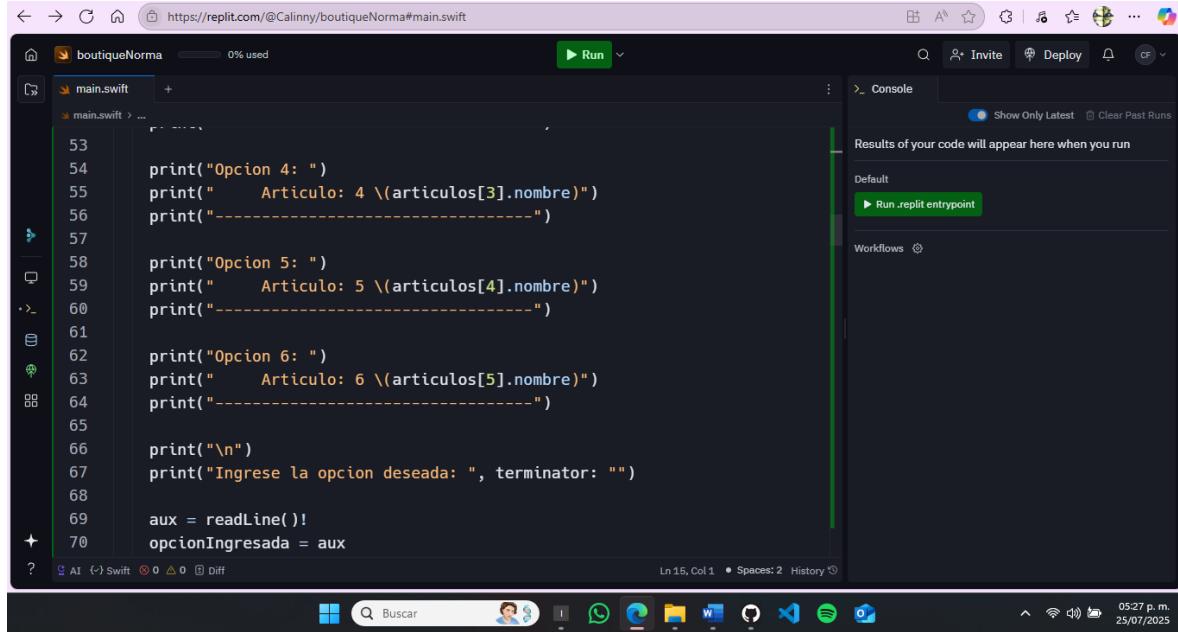
boutiqueNorma 0% used
main.swift + main.swift > ...
1 struct Articulo {
2     var nombre: String
3     var precio: Int
4     var cantidad: Int
5 }
6
7 var articulos: [Articulo] = [
8     Articulo(nombre: "Bluza", precio: 250, cantidad: 125),
9     Articulo(nombre: "Falda", precio: 399, cantidad: 150),
10    Articulo(nombre: "Bolza", precio: 350, cantidad: 90),
11    Articulo(nombre: "Camiza", precio: 219, cantidad: 110),
12    Articulo(nombre: "Pantalon", precio: 450, cantidad: 135),
13    Articulo(nombre: "Cartera", precio: 250, cantidad: 100)
14 ]
15
16 var aux: String = ""
17 var opcionIngresada: String = aux
18 var cuenta: Int = 0
  
```

Ln 15, Col 1 • Spaces: 2 History ⌂

AI Swift 0 ▲ 0 Diff

05:09 p.m. 25/07/2025

Se crean la estructura, la base para los arreglos y las variables que se utilizaran, para después crear el menú principal, donde se elige comprar o salir del programa, y el menú de opciones de artículos, en este ultimo se muestran los artículos disponibles y la opción de elegir uno de estos.



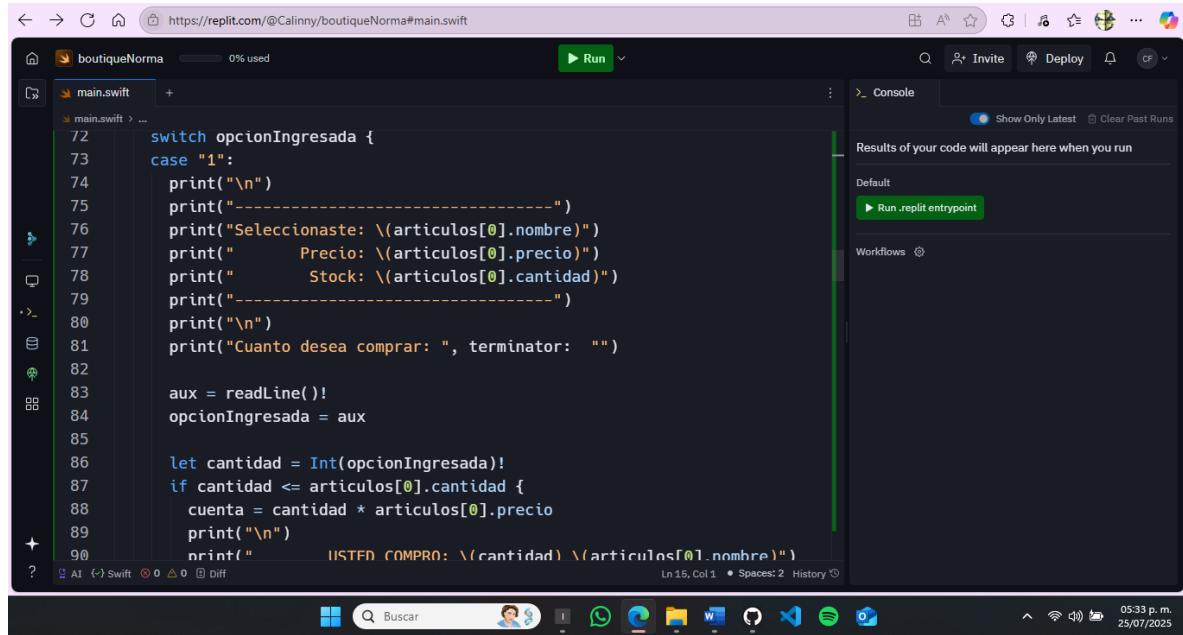
```

print("Opcion 4: ")
print("    Articulo: 4 \(articulos[3].nombre)")
print("-----")
print("Opcion 5: ")
print("    Articulo: 5 \(articulos[4].nombre)")
print("-----")
print("Opcion 6: ")
print("    Articulo: 6 \(articulos[5].nombre)")
print("-----")
print("\n")
print("Ingrese la opcion deseada: ", terminator: "")
  
```

Ln 15, Col 1 • Spaces: 2 History ⌂

AI Swift 0 ▲ 0 Diff

05:27 p.m. 25/07/2025



```

boutiqueNorma 0% used
main.swift + ...
main.swift > ...
72 switch opcionIngresada {
73     case "1":
74         print("\n")
75         print("-----")
76         print("Seleccionaste: \(articulos[0].nombre)")
77         print("      Precio: \(articulos[0].precio)")
78         print("      Stock: \(articulos[0].cantidad)")
79         print("-----")
80         print("\n")
81         print("Cuanto desea comprar: ", terminator: " ")
82
83         aux = readLine()!
84         opcionIngresada = aux
85
86         let cantidad = Int(opcionIngresada)!
87         if cantidad <= articulos[0].cantidad {
88             cuenta = cantidad * articulos[0].precio
89             print("\n")
90             print("USTED COMPRO: \(cantidad) \(articulos[0].nombre)")

```

Ln 16, Col 1 • Spaces: 2 History

AI Swift 0 0 Diff

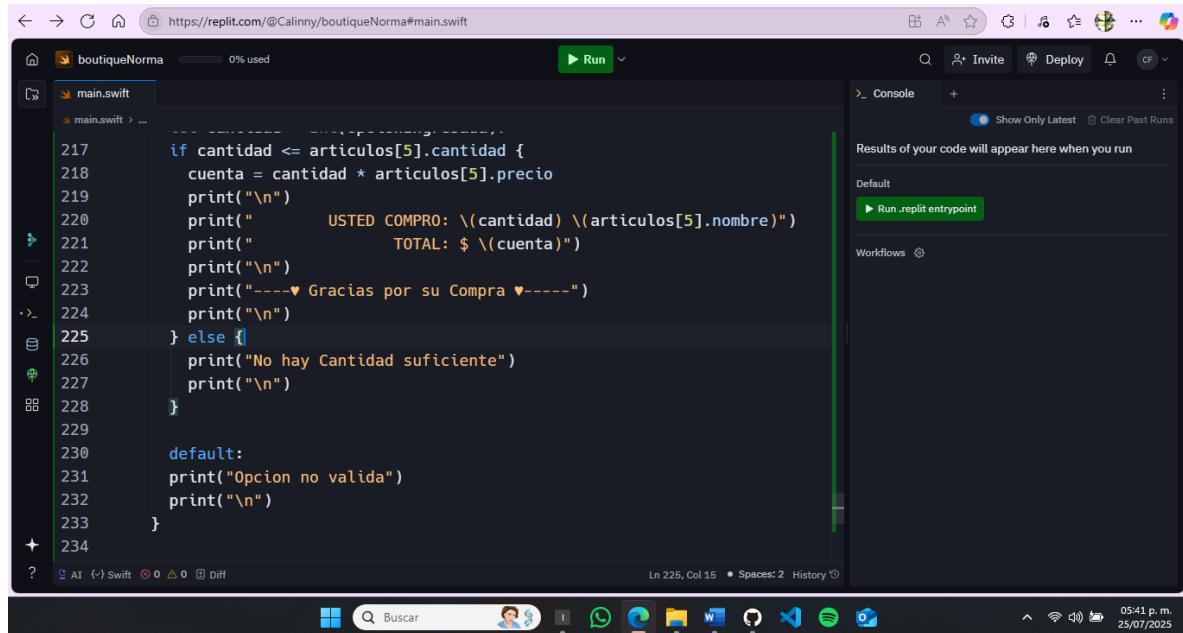
Run Only Latest Clear Past Runs

Default Run .replit entrypoint

Workflows

5:33 p.m. 25/07/2025

Con un *switch* se crean los *case* disponibles para cada uno de los arreglos ingresados, en cada uno de estos también se genera la compra de los productos, en caso de no tener el stock suficiente se genera el error. Por último, se le damos la opción de error al *switch* en caso de ingresar una opción no válida.



```

boutiqueNorma 0% used
main.swift + ...
main.swift > ...
217     if cantidad <= articulos[5].cantidad {
218         cuenta = cantidad * articulos[5].precio
219         print("\n")
220         print("      USTED COMPRO: \(cantidad) \(articulos[5].nombre)")
221         print("      TOTAL: $ \(cuenta)")
222         print("\n")
223         print("----► Gracias por su Compra ----")
224         print("\n")
225     } else {
226         print("No hay Cantidad suficiente")
227         print("\n")
228     }
229
230     default:
231         print("Opcion no valida")
232         print("\n")
233
234

```

Ln 225, Col 15 • Spaces: 2 History

AI Swift 0 0 Diff

Show Only Latest Clear Past Runs

Default Run .replit entrypoint

Workflows

5:41 p.m. 25/07/2025

Prueba del programa

The screenshot shows a Replit IDE interface with a dark theme. The top bar displays the URL <https://replit.com/@Calinny/boutiqueNorma>. The left sidebar contains project files like `main.swift` and `main.swift ~`. The main code editor area shows a Swift script for a boutique application:

```
boutiqueNorma 0% used
Stop

main.swift + main.swift > ...
19
20 while opcionIngresada != "2" {
21     print("-----")
22     print("-----♥ !!Bienvenidoii ♥-----")
23     print("-----♥--- Boutique Norma ---♥-----")
24     print("-----")
25     print("\n")
26     print("Opcion 1: Comprar Articulo")
27     print("Opcion 2: Salir")
28     print("\n")
29     print("-----")
30     print("Ingrese la opcion: ", terminator: "")
31
32     aux = readLine()!
33     opcionIngresada = aux
34
35     if opcionIngresada == "1"{
36         print("\n")
37         print("-----")
```

The right side of the interface shows the console output:

```
Console
Show Only Latest Clear Past Runs
Ask Assistant

swift main.s...
-----♥ !!Bienvenidoii ♥-----
-----♥--- Boutique Norma ---♥-----
-----
```

Output:

```
Opcion 1: Comprar Articulo
Opcion 2: Salir

Ingrese la opcion: 1

-----♥ Articulos Disponibles ♥-----

Opcion 1:
    Articulo: 1 Bluza

Opcion 2:
```

Corremos el programa y elegimos la opción *comprar artículo*, esto nos despliega el siguiente menú, el en el cual podemos elegir una de las opciones disponibles para observar sus datos, elegir la cantidad que se desea comprar y realizar la compra.

The screenshot shows a code editor interface for a Swift script named `main.swift`. The code is a simple menu-based application for a boutique named "Boutique Norma". It asks the user for an option (1 to buy an item or 2 to exit), handles input from the terminal, and prints a welcome message and a summary of the purchase.

```
boutiqueNorma 0% used Stop

main.swift
main.swift > ...

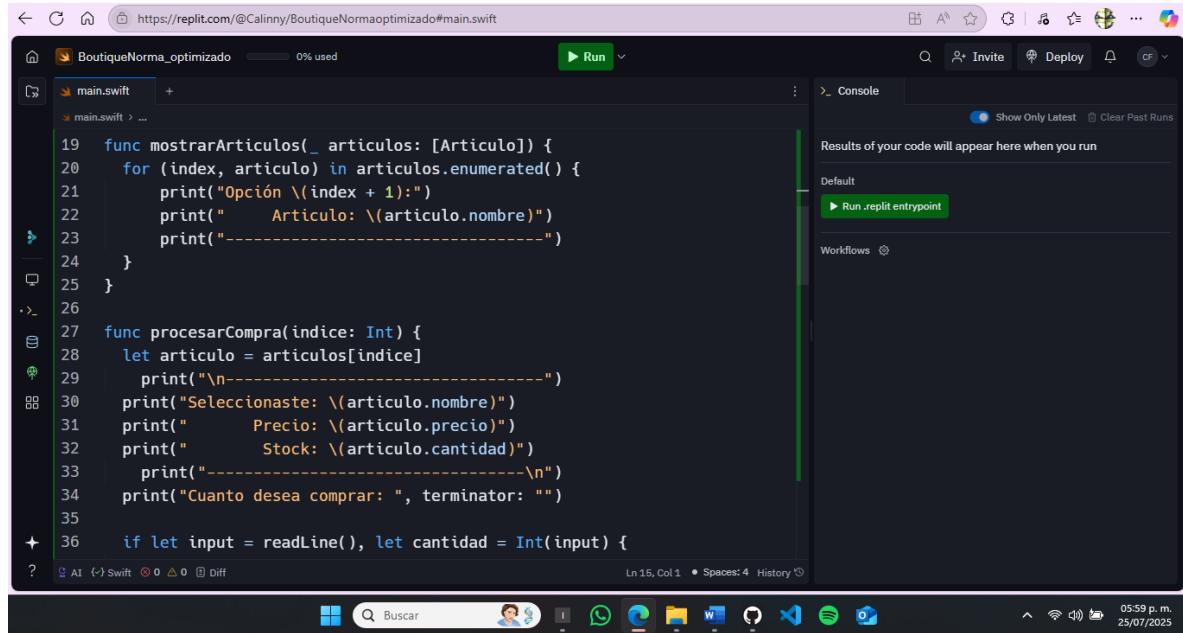
19
20 while opcionIngresada != "2" {
21     print("-----")
22     print("-----♥ !!Bienvenidoii ♥-----")
23     print("-----♥♥--- Boutique Norma ---♥♥---")
24     print("-----")
25     print("\n")
26     print("Opcion 1: Comprar Articulo")
27     print("Opcion 2: Salir")
28     print("\n")
29     print("-----")
30     print("Ingrese la opcion: ", terminator: "")
31
32     aux = readLine()!
33     opcionIngresada = aux
34
35     if opcionIngresada == "1"{
36         print("\n")
37         print("-----")
```

Console Output:

```
Show Only Latest Clear Past Runs
Ask Assistant

Ingrese la opcion deseada: 5
-----
Seleccionaste: Pantalon
Precio: 450
Stock: 135
-----
Cuanto desea comprar: 8
USTED COMPRO: 8 Pantalon
TOTAL: $ 3600
-----
-----♥ Gracias por su Compra ♥-----
-----♥ !!Bienvenidoii ♥-----
```

Versión optimizada del código



```

func mostrarArticulos(_ articulos: [Articulo]) {
    for (index, articulo) in articulos.enumerated() {
        print("Opción \(index + 1):")
        print("    Artículo: \(articulo.nombre)")
        print("-----")
    }
}

func procesarCompra(indice: Int) {
    let articulo = articulos[indice]
    print("\n-----")
    print("Seleccionaste: \(articulo.nombre)")
    print("    Precio: \(articulo.precio)")
    print("    Stock: \(articulo.cantidad)")
    print("-----\n")
    print("Cuanto desea comprar: ", terminator: "")
}

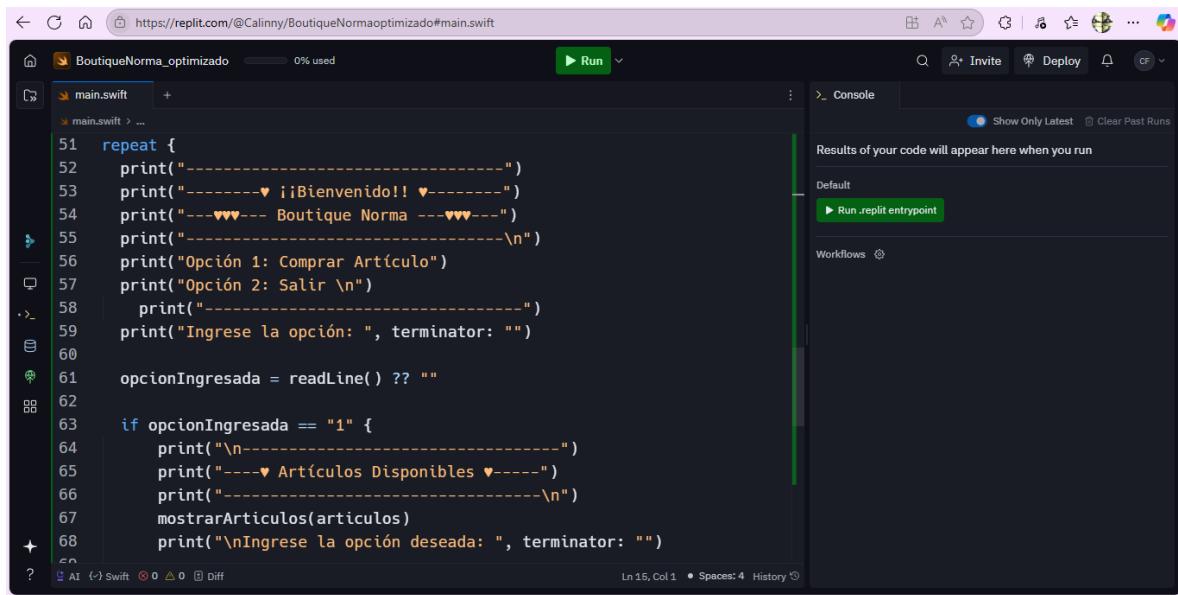
if let input = readLine(), let cantidad = Int(input) {
    procesarCompra(indice: cantidad)
}

```

The screenshot shows the Replit IDE interface with a dark theme. On the left is the code editor for a file named 'main.swift' under the project 'BoutiqueNorma_optimizado'. The code has been refactored to use functions like `mostrarArticulos()` and `procesarCompra()`. On the right is the 'Console' panel, which is currently empty. At the bottom, there's a toolbar with various icons and a status bar showing the date and time.

Mejoras realizadas:

- El uso de `func mostrarArticulos()` evita repetir el código para listar artículos.
- Se eliminan todos los `case` del `switch`, y se maneja dinámicamente el artículo seleccionado.
- Se implementa la función `procesarCompra()` para no repetir lógica por cada artículo.
- Uso de `enumerated()` para imprimir dinámicamente cada artículo.



```

repeat {
    print("-----")
    print("----> i;Bienvenido!! <----")
    print("----> ♥♥--- Boutique Norma ---♥♥---")
    print("-----\n")
    print("Opción 1: Comprar Artículo")
    print("Opción 2: Salir \n")
    print("-----")
    print("Ingrese la opción: ", terminator: "")
    opcionIngresada = readLine() ?? ""

    if opcionIngresada == "1" {
        print("\n-----")
        print("----> Artículos Disponibles <----")
        print("-----\n")
        mostrarArticulos(articulos)
        print("\nIngrese la opción deseada: ", terminator: "")
    }
} while opcionIngresada != "2"

```

This screenshot shows the enhanced version of the code. It includes a `repeat` loop that runs until the user enters '2'. Inside the loop, it prints a welcome message, displays the available articles, and then asks for the user's choice. The logic for handling the choice is moved into the `if opcionIngresada == "1"` block. The code editor, console, and other UI elements are visible in the background.

Prueba del código

The screenshot shows a Swift code editor interface with a terminal window on the right. The code in main.swift prints a welcome message and a menu. The terminal shows the game loop running, with user input '1' selected.

```
repeat {
    print("-----")
    print("-----♥ !!Bienvenido!! ♥-----")
    print("-----♥--- Boutique Norma ---♥---")
    print("-----\n")
    print("Opción 1: Comprar Artículo")
    print("Opción 2: Salir \n")
    print("-----")
    print("Ingrese la opción: ", terminator: "")
}

opcionIngresada = readLine() ?? ""

if opcionIngresada == "1" {
    print("\n-----")
    print("-----♥ Artículos Disponibles ♥-----")
    print("-----\n")
    mostrarArticulos(articulos)
    print("\nIngrese la opción deseada: ", terminator: "")
}
```

Console output:

```
-----♥--- Boutique Norma ---♥---  
-----  
Opción 1: Comprar Artículo  
Opción 2: Salir  
-----  
Ingrese la opción: 1  
-----  
-----♥ Artículos Disponibles ♥-----  
-----  
Opción 1:  
Articulo: Blusa  
-----  
Opción 2:  
Articulo: Falda  
-----  
Opción 3:  
Articulo: Bolza  
-----  
Opción 4:
```

De 250 líneas de la primer versión del código a 82 líneas de código ya optimizado.

```
23 print("-----")
24 }
25 }
26
27 func procesarCompra(indice: Int) {
28     let articulo = articulos[indice]
29     print("\n-----")
30     print("Seleccionaste: \(articulo.nombre)")
31     print("      Precio: \(articulo.precio)")
32     print("      Stock: \(articulo.cantidad)")
33     print("-----\n")
34     print("Cuanto desea comprar: ", terminator: "")
35
36     if let input = readLine(), let cantidad = Int(input) {
37         if cantidad <= articulo.cantidad {
38             cuenta = cantidad * articulo.precio
39             articulos[indice].cantidad -= cantidad
40             print("\nUSTED COMPRÓ: \(cantidad) \(articulo.nombre)")
41             print("-----")
42             print("TOTAL: $", cuenta)
43         }
44     }
45 }
```

Stop

Console

Show Only Latest

Clear Past Runs

swift main.s...

Opción 6:
Artículo: Cartera

Ingrese la opción deseada: 2

Seleccionaste: Falda
Precio: 399
Stock: 150

Cuanto desea comprar: 6

USTED COMPRÓ: 6 Falda
TOTAL: \$2394

----♥ Gracias por su Compra ♥----

Presione Enter para continuar...

CONCLUSIÓN

La realización de esta actividad ha permitido poner en práctica conocimientos fundamentales de programación en Swift mediante el desarrollo de una aplicación sencilla pero funcional que simula una tienda de ropa en línea. Esta experiencia no solo ha reforzado habilidades técnicas como la creación de estructuras, manejo de arreglos, uso de ciclos, condicionales y entrada de datos, sino que también ha fomentado el pensamiento lógico y la capacidad de resolución de problemas orientada a situaciones reales del entorno comercial.

Desde el punto de vista profesional, este tipo de ejercicios es altamente valioso, ya que el comercio electrónico sigue creciendo rápidamente, y contar con las habilidades necesarias para diseñar e implementar soluciones digitales es una ventaja competitiva en el mercado laboral actual. La automatización de procesos, como la compra y venta de productos, optimiza la experiencia del cliente, reduce errores humanos y permite ampliar el alcance de los negocios locales a una escala mucho mayor.

En la vida cotidiana, entender cómo funcionan las aplicaciones que utilizamos diariamente para comprar o gestionar servicios digitales nos brinda un enfoque más crítico y consciente de la tecnología que consumimos. Además, nos prepara para emprender nuestros propios proyectos, adaptándonos a un mundo cada vez más digitalizado. Esta actividad, aunque sencilla, representa un paso importante hacia el desarrollo de soluciones reales con impacto directo en la economía, el comercio y la vida diaria.

REFERENCIAS

- Programación by AristiDevs. (2024, 30 mayo).  CURSO: SWIFT y SWIFTUI desde CERO en ESPAÑOL - Programación IOS - TUTORIAL XCODE [Vídeo]. YouTube. <https://www.youtube.com/watch?v=f6WtmTBFNGM>
- José Dimas Luján Castillo. (2023, 27 enero). Clase 1 - Presentación del curso [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Hxmc9L30-3g>