



## Actividad | 2 | Aplicación 2

### Desarrollo de Aplicaciones Móviles III

Ingeniería en Desarrollo de Software

---



academiaglobal

TUTOR: Sandra Luz Lara Dévora

---

ALUMNO: Carlos Fco Estrada Salazar

---

FECHA: 12/Abril/2025

---

## INDICE

INTRODUCCIÓN	3
DESCRIPCIÓN	4
JUSTIFICACIÓN	5
DESARROLLO	6
Codificación	6
Prueba de la aplicación	9
CONCLUSIÓN	12
REFERENCIAS	13

GitHub Link Document:

GitHub Link Code:

Zip Google Drive:

## INTRODUCCIÓN

En la actualidad, la gestión eficiente del inventario es un aspecto clave para cualquier negocio, sin importar su tamaño. Un adecuado control de los productos permite optimizar los recursos, evitar pérdidas y mejorar la experiencia del cliente. En este contexto, el desarrollo de aplicaciones especializadas se ha convertido en una solución fundamental para automatizar procesos y facilitar la administración de los artículos disponibles en una tienda.

En esta actividad, se desarrollará una aplicación en Swift que servirá como un sistema de inventario para una tienda de la esquina. Esta aplicación permitirá a los empleados registrar productos, visualizar la lista de artículos, consultar la cantidad de productos en existencia y salir del sistema cuando sea necesario. La implementación se realizará en la plataforma Replit, lo que facilitará su desarrollo, prueba y ejecución en un entorno accesible y en línea.

El objetivo principal de este proyecto es proporcionar una herramienta sencilla, pero efectiva, que ayude a los empleados a mantener un mejor control del inventario. Con un menú interactivo, la aplicación garantizará que la información de los productos se registre y consulte de manera eficiente. Además, este ejercicio permitirá poner en práctica conocimientos de programación en Swift, incluyendo la manipulación de listas, estructuras de control y funciones.

A través de este desarrollo, se busca reforzar habilidades en el diseño y la implementación de aplicaciones orientadas a resolver problemas reales en el ámbito comercial. La solución propuesta será funcional y escalable, con la posibilidad de futuras mejoras e integración con otros sistemas, brindando así un aprendizaje práctico en el desarrollo de software.

## DESCRIPCIÓN

En el contexto de la actividad, se plantea la necesidad de una aplicación que facilite la gestión del inventario en una tienda de la esquina. Este tipo de establecimientos suelen manejar una gran variedad de productos, y su control eficiente es crucial para garantizar disponibilidad, evitar pérdidas y optimizar el proceso de ventas. Sin un sistema adecuado, los empleados pueden enfrentar dificultades al registrar nuevos productos, llevar un control de los artículos existentes y consultar la disponibilidad de los productos de manera rápida y precisa.

Para abordar este problema, se requiere el desarrollo de una aplicación en lenguaje Swift que proporcione una solución práctica y funcional. La aplicación deberá incluir un menú con cuatro opciones clave: registrar un artículo, ver la lista de artículos, consultar los productos en existencia y salir del sistema. Cada una de estas funciones tiene un propósito específico dentro de la gestión del inventario.

El registro de artículos permitirá a los empleados ingresar nuevos productos con detalles como nombre, cantidad y posiblemente categoría. La opción de visualizar la lista de artículos brindará una visión general de todos los productos registrados, facilitando el control y la toma de decisiones. La consulta de artículos en existencia permitirá verificar la disponibilidad de productos específicos, ayudando a mantener un stock adecuado y evitar quiebres de inventario. Finalmente, la opción de salir proporcionará una manera sencilla de cerrar la aplicación una vez que se haya realizado la gestión necesaria.

El uso de Swift como lenguaje de programación ofrece ventajas como rapidez, seguridad y facilidad de uso, lo que lo convierte en una opción ideal para desarrollar esta aplicación. La elección de la plataforma Replit permite que el desarrollo y ejecución del programa sean accesibles desde cualquier dispositivo con conexión a internet, sin la necesidad de configuraciones complejas.

Esta actividad representa un ejercicio práctico en el desarrollo de software, aplicando conocimientos en programación con Swift para solucionar un problema real dentro del entorno comercial.

## JUSTIFICACIÓN

La implementación de una aplicación en Swift para la gestión del inventario en una tienda de la esquina representa una solución eficiente y práctica que responde a la necesidad de mantener un control preciso sobre los productos disponibles. Un sistema de inventario digital reduce errores humanos, facilita la actualización de datos en tiempo real y mejora la eficiencia en la toma de decisiones relacionadas con las compras y ventas del negocio.

Uno de los principales motivos para optar por esta solución es la optimización del tiempo y los recursos. Un inventario manual o basado en papel puede ser propenso a errores, pérdida de datos y dificultades para consultar información rápidamente. En cambio, con una aplicación digital, los empleados pueden registrar artículos de manera estructurada, visualizar la lista completa de productos y verificar la existencia de artículos con solo unos cuantos comandos, asegurando un mejor flujo de trabajo.

El uso de Swift como lenguaje de desarrollo ofrece ventajas significativas. Swift es conocido por su rapidez, seguridad y facilidad de mantenimiento, lo que lo convierte en una excelente opción para crear aplicaciones funcionales y escalables. Además, la elección de la plataforma Replit facilita el acceso y la ejecución del programa desde cualquier dispositivo con conexión a internet, sin necesidad de configuraciones adicionales, lo que favorece su implementación en entornos de trabajo con recursos limitados.

Esta aplicación no solo mejora la organización del negocio, sino que también sienta las bases para futuras mejoras. Por ejemplo, en el futuro se podrían agregar funciones adicionales como alertas de productos con bajo stock, generación de reportes de ventas y sincronización con plataformas de punto de venta. De este modo, se ofrece una solución adaptable a las necesidades cambiantes del negocio.

El desarrollo de esta aplicación en Swift representa una respuesta efectiva al problema presentado en la actividad. Proporciona una herramienta confiable, accesible y fácil de usar para gestionar el inventario de la tienda de manera eficiente, mejorando así su operatividad y reduciendo el margen de error en el control de productos.

## DESARROLLO

### Codificación

```
struct Producto {
    var id: Int
    var nombre: String
    var cantidad: Int
}
```

Aquí definimos una **estructura** (*struct*) llamada *Producto*, que representa un artículo del inventario. Tiene tres propiedades:

- id: Identificador único del artículo.
- nombre: Nombre del artículo.
- cantidad: Cantidad disponible en el inventario.

```
var inventario: [Producto] = []
```

Aquí creamos un **arreglo vacío** de tipo *Producto*, donde almacenaremos los artículos registrados en el inventario.

```
func mostrarMenu() {
    print("\n\tMENÚ")
    print("1. Registrar artículos.")
    print("2. Ver lista de artículos.")
    print("3. Editar artículos.")
    print("4. Eliminar artículos.")
    print("5. Salir.")
    print("\nElija una opción (Número): ", terminator:
"" )
}
```

Esta función simplemente muestra el menú de opciones en pantalla.

El *terminator: ""* evita el salto de línea después del *print*, para que la entrada del usuario quede en la misma línea.

```
func registrarArticulo() {
    print("\n\t1. Registrar artículos")
    print("Por favor ingresa lo que se te pide:")
    print("Ingresa el ID del artículo:", terminator: " ")
    guard let id = Int(readLine() ?? "") else { return }
    print("Ingresa el nombre del artículo:", terminator: " ")
    let nombre = readLine() ?? ""
    print("Ingresa cuántos artículos registras:", terminator: " ")
    guard let cantidad = Int(readLine() ?? "") else { return }

    let nuevoProducto = Producto(id: id, nombre: nombre,
cantidad: cantidad)
    inventario.append(nuevoProducto)
    print("Artículo registrado con éxito.")
}
```

Esta función permite registrar un nuevo artículo:

1. Se pide el **ID** (conversión a entero).
2. Se pide el **nombre** del artículo.
3. Se pide la **cantidad** (conversión a entero).
4. Se crea una nueva instancia de *Producto* y se agrega al *inventario*.

El *guard let* evita que el programa falle si el usuario ingresa un valor no válido.

```
func verListaArticulos() {
    print("\n\t2. Lista de Productos")
    for producto in inventario {
        print("ID: \(producto.id) - Nombre: \(producto.nombre) -
    Cantidad: \(producto.cantidad)")
    }
}
```

Esta función recorre el arreglo *inventario* y muestra cada artículo en un formato legible.

```
func editarArticulo() {
    print("\n\t3. Editar artículos")
    print("ID del artículo que quieres editar:", terminator: " ")
    guard let id = Int(readLine() ?? "") else { return }

    if let index = inventario.firstIndex(where: { $0.id == id }) {
        print("ID: \(inventario[index].id) - Nombre: \(inventario[index].nombre) - Cantidad: \(
    inventario[index].cantidad)")
        print("Nuevo nombre del artículo:", terminator: " ")
        let nuevoNombre = readLine() ?? ""
        print("Nueva cantidad del artículo:", terminator: " ")
        guard let nuevaCantidad = Int(readLine() ?? "") else { return }

        inventario[index].nombre = nuevoNombre
        inventario[index].cantidad = nuevaCantidad
        print("Artículo editado con éxito.")
    } else {
        print("Artículo no encontrado.")
    }
}
```

1. Se solicita el ID del artículo a editar.
2. Se busca el artículo en el inventario usando *firstIndex(where:)*.
3. Si se encuentra, se muestran sus datos actuales.
4. Se solicita un nuevo nombre y cantidad.
5. Se actualizan los valores en el inventario.

```
func eliminarArticulo() {
    print("\n\t4. Eliminar artículos")
    print("Ingresa el ID del artículo a eliminar:", terminator: " ")
    guard let id = Int(readLine() ?? "") else { return }

    if let index = inventario.firstIndex(where: { $0.id == id }) {
        let producto = inventario[index]
        print("ID: \(producto.id) - Nombre: \(producto.nombre) - Cantidad: \(producto.cantidad)")
        print("¿Desea eliminar este artículo? (Sí/No):", terminator: " ")
        let confirmacion = readLine()?.lowercased()

        if confirmacion == "si" {
            inventario.remove(at: index)
            print("Artículo eliminado con éxito.")
        } else {
            print("Eliminación cancelada.")
        }
    } else {
        print("Artículo no encontrado.")
    }
}
```

1. Se pide el ID del artículo a eliminar.
2. Se busca en el inventario.
3. Si se encuentra, se muestran los datos del producto.
4. Se solicita confirmación para eliminar (Sí/No).
5. Si el usuario confirma, se borra con *remove(at:)*.

```
var ejecutando = true
while ejecutando {
    mostrarMenu()
    if let opcion = Int(readLine() ?? "") {
        switch opcion {
            case 1:
                registrarArticulo()
            case 2:
                verListaArticulos()
            case 3:
                editarArticulo()
            case 4:
                eliminarArticulo()
            case 5:
                ejecutando = false
                print("Saliendo de la aplicación...")
            default:
                print("Opción no válida. Intente de nuevo.")
        }
    }
}
```

Aquí tenemos el **bucle principal**, que:

1. Muestra el *menú*.
2. Recibe la opción elegida por el usuario.
3. Llama a la función correspondiente con *switch case*.
4. Si el usuario elige 5, cambia *ejecutando* a *false* y termina la ejecución.



## Prueba de aplicación

```

1 import Foundation
2
3 struct Producto {
4     var id: Int
5     var nombre: String
6     var cantidad: Int
7 }
8
9 var inventario: [Producto] = []
10
11 func mostrarMenu() {
12     print("\n\tMENÚ")
13     print("1. Registrar artículos.")
14     print("2. Ver lista de artículos.")
15     print("3. Editar artículos.")
16     print("4. Eliminar artículos.")
17     print("5. Salir.")
18     print("\nElija una opción (Número): ",

```

1. Registrar artículos  
Por favor ingresa lo que se te pide:  
Ingresa el ID del artículo: 3  
Ingresa el nombre del artículo: Camisa  
Ingresa cuántos artículos registras: 10  
Artículo registrado con éxito.

MENÚ  
1. Registrar artículos.  
2. Ver lista de artículos.  
3. Editar artículos.  
4. Eliminar artículos.  
5. Salir.

Elija una opción (Número): 1

1. Registrar artículos  
Por favor ingresa lo que se te pide:  
Ingresa el ID del artículo: 4  
Ingresa el nombre del artículo: Blusa  
Ingresa cuántos artículos registras: 10  
Artículo registrado con éxito.

Opción 1, la cual es registrar un producto, la misma pide, registro de ID, Nombre de producto y cantidad.

```

1 import Foundation
2
3 struct Producto {
4     var id: Int
5     var nombre: String
6     var cantidad: Int
7 }
8
9 var inventario: [Producto] = []
10
11 func mostrarMenu() {
12     print("\n\tMENÚ")
13     print("1. Registrar artículos.")
14     print("2. Ver lista de artículos.")
15     print("3. Editar artículos.")
16     print("4. Eliminar artículos.")
17     print("5. Salir.")
18     print("\nElija una opción (Número): ",

```

MENÚ  
1. Registrar artículos.  
2. Ver lista de artículos.  
3. Editar artículos.  
4. Eliminar artículos.  
5. Salir.

Elija una opción (Número): 2

2. Lista de Productos  
ID: 1 - Nombre: Pantalon - Cantidad: 10  
ID: 2 - Nombre: Pantalon mujer - Cantidad: 10  
ID: 3 - Nombre: Camisa - Cantidad: 10  
ID: 4 - Nombre: Blusa - Cantidad: 10

MENÚ  
1. Registrar artículos.  
2. Ver lista de artículos.  
3. Editar artículos.  
4. Eliminar artículos.  
5. Salir.

Elija una opción (Número):

La opción 2 despliega una lista de todos los productos registrados en la opción 1.

```

main.swift > () Producto > ...
return }
47
48     if let index = inventario.firstIndex(where: {
49         $0.id == id }) {
50         print("ID: \(inventario[index].id) - Nombre:
51         \(inventario[index].nombre) - Cantidad: \(
52         inventario[index].cantidad)")
53         print("Nuevo nombre del artículo:",
54         terminator: " ")
55         let nuevoNombre = readLine() ?? ""
56         print("Nueva cantidad del artículo:",
57         terminator: " ")
58         guard let nuevaCantidad = Int(readLine()) ??
59         "" else { return }
60
61         inventario[index].nombre = nuevoNombre
62         inventario[index].cantidad = nuevaCantidad
63         print("Artículo editado con éxito.")
64     } else {
65     }
66 }

```

```

ID: 3 - Nombre: Camisa - Cantidad: 10
ID: 4 - Nombre: Blusa - Cantidad: 10

MENÚ
1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

Elija una opción (Número): 3

3. Editar artículos
ID del artículo que quieres editar: 2
ID: 2 - Nombre: Pantalón mujer - Cantidad: 10
Nuevo nombre del artículo: Pantalón Campana mujer
Nueva cantidad del artículo: 10
Artículo editado con éxito.

MENÚ
1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

```

La opción 3 del programa edita un artículo previamente registrado

```

main.swift > () Producto > ...
return }
47
48     if let index = inventario.firstIndex(where: {
49         $0.id == id }) {
50         print("ID: \(inventario[index].id) - Nombre:
51         \(inventario[index].nombre) - Cantidad: \(
52         inventario[index].cantidad)")
53         print("Nuevo nombre del artículo:",
54         terminator: " ")
55         let nuevoNombre = readLine() ?? ""
56         print("Nueva cantidad del artículo:",
57         terminator: " ")
58         guard let nuevaCantidad = Int(readLine()) ??
59         "" else { return }
60
61         inventario[index].nombre = nuevoNombre
62         inventario[index].cantidad = nuevaCantidad
63         print("Artículo editado con éxito.")
64     } else {
65     }
66 }

```

```

MENÚ
1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

Elija una opción (Número): 2

2. Lista de Productos
ID: 1 - Nombre: Pantalón - Cantidad: 10
ID: 2 - Nombre: Pantalón Campana mujer - Cantidad: 10
ID: 3 - Nombre: Camisa - Cantidad: 10
ID: 4 - Nombre: Blusa - Cantidad: 10

MENÚ
1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

Elija una opción (Número):

```

Artículo editado en la opción 3.

```

terminator: " ")
51     let nuevoNombre = readLine() ?? ""
52     print("Nueva cantidad del artículo:",
terminator: " ")
53     guard let nuevaCantidad = Int(readLine() ??
"" ) else { return }
54
55     inventario[index].nombre = nuevoNombre
56     inventario[index].cantidad = nuevaCantidad
57     print("Artículo editado con éxito.")
58 } else {
59     print("Artículo no encontrado.")
60 }
61 }
62
63 func eliminarArticulo() {
64     print("\n\t4. Eliminar artículos")
65     print("Ingresa el ID del artículo a eliminar:",

```

Eliza una opción (Número): 4

4. Eliminar artículos  
Ingresa el ID del artículo a eliminar: 1  
ID: 1 - Nombre: Pantalón - Cantidad: 10  
¿Desea eliminar este artículo? (Si/No): si  
Artículo eliminado con éxito.

MENÚ

1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

Eliza una opción (Número): 2

2. Lista de Productos  
ID: 2 - Nombre: Pantalón Campana mujer - Cantidad: 10  
ID: 3 - Nombre: Camisa - Cantidad: 10  
ID: 4 - Nombre: Blusa - Cantidad: 10

MENÚ

La opción 4 elimina un artículo de la lista de artículos registrados.

```

55     inventario[index].nombre = nuevoNombre
56     inventario[index].cantidad = nuevaCantidad
57     print("Artículo editado con éxito.")
58 } else {
59     print("Artículo no encontrado.")
60 }
61 }
62
63 func eliminarArticulo() {
64     print("\n\t4. Eliminar artículos")
65     print("Ingresa el ID del artículo a eliminar:",
terminator: " ")
66     guard let id = Int(readLine() ?? "") else {
return }
67
68     if let index = inventario.firstIndex(where: {
$0.id == id }) {
69         let producto = inventario[index]
70         print("ID: \(producto.id) - Nombre: \(

```

1. Registrar artículos.  
2. Ver lista de artículos.  
3. Editar artículos.  
4. Eliminar artículos.  
5. Salir.

Eliza una opción (Número): 2

2. Lista de Productos  
ID: 2 - Nombre: Pantalón Campana mujer - Cantidad: 10  
ID: 3 - Nombre: Camisa - Cantidad: 10  
ID: 4 - Nombre: Blusa - Cantidad: 10

MENÚ

1. Registrar artículos.
2. Ver lista de artículos.
3. Editar artículos.
4. Eliminar artículos.
5. Salir.

Eliza una opción (Número): 5  
Saliendo de la aplicación...

La app no se cierra hasta seleccionar la opción 5 Salir.

## CONCLUSIÓN

La realización de esta actividad tiene una gran relevancia tanto en el campo laboral como en la vida cotidiana, especialmente en entornos donde el manejo de inventarios y el control de productos es crucial. A través del desarrollo de una aplicación básica en Swift para gestionar un inventario, se ponen en práctica conceptos esenciales de la programación estructurada, tales como estructuras de datos, condicionales, ciclos y funciones. Estos conocimientos son fundamentales para cualquier desarrollador que desee construir soluciones útiles y funcionales.

En el ámbito laboral, una aplicación de inventario permite optimizar el control de productos, facilita la toma de decisiones y reduce los errores humanos, como el doble registro o el olvido de actualizar la cantidad de artículos. Este tipo de herramientas es indispensable en tiendas, almacenes, oficinas o cualquier otro entorno donde se manejen bienes materiales. Además, permite a los empleados enfocarse en tareas de mayor valor añadido, dejando las operaciones repetitivas en manos del software.

En lo personal, esta práctica demuestra cómo la programación puede resolver problemas cotidianos de manera eficiente. El desarrollo de esta app fortalece la lógica de programación y demuestra cómo una herramienta digital puede mejorar la organización de cualquier tipo de inventario, desde una pequeña tienda hasta colecciones personales. Aprender a automatizar este tipo de procesos es una habilidad poderosa y cada vez más demandada en la era digital.

## REFERENCIAS

- MoureDev by Brais Moure. (2022, 14 enero). *¿CÓMO ser APPLE DEVELOPER? GUÍA de ESTUDIO Desde Cero [2022]* [Vídeo]. YouTube.  
<https://www.youtube.com/watch?v=-w3R4DEwd2g>
- José Dimas Luján Castillo. (2023, 27 enero). *Clase 1 - Presentación del curso* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Hxmc9L30-3g>