



Actividad | 2 | Aplicación 2.

Desarrollo de Aplicaciones Móviles III.

Ingeniería en Desarrollo de
Software.



academiaglobal

TUTOR: Sandra Luz Lara Dévora

ALUMNO: Carlos Ariel Nicolini

FECHA: 29/09/2025

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo.....	6
• Codificación	6
• Prueba de la aplicación.....	11
Conclusión.....	14
Referencias.....	15

Introducción

El lenguaje de programación Swift no solo ha gozado de una amplia aceptación, sino que también se ha convertido en una de las herramientas favoritas de los desarrolladores de software. Sus mejores características son algunas de las siguientes:

1. Swift es más fácil de leer.
2. También es fácil de mantener por que los programadores no necesitan esforzarse en la contabilidad.
3. El fallo de Swift en tiempo de ejecución se detendrá en la línea de código donde se haya usado una variable opcional nula, lo que evita errores en le código. Por lo tanto, Swift es un lenguaje de programación seguro.
4. Las enormes fugas de memoria que un programador puede tener en Objective-C son imposibles en Swift. Podemos decir que se integra con una gestión de memoria eficiente.
5. Tiene una estructura de código concisa.
6. Swift es un lenguaje de programación extremadamente rápido.
7. Con Swift, los espacios de nombres se basan en el destino al que pertenece un archivo de código. Esto significa que los programadores pueden diferenciar clases o valores mediante el identificador del espacio de nombres.
8. Admite bibliotecas dinámicas.
9. Con los patios de juego se fomenta la codificación interactiva.
10. Swift proporciona a la comunidad de desarrollo una forma directa de influir en un lenguaje para crear aplicaciones intuitivas.

Descripción

Contextualización:

Los empleados de la tienda de la esquina necesitan una aplicación para controlar diversas funciones del inventario. En esta se debe poder registrar y visualizar los productos para tener un control mejor. Se requiere que la aplicación se desarrolle en lenguaje Swift.

Actividad:

Crear la aplicación requerida en lenguaje Swift, de manera que funcione como un inventario. Esta deberá contar con un menú que tenga las siguientes especificaciones:

- Registrar un artículo.
- Ver la lista de artículos.
- Consultar los artículos en existencia.
- Opción de salir.

En este ejercicio realizaremos el código para un programa de inventario. La clase es de una explicación completa, solo que tuve que realizarle unos pequeños cambios al código ya que mis modificaciones habían ocasionado un mal funcionamiento, pero lo remedié de la manera que a mi gusto mostrara los resultados y demás opciones.

Espero que el trabajo presentado cumpla con lo solicitado y las expectativas. Muchas gracias por todo su apoyo profesora y por la explicación magistral, la cual fue muy directa y fácil de comprender.

Justificación

Para este nuevo ejercicio debemos realizar un programa de inventarios, donde podremos ingresar productos y la cantidad, además de poder realizar consultas generales del inventario en general o de algún producto en particular.

Continuamos utilizando y aprendiendo sobre el lenguaje Swift, ahora aplicaremos varios conceptos fundamentales de su programación y utilización como lo son la organización de datos con estructuras, lo cual nos va a permitir agrupar datos relacionados en un solo objeto, lo cual nos ayuda a mantener el código ordenado, además de que guardaremos datos en memoria y podremos consultarlos las veces que queramos mientras el programa este en uso, porque una vez cerrado esa información desaparecerá. Para poder guardar esa información hay diferentes métodos como en una base de datos de SQLite entre otras, archivos JSON, etc.

Fue un trabajo muy divertido de hacerlo y me da muchísimo mas conocimiento sobre esta grandiosa herramienta y lenguaje.

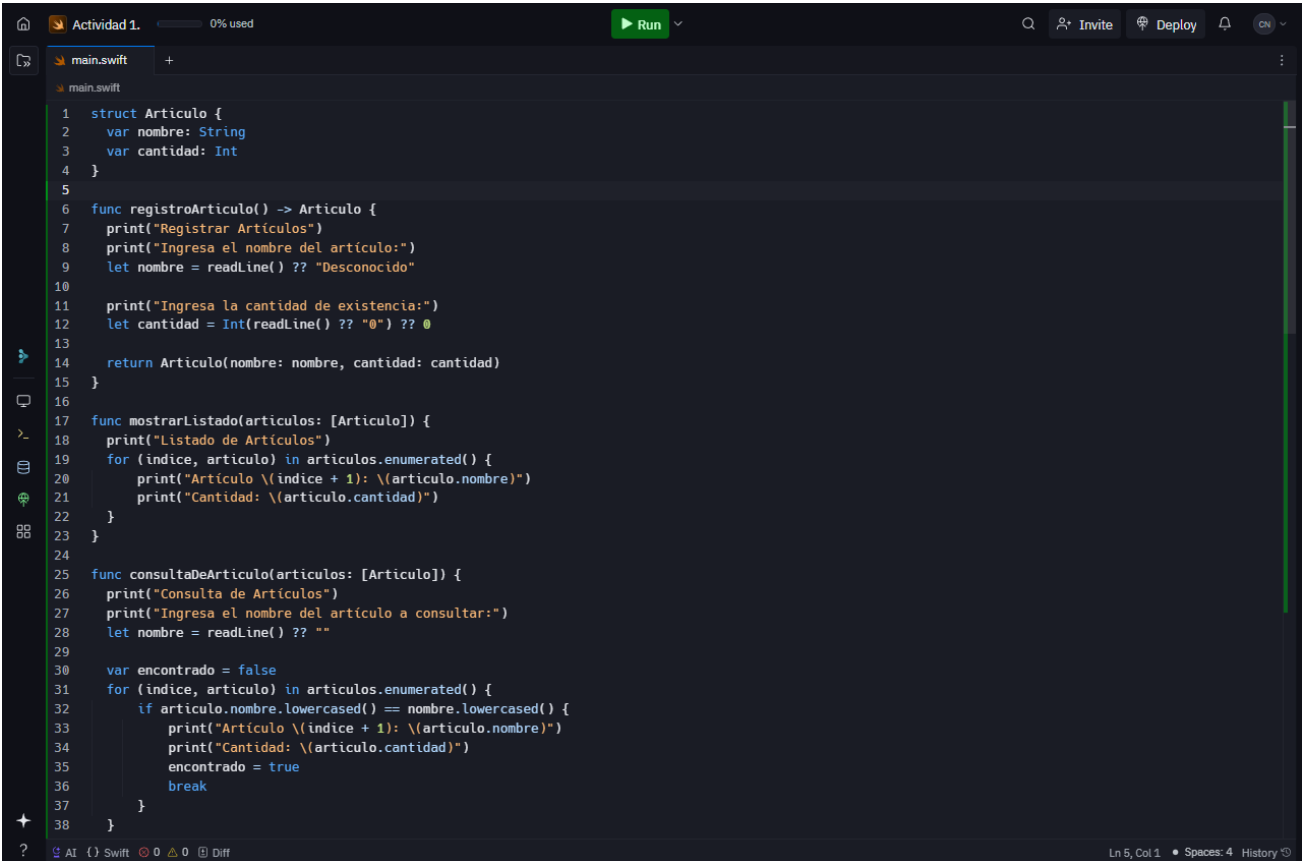
Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-III>

Desarrollo

Codificación

En esta parte del ejercicio realizamos una aplicación de inventario con ciertas funciones. A continuación se muestra el código realizado y una breve explicación:



```
1 struct Articulo {
2     var nombre: String
3     var cantidad: Int
4 }
5
6 func registroArticulo() -> Articulo {
7     print("Registrar Articulos")
8     print("Ingresa el nombre del articulo:")
9     let nombre = readLine() ?? "Desconocido"
10
11     print("Ingresa la cantidad de existencia:")
12     let cantidad = Int(readLine() ?? "0") ?? 0
13
14     return Articulo(nombre: nombre, cantidad: cantidad)
15 }
16
17 func mostrarListado(articulos: [Articulo]) {
18     print("Listado de Articulos")
19     for (indice, articulo) in articulos.enumerated() {
20         print("Articulo \(indice + 1): \(articulo.nombre)")
21         print("Cantidad: \(articulo.cantidad)")
22     }
23 }
24
25 func consultaDeArticulo(articulos: [Articulo]) {
26     print("Consulta de Articulos")
27     print("Ingresa el nombre del articulo a consultar:")
28     let nombre = readLine() ?? ""
29
30     var encontrado = false
31     for (indice, articulo) in articulos.enumerated() {
32         if articulo.nombre.lowercased() == nombre.lowercased() {
33             print("Articulo \(indice + 1): \(articulo.nombre)")
34             print("Cantidad: \(articulo.cantidad)")
35             encontrado = true
36             break
37         }
38     }
39 }
```

```

Actividad 1. 0% used
Run

main.swift
main.swift
38 }
39
40 if !encontrado {
41     print("Artículo no encontrado.")
42 }
43 }
44
45 // Variables globales
46 var articulos: [Articulo] = []
47 var opcion = 0
48
49 repeat {
50     print("""
51     \n--- Menú de inventario ---
52     1. Registro de artículo
53     2. Listado de artículos
54     3. Consulta de artículos
55     4. Salir del inventario
56     Selecciona una opción:
57     """)
58
59     if let input = readLine(), let seleccion = Int(input) {
60         opcion = seleccion
61
62         switch opcion {
63             case 1:
64                 let nuevoArticulo = registroArticulo()
65                 articulos.append(nuevoArticulo)
66                 print("*** Artículo registrado ***")
67
68             case 2:
69                 mostrarListado(articulos: articulos)
70
71             case 3:
72                 consultaDeArticulo(articulos: articulos)
73
74             case 4:
75                 print("Saliendo del programa...")
76         }
77     }
78 } while opcion != 4
79
80 } else {
81     print("Entrada inválida. Ingrese un número.")
82 }
83
84 } while opcion != 4

```

```

Actividad 1. 0% used
Run

main.swift
main.swift
59 if let input = readLine(), let seleccion = Int(input) {
60     opcion = seleccion
61
62     switch opcion {
63         case 1:
64             let nuevoArticulo = registroArticulo()
65             articulos.append(nuevoArticulo)
66             print("*** Artículo registrado ***")
67
68         case 2:
69             mostrarListado(articulos: articulos)
70
71         case 3:
72             consultaDeArticulo(articulos: articulos)
73
74         case 4:
75             print("Saliendo del programa...")
76
77         default:
78             print("Opción inválida. Elija opciones del 1 al 4.")
79     }
80 } else {
81     print("Entrada inválida. Ingrese un número.")
82 }
83
84 } while opcion != 4

```

A continuación, explicaremos lo que realiza el código:

-

```
1 struct Artículo {
2     var nombre: String
3     var cantidad: Int
4 }
```

Define una estructura llamada articulo con dos propiedades, nombre del tipo String y cantidad de tipo Int.

-

```
6 func registroArticulo() -> Artículo {
7     print("Registrar Artículos")
8     print("Ingresa el nombre del artículo:")
9     let nombre = readLine() ?? "Desconocido"
10
11     print("Ingresa la cantidad de existencia:")
12     let cantidad = Int(readLine() ?? "0") ?? 0
13
14     return Artículo(nombre: nombre, cantidad: cantidad)
15 }
```

Pide al usuario que ingrese un nombre para el articulo y la cantidad disponible. Si el usuario no ingresa nada el nombre será desconocido y la cantidad 0. Crea y devuelve un objeto Articulos con los datos ingresados.

-

```
17 func mostrarListado(articulos: [Artículo]) {
18     print("Listado de Artículos")
19     for (indice, articulo) in articulos.enumerated() {
20         print("Artículo \(indice + 1): \(articulo.nombre)")
21         print("Cantidad: \(articulo.cantidad)")
22     }
23 }
```

Recibe un arreglo de artículos y muestra cada uno, empezando con el articulo 1 y muestra el nombre y la cantidad ingresados. Utiliza enumerated() para tener acceso al índice del artículo.

-

```
25 func consultaDeArticulo(articulos: [Articulo]) {
26     print("Consulta de Artículos")
27     print("Ingresa el nombre del artículo a consultar:")
28     let nombre = readLine() ?? ""
29
30     var encontrado = false
31     for (indice, articulo) in articulos.enumerated() {
32         if articulo.nombre.lowercased() == nombre.lowercased() {
33             print("Artículo \(indice + 1): \(articulo.nombre)")
34             print("Cantidad: \(articulo.cantidad)")
35             encontrado = true
36             break
37         }
38     }
39
40     if !encontrado {
41         print("Artículo no encontrado.")
42     }
43 }
```

Permite consultar un artículo por su nombre (compara sin importar mayúsculas o minúsculas), si lo encuentra muestra su información, si no, muestra mensaje de Artículo no encontrado.

•

```
45 // Variables globales
46 var articulos: [Articulo] = []
47 var opcion = 0
48
49 repeat {
50     print("""
51     \n--- Menú de inventario ---
52     1. Registro de articulo
53     2. Listado de articulos
54     3. Consulta de articulos
55     4. Salir del inventario
56     Selecciona una opción:
57     """)
58
59     if let input = readLine(), let seleccion = Int(input) {
60         opcion = seleccion
61
62         switch opcion {
63             case 1:
64                 let nuevoArticulo = registroArticulo()
65                 articulos.append(nuevoArticulo)
66                 print("*** Artículo registrado ***")
67
68             case 2:
69                 mostrarListado(articulos: articulos)
70
71             case 3:
72                 consultaDeArticulo(articulos: articulos)
73
74             case 4:
75                 print("Saliendo del programa...")
76
77             default:
78                 print("Opción inválida. Elija opciones del 1 al 4.")
79         }
80     } else {
81         print("Entrada inválida. Ingresar un número.")
82     }
83
84 } while opcion != 4
```

Muestra un menú con 4 opciones. Lee la opción ingresada por el usuario (del 1 al 4) y ejecuta la acción correspondiente:

- 1- Registra un nuevo artículo en el inventario.
- 2- Muestra la lista de artículos registrados.

- 3- Permite consultar un artículo por nombre.
- 4- Termina el programa.

Si se agrega un numero invalido (que no sea del 1 al 4) o una letra, muestra un mensaje de error.

Prueba de la aplicación

A continuación, realizaremos las pruebas del aplicativo

Ingresamos dos registros de productos (Tenis y Camisetas) con la opción 1, además elegimos la opción 2 de listado de artículos donde podemos ver reflejados los artículos y sus cantidades ingresados al inventario.

The screenshot displays a Swift code editor with a file named 'main.swift'. The code defines a struct 'Articulo' with properties 'nombre' (String) and 'cantidad' (Int). It includes three functions: 'registroArticulo()' for adding items, 'mostrarListado(articulos: [Articulo])' for displaying the inventory, and 'consultaDeArticulo(articulos: [Articulo])' for searching by name. The console output on the right shows the program's execution: it starts with a menu, option 1 is selected to register 'Tenis' with a quantity of 6, then option 2 is selected to list items, showing 'Tenis' with quantity 6 and 'Camisetas' with quantity 4. Finally, option 3 is selected to search, and 'Camisetas' is found.

```

1 struct Articulo {
2     var nombre: String
3     var cantidad: Int
4 }
5
6 func registroArticulo() -> Articulo {
7     print("Registrar Articulos")
8     print("Ingresa el nombre del articulo:")
9     let nombre = readLine() ?? "Desconocido"
10
11     print("Ingresa la cantidad de existencia:")
12     let cantidad = Int(readLine() ?? "0") ?? 0
13
14     return Articulo(nombre: nombre, cantidad: cantidad)
15 }
16
17 func mostrarListado(articulos: [Articulo]) {
18     print("Listado de Articulos")
19     for (indice, articulo) in articulos.enumerated() {
20         print("Articulo \(indice + 1): \(articulo.nombre)")
21         print("Cantidad: \(articulo.cantidad)")
22     }
23 }
24
25 func consultaDeArticulo(articulos: [Articulo]) {
26     print("Consulta de Articulos")
27     print("Ingresa el nombre del articulo a consultar:")
28     let nombre = readLine() ?? ""
29
30     var encontrado = false
31     for (indice, articulo) in articulos.enumerated() {
32         if articulo.nombre.lowercased() == nombre.lowercased() {
33             print("Articulo \(indice + 1): \(articulo.nombre)")
34             print("Cantidad: \(articulo.cantidad)")
35             encontrado = true
36             break
37         }
38     }
39 }

```

Console Output:

```

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
1
Registrar Articulos
Ingresa el nombre del articulo:
Tenis
Ingresa la cantidad de existencia:
6
*** Artículo registrado ***

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
2
Listado de Articulos
Articulo 1: Tenis
Cantidad: 6
Articulo 2: Camisetas
Cantidad: 4

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
3
Ingresa el nombre del articulo a consultar:
Camisetas
Articulo 2: Camisetas
Cantidad: 4

```

Realizamos una consulta de artículo, en este caso ingresamos la palabra **Tenis** y nos muestra la cantidad que se tiene de dicho producto en el inventario.

```

1 struct Articulo {
2     var nombre: String
3     var cantidad: Int
4 }
5
6 func registroArticulo() -> Articulo {
7     print("Registrar Articulos")
8     print("Ingresa el nombre del artículo:")
9     let nombre = readline() ?? "Desconocido"
10
11     print("Ingresa la cantidad de existencia:")
12     let cantidad = Int(readline() ?? "0") ?? 0
13
14     return Articulo(nombre: nombre, cantidad: cantidad)
15 }
16
17 func mostrarListado(articulos: [Articulo]) {
18     print("Listado de Articulos")
19     for (indice, articulo) in articulos.enumerated() {
20         print("Artículo \(indice + 1): \(articulo.nombre)")
21         print("Cantidad: \(articulo.cantidad)")
22     }
23 }
24
25 func consultaDeArticulo(articulos: [Articulo]) {
26     print("Consulta de Artículos")
27     print("Ingresa el nombre del artículo a consultar:")
28     let nombre = readline() ?? ""
29
30     var encontrado = false
31     for (indice, articulo) in articulos.enumerated() {
32         if articulo.nombre.lowercased() == nombre.lowercased() {
33             print("Artículo \(indice + 1): \(articulo.nombre)")
34             print("Cantidad: \(articulo.cantidad)")
35             encontrado = true
36             break
37         }
38     }
39 }

```

Console Output:

```

--- Menú de inventario ---
1. Registro de artículo
2. Listado de artículos
3. Consulta de artículos
4. Salir del inventario
Selecciona una opción:
1
Registrar Artículos
Ingresa el nombre del artículo:
Camisetas
Ingresa la cantidad de existencia:
4
*** Artículo registrado ***

--- Menú de inventario ---
1. Registro de artículo
2. Listado de artículos
3. Consulta de artículos
4. Salir del inventario
Selecciona una opción:
2
Listado de Artículos
Artículo 1: Tennis
Cantidad: 6
Artículo 2: Camisetas
Cantidad: 4

--- Menú de inventario ---
1. Registro de artículo
2. Listado de artículos
3. Consulta de artículos
4. Salir del inventario
Selecciona una opción:
3
Consulta de Artículos
Ingresa el nombre del artículo a consultar:
Tennis
Artículo 1: Tennis
Cantidad: 6

--- Menú de inventario ---
1. Registro de artículo
2. Listado de artículos
3. Consulta de artículos
4. Salir del inventario
Selecciona una opción:

```

Ingresamos la opción 6, la cual no existe y nos envía el mensaje de opción invalida con el mensaje. Y por último seleccionamos la opción 4 y nos salimos del programa.

The screenshot shows a Swift IDE with a file named `main.swift` and a console window. The code defines a `struct Artículo` with `nombre` and `cantidad` properties. It includes functions for registering articles (`registroArticulo`), displaying a list (`mostrarListado`), and consulting an article (`consultaDeArticulo`). The console output shows the program's execution: it prompts for an article name and quantity, registers 'Camisetas' with a quantity of 4, displays a menu, lists the registered articles, and then shows an invalid option error when option 6 is selected.

```

1 struct Artículo {
2     var nombre: String
3     var cantidad: Int
4 }
5
6 func registroArticulo() -> Artículo {
7     print("Registrar Articulos")
8     print("Ingresa el nombre del artículo:")
9     let nombre = readLine() ?? "Desconocido"
10
11     print("Ingresa la cantidad de existencia:")
12     let cantidad = Int(readLine() ?? "0") ?? 0
13
14     return Artículo(nombre: nombre, cantidad: cantidad)
15 }
16
17 func mostrarListado(articulos: [Artículo]) {
18     print("Listado de Articulos")
19     for (indice, articulo) in articulos.enumerated() {
20         print("Artículo \(indice + 1): \(articulo.nombre)")
21         print("Cantidad: \(articulo.cantidad)")
22     }
23 }
24
25 func consultaDeArticulo(articulos: [Artículo]) {
26     print("Consulta de Articulos")
27     print("Ingresa el nombre del artículo a consultar:")
28     let nombre = readLine() ?? ""
29
30     var encontrado = false
31     for (indice, articulo) in articulos.enumerated() {
32         if articulo.nombre.lowercased() == nombre.lowercased() {
33             print("Artículo \(indice + 1): \(articulo.nombre)")
34             print("Cantidad: \(articulo.cantidad)")
35             encontrado = true
36             break
37         }
38     }
39 }

```

Console Output:

```

> swift main.s...
Ingresa el nombre del artículo:
Camisetas
Ingresa la cantidad de existencia:
4
*** Artículo registrado ***

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
2
Listado de Articulos
Artículo 1: Tennis
Cantidad: 6
Artículo 2: Camisetas
Cantidad: 4

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
3
Consulta de Articulos
Ingresa el nombre del artículo a consultar:
Tennis
Artículo 1: Tennis
Cantidad: 6

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:
6
Opción inválida. Elija opciones del 1 al 4.

--- Menú de inventario ---
1. Registro de articulo
2. Listado de articulos
3. Consulta de articulos
4. Salir del inventario
Selecciona una opción:

```

Conclusión

Este lenguaje de programación llamado Swift para la vida laboral puede abrir muchas puertas, ya que permite generar aplicaciones móviles seguras, con un código bien estructurado, el cual puede compartirse y puede ser ampliado por parte de otros equipos de trabajo lo cual lo convierte en muy colaborativo. Además, aunque es muy importante en la vida laboral, en nuestro día a día de la vida personal, puede ayudarnos muchísimo al poder crear apps como el inventario que hicimos en este trabajo para organizar los gastos de uno o también poder crear recordatorios personalizados de temas cotidianos.

Este lenguaje es muy sencillo y es muy buena idea que los jóvenes que empiezan en el mundo de la programación sean instruidos y enseñados en el puedan mejorar sus habilidades desde temprana edad.

Realizar este trabajo fue una experiencia divertida y de muchísimo valor, ya que la utilización constante hace que uno se acostumbre a la sintaxis y que resulte más fácil su programación.

Gracias profesora por sus explicaciones y haber compartido estos conocimientos

<https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-III>

Referencias

Mitra, M. (2016, February 25). 10 reasons to learn Swift programming language. Mantra Labs; Mantra Labs Pvt Ltd. <https://www.mantralabsglobal.com/blog/10-reasons-to-get-started-with-swift-programming-language/>