



Actividad [#2] - [Aplicación 2]

[Desarrollo de Aplicaciones Móviles III]

Ingeniería en Desarrollo de Software

Tutor: Sandra Luz Lara Dévora

Alumno: Alan David López Rojas

Fecha:/07/2023

Índice

Introducción.....	pág. 3
Descripción.....	pág. 4
Justificación.....	pág. 5
Desarrollo.....	pág. 6
Codificación.....	pág. 6
Prueba de aplicación.....	pág. 9
Conclusión.....	pág. 13
Bibliografía.....	pág. 14

Introducción

En el mundo empresarial y comercial, la gestión eficiente del inventario es fundamental para garantizar el éxito y la rentabilidad de cualquier negocio. Tener un control preciso de los productos disponibles, sus detalles y cantidades es esencial para tomar decisiones informadas y satisfacer las demandas de los clientes. En este contexto, la creación de una aplicación de inventario en lenguaje Swift se convierte en una herramienta valiosa para automatizar y optimizar este proceso.

La aplicación de inventario desarrollada en Swift proporcionará a los usuarios, ya sean empleados de una tienda, dueños de pequeñas empresas o emprendedores, una solución práctica y eficiente para administrar sus productos. Con esta aplicación, podrán registrar nuevos artículos, visualizar la lista completa de productos y consultar información detallada de cada uno de ellos.

Swift, como lenguaje de programación moderno y altamente versátil, ofrece un entorno propicio para el desarrollo de aplicaciones móviles y de escritorio. Su sintaxis clara y concisa, combinada con su capacidad de interoperabilidad con otras tecnologías de Apple, permite crear aplicaciones nativas de alta calidad y rendimiento.

Al utilizar Swift para crear una aplicación de inventario, se obtiene una solución robusta y confiable. La aplicación permitirá a los usuarios mantener un control preciso de su inventario, realizar un seguimiento de las existencias y garantizar una gestión eficiente de los productos. Esto, a su vez, se traduce en una mayor eficiencia operativa, reducción de pérdidas, mejor toma de decisiones y una experiencia más satisfactoria tanto para los usuarios como para los clientes.

Descripción

En el contexto de una tienda de la esquina, se requiere desarrollar una aplicación en lenguaje Swift para gestionar el inventario de productos. Esta aplicación permitirá a los empleados registrar, visualizar y consultar los artículos disponibles, proporcionando un control más efectivo y una mejor organización del inventario.

La actividad consiste en crear la aplicación de inventario utilizando Swift. La aplicación contará con un menú que ofrecerá opciones para realizar diversas acciones relacionadas con el inventario. Estas acciones incluirán registrar nuevos artículos, ver la lista completa de productos y realizar consultas específicas sobre un artículo en particular.

El objetivo principal de esta actividad es brindar a los empleados una herramienta práctica y eficiente para gestionar su inventario de manera más efectiva. La aplicación permitirá registrar nuevos artículos de forma rápida y sencilla, lo que garantizará que todos los productos disponibles estén debidamente registrados en el inventario.

Además, la posibilidad de visualizar la lista completa de artículos proporcionará a los empleados una visión general de todos los productos disponibles en la tienda. Esto les permitirá tener un panorama claro de los productos en existencia, incluyendo información relevante como el nombre, cantidad y precio.

La opción de consulta será útil para obtener detalles específicos sobre un artículo en particular. Los empleados podrán buscar y acceder rápidamente a la información detallada de un producto, como su descripción, precio y disponibilidad.

Justificación

La utilización de una aplicación de inventario en lenguaje Swift se presenta como una solución altamente beneficiosa y eficiente para la gestión del inventario en una tienda o negocio. La implementación de esta solución ofrece numerosas ventajas que mejoran significativamente el control y organización de los productos disponibles.

En primer lugar, la aplicación de inventario en Swift proporciona una mayor eficiencia y productividad en las tareas relacionadas con la gestión del inventario. Al contar con una interfaz intuitiva y amigable, facilita el registro de nuevos artículos de manera rápida y sencilla, evitando errores y omisiones en la actualización de los datos. Esto agiliza los procesos y optimiza el tiempo de los empleados, permitiéndoles dedicar más esfuerzo a otras actividades importantes.

Además, la aplicación de inventario en Swift permite una mejor organización y control de los productos disponibles. La visualización de la lista completa de artículos proporciona una visión general y ordenada de todos los productos en existencia, lo que facilita la planificación y toma de decisiones informadas. Los empleados pueden acceder rápidamente a información relevante como el nombre, cantidad y precio de los productos, lo que les permite realizar un seguimiento más preciso del inventario y evitar la falta de existencias.

Otra ventaja importante es la precisión de la información. La aplicación garantiza que los datos del inventario estén actualizados y libres de errores. Al registrar electrónicamente los detalles de cada artículo, se reducen los errores humanos y se evita la duplicación de datos. Esto proporciona una base sólida para la toma de decisiones y la generación de informes precisos sobre el estado del inventario.

La facilidad de consulta y análisis es otro aspecto destacado de esta solución. Los empleados pueden buscar y consultar información detallada sobre un artículo específico de manera rápida y eficiente. Esto facilita la atención al cliente, ya que los empleados pueden proporcionar información precisa y actualizada sobre los productos. Además, la aplicación permite generar informes y análisis a partir de los datos del inventario, lo que brinda una visión más completa y ayuda en la toma de decisiones estratégicas.

Desarrollo

❖ Codificación

```
import Foundation

struct Artículo {
    var nombre: String
    var cantidad: Int
}

func registrarArticulo() -> Artículo {
    print("Registrar artículo")
    print("Por favor, ingrese el nombre del artículo:")
    if let nombre = readLine(), !nombre.isEmpty {
        print("Ingrese la cantidad de artículos:")
        if let cantidadString = readLine(), let cantidad = Int(cantidadString) {
            return Artículo(nombre: nombre, cantidad: cantidad)
        }
    }
    print("Datos inválidos. El artículo no pudo ser registrado.")
    return Artículo(nombre: "", cantidad: 0)
}

func mostrarListaArticulos(articulos: [Artículo]) {
    print("Lista de artículos:")
    for (indice, articulo) in articulos.enumerated() {
```

```

        print("Artículo \((indice + 1): \((articulo.nombre)")
        print("Cantidad: \((articulo.cantidad)")
    }
}

func consultarArticulos(articulos: [Articulo]) {
    print("Artículos en existencia:")
    for (indice, articulo) in articulos.enumerated() {
        print("Artículo \((indice + 1): \((articulo.nombre)")
        print("Cantidad: \((articulo.cantidad)")
    }
}

func main() {
    var articulos: [Articulo] = []

    var opcion: Int?
    repeat {
        print("\nMENU")
        print("1. Registrar artículo")
        print("2. Ver lista de artículos")
        print("3. Consultar artículos en existencia")
        print("4. Salir")
        print("\nElige una opción (número):")

        if let input = readLine(), let numero = Int(input) {
            opcion = numero

            switch opcion {
            case 1:
                let nuevoArticulo = registrarArticulo()
                articulos.append(nuevoArticulo)
                print("\nArtículo registrado exitosamente.")
            case 2:
                mostrarListaArticulos(articulos: articulos)
            case 3:
                consultarArticulos(articulos: articulos)
            case 4:
                print("\n¡Hasta luego!")
            default:
                print("\nOpción inválida. Por favor, elige una opción válida.")
            }
        } else {
            print("\nOpción inválida. Por favor, elige una opción válida.")
        }
    } while opcion != 4
}

main()

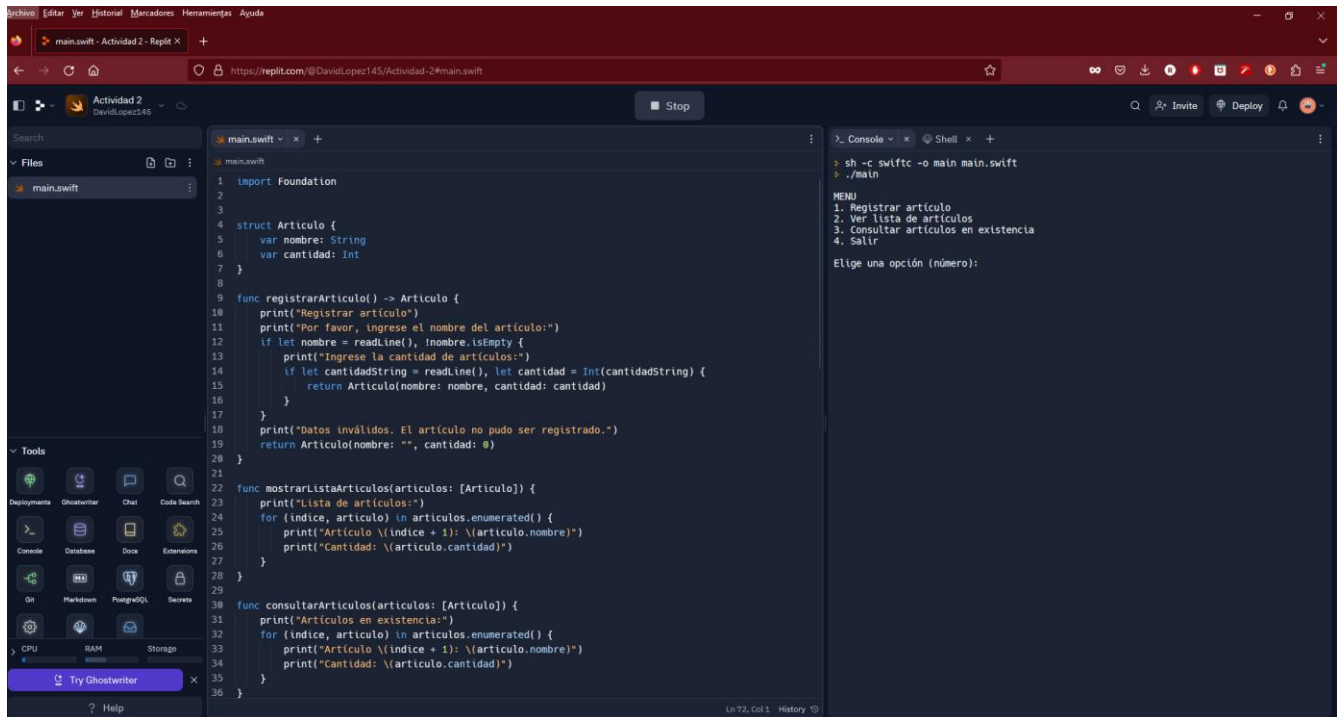
```

Este programa permite registrar, ver y consultar artículos en un inventario. Al iniciar, muestra un menú con las opciones disponibles. Dependiendo de la opción seleccionada, el programa ejecuta la función correspondiente. Al registrar un artículo, se solicita el nombre y la cantidad, y se agrega a la lista de artículos. Al ver la lista de artículos, se muestra el nombre y la cantidad de cada artículo registrado. Al consultar los artículos en existencia, se muestra el nombre y la cantidad de cada artículo registrado.

El programa se ejecuta hasta que se elija la opción de salir (4), momento en el cual finaliza la ejecución.

1. Primero, se define una estructura llamada **Articulo** que representa un artículo del inventario. Esta estructura tiene dos propiedades: **nombre** (String) para almacenar el nombre del artículo y **cantidad** (Int) para almacenar la cantidad disponible del artículo.
2. Luego, se definen varias funciones:
 - La función **registrarArticulo()** solicita al usuario ingresar el nombre y la cantidad de un artículo y devuelve un objeto **Articulo** con esos valores. Si los datos ingresados son válidos, se crea un nuevo objeto **Articulo** y se retorna. De lo contrario, se devuelve un objeto **Articulo** con valores vacíos y cantidad cero.
 - La función **mostrarListaArticulos(articulos:)** toma una lista de artículos y muestra en la consola el nombre y la cantidad de cada artículo.
 - La función **consultarArticulos(articulos:)** también toma una lista de artículos y muestra en la consola el nombre y la cantidad de cada artículo.
 - La función **main()** es la función principal del programa. Dentro de esta función, se crea una variable llamada **articulos** que almacena la lista de artículos del inventario.
3. Luego, se utiliza un bucle **repeat-while** para mostrar el menú y ejecutar la opción seleccionada hasta que se elija la opción de salir (4).
 - Dentro del bucle, se muestra el menú y se solicita al usuario que elija una opción. Si el usuario ingresa un número válido, se almacena en la variable **opcion**.
 - Se utiliza una estructura **switch** para manejar cada una de las opciones:
 - Si se elige la opción 1, se llama a la función **registrarArticulo()** para registrar un nuevo artículo. El artículo se agrega a la lista **articulos** y se muestra un mensaje de confirmación.
 - Si se elige la opción 2, se llama a la función **mostrarListaArticulos(articulos:)** para mostrar la lista de artículos.
 - Si se elige la opción 3, se llama a la función **consultarArticulos(articulos:)** para mostrar los artículos en existencia.
 - Si se elige la opción 4, se muestra un mensaje de despedida y el programa termina.
 - Si se elige cualquier otra opción, se muestra un mensaje de error.
4. El bucle se repite hasta que se elija la opción de salir (4), momento en el cual el programa finaliza su ejecución.

❖ Prueba de la aplicación

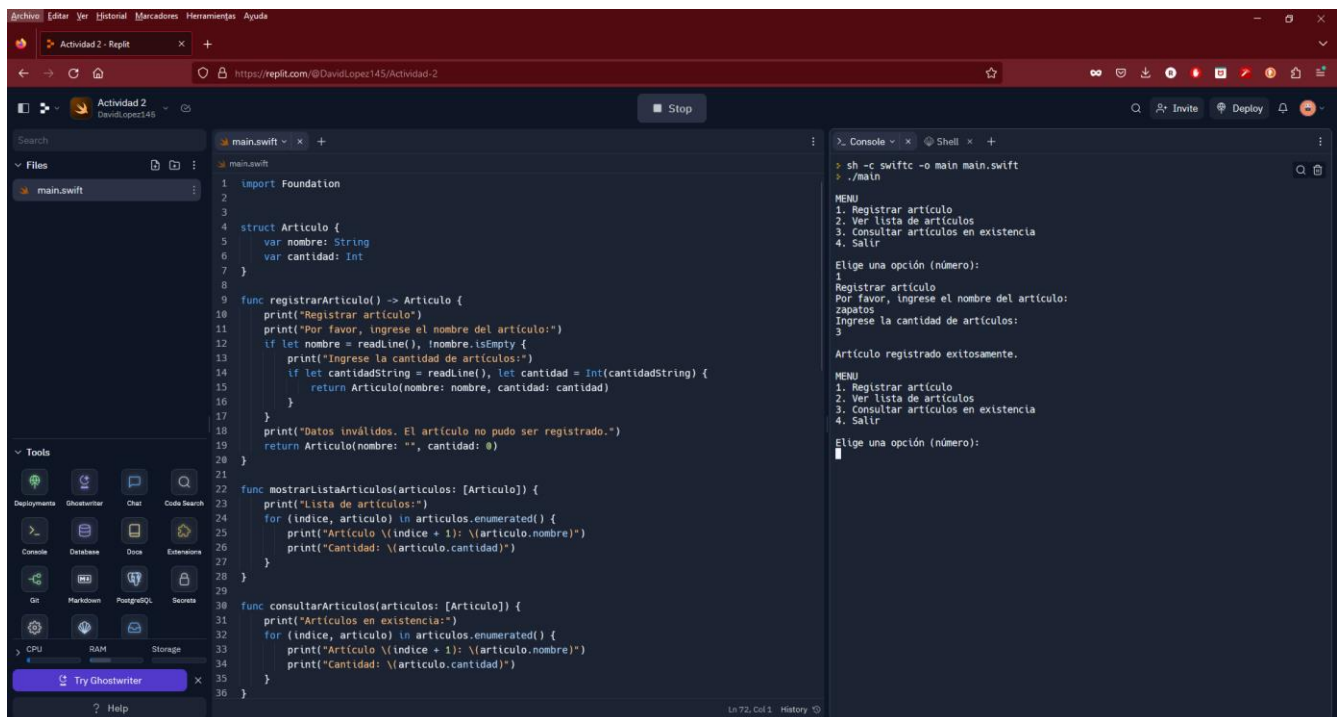


The screenshot shows a Replit environment with a Swift file named `main.swift`. The code defines a `struct` for articles and two functions: `registrarArticulo` and `mostrarListaArticulos`. The `registrarArticulo` function prompts the user for an article name and quantity, and the `mostrarListaArticulos` function lists the registered articles. The console shows the initial menu and the prompt to choose an option.

```
1 import Foundation
2
3
4 struct Articulo {
5     var nombre: String
6     var cantidad: Int
7 }
8
9 func registrarArticulo() -> Articulo {
10     print("Registrar artículo")
11     print("Por favor, ingrese el nombre del artículo:")
12     if let nombre = readline(), !nombre.isEmpty {
13         print("Ingrese la cantidad de artículos:")
14         if let cantidadString = readline(), let cantidad = Int(cantidadString) {
15             return Articulo(nombre: nombre, cantidad: cantidad)
16         }
17     }
18     print("Datos inválidos. El artículo no pudo ser registrado.")
19     return Articulo(nombre: "", cantidad: 0)
20 }
21
22 func mostrarListaArticulos(articulos: [Articulo]) {
23     print("Lista de artículos:")
24     for (indice, articulo) in articulos.enumerated() {
25         print("Artículo \(indice + 1): \(articulo.nombre)")
26         print("Cantidad: \(articulo.cantidad)")
27     }
28 }
29
30 func consultarArticulos(articulos: [Articulo]) {
31     print("Artículos en existencia:")
32     for (indice, articulo) in articulos.enumerated() {
33         print("Artículo \(indice + 1): \(articulo.nombre)")
34         print("Cantidad: \(articulo.cantidad)")
35     }
36 }
```

Console output:

```
> sh -c swiftc -o main main.swift
> ./main
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
```



The screenshot shows the same Replit environment after the first test run. The code is identical to the previous screenshot. The console output now shows the result of the first test run, where the user entered '1' to register an article, provided the name 'zapatos' and the quantity '3'. The output confirms that the article was registered successfully and shows the updated menu.

```
> sh -c swiftc -o main main.swift
> ./main
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
1
Registrar artículo
Por favor, ingrese el nombre del artículo:
zapatos
Ingrese la cantidad de artículos:
3
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
```

The screenshot shows a Swift Playground window titled "Actividad 2 - Regit" with the URL "https://replit.com/@DavidLopez145/Actividad-2". The code in the main.swift file is as follows:

```
1 import Foundation
2
3
4 struct Artículo {
5     var nombre: String
6     var cantidad: Int
7 }
8
9 func registrarArticulo() -> Artículo {
10     print("Registrar artículo")
11     print("Por favor, ingrese el nombre del artículo:")
12     if let nombre = readLine(), !nombre.isEmpty {
13         print("Ingrese la cantidad de artículos:")
14         if let cantidadString = readLine(), let cantidad = Int(cantidadString) {
15             return Artículo(nombre: nombre, cantidad: cantidad)
16         }
17     }
18     print("Datos inválidos. El artículo no pudo ser registrado.")
19     return Artículo(nombre: "", cantidad: 0)
20 }
21
22 func mostrarListaArticulos(articulos: [Artículo]) {
23     print("Lista de artículos:")
24     for (indice, articulo) in articulos.enumerated() {
25         print("Artículo \(indice + 1): \(articulo.nombre)")
26         print("Cantidad: \(articulo.cantidad)")
27     }
28 }
29
30 func consultarArticulos(articulos: [Artículo]) {
31     print("Artículos en existencia:")
32     for (indice, articulo) in articulos.enumerated() {
33         print("Artículo \(indice + 1): \(articulo.nombre)")
34         print("Cantidad: \(articulo.cantidad)")
35     }
36 }
```

The console output shows the initial menu:

```
sh -c swiftc -o main main.swift
./main
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
1
Registrar artículo
Por favor, ingrese el nombre del artículo:
zapatos
Ingrese la cantidad de artículos:
3
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
2
Lista de artículos:
Artículo 1: zapatos
Cantidad: 3
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
```

The screenshot shows the same Swift Playground window after several updates to the code. The main.swift file now includes:

```
1 import Foundation
2
3
4 struct Artículo {
5     var nombre: String
6     var cantidad: Int
7 }
8
9 func registrarArticulo() -> Artículo {
10     print("Registrar artículo")
11     print("Por favor, ingrese el nombre del artículo:")
12     if let nombre = readLine(), !nombre.isEmpty {
13         print("Ingrese la cantidad de artículos:")
14         if let cantidadString = readLine(), let cantidad = Int(cantidadString) {
15             return Artículo(nombre: nombre, cantidad: cantidad)
16         }
17     }
18     print("Datos inválidos. El artículo no pudo ser registrado.")
19     return Artículo(nombre: "", cantidad: 0)
20 }
21
22 func mostrarListaArticulos(articulos: [Artículo]) {
23     print("Lista de artículos:")
24     for (indice, articulo) in articulos.enumerated() {
25         print("Artículo \(indice + 1): \(articulo.nombre)")
26         print("Cantidad: \(articulo.cantidad)")
27     }
28 }
29
30 func consultarArticulos(articulos: [Artículo]) {
31     print("Artículos en existencia:")
32     for (indice, articulo) in articulos.enumerated() {
33         print("Artículo \(indice + 1): \(articulo.nombre)")
34         print("Cantidad: \(articulo.cantidad)")
35     }
36 }
```

The console output shows the application's state after multiple interactions:

```
Elige una opción (número):
1
Registrar artículo
Por favor, ingrese el nombre del artículo:
reloj
Ingrese la cantidad de artículos:
1
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
1
Registrar artículo
Por favor, ingrese el nombre del artículo:
laptop
Ingrese la cantidad de artículos:
5
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
3
Artículos en existencia:
Artículo 1: zapatos
Cantidad: 3
Artículo 2: reloj
Cantidad: 1
Artículo 3: laptop
Cantidad: 5
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
```

```
1 import Foundation
2
3
4 struct Artículo {
5     var nombre: String
6     var cantidad: Int
7 }
8
9 func registrarArticulo() -> Artículo {
10     print("Registrar artículo")
11     print("Por favor, ingrese el nombre del artículo:")
12     if let nombre = readLine(), !nombre.isEmpty {
13         print("Ingrese la cantidad de artículos:")
14         if let cantidadString = readLine(), let cantidad = Int(cantidadString) {
15             return Artículo(nombre: nombre, cantidad: cantidad)
16         }
17     }
18     print("Datos inválidos. El artículo no pudo ser registrado.")
19     return Artículo(nombre: "", cantidad: 0)
20 }
21
22 func mostrarListaArticulos(articulos: [Artículo]) {
23     print("Lista de artículos:")
24     for (indice, articulo) in articulos.enumerated() {
25         print("Artículo \(indice + 1): \(articulo.nombre)")
26         print("Cantidad: \(articulo.cantidad)")
27     }
28 }
29
30 func consultarArticulos(articulos: [Artículo]) {
31     print("Artículos en existencia:")
32     for (indice, articulo) in articulos.enumerated() {
33         print("Artículo \(indice + 1): \(articulo.nombre)")
34         print("Cantidad: \(articulo.cantidad)")
35     }
36 }
```

Por favor, ingrese el nombre del artículo:
reloj
Ingrese la cantidad de artículos:
1
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
1
Registrar artículo
Por favor, ingrese el nombre del artículo:
laptop
Ingrese la cantidad de artículos:
5
Artículo registrado exitosamente.
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
3
Artículos en existencia:
Artículo 1: zapatos
Cantidad: 3
Artículo 2: reloj
Cantidad: 1
Artículo 3: laptop
Cantidad: 5
MENU
1. Registrar artículo
2. Ver lista de artículos
3. Consultar artículos en existencia
4. Salir
Elige una opción (número):
4
¡Hasta luego!

<https://replit.com/join/pxizvefjje-davidlopez145>

Conclusión

En conclusión, la realización de la actividad de crear una aplicación de inventario en lenguaje Swift para gestionar el inventario de una tienda o negocio resulta de suma importancia tanto en el campo laboral como en la vida cotidiana.

En el ámbito laboral, contar con una aplicación de inventario eficiente y precisa facilita enormemente las tareas relacionadas con la gestión y control de los productos disponibles. Permite un registro rápido y preciso de nuevos artículos, una visualización ordenada y completa del inventario, y la posibilidad de consultar información detallada sobre cada producto. Esto se traduce en una mayor eficiencia operativa, ahorro de tiempo y recursos, y una toma de decisiones más informada. Además, la generación de informes y análisis a partir de los datos del inventario brinda una visión estratégica que contribuye al crecimiento y éxito del negocio.

En la vida cotidiana, la aplicación de inventario también resulta valiosa. Puede ser utilizada por pequeños comerciantes, emprendedores o incluso en el hogar, para mantener un control de los productos y evitar la escasez o exceso de los mismos. La aplicación facilita la organización y planificación de las compras, asegurando que los elementos necesarios estén siempre disponibles y evitando compras innecesarias o duplicadas.

Bibliografía

Stack Overflow Contributors, Aprendizaje Swift Language, 2019, Manual-informática, En línea:
<https://manual-informatica.com/download-file.html>