



**Actividad |2|****Aplicación 2.**

**Desarrollo de Aplicaciones Móviles III.**

Ingeniería en Desarrollo de Sofwtare.

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 Objetive-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 SQLilte entre otras, archivos JSON, etc.

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

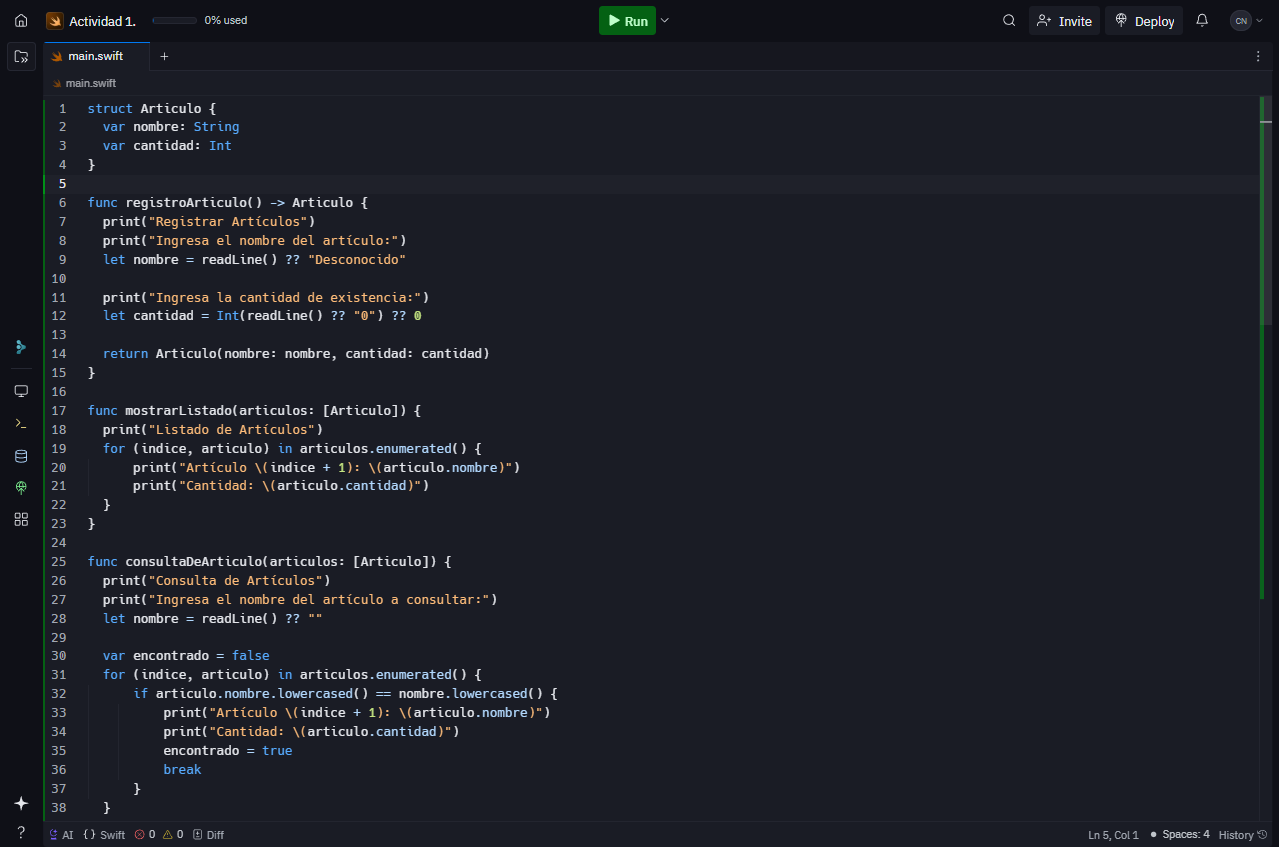
Este trabajo fue subido al siguiente enlace de GitHub

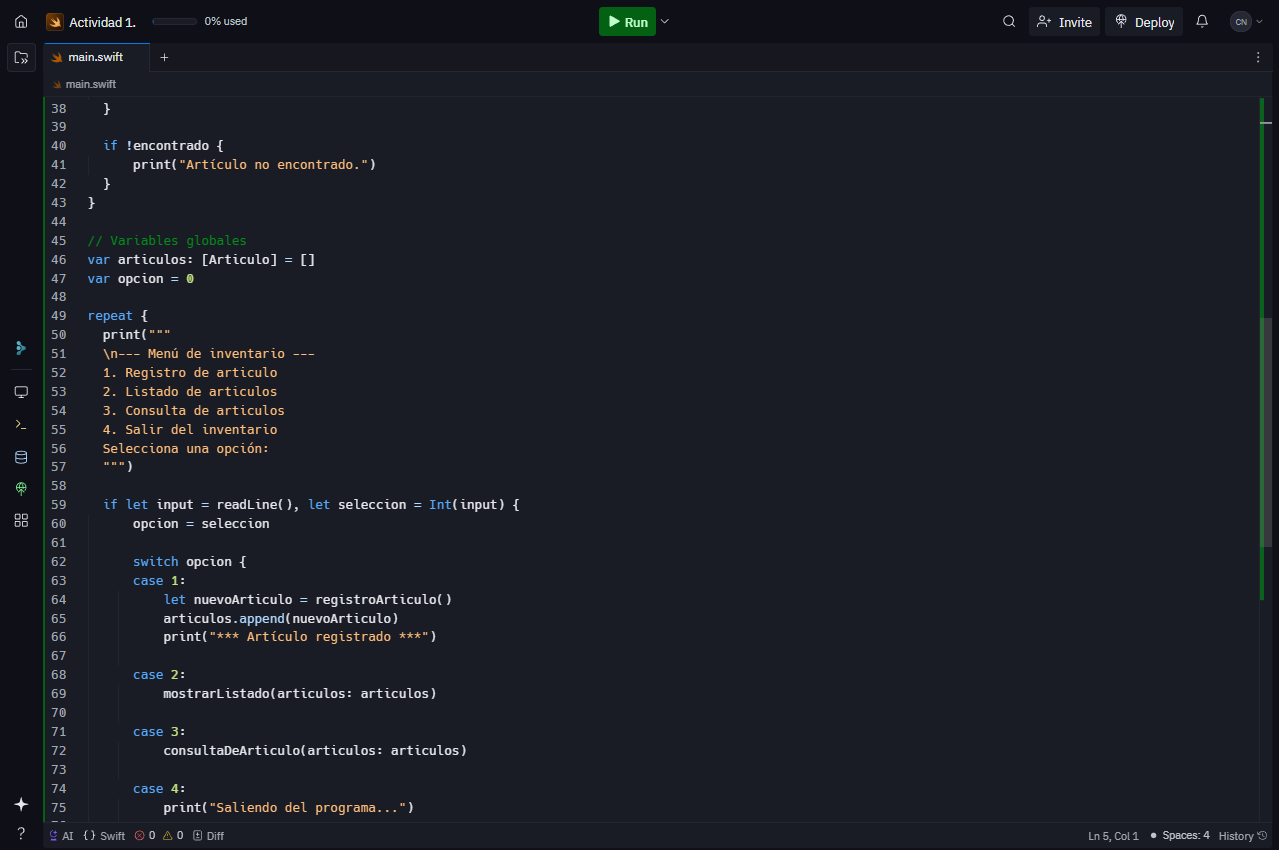
https://github.com/CarlosNico/Desarrollo-de-Aplicaciones-M-viles-III

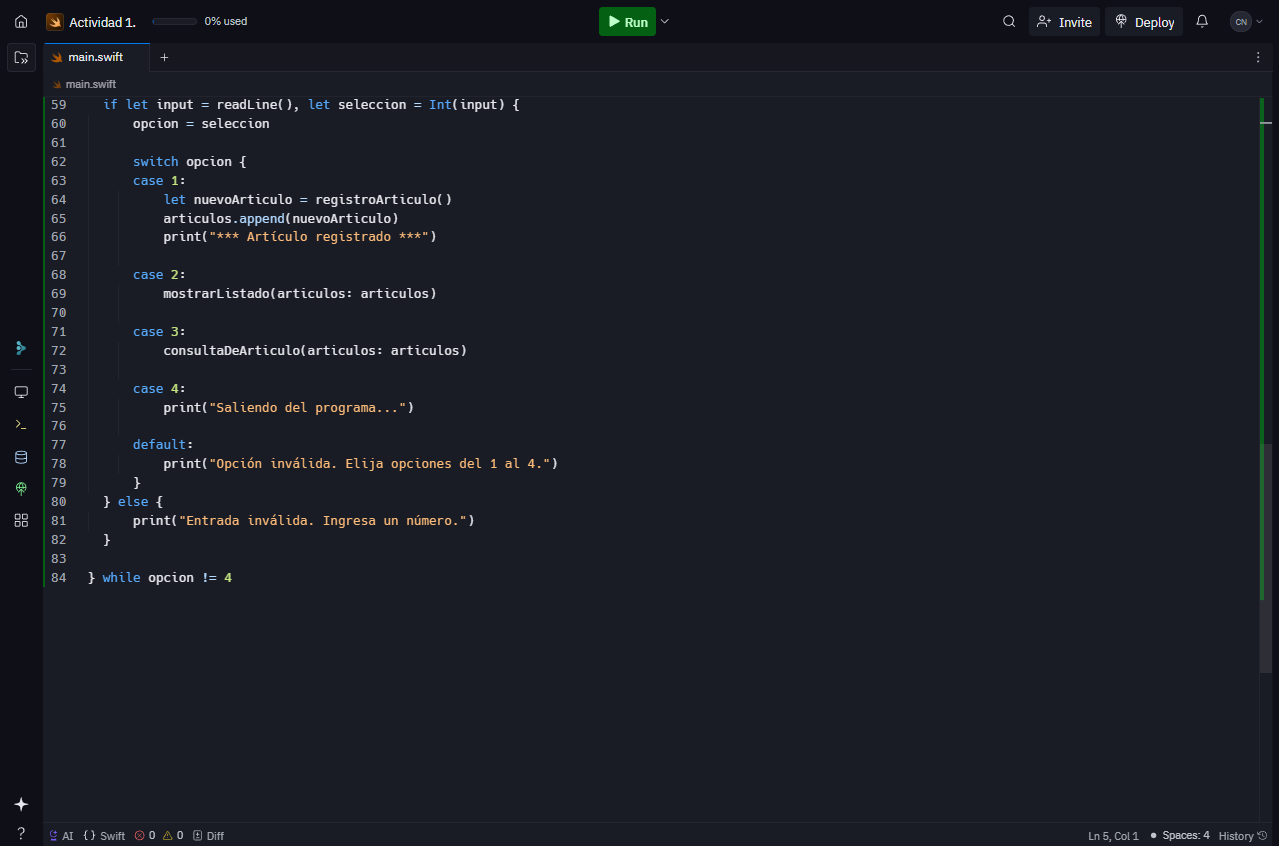
**Desarrollo**

**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:



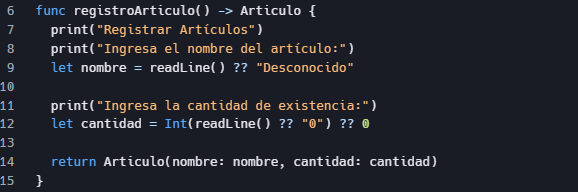




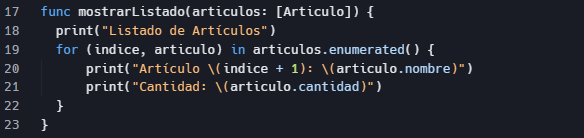
A continuación, explicaremos lo que realiza el código:



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



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.



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.



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



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

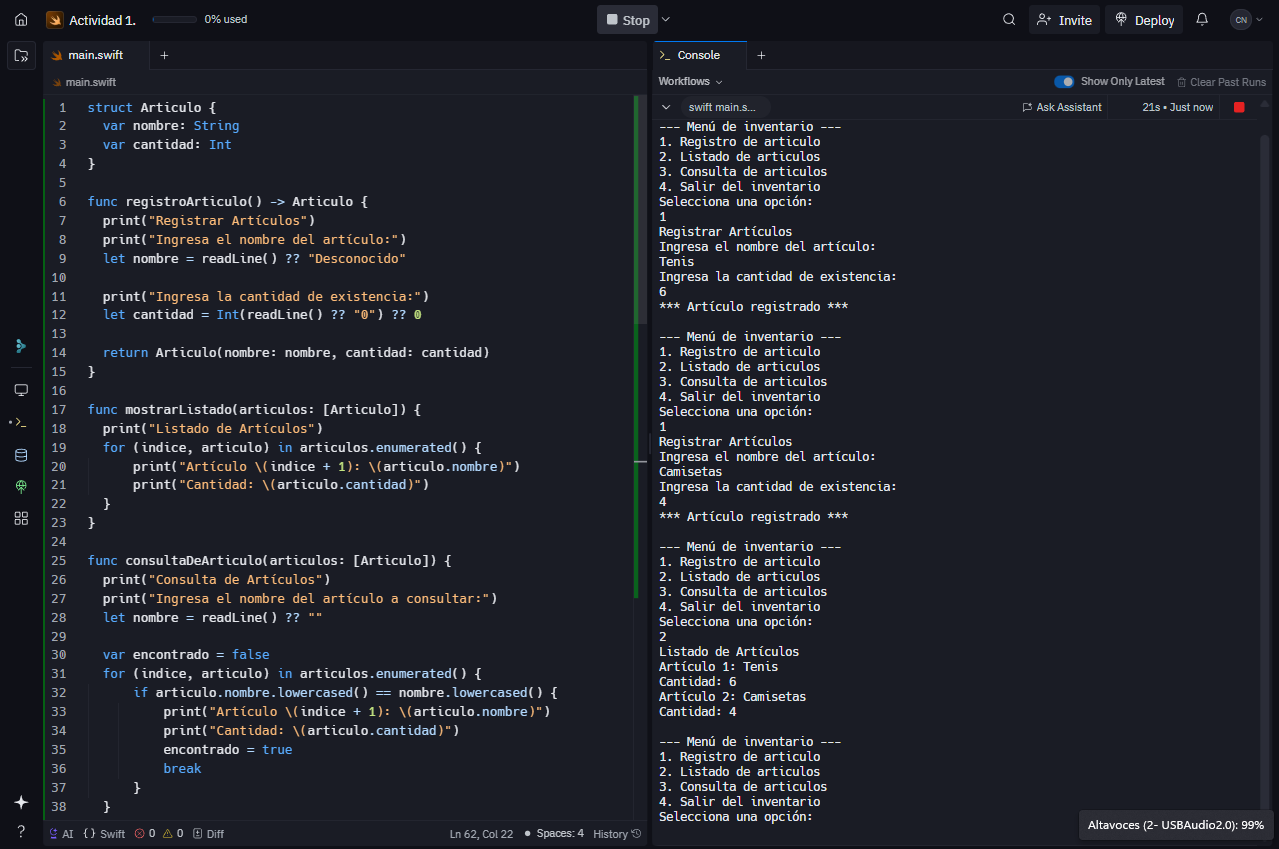
1. Registra un nuevo artículo en el inventario.
2. Muestra la lista de artículos registrados.
3. Permite consultar un artículo por nombre.
4. Termina el programa.

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

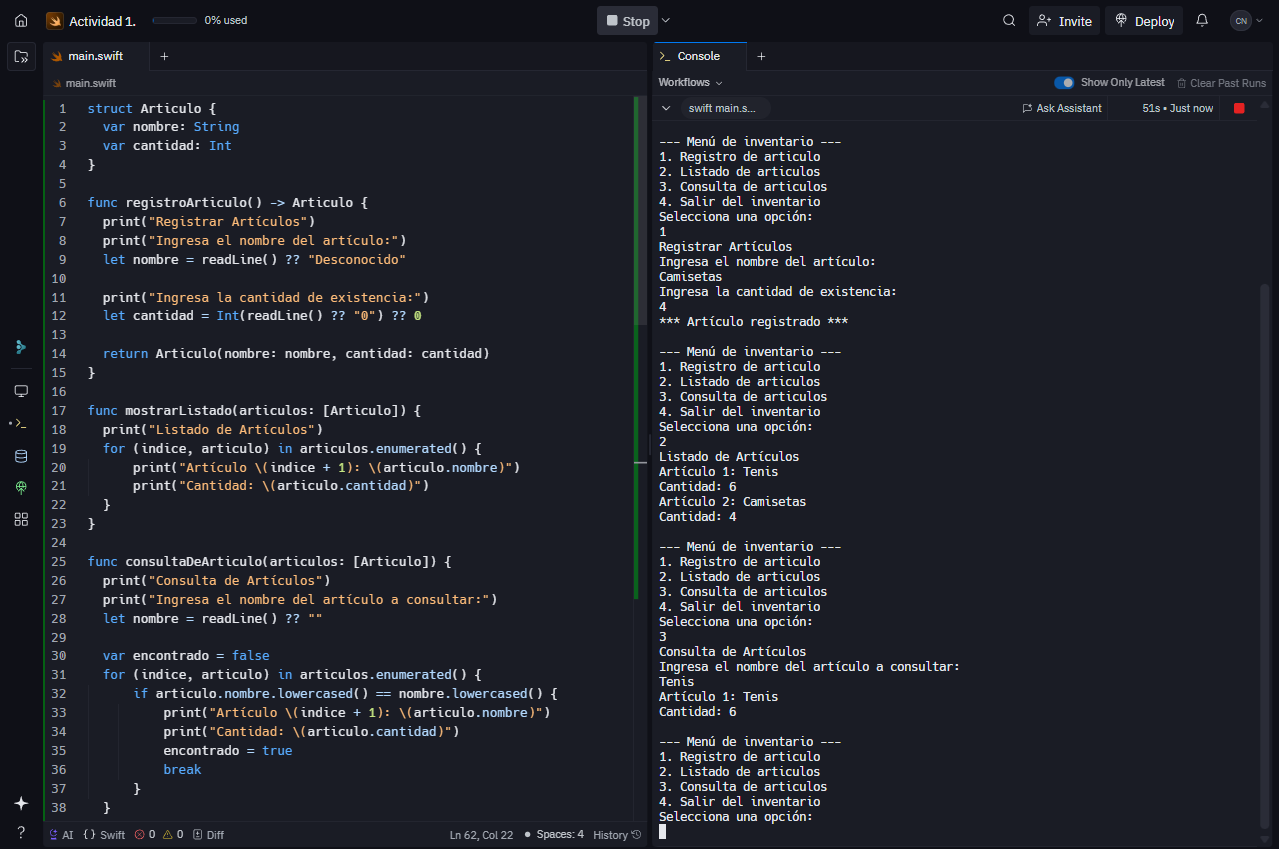
**Prueba de la aplicación**

A continuación, realizaremos las pruebas del aplicativo

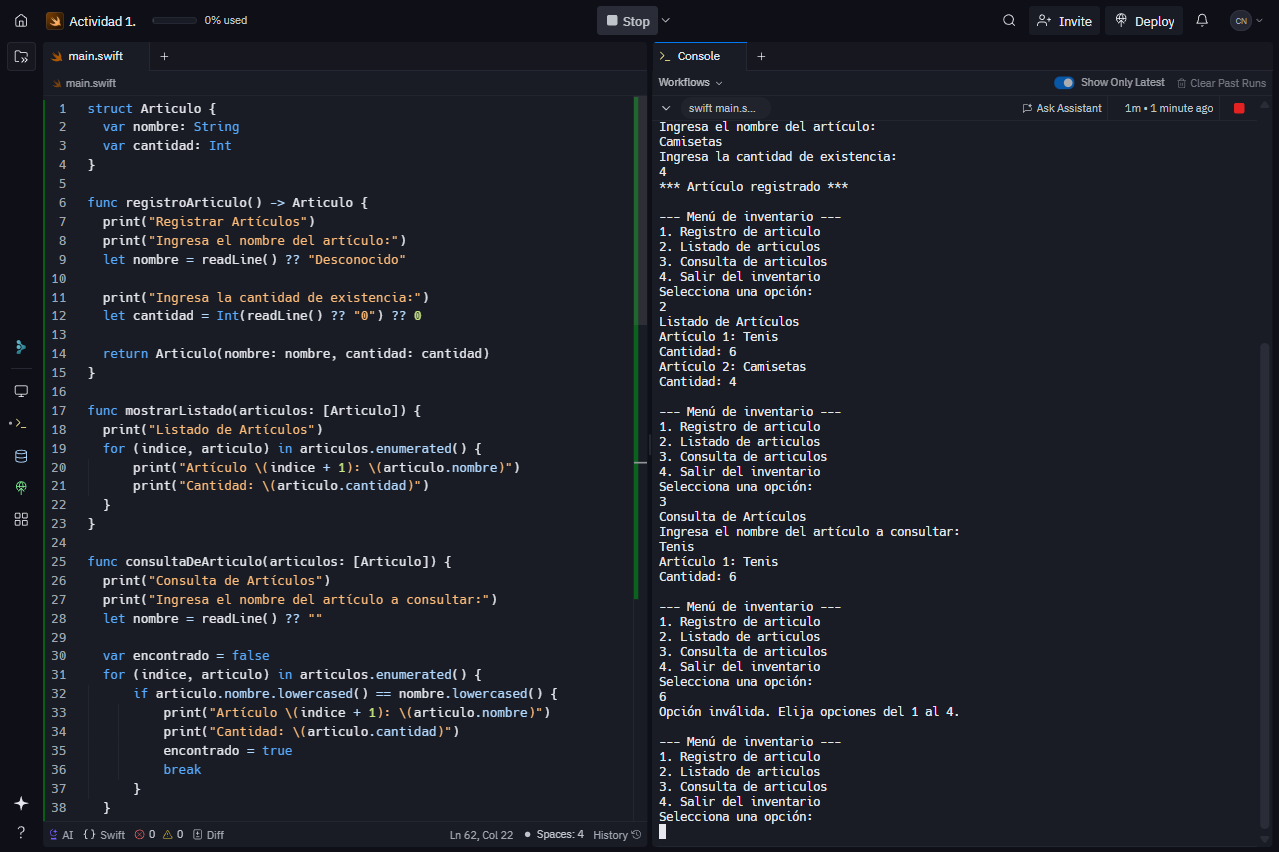
Ingresamos dos registros de productos (Tenis y Camisetas) con la opción 1, además elegimos la opción 2 de listado de artículos donde podemos ver reflejados los artículos y sus cantidades ingresados al inventario.



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



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.



**Conclusión**

El análisis estructurado visualiza un sistema desde la perspectiva de los

datos que fluyen a través de él. La función del sistema se describe

mediante procesos que transforman los flujos de datos.

El análisis estructurado aprovecha la ocultación de información a través

del análisis de descomposición sucesiva. Esto permite enfocar la

atención en los detalles pertinentes y evita la confusión al mirar detalles

irrelevantes. A medida que aumenta el nivel de detalle, se reduce la

amplitud de la información

El análisis estructurado visualiza un sistema desde la perspectiva de los

datos que fluyen a través de él. La función del sistema se describe

mediante procesos que transforman los flujos de datos.

El análisis estructurado aprovecha la ocultación de información a través

del análisis de descomposición sucesiva. Esto permite enfocar la

atención en los detalles pertinentes y evita la confusión al mirar detalles

irrelevantes. A medida que aumenta el nivel de detalle, se reduce la

amplitud de la información

En conclusión, los lenguajes se desarrollaron con el objetivo de ser más accesibles y

entendibles por la mayoría de programadores, de manera que los programadores

pudieran concentrarse más en resolver la tarea o los problemas y no en el lenguaje que

la maquina tenía que entender

En conclusión, los lenguajes se desarrollaron con el objetivo de ser más accesibles y

entendibles por la mayoría de programadores, de manera que los programadores

pudieran concentrarse más en resolver la tarea o los problemas y no en el lenguaje que

la maquina tenía que entender

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 mu 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

En conclusión, los lenguajes se desarrollaron con el objetivo de ser más accesibles y

entendibles por la mayoría de programadores, de manera que los programadores

pudieran concentrarse más en resolver la tarea o los problemas y no en el lenguaje que

la maquina tenía que entender

**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/*