Actividad |2| Inventario Aplicación 2.

Desarrollo de Aplicaciones Móviles III.

Ingeniería en Desarrollo de Software.

TUTOR: Sandra Luz Lara Devora.

ALUMNO: Ramón Ernesto Valdez Felix.

FECHA: 05/08/2025.

Introducción.

En esta segunda actividad de la materia de Desarrollo de Aplicaciones Móviles III, donde tenemos que realizar la documentación de la creación de una aplicación de un control de inventario eficiente que es crucial para el éxito. Para abordar esta necesidad, se desarrollará una aplicación intuitiva diseñada para optimizar la gestión de productos. Esta herramienta permitirá a los empleados registrar y visualizar fácilmente los artículos, asegurando un seguimiento preciso y un control superior sobre el stock disponible. La aplicación, que será desarrollada en Swift, ofrecerá una interfaz clara y funcional. Contará con un menú principal que facilitará diversas operaciones esenciales: la opción de registrar un nuevo artículo, la capacidad de ver la lista completa de artículos existentes, y la funcionalidad para consultar los artículos en existencia por nombre y su disponibilidad. Finalmente, incluirá una opción de salida para una navegación sencilla. Esta solución tecnológica busca simplificar las tareas diarias y mejorar significativamente la eficiencia operativa.

Descripción.

En esta segunda actividad de la materia de Desarrollo de Aplicaciones Móviles III, nos enfocamos en la documentación y creación de una aplicación de control de inventario. Un sistema eficiente es crucial para el éxito de cualquier tienda de la esquina, y para abordar esta necesidad, se desarrollará una aplicación intuitiva diseñada para optimizar la gestión de productos.

Esta herramienta permitirá a los empleados registrar y visualizar fácilmente los artículos, asegurando un seguimiento preciso y un control superior sobre el stock disponible. La aplicación, que será desarrollada en Swift, ofrecerá una interfaz clara y funcional. Contará con un menú principal que facilitará diversas operaciones esenciales: la opción de registrar un nuevo artículo, la capacidad de ver la lista completa de artículos existentes, y la funcionalidad para consultar los artículos en existencia por nombre y su disponibilidad. Finalmente, incluirá una opción de salida para una navegación sencilla. Esta solución tecnológica busca simplificar las tareas diarias y mejorar significativamente la eficiencia

operativa.

Justificación.

Este documento presenta una justificación para el desarrollo de una aplicación de control de inventario, esencial para la optimización de las operaciones en una tienda de la esquina. Un sistema de gestión de inventario robusto es fundamental para evitar pérdidas por caducidad, garantizar la disponibilidad de productos y mejorar la eficiencia en la reposición de stock. La App de Control de Inventario, desarrollada en Swift, aborda estas necesidades mediante una interfaz intuitiva y funcional.

La aplicación permitirá a los empleados registrar, visualizar y consultar artículos de manera ágil, lo que se traduce en un seguimiento preciso del stock y una reducción en los errores humanos. Con funciones clave como el registro de nuevos productos y la consulta por nombre, se simplifican las tareas diarias. El objetivo es proporcionar una herramienta tecnológica que mejore la eficiencia operativa y permita una toma de decisiones más informada, asegurando así el éxito y la competitividad de la tienda.

Entre otros puntos adicionales a utilizar en la justificación para la realizar de la documentación de esta actividad que son los siguientes:

- PDF de está actividad en el portafolio GitHub.
- Descarga el script genera de swift y subirlo como parte de la actividad a GitHub
- Anexa link de GitHub en documento.
- Utilizar la herramienta del entorno de trabajo online Replit o XCode que se usarán con lenguaje de programación: Swift.

Desarrollo.

En esta segunda actividad de la materia continuamos con el desarrollo de una aplicación de control de inventario, aquí tomaremos en cuenta que utilizaremos la consola de programación online Repli con el

cual utilizaremos el lenguaje de programación Swift. Esto será documentado en los puntos siguientes de este documento donde nuestro propósito central es garantizar una interpretación de manera correcta del desglose y uso del script de la aplicación a crear.

Link: GitHub. **Link: Script.**

Codificación.

En este punto de la actividad realizaremos un breve descripción de los bloques de líneas del script de la aplicación de inventario que permitirá el registro, almacenaje de los registros, ver los artículos y hacer consultas individuales de artículos o productos.

Codificacion de app.

Este bloque de código, escrito en Swift, define la lógica central para una aplicación de gestión de inventario.

Se estructura en dos partes principales:

La primera parte define una estructura llamada Product. Esta estructura actúa como un molde o plantilla para representar un artículo individual en el inventario. Contiene tres propiedades fundamentales:

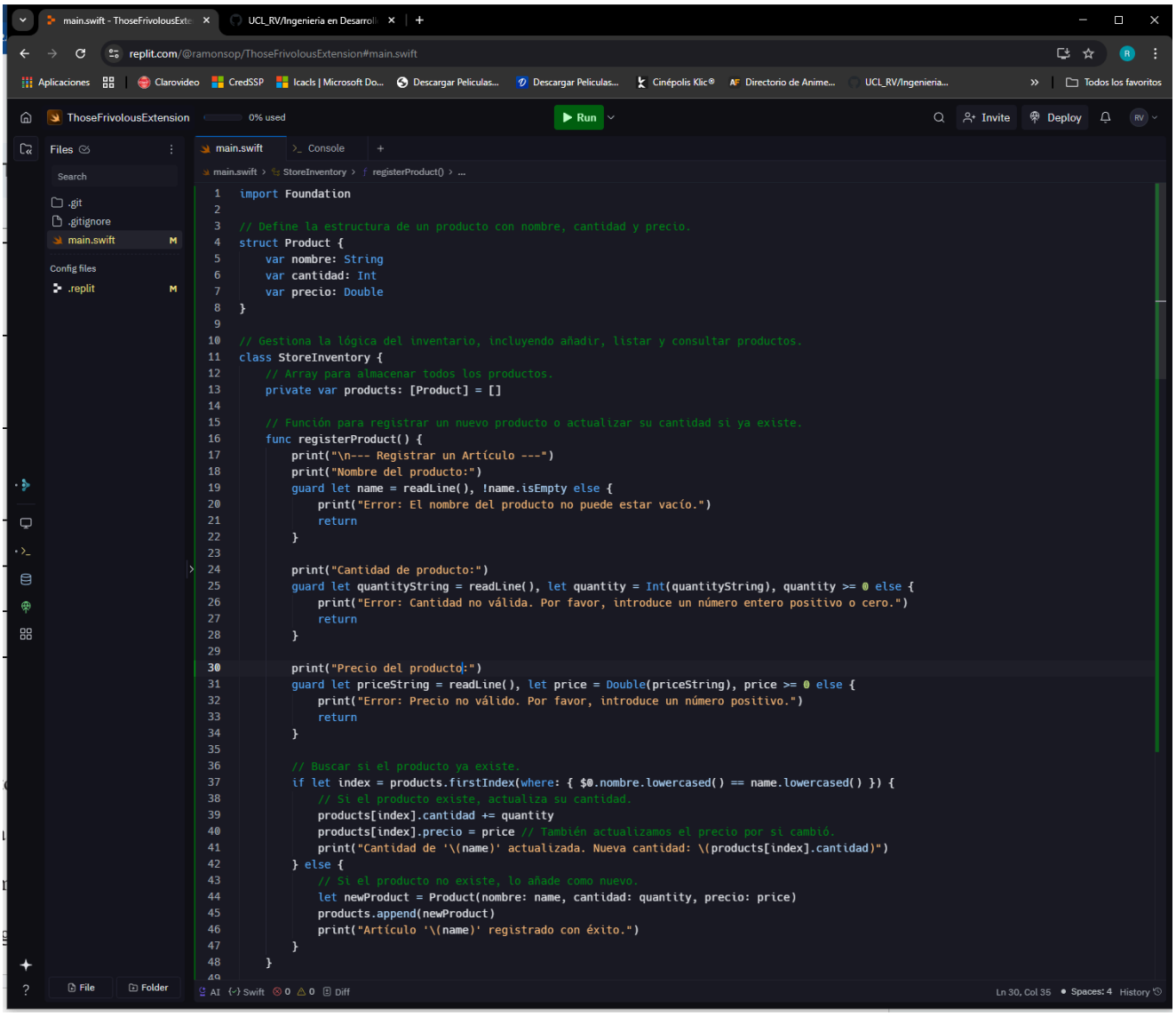
- nombre: Un String para el nombre del producto.
- cantidad: Un Int para la cantidad disponible de ese producto en stock.
- precio: Un Double para el precio de venta del producto.

2. Clase StoreInventory

La segunda parte es una clase llamada StoreInventory, que encapsula toda la funcionalidad para gestionar la lista de productos.

- Propiedad products: Es un arreglo privado de tipo [Product] que almacena todos los productos del inventario. La palabra clave private asegura que este arreglo solo se pueda modificar desde dentro de la misma clase.
- Función registerProduct(): Ésta es la función principal que maneja el registro de nuevos productos o la actualización de los existentes.
    - Solicita al usuario el nombre, la cantidad y el precio del producto a través de la consola (readLine()).
    - Utiliza la sentencia guard let para validar que las entradas del usuario sean válidas.
    - Busca si el producto ya existe en el inventario, ignorando mayúsculas y minúsculas (lowercased()).
    - Si el producto ya existe, actualiza su cantidad sumando la nueva cantidad ingresada y también actualiza el precio.
    - Si el producto no existe, crea una nueva instancia de la estructura Product con los datos proporcionados y la añade al arreglo products.

7



Este bloque de código Swift contiene tres funciones clave dentro de la clase StoreInventory, diseñadas para gestionar y visualizar la información del inventario de una tienda.

listAllProducts()

Esta función se encarga de mostrar todos los productos registrados en el inventario.

Verificación inicial: Primero, comprueba si el array products está vacío. Si lo está, imprime un mensaje indicando que no hay artículos y la función termina.

Ordenación: Si hay productos, los ordena alfabéticamente por su nombre ($0.nombre.lowercased() <

$1.nombre.lowercased()) para una presentación más clara y legible.

Iteración y visualización: Finalmente, recorre el array de productos ordenados y muestra en la consola el Nombre, la Cantidad y el Precio de cada artículo. El precio se formatea para mostrar siempre dos decimales ("%.2f").

consultProductByName()

Esta función permite a los usuarios buscar y consultar la información de un producto específico por su nombre.

Entrada del usuario: Solicita al usuario que introduzca el nombre del artículo a buscar.

Validación: Utiliza guard let para asegurar que el nombre introducido no esté vacío.

Búsqueda: Busca el producto en el array products utilizando first(where:). La búsqueda es insensible a mayúsculas y minúsculas (.lowercased()).

Resultados:

Si se encuentra el producto (foundProduct), imprime todos sus detalles (Nombre, Cantidad, Precio).

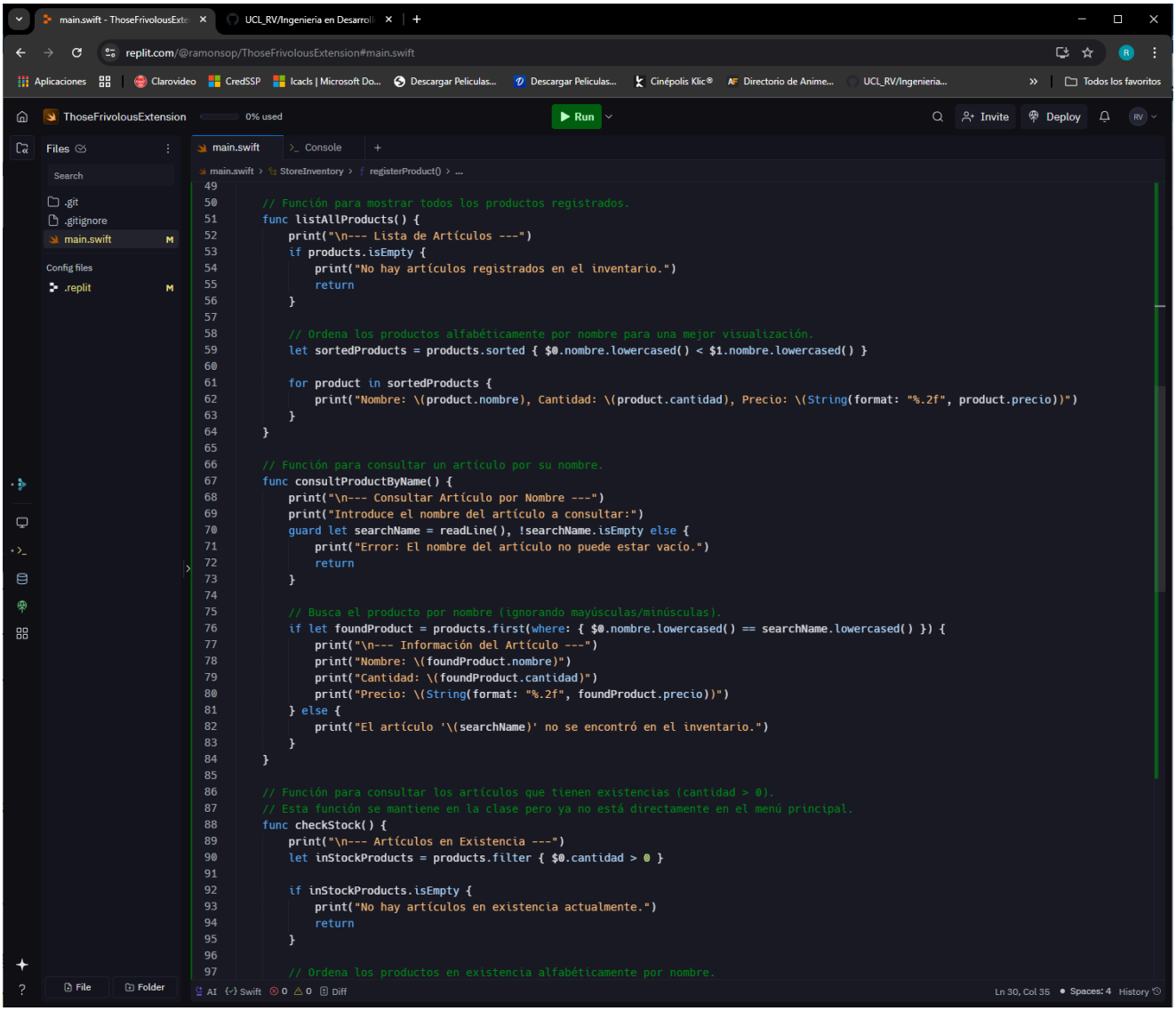
Si el producto no se encuentra, imprime un mensaje indicando que el artículo no existe en el inventario.

checkStock()

Esta función tiene como objetivo listar todos los productos que actualmente tienen existencias, es decir, una cantidad mayor que cero.

Filtrado: Utiliza el método filter para crear un nuevo array (inStockProducts) que solo contiene los productos cuya propiedad cantidad es mayor que 0.

Verificación: Comprobar si este nuevo array de productos en existencia está vacío. Si lo está, mprime un mensaje informando que no hay artículos en stock.



Este bloque de código, escrito en Swift, es la parte final de la aplicación de inventario. Contiene el cierre de una función y la lógica principal que ejecuta la aplicación completa. Se divide en dos secciones clave:

1. Cierre de la función checkStock()

Las primeras líneas corresponden a la parte final de la función checkStock(), que se encarga de

mostrar los productos en existencia.

Ordenación: La línea let sortedInStockProducts = inStockProducts.sorted { $0.nombre.lowercased() < $1.nombre.lowercased() } toma el array de productos que tienen existencias (inStockProducts) y los ordena alfabéticamente por nombre para una mejor visualización.

Visualización: El bucle for itera sobre el array ordenado y muestra en la consola el nombre, la cantidad y el precio de cada producto en stock. El precio se formatea a dos decimales.

2. Función runInventoryApp() y Bucle Principal

Esta es la sección más importante, ya que orquesta el flujo de toda la aplicación.

Inicialización: Se crea una instancia de la clase Stored Inventory llamada inventory, que será el objeto que gestiona toda la lógica del inventario. La variable running se inicia en true para controlar el bucle principal.

Bucle while: El bucle while running es el corazón de la aplicación. Mantiene el programa en ejecución hasta que el usuario decide salir.

Menú Principal: Dentro del bucle, se imprime un menú con las opciones disponibles:

Registrar un artículo: Llama a inventory.registerProduct().

Ver la lista de artículos: Llama a inventory.list All Products().

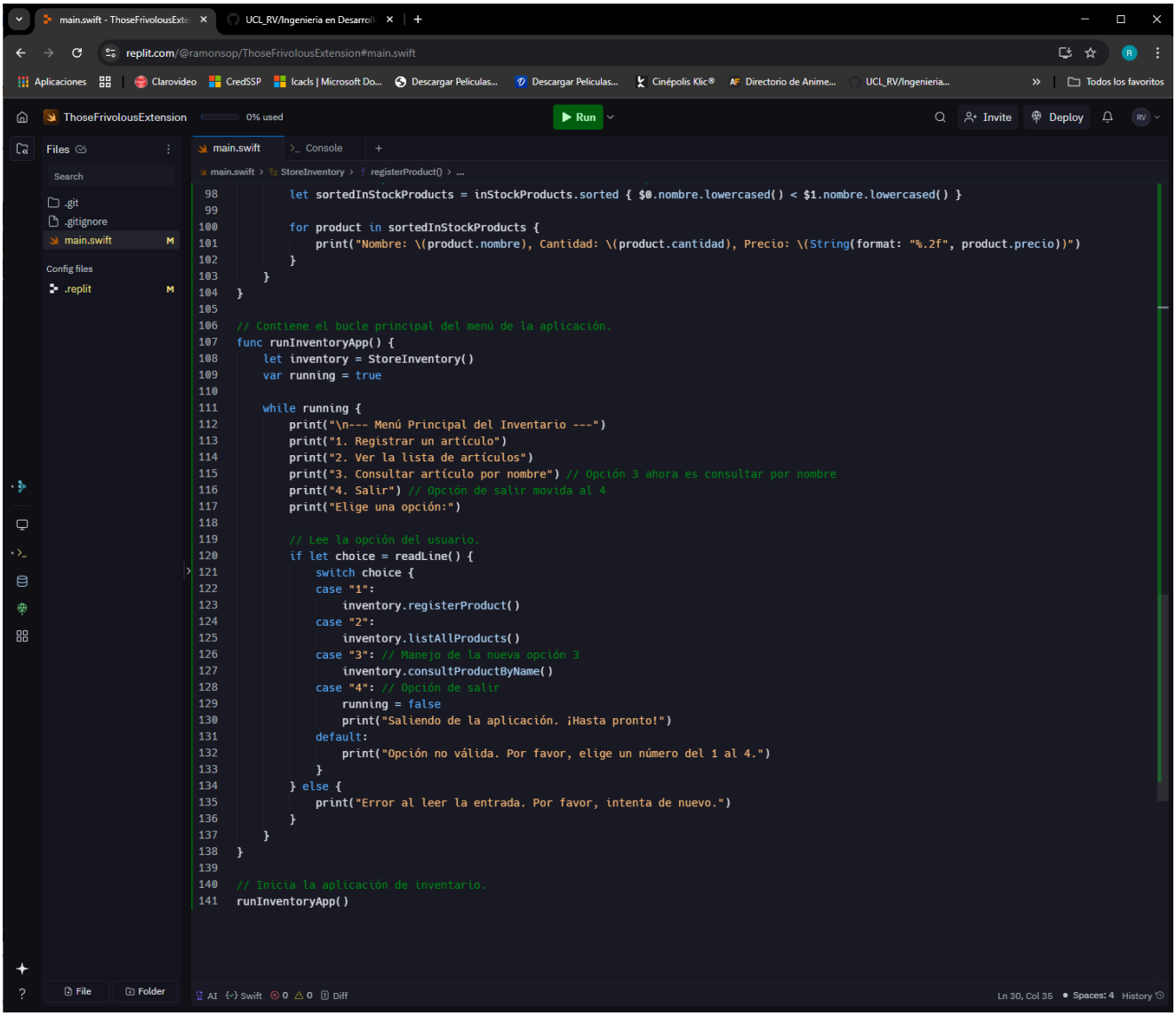
Consultar artículo por nombre: Llama a inventory.consult Product ByName().

Salir: Establece running en false para terminar el bucle y la aplicación.

Manejo de la entrada: El if let choice = readLine() lee la opción del usuario. El bloque switch evalúa la entrada y llama a la función correspondiente según el número elegido.

Mensajes de error: El default en el switch maneja las opciones no válidas, y el else del if maneja errores de lectura, proporcionando una experiencia más robusta para el usuario.

Llamada a la función: Finalmente, la línea runInventoryApp() inicia la ejecución de todo el programa.



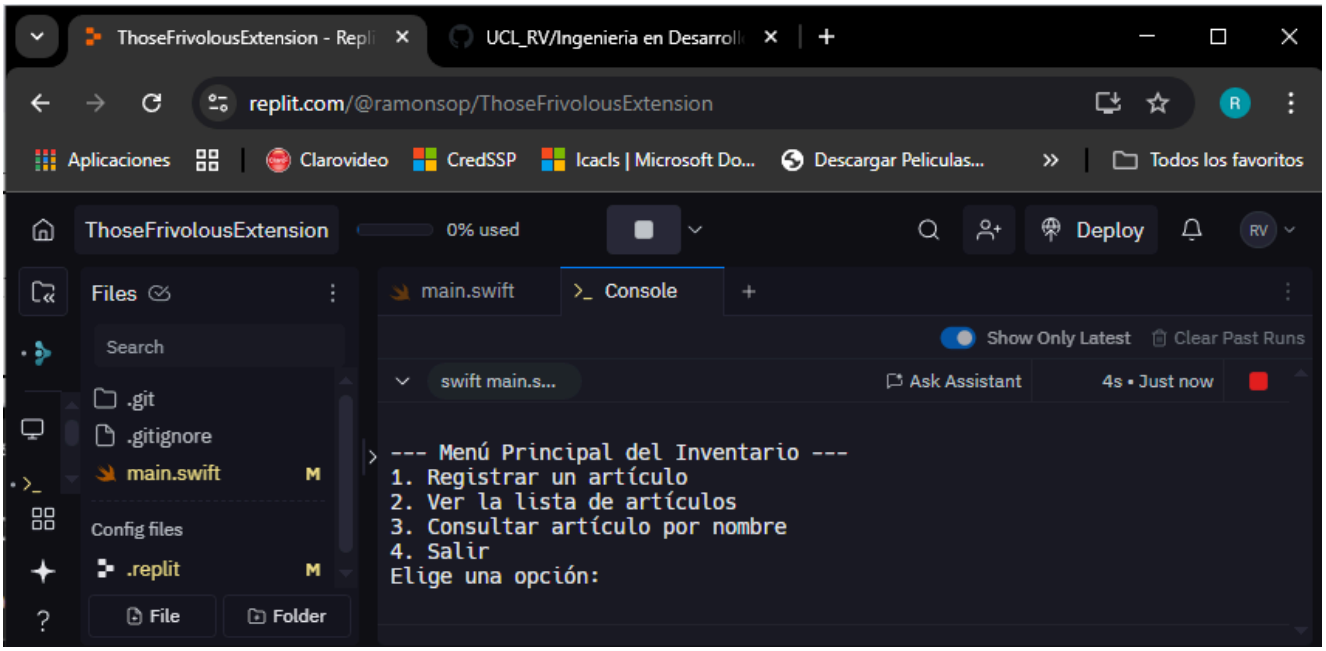
Prueba de la aplicación.

En este punto realizaremos la ejecución del código de la aplicación que mostrará el menú principal del inventario en el cual podremos registrar productos, ver el listado de productos, consultar algún producto y salir de la aplicación. Anexamos la evidencia de cada uno de los resultados agregando una screenshot

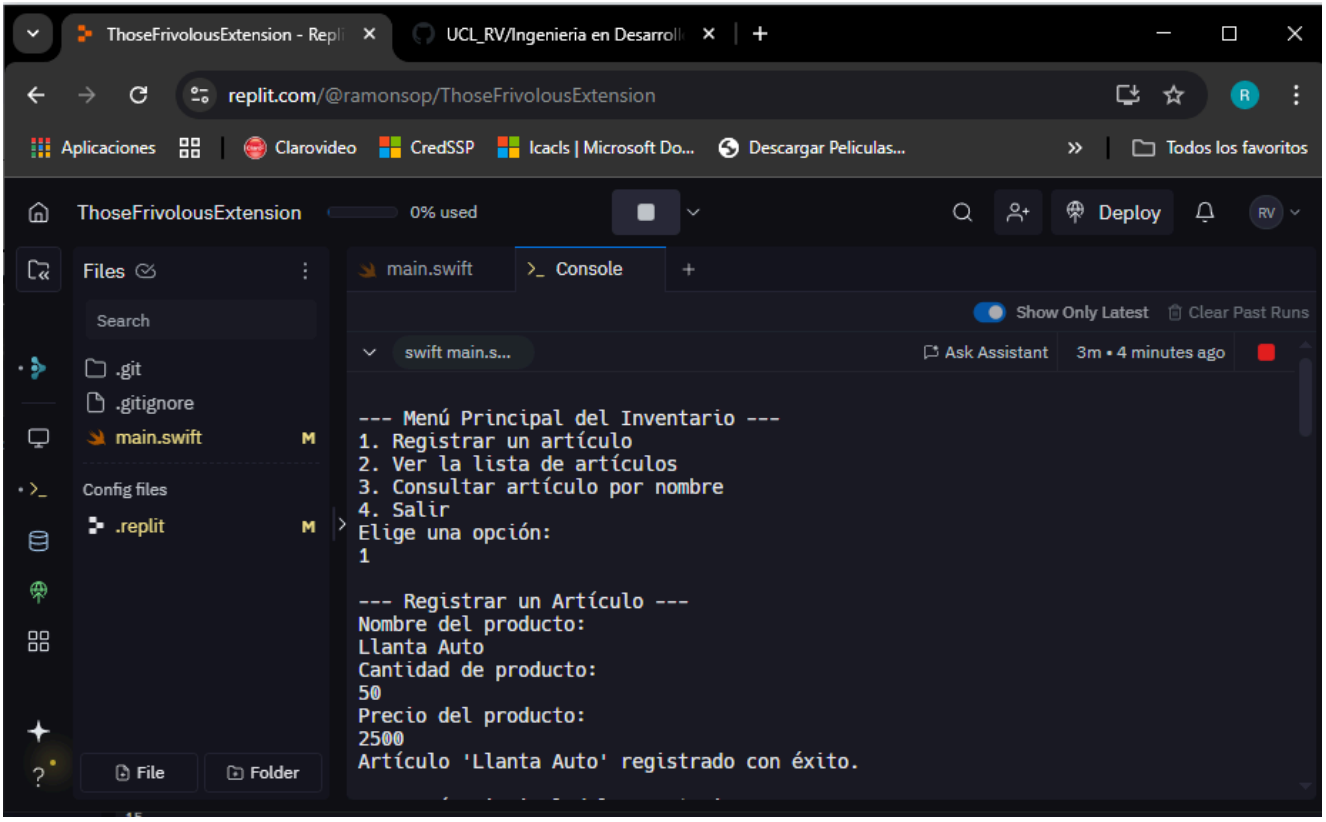
por resultado obtenido.

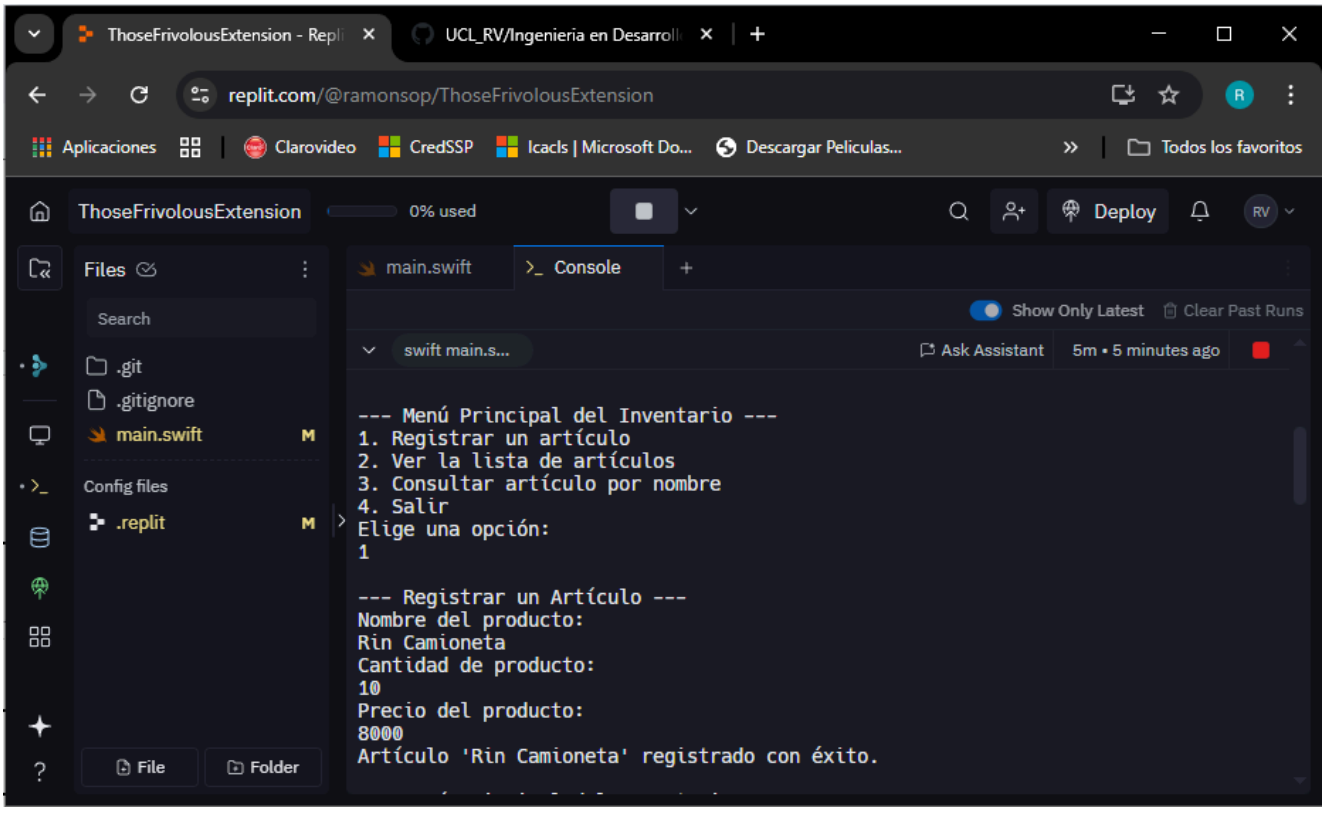
Pruebas de ejecución de la app:

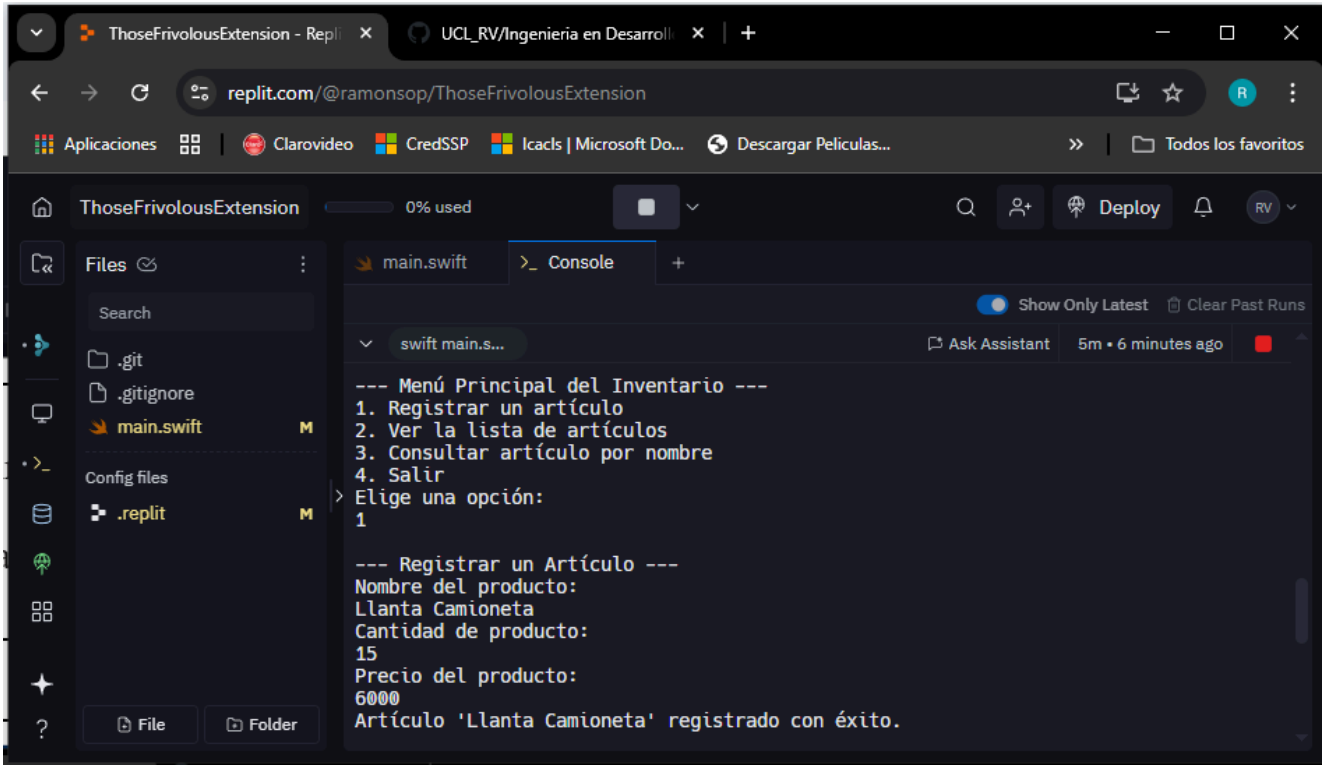
En esta pantalla nos muestra el menú de la aplicación que se creó para la actividad.

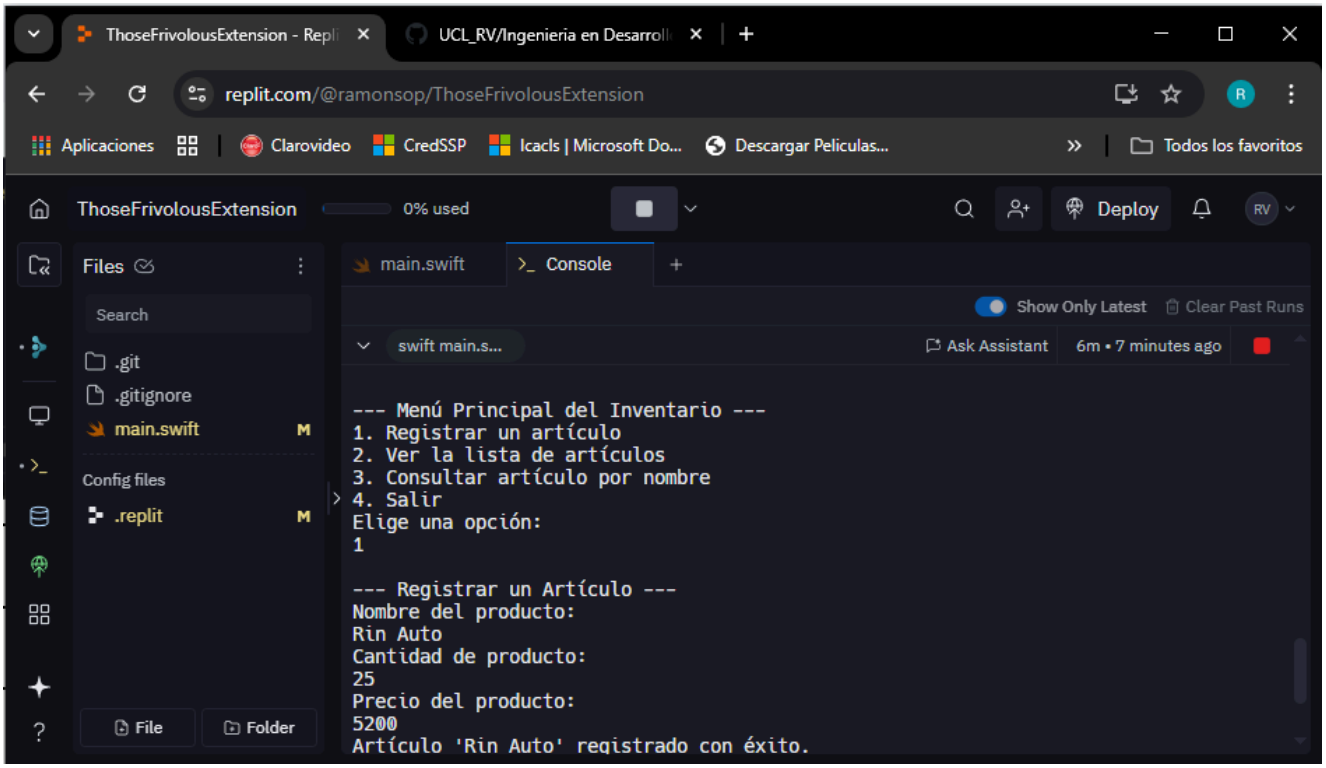


En estas 4 pantallas con las que continuaremos la actividad están dedicadas a la alta de 4 artículos para auto y camioneta. Anexamos las imágenes de evidencia.

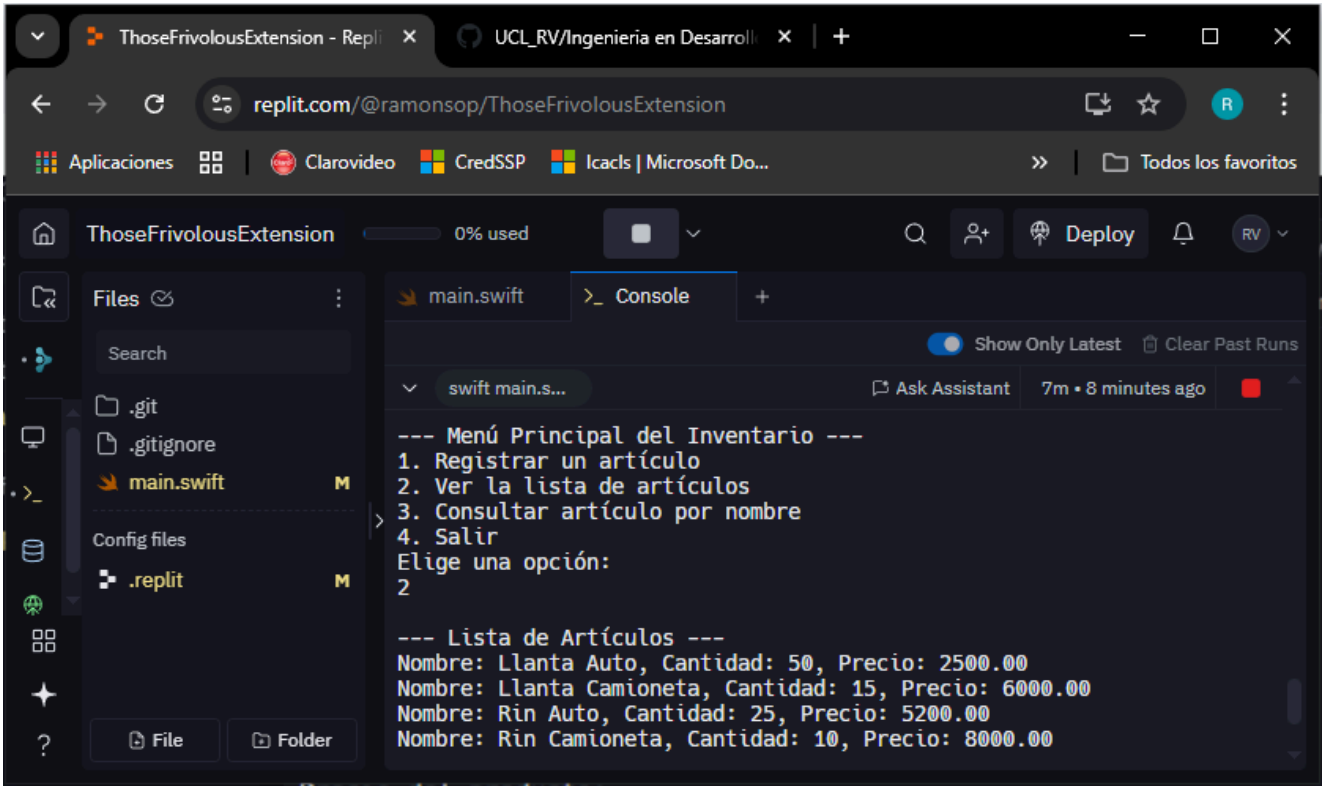

ThoseFrivolousExtension - Repl
UCL_RV/Ingenieria en Desarroll
replit.com/@ramonsop/ThoseFrivolousExtension
Aplicaciones
Clarovideo
CredSSP
Icacls | Microsoft Do...
Descargar Peliculas...
Todos los favoritos
ThoseFrivolousExtension
0% used
Deploy
RV
Files
Search
.git
.gitignore
main.swift
M
Config files
.replit
M
File
Folder
main.swift
Console
Show Only Latest
Clear Past Runs
swift main.s...
Ask Assistant
3m • 4 minutes ago
--- Menú Principal del Inventario ---
1. Registrar un artículo
2. Ver la lista de artículos
3. Consultar artículo por nombre
4. Salir
Elige una opción:
1
--- Registrar un Artículo ---
Nombre del producto:
Llanta Auto
Cantidad de producto:
50
Precio del producto:
2500
Artículo 'Llanta Auto' registrado con éxito.


ThoseFrivolousExtension - Repl
UCL_RV/Ingenieria en Desarroll
replit.com/@ramonsop/ThoseFrivolousExtension
Aplicaciones
Clarovideo
CredSSP
Icacls | Microsoft Do...
Descargar Peliculas...
Todos los favoritos
ThoseFrivolousExtension
0% used
Deploy
RV
Files
Search
.git
.gitignore
main.swift
M
Config files
.replit
M
File
Folder
main.swift
Console
Show Only Latest
Clear Past Runs
swift main.s...
Ask Assistant
5m • 5 minutes ago
--- Menú Principal del Inventario ---
1. Registrar un artículo
2. Ver la lista de artículos
3. Consultar artículo por nombre
4. Salir
Elige una opción:
1
--- Registrar un Artículo ---
Nombre del producto:
Rin Camioneta
Cantidad de producto:
10
Precio del producto:
8000
Artículo 'Rin Camioneta' registrado con éxito.

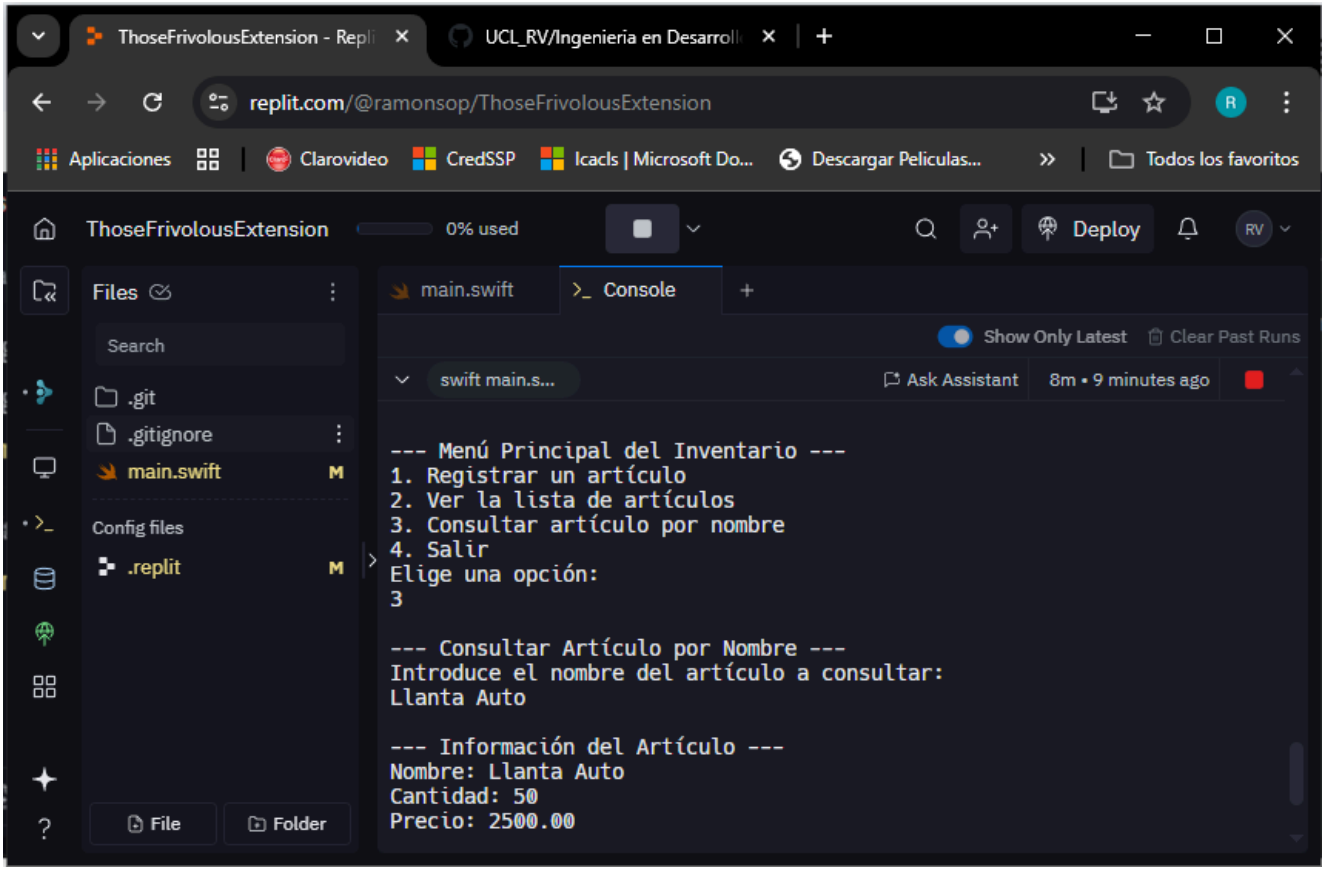




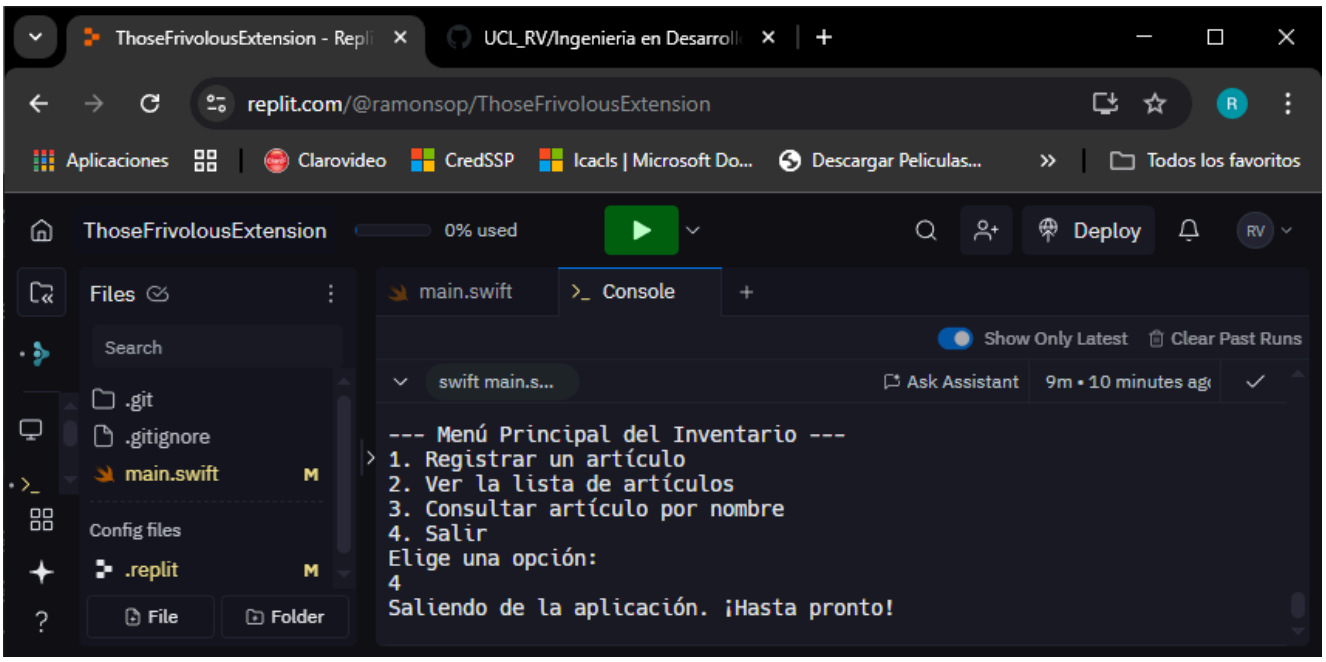
En esta pantalla utilizamos la opción 2 del menú que nos permite ver el listado de todos los artículos que fueron registrados con tres características: nombre, cantidad y precio. Anexando la imagen de evidencia



En esta pantalla utilizamos la opción 3 del menú que nos permite consultar un artículo por nombre y permite ver la información del artículo que fue consultado mostrando las tres características: nombre, cantidad y precio. Anexando la imagen de evidencia



En la última pantalla mostraremos la evidencia de la opción 4 del menú que nos permite salir de la aplicación.



Conclusion.

En conclusión: El desarrollo de esta aplicación de control de inventario en Swift representa un ejercicio crucial en la creación de herramientas prácticas que impactan directamente en la eficiencia operativa. La capacidad de registrar, listar y consultar productos de manera ágil es fundamental para cualquier negocio, grande o pequeño.

La relevancia de esta actividad va más allá de la simple codificación. A nivel laboral, demuestra la habilidad de traducir una necesidad real de un cliente para optimizar la gestión de stock en una solución tecnológica funcional y escalable. En la vida cotidiana, la lógica detrás de esta aplicación es aplicable a la organización personal, desde la gestión de una despensa hasta el control de activos del hogar. Comprender cómo un sistema puede automatizar y simplificar tareas manuales es una habilidad invaluable que mejora la productividad y minimiza errores, sentando las bases para proyectos de desarrollo de aplicaciones móviles más complejos y robustos en el futuro. Como dato personal y adicional, se me hizo muy fácil el poder realizar esta actividad debido a que la consola de programación es muy intuitiva y aporta mucho apoyo al desarrollador con el autoayuda o autocompletar al estar desarrollando alguna aplicación.

Referencias.

Gemini - chat to supercharge your ideas. (n.d.). Gemini. Retrieved January 9, 2025, from https://gemini.google.com/

Ingeniería en desarrollo de software. (n.d.). Edu.Mx. Retrieved January 9, 2025, from https://umi.edu.mx/coppel/IDS/login/index.php