

Actividad 2 - Aplicación 2.

Desarrollo de Aplicaciones Móviles 3.

Ingeniería en Desarrollo de Software

Tutor: Sandra Luz Lara Dévora.

Alumno: Homero Ramirez Hurtado.

Fecha: 18 de Julio del 2024.



Índice.

. Introducción.

. Descripción.

. Justificación.

. Desarrollo.

- Codificación.
- Prueba de la Aplicación.

. Conclusión.

. Referencias.



Introducción.

La tienda de la esquina necesita una aplicación que permita a los empleados llevar un control eficiente de su inventario. Esta aplicación debe ser fácil de usar y proporcionar las siguientes funcionalidades clave:

1. Registro de Artículos: Los empleados deben poder agregar nuevos productos al inventario. Esto incluye detalles como nombre del producto, cantidad disponible, precio y cualquier otra información relevante.
2. Visualización de la Lista de Artículos: La aplicación debe mostrar una lista completa de todos los productos en existencia. Los empleados podrán buscar, filtrar y ordenar esta lista según sus necesidades.
3. Consulta de Existencias: Los empleados deben poder verificar rápidamente cuántos artículos quedan en stock para cada producto. Esto les ayudará a tomar decisiones informadas sobre reabastecimiento.
4. Opción de Salir: Al finalizar su turno o tarea, los empleados podrán cerrar la aplicación de manera ordenada.

Tecnología y Lenguaje

Para desarrollar esta aplicación, utilizaremos el lenguaje Swift. Swift es un lenguaje moderno, seguro y eficiente para el desarrollo de aplicaciones en plataformas iOS, macOS y watchOS. Algunas de las ventajas de Swift son:

- Sintaxis Clara: Swift tiene una sintaxis más legible y concisa que otros lenguajes, lo que facilita la programación.
- Seguridad Integrada: Swift incluye características de seguridad, como la prevención de errores de acceso a memoria y la gestión automática de memoria.
- Compatibilidad con Plataformas Apple: Dado que la tienda de la esquina probablemente utilice dispositivos Apple (como iPads o iPhones), Swift es una elección natural.

Descripción.

La aplicación de inventario diseñada para la tienda de la esquina es una herramienta esencial para optimizar la gestión de productos:

1. Registro de Artículos:
 - Los empleados pueden agregar nuevos productos al inventario mediante un formulario intuitivo. Ingresarán detalles como nombre, descripción, precio y cantidad disponible.
 - La aplicación valida la entrada para evitar errores y garantizar datos precisos.
2. Lista de Artículos:
 - La pantalla principal muestra una lista completa de todos los productos almacenados.
 - Los empleados pueden buscar artículos específicos, filtrar por categoría o ver todo el inventario de un vistazo.

3. Consulta de Existencias:

- Al seleccionar un artículo, se muestra su información detallada, incluyendo la cantidad actual en stock.
- Los umbrales de existencias mínimas pueden destacarse para alertar a los empleados sobre posibles reabastecimientos.

4. Salida Ordenada:

- Cuando los empleados finalizan su turno o tarea, pueden cerrar la aplicación de manera segura.
- La aplicación guarda automáticamente los cambios realizados.

Beneficios de Swift

El uso del lenguaje Swift ofrece ventajas significativas:

1. Eficiencia: Swift está diseñado para un rendimiento óptimo, lo que garantiza una aplicación ágil y receptiva.
2. Seguridad: Su sintaxis moderna y características de seguridad integradas reducen errores y vulnerabilidades.
3. Compatibilidad con Plataformas Apple: Swift es la elección natural para dispositivos iOS y macOS.

Justificación.

Optimización de la Gestión

La implementación de una aplicación de inventario en la tienda de la esquina es crucial por varias razones:

1. Eficiencia Operativa:

- La gestión manual del inventario es propensa a errores y consume tiempo valioso. Una aplicación automatizada agiliza las tareas de registro, seguimiento y consulta.
- Los empleados podrán dedicar más tiempo a atender a los clientes y menos a la administración de productos.

2. Reducción de Pérdidas y Desperdicio:

- Con una aplicación, se evitarán situaciones como la falta de stock o la sobrecompra de ciertos artículos.
- Las alertas por existencias bajas permitirán reabastecer a tiempo, evitando pérdidas por falta de productos.

3. Mejora en la Toma de Decisiones:

- La aplicación proporcionará datos en tiempo real sobre el inventario. Los gerentes podrán analizar tendencias y tomar decisiones informadas.
- ¿Qué productos se venden más rápido? ¿Cuáles necesitan promoción? La aplicación responderá estas preguntas.

Ventajas de Swift

La elección de Swift como lenguaje de desarrollo refuerza aún más esta justificación:

1. Velocidad y Rendimiento:
 - Swift está diseñado para ser rápido. Las operaciones de búsqueda y actualización de inventario serán ágiles y responsivas.
 - En una tienda con alta rotación de productos, esto marca la diferencia.
2. Seguridad Integrada:
 - Swift previene errores comunes, como desbordamientos de memoria o referencias nulas.
 - En un contexto de datos críticos como el inventario, la seguridad es primordial.

Desarrollo.

Codificación:

```
struct Artículo {  
    var Nombre: String  
    var Cantidad: Int  
}  
  
func RegistroArticulos() -> Artículo{  
    print ("Resgistar Artículo")  
    print ("Ingresa el nombre del articulo")  
    let nombre = readLine()!  
    print ("Ingresa la cantidad de existencia")  
    let cantidad = Int(readLine()!)!  
    return Artículo(Nombre: nombre, Cantidad: cantidad)  
}  
  
func mostrarListado(articulos: [Artículo]){  
    print ("Listado de articulos")  
    for (indice, articulo) in articulos.enumerated(){  
        print ("Artículo \(indice + 1): \(articulo.Nombre)")  
        print ("Cantidad: \(articulo.Cantidad)")  
    }  
}
```

```
}
```

```
}
```

```
func consltadeArticulos(articulos: [Articulo]){  
    print ("Consulta de articulos")  
    print ("Ingresa el nombre del articulo a consultar")  
    let nombre = readLine()!  
    for (indice, articulo) in articulos.enumerated(){  
        if articulo.Nombre == nombre{  
            print ("Articulo \((indice + 1): \((articulo.Nombre)")  
            print ("Cantidad: \((articulo.Cantidad)")  
            break  
        }  
    }  
}
```

```
var Articulos: [Articulo] = []
```

```
var opcion = 0
```

```
repeat{  
    print ("1.- Registro")  
    print ("2.- Listado")  
    print ("3.- Consulta")  
    print ("4.- Salir")  
    print ( "Selecciona una opcion")  
    opcion = (Int(readLine()!))!
```

```
switch opcion{
    case 1:
        print ("Registro")
        let nuevoArticulo = RegistroArticulos()
        Articulos.append(nuevoArticulo)
        print ("****Articulo registrado****")
    case 2:
        print ("Listado")
        mostrarListado(articulos: Articulos)
    case 3:
        print ("Consulta")
        consltadeArticulos(articulos: Articulos)
    case 4:
        print ("Salir")
        default:
            print ("Opcion invalida")
    }
} while opcion != 4
```

Prueba de la Aplicación:

```
1 struct Artículo {
2   var Nombre: String
3   var Cantidad: Int
4 }
5 func RegistroArticulos() -> Artículo{
6   print ("Registrar Artículo")
7   print ("Ingresa el nombre del artículo")
8   let nombre = readLine()!
9   print ("Ingresa la cantidad de existencia")
10  let cantidad = Int(readLine()!)
11  return Artículo(Nombre: nombre, Cantidad: cantidad)
12 }
13 func mostrarListado(articulos: [Artículo]){
14   print ("Listado de artículos")
15   for (indice, articulo) in articulos.enumerated(){
16     print ("Artículo \(indice + 1): \(articulo.Nombre)")
17     print ("Cantidad: \(articulo.Cantidad)")
18   }
19 }
20
21
22
23 func consltadeArticulos(articulos: [Artículo]){
24   print ("Consulta de artículos")
25   print ("Ingresa el nombre del artículo a consultar")
26   let nombre = readLine()!
27   for (indice, articulo) in articulos.enumerated(){
28     if articulo.Nombre == nombre{
29       print ("Artículo \(indice + 1): \(articulo.Nombre)")
30       print ("Cantidad: \(articulo.Cantidad)")
31       break
32     }
33   }
34 }
```

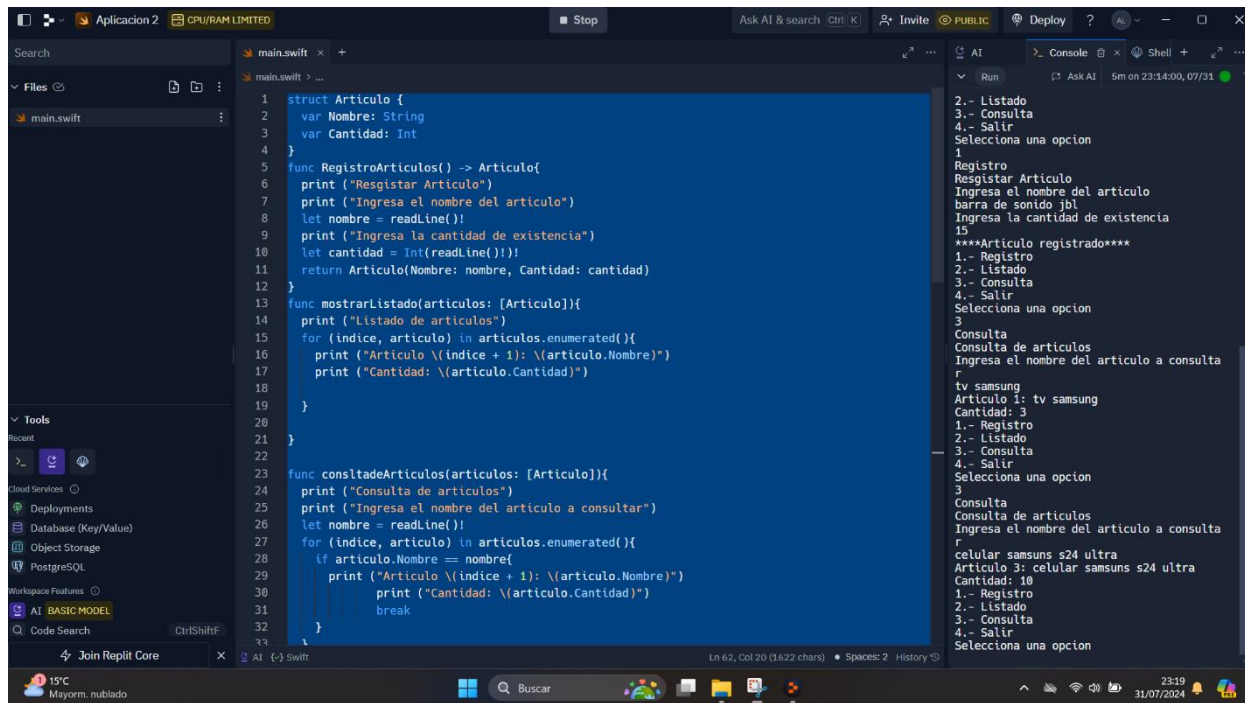
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
1
Registro
Registrar Artículo
Ingresa el nombre del artículo
tv samsung
Ingresa la cantidad de existencia
10
****Artículo registrado****
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
1
Registro
Registrar Artículo
Ingresa el nombre del artículo
notebook asus
Ingresa la cantidad de existencia
5
****Artículo registrado****
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
1
Registro
Registrar Artículo
Ingresa el nombre del artículo
celular samsuns s24 ultra
Ingresa la cantidad de existencia
10
****Artículo registrado****
1.- Registro
2.- Listado
3.- Consulta
4.- Salir

Registro.

```
1 struct Artículo {
2   var Nombre: String
3   var Cantidad: Int
4 }
5 func RegistroArticulos() -> Artículo{
6   print ("Registrar Artículo")
7   print ("Ingresa el nombre del artículo")
8   let nombre = readLine()!
9   print ("Ingresa la cantidad de existencia")
10  let cantidad = Int(readLine()!)
11  return Artículo(Nombre: nombre, Cantidad: cantidad)
12 }
13 func mostrarListado(articulos: [Artículo]){
14   print ("Listado de artículos")
15   for (indice, articulo) in articulos.enumerated(){
16     print ("Artículo \(indice + 1): \(articulo.Nombre)")
17     print ("Cantidad: \(articulo.Cantidad)")
18   }
19 }
20
21
22
23 func consltadeArticulos(articulos: [Artículo]){
24   print ("Consulta de artículos")
25   print ("Ingresa el nombre del artículo a consultar")
26   let nombre = readLine()!
27   for (indice, articulo) in articulos.enumerated(){
28     if articulo.Nombre == nombre{
29       print ("Artículo \(indice + 1): \(articulo.Nombre)")
30       print ("Cantidad: \(articulo.Cantidad)")
31       break
32     }
33   }
34 }
```

Selecciona una opcion
3
Consulta
Consulta de artículos
Ingresa el nombre del artículo a consulta
r
tv samsung
Artículo 1: tv samsung
Cantidad: 3
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
3
Consulta
Consulta de artículos
Ingresa el nombre del artículo a consulta
r
celular samsuns s24 ultra
Artículo 3: celular samsuns s24 ultra
Cantidad: 10
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
2
Listado
Listado de artículos
Artículo 1: tv samsung
Cantidad: 3
Artículo 2: notebook asus
Cantidad: 5
Artículo 3: celular samsuns s24 ultra
Cantidad: 10
Artículo 4: barra de sonido jbl
Cantidad: 15
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion

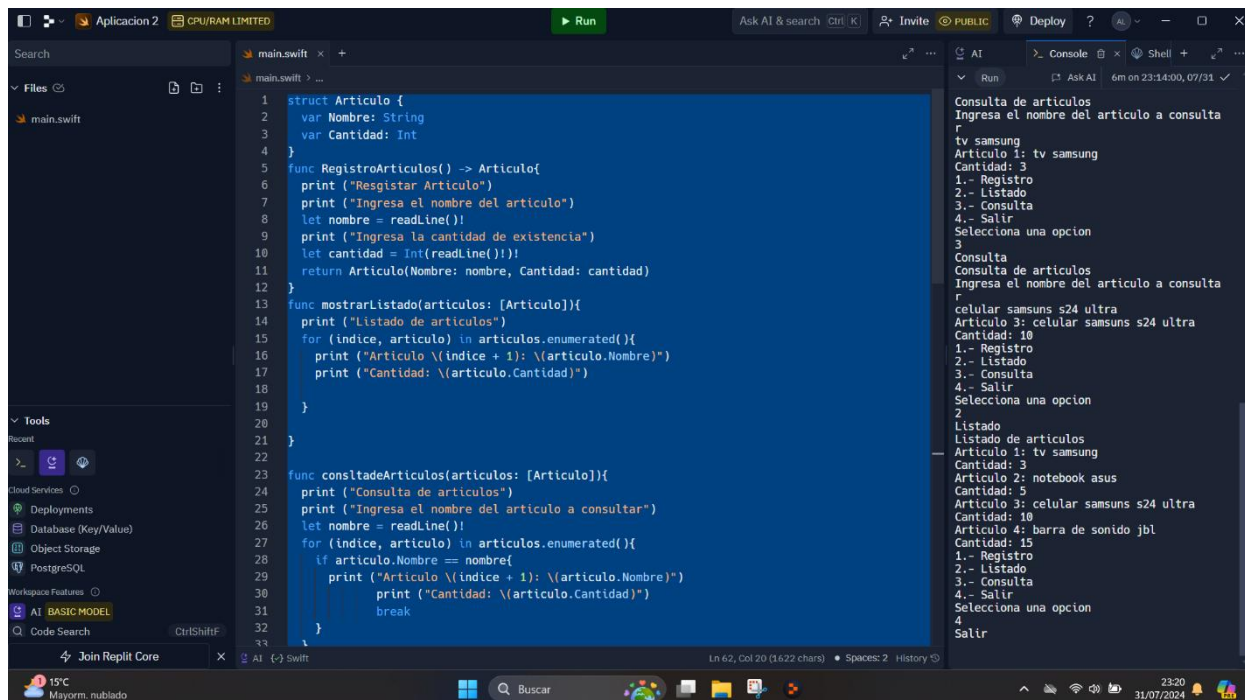
Listado.



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

2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
1
Registro
Registrar Artículo
Ingresa el nombre del articulo
barra de sonido jbl
Ingresa la cantidad de existencia
15
****Articulo registrado****
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
3
Consulta
Consulta de articulos
Ingresa el nombre del articulo a consulta
r
tv samsung
Articulo 1: tv samsung
Cantidad: 3
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
3
Consulta
Consulta de articulos
Ingresa el nombre del articulo a consulta
r
celular samsuns s24 ultra
Articulo 3: celular samsuns s24 ultra
Cantidad: 10
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion

Consulta.



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

Consulta de articulos
Ingresa el nombre del articulo a consulta
r
tv samsung
Articulo 1: tv samsung
Cantidad: 3
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
3
Consulta
Consulta de articulos
Ingresa el nombre del articulo a consulta
r
celular samsuns s24 ultra
Articulo 3: celular samsuns s24 ultra
Cantidad: 10
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
2
Listado
Listado de articulos
Articulo 1: tv samsung
Cantidad: 3
Articulo 2: notebook asus
Cantidad: 5
Articulo 3: celular samsuns s24 ultra
Cantidad: 10
Articulo 4: barra de sonido jbl
Cantidad: 15
1.- Registro
2.- Listado
3.- Consulta
4.- Salir
Selecciona una opcion
4
Salir

Salida.

Conclusión.

En este proyecto, desarrollamos una aplicación de inventario en el lenguaje Swift utilizando la plataforma en línea Replit. Nuestro objetivo era proporcionar una solución sencilla y funcional para gestionar artículos y sus cantidades disponibles.

- **Diseño de la Aplicación:**

Creamos una interfaz de usuario minimalista con un menú que ofrecía las siguientes opciones:

1. Registrar un artículo: Permite al usuario ingresar los detalles de un nuevo artículo.
2. Ver la lista de artículos: Muestra todos los artículos registrados.
3. Consultar existencias: Proporciona la cantidad disponible de un artículo específico.
4. Salir: Finaliza la aplicación.

La simplicidad del diseño facilita la navegación y el uso.

- **Persistencia de Datos:**

1. Replit nos brindó un espacio de almacenamiento persistente. Los datos se guardaban automáticamente entre sesiones.
2. La persistencia aseguró que los registros no se perdieran al cerrar la aplicación.

- **Pruebas y Validación:**

1. Realizamos pruebas exhaustivas para verificar que todas las funciones funcionaran correctamente.
2. Validamos la entrada del usuario para evitar errores y garantizar la integridad de los datos.

En resumen, nuestra aplicación de inventario en Swift demostró cómo combinar conceptos de programación, diseño de interfaces y persistencia de datos para crear una herramienta práctica. Aunque Replit fue nuestro lienzo virtual, las habilidades adquiridas aquí pueden aplicarse en proyectos más amplios y desafiantes.

Referencias.

Video de la Tutoría 2.