

Ingeniería en Desarrollo de Software

Actividad: Número 1.

Nombre de la Actividad: Instalación XCode/Tienda de
ropa.

Nombre del Curso: Desarrollo de aplicaciones móviles IV.

Tutor: Marco Alonso Rodríguez Tapia.

Alumno: Omar Juárez Carmona.

Fecha: 11 - Noviembre – 2023.

INDICE

Contextualización y actividad.....	3
Introducción.....	4
Descripción.....	6
Justificación.....	7
Desarrollo.....	8
- Codificación.....	8
- Prueba del programa.....	13
Conclusión.....	16
Referencias y link.....	17

CONTEXTUALIZACION Y ACTIVIDAD

Contextualización:

La Boutique Norma necesita un programa que funcione como una tienda de ropa en línea. Se requiere que se puedan mostrar productos y el cliente realice su compra en la misma app. Para realizarla, se deberá utilizar el lenguaje de programación Swift.

Actividad:

Realizar la instalación de XCode o utilizar el compilador online. A su vez, crear el programa en lenguaje Swift, y realizar lo siguiente:

- Mostrar 4 artículos a la venta con su nombre, precio y stock.
- Menú de selección para el cliente, donde podrá elegir si quiere comprar un artículo o salir.
- Al realizar la compra, el programa deberá decir qué artículo compró el cliente y la cantidad a pagar por el mismo.

INTRODUCCION

Swift es un lenguaje de programación poderoso e intuitivo creado por Apple para desarrollar aplicaciones de iOS, Mac, Apple TV y Apple Watch. Es fácil de usar y de código abierto, lo que lo hace ideal para que cualquier persona con una idea pueda hacerla realidad. Los desarrolladores pueden programar de una forma más segura y confiable, ahorrar tiempo y enriquecer la experiencia con las aplicaciones. Swift es rápido y eficiente, proporciona información en tiempo real y puede incorporarse fácilmente al código de Objective-C existente. Los desarrolladores están haciendo cosas asombrosas con Swift, como crear aplicaciones para Lyft, Khan Academy, Clear Sky Guide, LinkedIn, Kickstarter, Eventbrite y Hipmunk. Además, las aplicaciones en Swift son pura velocidad. Por ejemplo, un algoritmo común de búsqueda se completa mucho más rápido en Swift que en Objective-C o Python.

2.7. Numerosas universidades e instituciones académicas de todo el mundo están enseñando a programar con Swift y Xcode en el Mac, lo que permite a los estudiantes contar con las mejores herramientas para crear aplicaciones increíbles. ¡Y lo mejor de todo es que Swift es gratis y de código abierto!

Swift es un lenguaje de programación de alto nivel desarrollado por Apple. Es un lenguaje de programación multiparadigma que admite programación orientada a objetos, programación imperativa y programación funcional.

Para programar funciones avanzadas en Swift, primero debes tener una comprensión sólida de las funciones básicas. Las funciones son bloques de código que realizan una tarea específica y se pueden llamar en cualquier lugar de tu programa. En Swift, las funciones son ciudadanos de primera clase, lo que significa que se pueden pasar como argumentos a otras funciones y se pueden devolver como valores.

Una vez que tengas una comprensión sólida de las funciones básicas, puedes comenzar a explorar las funciones avanzadas en Swift. Algunas de las características avanzadas de las funciones en Swift incluyen:

- ✓ Funciones anidadas
- ✓ Funciones genéricas
- ✓ Funciones con parámetros variádicos
- ✓ Funciones con valores predeterminados
- ✓ Funciones con parámetros de entrada y salida

Sin más preámbulo, continuemos con nuestra primera actividad conociendo la programación con funciones avanzadas en el lenguaje de programación Swift.

DESCRIPCION

La Boutique Norma necesita un programa que funcione como una tienda de ropa en línea. Se requiere que se puedan mostrar productos y el cliente realice su compra en la misma app. Para realizarla, se deberá utilizar el lenguaje de programación Swift.

En esta primera actividad de nuestra materia de Desarrollo de aplicaciones móviles IV, vamos a seguir utilizando la herramienta de trabajo donde hemos venido programando hasta el momento en línea, y estoy haciendo mención de Replit.

Utilizaremos esta herramienta de trabajo de Replit en donde programaremos nuestra aplicación en el lenguaje de Swift, que como anteriormente, ya hemos dado una breve reseña de este lenguaje de programación a utilizar.

Crearemos una app con este lenguaje de programación de Swift, en donde el objetivo de dicha app, sea simular una tienda de ropa en línea, en donde cliente pueda decidir con una de las opciones que es lo que desea adquirir de nuestra tienda, llámese Zapatos, Playeras, Pantalones, Sombreros o Calcetines, y al momento de realizar o seleccionar dicha opción, pueda teclear la cantidad que desea adquirir del producto, y ya teniendo la cantidad a comprar, que en automático descuenta las prendas seleccionadas y se las descuenta al inventario que tenemos en tienda, como así mismo, le indique a cliente cuanto ha sido el precio a pagar por haber adquirido dicho artículo en nuestra tienda.

Sin más preámbulo, vamos a comenzar nuestra codificación en nuestra herramienta de trabajo, para así poder visualizar corriendo correctamente nuestra app de nuestra tienda de ropa.

JUSTIFICACION

Swift es un lenguaje de programación de alto nivel desarrollado por Apple. Fue diseñado para ser seguro, rápido y fácil de usar. Swift es un lenguaje de programación moderno que se utiliza para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS. Es un lenguaje de programación de código abierto, lo que significa que cualquier persona puede contribuir a su desarrollo y mejora. Swift es un lenguaje de programación muy popular y se utiliza en muchas aplicaciones populares en la App Store.

La importancia de programar en Swift radica en su facilidad de uso, seguridad y velocidad. Swift es fácil de aprender y usar, lo que lo hace ideal para principiantes y expertos por igual. Además, Swift es un lenguaje de programación seguro que ayuda a prevenir errores comunes en el código. También es muy rápido, lo que significa que las aplicaciones desarrolladas en Swift son muy eficientes.

En resumen, Swift es un lenguaje de programación moderno, seguro y rápido que se utiliza para desarrollar aplicaciones para iOS, macOS, watchOS y tvOS. La facilidad de uso, seguridad y velocidad son algunas de las razones por las que Swift es tan popular entre los desarrolladores.

Ventajas de programar en lenguaje de programación :

- Proceso de desarrollo más rápido.
- Rendimiento y seguridad.
- Gestión de memoria simplificada.
- Soporte multidispositivo.
- Comunidad de código abierto y una alta gama de recursos.

Por estas razones es recomendable el poder escribir programación en el lenguaje de Swift y es por eso que se le recomienda al público programador a utilizar esta herramienta de trabajo para programadores.

DESARROLLO

Codificación

```
struct Artículo {  
    let nombre: String?  
    let precio: Double?  
    var stock: Int?  
}  
  
var articulos = [  
    Artículo(nombre: "Zapatos", precio: 700, stock: 20),  
    Artículo(nombre: "Playeras", precio: 200, stock: 30),  
    Artículo(nombre: "Pantalones", precio: 450, stock: 40),  
    Artículo(nombre: "Sombrero", precio: 200, stock: 10),  
    Artículo(nombre: "Calcetines", precio: 75, stock: 60)  
]  
  
var aux: String = ""  
var opcionIngresada: String = aux  
var cuenta: Double = 0.0  
  
while opcionIngresada != "2" {  
    print("***** Bienvenidos a mi tienda online *****")  
    print("***** ARTICULOS *****")  
    print("***** ----- *****")  
  
    print("_____")  
    print(" Artículo 1 \"(articulos[0].nombre!)")
```



```
print(" Precio \(%s\)" % articulos[0].precio!)
print(" Stock: \(%s\)" % articulos[0].stock!)
print("_____")
```

```
print(" Artículo 2 \(%s\)" % articulos[1].nombre!)
print(" Precio \(%s\)" % articulos[1].precio!)
print(" Stock: \(%s\)" % articulos[1].stock!)
print("_____")
```

```
print(" Artículo 3 \(%s\)" % articulos[2].nombre!)
print(" Precio \(%s\)" % articulos[2].precio!)
print(" Stock: \(%s\)" % articulos[2].stock!)
print("_____")
```

```
print(" Artículo 4 \(%s\)" % articulos[3].nombre!)
print(" Precio \(%s\)" % articulos[3].precio!)
print(" Stock: \(%s\)" % articulos[3].stock!)
print("_____")
```

```
print(" Artículo 5 \(%s\)" % articulos[4].nombre!)
print(" Precio \(%s\)" % articulos[4].precio!)
print(" Stock: \(%s\)" % articulos[4].stock!)
print("_____")
```

```
print("1.- Comprar articulo")
print("2.- Salir")
print("_____")
```

```

aux = readLine()!

opcionIngresada = aux

switch opcionIngresada {
case "1":
    print("_____")
    print("Ingresa el número de articulo a comprar: ")
    aux = readLine()!
    opcionIngresada = aux

    switch opcionIngresada {
    case "1":
        print("Ingresa la cantidad de zapatos que desea comprar: ")
        aux = readLine()!
        opcionIngresada = aux

        let cantidadIngresada = Int(opcionIngresada)!
        if cantidadIngresada <= articulos[o].stock! {
            /// Se puede comprar
            articulos[o].stock! = articulos[o].stock! - cantidadIngresada
            cuenta = Double(cantidadIngresada) * articulos[o].precio!
            print("Total a pagar: \(cuenta)")
            print("***** \nGRACIAS POR SU VALIOSA COMPRA, ESPERO
VUELVA PRONTO A COPPEL *****")
        } else {
            print("No hay suficiente stock, lo sentimos!")
        }
    }
}

```

```
}
```

```
case "2":
```

```
    print("Ingresa la cantidad de playeras que desea comprar: ")
```

```
    aux = readLine()!
```

```
    opcionIngresada = aux
```

```
    let cantidadIngresada = Int(opcionIngresada)!
```

```
    if cantidadIngresada <= articulos[1].stock! {
```

```
        articulos[1].stock! = articulos[1].stock! - cantidadIngresada
```

```
        cuenta = Double(cantidadIngresada) * articulos[1].precio!
```

```
        print("Total a pagar: \"(cuenta)\")
```

```
        print("***** \nGRACIAS POR SU VALIOSA COMPRA, ESPERO  
VUELVA PRONTO A COPPEL *****")
```

```
    } else {
```

```
        print("No hay suficiente stock, lo sentimos!")
```

```
    }
```

```
case "3":
```

```
    print("Ingresa la cantidad de pantalones que desea comprar: ")
```

```
    aux = readLine()!
```

```
    opcionIngresada = aux
```

```
    let cantidadIngresada = Int(opcionIngresada)!
```

```
    if cantidadIngresada <= articulos[2].stock! {
```

```

        articulos[2].stock! = articulos[2].stock! - cantidadIngresada

        cuenta = Double(cantidadIngresada) * articulos[2].precio!

        print("Total a pagar: \"(cuenta)\")

        print("***** \nGRACIAS POR SU VALIOSA COMPRA, ESPERO
VUELVA PRONTO A COPPEL *****")

    } else {

        print("No hay suficiente stock, lo sentimos!")

    }

default:

    print("Opción no válida")

}

case "2":

    //Salir

    print("Hasta pronto!")

    aux = readLine()

default:

    print("Opcion inválida")

    aux = readLine()

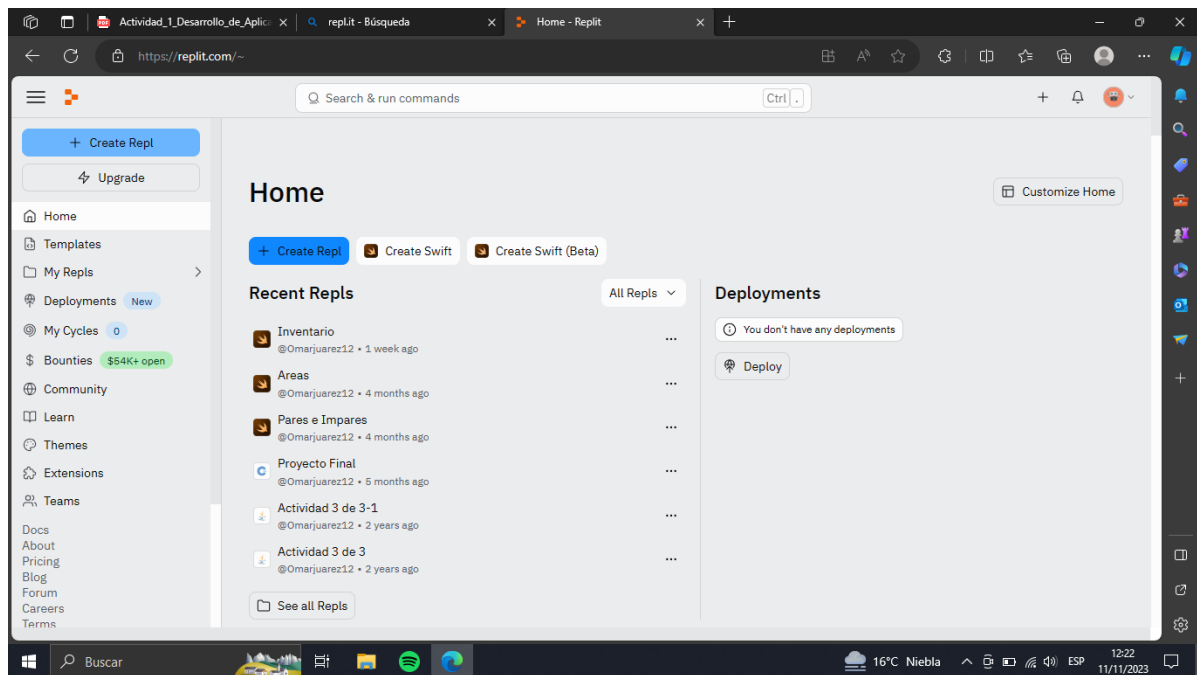
}

}

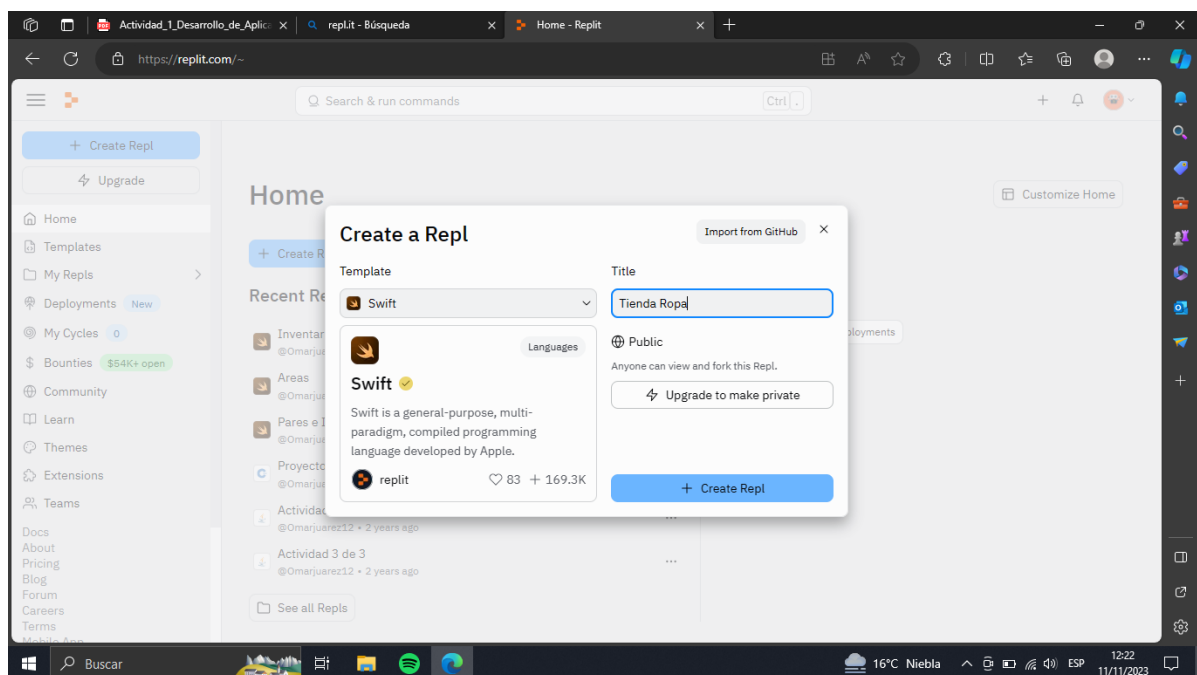
```

Prueba del programa

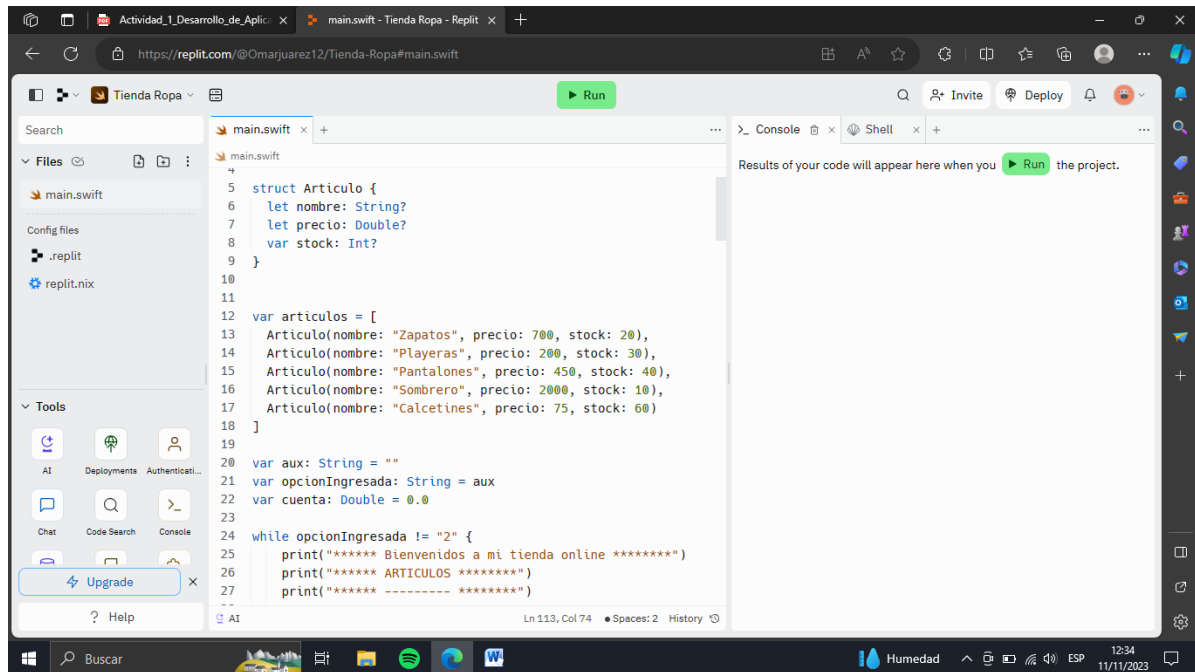
Creando Replit nuevo.



Poniendo nombre a nuevo Replit.



Codificando nuestro proyecto Tienda Ropa.

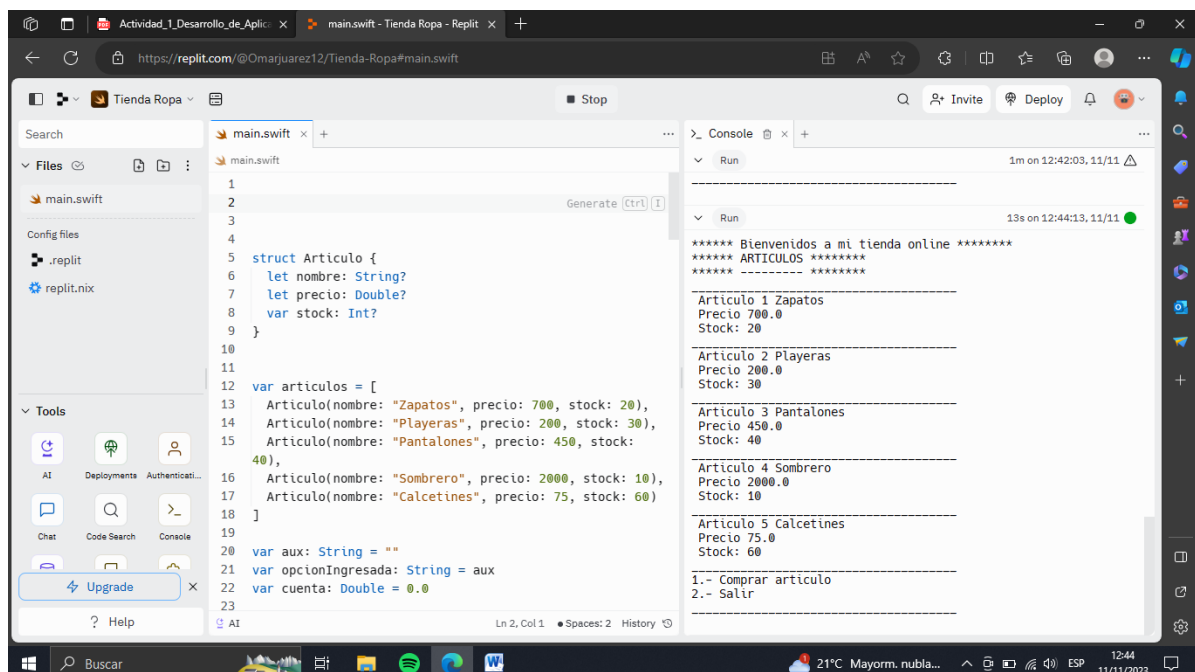


The screenshot shows a Replit IDE window titled 'Tienda Ropa'. The main editor displays a Swift file named 'main.swift' with the following code:

```
1 struct Artículo {
2     let nombre: String?
3     let precio: Double?
4     var stock: Int?
5 }
6
7 var articulos = [
8     Artículo(nombre: "Zapatos", precio: 700, stock: 20),
9     Artículo(nombre: "Playeras", precio: 200, stock: 30),
10    Artículo(nombre: "Pantalones", precio: 450, stock: 40),
11    Artículo(nombre: "Sombrero", precio: 2000, stock: 10),
12    Artículo(nombre: "Calcetines", precio: 75, stock: 60)
13]
14
15 var aux: String = ""
16 var opcionIngresada: String = aux
17 var cuenta: Double = 0.0
18
19 while opcionIngresada != "2" {
20     print("***** Bienvenidos a mi tienda online *****")
21     print("***** ARTICULOS *****")
22     print("***** ----- *****")
23 }
```

The console on the right shows the output of the code, which is currently empty. The status bar at the bottom indicates 'Ln 113, Col 74' and 'Spaces: 2'.

Ejecutando nuestro proyecto Tienda Ropa.



The screenshot shows the same Replit IDE window, but now the code is being executed. The console on the right displays the output of the code, which is a list of items and their prices:

```
***** Bienvenidos a mi tienda online *****
***** ARTICULOS *****
***** ----- *****

Artículo 1 Zapatos
Precio 700.0
Stock: 20

Artículo 2 Playeras
Precio 200.0
Stock: 30

Artículo 3 Pantalones
Precio 450.0
Stock: 40

Artículo 4 Sombrero
Precio 2000.0
Stock: 10

Artículo 5 Calcetines
Precio 75.0
Stock: 60

1.- Comprar artículo
2.- Salir
```

The status bar at the bottom indicates 'Ln 2, Col 1' and 'Spaces: 2'.

Probando nuestro proyecto Tienda Ropa.

The screenshot shows a Replit environment with a Swift file named `main.swift` and a console window. The code defines a `struct Articulo` with properties `nombre`, `precio`, and `stock`. It then creates an array of `Articulo` objects representing different clothing items. The console output shows the results of running the code, including a total payment of 600.0, a thank you message, and a list of items with their prices and stock levels.

```
1
2
3
4
5 struct Articulo {
6     let nombre: String?
7     let precio: Double?
8     var stock: Int?
9 }
10
11
12 var articulos = [
13     Articulo(nombre: "Zapatos", precio: 700, stock: 20),
14     Articulo(nombre: "Playeras", precio: 200, stock: 30),
15     Articulo(nombre: "Pantalones", precio: 450, stock:
16 40),
17     Articulo(nombre: "Sombrero", precio: 200, stock: 10),
18     Articulo(nombre: "Calcetines", precio: 75, stock: 60)
19 ]
20
21 var aux: String = ""
22 var opcionIngresada: String = aux
23 var cuenta: Double = 0.0
24
```

Console Output:

```
Run
Total a pagar: 600.0
*****
GRACIAS POR SU VALIOSA COMPRA, ESPERO VUELVA PRONTO A COPPEL *
*****
***** Bienvenidos a mi tienda online *****
***** ARTICULOS *****
***** ----- *****

Articulo 1 Zapatos
Precio 700.0
Stock: 20

Articulo 2 Playeras
Precio 200.0
Stock: 27

Articulo 3 Pantalones
Precio 450.0
Stock: 40

Articulo 4 Sombrero
Precio 200.0
Stock: 10

Articulo 5 Calcetines
Precio 75.0
Stock: 60

1.- Comprar articulo
2.- Salir
```

Ejecución correcta de nuestro proyecto Tienda Ropa.

The screenshot shows a Replit environment with a Swift file named `main.swift` and a console window. The code defines a `struct Articulo` with properties `nombre`, `precio`, and `stock`. It then creates an array of `Articulo` objects representing different clothing items. The console output shows the results of running the code, including a total payment of 600.0, a thank you message, and a list of items with their prices and stock levels.

```
1
2
3
4
5 struct Articulo {
6     let nombre: String?
7     let precio: Double?
8     var stock: Int?
9 }
10
11
12 var articulos = [
13     Articulo(nombre: "Zapatos", precio: 700, stock: 20),
14     Articulo(nombre: "Playeras", precio: 200, stock: 30),
15     Articulo(nombre: "Pantalones", precio: 450, stock:
16 40),
17     Articulo(nombre: "Sombrero", precio: 200, stock: 10),
18     Articulo(nombre: "Calcetines", precio: 75, stock: 60)
19 ]
20
21 var aux: String = ""
22 var opcionIngresada: String = aux
23 var cuenta: Double = 0.0
24
```

Console Output:

```
Run
GRACIAS POR SU VALIOSA COMPRA, ESPERO VUELVA PRONTO A COPPEL *
*****
***** Bienvenidos a mi tienda online *****
***** ARTICULOS *****
***** ----- *****

Articulo 1 Zapatos
Precio 700.0
Stock: 20

Articulo 2 Playeras
Precio 200.0
Stock: 27

Articulo 3 Pantalones
Precio 450.0
Stock: 40

Articulo 4 Sombrero
Precio 200.0
Stock: 10

Articulo 5 Calcetines
Precio 75.0
Stock: 60

1.- Comprar articulo
2.- Salir

2
Hasta pronto!
```

CONCLUSION

En esta actividad número uno de nuestra materia de desarrollo de aplicaciones móviles IV, hemos aprendido a programar con funciones avanzadas en Swift.

Debemos de saber que las funciones en Swift son muy potentes y permiten manipular la lógica de tu aplicación. Cuanto más pequeñas sean las funciones, más fácil serán de reutilizar por toda tu aplicación. Las funciones en Swift agrupan código relacionado para realizar tareas.

Las funciones pueden retornar o no valores. En Swift, las funciones tienen un tipo. El tipo de una función consiste en los tipos de los parámetros de entrada y el tipo de retorno.

Las funciones también pueden devolver tuplas. No las suelo utilizar mucho, pero en ocasiones son muy útiles.

Al usar una función, podemos crear dos nombres para el parámetro. El primero es el nombre del parámetro que se usa desde fuera de la función, y el segundo es para usarlo desde dentro del scope de la función.

La sintaxis de las funciones en Swift se considera bastante flexible; es así como, en la guía de Apple sobre este lenguaje de programación, se menciona que con las funciones en Swift puedes expresar muchas cosas, desde una función simple de estilo C que no tenga nombres de parámetros, hasta un método complejo de estilo Objective-C que tenga nombres y etiquetas de argumentos para cada uno de los parámetros.

REFERENCIAS

Sign up. (n.d.). Replit. Retrieved October 30, 2023, from <https://replit.com/~>

SwiftBeta. (2020, August 25). *FUNCIONES en Swift en Español - Curso Swift*.

SwiftBeta. <https://www.swiftbeta.com/funciones-en-swift/>

Swift lenguaje de programación. (n.d.). Bing. Retrieved October 30, 2023, from

<https://www.bing.com/search?q=swift+lenguaje+de+programaci%C3%B3n&qsr=RQ&pq=lenguaje+de+swift&sc=10-17&cvid=FCE2C003BB394F77BA724AA4F440DAFA&FORM=QBRE&sp=1&lq=0&showconv=1>

LINKS

[main.swift - Tienda Ropa - Replit](#)

[Omarsitho1988 \(github.com\)](#)

https://drive.google.com/drive/folders/1zuQGG7Yr_K4HYqVS2uj569fGpslifYol?usp=sharing