UNIVERSIDAD DON BOSCO
Ingeniería en Ciencias de la Computación

Desarrollo de Software para Móviles
Ing. Alexander Sigüenza
Grupo Teórico I

Proyecto Kotlin 1

Nombre:
Oscar Alexander Guevara Rodríguez

Carnet:
GR222756

El presente código busca ser una experiencia de tienda para un usuario común que hace sus compras día a día.

La idea principal del código se basa en una tienda la cual posee un inventario y un carrito de compras para el usuario.

```
|------------***Bienvenido a Tienda Naranja***-------------|
| 1. Ver productos disponibles                              |
| 2. Agregar producto al carrito                            |
| 3. Ver carrito                                            |
| 4. Eliminar producto del carrito                          |
| 5. Confirmar compra                                       |
| 6. Salir                                                  |
|___________________________________________________________|
Digite una opción (1-6):
```

El usuario es recibido por un menú con distintas opciones, las cuales le ayudarán a tener una experiencia completa dentro de la tienda. A continuación, se evaluará cada una de las opciones y su función.

Opción 1: Ver productos disponibles. Esta opción está destinada a que el usuario vea los productos disponibles en la tienda, sus precios, la cantidad en existencia y un identificador (ID) que le permitirá interactuar con estos productos en las demás opciones. Una vez mostrado el usuario regresará nuevamente al menú principal de forma automática.

```
---------*Productos Disponibles*---------
id   | Producto         | Precio ($)| Existencia
1    | Tomahawk         | 14.00     |7
2    | Arrachera        | 14.50     |25
3    | Ribeye           | 18.00     |15
4    | Huevos           | 6.00      |21
5    | Grasa animal     | 11.50     |13
6    | Costilla         | 9.50      |20
7    | Lomo             | 7.00      |23
8    | Wagyu Beef       | 75.00     |10
9    | Queso Fresco     | 5.00      |20
10   | Miel natural     | 10.00     |12
11   | Leche cruda      | 8.00      |15
12   | Crema            | 5.00      |18
```

Opción 2. Agregar producto al carrito. Una vez visto los productos disponibles el usuario puede añadir un producto al carrito por medio del ID. Se le preguntará el ID del producto que desee agregar y la cantidad que llevará de dicho producto. Una vez hecho el producto se agregará al carrito y el usuario volverá al menú principal. Dentro del código el producto es restado del inventario de manera temporal, mientras el usuario lo tenga en su carrito.

```
Digite una opción (1-6): 2
Ingrese el ID del producto que desea agregar: 6
Ingrese la cantidad: 2
Producto agregado al carrito exitosamente!
Regresando al menú principal...
```

Opción 3. Ver carrito. Cuando el usuario agregue sus productos, tendrá la posibilidad de ver que productos hay en su carrito, cantidad y precio y un total momentáneo que no tiene el IVA, ya que esté se agrega en la facturación.

```
Digite una opción (1-6): 3
Productos en carrito:
 Costilla            | Cantidad: 2 | Precio:  9.50      $ | Total: 19.0$
Total sin IVA: 19.0$
```

Opción 4. Eliminar producto del carrito. Al momento de comprar el usuario puede arrepentirse de haber agarrado un producto en su carrito, es por ello por lo que tiene la opción de eliminar dicho producto de su carrito. Cuando esto sucede el producto será eliminado del carrito y será devuelto al inventario.

```
Digite una opción (1-6): 4
Ingrese el ID del producto a eliminar: 6
Producto eliminado del carrito.
```

```
Digite una opción (1-6): 3
Productos en carrito:
Carrito vacío! Agregue un producto
```

Opción 5. Cuando el usuario ya tiene los productos que llevará y no seguirá interactuando más, puede confirmar su compra, es acá cuando se le generará una factura y se incluirá el IVA como impuesto, dando un resumen de su compra y un total final.

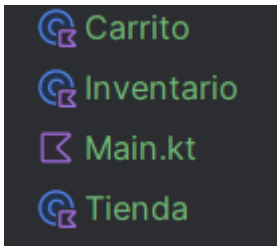
```
Digite una opción (1-6): 5
-------------------------------- Factura ----------------------------------
 Grasa animal        | Cantidad: 2 | Precio:  11.50     $ | Subtotal: 23.0$
| Total sin IVA: 23.0$                                                    |
| IVA (13%): 0.0$                                                         |
---------------------------------------------------------------------------
| Total Final: 23.0$                                                      |
-------------------------- GRACIAS POR SU COMPRA --------------------------
```

Opción 6. Salir. Luego de generar la factura el usuario será devuelto al menú principal si él quisiera seguir comprando, no obstante, si ya terminó de comprar o simplemente no hubo ningún producto que le llamara la atención o simplemente quería ver los productos, puede salir del sistema en todo momento con la opción 6, dando así por terminado el programa.

```
Digite una opción (1-6): 6
Gracias por su visita, usted está saliendo...

Process finished with exit code 0
```

Código y estructura.
El código está estructurado en un archivo principal llamado main.kt, 3 archivos clases, llamados Carrito.kt, Inventario.kt y Tienda.kt y un archivo inventario.txt. Las clases utilizadas son Carrito, Inventario y Tienda.



Main.kt. En este archivo se ubica el menú y la creación de una instancia de tienda, con el cual el usuario interactúa en todo momento, desde este archivo se invocan los métodos de la clase tienda, dependiendo de la opción del menú que el usuario elija. En palabras simples gestiona el menú de la tienda, permitiendo al usuario ver productos, agregar y eliminar artículos del carrito, confirmar la compra o salir del programa

```kotlin
when (opcion) {
    1 -> tienda.mostrarProductos()
    2 -> tienda.agregarAlCarrito()
    3 -> tienda.mostrarCarrito()
    4 -> tienda.eliminarDelCarrito()
    5 -> tienda.confirmarCompra()
    6 -> {
        println("Gracias por su visita, usted está saliendo...")
        tienda.restaurarInventario()
    }
    else -> {
        println("Opción inválida, asegúrese de ingresar una de las opciones (1-
        println("Regresando al menu principal...")
        Thread.sleep( millis: 3000)  // espera de 3 segundos antes de mostrar menú
    }
```

Tienda.kt. En este archivo se ubica la clase Tienda, la cual se encarga de la gestión del inventario y el carrito de compras, ofreciendo métodos para ver productos, agregar y eliminar productos.

La clase posee dos atributos privados, los cuales son inventario (objeto de la clase Inventario) que gestiona los productos y carrito (objeto de la clase carrito) que maneja los productos que el usuario seleccione.

```kotlin
private val inventario = Inventario() //nue
private val carrito = Carrito(inventario)//
```

Luego la clase tienda tiene varios métodos los cuales funcionan como un puente de conexión para llamar a las funciones de Inventario o Carrito.

```
fun mostrarProductos() { //mostrar productos en existencia  new *
    inventario.mostrarProductos() // llamada al metodo de la clase i
}

fun agregarAlCarrito() { //agregar productos al carrito  new *
    carrito.agregarProducto() // llamada al metodo de la clase carri
}

fun mostrarCarrito() { // mostrar productos dentro del carrito  new *
    carrito.mostrarCarrito()
}

fun eliminarDelCarrito() { //eliminar productos del carrito  new *
    carrito.eliminarProducto()
}

fun confirmarCompra() { //confirmar compra y generar factura  new *
    carrito.confirmarCompra()
}

fun restaurarInventario() { //devolver productos si el usuario sale
    carrito.restaurarInventario()
}
```

Carrito.kt: En este archivo se ubica la clase Carrito, la cual se encarga de gestionar la compra de los productos. Permite agregar, eliminar, confirmar compras.

```
class Carrito(private val inventario: Inventario) {  new *
    private val carrito = mutableListOf<String>()
```

La clase Carrito recibe un objeto de la clase Inventario como parámetro, es utilizado para el manejo del inventario cuando el usuario interactúe con los productos. Luego se declara una lista mutable donde se añadirán los productos agregados al carrito.

Está conformada por las funciones:

agregarProducto(): Permite al usuario agregar un producto al carrito.
mostrarCarrito(): Ver productos del carrito
eliminarProducto(): Eliminar un producto del carrito y devolverlo al inventario

restaurarInventario(): Devolver los productos del carrito del usuario al inventario si este sale sin confirmar una compra.
confirmarCompra(): Generar la factura, totales, inclusión de impuestos y breve resumen de los productos.

Inventario.kt. Se encarga principalmente de la gestión de productos en el archivo de texto donde se encuentra el inventario (inventario.txt).

```
private val archivo = File( pathname: "C:\\Users\\oguev\\IdeaProjects\\dsm-carrito\\inventario.txt")
val productos = mutableListOf<String>() //creacion de lista mutable con los productos

init {  new *
    if (archivo.exists()) { //verifica si existe
        BufferedReader(FileReader(archivo)).use { reader ->
            reader.forEachLine { productos.add(it) } //lee cada linea y lo agrega a la lista
        }
    }
}
```

El archivo del inventario es cargado durante la ejecución del programa y se crea una lista mutable con todos los productos del archivo inventario.txt

Entre sus funciones se encuentran:

mostrarProductos(): Utilizada para la impresión en consola de los productos obtenidos del archivo inventario.txt

actualizarInventario(): Recibe el id del producto como parámetro y la cantidad actualizada, esta función es invocada cuando un usuario elige un determinado producto. El inventario se actualizará de manera temporal mientras el usuario tenga el producto en su carrito.

obtenerCantidad(): Esta función obtiene la cantidad de productos que el usuario había tomado previamente en su carrito. En caso de que el usuario no haya completado su compra, devuelve la cantidad en stock de un producto específico.

guardarArchivo(): Guardar el archivo.txt con los cambios realizados durante la ejecución del programa.

Inventario.txt: acá se encuentra la lista de todos los productos, sus ID, sus nombres, sus precios y cantidades disponibles. Este archivo es modificado a lo largo de la ejecución del programa