



Trabajo de investigación

Carrito de Compras en Kotlin con Facturación, en modo consola

- Desarrollo de Software para Móviles DSM941 G01T -

Docente:

Ing. Alexander Alberto Siguenza Campos

Integrantes:

- William Alberto García Gómez - GG212522
- Christian Yahir López Hernández - LH212531
- Yahir Stewart Sibrian Arriola - SA212551
- Alberto Joseph Mendoza Moreno - MM200462
- Juan Jose Lue Valdez -LV231966

Fecha:

23 de marzo de 2025

Índice

Carrito en Kotlin.....	3
Producto.kt.....	3
Inventario.kt.....	3
CarritoDeCompras.kt.....	3
CarritoDeCompras.kt.....	4
Main.kt.....	5
GitHub.....	6
¿Cómo ejecutar el proyecto?.....	6
Descarga manual.....	6
Clonar el proyecto.....	7

Carrito en Kotlin

Producto.kt

Clase Producto, que representa los productos disponibles en el inventario.

Atributos:

- id: Identificador único del producto.
- nombre: Nombre del producto.
- precio: Precio unitario del producto.
- cantidadDisponible: Cantidad de unidades disponibles

```
data class Producto(
    val id: Int,
    val nombre: String,
    val precio: Double,
    var cantidadDisponible: Int
)
```

Esta clase usa un Data Class para facilitar la manipulación y comparación de objetos

Inventario.kt

Función obtenerInventario(), la cual devuelve una lista de productos predefinidos. Cada producto está representado por una instancia de la clase Producto. El inventario se carga en memoria al inicio del programa, permitiendo su consulta y modificación.

```
fun obtenerInventario(): MutableList<Producto> {
    return mutableListOf(
        Producto(1, "producto 1", 20.0, 10),
        Producto(2, "producto 2", 10.0, 5),
        Producto(3, "producto 3", 40.0, 30),
        Producto(4, "producto 4", 50.0, 5),
        Producto(5, "producto 5", 10.0, 30),
        Producto(6, "producto 6", 20.0, 10),
        Producto(7, "producto 7", 60.0, 20),
        Producto(8, "producto 8", 30.0, 8),
        Producto(9, "producto 9", 25.0, 9)
    )
}
```

CarritoDeCompras.kt

Clase CarritoDeCompra donde se maneja la selección y eliminación de productos. Utiliza una estructura de datos

```
mutableMapOf<Int, Pair<Producto, Int>>()
```

- La clave es el ID del producto.
- El valor es un par (Producto, cantidad seleccionada).

Proporciona mensajes de error en caso de cantidades inválidas o productos inexistentes.

Métodos principales:

– **agregarProducto()**: Permite añadir productos al carrito validando su disponibilidad.

```
fun agregarProducto(producto: Producto, cantidad: Int) {
```

– **eliminarProducto()**: Permite eliminar productos, devolviendo la cantidad al inventario si es necesario.

```
fun eliminarProducto(productId: Int, cantidad: Int)
```

– **mostrarCarrito()**: Muestra el contenido del carrito con los productos, cantidades y precios.

```
fun mostrarCarrito()
```

– **calcularTotal()**: Calcula el precio total de todos los productos en el carrito.

```
fun calcularTotal(): Double
```

CarritoDeCompras.kt

Genera y muestra un resumen de la compra.

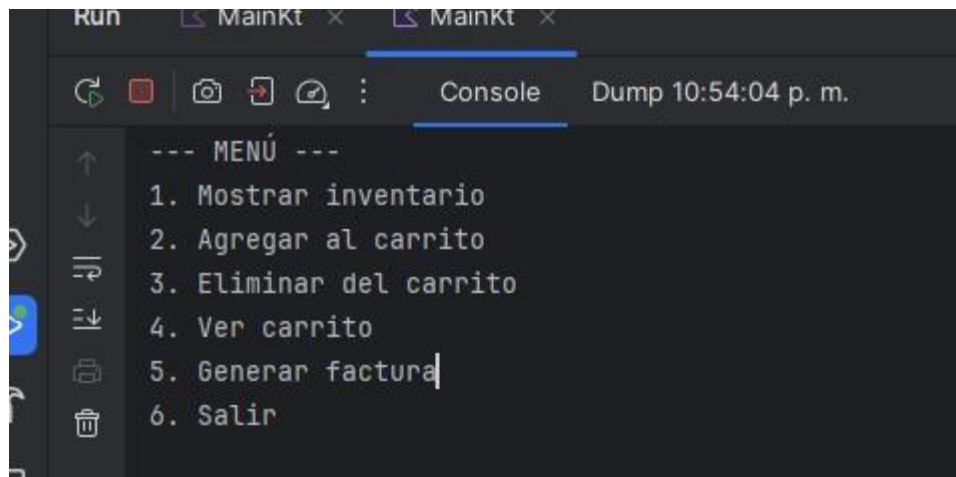
– **generarFactura()**: Muestra los productos en el carrito junto con sus cantidades, precios y el total a pagar.

```
class Factura(private val carrito: CarritoDeCompras) {  @ ChrisHz
    fun generarFactura() {  @ ChrisHz
        println("\n----- FACTURA -----")
        carrito.mostrarCarrito()
        val total = carrito.calcularTotal()
        println("\nTOTAL A PAGAR: $$total")
    }
}
```

Main.kt

Inicia y gestiona la interacción con el usuario a través de un menú en consola.

1. Mostrar inventario: muestra los productos disponibles en el sistema.
2. Agregar productos al carrito: permite seleccionar un producto por su ID y añadirlo al carrito.
3. Eliminar productos del carrito: remueve un producto parcial o totalmente del carrito
4. Ver carrito: muestra los productos seleccionados, sus precios y el total actual de la compra.
5. Generar factura: imprime en pantalla el resumen de la compra y el total a pagar.
6. Salir: finaliza la aplicación.

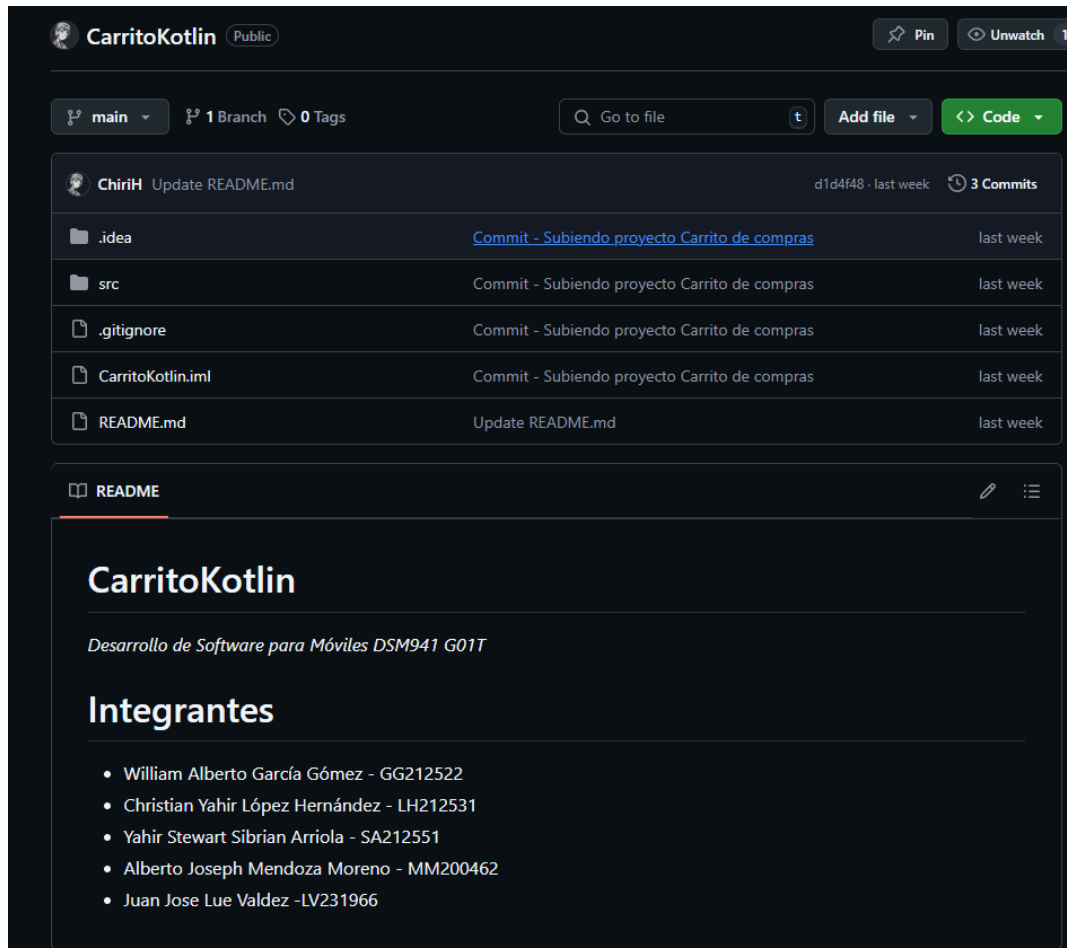


The screenshot shows an IDE's console window with a dark theme. At the top, there are tabs for 'Run' and 'MainKt'. Below the tabs, there's a toolbar with icons for running, stopping, and debugging. The console output displays a menu titled '--- MENÚ ---' followed by six numbered options: '1. Mostrar inventario', '2. Agregar al carrito', '3. Eliminar del carrito', '4. Ver carrito', '5. Generar factura', and '6. Salir'. The cursor is positioned at the end of the fifth option.

```
--- MENÚ ---  
1. Mostrar inventario  
2. Agregar al carrito  
3. Eliminar del carrito  
4. Ver carrito  
5. Generar factura  
6. Salir
```

GitHub

Repositorio: <https://github.com/ChiriH/CarritoKotlin>

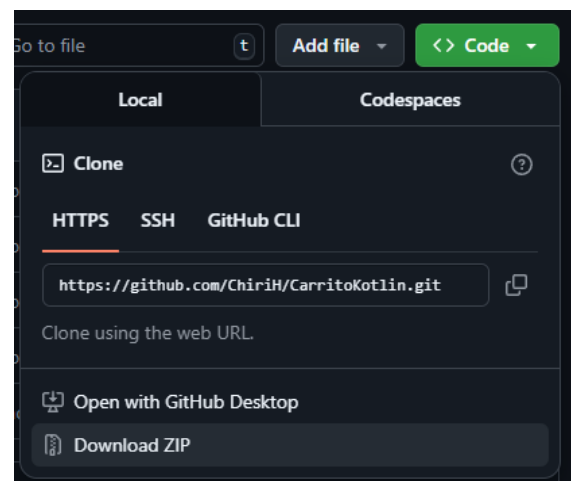


¿Cómo ejecutar el proyecto?

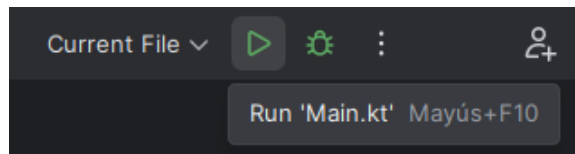
Se puede acceder al repositorio y descargar de manera manual o clonar el proyecto.

Descarga manual

1. Acceder al repositorio en GitHub y descargar el código como **archivo ZIP**.
2. Extraer el archivo ZIP en una carpeta local.



3. Abrir **IntelliJ IDEA** y seleccionar la opción **Abrir Proyecto**.
4. Ejecutar el archivo Main.kt



Clonar el proyecto

Clonar el repositorio directamente con el siguiente comando en la terminal: `git clone https://github.com/ChiriH/CarritoKotlin.git`

Luego:

1. Abrir IntelliJ IDEA.
2. Seleccionar Abrir Proyecto y elegir la carpeta donde se clonó el repositorio.
3. Ejecutar Main.kt para iniciar el programa.