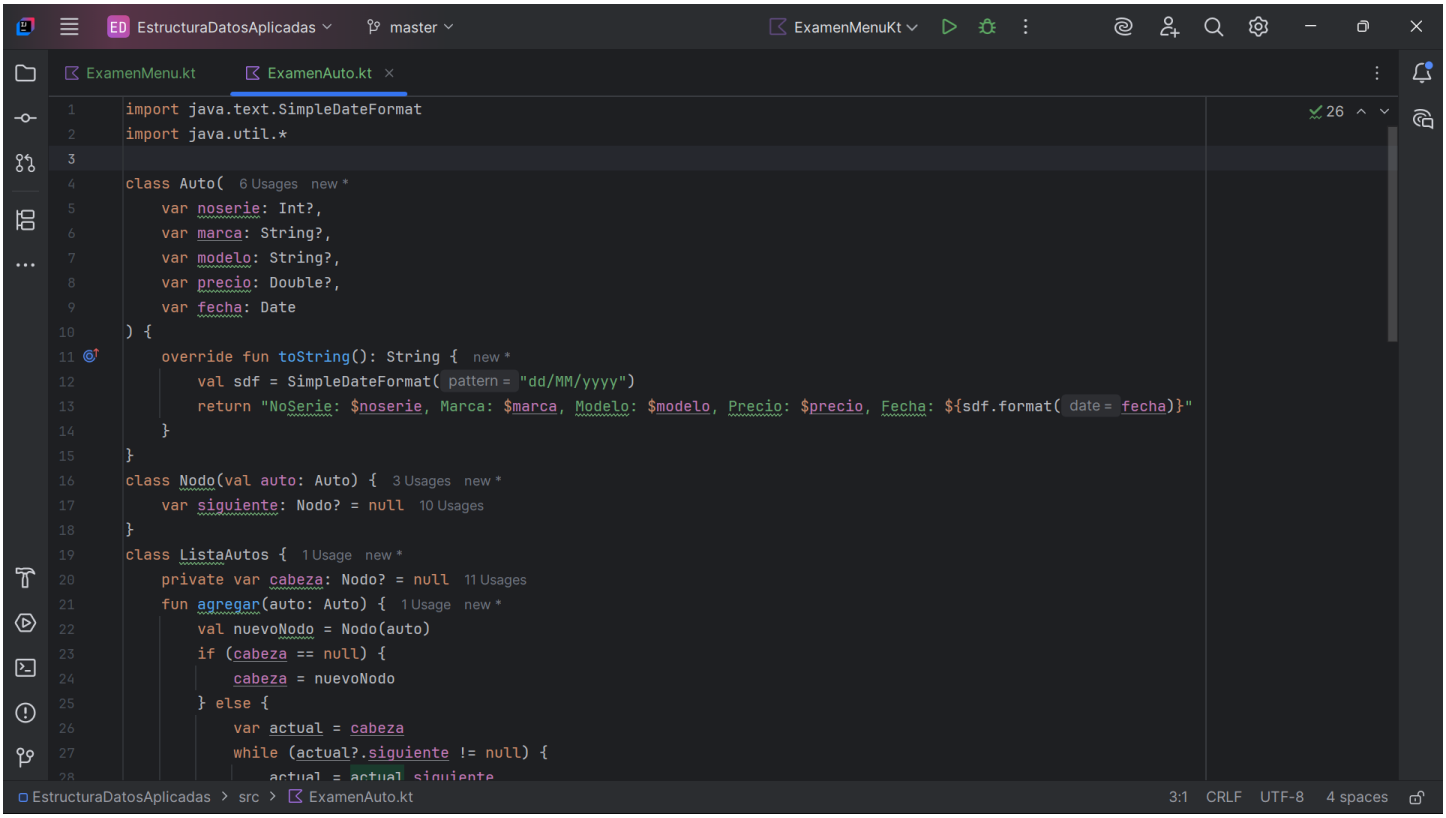
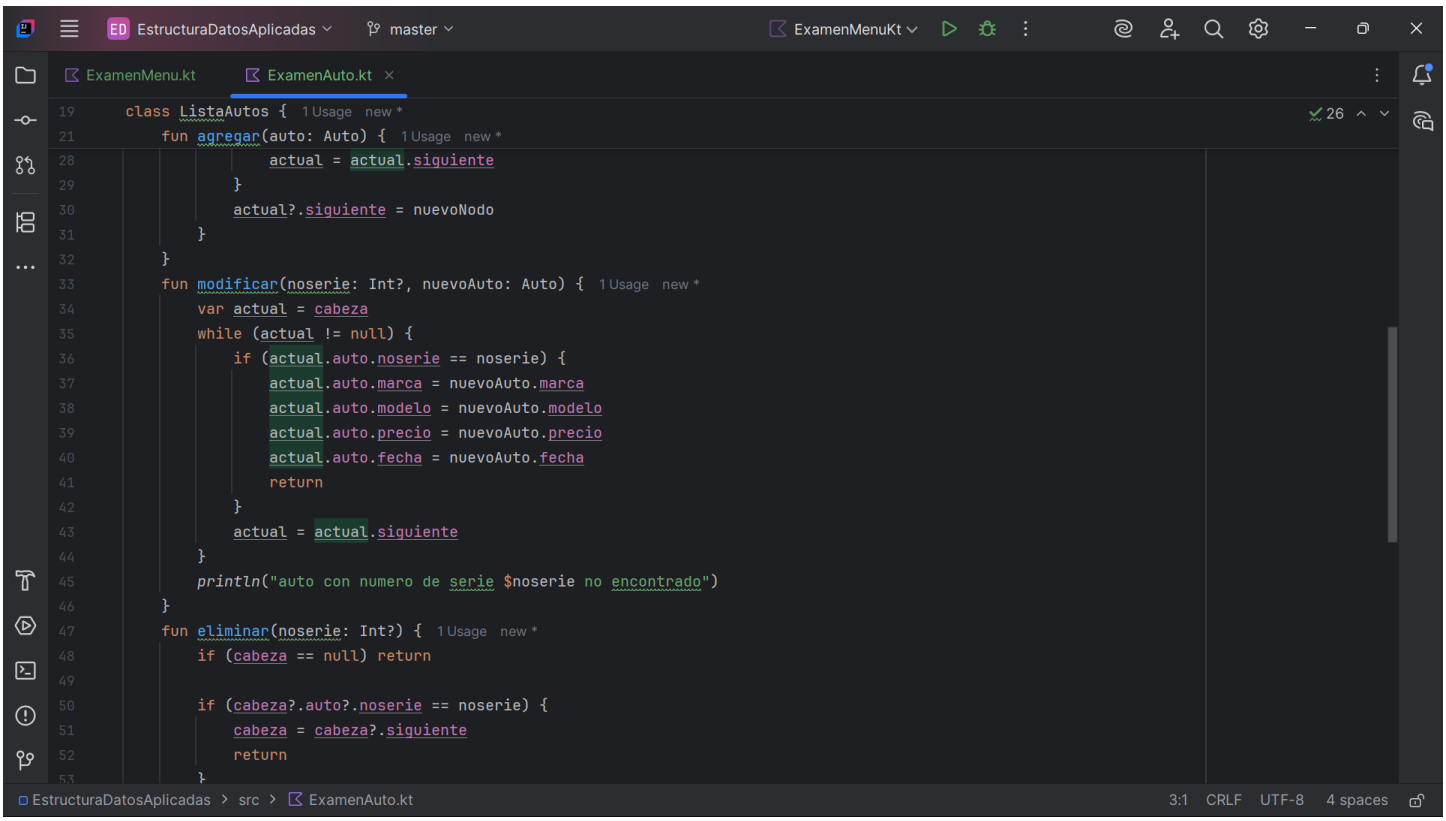
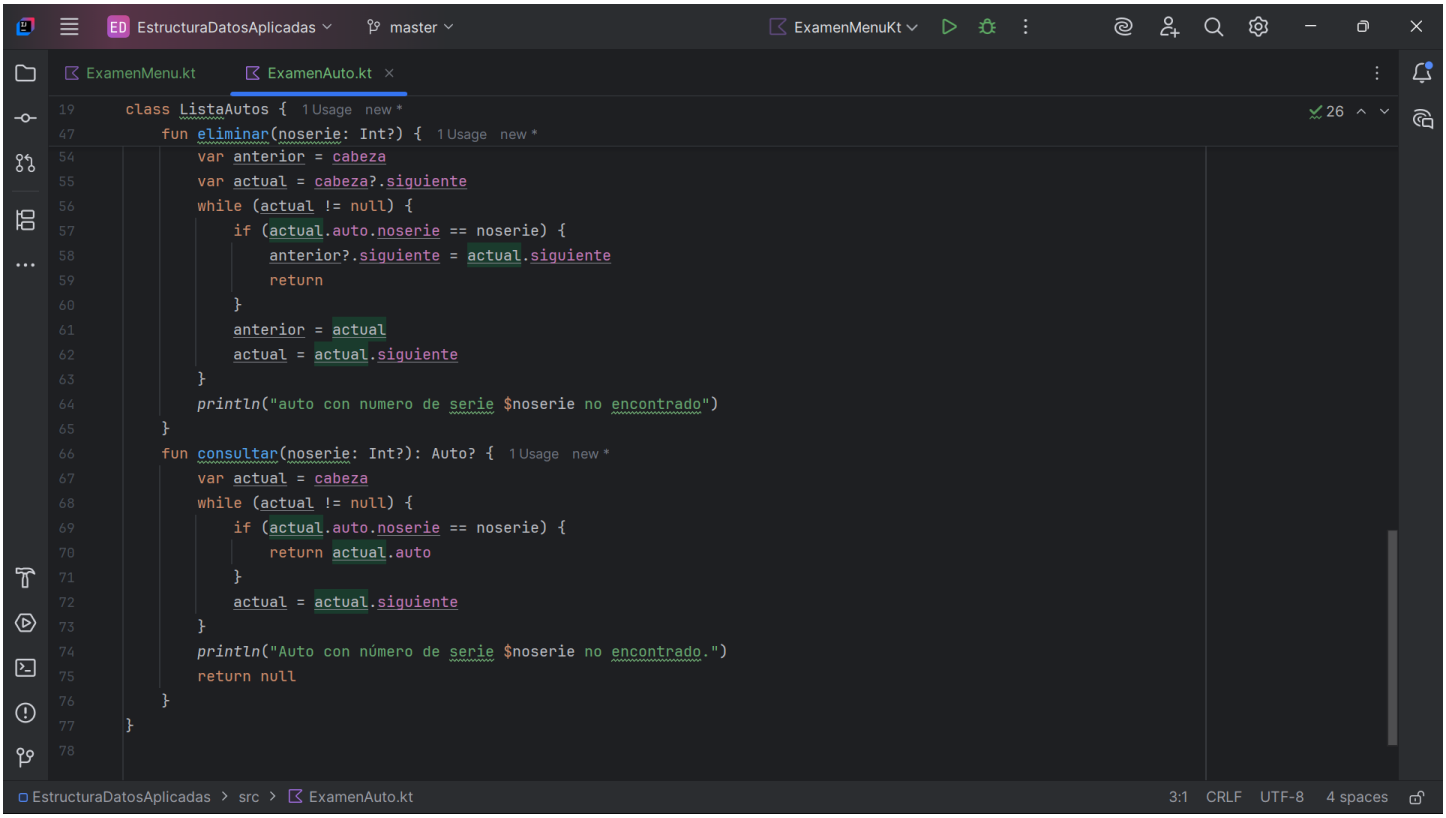
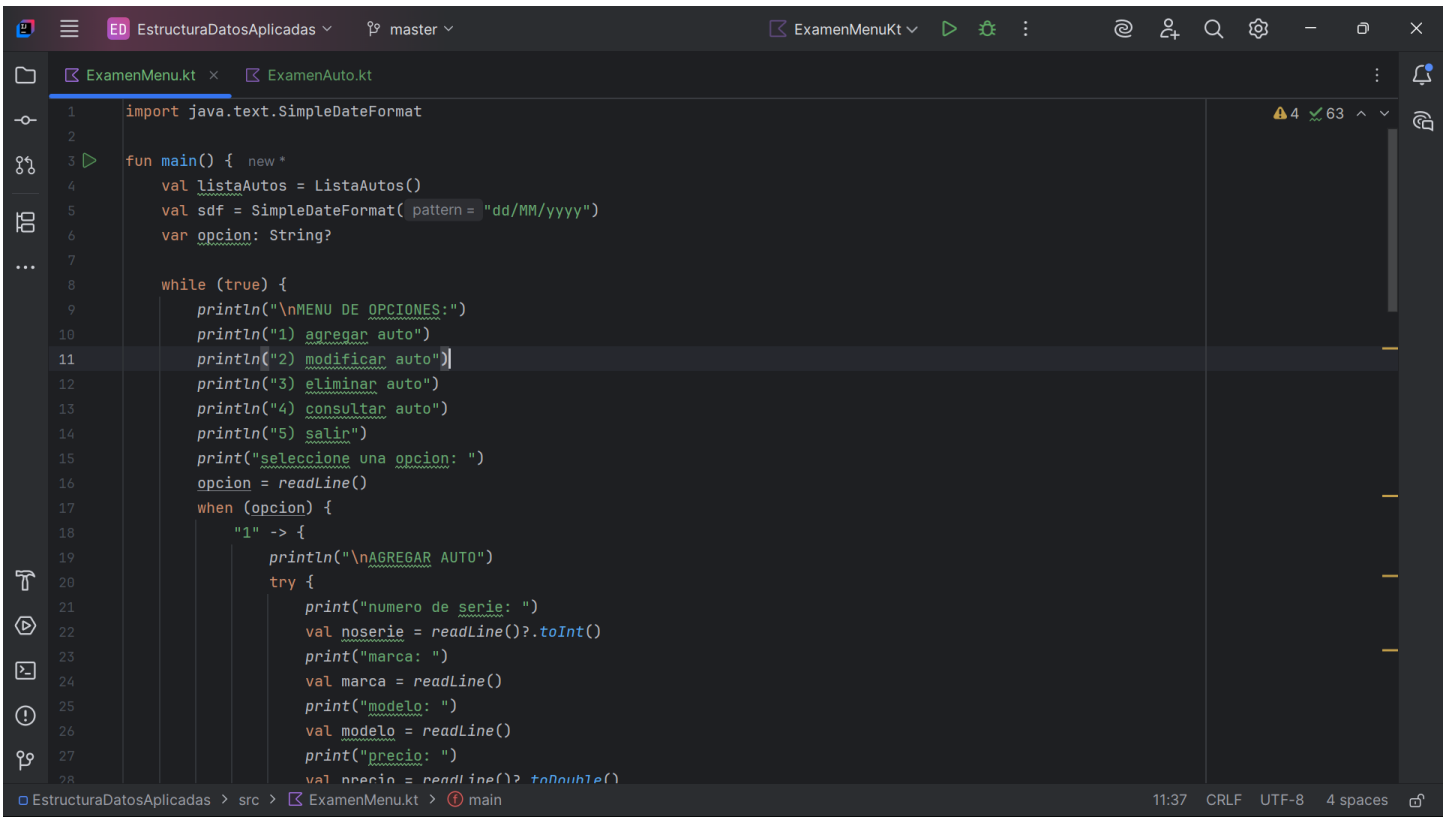
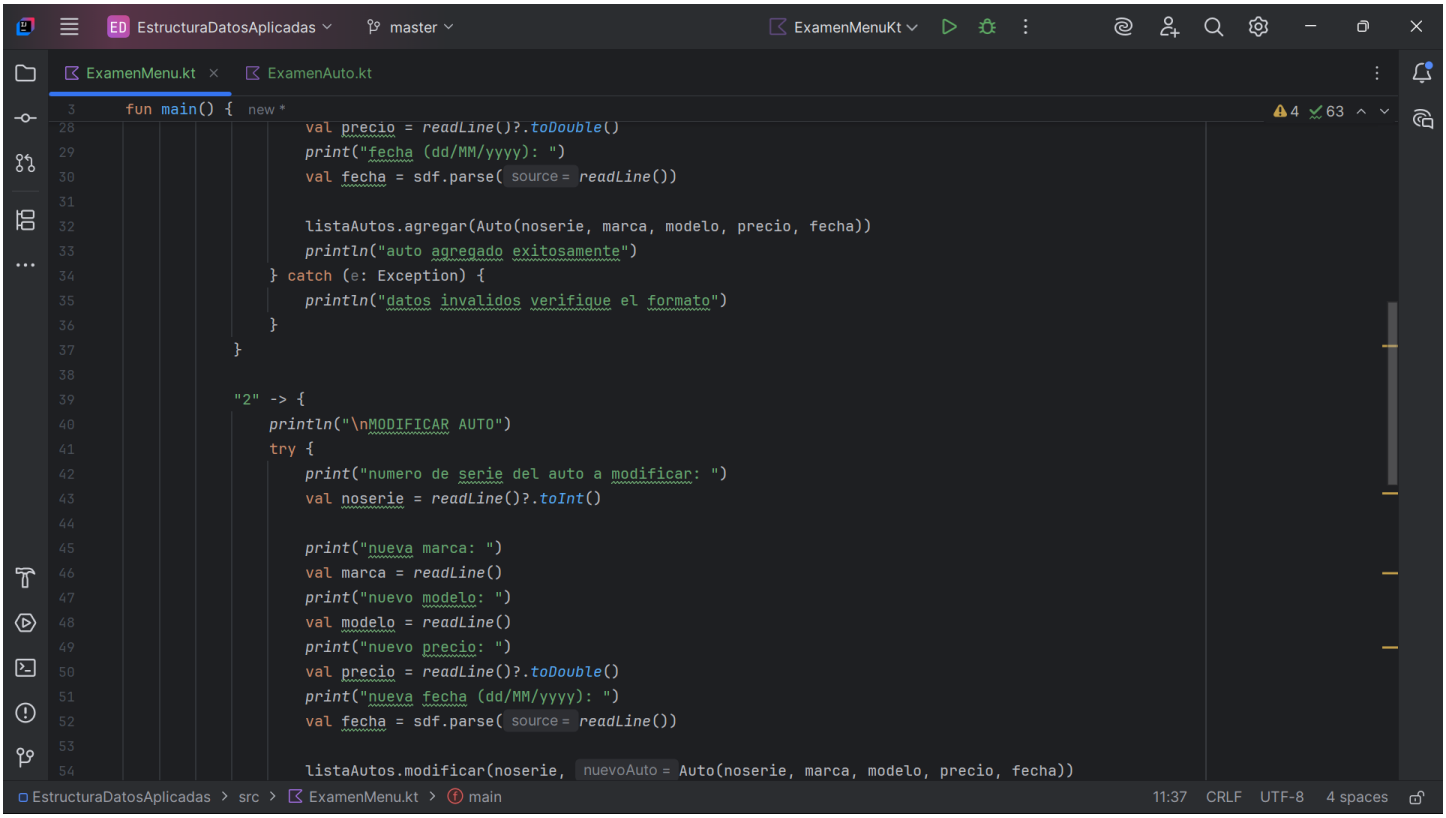
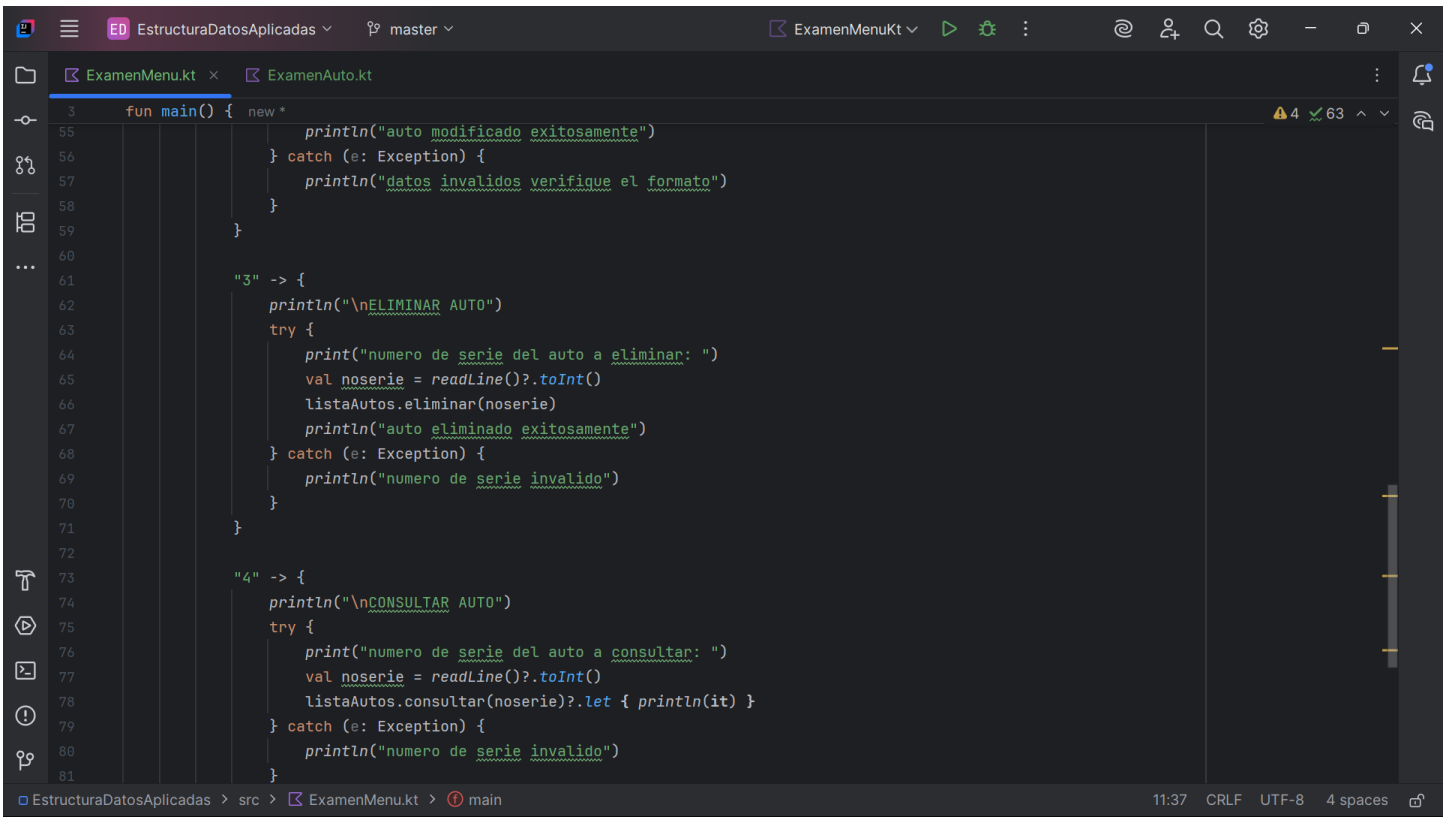

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
import java.text.SimpleDateFormat
import java.util.*
class Auto(6 Usages new *
var noserie: Int?,
var marca: String?,
var modelo: String?,
var precio: Double?,
var fecha: Date
) {
override fun toString(): String { new *
val sdf = SimpleDateFormat(pattern = "dd/MM/yyyy")
return "NoSerie: $noserie, Marca: $marca, Modelo: $modelo, Precio: $precio, Fecha: ${sdf.format(date = fecha)}"
}
}
class Nodo(val auto: Auto) { 3 Usages new *
var siguiente: Nodo? = null 10 Usages
}
class ListaAutos { 1 Usage new *
private var cabeza: Nodo? = null 11 Usages
fun agregar(auto: Auto) { 1 Usage new *
val nuevoNodo = Nodo(auto)
if (cabeza == null) {
cabeza = nuevoNodo
} else {
var actual = cabeza
while (actual?.siguiente != null) {
26
EstructuraDatosAplicadas > src > ExamenAuto.kt
3:1 CRLF UTF-8 4 spaces

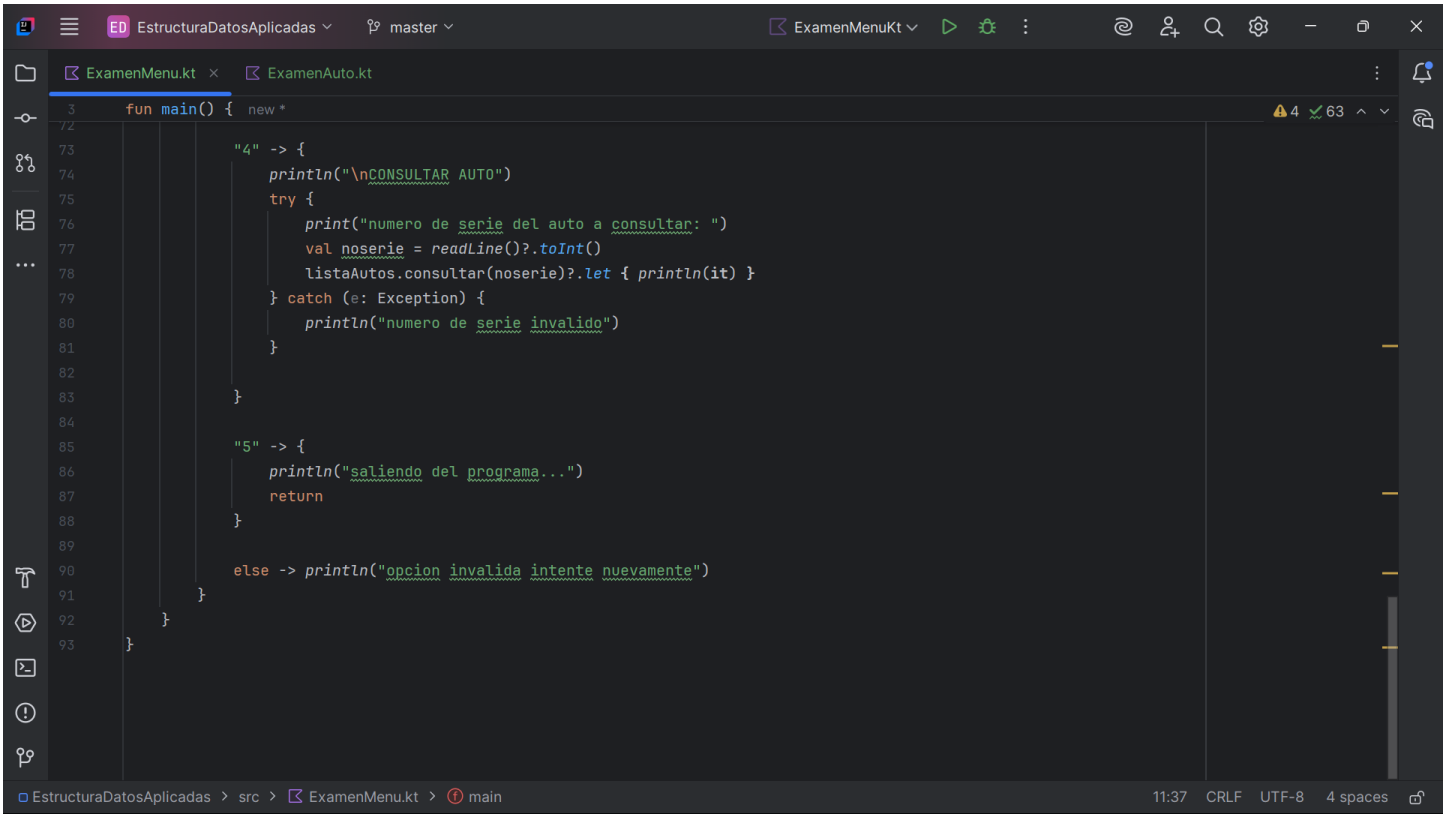

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
class ListaAutos { 1 Usage new *
fun agregar(auto: Auto) { 1 Usage new *
actual = actual.siguiente
}
actual?.siguiente = nuevoNodo
}
}
fun modificar(noserie: Int?, nuevoAuto: Auto) { 1 Usage new *
var actual = cabeza
while (actual != null) {
if (actual.auto.noserie == noserie) {
actual.auto.marca = nuevoAuto.marca
actual.auto.modelo = nuevoAuto.modelo
actual.auto.precio = nuevoAuto.precio
actual.auto.fecha = nuevoAuto.fecha
return
}
actual = actual.siguiente
}
println("auto con numero de serie $noserie no encontrado")
}
fun eliminar(noserie: Int?) { 1 Usage new *
if (cabeza == null) return
if (cabeza?.auto?.noserie == noserie) {
cabeza = cabeza?.siguiente
return
}
26
EstructuraDatosAplicadas > src > ExamenAuto.kt
3:1 CRLF UTF-8 4 spaces

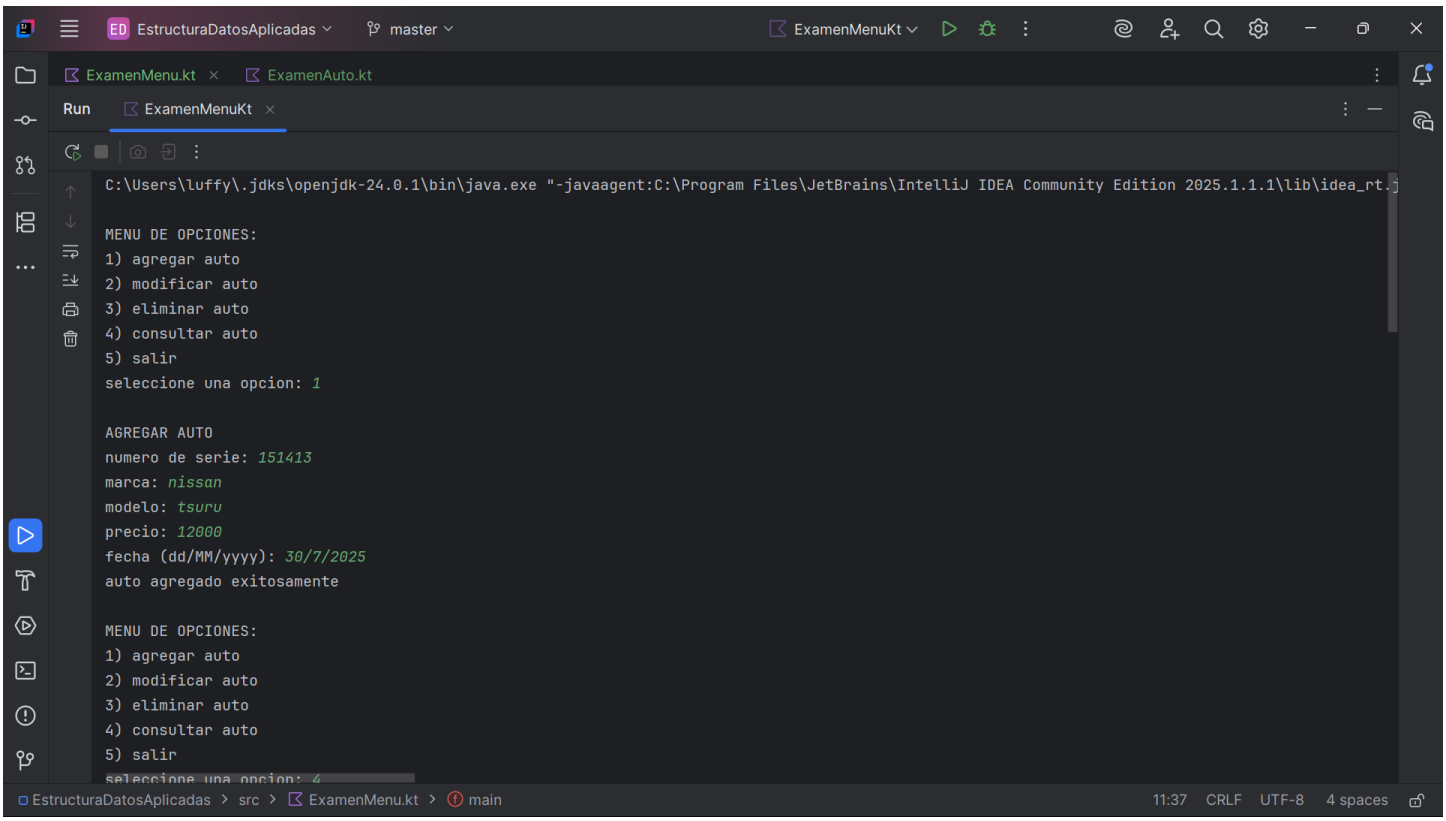

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
class ListaAutos { 1 Usage new *
fun eliminar(noserie: Int?) { 1 Usage new *
var anterior = cabeza
var actual = cabeza?.siguiente
while (actual != null) {
if (actual.auto.noserie == noserie) {
anterior?.siguiente = actual.siguiente
return
}
anterior = actual
actual = actual.siguiente
}
println("auto con numero de serie $noserie no encontrado")
}
fun consultar(noserie: Int?): Auto? { 1 Usage new *
var actual = cabeza
while (actual != null) {
if (actual.auto.noserie == noserie) {
return actual.auto
}
actual = actual.siguiente
}
println("Auto con número de serie $noserie no encontrado.")
return null
}
}
EstructuraDatosAplicadas > src > ExamenAuto.kt
3:1 CRLF UTF-8 4 spaces

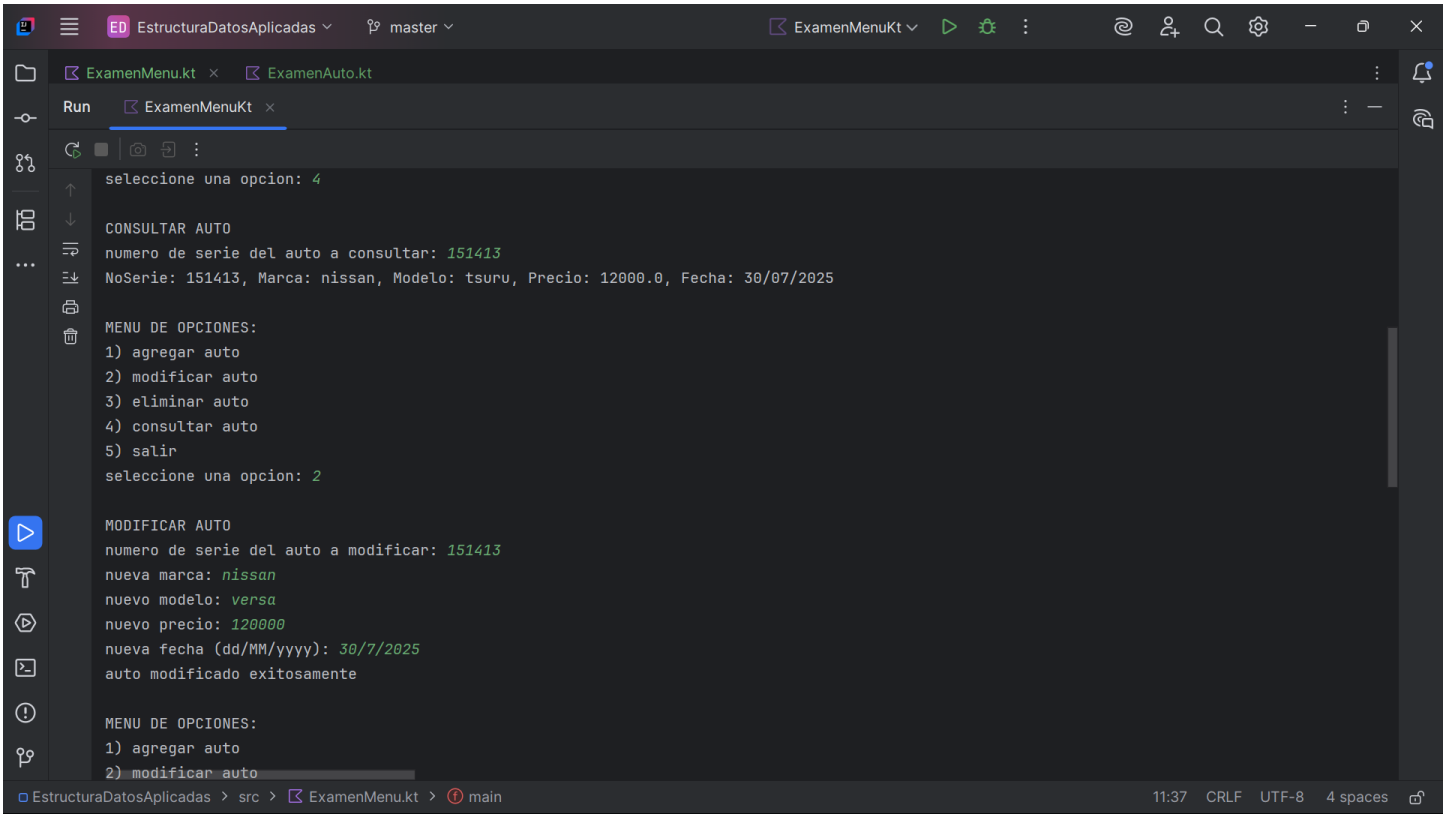

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
import java.text.SimpleDateFormat
fun main() { new *
val listaAutos = ListaAutos()
val sdf = SimpleDateFormat(pattern = "dd/MM/yyyy")
var opcion: String?
while (true) {
println("\nMENU DE OPCIONES:")
println("1) agregar auto")
println("2) modificar auto")
println("3) eliminar auto")
println("4) consultar auto")
println("5) salir")
print("seleccione una opcion: ")
opcion = readLine()
when (opcion) {
"1" -> {
println("\nAGREGAR AUTO")
try {
print("numero de serie: ")
val noserie = readLine()?.toInt()
print("marca: ")
val marca = readLine()
print("modelo: ")
val modelo = readLine()
print("precio: ")
EstructuraDatosAplicadas > src > ExamenMenu.kt > main
11:37 CRLF UTF-8 4 spaces

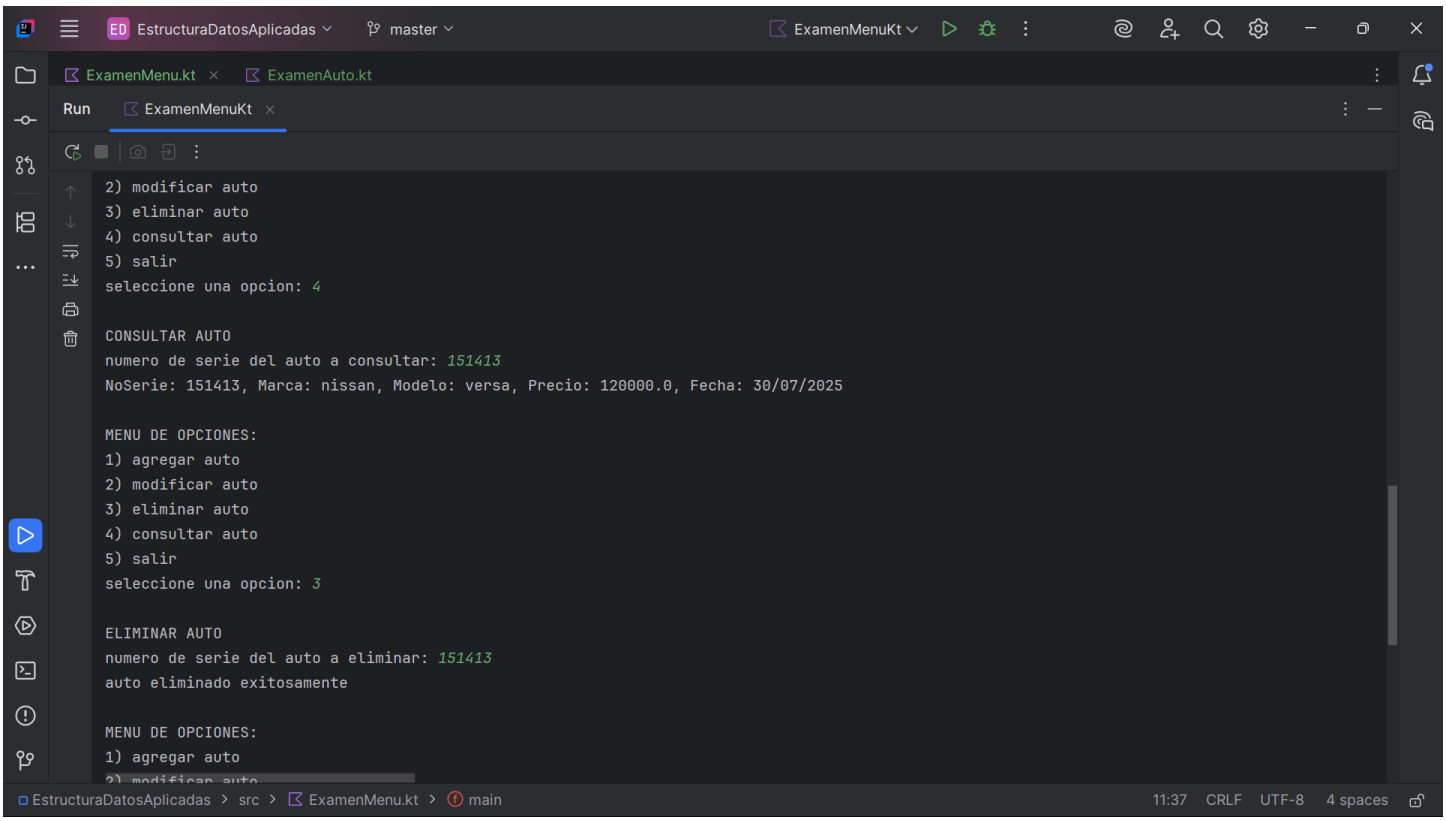

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
fun main() { new *
val precio = readLine()?.toDouble()
print("fecha (dd/MM/yyyy): ")
val fecha = sdf.parse(source = readLine())
listaAutos.agregar(Auto(noserie, marca, modelo, precio, fecha))
println("auto agregado exitosamente")
} catch (e: Exception) {
println("datos invalidos verifique el formato")
}
}
"2" -> {
println("\nMODIFICAR AUTO")
try {
print("numero de serie del auto a modificar: ")
val noserie = readLine()?.toInt()
print("nueva marca: ")
val marca = readLine()
print("nuevo modelo: ")
val modelo = readLine()
print("nuevo precio: ")
val precio = readLine()?.toDouble()
print("nueva fecha (dd/MM/yyyy): ")
val fecha = sdf.parse(source = readLine())
listaAutos.modificar(noserie, nuevoAuto = Auto(noserie, marca, modelo, precio, fecha))
EstructuraDatosAplicadas > src > ExamenMenu.kt > main
11:37 CRLF UTF-8 4 spaces

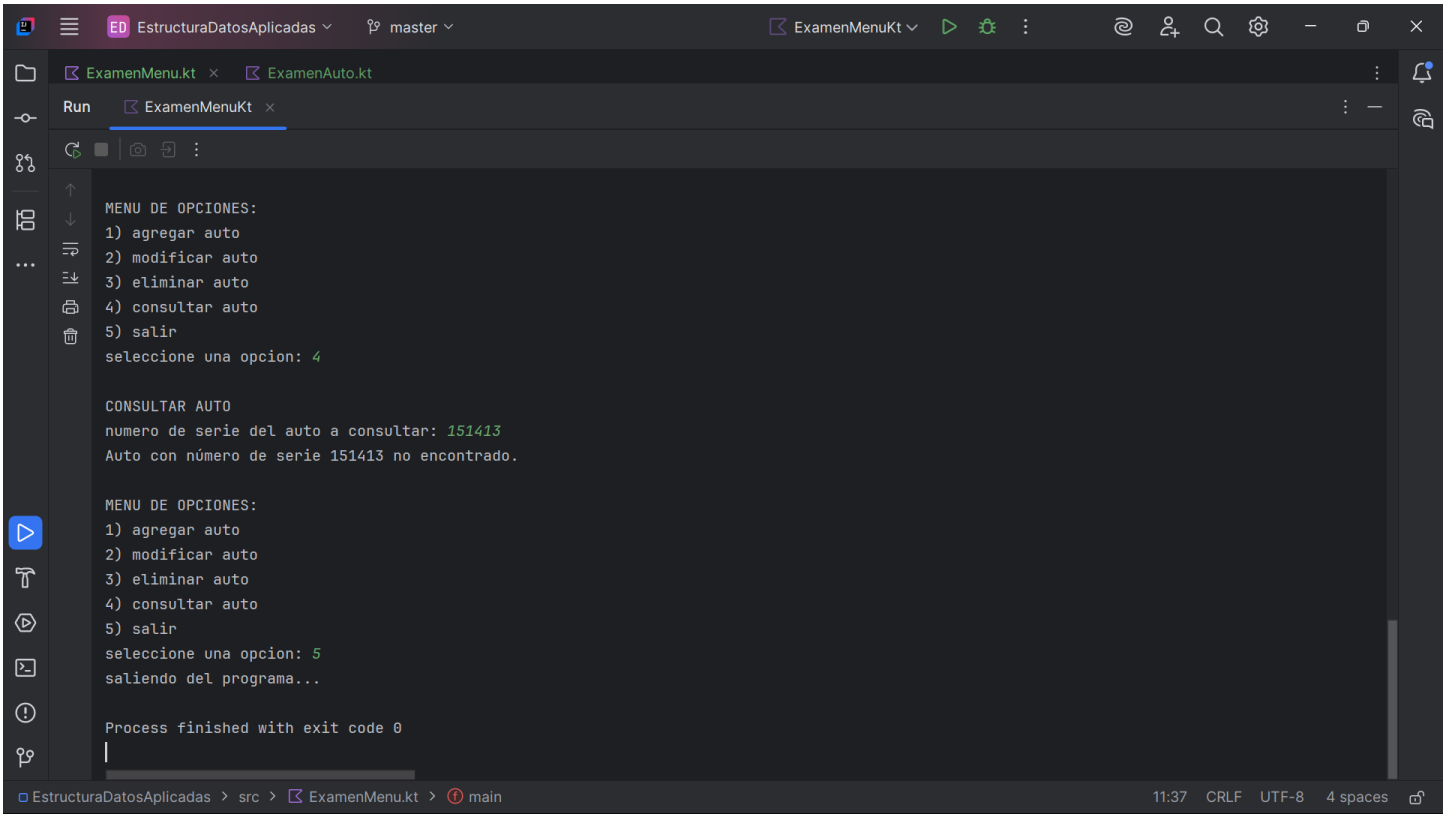

EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
fun main() { new *
println("auto modificado exitosamente")
} catch (e: Exception) {
println("datos invalidos verifique el formato")
}
}
"3" -> {
println("\nELIMINAR AUTO")
try {
print("numero de serie del auto a eliminar: ")
val noserie = readLine()?.toInt()
listaAutos.eliminar(noserie)
println("auto eliminado exitosamente")
} catch (e: Exception) {
println("numero de serie invalido")
}
}
"4" -> {
println("\nCONSULTAR AUTO")
try {
print("numero de serie del auto a consultar: ")
val noserie = readLine()?.toInt()
listaAutos.consultar(noserie)?.let { println(it) }
} catch (e: Exception) {
println("numero de serie invalido")
}
EstructuraDatosAplicadas > src > ExamenMenu.kt > main
11:37 CRLF UTF-8 4 spaces


EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
fun main() { new *
"4" -> {
println("\nCONSULTAR AUTO")
try {
print("numero de serie del auto a consultar: ")
val noserie = readLine()?.toInt()
listaAutos.consultar(noserie)?.let { println(it) }
} catch (e: Exception) {
println("numero de serie invalido")
}
}
"5" -> {
println("saliendo del programa...")
return
}
else -> println("opcion invalida intente nuevamente")
EstructuraDatosAplicadas › src › ExamenMenu.kt › main
11:37 CRLF UTF-8 4 spaces


EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
Run
ExamenMenuKt
C:\Users\luffy\.jdks\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.j
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 1
AGREGAR AUTO
numero de serie: 151413
marca: nissan
modelo: tsuru
precio: 12000
fecha (dd/MM/yyyy): 30/7/2025
auto agregado exitosamente
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 4
EstructuraDatosAplicadas › src › ExamenMenu.kt › main
11:37 CRLF UTF-8 4 spaces


EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
Run
ExamenMenuKt
seleccione una opcion: 4
CONSULTAR AUTO
numero de serie del auto a consultar: 151413
NoSerie: 151413, Marca: nissan, Modelo: tsuru, Precio: 12000.0, Fecha: 30/07/2025
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 2
MODIFICAR AUTO
numero de serie del auto a modificar: 151413
nueva marca: nissan
nuevo modelo: versa
nuevo precio: 120000
nueva fecha (dd/MM/yyyy): 30/7/2025
auto modificado exitosamente
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
EstructuraDatosAplicadas
src
ExamenMenu.kt
main
11:37
CRLF
UTF-8
4 spaces


EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
Run
ExamenMenuKt
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 4
CONSULTAR AUTO
numero de serie del auto a consultar: 151413
NoSerie: 151413, Marca: nissan, Modelo: versa, Precio: 120000.0, Fecha: 30/07/2025
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 3
ELIMINAR AUTO
numero de serie del auto a eliminar: 151413
auto eliminado exitosamente
MENU DE OPCIONES:
1) agregar auto
EstructuraDatosAplicadas
src
ExamenMenu.kt
main
11:37
CRLF
UTF-8
4 spaces


EstructuraDatosAplicadas
master
ExamenMenuKt
ExamenMenu.kt
ExamenAuto.kt
Run
ExamenMenuKt
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 4
CONSULTAR AUTO
numero de serie del auto a consultar: 151413
Auto con número de serie 151413 no encontrado.
MENU DE OPCIONES:
1) agregar auto
2) modificar auto
3) eliminar auto
4) consultar auto
5) salir
seleccione una opcion: 5
saliendo del programa...
Process finished with exit code 0
EstructuraDatosAplicadas
src
ExamenMenu.kt
main
11:37
CRLF
UTF-8
4 spaces