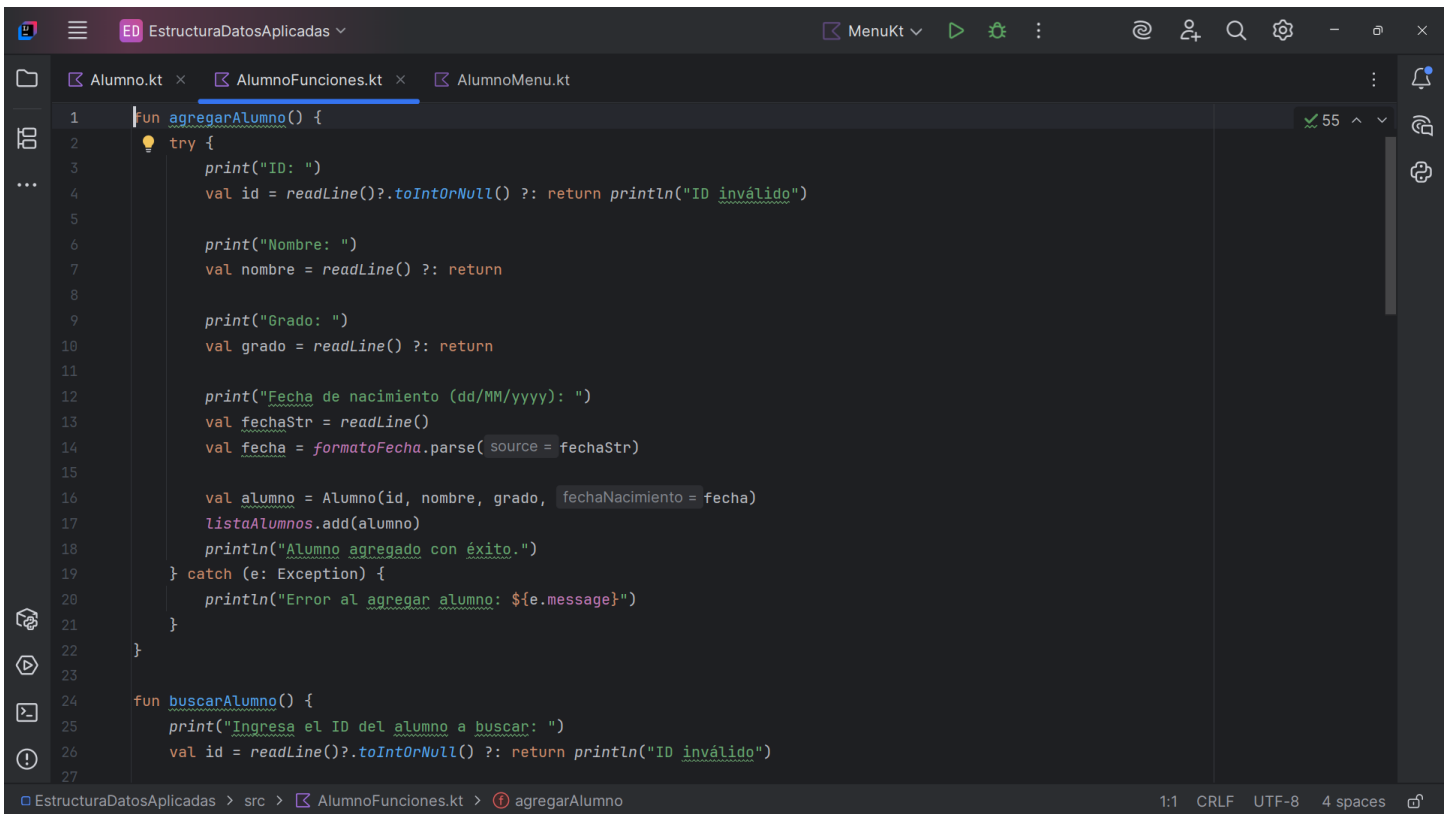


```
1 import java.text.SimpleDateFormat
2 import java.util.*
3
4 data class Alumno(var id: Int, var nombre: String, var grado: String, var fechaNacimiento: Date)
5
6 val listaAlumnos = arrayListOf<Alumno>()
7 val formatoFechaNacimiento = SimpleDateFormat(pattern = "dd/MM/yyyy")
8
```

EstructuraDatosAplicadas > src > Alumno.kt 8:1 CRLF UTF-8 4 spaces



```
1 fun agregarAlumno() {
2     try {
3         print("ID: ")
4         val id = readLine()?.toIntOrNull() ?: return println("ID inválido")
5
6         print("Nombre: ")
7         val nombre = readLine() ?: return
8
9         print("Grado: ")
10        val grado = readLine() ?: return
11
12        print("Fecha de nacimiento (dd/MM/yyyy): ")
13        val fechaStr = readLine()
14        val fecha = formatoFecha.parse(source = fechaStr)
15
16        val alumno = Alumno(id, nombre, grado, fechaNacimiento = fecha)
17        listaAlumnos.add(alumno)
18        println("Alumno agregado con éxito.")
19    } catch (e: Exception) {
20        println("Error al agregar alumno: ${e.message}")
21    }
22 }
23
24 fun buscarAlumno() {
25     print("Ingresa el ID del alumno a buscar: ")
26     val id = readLine()?.toIntOrNull() ?: return println("ID inválido")
27 }
```

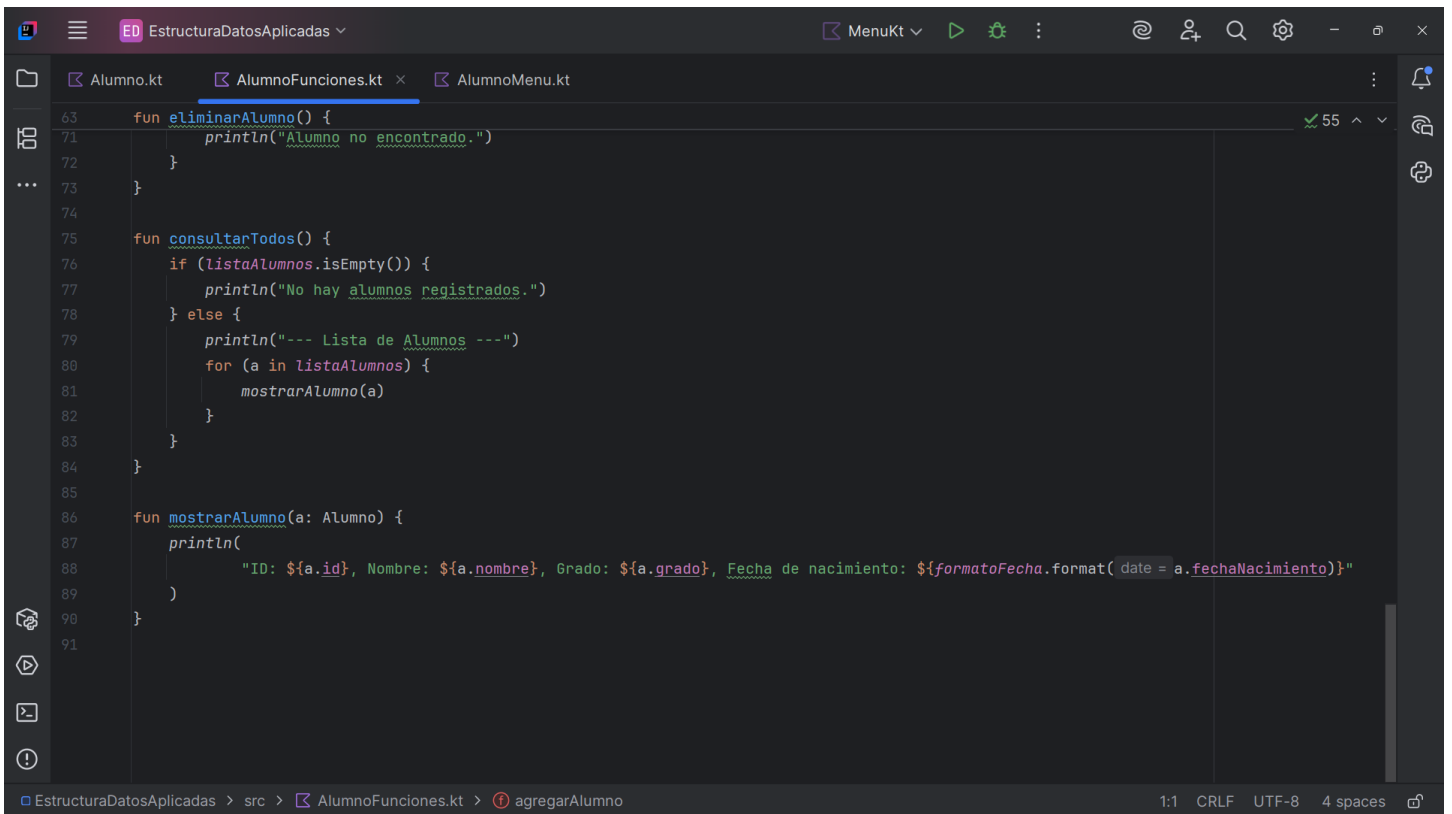
EstructuraDatosAplicadas > src > AlumnoFunciones.kt > agregarAlumno 1:1 CRLF UTF-8 4 spaces

```
ED EstructuraDatosAplicadas
MenuKt
Alumno.kt
AlumnoFunciones.kt
AlumnoMenu.kt

24 fun buscarAlumno() {
25     val alumno = listaAlumnos.find { it.id == id }
26     if (alumno != null) {
27         println("Alumno encontrado:")
28         mostrarAlumno(a = alumno)
29     } else {
30         println("Alumno no encontrado.")
31     }
32 }
33
34 fun modificarAlumno() {
35     print("Ingresa el ID del alumno a modificar: ")
36     val id = readLine()?.toIntOrNull() ?: return println("ID inválido")
37
38     val alumno = listaAlumnos.find { it.id == id }
39     if (alumno != null) {
40         print("Nuevo nombre ({alumno.nombre}): ")
41         val nombreNuevo = readLine()
42         if (!nombreNuevo.isNullOrEmpty()) alumno.nombre = nombreNuevo
43
44         print("Nuevo grado ({alumno.grado}): ")
45         val gradoNuevo = readLine()
46         if (!gradoNuevo.isNullOrEmpty()) alumno.grado = gradoNuevo
47
48         print("Nueva fecha de nacimiento (dd/MM/yyyy): ")
49         val fechaNuevaStr = readLine()
50         if (!fechaNuevaStr.isNullOrEmpty()) {
51             alumno.fechaNacimiento = formatoFecha.parse(source = fechaNuevaStr)
52         }
53     }
54     println("Alumno modificado correctamente.")
55 } else {
56     println("Alumno no encontrado.")
57 }
58 }
59
60 fun eliminarAlumno() {
61     print("Ingresa el ID del alumno a eliminar: ")
62     val id = readLine()?.toIntOrNull() ?: return println("ID inválido")
63
64     val eliminado = listaAlumnos.removeIf { it.id == id }
65     if (eliminado) {
66         println("Alumno eliminado.")
67     } else {
68         println("Alumno no encontrado.")
69     }
70 }
71
72 fun consultarTodos() {
73     if (listaAlumnos.isEmpty()) {
74         println("No hay alumnos registrados.")
75     } else {
76         println("--- Lista de Alumnos ---")
77     }
78 }
```

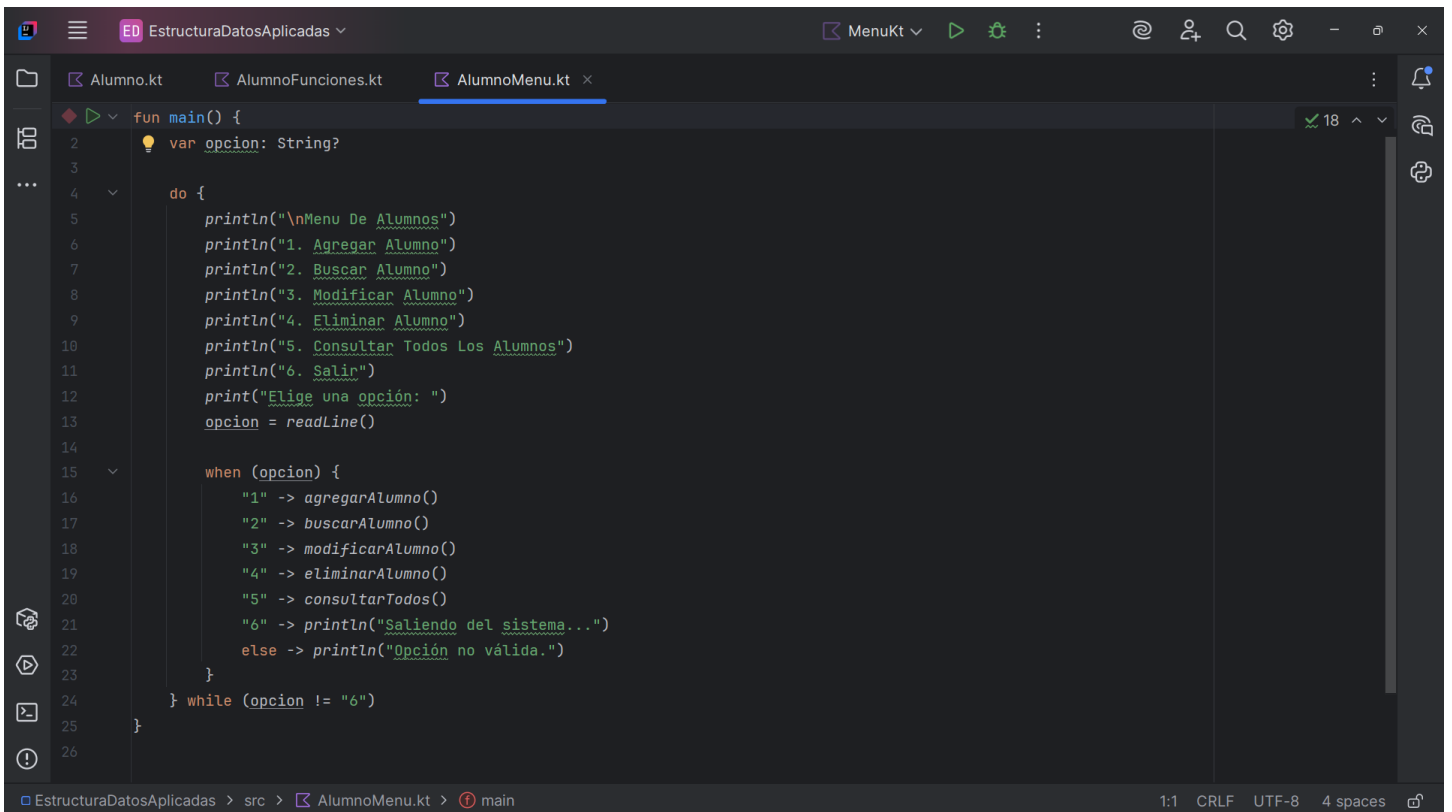
```
ED EstructuraDatosAplicadas
MenuKt
Alumno.kt
AlumnoFunciones.kt
AlumnoMenu.kt

37 fun modificarAlumno() {
54     alumno.fechaNacimiento = formatoFecha.parse(source = fechaNuevaStr)
55 }
56
57     println("Alumno modificado correctamente.")
58 } else {
59     println("Alumno no encontrado.")
60 }
61 }
62
63 fun eliminarAlumno() {
64     print("Ingresa el ID del alumno a eliminar: ")
65     val id = readLine()?.toIntOrNull() ?: return println("ID inválido")
66
67     val eliminado = listaAlumnos.removeIf { it.id == id }
68     if (eliminado) {
69         println("Alumno eliminado.")
70     } else {
71         println("Alumno no encontrado.")
72     }
73 }
74
75 fun consultarTodos() {
76     if (listaAlumnos.isEmpty()) {
77         println("No hay alumnos registrados.")
78     } else {
79         println("--- Lista de Alumnos ---")
80     }
81 }
```



```
63 fun eliminarAlumno() {
71     println("Alumno no encontrado.")
72 }
73 }
74
75 fun consultarTodos() {
76     if (listaAlumnos.isEmpty()) {
77         println("No hay alumnos registrados.")
78     } else {
79         println("--- Lista de Alumnos ---")
80         for (a in listaAlumnos) {
81             mostrarAlumno(a)
82         }
83     }
84 }
85
86 fun mostrarAlumno(a: Alumno) {
87     println(
88         "ID: ${a.id}, Nombre: ${a.nombre}, Grado: ${a.grado}, Fecha de nacimiento: ${formatoFecha.format(date = a.fechaNacimiento)}"
89     )
90 }
91 }
```

EstructuraDatosAplicadas > src > AlumnoFunciones.kt > agregarAlumno 1:1 CRLF UTF-8 4 spaces



```
fun main() {
2     var opcion: String?
3
4     do {
5         println("\nMenu De Alumnos")
6         println("1. Agregar Alumno")
7         println("2. Buscar Alumno")
8         println("3. Modificar Alumno")
9         println("4. Eliminar Alumno")
10        println("5. Consultar Todos Los Alumnos")
11        println("6. Salir")
12        print("Elige una opción: ")
13        opcion = readLine()
14
15        when (opcion) {
16            "1" -> agregarAlumno()
17            "2" -> buscarAlumno()
18            "3" -> modificarAlumno()
19            "4" -> eliminarAlumno()
20            "5" -> consultarTodos()
21            "6" -> println("Saliendo del sistema...")
22            else -> println("Opción no válida.")
23        }
24    } while (opcion != "6")
25 }
26 }
```

EstructuraDatosAplicadas > src > AlumnoMenu.kt > main 1:1 CRLF UTF-8 4 spaces

```
ED EstructuraDatosAplicadas AlumnoMenuKt
Run AlumnoMenuKt
C:\Users\luffy\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.jar"
Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 1
ID: 1
Nombre: Erick
Grado: 4
Fecha de nacimiento (dd/MM/yyyy): 21/09/2001
Alumno agregado con éxito.

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 1
```

```
ED EstructuraDatosAplicadas AlumnoMenuKt
Run AlumnoMenuKt
Elige una opción: 1
ID: 2
Nombre: Ivan
Grado: 3
Fecha de nacimiento (dd/MM/yyyy): 12/11/2005
Alumno agregado con éxito.

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 1
ID: 3
Nombre: Daniela
Grado: 2
Fecha de nacimiento (dd/MM/yyyy): 29
Error al agregar alumno: Unparseable date: "29"

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
```

The screenshot shows an IDE with the following components:

- Top Bar:** Includes icons for file explorer, search, and other IDE functions. The active file is `AlumnoMenuKt`.
- Editor:** Displays the `AlumnoMenuKt` file. The code is in Kotlin and defines a menu for managing students. The menu options are:
 - 1. Agregar Alumno
 - 2. Buscar Alumno
 - 3. Modificar Alumno
 - 4. Eliminar Alumno
 - 5. Consultar Todos Los Alumnos
 - 6. Salir
- Run Console:** Shows the output of the program. The user has selected option 5, and the program displays a list of two students:
 - ID: 1, Nombre: Erick, Grado: 4, Fecha de nacimiento: 21/09/2001
 - ID: 2, Nombre: Ivan, Grado: 3, Fecha de nacimiento: 12/11/2005

The screenshot shows an IDE with the following components:

- Top Bar:** Displays the project name "EstructuraDatosAplicadas" and the active file "AlumnoMenuKt".
- Run Bar:** Shows the "Run" button and the active file "AlumnoMenuKt".
- Terminal/Console:** Displays the output of the program execution.

```
Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 2
Ingresa el ID del alumno a buscar: 2
Alumno encontrado:
ID: 2, Nombre: Ivan, Grado: 4, Fecha de nacimiento: 12/11/2005

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 4
Ingresa el ID del alumno a eliminar: 1
Alumno eliminado.
```
- Bottom Bar:** Shows the file path "EstructuraDatosAplicadas > src > AlumnoMenu.kt" and the current line and column "1:1 CRLF UTF-8 4 spaces".

ED EstructuraDatosAplicadas

AlumnoMenuKt

AlumnoMenuKt

AlumnoMenuKt

Run AlumnoMenuKt

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 5
--- Lista de Alumnos ---
ID: 2, Nombre: Ivan, Grado: 4, Fecha de nacimiento: 12/11/2005

Menu De Alumnos
1. Agregar Alumno
2. Buscar Alumno
3. Modificar Alumno
4. Eliminar Alumno
5. Consultar Todos Los Alumnos
6. Salir
Elige una opción: 6
Saliendo del sistema...

Process finished with exit code 0

EstructuraDatosAplicadas > src > AlumnoMenu.kt > main 1:1 CRLF UTF-8 4 spaces