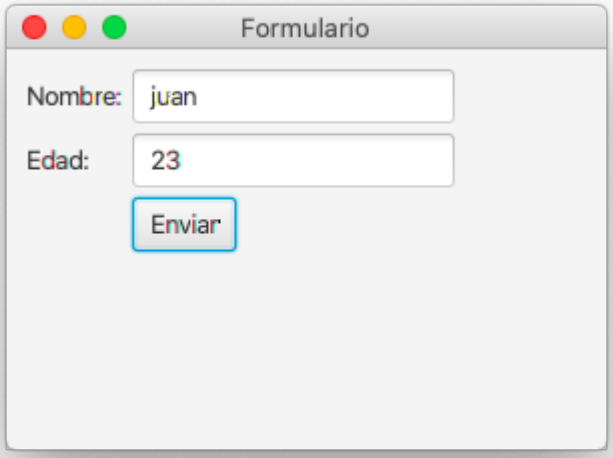
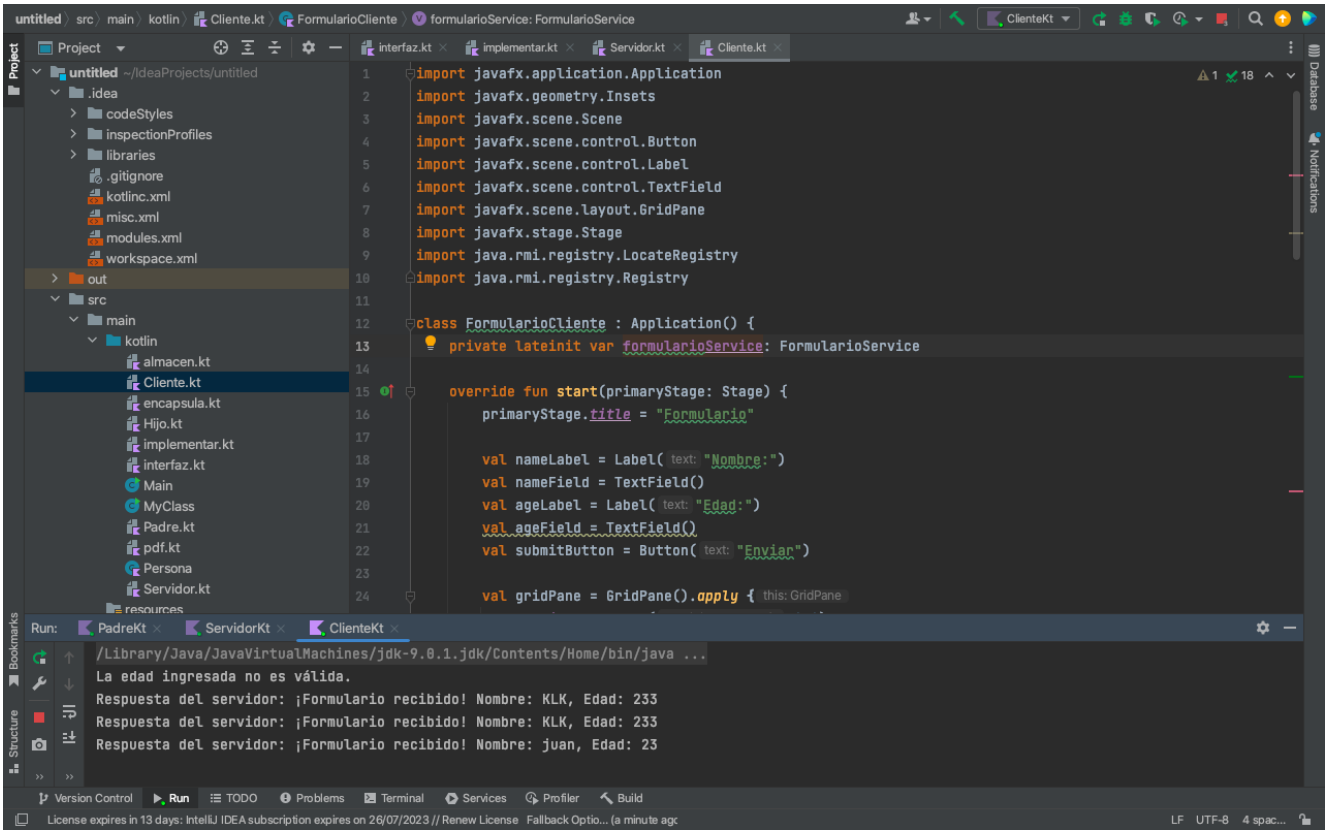
Envío de datos de un formulario entre cliente y servidor mediante RMI en JavaFX con Kotlin

Ejemplo simplificado de cómo implementar el envío de un formulario en JavaFX utilizando RMI entre un cliente y un servidor en Kotlin.





Interfaz remota (FormularioService.kt)

```
import java.rmi.Remote
import java.rmi.RemoteException

interface FormularioService : Remote {
    @Throws(RemoteException::class)
    fun enviarFormulario(nombre: String, edad: Int): String
}
```

Implementación del servicio remoto (FormularioServiceImpl.kt)

```
import java.rmi.RemoteException
import java.rmi.server.UnicastRemoteObject

class FormularioServiceImpl : UnicastRemoteObject(), FormularioService {
    @Throws(RemoteException::class)
```

```
    override fun enviarFormulario(nombre: String, edad: Int):
String {
        // Aquí puedes realizar cualquier lógica adicional con
los datos recibidos
         return "¡Formulario recibido! Nombre: $nombre, Edad:
$edad"
    }
}
```

Servidor RMI (FormularioServer.kt)

```
import java.rmi.registry.LocateRegistry
import java.rmi.registry.Registry

fun main() {
    try {
        val service = FormularioServiceImpl()

                        val   registry:   Registry   =
LocateRegistry.createRegistry(1099)
        registry.rebind("FormularioService", service)

            println("Servidor RMI iniciado. Esperando
conexiones...")
    } catch (e: Exception) {
        println("Error en el servidor: ${e.message}")
        e.printStackTrace()
    }
}
```

Cliente JavaFX (FormularioCliente.kt)

```
import javafx.application.Application
import javafx.geometry.Insets
import javafx.scene.Scene
import javafx.scene.control.Button
import javafx.scene.control.Label
import javafx.scene.control.TextField
import javafx.scene.layout.GridPane
```

```
import javafx.stage.Stage
import java.rmi.registry.LocateRegistry
import java.rmi.registry.Registry

class FormularioCliente : Application() {
    private lateinit var formularioService: FormularioService

    override fun start(primaryStage: Stage) {
        primaryStage.title = "Formulario"

        val nameLabel = Label("Nombre:")
        val nameField = TextField()
        val ageLabel = Label("Edad:")
        val ageField = TextField()
        val submitButton = Button("Enviar")

        val gridPane = GridPane().apply {
            padding = Insets(10.0)
            hgap = 5.0
            vgap = 5.0
        }

        gridPane.add(nameLabel, 0, 0)
        gridPane.add(nameField, 1, 0)
        gridPane.add(ageLabel, 0, 1)
        gridPane.add(ageField, 1, 1)
        gridPane.add(submitButton, 1, 2)

        submitButton.setOnAction {
            val nombre = nameField.text
            val edad = ageField.text.toIntOrNull()

            if (edad != null) {
                                    val resultado =
formularioService.enviarFormulario(nombre, edad)
                println("Respuesta del servidor: $resultado")
            } else {
                println("La edad ingresada no es válida.")
            }
        }
```

```
        val scene = Scene(gridPane, 300.0, 200.0)

        primaryStage.scene = scene
        primaryStage.show()
    }

    override fun init() {
        try {
                            val registry: Registry =
LocateRegistry.getRegistry("localhost", 1099)
                                formularioService =
registry.lookup("FormularioService") as FormularioService
        } catch (e: Exception) {
              println("Error al conectarse al servidor:
${e.message}")
            e.printStackTrace()
        }
    }
}

fun main(args: Array<String>) {
    Application.launch(FormularioCliente::class.java, *args)
}
```

En este ejemplo, se crea una interfaz remota FormularioService que define el método enviarFormulario() para enviar el formulario desde el cliente al servidor. Luego, se implementa la clase FormularioServiceImpl que proporciona la implementación del servicio remoto.

En el servidor RMI (FormularioServer.kt), se inicia el servicio remoto y se vincula al registro RMI en el puerto 1099.

En el cliente JavaFX (FormularioCliente.kt), se crea una interfaz de usuario con campos de texto para el nombre y la edad, así como un botón para enviar el formulario. Al hacer clic en el botón, se recopilan los datos del formulario y se llama al método enviarFormulario() del servicio remoto en el servidor.

Es importante tener en cuenta que este es un ejemplo simplificado y no incluye medidas de seguridad ni manejo de excepciones completo. Para un uso en producción, se recomienda agregar validaciones, manejo de errores y considerar aspectos de seguridad adecuados.

Recuerda que también debes asegurarte de que los archivos de clases se compilen y ejecuten correctamente en el entorno en el que estás trabajando y que se configuren adecuadamente los archivos de políticas de seguridad para permitir la comunicación RMI.