

The background of the page is an abstract composition. It features a dense cluster of thin, dark blue and black lines that sweep across the left side of the frame. Interspersed among these lines are numerous small, solid black dots. The overall effect is one of dynamic movement and complexity, reminiscent of a network diagram or a microscopic view of a material structure. The right side of the page is mostly white, providing a clean space for the text.

# ENVIARXMLPROYECTO

---

AUTOR: Magdalena Rios Guardiola

## **Indice:**

1. Especificaciones
2. Diseño
3. Requisitos
4. Instalación
5. Pruebas
6. Instrucciones de ejecución

# 1. Especificaciones

Los argumentos a pasar al jar serian en el siguiente orden IP, puerto, identificador.

Como no se han especificado restricciones sobre los argumentos se ha optado por aplicar estos:

1. IP sirve cualquier ip4 valida con valores decimales.
2. Puerto sirve cualquier puerto valido entre 0 y 65535
3. Identificador sirve cualquier String

# 2. Diseño

Es un proyecto Maven hecho en IntelliJ con Test de integración basados en Junit. Se ha programa esta hecho en idioma Java usando JDK17, se prefirió sobre el 11 lts recomendado debido a las recomendaciones de seguridad de Oracle.

Se separaron la interfaz de usuario de la lógica de negocio (proyecto),

Por un lado estaría la interfaz de usuario que seria la clase Ejecutable donde se iniciaría el programa y interactuaría con el usuario. En esta parte seria donde entra, por linea de comandos, los argumentos. En caso de no proporcionarlos por linea de comando se pedirían vía consola.

Por otro lado tendríamos la lógica de negocio la clase ConexionCliente y los paquetes de actions, exceptions y utils.

- La clase ConexionCliente tal como su nombre indica seria para la conexión con el cliente, enviar los datos y cerrar dicha conexión una vez finalizado el envío.
- En el paquete actions se encontraría la clase EnviarXmlActions crearía el XML y lo enviaría.
- En el paquete exceptions se encuentran las distintas clases donde se recogerían las excepciones: ErrorCerrandoConexionException, ErrorDeConexionException, ErrorEnviandoDatosException, IPNOValidaException, PuertoNOValidoException
- En el paquete utils se encontraría la clase ValidacionUtil que valida tanto el puerto como la Ip para que cumplen los requisitos exigidos.
- Se generan una serie de test de integración para la realización del código, a partir de los cuales se realizaron los métodos ( Tdd).

### 3. Requisitos

Se necesita como mínimo debido a la compilación con JDK17 el tener instalado JRE 17.0.7.

### 4. Instalación

Tener instalado JRE 17.0.7. y en caso de querer usar el .exe disponer del .jar en una misma carpeta.

### 5. Pruebas

Usando Junit se han testado los distintos métodos de la clase ValidacionUtil.

En el caso de las clases ConexionCliente y EnviarXmlAction se han usado pruebas de integración. En las que es necesario lanzar dos test simultáneamente primero se lanzaría uno de la clase ServerITTest (con dos objetivos primero crear un servidor y segundo confirmar que ese servidor recibe los datos correctos) y posteriormente se lanzaría uno de la clase EnviarXmlActionTest, los test están emparejados de la siguiente forma:

- envioDatosCorrectos con recepcionServeCorrecto.
- envioDatosINCorrectos con recepcionServeInCorrecto.

### 6. Instrucciones de Ejecución

Para ejecutar el programa solo hay que acceder desde una consola y escribir los siguientes comandos:

```
java -jar EnviarXMLProyecto.jar ip puerto identificador
```

solo hay que sustituir los argumentos por los valores deseados.