

## MANUAL DE USUARIO

**Nombre del proyecto:** Sistema de Monitoreo por Puerto Serial

**Plataforma:** Windows

**Lenguaje:** Java

**Entorno de desarrollo:** Visual Studio Code

---

### Requisitos del Sistema

- Sistema operativo: Windows 10 o superior
  - Java JDK 8 o superior instalado
  - Cable USB para conectar el dispositivo externo (Arduino)
  - Biblioteca `jSerialComm-2.11.0.jar` incluida
- 

### Instalación y Ejecución

1. **Descargar y descomprimir** el proyecto.
  2. Abrir la carpeta `PROYECTO-main` en Visual Studio Code.
  3. Asegurarse de tener configurado el entorno Java en VSCode.
  4. Conectar el dispositivo Arduino por puerto USB.
  5. Compilar y ejecutar el archivo principal (`MenuComando.java` o `arduinoc.java`).
  6. El sistema comenzará a mostrar las pantallas de interacción, lectura de datos o comandos al Arduino.
- 

### Uso del Sistema

- Al iniciar, se presentará una **pantalla inicial** donde se debe seleccionar una opción.
  - Se pueden enviar comandos al Arduino desde el menú o visualizar datos.
  - El sistema permite la **lectura de datos** del Arduino y su visualización en pantalla.
  - El sistema puede tener varias **pantallas secundarias** que gestionan entradas y salidas del hardware.
- 

### Flujo de Uso del Sistema

#### 1. Pantalla Inicial (`pantallainicial1`)

- Pantalla gráfica que contiene botones principales.
- Al iniciar, intenta detectar el puerto serial de Arduino automáticamente.
- Muestra botones de navegación a otras pantallas como `Pantalla2` o `MenuComando`.

#### 2. Pantalla de Comandos (`MenuComando`)

- Permite interactuar con Arduino usando botones virtuales.
  - Envía comandos al puerto serial.
  - Puede manejar órdenes como subir volumen, cambiar canal, o activar funciones.
-

### 3. Pantalla Visual (**Pantalla2**)

- Muestra imágenes o números.
- Reacciona tanto a botones GUI como a señales del control remoto IR.
- Si el usuario presiona un botón en el control, se cambia la pantalla actual.

## Control Remoto IR

- El sistema reconoce códigos IR que llegan desde Arduino.
- Los códigos activan distintas pantallas o acciones:
  - Cambiar a **Pantalla2** -> Boton 100+
  - Cambiar a **Menu Comando** -> Boton 200+
  - Mostrar números del 0 al 9 -> Botones del 0 al 9
  - Volumen y navegación -> Botones de volumen y navegación
  - Salir del Programa -> Botón EO



## Funcionalidad de Base de Datos

- Archivo local **datos\_arduino.db**
- Se utiliza para guardar historial de comandos o eventos
- Clases relacionadas:
  - **ArduinoDataDAO**: conexión a la base y queries
  - **DatabaseInitializer**: crea las tablas si no existen

## Solución de Problemas

Problema	Solución
No se detecta Arduino	Verifica conexión y que el puerto COM esté libre
No responde el control IR	Asegúrate de que el Arduino esté enviando los códigos IR correctamente
Error: No hay atributo Main-Class	Verifica el contenido del <b>manifest.txt</b>

## Créditos y Contacto

### Autores: [

- Galarraga Andres
- Churuchumbi Sofia
- Calapi Kevin
- Charanchi Kevin
- Carvajal Alan

]

**GitHub:** <https://github.com/sofi-hobi/PROYECTO>

**Licencia:** Uso educativo