



v2.2

SerloT App

Manual de usuario para la configuración y gestión de dispositivos IoT mediante interfaz gráfica de escritorio.

Plataforma Windows 10 / 11 x64

Modelos SERIOT 3 / SERLOG

Backend seriot-0.0.1-SNAPSHOT.jar

Revisión 2026

Tabla de Contenidos

INTRODUCCIÓN

- 1 Descripción general
- 2 Requisitos del sistema
- 3 Instalación
- 4 Primer arranque

CONFIGURACIÓN

- 5 Panel principal
- 6 Configuración serie
- 7 Configuración WiFi
- 8 Conexión móvil (GPRS)
- 9 Sincronización de hora

VARIABLES MODBUS

- 10 Gestión de variables
 - ↳ Tipos de datos
 - ↳ Tipos de registro
 - ↳ Crear / editar variable
 - ↳ Ejemplos prácticos
- 11 Importar / Exportar

REFERENCIA

- 12 API interna
- 13 Mensajes de error
- 14 Atajos de teclado
- 15 Glosario

HARDWARE SERIOT

- 16 Descripción del dispositivo
- 17 Instalación y conexiones
- 18 Indicadores LED
- 19 Comandos AT
- 20 Configuración remota SMS
- 21 Firmware y memoria

1. Descripción general

SerIoT App es una aplicación de escritorio que proporciona una interfaz gráfica para configurar dispositivos SerIoT conectados al PC.

Comunicación serie

Detección automática del dispositivo por puerto COM y configuración de parámetros serie.

Redes WiFi / Móvil

Ajuste de conectividad inalámbrica (SSID, IP) y datos móviles GPRS del dispositivo.

Sincronización horaria

Sincroniza el reloj interno del dispositivo con la hora del sistema operativo.

Variables Modbus

Creación y gestión de variables con 11 tipos de dato y registros Holding / Input / Write.

Importar / Exportar


Guarda y restaura configuraciones completas del dispositivo en archivos locales.

Data Logger

Soporte especial para dispositivos en modo registrador de datos sin conexión de red.

2. Requisitos del sistema


Componente	Mínimo	Recomendado
Sistema operativo	Windows 10 x64	Windows 10 / 11 x64
RAM	2 GB	4 GB o más
Espacio en disco	500 MB	1 GB
Java Runtime	JRE 8 o superior (incluido en el paquete)	
Puerto USB / COM	Disponibile para el dispositivo SerIoT	
Puerto de red local	Puerto 8080 libre en localhost	

 El puerto **8080** debe estar libre en `localhost`. Si otro servicio lo ocupa, el backend no arrancará y la aplicación mostrará el error "*Service Down*".

3. Instalación

La aplicación se distribuye a través de un ejecutable (InstallSeriotV2.2AppX64 última versión). El cual ejecutará todo el proceso de instalación.

- 1 Descomprime `InstallSeriotV2.2AppX64` en la carpeta deseada.
- 2 Instale. Siga el paso a paso.
- 3 Conecta el dispositivo SerIoT al PC mediante cable USB **antes** de abrir la aplicación.
- 4 Ejecuta `SeriotAppV2.exe` con doble clic.

 Se creará un acceso directo en el escritorio apuntando a `SeriotAppV2.exe` para facilitar el acceso diario.

4. Primer arranque

Al ejecutar la aplicación ocurren los siguientes pasos de forma automática:

- 1 Se muestra la **pantalla de splash** animada mientras se inicializa el entorno.
- 2 El proceso principal enumera los puertos COM en busca de un dispositivo SerIoT.
- 3 Se lanza el backend Java en segundo plano. Se realizan hasta **5 intentos** de conexión al endpoint `/check-health`.
- 4 Una vez el backend responde, se carga el **panel principal** con la información del dispositivo detectado.

Pantalla: dispositivo no encontrado



Fig. 1 — Pantalla mostrada cuando no hay dispositivo SerIoT conectado al iniciar.

i Si ves esta pantalla con el dispositivo conectado, desconecta y reconecta el cable USB, espera 5 segundos y reinicia la aplicación.

5. Panel principal (Dashboard)

Tras la detección exitosa del dispositivo se muestra el panel principal con un resumen del estado actual en tiempo real.

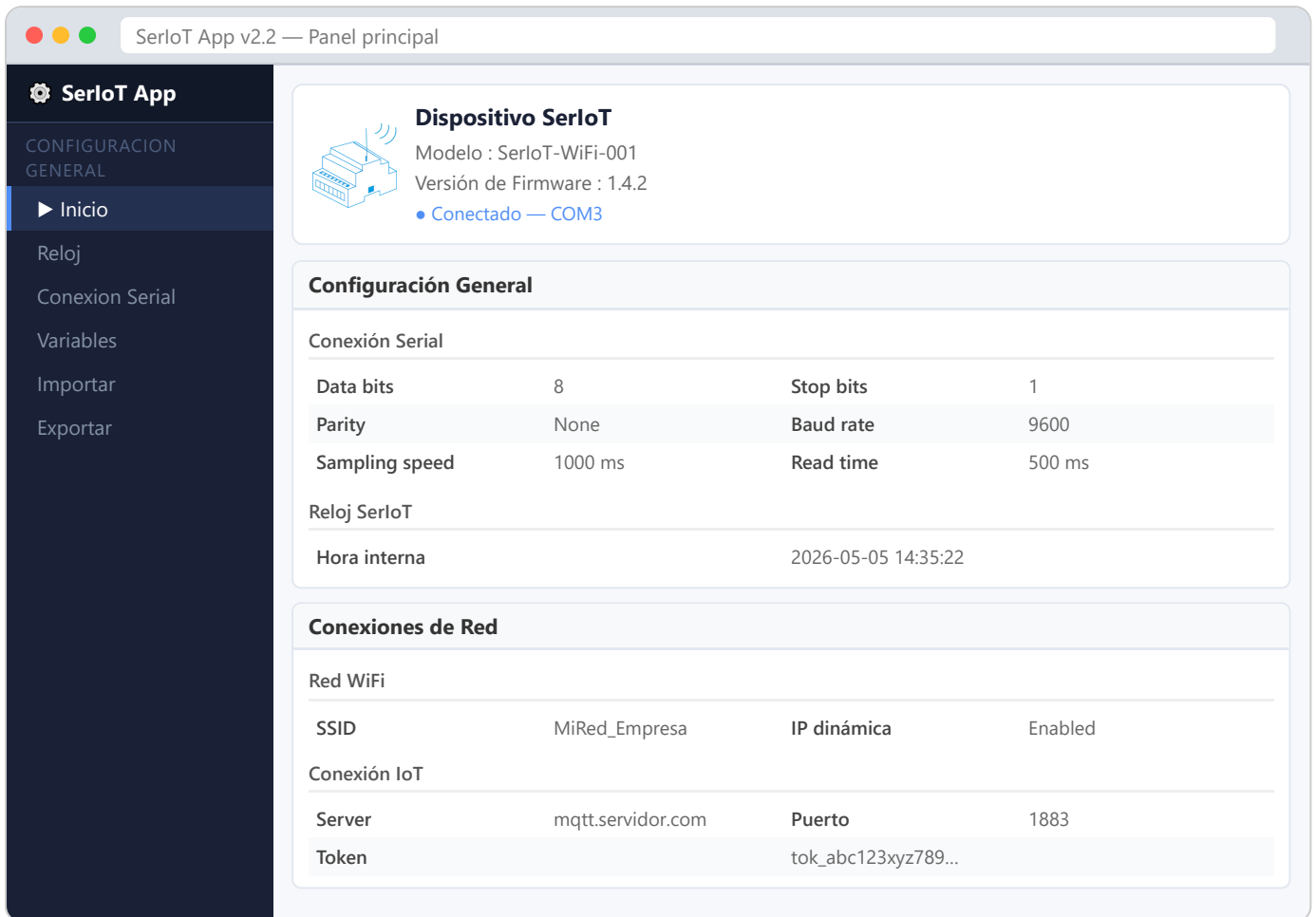


Fig. 2 — Panel principal con estado del dispositivo, configuración serie y red.

6. Configuración serie

Ajusta los parámetros de la conexión serie entre el PC y el dispositivo. Los cambios se aplican al pulsar **Guardar**.



Fig. 3 — Formulario de configuración de conexión serie.

Parámetro	Opciones	Descripción
Data bits	7 · 8	Bits de datos por trama
Stop bits	1 · 2	Bits de parada por trama
Parity	None · Odd · Even	Control de paridad
Baud rate	1200 · 2400 · 4800 · 9600 · 19200 · 38400 · 57600 · 115200	Velocidad de transmisión
Sampling time	Número en ms	Intervalo entre lecturas de variables
Read time	Número en ms	Tiempo máximo de espera por respuesta Modbus

Ejemplo — Configuración para PLC Siemens S7-1200 (Modbus RTU)

Un PLC Siemens S7-1200 con comunicación Modbus RTU suele requerir paridad *Even* a 19 200 baud:

Data bits

8

Baud rate

19200

Stop bits

1

Sampling time

2000

Parity

Even

Read time

1000



Una paridad incorrecta provoca errores de trama y pérdida de datos. Consulta el manual del PLC para confirmarlo.

8. Conexión móvil (GPRS)

Disponible para dispositivos de tipo **Móvil**. Configura la conexión de datos mediante red celular.

Campo	Descripción	Ejemplo
APN	Nombre del punto de acceso del operador	(ej. <code>internet.operador.com</code>)
Usuario	Nombre de usuario autenticación de operador (si aplica)	(ej. <code>movistar</code>)
Password	Contraseña del APN (si aplica)	(ej. <code>movistar</code>)

Ejemplo — APN para operadores comunes

Movistar Colombia

APN

Usuario

Password

Claro Colombia

APN

Usuario

Password

9. Reloj de hora

Sincroniza reloj sistema con PC. **Sincronizar**.

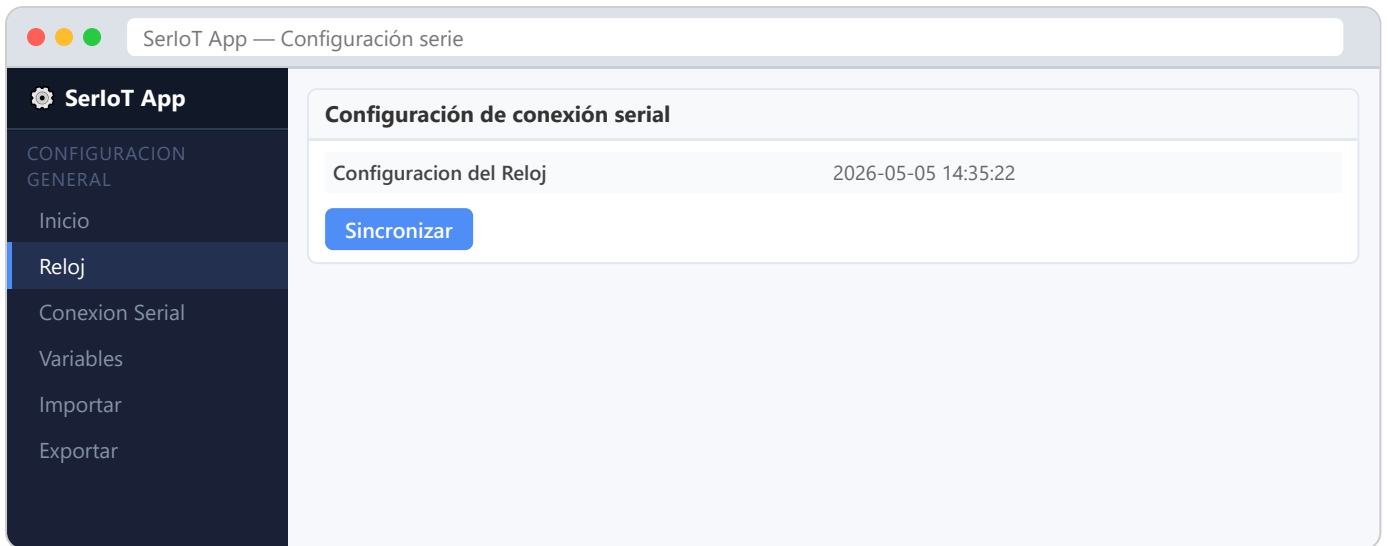


Fig. 4 — Formulario de sincronización.

Sincronización de hora

Sincroniza el reloj interno del dispositivo con la hora del sistema operativo donde se ejecuta SerloT App.

- 1 Accede a **Configuración** → **Hora** en el menú.
- 2 La pantalla muestra la hora actual del PC en tiempo real.
- 3 Pulsa **Sincronizar** para enviar la hora al dispositivo.

 Asegúrate de que la hora del PC sea correcta (incluyendo zona horaria) antes de sincronizar.

10. Gestión de variables

El módulo de variables permite definir, editar y eliminar las variables Modbus que el dispositivo lee de los equipos conectados. Cada variable mapea un registro Modbus a un nombre legible por la plataforma IoT.



Fig. 5 — Tabla de variables con fila 1 expandida mostrando campos secundarios.

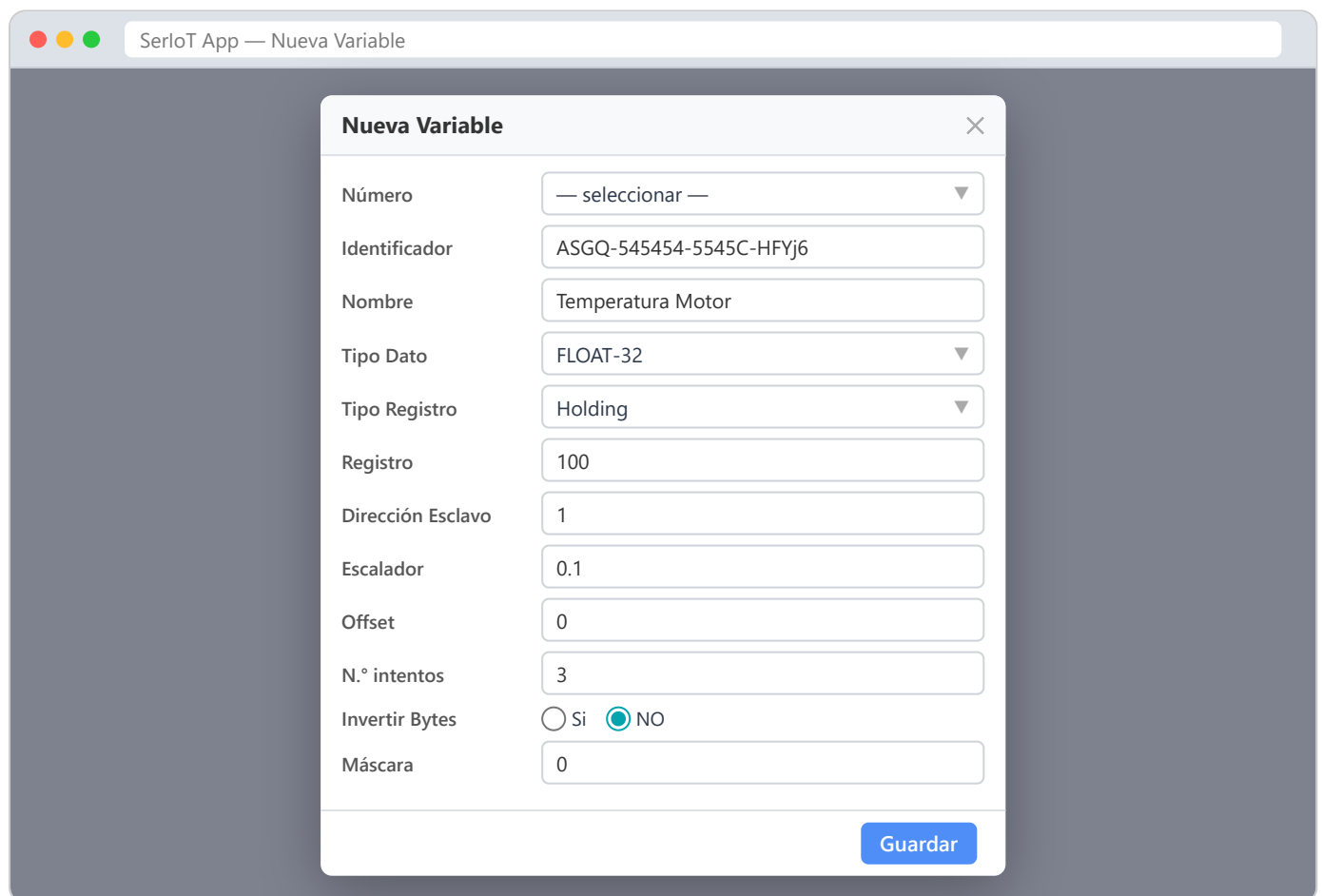
Tipos de datos soportados

Tipo	Descripción	Rango
UINT-16	Entero sin signo 16 bits	0 – 65 535
UINT-32	Entero sin signo 32 bits	0 – 4 294 967 295
UINT-64	Entero sin signo 64 bits	0 – $2^{64}-1$
INT-16	Entero con signo 16 bits	-32 768 – 32 767
INT-32	Entero con signo 32 bits	$\pm 2\,147\,483\,647$
INT-64	Entero con signo 64 bits	$\pm 9.2 \times 10^{18}$
FLOAT-32	Flotante simple precisión IEEE 754	~7 cifras sig.
FLOAT-64	Flotante doble precisión IEEE 754	~15 cifras sig.
BIT-MASK	Máscara de bits sobre un registro	Configurable
MOD10	Valor en módulo 10	Específico de protocolo
MODICON-FLOAT	Flotante Modicon con word swap	Compatible PLC MODICON

Tipos de registro Modbus

Tipo	Función Modbus	Descripción
Holding	FC 03 — Read Holding Registers	Lectura/escritura de uso general (el más común)
Input	FC 04 — Read Input Registers	Solo lectura, típicamente mediciones de sensores
Write Single	FC 06 — Write Single Register	Escritura directa de un registro de 16 bits

Crear / editar variable — formulario modal



SerIoT App — Nueva Variable

Nueva Variable ✕

Número: — seleccionar —

Identificador: ASGQ-545454-5545C-HFYj6

Nombre: Temperatura Motor

Tipo Dato: FLOAT-32

Tipo Registro: Holding

Registro: 100

Dirección Esclavo: 1

Escalador: 0.1

Offset: 0

N.º intentos: 3




Invertir Bytes: Si NO

Máscara: 0

Guardar

Fig. 6 — Modal de nueva variable completado con datos de un sensor de temperatura.

- 1 Pulsa + **Nueva variable** (esquina superior derecha de la tabla).
- 2 Completa los campos requeridos: Identificador, Nombre, Tipo Dato, Tipo Registro, Registro, Dirección Esclavo, Escalador y N.º intentos.
- 3 Pulsa **Guardar**. La variable aparecerá en la tabla.

 Para **editar**: haz doble clic sobre la celda (edición en línea) o pulsa  para abrir el formulario completo. Para **eliminar**: pulsa  y confirma el diálogo.

Ejemplos prácticos

Ejemplo 1 — Sensor de temperatura (FLOAT-32, Holding)

El sensor entrega un valor crudo de 235 = 23.5 °C. El escalador **0.1** realiza la conversión.

Identificador

Registro

Nombre

Dir. Esclavo

Tipo Dato

Escalador

Tipo Registro

Offset

Fórmula: $\text{valor_real} = \text{crudo} \times \text{escalador} + \text{offset} \rightarrow 235 \times 0.1 + 0 = \mathbf{23.5\text{ °C}}$

Ejemplo 2 — Sensor de presión (UINT-16, Input)

Sensor de presión en psi en esclavo dirección 2. El valor crudo ya está en la unidad deseada (escalador = 1).

Identificador	Registro
<input type="text" value="presion_bomba"/>	<input type="text" value="200"/>
Nombre	Dir. Esclavo
<input type="text" value="Presión Bomba"/>	<input type="text" value="2"/>
Tipo Dato	Escalador
<input type="text" value="UINT-16"/>	<input type="text" value="1"/>
Tipo Registro	N.º intentos
<input type="text" value="Input (FC04)"/>	<input type="text" value="3"/>

Ejemplo 3 — Estado ON/OFF con máscara de bit (BIT-MASK)

Registro de estado de 16 bits donde el **bit 0** indica si el motor está encendido. Máscara **1** extrae solo ese bit.

Identificador	Registro
<input type="text" value="estado_motor"/>	<input type="text" value="300"/>
Nombre	Dir. Esclavo
<input type="text" value="Estado ON/OFF Motor"/>	<input type="text" value="1"/>
Tipo Dato	Escalador
<input type="text" value="BIT-MASK"/>	<input type="text" value="1"/>
Tipo Registro	Máscara
<input type="text" value="Holding (FC03)"/>	<input type="text" value="1"/>

Resultado: registro = `0x0001` → estado = **1 (ON)** / registro = `0x0000` → estado = **0 (OFF)**


11. Importar / Exportar configuración

Exportar

- 1 Ve a **Exportar** en el menú lateral.
- 2 Selecciona las configuraciones a incluir (serie, red, variables...).
- 3 Pulsa **Exportar** y elige la ruta de guardado del archivo.

Importar

- 1 Ve a **Importar** en el menú lateral.
- 2 Pulsa **Seleccionar archivo** y abre un fichero previamente exportado.
- 3 Revisa el resumen de cambios y pulsa **Aplicar**.

 Importar **sobrescribe** los valores actuales del dispositivo. Exporta primero si quieres conservar la configuración como copia de seguridad.

12. API interna

El backend expone una API REST en `http://localhost:8080/seriot/` — útil para diagnóstico o integraciones avanzadas.

Endpoint	Método	Descripción
<code>/seriot/check-health</code>	GET	Comprueba que el backend está activo
<code>/seriot/device</code>	GET	Información del dispositivo conectado
<code>/seriot/serial-config</code>	GET / POST	Leer / guardar configuración serie
<code>/seriot/network-config</code>	GET / POST	Leer / guardar configuración de red
<code>/seriot/time</code>	GET / POST	Leer / sincronizar hora
<code>/seriot/variables</code>	GET / POST / DELETE	CRUD completo de variables Modbus
<code>/actuador/shutdown</code>	POST	Apagado controlado del backend

GET `/seriot/device`

```
{
  "model":      "SerIoT-WiFi-001",
  "version":    "1.4.2",
  "firmware_date": "2025-11-20",
  "port":      "COM3",
  "type":      "WIFI"
}
```

GET `/seriot/variables` → respuesta

```
[
  {
    "number": 1, "id": "temp_motor", "name": "Temperatura Motor",
    "data_type": "FLOAT-32", "register_type": "HOLDING",
    "register": 100, "slave_address": 1,
    "scaler": 0.1, "offset": 0, "number_of_tries": 3,
    "reverse_byte": 0, "bit_mask": 0
  }
]
```

```
  },  
  {  
    "number": 2, "id": "presion_bomba", "name": "Presión Bomba",  
    "data_type": "UINT-16", "register_type": "INPUT",  
    "register": 200, "slave_address": 2,  
    "scaler": 1, "offset": 0, "number_of_tries": 3,  
    "reverse_byte": 0, "bit_mask": 0  
  }  
]
```

⚠ 13. Mensajes de error

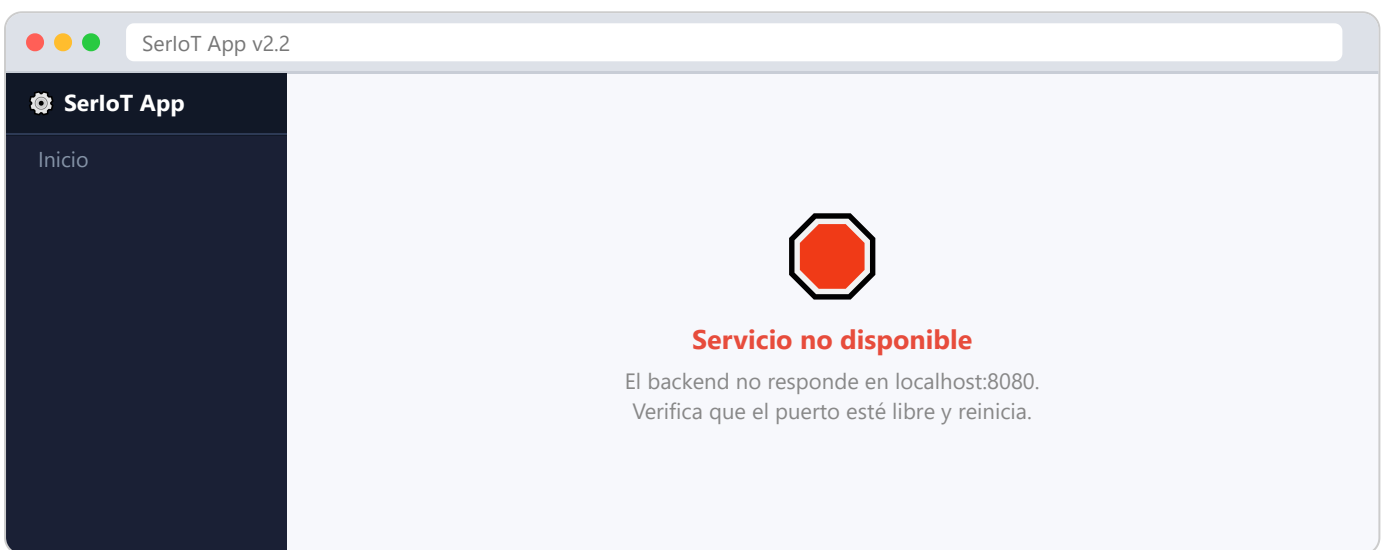


Fig. 7 — Pantalla de error cuando el backend Java no puede arrancar.

Mensaje	Causa probable	Solución
Device Not Found	Ningún dispositivo SerIoT en puertos COM	Conecta el dispositivo y reinicia
Service Down	Puerto 8080 ocupado u otro error de backend	Libera el puerto 8080 y reinicia
5 reintentos fallidos	El backend tardó demasiado en responder	Espera unos segundos y recarga
Campo requerido	Formulario guardado con campos vacíos	Completa los campos marcados en rojo
Campo debe ser un número	Texto en campo numérico	Ingresa solo dígitos
Límite de variables	Máximo de variables del dispositivo alcanzado	Elimina variables no utilizadas

14. Atajos de teclado


Atajo	Acción
F5	Recargar la vista actual
Ctrl + R	Recargar la ventana
Ctrl + Shift + I	Abrir DevTools (solo modo desarrollo)
Alt + F4	Cerrar la aplicación
Doble clic en celda	Activar edición en línea en la tabla de variables
Enter / Tab	Confirmar edición en línea
Esc	Cancelar edición o cerrar modal

15. Glosario

Término	Definición
Modbus RTU	Protocolo serie estándar para automatización industrial, basado en tramas binarias
Holding Register	Zona de memoria lectura/escritura en un esclavo Modbus (función 03)
Input Register	Zona de memoria de solo lectura en un esclavo Modbus (función 04)
Esclavo / Slave	Dispositivo que responde peticiones Modbus; dirección única 1–247
Escalador (Scaler)	Factor multiplicador para convertir valor crudo a unidades reales
Offset	Valor sumado tras el escalador: $valor_real = crudo \times scaler + offset$
Byte Reversal	Intercambio de bytes para compatibilidad Big Endian / Little Endian
BIT-MASK	Operación AND para aislar un bit concreto dentro de un registro de 16 bits
Data Logger	Modo del dispositivo para registro local de datos sin conectividad de red
APN	Access Point Name — punto de acceso del operador celular para datos
DHCP	Protocolo de configuración dinámica de host — asigna IP automáticamente

16. Descripción del dispositivo SerIoT

El **SerIoT** es un Gateway para uso industrial que permite conectar a internet cualquier equipo, máquina o dispositivo con comunicación serial **Modbus RTU**, para efectuar registro de datos en plataformas IoT. Opera como **Maestro Modbus** y de manera autónoma (sin Middleware) lee registros de esclavos a intervalos configurables, les adiciona marca de tiempo y los almacena hasta poder transmitirlos por red celular o WiFi.

-  No requiere APN privadas, VPN ni IP fijas. Permite desplegar sistemas de monitoreo con poca infraestructura en poco tiempo y a bajo costo.

Principales características

Conectividad celular

Tecnología 2G, 2G/3G/4G o 2G/Cat-M1 según modelo. Protocolos TCP/IP, UDP, HTTP, HTTPS.

Modbus RTU Maestro

Soporta hasta 31 esclavos Modbus. Funciones 03 (Holding) y 04 (Input Register).

Memoria flash 16 MB

Almacena datos que no puede transmitir por inestabilidad de red, para reenvío posterior.

Hasta 70 variables

13 tipos de datos soportados (UINT, INT, FLOAT, BIT_MASK, MOD10, MODICON...).

Alimentación flexible

90–230 VAC / 50–60 Hz o 9–30 VDC. Montaje sobre riel DIN.

Aislamiento óptico

Puerto RS485 aislado ópticamente para protección de instalaciones industriales.

Características técnicas

Parámetro	Valor / Descripción
Tecnología celular	2G · 2G/3G/4G · 2G/Cat-M1 (según modelo)
Protocolos de red	TCP/IP, UDP, HTTP, HTTPS
SIM	1 × SIM Card 15 × 25 mm
Memoria interna	16 MB Flash
Puerto serial	1 × RS485 aislado ópticamente
Baud Rate RS485	2400 – 115 200 bps
Protocolo serial	Modbus RTU (Maestro)
Puerto USB	1 × USB tipo B (configuración / firmware)
Alimentación	90–230 VAC / 50–60 Hz ó 9–30 VDC
Dimensiones	88 × 72 × 59 mm
Montaje	Riel DIN
Variables máximas	70 variables
Esclavos Modbus máx.	31 dispositivos

17. Instalación y conexiones

El montaje del SerIoT se efectúa sobre **riel DIN**. Se requieren dos conexiones principales:

Alimentación

Bornes **1 y 2** marcados como *90–230 VAC*. También acepta 9–30 VDC en los mismos terminales.

RS485

Bornes **22 (B), 24 (A), 21 (GND)**. Conectar en paralelo el SerIoT y todos los esclavos Modbus.

Conexión RS485 — Reglas de cableado

- 1 Conectar el SerIoT y todos los esclavos Modbus **en paralelo** sobre los cables A, B y GND.
- 2 Para redes largas o con muchos dispositivos, añadir una resistencia de **120 Ω** en los extremos del bus para garantizar acople de impedancias.
- 3 La resistencia de terminal está **incorporada** en el SerIoT; se activa haciendo un puente entre los terminales **23 (RT) y 24 (A)**.
- 4 Si solo se conecta el SerIoT y un esclavo en una red corta, **no es necesaria** la resistencia terminal.



No usar conexiones en estrella ni derivaciones intermedias. Esto puede generar corrupción de datos en la red RS485. El cableado debe ser lineal (daisy-chain).


18. Indicadores LED

El SerIoT dispone de tres LEDs en su parte frontal para indicar el estado de funcionamiento.

LED	Estado	Significado
Rojo	Apagado	Equipo sin energía — verificar fuente de alimentación
	Encendido	Equipo energizado correctamente
Amarillo (ST)	500 ms ON / 500 ms OFF	Modo de funcionamiento normal
	1000 ms ON / 1000 ms OFF	Modo de configuración (AT activo)
Verde (LK) — 2G	500 ms OFF / 200 ms ON	Sin enganche a red 2G — verificar SIM y antena
	3000 ms OFF / 200 ms ON	Enganchado a red 2G — enviará datos si está correctamente configurado
Verde (LK) — 3G	3000 ms OFF / 200 ms ON	Sin enganche a red 3G — verificar SIM y antena
	3000 ms ON / 200 ms OFF	Enganchado a red 3G — enviará datos si está correctamente configurado

19. Configuración por comandos AT

El SerIoT puede configurarse directamente mediante comandos AT a través del puerto USB. Para ingresar al modo AT enviar `AT+CONFIG` (el módulo responde `OK`). Para salir enviar `AT+ESC`.

 Al ingresar al modo AT, el LED amarillo (ST) cambia a **1000 ms ON / 1000 ms OFF**. Usar cualquier terminal serie (ej. PuTTY) a la velocidad configurada.

AT+SERIAL — Puerto serie

Acción	Comando
Consultar	<code>AT+SERIAL?</code>
Configurar	<code>AT+SERIAL=<bd,parity></code>
Parámetros	bd: 1200 · 2400 · 4800 · 9600 · 19200 · 38400 · 56000 · 57600 · 115200 parity: NONE · EVEN · ODD

Ejemplo AT+SERIAL

```
AT+SERIAL?
+SERIAL: 9600,NONE

AT+SERIAL=115200,EVEN
OK

AT+SERIAL?
+SERIAL: 115200,EVEN
```

AT+CLOCK — Fecha y hora

Acción	Comando
Consultar	<code>AT+CLOCK?</code>
Configurar	<code>AT+CLOCK=<yymmddhhmmss></code>
Formato	YYMMDDHHMMSS — año/mes/día/hora/minuto/segundo

Ejemplo AT+CLOCK

```
AT+CLOCK?
+CLOCK: 260505143023
```

```
AT+CLOCK=260505153000
```

```
OK
```

AT+SERVER — Servidor IoT

Acción	Comando
Consultar	<code>AT+SERVER?</code>
Configurar	<code>AT+SERVER=<"ip",port></code>
Parámetros	ip: dirección IP o URL port: puerto del servidor

Ejemplo AT+SERVER

```
AT+SERVER?
```

```
+SERVER: "190.249.164.243",5000
```

```
AT+SERVER="iot.miempresa.com",8883
```

```
OK
```

AT+TOKEN — Token de seguridad

Acción	Comando
Consultar	<code>AT+TOKEN?</code>
Configurar	<code>AT+TOKEN=<token></code>
Nota	Si el servidor no requiere token, ingresar al menos un carácter alfanumérico. Campo vacío impide el envío de datos.

Comandos APN (red celular)

Comando	Descripción	Ejemplo
<code>AT+APN= <apn></code>	Nombre del punto de acceso del operador	<code>AT+APN=internet.movistar.com.co</code>
<code>AT+USER= <user></code>	Usuario del APN	<code>AT+USER=movistar</code>
<code>AT+PASS= <pass></code>	Contraseña del APN	<code>AT+PASS=movistar</code>
<code>AT+STIME= <s></code>	Tiempo de muestreo en segundos entre peticiones de variables	<code>AT+STIME=60</code>

Comandos de variables por AT

Comando	Descripción
<code>AT+CLEANVAR<n></code>	Borra de memoria la variable número <code>n</code>
<code>AT+VARID<n>? / AT+VARID<n>= <id></code>	Consultar / configurar el ID alfanumérico de la variable <code>n</code>
<code>AT+VARNAME<n>? / AT+VARNAME<n>= <name></code>	Consultar / configurar el nombre descriptivo de la variable <code>n</code>
<code>AT+VARVAL<n>? / AT+VARVAL<n>=<...></code>	Consultar / configurar todos los parámetros de la variable (ver formato abajo)

El formato completo de `AT+VARVAL` es:

Formato `AT+VARVAL`

```
AT+VARVAL<n>=<aslv,reg,rt,dt,ret,sw,bm,sc,off>
```

`aslv` : Dirección del esclavo Modbus

`reg` : Dirección del registro

rt : Tipo de registro (3=Holding, 4=Input)
dt : Tipo de dato (0-13, ver tabla tipos de datos)
ret : Número de intentos
sw : Invertir bytes (0=Big Endian, 1=Little Endian)
bm : Máscara de bits
sc : Escalador
off : Offset

Ejemplo AT+VARVAL


```
AT+VARVAL1?  
+VARVAL1: 1,100,3,6,3,0,0,0.1,0.0  
  
AT+VARVAL1=1,100,3,6,3,0,0,0.1,0.0  
OK
```

Comandos Gestion Memoria SD SERLOG- (Si aplica)

Comando	Descripción
<code>AT+SDUMP?</code>	Envia el contenido del ultimo archivo CSV guardado en la SD por el puerto USB
<code>AT+SDUMP= <name></code>	Envia el contenido del archivo descrito por <code>name</code> guardado en la memoria SD por el el puerto USB
<code>AT+SDLS?</code>	Envia la lista de arvhivos de la SD por el puerto USB
<code>AT+SDEL= <name></code>	Borra el archivo de la memoria SD descrito por <code>name</code>
<code>AT+SDFMT= <YES></code>	Formatea completamente la memoria SD. Confirmacion explicita <code>YES</code>
<code>AT+VCOM?</code>	Consulta el estado del flujo en el puerto serial. Si sta habilitado todo lo que se guarda en la SD es enviado al puerto USB
<code>AT+VCOM= <estado></code>	Establece el estado en el flujo en el puerto USB. <code>estado</code> 0 - Deshabilitado / 1 - Habilitado

20. Configuración remota por SMS

A partir del **firmware v2.4**, el SerloT con conectividad celular soporta configuración remota mediante mensajes de texto SMS. Permite modificar cualquier parámetro configurable desde la aplicación.

 La comunicación es **unidireccional**: el SerloT solo recibe SMS, no genera respuesta. Para verificar que el parámetro cambió, observar el comportamiento del equipo en tiempo real (p. ej. confirmar nuevo tiempo de muestreo viendo los envíos de datos).

Procedimiento

- 1 Conocer el número de celular asociado a la SIM CARD instalada en el SerloT.
- 2 Enviar el comando AT directamente como cuerpo del SMS — **no es necesario** enviar `AT+CONFIG` antes ni `AT+ESC` después.
- 3 Esperar el tiempo de muestreo configurado y verificar que el cambio surtió efecto.

Comandos compatibles con SMS

Comando SMS

Parámetro configurado

`AT+SERIAL=<bd,parity>`

Velocidad y paridad del puerto serie

`AT+CLOCK=<yymmddhhmmss>`

Fecha y hora interna

`AT+SERVER=<"ip",port>`

Servidor IoT (IP y puerto)

`AT+TOKEN=<token>`

Token de autenticación

`AT+APN=<apn>`

APN del operador celular

`AT+USER=<user>`

Usuario del APN

`AT+PASS=<pass>`

Contraseña del APN

`AT+STIME=<s>`

Tiempo de muestreo (segundos)

`AT+CLEANVAR<n>`

Eliminar variable n

`AT+VARID<n>=<id>`

ID de la variable n

`AT+VARNAME<n>=<name>`

Nombre de la variable n

`AT+VARVAL<n>=<aslv,reg,rt,dt,ret,sw,bm,sc,off>`

Parámetros completos de la variable n

Ejemplo — Cambiar tiempo de muestreo a 60 s por SMS

Si la SIM del SerloT tiene el número `+57 322 236 4055`, enviar el siguiente SMS:

Cuerpo del SMS

```
AT+STIME=60
```


El SerloT comenzará a enviar datos cada 60 segundos tras recibir el comando.

21. Firmware y memoria

Actualizar el firmware


La actualización se realiza por el **puerto USB** usando la aplicación **KinetisFlashTool**.

- 1 Descarga e instala **KinetisFlashTool** desde www.gndelectronics.com.
- 2 Abre la aplicación y selecciona **USB + HIB** en el campo *Port Set* (esquina superior izquierda).
- 3 En *Target Address* ingresa el valor hexadecimal `0x0000A000`.
- 4 Clic en **Browse** y selecciona el archivo `.bin` con el nuevo firmware.
- 5 Conecta el SerIoT por USB al PC, **reinícialo** y antes de que pasen 5 segundos clic en **Update**.
- 6 Espera a que la barra de progreso llegue al **100%**. El proceso habrá concluido correctamente.

 Se recomienda actualizar al firmware más reciente disponible para obtener las últimas correcciones y funciones, incluyendo configuración por SMS (disponible desde v2.4).

Memoria flash — comportamiento

El SerIoT posee **16 MB de Flash**. Cuando no puede transmitir datos por inestabilidad de red, los almacena localmente y reintenta el envío cuando la conectividad se restaura, evitando pérdida de datos.

 La única forma de recuperar datos almacenados en memoria es a través de la red celular. **No existe función para extracción local de datos.**

Aspecto	Detalle
Capacidad	16 MB Flash
Autonomía	Depende del número de variables y el tiempo de muestreo. Un tiempo de muestreo pequeño reduce el tiempo de almacenamiento.
Recuperación	Automática vía red celular al restaurar conectividad
Extracción local	No disponible

Borrar la memoria (reset de fábrica)



Exporta la configuración (ver sección 11) antes de borrar la memoria. Al borrar, se pierden también todos los parámetros configurados y el equipo regresa a valores de fábrica.

- 1 Apaga el SerloT.
- 2 Usa la punta de un clip o similar para presionar el **botón de reset**, ubicado en la parte inferior lateral izquierda.
- 3 Manteniendo el botón presionado, **enciende** el SerloT.
- 4 Mantén el botón hasta que el **LED ST se encienda**, luego suéltalo.
- 5 Se borrarán todos los datos en memoria y el equipo volverá a parámetros de fábrica.