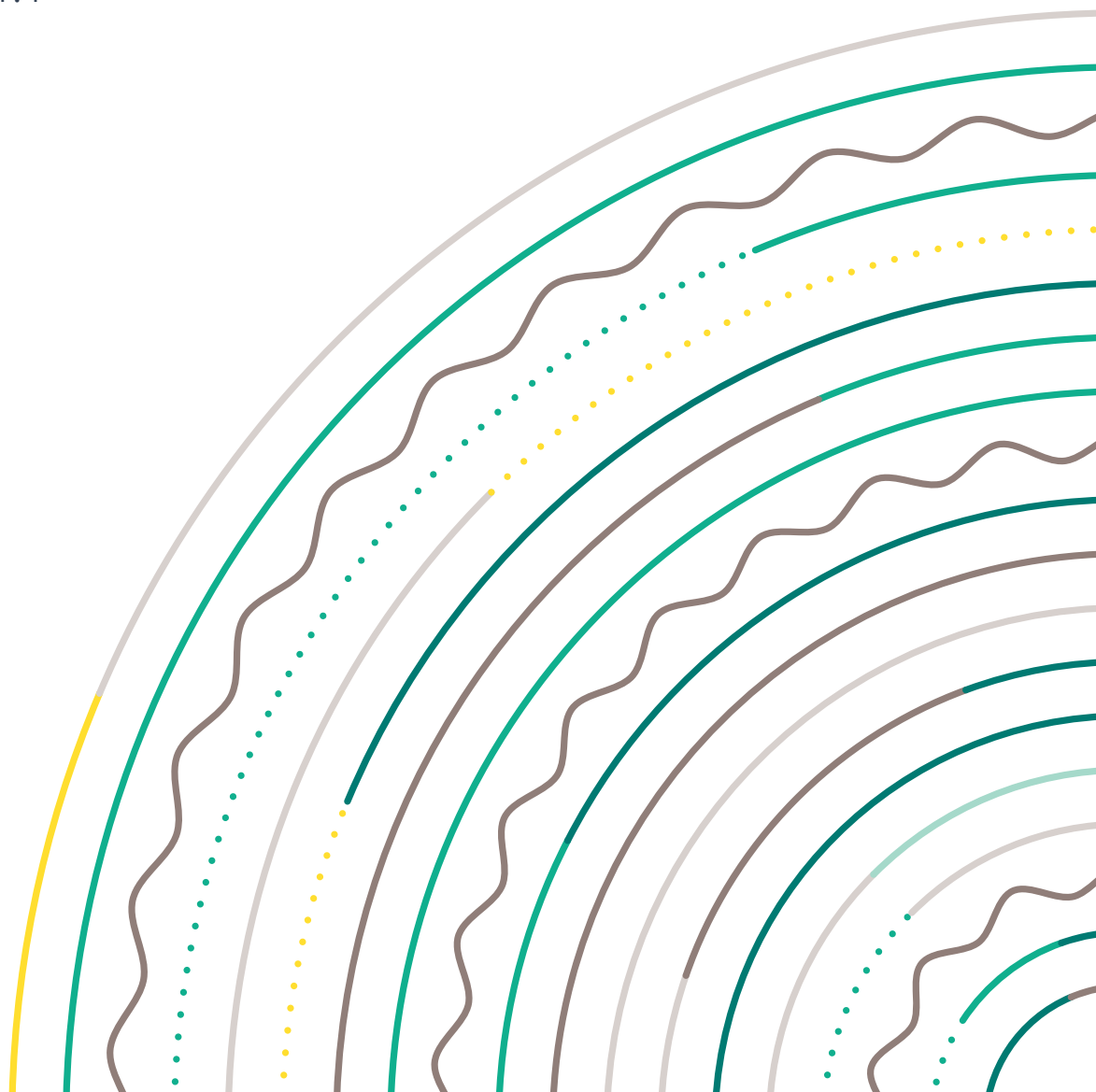


# Lifedomus

## Modbus

31/01/2018

Versión 1.4



# Protocolo Modbus TCP, modo «Maestro»

## Índice

1	Requisitos previos .....	4
2	Conector Modbus.....	4
2.1	Adición de un conector Modbus TCP .....	4
2.2	Configuración de un conector Modbus TCP .....	4
3	Equipo Modbus .....	6
3.1	Propiedades de un equipo .....	6
3.1.1	Configuración del campo «lectura» de una propiedad .....	6
3.1.2	Configuración del campo «escritura» de una propiedad .....	6
3.2	Variables de un equipo .....	7
3.2.1	Gestión de variables .....	7
3.2.2	Configuración de una variable .....	8
3.3	Comandos de un equipo .....	10
4	Creación de la interfaz «Design Studio».....	11



## 1 Requisitos previos

Para poder seguir correctamente esta documentación son necesarios algunos requisitos previos:

- ✓ Adición, configuración y utilización de un conector en «Config Studio»
- ✓ Creación de un equipo genérico Lifedomus
- ✓ Protocolo Modbus
- ✓ Bases sobre las operaciones lógicas binarias para entender y utilizar la «máscara»

## 2 Conector Modbus

La comunicación entre Lifedomus y un bus Modbus necesita la utilización de una pasarela Modbus TCP/IP.

Como todos los demás protocolos, la comunicación se realiza a través de un conector Lifedomus. Actualmente, Lifedomus se comporta como «**maestro**» en el bus. Posteriormente se implementará un modo «esclavo».

### 2.1 Adición de un conector Modbus TCP

La adición de un conector «Modbus TCP/IP» se realiza de igual modo que para cualquier otro conector.

### 2.2 Configuración de un conector Modbus TCP

Los elementos de configuración de un conector Modbus TCP son:

- Dirección IP: dirección IP de la pasarela Modbus TCP
- Puerto IP: puerto de la pasarela Modbus TCP (por defecto: 502)

**Modbus TCP/IP**  
 Protocole : Modbus  
 Catégorie : Interfaces IP  
 Interface : Modbus TCP/IP

⋮
➡
✎
✕

**Label :**

**Label court :**

**Description :**

**Propriétés :**

**Adresse IP**


**Port IP**

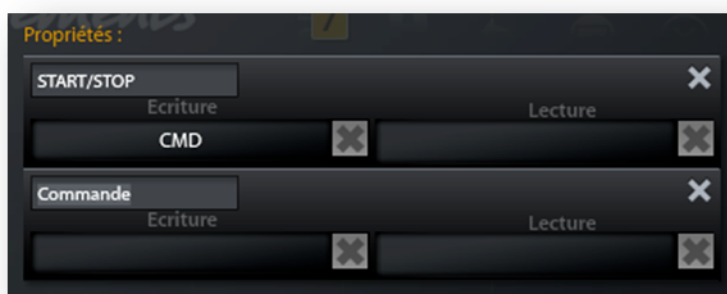
⏪
⏩

## 3 Equipo Modbus

Un conector Modbus funciona exclusivamente con equipos Lifedomus de tipo «**Universal**». Por tanto, hay que agregar un «**Equipo (Genérico)**» y vincularlo al conector Modbus creado anteriormente.

### 3.1 Propiedades de un equipo

Para agregar una propiedad al equipo que se acaba de crear, use el icono  de la parte inferior derecha. Una propiedad permitirá la lectura y escritura de datos en uno o varios registros del esclavo Modbus.



#### 3.1.1 Configuración del campo «lectura» de una propiedad

Al hacer clic en el campo «lectura» de una propiedad, aparecerá una nueva ventana y sus datos de acceso a las diferentes variables creadas. Puede crear nuevas variables.

Seleccionar una **variable** y validar la ventana permite vincular el campo de lectura de la propiedad a esta variable. Se hace accesible en Lifedomus un retorno de estado correspondiente al valor de esta variable.

En la sección [Variables de un equipo](#) de este documento se describen el funcionamiento y los detalles de configuración de una variable.

#### 3.1.2 Configuración del campo «escritura» de una propiedad

Al hacer clic en el campo «Escritura» de una propiedad, aparecerá una nueva ventana y sus datos de acceso a las diferentes variables y comandos creados.

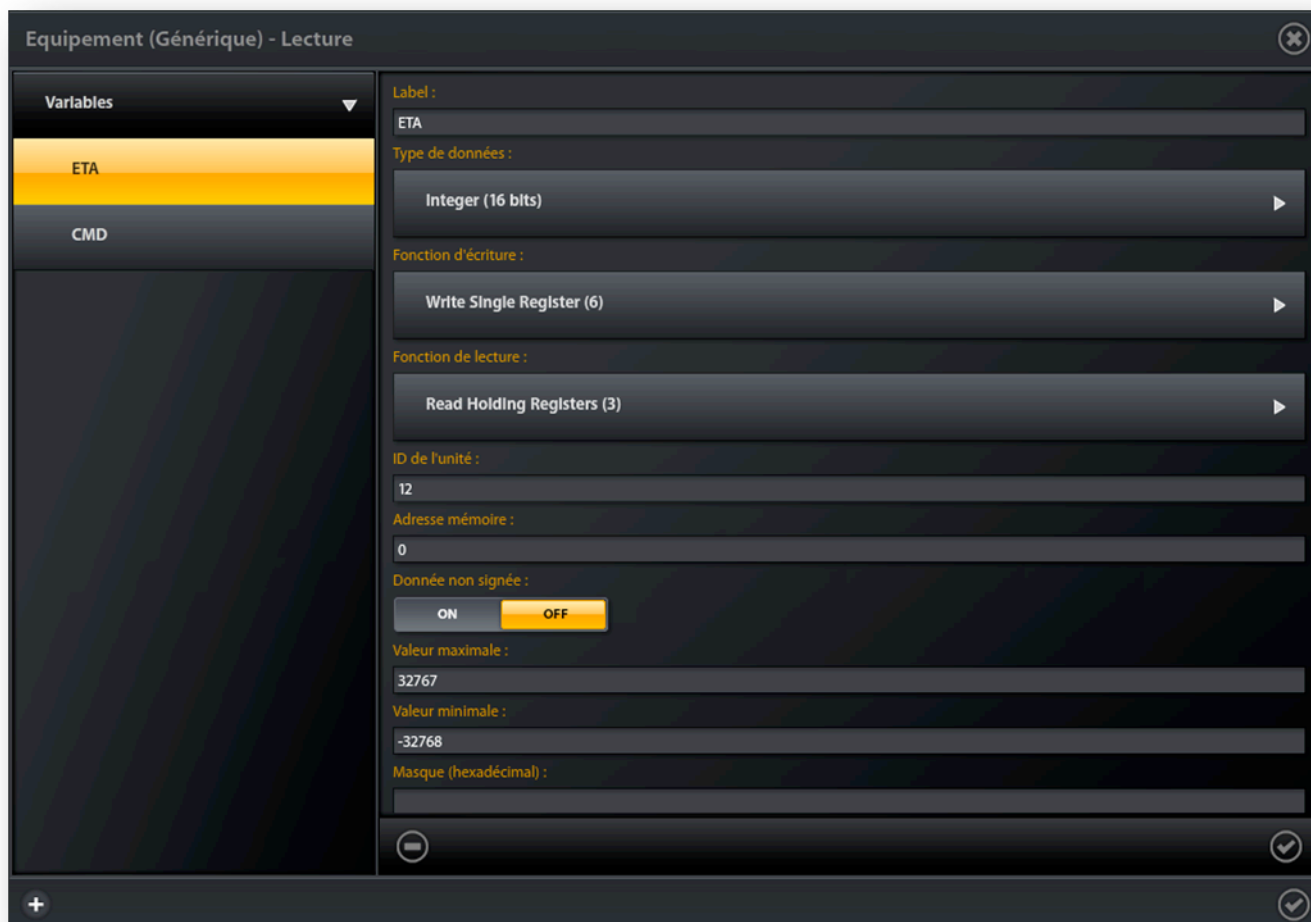
Seleccionar una **variable** y validar la ventana permite vincular el campo de escritura de la propiedad a esta variable. Se hace accesible en Lifedomus una acción que permite la escritura en la variable.

Seleccionar un **comando** y validar la ventana permite vincular el campo de escritura de la propiedad a este comando. Se hace accesible en Lifedomus una acción que permite la ejecución de este comando.

En la sección [Comandos de un equipo](#) de este documento se describen el funcionamiento y los detalles de configuración de un comando.

## 3.2 Variables de un equipo


Una variable Modbus es un dato en Lifedomus que representa una zona de memoria (uno o varios registros) de un esclavo del bus.



The screenshot shows a software interface titled "Equipement (Générique) - Lecture". On the left, there is a sidebar with a "Variables" dropdown menu, a list containing "ETA" (highlighted in orange), and a "CMD" button. The main area on the right is for configuring the selected variable. It includes fields for "Label" (ETA), "Type de données" (Integer (16 bits)), "Fonction d'écriture" (Write Single Register (6)), "Fonction de lecture" (Read Holding Registers (3)), "ID de l'unité" (12), "Adresse mémoire" (0), a "Donnée non signée" toggle (set to OFF), "Valeur maximale" (32767), "Valeur minimale" (-32768), and "Masque (hexadécimal)". At the bottom, there are three buttons: a plus sign (+) for adding, a checkmark in a circle (✓) for saving, and a minus sign in a circle (⊖) for deleting.

La pantalla de gestión de variables siguiente permite agregar, configurar y eliminar variables Modbus. Estas variables creadas son específicas al equipo actualmente en curso de modificación.

### 3.2.1 Gestión de variables

La adición de una variable se realiza seleccionando la lista «Variables» y haciendo clic en  (parte inferior izquierda).

El botón  permite guardar las modificaciones realizadas en la variable y su asociación con la propiedad (en lectura o escritura) seleccionada previamente.

El botón  permite la eliminación de la variable seleccionada.

### 3.2.2 Configuración de una variable

La configuración de una variable se realiza seleccionándola en la lista de variables. Los elementos de configuración permiten definir con precisión la forma en que se leerán e interpretarán las zonas de memoria.

- **Etiqueta:** la etiqueta permite simplemente asignar un título a la variable.
- **Tipo de datos:** el tipo de datos que quiere manejar en Lifedomus: booleano, número entero, separador decimal (IEEE 754) o valor hexadecimal. Los valores de 16 bits o menos se leen o escriben en un solo registro, mientras que los valores de un tipo de 32 bits se leerán o escribirán en dos registros consecutivos.
- La **función de lectura** y la **función de escritura** son las funciones Modbus que se usarán para leer y escribir las zonas de memoria correspondientes a esta variable en el bus. Generalmente, estas funciones se precisan en la documentación del fabricante.
- **ID de unidad** o «UnitID» (1 a 255): este campo es específico al protocolo Modbus TCP y permite identificar un esclavo en el bus.
- **Dirección de memoria** (1 a 65535): la dirección de memoria es la dirección del registro que quiere leer o escribir. Cuando se selecciona un tipo de datos de 32 bits, los registros leídos o escritos son el registro indicado en el campo «dirección de memoria» y el registro siguiente.  
Ejemplo: un entero (32 bits) leído o escrito en la dirección de memoria 2202 se leerá o escribirá de hecho en las direcciones 2202 y 2203.
- **Dato no firmado:** la activación de este campo permite considerar el dato como un dato no firmado, es decir un valor positivo. Cambiar este valor actualiza los campos «Valor mínimo» y «Valor máximo».
- **Valor mínimo y máximo:** estos valores se definen por defecto en función del «**tipo de datos**» seleccionado y de la activación o desactivación de los «**datos firmados**». Ejemplos:
  - Un Booleano para valor 0 (falso) o 1 (verdadero)
  - Un Entero (16 bits) firmado tiene un valor comprendido entre -32768 y 32767
  - Un Entero (16 bits) sin firmar tiene un valor comprendido entre 0 y 65535

Los valores mínimo y máximo definidos por defecto son los límites de los valores leídos o escritos en Lifedomus. Si no es posible superar estos límites, es muy posible restringirlos.

- **Máscara (hexadecimal):** este campo permite asignar una máscara al registro o registros. Se debe definir en formato hexadecimal. Ejemplo: FFF7, 0xFFF7 o 16#FFF7
  - **En caso de lectura:** para determinar el valor final disponible en Lifedomus, se realiza un «Y» lógico entre la máscara y el valor leído en el bus.
  - **En caso de escritura:** solo los bits que presentan un «1» en la máscara se cambiarán en el(los) registro(s) destino.

El uso de la máscara añade una gran potencia a la manipulación de datos.

La sección [Ejemplos y particularidades de la máscara](#) de este documento incluye ejemplos prácticos.

- **Invertir bytes:** generalmente, en la documentación del fabricante se precisa si hay que utilizar este parámetro. Permite invertir el byte alto y el byte bajo durante la lectura y la escritura del registro. Un registro tiene 16 bits, o 2 bytes. Si un registro tiene por valor 0x7F00, invertir los bytes permite leer: 0x007F.
- **Invertir palabras:** una palabra tiene 16 bits (la dimensión de un registro). La inversión de palabras es útil durante la lectura y la escritura de varios registros, cuando el tipo de datos seleccionado es Entero (32 bits) o Flotante (32 bits).  
Ejemplo: los registros n° 2202 y n° 2203 se usan para leer o escribir un entero de 32 bits. Sin inversión de palabras, el registro n°2202 contiene los bits de mayor peso, así como el bit de signo (si el valor está firmado) mientras que el registro n°2203 contiene los bits de menor peso. Con la inversión de palabras, el registro n°2202 contiene los bits de menor peso y el registro n°2203 los de mayor peso y el bit de signo.
- **Coeficiente:** este campo es un coeficiente multiplicador aplicado al valor leído en el bus (dividido durante la escritura).  
 Lectura: «Valor Lifedomus» = «Valor bus» x «Coeficiente»  
 Escritura: «Valor bus» = «Valor Lifedomus»/«Coeficiente»
- **Unidad:** este campo facultativo permite seleccionar la unidad del valor en Lifedomus.



## Ejemplos y particularidades de la máscara

La máscara permite tener en cuenta solo ciertos bits del registro en la lectura y la escritura del bus. Se puede usar para datos de tipo entero (16 o 32 bits), hexadecimal o booleano.

Los valores de los ejemplos siguientes se presentan en formato hexadecimal y binario para facilitar la comprensión del uso de la máscara durante la lectura y la escritura en el bus.

### 3.2.2.1 Uso de la máscara con el tipo «entero» o «hexadecimal» de Lifedomus

#### Ejemplo:

- El registro #3303 contiene el siguiente valor: **0x12FF** (**0001 0010 1111 1111**).
- La máscara es: **0x007F** (**0000 0000 0111 1111**).
- Una lectura de la variable dará para el valor de retorno de estado **«máscara» Y «valor en el bus»**:  
**0x007F** (**0000 0000 0111 1111**) Y **0x12FF** (**0001 0010 1111 1111**) = **0x007F** (**0000 0000 0111 1111**)
- La escritura del valor **0x0000** (**0000 0000 0000 0000**) en la variable generará la escritura del valor:  
**0x1280** (**0001 0010 1000 0000**) en el bus:
- Una nueva lectura de la variable dará para el valor de retorno de estado:  
**0x007F** (**0000 0000 0111 1111**) Y **0x1280** (**0001 0010 1000 0000**) = **0x0000** (**0000 0000 0000 0000**)
- La escritura del valor **0x0F22** (**0000 1111 0010 0010**) en la variable generará la escritura de:  
**0x12A2** (**0001 0010 1010 0010**) en el bus.

### 3.2.2.2 Uso de la máscara con el tipo «booleano» de Lifedomus

La utilización de la máscara con un valor de tipo booleano puede ser frecuente en Modbus. De hecho, varios valores booleanos, «VERDADERO/FALSO», «Activado/Desactivado»,... se suelen registrar en un único registro. Cada bit representa un valor booleano («VERDADERO» si el o los bits en cuestión están en «1», «FALSO» en caso contrario).

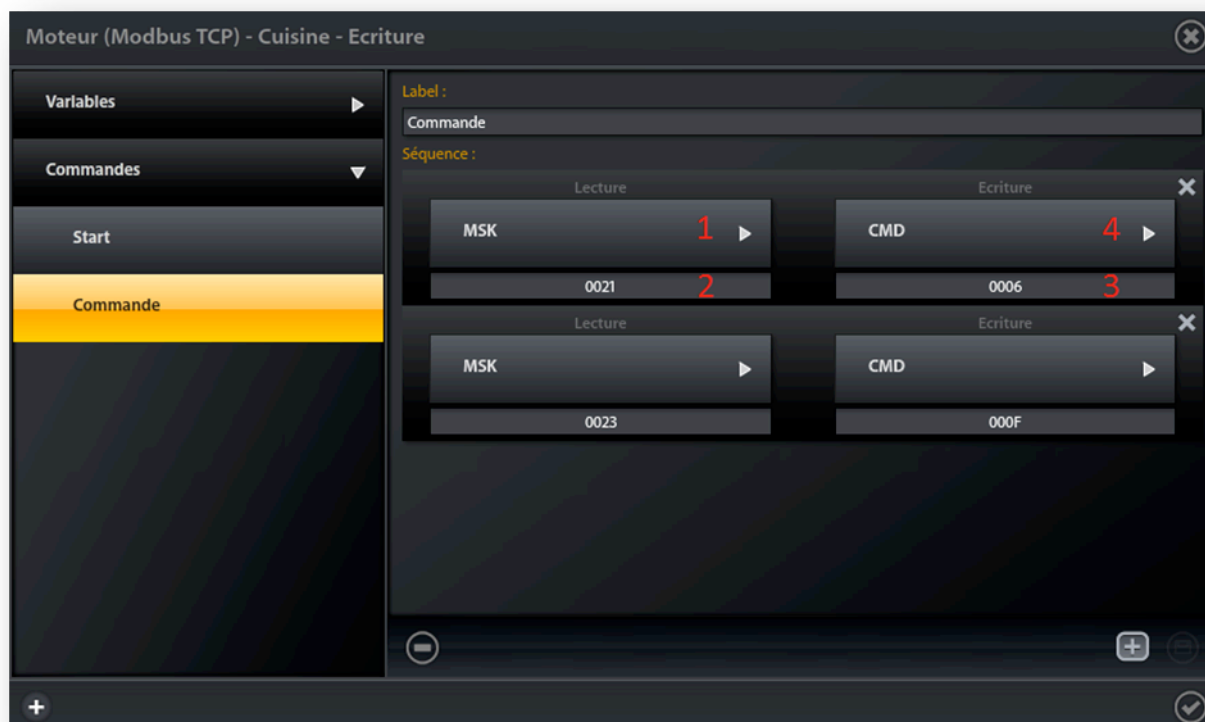
#### Ejemplo:

- El registro #3303 contiene el siguiente valor: **0x12FF** (**0001 0010 1111 1111**).
- La máscara es: **0x0080** (**0000 0000 1000 0000**)
- **0x0080** (**0000 0000 1000 0000**) Y **0x12FF** (**0001 0010 1111 1111**) = **0x0080** (**0000 0000 1000 0000**)  
 Al ser el resultado igual a la máscara, una lectura de la variable dará el valor de retorno de estado:  
**VERDADERO**
- La escritura del valor **FALSO** escribirá en el bus: **«inverso de la máscara» Y «valor del bus»**  
**0x0F22** (**1111 1111 0111 1111**) Y **0x12FF** (**0001 0010 1111 1111**) = **0x127F** (**0001 0010 0111 1111**)
- **0x0080** (**0000 0000 1000 0000**) Y **0x127F** (**0001 0010 0111 1111**) = **0x0080** (**0000 0000 0000 0000**)  
 Al ser el resultado diferente de la máscara, una lectura de la variable dará para el valor: **FALSO**
- La escritura del valor **VERDADERO** escribirá en el bus: **«máscara» Y «valor en el bus»**  
**0x0080** (**0000 0000 1000 0000**) Y **0x127F** (**0001 0010 0111 1111**) = **0x12FF** (**0001 0010 1111 1111**)

### 3.3 Comandos de un equipo

La pantalla de gestión de comandos permite agregar, configurar y eliminar comandos Modbus.

Como las variables, estos comandos creados son específicos al equipo actualmente en curso de modificación.



Cada elemento de la secuencia de un comando se ejecuta en el orden introducido y funciona del modo siguiente: si la variable de la izquierda (1) es igual al valor indicado a la izquierda (2), el valor indicado a la derecha (3) se escribe en la variable de la derecha (4). Cabe señalar que si el campo de lectura (2) queda vacío, se validará la condición y se ejecutará la escritura.

Ejemplo: estos comandos se usan en algunos casos como el del arranque de un motor: si el motor está desconectado, se conecta, si está parado, se pone en marcha y se define su velocidad.

## 4 Creación de la interfaz «Design Studio»

Los equipos Modbus que están en Lifiedomus de equipos genéricos, no tienen widget por defecto en Design Studio. Por tanto es necesario usar el modo edición para construir la interfaz de supervisión.

*Consulte la documentación relativa al «What I See» y al «What I Do» de «Design Studio».*