

## CODESYS – Nivel 2

### Descripción del curso

El Curso de CoDeSys Avanzado está orientado a profesionales que desean especializarse en el desarrollo de aplicaciones de automatización utilizando el entorno CoDeSys, integrando programación en Texto Estructurado (ST), comunicación industrial entre PLC y diseño de interfaces HMI.

El curso se desarrolla bajo una metodología práctica basada en experiencia real en automatización industrial, permitiendo al participante implementar soluciones completas que integran lógica de control, redes industriales y visualización, aplicadas a sistemas industriales modernos



### Contenido del curso

#### DÍA 1 – Fundamentos de CoDeSys y Programación en ST

##### 1. Tipos de Datos y Variables

- Declaración de variables y buenas prácticas.
- Tipos de datos estándar (BOOL, INT, REAL, STRING)
- Tipos definidos por el usuario: STRUCT y ENUM

##### 2. Programación en Texto Estructurado (ST) y Algoritmia

- Sintaxis avanzada en ST: Sentencias IF, CASE, FOR, WHILE.
- Operaciones matemáticas y manipulación de strings.
- Creación de Funciones (FUN) y Bloques de Función (FB) personalizados.
- Concepto de encapsulamiento.

##### 3. Taller Práctico 1

#### Bloque de función para control de penetración – perforación

- Adquisición y escalamiento de señales analógicas (presión, torque, RPM).
- Generación de alarmas: sobrecarga, baja eficiencia, sensor fuera de rango.
- Uso de STRUCT para agrupación de variables y alarmas.

#### DÍA 2 – Control de Flujo y Estructuras Avanzadas

##### 4. Estructuras de control y datos

- Programación Orientada a Objetos (POO) en PLC
- Introducción a la POO industrial: Interfaces, Métodos y Propiedades.
- Herencia y Extensión de bloques.
- Gestión de Punteros (POINTER TO) y Referencias.
- Estructuras de datos complejas (STRUCT).

## 5. Taller Práctico 2

### Librería de subsistemas de perforadora. Uso de marcas, registros y palabras

- Diseño de clase base para subsistemas (rotación, avance, fluidos).
- Implementación de herencia en bloques de función (FB).
- Creación de métodos y propiedades para control de estados.
- Gestión de interlocks entre subsistemas.

## 6. Secuenciadores y Gestión de Errores

- Programación de máquinas de estado con SFC (Sequential Function Chart). Pasos, transiciones y acciones.
- Configuración avanzada de Alarm Configuration. Históricos de alarmas, clases de alarmas y banners de visualización.

## 7. Taller Práctico 3

### Ciclo automático de perforación.

- Diseño de máquina de estados para ciclo de perforación.
- Secuencia: rotación → avance → inyección → perforación → retorno.
- Integración de condiciones de seguridad y parada de emergencia.
- Configuración de alarmas y fallos operativos.
- Simulación del ciclo completo con diferentes escenarios.

## DÍA 3 & 4 – Motion y/o Desarrollo de interfaces HMI en Codesys

### 8. Movimiento y Control de Ejes (SoftMotion)

- Introducción a CODESYS SoftMotion. Configuración de ejes virtuales y reales.
- Movimientos básicos: MC\_Power, MC\_MoveAbsolute, MC\_MoveVelocity. Sincronización básica.

## 9. Taller Práctico 4

### Control de posicionamiento del brazo de perforación.

- Configuración de eje virtual en CODESYS SoftMotion.

- Movimientos básicos: posicionamiento absoluto y manual.
- Implementación de límites de seguridad y zonas de trabajo.
- Desarrollo de interfaz HMI (modo manual/automático, indicadores).
- Pruebas de posicionamiento y visualización en tiempo real.

## 10. WebVisu

- Creación de interfaces gráficas
- Botones, indicadores, tendencias
- Gestión de alarmas (Alarm Configuration)

## DÍA 5 – Comunicaciones e Industria 4.0

### 11. Fundamentos de Redes Industriales

- Configuración de redes industriales: Modbus TCP/IP en CoDeSys
- Escaneo de dispositivos en red.
- Comunicación de alto nivel: Configuración de Servidor OPC UA para intercambio de datos con SCADA/ERP y envío de datos vía MQTT.

## 12. Taller Práctico 5

### Proyecto Integrador: Sistema automatizado de perforación con monitoreo remoto.

- Integración de subsistemas (rotación, avance, fluidos, posicionamiento).
- Control del ciclo completo mediante SFC.
- Procesamiento de datos en ST (ROP, profundidad, eficiencia).
- Publicación de variables vía OPC UA para monitoreo externo.
- Visualización en HMI: estado, alarmas y producción.
- Pruebas funcionales del sistema completo (operación normal y fallos).