

Trabajo Fin de Máster Ingeniería de Tecnologías Industriales

Control y monitorización de una EDAR usando Codesys y Wonderware

Autor: Álvaro Gómez Díaz

Tutor: Francisco Salas Gómez

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla



Trabajo Fin de Máster
Ingeniería de Tecnologías Industriales

Control y monitorización de una EDAR usando Codesys y Wonderware

Autor:

Álvaro Gómez Díaz

Tutor:

Francisco Salas Gómez

Profesor Titular

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Agradecimientos

Cuando miro atrás y veo todo el camino que he recorrido me llena de orgullo saber lo que he podido hacer, estoy haciendo y muy probablemente seré capaz de hacer. Todo esto al fin y al cabo no es cosa solo de mi trabajo y esfuerzo, toda las personas que me han acompañado y apoyado han hecho que este camino haya sido mucho más fácil de recorrer. Es por eso que este proyecto se lo dedico a ellos.

Álvaro Gómez Díaz

Sevilla, 2023

Resumen

Este proyecto se ha basado en el proceso de un sistema de una estación de aguas residuales, centrandose sobre todo en la parte primaria y secundaria de dicho proceso. Se ha intentado hacer un modelo lo más realista posible utilizando referencias desde para la implementación de sensores hasta para las ecuaciones de simulación en el desarrollo de algunos parametros tales como la evolución de bacterias. Se ha utilizado un mismo programa como controlador y simulador del proceso (Codesys) junto como un programa para hacer de HMI (AVEVA System Platform). Dichos programas han tenido poca interacción en conjunto mediante comunicación via OPC UA por lo que este proyecto puede ser considerado innovador al respecto de la comunicación entre ambos programas.

Abstract

This project has been based on the process of a wastewater station system, focusing mainly on the primary and secondary part of said process. An attempt has been made to make a model as realistic as possible using references from the implementation of sensors to the simulation equations in the development of some parameters such as the evolution of bacteria. The same program has been used as controller and process simulator (Codesys) together with a program to make HMI (AVEVA System Platform). These programs have had little interaction together through communication via OPC UA, so this project can be considered innovative in terms of communication between both programs.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1. Planteamiento	1
1.1. Objetivos	1
2. Sistema a controlar	3
2.1. Descripción detallada	4
3. Control del sistema	9
3.1. Modo automático	9
3.2. Modo Manual	18
3.3. Reseteo	18
4. Pantalla de explotación HMI	19
4.1. Representación en Intouch	19
5. Conclusiones	29
Apéndice A. Guía instalación	31
A.1. Instalación de Wonderware	31
A.2. Configuración para la licencia	32
A.3. Posibles problemas	33
Apéndice B. Comunicación	35
B.1. Pasos a seguir	35
B.2. Posibles problemas	39
Apéndice C. Planteamiento	41
C.1. Código simulación del sistema	41
C.2. Modo Reseteo código	45
C.3. Modo Manual código	46
<i>Índice de Figuras</i>	47
<i>Índice de Códigos</i>	49
<i>Bibliografía</i>	51

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1. Planteamiento	1
1.1. Objetivos	1
2. Sistema a controlar	3
2.1. Descripción detallada	4
2.1.1. Tanques decantadores	4
2.1.2. Tanque reactor	5
2.1.3. Tanque digestor de lodos	5
2.1.4. Sensores	6
Tanques decantadores	6
Tanque reactor	6
Tanque digestor	7
3. Control del sistema	9
3.1. Modo automático	9
3.1.1. Control de válvulas	9
3.1.2. Control de temperatura	13
3.1.3. Control de bacterias	16
3.2. Modo Manual	18
3.3. Reseteo	18
4. Pantalla de explotación HMI	19
4.1. Representación en Intouch	19
4.1.1. Implementación variables	20
4.1.2. Pantalla principal	21
Visualización	22
Cambio de modo	24
Cambio de ventana	24
Inputs modo manual	24
4.1.3. Pantallas evolución temperatura y bacterias	25
4.1.4. Pantalla modos de operación	26
5. Conclusiones	29
Apéndice A. Guía instalación	31
A.1. Instalación de Wonderware	31
A.2. Configuración para la licencia	32

A.3. Posibles problemas	33
Apéndice B. Comunicación	35
B.1. Pasos a seguir	35
B.1.1. Codesys	35
B.1.2. Wonderware	37
B.2. Posibles problemas	39
Apéndice C. Planteamiento	41
C.1. Código simulación del sistema	41
C.2. Modo Reseteo código	45
C.3. Modo Manual código	46
<i>Índice de Figuras</i>	47
<i>Índice de Códigos</i>	49
<i>Bibliografía</i>	51

1 Planteamiento

Este proyecto ha sido realizado con el fin de mostrar todas las posibilidades de uso que presta el software de AVEVA System Platform para realizar un entorno HMI de un sistema. También se muestra la posibilidad que existe de conectar mediante protocolo OPC UA este programa a Codesys.

1.1 Objetivos

El objetivo de este proyecto ha sido evaluar las funcionalidades del software AVEVA System Platform y la posibilidad de comunicarse con otros dispositivos usando el protocolo OPC-UA, para su posible incorporación a la enseñanza de los mecanismos y recursos para la automatización de plantas industriales. Con objeto de facilitar el desarrollo del proyecto y dado que no se dispone de plantas reales con la complejidad necesaria para probar las posibilidades del SCADA se ha optado por realizar un modelo y controlarlo usando el software de Codesys, de uso gratuito para simulación. Todo el desarrollo y el funcionamiento del proyecto se ha hecho utilizando como único recurso un portátil personal, solo con la necesidad de adquirir una licencia para el software de AVEVA System Platform, también existe un modo demo del programa con funcionalidades más reducidas. Gracias a esto, cualquier persona desde su propio PC personal puede realizar este proyecto a la vez que interactuar con todos los elementos de control y visualización que mostraremos más adelante. Esto puede ser una gran mejora a la hora de la enseñanza en remoto ya que el propio estudiante no necesita de ningún tipo de PLC y puede establecer conexión de forma local entre ambos programas sin problema alguno.

Para explicar el proyecto que se ha ejecutado para este TFM, se ha decidido dividir el trabajo en varios apartados. En el capítulo 2 se contará exactamente como se ha simulado el sistema a controlar, en el apartado 3 se contará como se va a controlar este sistema, en el apartado 4 se pasará a explicar el desarrollo de la pantalla de explotación o HMI con la cual estará en contacto el usuario y por último se hablará de las conclusiones a las que se ha llegado durante este proyecto. También se indicarán anexos sobre la instalación de los programas, el proceso para comunicarlos entre sí y por último el código que se ha utilizado dentro de nuestro programa.

2 Sistema a controlar

Para este proyecto se ha decidido controlar una estación de depuración de aguas residuales más conocido por la abreviatura EDAR. Estas estaciones como bien indica su nombre tienen como función la limpieza de aguas residuales y aprovechamiento de los posibles desechos tales como el fango seco para producir biogas). Todos estos procesos se pueden ver reflejados en la siguiente web certificada acerca del estudio de las EDAR. [3]. A continuación se muestra una figura con los procesos principales:

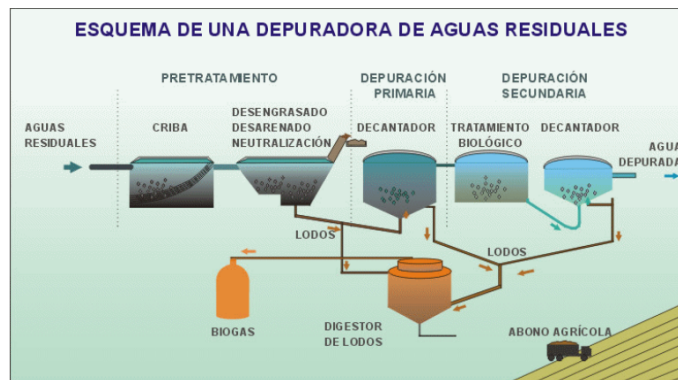


Figura 2.1 Ejemplo de EDAR.

En nuestro caso nos hemos centrado más concretamente en la depuración primaria y secundaria del proceso.

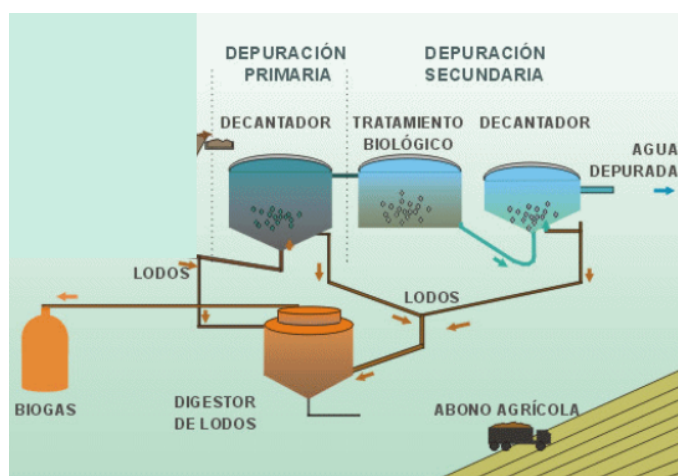


Figura 2.2 Ejemplo específico de EDAR.

2.1 Descripción detallada

Para este caso hemos tomado como referencia el siguiente sistema para simular un EDAR.

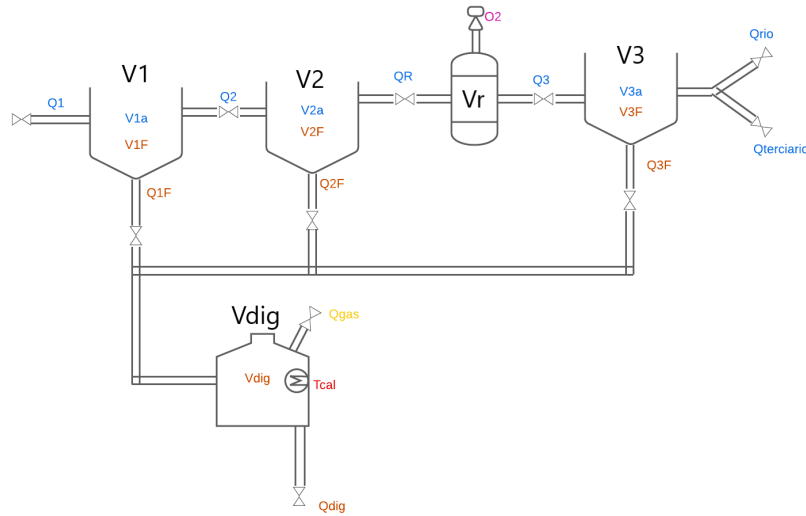


Figura 2.3 Representación de sistema a controlar.

Este sistema se compone de diferentes tipos de tanques:

- 1. Tanques decantadores:** En estos tanques se filtrará el agua residual dejando en la parte inferior los sedimentos que pueda tener. Estos tanques se pueden ver representados en la figura 2.3 como V_1 , V_2 y V_3 .
- 2. Tanque reactor:** En este tanque se utilizarán bacterias para eliminar posibles materia orgánica que pueda existir en el agua residual (Nitrógeno, Carbono...). Este tanque se puede ver representado en la figura 2.3 como V_r .
- 3. Tanque digestor de lodos:** En este tanque se calentará el fango para evaporar y conseguir biogás a partir de él. Este tanque se puede ver representado en la figura 2.3 como V_{dig} .

2.1.1 Tanques decantadores

Como se observa en la imagen tendremos tres tanques decantadores los cuales tendrán una cantidad de agua residual sin sedimentos (V_a) y una cantidad de sedimentos (V_f). Cada uno de estos tanques decantadores tendrá una entrada de agua residual y dos de salida (a excepción del tercer tanque que tendrá una posible salida adicional. El agua residual tendrá cierto porcentaje de agua sin sedimentos ($1 - X_1$) y otro porcentaje que será de sedimentos o fangos (X_1), este valor será variable en el primer tanque mientras que en los demás tendrá un valor de $X_1 = 0.2$. Por otro lado también se tendrá una válvula de salida para el agua sin sedimentos (Q_a) y una válvula de salida para los sedimentos acumulados (Q_f). Debido a que trataremos con sedimentos acumulados y con agua sin sedimentos diferenciaremos en dos volúmenes diferentes dentro de los tanques decantadores " V_a " para volúmenes sin sedimentos y " V_f " para volúmenes de sedimentos acumulados.

$$\Delta V_1 = \Delta V_{1a} + \Delta V_{1f} \quad (2.1)$$

$$\Delta V_{1a} = Q_1 \cdot (1 - X_1) - Q_2 \quad (2.2)$$

$$\Delta V_{1f} = Q_1 \cdot X_1 - Q_{1f} \quad (2.3)$$

$$\Delta V_2 = \Delta V_{2a} + \Delta V_{2f} \quad (2.4)$$

$$\Delta V_{2a} = Q_2 \cdot 0.8 - Q_r \quad (2.5)$$

$$\Delta V_{2f} = Q_2 \cdot 0.2 - Q_{2f} \quad (2.6)$$

$$\Delta V_3 = \Delta V_{3a} + \Delta V_{3f} \quad (2.7)$$

$$\Delta V_{3a} = Q_3 \cdot 0.8 - Q_{rio} - Q_{terc} \quad (2.8)$$

$$\Delta V_{3f} = Q_3 \cdot 0.2 - Q_{3f} \quad (2.9)$$

2.1.2 Tanque reactor

En este tanque reactor como bien se ha nombrado antes se llevará a cabo una descomposición de los elementos orgánicos que puedan existir en las aguas residuales. Para ello se usarán bacterias las cuales se introducirán en el tanque reactor y se alimentarán de estos elementos antes nombrados. Para la ecuación que nos indicara el crecimiento de las bacterias en función de la cantidad de oxígeno nos hemos basado en la ecuación de Monod 2.10, esta ecuación es un modelo matemático para el crecimiento de microorganismos [4]:

$$\lambda = \lambda_{max} \cdot \frac{S}{K_s + S} \quad (2.10)$$

Este modelo matemático nos da el crecimiento de dichas bacterias en función de la cantidad de sustrato limitante S , de la tasa máxima de crecimiento específica λ_{max} y la constante de media velocidad K_s .

En nuestro modelo estas bacterias ven limitado su crecimiento en función de cantidad de oxígeno que se le inyecte para prosperar, basándonos en el mismo modelo matemático pero cambiando que las propias bacterias necesitan de oxígeno para seguir viviendo hemos obtenido la siguiente ecuación:

$$\Delta Bacterias = \lambda_{max} \cdot \frac{O_2 - 0.1 \cdot Bacterias_{actuales}}{O_2 + K_b} \quad (2.11)$$

Por otro lado, este tanque, al no ser decantador, solo nos servirá saber la cantidad de agua residual que se contiene dentro de él, sin diferenciar el porcentaje de sedimentos y agua. Por consiguiente tendríamos la siguiente función respecto a la variación de volúmenes:

$$\Delta V_r = Q_r - Q_3 \quad (2.12)$$

2.1.3 Tanque digestor de lodos

Este tanque se llenará de los sedimentos o lodos provenientes de los tanques decantadores, una vez lo suficientemente lleno se empezará a calentar dicho lodo con un calentador el cual se irá regulando su temperatura para que la temperatura de los lodos se mantenga alrededor de 90°C. La siguiente ecuación reflejará el cambio de temperatura global de dichos lodos en función de los lodos entrantes (que tendrán una temperatura aproximada de 20°C), la temperatura de los lodos que se encuentre actualmente en el tanque y la temperatura del calentador.

$$T_{digactual} = \frac{K_1 \cdot T_f \cdot (Q_{1f} + Q_{2f} + Q_{3f}) + T_{dig} \cdot V_{dig} + K_2 \cdot T_{cal}}{(Q_{1f} + Q_{2f} + Q_{3f}) + V_{dig} + (K_1 + K_2)} \quad (2.13)$$

Reflejando las anteriores variables los siguientes valores:

1. T_{dig} : Esta variable hace referencia a la temperatura actual de los fangos que se encuentran en el tanque

2. T_f : Este es el valor de la temperatura a la cual entran los fangos provenientes de los tanques de decantación
3. V_{dig} : Es el volumen actual de los fangos que se encuentran en el tanque
4. T_{cal} : Temperatura actual del calentador que se encuentra dentro del tanque
5. K_1 : Constante que hace referencia a la cantidad de calor que transmiten los fangos al sistema global
6. K_2 : Constante que hace referencia a la cantidad de calor que transmiten el calentador al sistema global

Para la formula de esta transferencia de calor nos hemos basado en las ecuaciones dadas en el temario de transferencia de calor entre ellas la fórmula de temperatura de dos cuerpos en el estado estacionario:

$$T = \frac{K_1 \cdot T_1 \cdot L_2 + K_2 \cdot T_2 \cdot L_1}{K_1 \cdot L_2 + K_2 \cdot L_1} \quad (2.14)$$

Adicionalmente también se simulará el movimiento de los fangos dentro del tanque teniendo tres posibles válvulas de entradas (proveniente de los tres tanques de decantación) y una válvula de salida por donde saldrán el fango seco.

$$\Delta V_{dig} = Q_{1f} + Q_{2f} + Q_{3f} - Q_{dig} \quad (2.15)$$

Por último el biogás generado debido al calentamiento de estos fangos dentro del tanque será retirado del mismo una vez se sobre pase cierto umbral de presión dentro del propio tanque. Dicho biogás será expulsado del tanque mediante la apertura de una válvula incluida en la parte superior del tanque. Aquí se dispone las ecuaciones que se han usado para la simulación de ese incremento de gas y presión dentro del tanque.

$$V_{gas} := V_{gas} + K_g \cdot dig \cdot V_{dig} \cdot 0.1 \quad (2.16)$$

$$P := (V_{gas} / (13 - V_{dig})) \cdot K_p \quad (2.17)$$

2.1.4 Sensores

Para el control de todo el sistema se han implementado una serie de sensores, los cuales se han dividido en función de su localización dentro del propio sistema

Tanques decantadores

- Sensores de nivel: Estos sensores se activarán una vez el liquido del agua residual haya alcanzado cierto nivel en el tanque tendremos los siguientes sensores en los tanques decantadores. Hay que tener en cuenta que la letra X indicará al tanque en el que se encuentra el sensor pudiendo ser este 1, 2 o 3:
 - **SXA**: Este sensor se encuentra en la parte superior del tanque e indicará cuando el tanque esta lleno de liquido, siendo este limite 9 litros.
 - **SXM**: Este sensor se encuentra en la parte media del tanque e indicará cuando el liquido ha llegado al limite de 5 litros
 - **SXB**: Este sensor se encontrará en la parte baja del tanque y marcará cuando el liquido ha llegado al limite de 2 litros.
- Sensores de espesor: Estos sensores marcarán la cantidad de sedimentos que se encuentran en el agua residual frente a la cantidad de agua residual total [2]. Debido a que hay varios tanques decantadores se utilizará una X que se sustituirá en cada sensor en función del tanque donde se encuentre:
 - **SXF**: Este sensor de espesor se simulará en nuestro sistema dándonos un resultado de la cantidad de sedimentos entre la cantidad de agua residual total

Tanque reactor

- Sensores de nivel: Al igual que en los sensores de altura de los tanques decantadores tendremos tres diferentes sensores en el tanque reactor
 - **SRA**: Este sensor se encontrará en el limite de 9 litros de tanque reactor por lo que una vez llegue a dicho limite el liquido se activará este sensor.

- **SRM**: Este sensor se encontrará en el límite de 5 litros de tanque por lo que una vez llegue a dicho límite el líquido se activará este sensor.
- **SRB**: Este sensor se encontrará en el límite de 2 litros de tanque por lo que una vez llegue a dicho límite el líquido se activará este sensor.
- Sensor de bacterias: Este sensor medirá la cantidad de bacterias que se encuentran en el agua residual, midiéndolo a partir de la evolución de los sustratos que hay en el líquido.

Tanque digestor

- Sensores de altura: Al igual que en los sensores de altura de los tanques decantadores tendremos tres diferentes sensores en el tanque reactor
 - **SDig**: Este sensor se encontrará en el límite de 9 litros de tanque por lo que una vez llegue a dicho límite el fango se activará este sensor.
 - **SDigM**: Este sensor se encontrará en el límite de 4 litros de tanque por lo que una vez llegue a dicho límite el fango se activará este sensor.
 - **SDigB**: Este sensor se encontrará en la parte más baja del tanque por lo que en cuanto detecte que hay algo de fango o sedimentos en el interior del tanque de digestión se activará el sensor.
- Sensor de temperatura: Este sensor de temperatura (el cual se llamará Tdig en nuestro programa) medirá la temperatura a la que se encuentre el fango almacenado en el tanque. Este sensor nos ayudará a controlar junto con el calentador la temperatura a la que queremos que llegue a estar el fango para poder secarlo y poder usarlo como abono.
- Sensor de presión: Este sensor (el cual será llamado en nuestro programa como P) medirá la presión a la que se encuentra el tanque de digestión, si supera cierto umbral de presión debido a los gases generados en el calentamiento de los fangos, se podrá expulsar mediante una válvula que se encuentra en la parte superior del tanque.

3 Control del sistema

Diferenciaremos dos tipos de modos diferentes de funcionamiento, teniendo otro modo de transición entre ellos obligatorio. Aquí una breve descripción de las especificaciones de control para cada uno de estos modos:

1. **Automático:** Este modo realizará un control automático de los actuadores, tanto de válvulas como del calentador o el inyector de bacterias, utilizando valores de referencia para estos dos últimos por parte del usuario y los sensores que se sitúan en todos los tanques. Este modo será el modo en el que prioritariamente estaría trabajando el sistema EDAR
2. **Manual:** Este modo hace que todos los actuadores queden a disposición del usuario desde la maquina HMI, ya que con un simple click puede abrir o cerrar válvulas, subir o bajar la temperatura del calentador o controlar la cantidad de oxígeno que le entra a las bacterias del reactor.
3. **Reseteo:** Este modo tiene como función restablecer el sistema para que pueda asegurarse el pleno funcionamiento del modo automático.

3.1 Modo automático

En este modo se controlará de forma automática todas los actuadores que se han mencionado anteriormente. Para este control se puede dividir el sistema en tres subsistemas: válvulas (de líquido y de gas), calentador y válvula de oxígeno.

3.1.1 Control de válvulas

Las válvulas a controlar en este caso son las siguientes que se muestran en esta imagen:

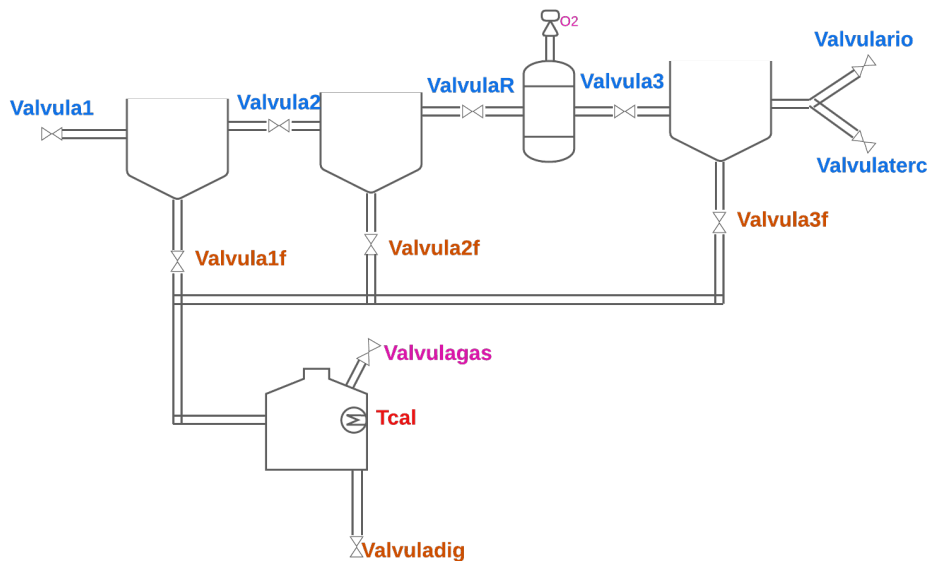


Figura 3.1 Válvulas del sistema.

- Valvula1:** Esta válvula es la entrada de agua residual al primer tanque de decantación, en principio su apertura estará limitada solo al sensor S1A el cual nos indicará si el tanque de decantación 1 esta lleno o no.



Figura 3.2 Control valvula 1.

- Valvula2:** Esta valvula conecta el tanque decantador 1 con el tanque decantador 2, de forma que todo el agua residual que le llega al tanque decantador 2 es a traves de esta valvula. Para que esta valvula se active se necesita que se cumplan 3 condicionantes: que el tanque decantador 1 este relleno hasta la mitad al menos (S1M activo), que el tanque decantador 2 no este completamente lleno (S2A negativo) y que la cantidad de sedimentos del tanque respecto al total del agua residual sea menor del 50% (SF1<0.5).

Para que vuelva a cerrarse esa valvula se ha implementado que sea si se cumple cualquier de estos tres condicionantes: que el tanque decantador 1 este a un nivel más bajo del sensor S1B (S1B negativo), que el tanque decantador este lleno (S2A activo) o que la cantidad de sedimentos del tanque respecto al total sea mayor que el 50%



Figura 3.3 Control valvula 2.

- ValvulaR:** Para el llenado del reactor es un poco más complejo ya que una vez este lo suficientemente lleno se empezará a ejecutar el proceso de inyección bacteriana para eliminar los sustratos y mientras dure ese proceso no se podrán añadir más agua a tratar. Para ello hemos dividido el control en diferentes

fases:

- **Llenado:** Esta variable nos indicará que el reactor se encuentra en fase de llenado, lo cual se producirá cuando se den tres factores: el reactor este vacío (SRB negativo), no este el proceso en fase de vaciado (Vaciado negativo), ni se haya cumplido el tiempo de espera Temp1 en el caso de que no se haya llenado al completo. Una vez nos encontremos en esta fase se deben de cumplir cuatro condicionantes más, los cuales son similares a los mencionados en los otros tanques, estos son: que el tanque 2 se haya llenado hasta la mitad al menos (S2M activo), que el tanque reactor no esta totalmente lleno (SRA negativo), que el proceso no este en fase Vaciado y que el espesor del tanque 2 sea menor al 50 % (SF2<0.5).



Figura 3.4 Control valvula Reactor.

- **Inyección Bacterias:** Como ya se ha mencionado antes, se dispondrá un contador de tiempo ya que si pasa el tanque cierto tiempo sin llenarse completamente (pero si lleno al menos hasta la mitad) se empezará con el proceso de inyección de bacterias. Una vez se haya llegado a ese tiempo de espera o al limite maximo de capacidad del tanque se activará una variable llamada InyectBact, la cual activará el control del oxígeno para el desarrollo de estas bacterias. Este control se explicará un poco más adelante en esta sección.



Figura 3.5 Control inyección bacterias.

- **Valvula3:** Esta variable se corresponde con el control de la valvula de salida del reactor y por tanto de la valvula de entrada al tanque decantador 3. Por ello su actuación dependerá del proceso que este transcurriendo en el propio reactor. Como hemos comentado anteriormente dentro de este reactor habrá dos fases diferenciadas, llenado e inyección de bacterias. Una vez se haya completado esta última, la cual se indicará mediante la variable Temp2, se empezará con el siguiente proceso
- **Vaciado:** Esta variable nos indicará que la fase de vaciado del reactor ha comenzado. La valvula 3 por tanto se abrirá solo si esta variable esta activa y el tanque decantador 3 no está completamente lleno (S3A negativo).



Figura 3.6 Control valvula 1.

- Valvula1f:** Este tipo de actuador funciona igual para las valvulas de fango de cualquier tanque decantador. Estas valvulas se activarán si se producen 5 condicionantes: el tanque de digestión no está completamente lleno, la presión dentro del tanque es menor a 5 atmosferas, hay liquido en el tanque decantador correspondiente, no es está expulsando fango ya calentado (Temp3 negativo) y el sensor de espesor nos marca que al menos un 50% es fango (desactivadonse cuando este porcentaje llega a 10%) o sedimentos dentro de ese agua residual.



Figura 3.7 Control válvula fango tanque 1.



Figura 3.8 Control válvulas fango.

- Valvulaterc y ValvulaRio:** Estas válvulas son las válvulas de salida del último tanque decantador, pudiendo el usuario escoger si el agua tratada va al río, a uso terciario o a ambos. Para el control de su apertura nos basaremos en los siguientes condicionantes: el usuario debe haber activado la variable para elegir si va al río, a uso terciario o ambos, la cantidad de agua tratada debe llegar al menos a la mitad del tanque y tiene que haber un espesor menor del 50% respecto a la cantidad de agua total.



Figura 3.9 Control valvula río y terciario.

- Valvulagas:** Esta válvula controlará la cantidad de gas que se encuentra en el tanque de digestión. Dicho gas se generará debido al calentamiento del fango a tratar. Esta válvula se activará cuando la presión sea mayor a 5 atmósferas y se desactivará cuando llegue a una presión menor que 4 atmósferas.



Figura 3.10 Control válvula gas.

- Valvuladig:** La válvula de salida del tanque digestión se activará una vez el fango haya llegado a la temperatura objetivo y se haya mantenido en esa misma temperatura durante un periodo de tiempo prolongado (Temp3 activo), a la vez que sigue habiendo fango dentro de ese tanque de digestión, una vez se vacíe por completo empezará de nuevo la fase de calentamiento del fango mediante el calentador.



Figura 3.11 Control válvula dig.

3.1.2 Control de temperatura

Para el control de la temperatura se usará un controlador PID el cual se diseñará usando el metodo de Ziegler-Nichols, para ello mediante el modo manual del programa se enviará una señal de entrada tipo escalón, donde se obtendrá el siguiente gráfico:

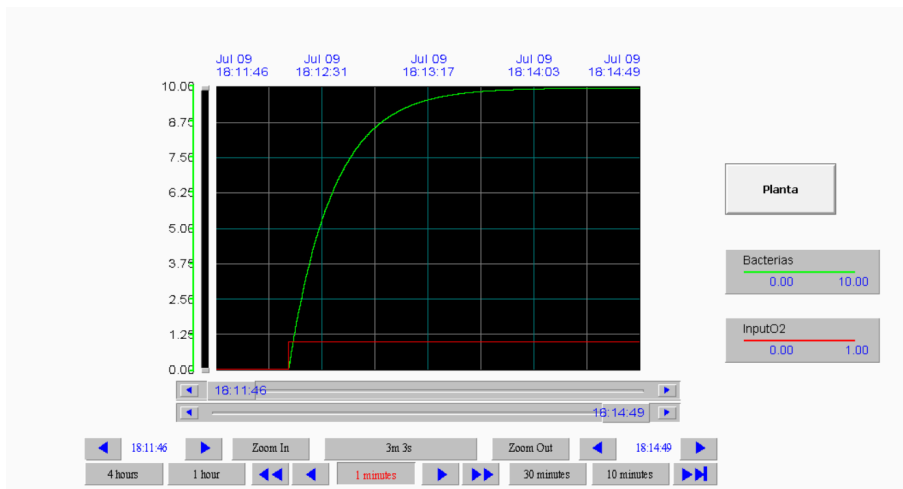


Figura 3.12 Representación respuesta sistema temperatura a escalón.

Una vez hecho esto observando la gráfica podremos obtener algunos valores como L (tiempo de retardo), τ (tiempo del sistema) y K (constante del sistema). Consiguiendo los siguientes valores:

- $\tau = 57$
- $L = 0.15$
- $K = 1$

A partir de estos datos se puede hacer una primera aproximación para el diseño de los parámetros de nuestro controlador, usando la fórmula de Ziegler-Nichols para el diseño de los parámetros de control:

Controlador	K_p	τ_i	τ_d
P	$\frac{\tau}{KL}$	∞	0
PI	$0.9 \frac{\tau}{KL}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{\tau}{KL}$	$2L$	$0.5L$

Tabla de Ziegler Nichols – Método 1

Figura 3.13 Fórmula de Ziegler-Nichols.

A partir de esta formula sacaremos los siguientes parámetros:

- $K_p = \frac{1.2 \cdot \tau}{K \cdot L} = 456$
- $T_i = 2 \cdot L = 0.3$
- $T_d = 0.5 \cdot L = 0.07$

Utilizando este controlador en nuestro sistema y simulándolo tenemos la siguiente gráfica:

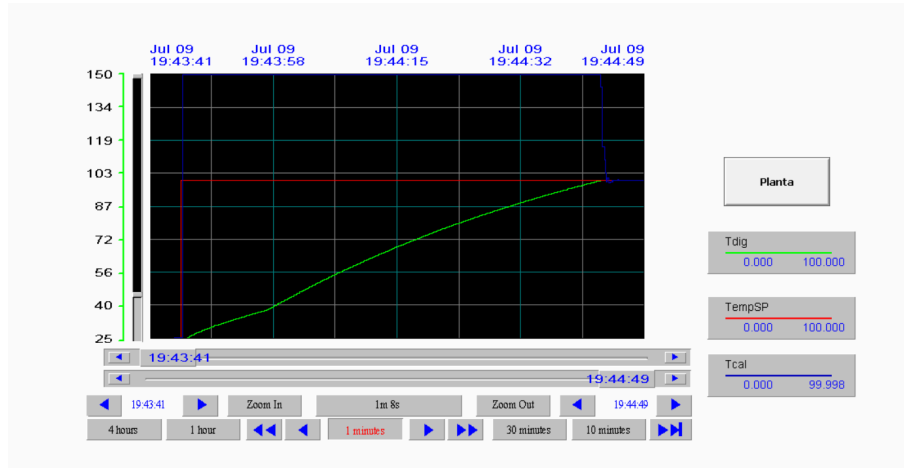


Figura 3.14 Controlador con setpoint en 100° temperatura 1a iteración.

Analizando esta gráfica vemos que cuadra para el control de nuestro sistema, sin embargo debido a que no podemos tener una entrada que varíe tan repentinamente, necesitaremos bajar el valor de nuestro constante proporcional para no tener variaciones tan bruscas a la entrada. A continuación se probará con los siguientes valores:

- $K_p = \frac{1.2 \cdot \tau}{K \cdot L} = 4$
- $T_i = 2 \cdot L = 0.3$
- $T_d = 0.5 \cdot L = 0.07$

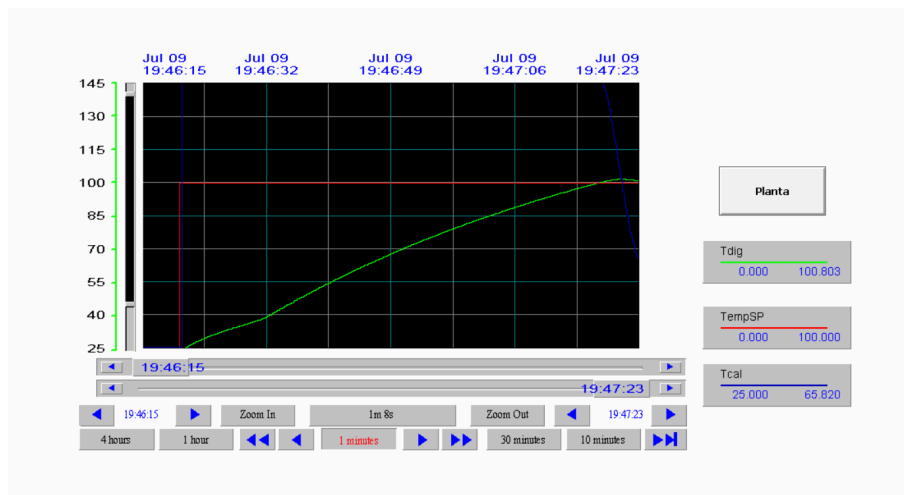


Figura 3.15 Controlador con setpoint en 100° temperatura 2a iteración.

Observamos que se sigue dando la misma problemática que anteriormente comentamos por lo que esta vez se baja tanto el valor de la constante proporcional como el de la constante integral dejándonos con los siguientes valores:

- $K_p = \frac{1.2 \cdot \tau}{K \cdot L} = 0.02$
- $T_i = 2 \cdot L = 0.05$
- $T_d = 0.5 \cdot L = 0.07$

Usando estos últimos parámetros nos encontramos con la siguiente respuesta del sistema:

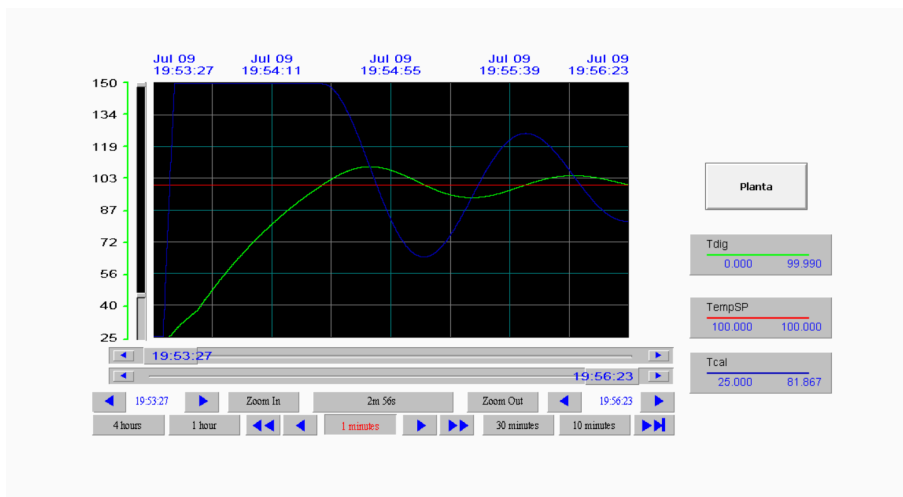


Figura 3.16 Controlador con setpoint en 100° temperatura 3a iteración.

Finalmente se usará estos últimos valores para el controlador priorizando una menor variación de la señal de entrada a la rapidez del sistema a llegar al setpoint.

Por lo tanto, usando los últimos parámetros antes mencionados, este control usará la siguiente fórmula:

$$G(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right) \quad (3.1)$$

$$= 0.02 \cdot \left(1 + \frac{1}{0.05 \cdot s} + 0.07 \cdot s\right) \quad (3.2)$$

$$= \frac{0.0014 \cdot s^2 + 0.02 \cdot s + 0.4}{s} \quad (3.3)$$

Una vez se haya mantenido la temperatura estable durante unos 5 segundos se activará la variable Temp3 para la apertura de la válvula de salida del tanque de digestión. Usando un setpoint para dicha temperatura de 100°C y partiendo de temperatura ambiente 25°C tenemos la siguiente evolución.

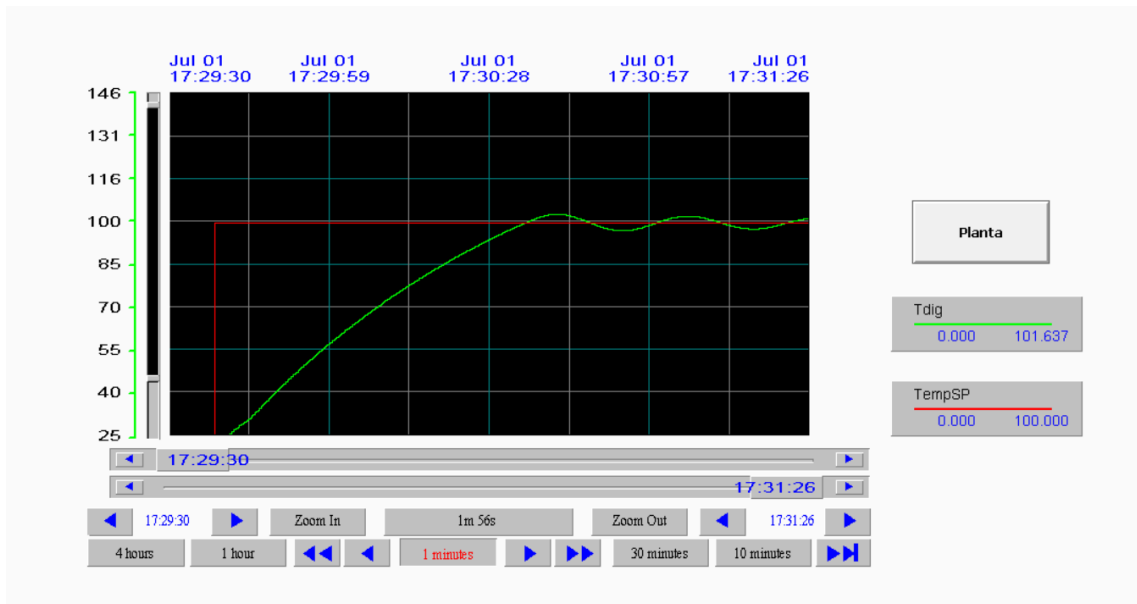


Figura 3.17 Control temperatura fango dentro del tanque de digestión.

3.1.3 Control de bacterias

Para el control de bacterias se usará un controlador PID el cual se diseñará usando el método visto anteriormente de Ziegler-Nichols, para ello mediante el modo manual del programa se enviará una señal de entrada tipo escalón, donde se obtendrá el siguiente gráfico:

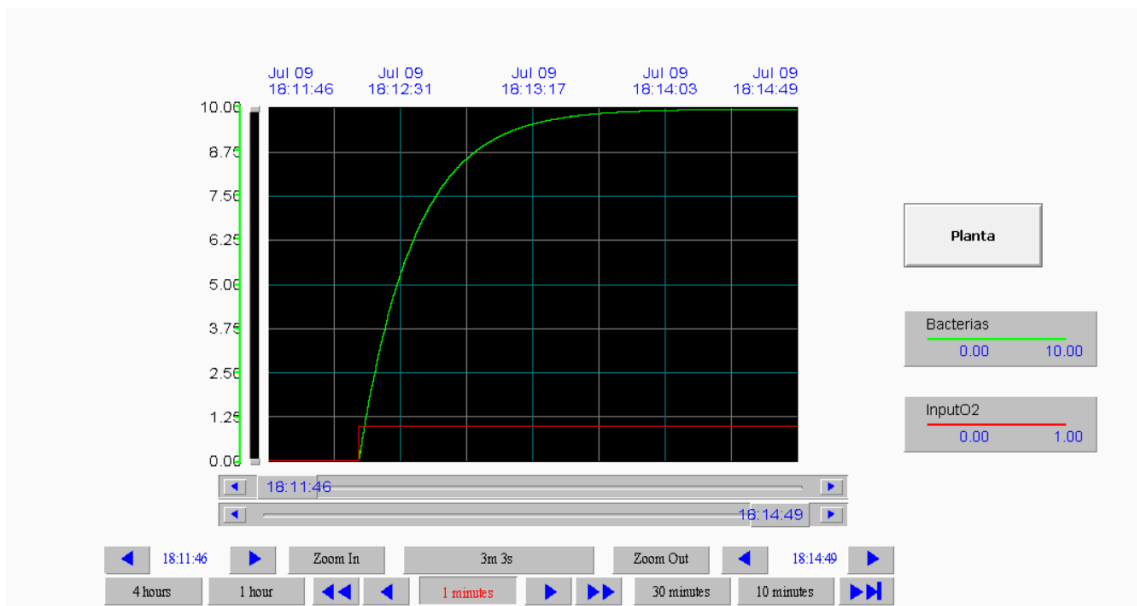


Figura 3.18 Representación respuesta sistema bacterias a escalón.

Una vez hecho esto observando la gráfica podremos obtener algunos valores como L (tiempo de retardo), τ (tiempo del sistema) y K (constante del sistema). Consiguendo los siguientes valores:

- $\tau = 20$
- $L = 0.15$
- $K = 10$

A partir de estos datos se puede hacer una primera aproximación para el diseño de los parámetros de nuestro controlador:

- $K_p = \frac{1.2 \cdot \tau}{K \cdot L} = 16$
- $T_i = 2 \cdot L = 0.3$
- $T_d = 0.5 \cdot L = 0.07$

Utilizando este controlador en nuestro sistema y simulándolo tenemos la siguiente gráfica:

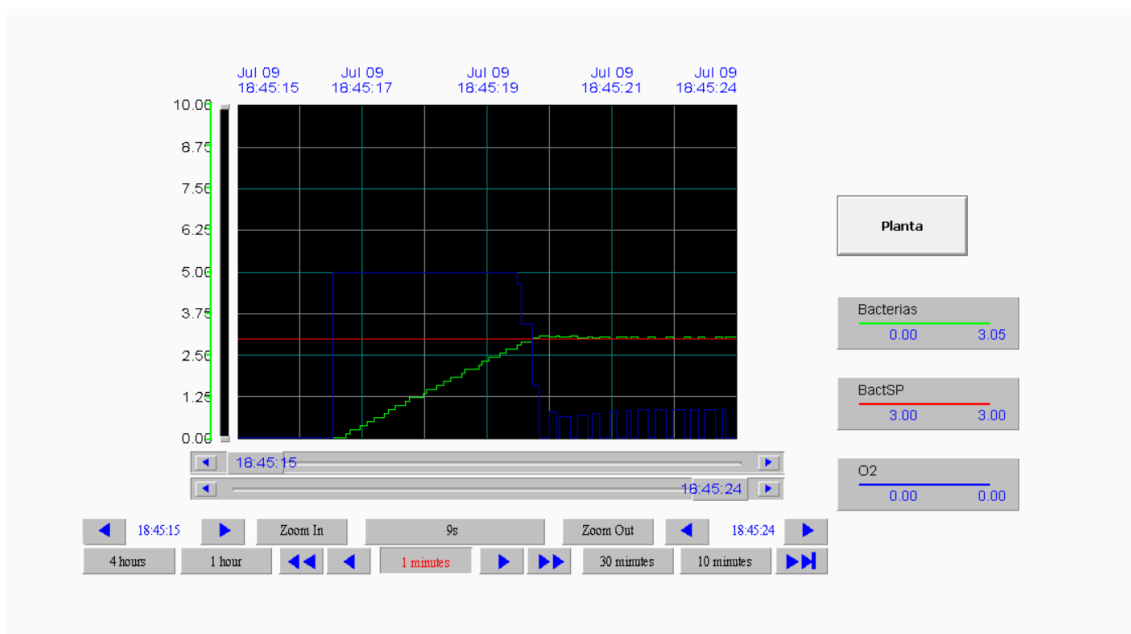


Figura 3.19 Controlador con setpoint en 3mg bacterias 1a iteración.

Se observa que hay una oscilación en régimen permanente, esto puede ser debido a que tenemos una constante K_p algo grande, para ello bajamos esta constante a 4, quedándonos así con los siguientes valores:

- $K_p = 4$
- $T_i = 0.3$
- $T_d = 0.07$

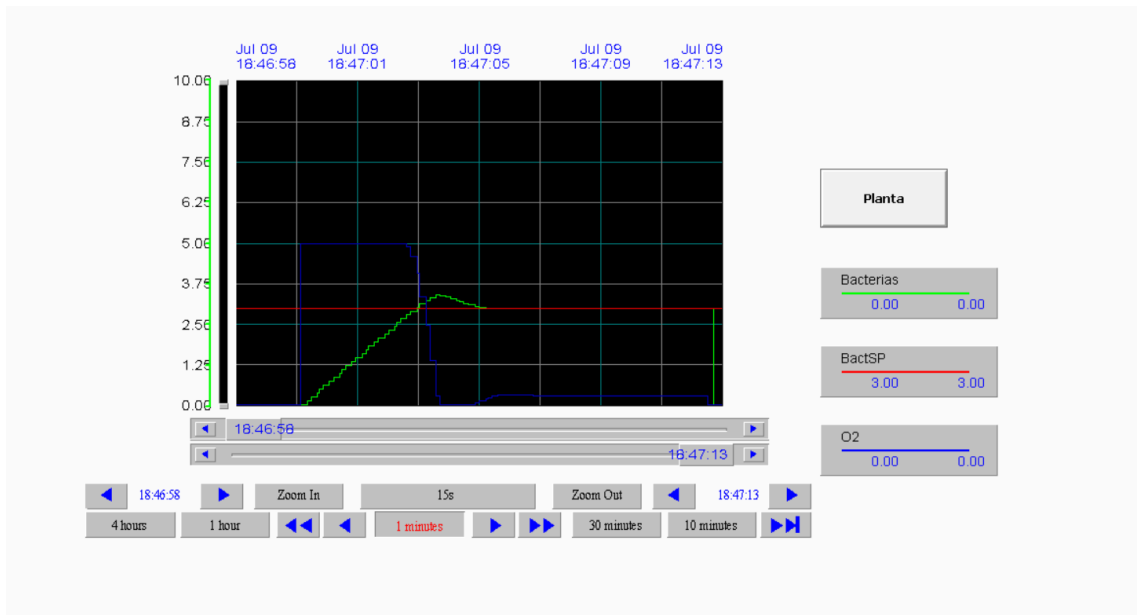


Figura 3.20 Controlador con setpoint en 3mg bacterias 2a iteración.

Finalmente se ve como tenemos un control óptimo y sin variaciones en la señal de entrada en régimen permanente.

Por lo tanto, usando los últimos parámetros antes mencionados, este control usará la siguiente formula:

$$G(s) = K_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right) \quad (3.4)$$

$$= 4 \cdot \left(1 + \frac{1}{0.3 \cdot s} + 0.07 \cdot s\right) \quad (3.5)$$

$$= \frac{0.28 \cdot s^2 + 4 \cdot s + 13.33}{s} \quad (3.6)$$

Una vez se haya mantenido durante un tiempo constante la cantidad de bacterias en torno al setpoint se activará la variable Temp2 dando comienzo al vaciado del tanque.

3.2 Modo Manual

Para el modo manual simplemente añadiremos unas variables adicionales para que el usuario pueda activar o desactivar cualquier actuador desde el HMI. Más adelante se puede observar dicho código implementado.

3.3 Reseteo

Para el paso del modo automático al modo manual no habrá ningún paso previo, sin embargo para el cambio de modo manual a automático si habrá un proceso intermedio con objeto de poner el sistema en condiciones iniciales para comenzar de forma correcta el control en modo automático. Este modo (llamado Reseteo en nuestro programa) vaciará por completo todos los tanques vaciando todo el fango sin llegar a calentarlo y el agua residual se expulsará por la válvula terciaria. En el momento que todos los sensores de los tanques den negativo y por lo tanto estén vacíos los tanques, se pasará de nuevo al modo automático de funcionamiento del sistema.

4 Pantalla de explotación HMI

4.1 Representación en Intouch

Para la representación de nuestro HMI o SCADA, es decir, la herramienta o interfaz que visualizar el usuario encargado de la planta para monitorizar y controlar la planta, se usará el programa AVEVA Intouch 2020, dicho programa deberá solo podrá abrirse como administrador del dispositivo, en este caso el portátil o PC que este usando el usuario. Este programa tiene multitud de funcionalidades y herramientas para la visualización de sistemas, sin embargo para este caso solo usaremos las herramientas que se disponen en la siguiente imagen:

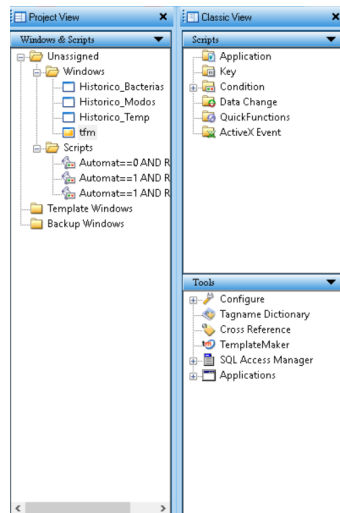


Figura 4.1 Herramientas usadas en Intouch .

Dentro de nuestro programa se podrán usar diferentes pestañas para visualizar diferentes partes del sistema a controlar, para nuestro caso hemos creado un total de cuatro pestañas, las cuales tienen las siguientes funciones:

- **Tfm**: Esta será la pantalla principal del proyecto, en esta pantalla podremos visualizar el estado de todos los sensores y actuadores en tiempo real, a la vez que introducir nuevos setpoints o pasar a controlar los actuadores de forma manual.
- **Histórico_bacterias**: En esta pantalla se podrá visualizar el estado de la cantidad de bacterias de forma gráfica que hay en todo momento en el tanque reactor, pudiendo también compararlo con la cantidad de referencia que se le ha pasado por parte del usuario. Estos datos serán guardados por el programa durante un día completo, pudiendo comparar los resultados que se han obtenido a lo largo de todo un día.

- **Histórico_temp:** En esta pantalla, al igual que en la anterior pantalla, podremos visualizar una variable a controlar que en este caso será la temperatura del fango que se encuentra dentro del tanque de digestión. Este dato junto con la temperatura de referencia dada por el usuario podrán visualizarse durante todo un día, al igual que en el caso de las bacterias.
- **Histórico_Modos:** La última pantalla accesible por el usuario se da para que el usuario pueda tener un seguimiento de los modos que se han ido usando a lo largo del control del sistema, pudiendo observar en todo momento que modos estaban en uso y a que hora se daban estos.

Adicionalmente a estas pestañas podremos incluir unos pequeños scripts que estarán funcionando de forma permanente mientras se ejecuta el programa. Los scripts usados para nuestro programa es para la funcionalidad de la pestaña Historio_Modos, la cual se explicará en el apartado 4.1.4

4.1.1 Implementación variables

Para el uso de las variables que hemos compartido a través de OPC UA provenientes del programa de Codesys, lo cual se comenta en el capítulo B, tendremos que hacer una configuración previa en el programa de AVEVA Intouch. Para hacer esto lo primero que deberemos introducir son estas variables a nuestro programa de Intouch. Para ello iremos al apartado de herramientas mostradas en la anterior imagen, más concretamente a *Tagname Dictionary*.

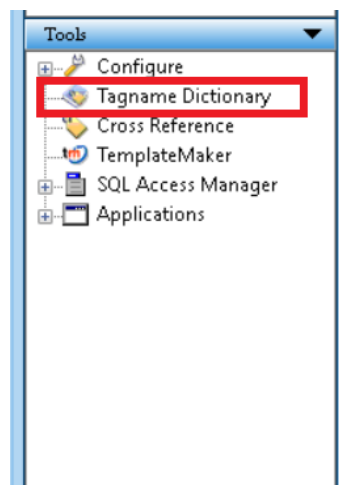


Figura 4.2 Tagname Dictionary.

A continuación se deberá añadir la variable que se quiera incluir al SCADA, dicha variable tiene que estar configurada anteriormente en la comunicación dentro del programa System Management Console de AVEVA. Para hacer esto pincharemos en New y agregaremos la variable como una variable de entrada/salida como se ve en la siguiente imagen (diferenciando también el tipo de variable si es booleana, entera, real...)

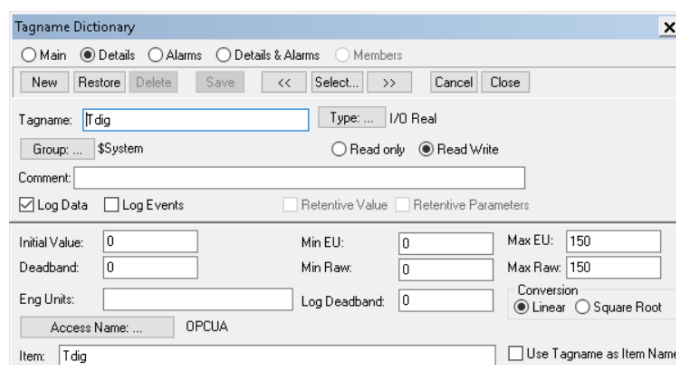


Figura 4.3 Configuración variable .

Una vez hecho esto debemos de configurar cual es la instancia que se creo en el programa SMC por donde van a pasarnos esas variables, para ello se pinchará en apartado Access Name e introduciremos dicha instancia si no se ha creado.

Figura 4.4 Grupo de acceso por donde se comunicará la variable.

Por último introduciremos el nombre con el que se había nombrado dicha variable en dicha instancia en el apartado de Item. También hay que añadir que se podrá incluir a la configuración de la variable si solamente va a ser de lectura o de escritura/lectura, si tiene un rango máximo o mínimo o si quiere que se guarde todos los valores que se hayan obtenido durante un periodo de tiempo (se explicará en el apartado 4.1.3

Figura 4.5 Detalles configuración variable.

4.1.2 Pantalla principal

La pantalla principal para la visualización del proyecto será la que se muestra en la siguiente imagen:

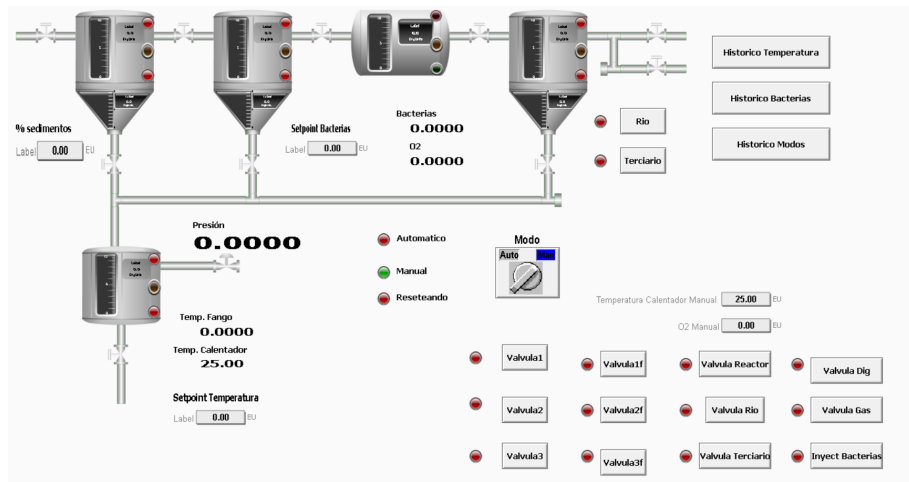


Figura 4.6 Ventana principal del HMI .

Esta pantalla es la que lleva más complejidad por lo cual pasaremos a explicarla por partes tal y como se refleja en esta imagen:

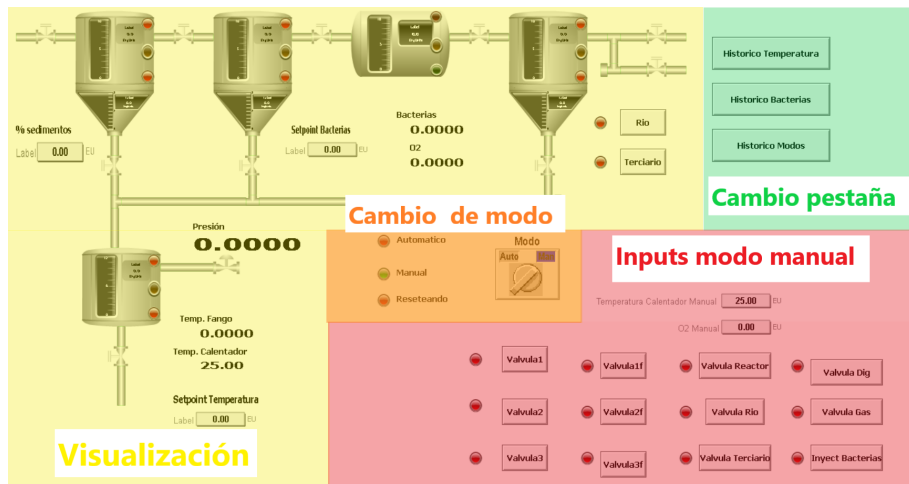


Figura 4.7 Ventana principal por partes.

Visualización

Para la representación del sistema se usarán diferentes tipos de elementos entre ellos los siguientes:

- Tanques de liquido:** Estos tanques pueden configurarse varios parámetros, entre ellos la variable que indicará el nivel actual del liquido, la cantidad mínima y máxima de liquido que puede albergar dicho tanque e incluso el color del propio tanque.



Figura 4.8 Configuración elemento tanque.

Para la representación del tanque decantador se ha usado dos tipos de tanques distintos, uno cilíndrico para el agua sin sedimentos (la cual se encontrará en la parte superior y recogerá los valores de las variables V1a, V2a o V3a) y uno cónico para el fango o sedimentos del agua residual (la cual se encontrará en la parte inferior en color marrón y recogerá valores de las variables V2f, V2f o V3f).

- **Válvulas:** Las válvulas que se representan en el SCADA podrán servir de ayuda al usuario ya que se podrá visualizar si están abiertas o cerradas en todo momento, estando abiertas cuando se encuentren en color verde y estando cerradas cuando no presenten color. Para recoger esta información dispondremos en la configuración del elemento de la válvula la variable que controle la actuación de dicha válvula.

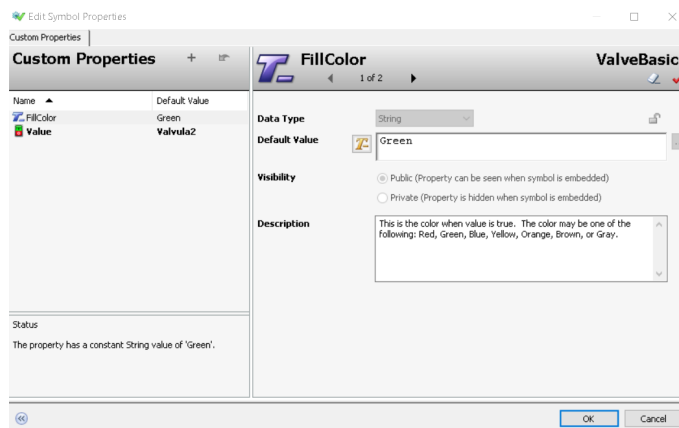


Figura 4.9 Configuración válvulas.

- **Tuberías:** Las tuberías solamente tendrán un carácter visual para el usuario ya que al no recoger ningún tipo de evidencia del estado de la misma a través de sensores el usuario no puede el estado actual de las mismas a través del HMI
- **Alarmas:** En este caso se ha decidido implementar alarmas visuales para mostrar los niveles de llenado de los tanques y cuando están activas las salidas de río y terciario.
- **Displays numéricos:** Se han añadido displays numéricos para señalar valores importantes como las temperaturas (tanto del calentador como del propio fango), las bacterias actuales y el oxígeno que se encontraba en el tanque reactor y la presión a la que se encontraba el tanque de digestión. Adicionalmente a esto, se han añadido displays que servían de inputs numéricos para la temperatura de referencia, el setpoint de bacterias a conseguir y la cantidad de sedimentos que recogía el primer tanque decantador.
- **Botones:** Los botones que se disponen en la visualización tienen el objetivo de activar o desactivar las variables que se añadan en dicho elemento. Para que el usuario cambie el valor de la variable booleana

al pinchar en el botón éste deberá configurarse como un botón de tipo Toggle, tal y como se muestra en la imagen.

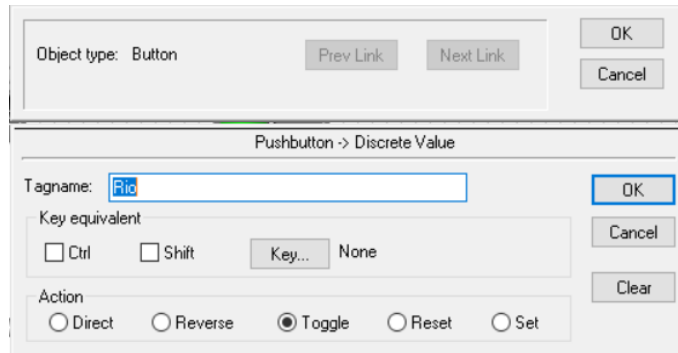


Figura 4.10 Configuración botones .

Cambio de modo

Para el cambio de modo se usará un switch el cual solo podrá conmutar una variable para que este activada o desactivada, la cual en este caso será la variable *Automat*. Seguidamente para poder representar el modo reseteando, se añadirá la variable *Reset* y se implementarán tres alarmas visuales los cuales nos mostrarán el modo actual de control en el cual se encuentra el sistema.

Cambio de ventana

Para poder visualizar las otras ventanas antes nombradas tendremos unos botones los cuales al pinchar nos abrirá la pestaña correspondiente elegida. Para configurar el botón tendremos que elegir *Show Window* en vez de *Discrete Value* en la sección de *Touch Pushbuttons*.

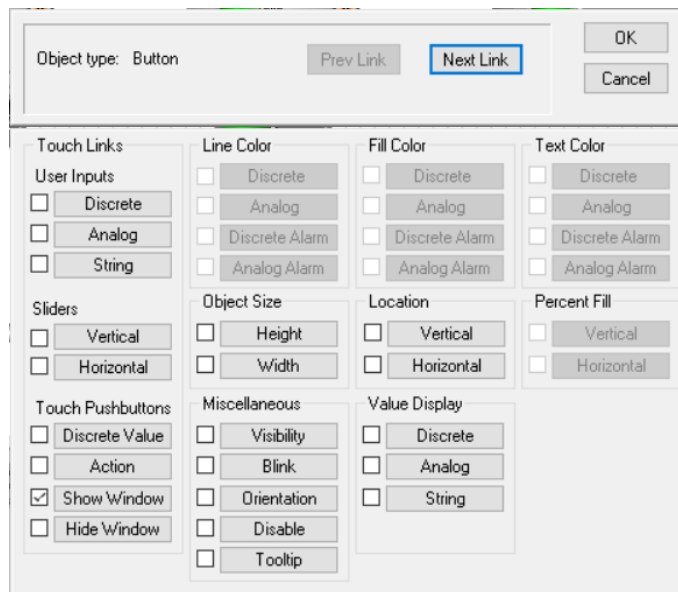


Figura 4.11 Configuración acción de botón.

Inputs modo manual

Estos inputs serán tanto de booleanos, en los cuales se usarán botones como en anteriores ocasiones, como numéricos, en los cuales se usarán displays con entrada numérica. Sin embargo estos botones y displays solo trabajarán con las variables que se usarán en el modo manual y no sobre las variables de los actuadores de forma directa, por lo que el valor que tomen solo se llevará a cabo en el sistema una vez este pase a modo manual.

4.1.3 Pantallas evolución temperatura y bacterias

La pantalla de evolución de temperatura y bacterias tendrán la siguiente apariencia:

Figura 4.12 Pantalla de evolución.

Para estas pantallas tendremos que usar un total de tres nuevos elementos para su representación:

- **Historical Trend Chart:** este elemento será la gráfica en sí la cual nos representará la evolución de las variables seleccionadas, dichas variables tienen que tener activadas el campo "Log Data" para poder ser representadas en la gráfica. Para introducir las variables a visualizar deberemos ir a la sección "Pensz" añadir dichas variables.

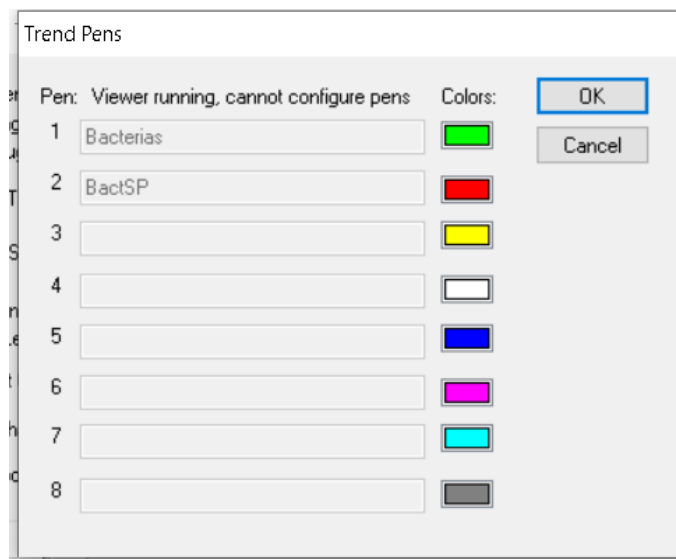


Figura 4.13 Configuración entradas elemento Historical Tren Chart.

De forma adicional se crearán unas variables adicionales, las cuales podremos renombrar como nosotros queramos. Estas variables, que en nuestro caso se llamarán X e Y, se recogerá la información recibida en la gráfica y deberán utilizarla los dos siguientes elementos que se mostraran a continuación.

- **Trend Panel Chard:** Es un elemento que nos servirá para poder movernos de una forma más precisa a través de la gráfica, pudiendo hacer zoom in o zoom out a cualquier lugar de la gráfica, también nos dispondrá el tiempo que se está mostrando en la gráfica actualmente. Para hacer uso de este elemento deberemos incluir en la configuración del mismo las variables generadas por la gráfica llamadas en este caso X e Y.

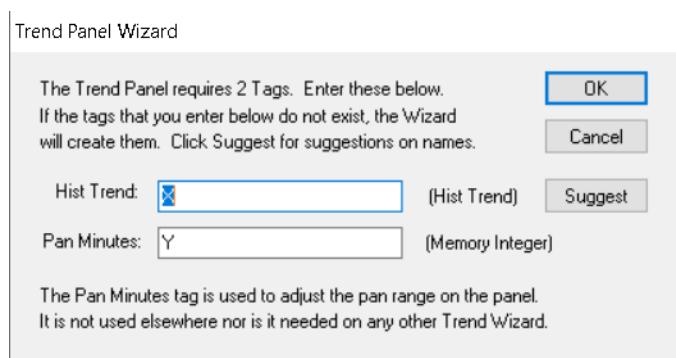


Figura 4.14 Configuración TrendPanel.

- Trend Legend:** Este elemento nos marcará con precisión el valor que toma la variable seleccionada en el momento del tiempo que estemos seleccionando en ese instante. Debido a que para nuestro proyecto queremos visualizar el estado de la variable más su setpoint se necesitarán dos elementos de este tipo para recoger los valores de ambas variables. Al igual que en el anterior elemento deberemos añadir las variables creadas por el elemento *Historical Trend Chard*.

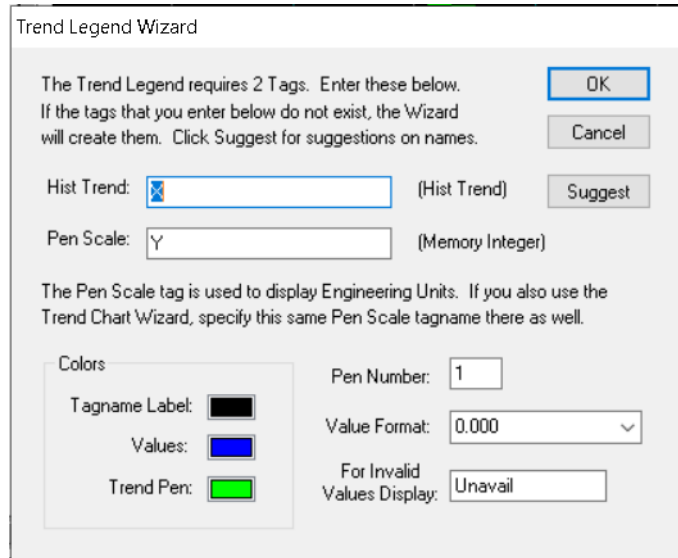


Figura 4.15 Configuración TrendLegend.

4.1.4 Pantalla modos de operación

La pantalla para poder visualizar los modos de operación usados en todo momento del proyecto usará un solo elemento, sin embargo habrá que hacer unas configuraciones en las variables que se van a usar para este elemento.

- Se deberán crear tres variables adicionales para representar cada modo de control, estas variables serán en este caso propias del programa y no de entrada/salida como las que se han creado anteriormente
- Estas variables tendrán que ir a otro grupo diferente que las demás variables anteriormente creadas, para ello crearemos un grupo diferente llamado "Modos".

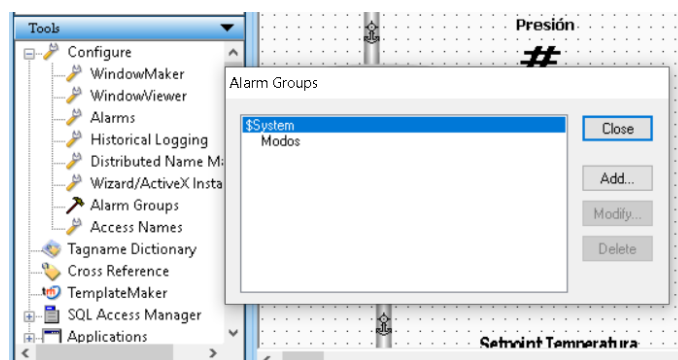


Figura 4.16 Creación de nuevo Alarm Group.

- A continuación, se deberá ir a la sección Alarms, aquí se deberá rellenar cuando se activará la alarma en nuestra pantalla de operación (en este caso cuando estén activos) y el comentario que se mostrará.
- Por último se crearán unos scripts para activar o desactivar estas nuevas variables con las variables de entrada/salida que teníamos en un principio dispuestas en nuestro HMI y controlador. Aquí una muestra de uno de esos scripts.

5 Conclusiones

Una gran parte del tiempo dedicado al proyecto ha sido para estudiar como conectar los programas usados (Codesys y Aveva System Platform). Una vez realizado esto, que es lo que ha sido más costoso, se ha desarrollado todo el modelado del sistema junto con su control y visualización en estos programas. Debido a la falta de tiempo para dedicarle a estos puntos puede haber diversos puntos de mejora respecto al proyecto. Esos posibles puntos de mejora son los siguientes:

- La simulación de sistema podría ser en tiempo real y no de forma discreta como se hace actualmente con el programa de Codesys. Para hacer está simulación del sistema en tiempo real se podría usar un programa como Matlab comunicándolo también mediante protocolo OPC UA.
- En el caso de haber añadido el programa de Matlab para la simulación del sistema nos hubiera dado pie a un mejor estudio del sistema ayudando así a conseguir un mejor control mediante el uso de ciertos métodos de diseño de PID tales como Ziegler-Nichols [1]. Dicho método debido a que se nos presenta un sistema discreto nos hace mucho más difícil observar la evolución del sistema y el punto de inflexión de la gráfica (método de sintonización por la respuesta al escalón)
- Respecto al entorno HMI de nuestro proyecto podría haberse implementado más ventanas. Aunque en este caso no se observaba que fueran tan interesantes como las dispuestas en este proyecto, podrían añadirse más ventanas para reflejar en un histórico la activación y desactivación de los actuadores y/o sensores. Esto podría ayudar al usuario a descubrir cual es la raíz de los problemas que puedan surgir durante el funcionamiento de la planta.

Apéndice A

Guía instalación

A.1 Instalación de Wonderware

Para este proyecto utilizaremos Wonderware System Platform 2020. Para la instalación utilizaremos el siguiente enlace de descarga para obtener el ejecutable de dicho programa:
<https://softwaresupportsp.aveva.com/>

Una vez hecho esto, pasaremos a instalar el programa. Nos saldrá la siguiente pestaña donde tendremos que escoger en *Product Based Selection*

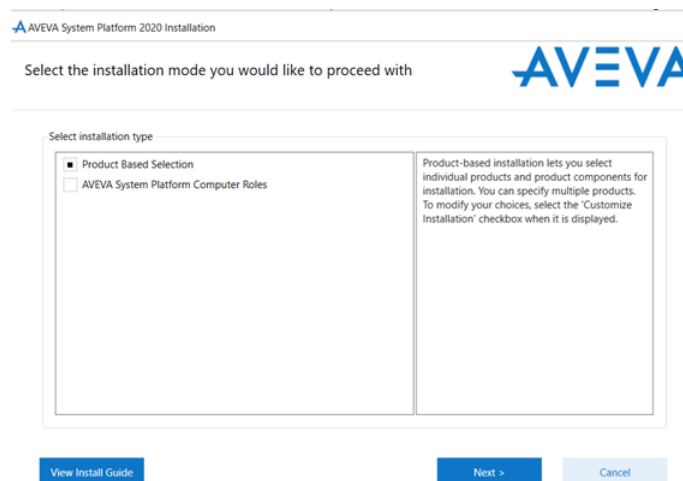


Figura A.1 Instalación programa AVEVA System Platform.

Lo siguiente tendremos que escoger todos los subprogramas que se encuentran a excepción del Intouch Access Anywhere y Historian.

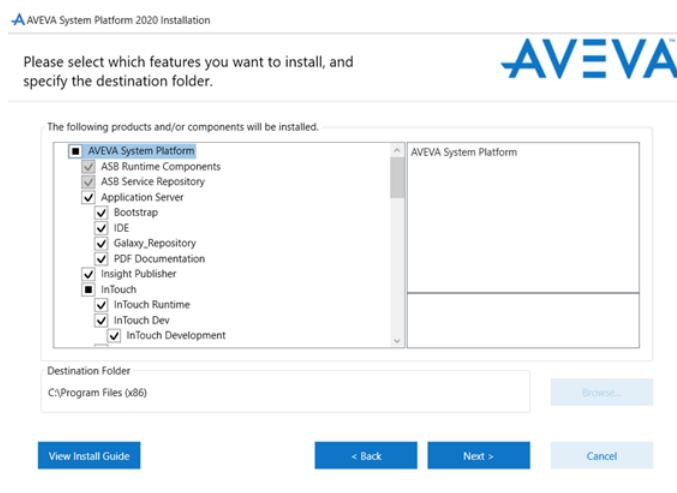


Figura A.2 Instalación por producto.

A.2 Configuración para la licencia

Para la configuración de la licencia deberemos ir a la carpeta de AVEVA y escoger el servicio de Enterprise License Manager, una vez hecho esto se nos abrirá en el explorador la siguiente pestaña



Figura A.3 Interfaz licencia.

Deberá aparecer nuestro PC y tendremos que pinchar sobre él, después nos saldrá la siguiente pestaña

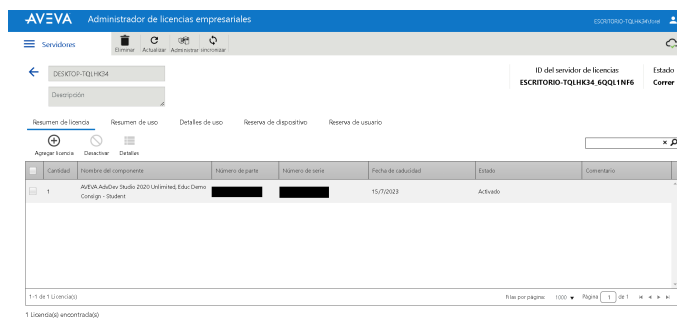


Figura A.4 Interfaz licencia.

Deberemos escoger la licencia correspondiente para poder utilizar los servicios de AVEVA, como podemos observar en la anterior imagen. Por último deberemos escoger nuestro PC como el servidor de administración del sistema, para ello abriremos el configurador de AVEVA, iremos a la opción System Management Server y escogeremos la siguiente opción tal y como sale en la siguiente imagen

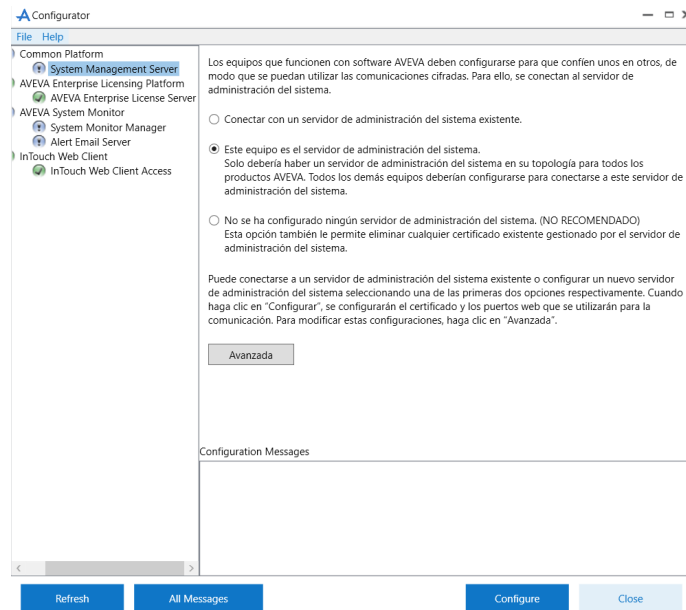


Figura A.5 Configurator AVEVA.

Por último le daremos a configure y ya podremos utilizar nuestro programa de AVEVA en nuestro ordenador personal.

A.3 Posibles problemas

Un posible problema que se nos puede generar a la hora de instalar este programa es que durante el proceso de aceptación de términos de uso no se nos muestre dicho documento y por lo tanto no nos de acceso a la opción de aceptarlos. En la siguiente imagen se muestra dicho problema:

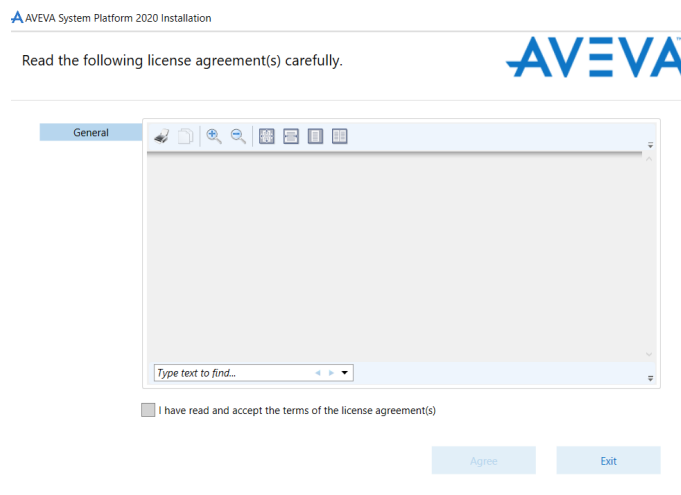


Figura A.6 Problema durante instalación .

Si llega a suceder este problema se deberá actuar de la siguiente forma:

1. Accederemos al Panel de Control de nuestro PC y seguidamente iremos a Programas tal y como se indica en la siguiente imagen.



Figura A.7 Panel de Control.

2. Una vez ahí accederemos a programas y características y por último actualizaciones instaladas.

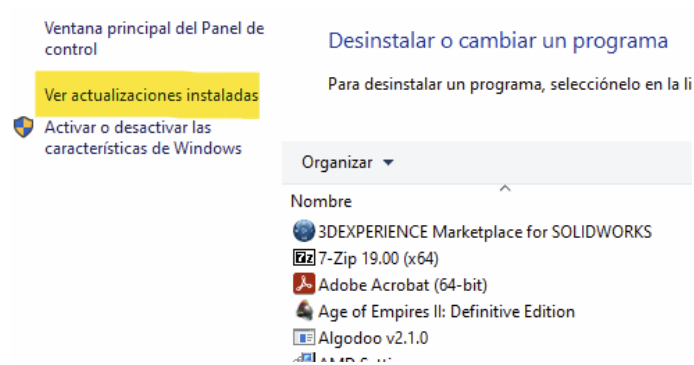


Figura A.8 Actualizaciones de programas.

3. Una vez hayamos accedido a dicha pestaña habrá que buscar actualizaciones que puedan entrar en conflicto con la instalación del System Platform de AVEVA para ello tendremos que buscar actualizaciones con el siguiente nombre:

- KB5020872
- KB5020874

Hecho esto podremos volver al proceso de instalación del programa de AVEVA y seguir con el mismo.

Apéndice B

Comunicación

Para comunicar ambos programas (AVEVA System Platform y Codesys) hemos optado por comunicarlos mediante protocolo OPC UA. Para este caso hemos elegido una tasa de refresco de las variables a comunicar de 0.1 segundos, debido a que la propia simulación tiene el mismo tiempo de ejecución es el tiempo máximo para esta tasa de refresco.

B.1 Pasos a seguir

Lo primero será configurar cada programa para poder habilitar la comunicación mediante este protocolo.

B.1.1 Codesys

Por la parte de Codesys tendremos que configurar los siguientes parámetros para poder comunicar ambos programas. Primero se deberá iniciar el *Soft PLC* para poder realizar la simulación del proyecto desde nuestro propio portátil sin necesidad de utilizar un PLC interno. Para ello nos iremos a nuestra barra de tareas e iniciaremos nuestro programa *Codesys Control Win Systray - x64* tal y como sale en la siguiente imagen

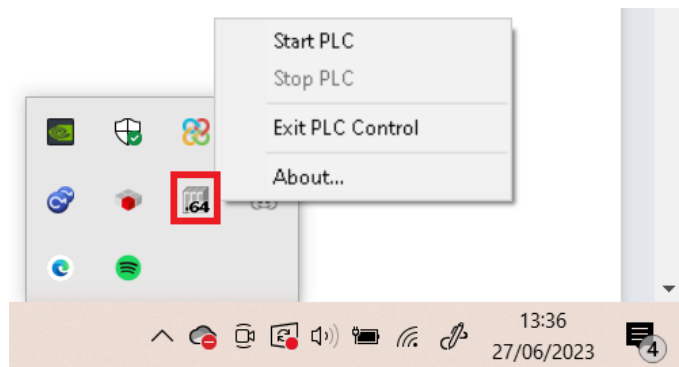


Figura B.1 Codesys Control Win Systray.

Una vez hecho esto, volviendo al programa de Codesys, deberemos ir al apartado configuración de comunicación dentro de la pestaña device.

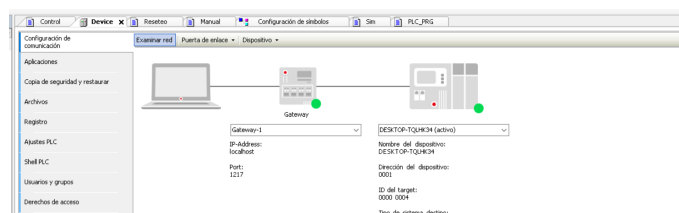


Figura B.2 Configuración comunicación Codesys.

A continuación se buscará nuestro PLC y estableceremos conexión con él.

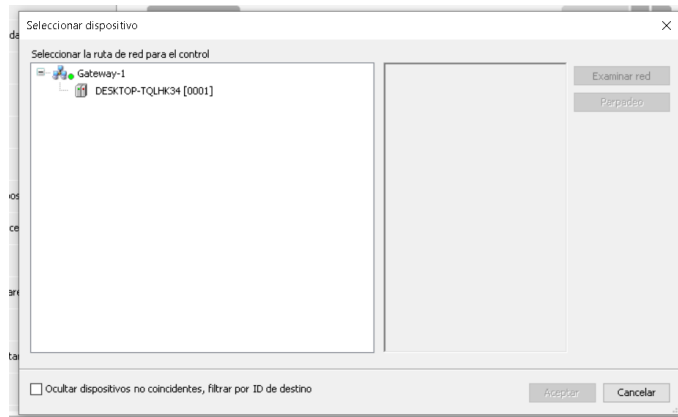


Figura B.3 Búsqueda de Soft PLC para conexión con Codesys.

El siguiente paso será definir las variables que se vayan a compartir con nuestro programa de HMI (que en este caso es Wonderware), para esto se deberá ir al apartado configuración de símbolos y marcar las variables de nuestro programa que queramos comunicar tal y como se aprecia en la siguiente imagen.

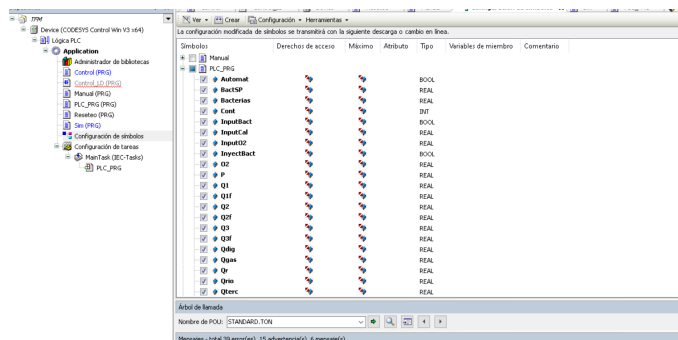


Figura B.4 Variables a compartir con Wonderware.

Una vez configuradas todas las variables que se vayan a comunicar, se tendrá que volver al apartado configuración de comunicación, clicando esta vez en configuración de seguridad del dispositivo, aquí una imagen de como acceder esta opción.

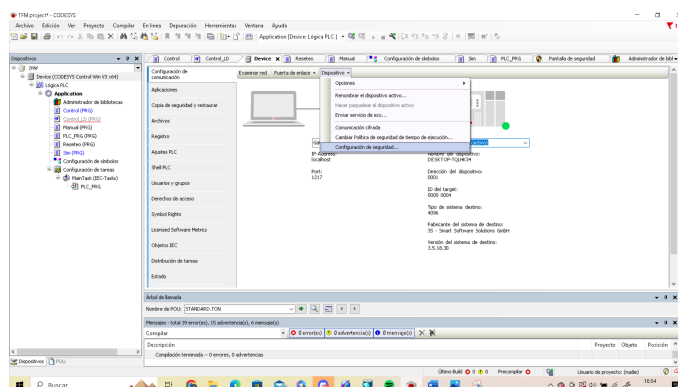


Figura B.5 Configuración de seguridad de la comunicación.

Una vez hecho esto se deberá poner la siguiente configuración, haciendo hincapié en los siguientes puntos:

1. **CommunicationMode:** Este modo definirá si la comunicación esta cifrada o no, en este caso pondremos como opción **ALL** para permitir todo tipo de modos de comunicación.
2. **Activation:** Tendrá que estar activado para que Codesys funcione como servidor de la comunicación mediante OPC UA.
3. **UserMgmtAllowAnonymous:** Este apartado se tendrá que aceptar ya que su función es permitir o negar el acceso de un nodo cliente que se intente conectar con Codesys de forma anonima.

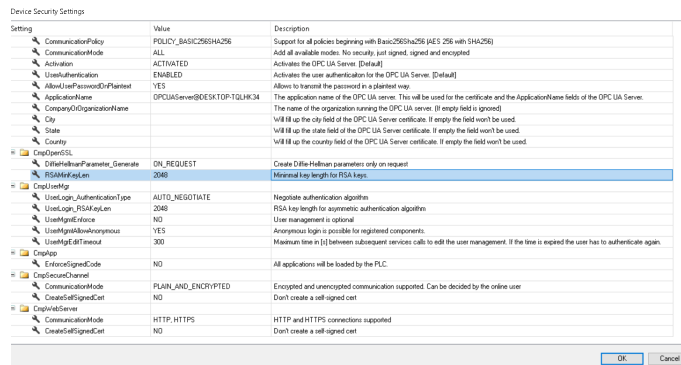


Figura B.6 Configuración usada para la comunicación de este proyecto.

B.1.2 Wonderware

Para la configuración de wonderware deberemos realizar los siguientes pasos. Primero se tendrá que abrir el programa Aveva System Management Console, una vez hecho esto iremos al siguiente apartado para activar la instancia de la puerta de enlace.

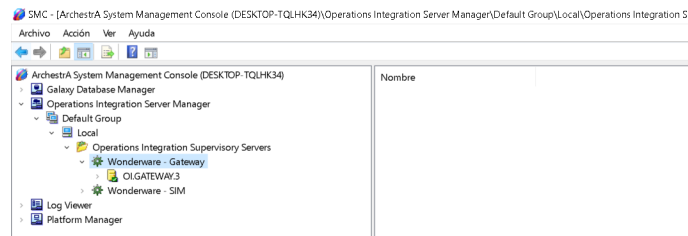


Figura B.7 Aveva SMC.

Dentro de esta instancia tendremos que crear una nueva subinstancia para establecer la conexión mediante protocolo OPC UA tal y como sale en la imagen.

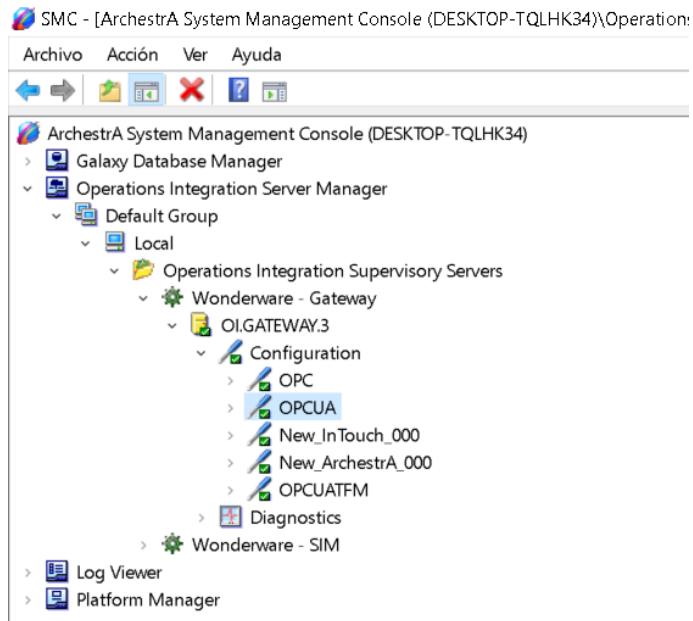


Figura B.8 Creación de instancia para conexión mediante protocolo OPC UA.

Dentro de este apartado se tendrá que rellenar con los siguientes datos que se ven a continuación.

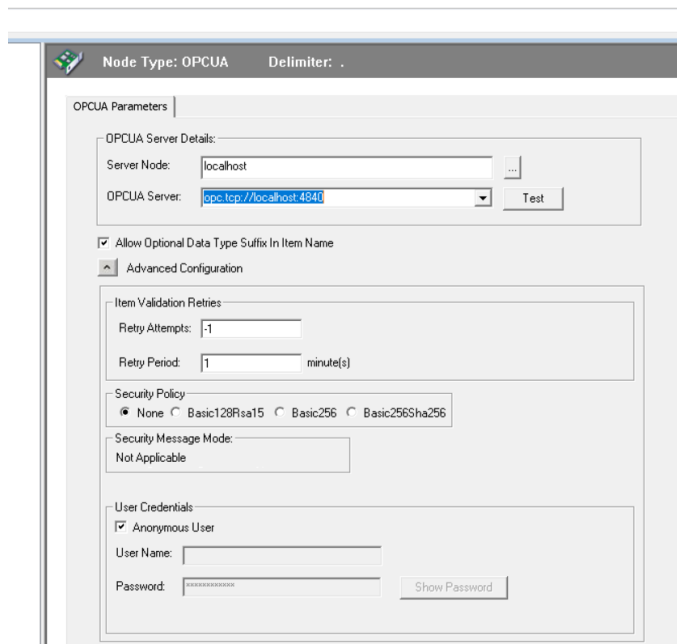
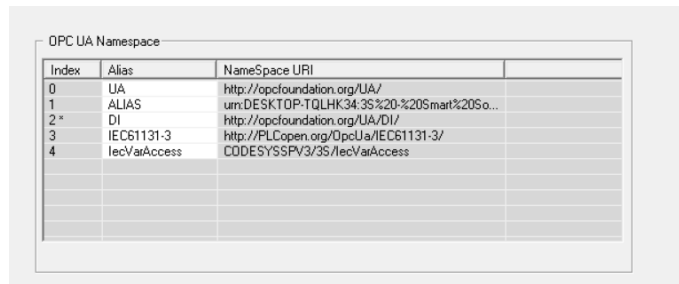


Figura B.9 Configuración con los datos del nodo servidor.

Tenemos que fijarnos en el nombre del servidor (que por defecto será esa dirección en nuestro portátil) y en las credenciales que, al igual que pusimos en la configuración en la comunicación del programa de Codesys, añadiremos un usuario anónimo para la conexión.

Para confirmar esta conexión debemos pulsar en el botón de Test, si esta conexión se realiza con éxito nos aparecerá relleno el apartado de OPC UA Namespaces tal y como aparece en la siguiente imagen.



Index	Alias	NameSpace URI
0	UA	http://opcfoundation.org/UA/
1	ALIAS	urn:DESKTOP-TQLHK34:3S%20-%20Smart%20So...
2*	DI	http://opcfoundation.org/UA/DI/
3	IEC61131-3	http://PLCopen.org/OPcUa/IEC61131-3/
4	IecVarAccess	CODESYSSPV3/3S/IecVarAccess

Figura B.10 Valores obtenidos una vez la conexión se realiza con éxito.

Por ultimo deberemos crear una OPCUAGroup Connection, que simplemente es un nuevo apartado donde añadiremos las variables las cuales queremos comunicar. En este caso hay que añadir que si queremos escribir sobre variables desde nuestro entorno HMI (como es el caso) deberemos confirmar en la casilla Demand Read. Una vez hecho esto clicaremos en el botón Browse OPCUA Server e iremos añadiendo todas nuestras variables.

B.2 Posibles problemas

A continuación se detallará uno de los problemas que se tuvo a la hora de realizar dicha comunicación entre ambos programas. Primero debemos empezar recalando que cuando el programa de AVEVA (Systema Management Console o para abreviar SMC) se conecta mediante protocolo OPC UA como cliente a otro nodo que haga de servidor, el programa SMC envía un certificado el cual tiene que ser aceptado por la parte del nodo servidor para poder establecer dicha comunicación. De forma estandar con otros programas estilo "hubs" tales como Cogent DataHub este certificado es aprobado directamente sin ningun tipo de configuración adicional. Sin embargo, a la hora de realizar una primera conexión con Codesys, funcionando éste como nodo servidor, se observa que dicho certificado se pone en la zona de certificados "Untrusted". Para acceder a tales certificados tendremos que hacer los siguientes pasos.

1. Se deberá acceder a la pantalla de seguridad mediante la barra de herramientas de Codesys, pinchando en Ver y luego seleccionando la opción de Pantalla de seguridad.

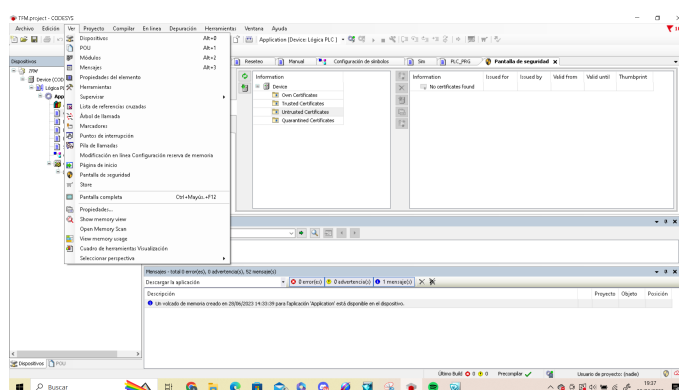


Figura B.11 Acceso a pantalla de seguridad.

2. Seguidamente se accederá a la pestaña "Devices" de ahí tendremos que mover el certificado de Wonderware de la zona de certificados "Untrusted" a "Trusted". Una vez hecho esto nos reconocerá el certificado como uno válido y nos dejará conectarnos sin problemas.

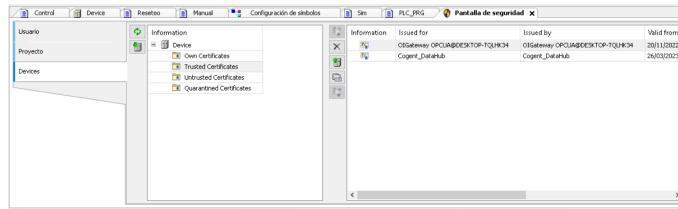


Figura B.12 Zona de certificados.

Apéndice C

Planteamiento

C.1 Código simulación del sistema

Código C.1 Código simulación.

```
// linea de agua
IF Valvula1=TRUE THEN
  V1a:=V1a+Q*(1-X1);
  V1f:=V1f+Q*X1;
END_IF

IF V1a>=0.1 AND Valvula2=TRUE THEN
  V1a:=V1a-Q;
  V2a:=V2a+Q*(1-X2);
  V2f:=V2f+Q*X2;
END_IF
IF V1a<0.1 AND Valvula2=TRUE THEN
  V2a:=V2a+V1a;
  V2f:=V2f+V1a*X2;
  V1a:=0;
END_IF

IF V2a>=0.1 AND ValvulaR=TRUE THEN
  V2a:=V2a-Q;
  Vr:=Vr+Q;
END_IF
IF V2a<0.1 AND ValvulaR=TRUE THEN
  Vr:=Vr+V2a;
  V2a:=0;
END_IF

IF Vr>=0.1 AND Valvula3=TRUE THEN
  Vr:=Vr-Q;
  V3a:=V3a+(1-X3)*Q;
  V3f:=V3f+Q*X3;
END_IF

IF Vr<0.1 AND Valvula3=TRUE THEN
  V3a:=V3a+(1-X3)*Vr;
  V3a:=V3a+X3*Vr;
  Vr:=0;
```

```
END_IF

IF V3a>=0.1 AND ValvulaRio=TRUE THEN
  V3a:=V3a-Q;
END_IF

IF V3a<0.1 AND ValvulaRio=TRUE THEN
  V3a:=0;
END_IF

IF V3a>=0.1 AND ValvulaTerc=TRUE THEN
  V3a:=V3a-Q;
END_IF

IF V3a<0.1 AND ValvulaTerc=TRUE THEN
  V3a:=0;
END_IF

// linea de fango
IF Valvula1f=TRUE AND V1f>=0.1 THEN
  V1f:=V1f-Q;
  Vdig:=Vdig+Q;
  Q1f:=0.1;
ELSIF Valvula1f=TRUE AND V1f<0.1 AND V1f>0 THEN
  Vdig:=Vdig+V1f;
  Q1f:=V1f;
  V1f:=0;
ELSE
  Q1f:=0;
END_IF

IF Valvula2f=TRUE AND V2f>=0.1 THEN
  Vdig:=Vdig+Q;
  V2f:=V2f-Q;
  Q2f:=0.1;
ELSIF Valvula2f=TRUE AND V2f>0 AND V2f<0.1 THEN
  Vdig:=Vdig+V2f;
  Q2f:=V2f;
  V2f:=0;
ELSE
  Q2f:=0;
END_IF

IF V3f>=0.1 AND Valvula3f=TRUE THEN
  V3f:=V3f-Q;
  Vdig:=Vdig+Q;
  Q3f:=0.1;
ELSIF Valvula3f=TRUE AND V3f>0 AND V3f<0.1 THEN
  Vdig:=Vdig+V3f;
  Q3f:=V3f;
  V3f:=0;
ELSE
  Q3f:=0;
END_IF
```



```

IF Vdig>=0.1 AND ValvulaDig=TRUE THEN
  Vdig:=Vdig-Q;
END_IF
IF Vdig<0.1 AND Vdig>0 AND ValvulaDig=TRUE THEN
  Vdig:=0;
END_IF
// oxigeno reactor
IF InyectBact=TRUE THEN
  Bacterias:=Bacterias+landa*((O2-0.1*Bacterias)/((O2)+Kb));

ELSE
  Bacterias:=0;
END_IF

// digestión anaeroba
IF Vdig>0 OR Q1f>0 OR Q2f>0 OR Q3f>0 THEN
  Tdig:=((K1*((Tf*(Q1f+Q2f+Q3f)+Tdig*Vdig)/((Q1f+Q2f+Q3f)+Vdig))+K2*Tcal)/(K1+K
  2));
END_IF
IF Tdig>70 THEN
  Vgas:=Vgas+Kg*Tdig*Vdig;
ELSE
  Vgas:=Vgas;
END_IF

IF ValvulaGas=TRUE AND Vgas>=0.1 THEN
  Vgas:=Vgas-Q;
ELSIF ValvulaGas=TRUE AND Vgas<0.1 THEN
  Vgas:=0;
END_IF

P:= (Vgas/(13-Vdig))*Kp;
//sensores
V1:=V1f+V1a;
IF V1>9 THEN
  S1A:=TRUE;
ELSE
  S1A:=FALSE;
END_IF
IF V1>5 THEN
  S1M:=TRUE;
ELSE
  S1M:=FALSE;
END_IF
IF V1>2 THEN
  S1B:=TRUE;
ELSE
  S1B:=FALSE;
END_IF
V2:=V2f+V2a;
IF V2>9 THEN
  S2A:=TRUE;
ELSE
  S2A:=FALSE;
END_IF
IF V2>5 THEN

```

```
S2M:=TRUE;
ELSE
  S2M:=FALSE;
END_IF
IF V2>2 THEN
  S2B:=TRUE;
ELSE
  S2B:=FALSE;
END_IF
IF Vr>9 THEN
  SRA:=TRUE;
ELSE
  SRA:=FALSE;
END_IF
IF Vr>5 THEN
  SRM:=TRUE;
ELSE
  SRM:=FALSE;
END_IF
IF Vr>0 THEN
  SRB:=TRUE;
ELSE
  SRB:=FALSE;
END_IF
V3:=V3f+V3a;
IF V3>9 THEN
  S3A:=TRUE;
ELSE
  S3A:=FALSE;
END_IF
IF V3>5 THEN
  S3M:=TRUE;
ELSE
  S3M:=FALSE;
END_IF
IF V3>2 THEN
  S3B:=TRUE;
ELSE
  S3B:=FALSE;
END_IF
IF V1a>0 THEN
  SF1:=V1f/V1;
ELSIF V1f>0 THEN
  SF1:=10;
END_IF
IF V2a>0 THEN
  SF2:=V2f/V2;
ELSIF V2f>0 THEN
  SF2:=10;
END_IF
IF V3a>0 THEN
  SF3:=V3f/V3;
ELSIF V3f>0 THEN
  SF3:=10;
END_IF
IF Vdig>9 THEN
```

```
Sdig:=TRUE;
ELSE
  Sdig:=FALSE;
END_IF
IF Vdig>0 THEN
  SdigB:= TRUE;
ELSE
  SdigB:= FALSE;
END_IF
IF Vdig>4 THEN
  SdigM:= TRUE;
ELSE
  SdigM:= FALSE;
END_IF
```

C.2 Modo Reseteo código

Código C.2 Código modo Reset.

```
// Control agua valvula 1
Valvula1:=FALSE;

// Control agua valvula 2
Valvula2:=TRUE;

// Control agua valvula reactor
ValvulaR:=TRUE;

// Control agua valvula fango
Valvula1f:=TRUE;
Valvula2f:=TRUE;
Valvula3f:=TRUE;

// Control agua valvula 3
Valvula3:=TRUE;

// Control agua valvula rio
Valvulario:=FALSE;

// Control agua valvula terc
Valvulaterc:=TRUE;

// Control agua valvula dig
Valvuladig:=TRUE;

// Control agua valvula gas
Valvulagas:=TRUE;

//Fin de reset
IF SdigB=FALSE AND S1B=FALSE AND S2B=FALSE AND S3B=FALSE AND SRB=FALSE AND P<1
  THEN

  Reset:=TRUE;
END_IF
```

C.3 Modo Manual código

Código C.3 Código modo Manual.

```
// Control agua valvula 1
Valvula1:=Valvula1M;

// Control agua valvula 2
Valvula2:=Valvula2M;

// Control agua valvula reactor
ValvulaR:=ValvulaRM;

//Inyectar bacterias
IF InputBact=TRUE THEN
  InjectBact:=TRUE;
ELSE
  InjectBact:=FALSE;
END_IF
O2:=InputO2;

// Control agua valvula 3
Valvula3:=Valvula3M;

//Control agua valvula rio
ValvulaRio:=ValvulaRioM;

//Control agua valvula terciario
ValvulaTerc:=ValvulaTercM;

//Control fango valvulas
Valvula1f:=Valvula1fM;
Valvula2f:=Valvula2fM;
Valvula3f:=Valvula3fM;

//Control valvula gas
Valvulagas:=ValvulagasM;

//Control temperatura dig
IF InputCal>25 THEN
  Tcal:=InputCal;
ELSE
  Tcal:=25;
END_IF

//Control valvula dig
ValvulaDig:=ValvulaDigM;
```

Índice de Figuras

2.1.	Ejemplo de EDAR	3
2.2.	Ejemplo específico de EDAR	3
2.3.	Representación de sistema a controlar	4
3.1.	Válvulas del sistema	10
3.2.	Control válvula 1	10
3.3.	Control válvula 2	10
3.4.	Control válvula Reactor	11
3.5.	Control inyección bacterias	11
3.6.	Control válvula 1	12
3.7.	Control válvula fango tanque 1	12
3.8.	Control válvulas fango	12
3.9.	Control válvula río y terciario	12
3.10.	Control válvula gas	12
3.11.	Control válvula dig	13
3.12.	Representación respuesta sistema temperatura a escalón	13
3.13.	Fórmula de Ziegler-Nichols	13
3.14.	Controlador con setpoint en 100 ^o temperatura 1a iteración	14
3.15.	Controlador con setpoint en 100 ^o temperatura 2a iteración	14
3.16.	Controlador con setpoint en 100 ^o temperatura 3a iteración	15
3.17.	Control temperatura fango dentro del tanque de digestión	16
3.18.	Representación respuesta sistema bacterias a escalón	16
3.19.	Controlador con setpoint en 3mg bacterias 1a iteración	17
3.20.	Controlador con setpoint en 3mg bacterias 2a iteración	18
4.1.	Herramientas usadas en Intouch	19
4.2.	TagName Dictionary	20
4.3.	Configuración variable	20
4.4.	Grupo de acceso por donde se comunicará la variable	21
4.5.	Detalles configuración variable	21
4.6.	Ventana principal del HMI	22
4.7.	Ventana principal por partes	22
4.8.	Configuración elemento tanque	23
4.9.	Configuración válvulas	23
4.10.	Configuración botones	24
4.11.	Configuración acción de botón	24
4.13.	Configuración entradas elemento Historical Tren Chard	25
4.14.	Configuración TrendPanel	25
4.15.	Configuración TrendLegend	26
4.16.	Creación de nuevo Alarm Group	26
4.17.	Script usado para activación/desactivación de modos de control	27

4.18.	Ventana de modos de control	27
A.1.	Instalación programa AVEVA System Platform	31
A.2.	Instalación por producto	32
A.3.	Interfaz licencia	32
A.4.	Interfaz licencia	32
A.5.	Configurador AVEVA	33
A.6.	Problema durante instalación	33
A.7.	Panel de Control	34
A.8.	Actualizaciones de programas	34
B.1.	Codesys Control Win Systray	35
B.2.	Configuración comunicación Codesys	35
B.3.	Busqueda de Soft PLC para conexión con Codesys	36
B.4.	Variables a compartir con Wonderware	36
B.5.	Configuración de seguridad de la comunicación	36
B.6.	Configuración usada para la comunicación de este proyecto	37
B.7.	Aveva SMC	37
B.8.	Creación de instancia para conexión mediante protocolo OPC UA	38
B.9.	Configuración con los datos del nodo servidor	38
B.10.	Valores obtenidos una vez la conexión se realiza con éxito	39
B.11.	Acceso a pantalla de seguridad	39
B.12.	Zona de certificados	40

Índice de Códigos

C.1.	Código simulación	41
C.2.	Código modo Reset	45
C.3.	Código modo Manual	46

Bibliografía

- [1] Picuino, *Método de ziegler-nichols*, última consulta Julio de 2023. <https://www.picuino.com/es/control-ziegler-nichols.html>.
- [2] Tecpa, *Control y sensores en una edar*, última consulta Julio de 2023. <https://www.tecpa.es/control-del-proceso-en-una-edar/>.
- [3] _____, *Tratamientos y procesos que se dan en una edar*, última consulta Julio de 2023. <https://www.tecpa.es/tratamientos-procesos-depuracion-aguas-residuales/>.
- [4] Wikipedia, *Ecuación de monod, evolución de microorganismos*, última consulta Julio de 2023. https://es.wikipedia.org/wiki/Ecuación_de_Monod.