Universidad de Valladolid

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

Interfaz gráfica de aplicaciones de control desarrolladas con Python

Autor:

Hernández Sánchez, Sergio

Tutores:

Acebes Arconada, Luis Felipe
Pablos de la Fuente, Cristian
Departamento de Ingeniería de Sistemas y Automática

Valladolid, Julio 2019.

RESUMEN, PALABRAS CLAVE

La revolución industrial 4.0 tiene entre sus bases la optimización de la operación de procesos industriales, permitiendo así desarrollar herramientas de cálculo potentes y eficientes en estos procesos. Sin embargo, esta revolución no es un indicativo fiable de su presencia en la cotidianidad de los entornos industriales, en muchas ocasiones por falta de conocimientos de utilización de los operarios. Por esto el presente TFG, propone el diseño de interfaz gráfica que permita la comunicación entre un algoritmo desarrollado en Python, que busca optimizar la simulación de una planta azucarera para mejorar su competitividad en el mercado eléctrico español. La interfaz gráfica que comunica al operario y la optimización, ha sido realizada a través del software industrial Wonderware, mediante el protocolo de comunicaciones OPC UA. Como resultado, gracias al trabajo realizado cualquier operario sería capaz de lanzar optimizaciones sin necesitar conocimiento alguno de programación.

Palabras clave: HMI, Comunicaciones Industriales, Simulación, Optimización, Cogeneración.

ABSTRACT, KEYWORDS

The industrial revolution 4.0 has among its bases the optimization of the operation of industrial processes, allowing to develop powerful and efficient calculation tools in these processes. However, this revolution is not a reliable indicator of its presence in the daily life of industrial environments, often due to lack of knowledge of the use of operators. This is why the present TFG proposes the graphic interface design that allows the communication between an algorithm developed in Python, which seeks to optimize the simulation of a sugar plant to improve its competitiveness in the Spanish electricity market. The graphical interface that communicates to the operator and the optimization, has been realized through the industrial software Wonderware, by means of the protocol of communications OPC UA. As a result, thanks to the work done, any operator would be able to launch optimizations without needing any knowledge of programming.

Keywords: HMI, Industrial Comunications, Simulation, Optimization, Cogeneration.

Índice

INTRODUCCIÓN

La llegada de la industria 4.0 a las empresas ha supuesto un aumento de su competitividad a través de la adaptación a los nuevos cambios tecnológicos, puesto que esta revolución incorpora tecnologías avanzadas de producción, que permiten a las fábricas mejorar su proceso y hacerlo más eficiente. La incorporación de estas tecnologías engloba varias herramientas clave para integrar en las plantas de producción, entre las que se encuentra la optimización de procesos.

Gracias a las técnicas de optimización, la necesidad latente de las empresas de mejorar su eficiencia, puede ser resuelta, puesto que puede optar a aumentar sus beneficios económicos, así como ser más responsables a nivel medioambiental.

Sin embargo, su implementación en las industrias es escasa debido en muchas ocasiones, a la complejidad en la utilización de estas herramientas, puesto que muchas veces requieren de conocimientos de programación. Por esto, se debe perseguir facilitar el acercamiento de los operarios de las industrias a estas herramientas, beneficiándose así tanto las empresas como sus empleados, buscando sobrepasar el obstáculo que supone la falta de conocimientos de programación.

El presente Trabajo de Fin de Grado, a partir de ahora TFG, pretende presentar una propuesta gráfica utilizando el software Wonderware™ de una aplicación, que permita gestionar conjuntamente la producción de una planta azucarera y la generación de electricidad de un sistema de cogeneración asociado, con el fin de que se aumente la competitividad de la planta, vendiendo electricidad en el mercado diario eléctrico español. La gestión del excedente eléctrico de la industria azucarera en la que se centra este caso de estudio, tiene en cuenta le legislación de regulación del mercado, la ley de oferta y demanda, así como la incertidumbre del precio eléctrico.

Para testear, la interfaz gráfica propuesta, se utilizará una herramienta de simulación, relacionada con el proceso de la industria azucarera, y el modelo de optimización, que se vincula con la planta del sistema de cogeneración. La herramienta de simulación implementa una capa de comunicaciones con el protocolo OPC UA que será el utilizado para establecer la comunicación con la interfaz gráfica, mientras que la optimización no lleva ninguna capa de comunicaciones, por lo tanto se deberá realizar esta implementación.

La estructura seguida en este TFG ha sido la siguiente: primeramente, se presentan una serie de conceptos clave para entender el funcionamiento de la interfaz desarrollada. A continuación, se explica en detalle el caso de estudio en el que se enmarca este trabajo. Después, se expone la arquitectura de la interfaz desarrollada, así como una explicación de diversos

retos que se han ido superando. Posteriormente, se muestran los resultados de esta propuesta enseñando la secuencia de pasos que debe seguir el operario para ejecutar una optimización de manera correcta. También, se presenta una serie de pruebas que verifica el funcionamiento de la aplicación realizada, contemplando posibles situaciones a las que puede enfrentarse el usuario en interacción con la aplicación.

En su apartado final, se explican las conclusiones a las que se ha llegado a lo largo del desarrollo de este trabajo.

Así, este TFG se divide en dos partes, por un lado la conceptualización teórica de la propuesta, y por otro lado la ejecución práctica, que puede observarse en los Anexos que acompañan al documento, derivados del trabajo de campo y que complementan todo lo expuesto.

OBJETIVOS

El presente TFG tiene como objetivo principal diseñar la interfaz gráfica hombre-máquina (HMI) para comunicar las herramientas que han sido impuestas para su desarrollo, es decir, el proceso de simulación de una industria azucarera y la herramienta encargada de optimizar el sistema de cogeneración asociado a dicha industria. Este objetivo general se desglosa en objetivos específicos con el fin de facilitar su consecución en el proceso de diseño e implementación, teniendo en cuenta los requisitos establecidos por la tutorización de este TFG, siendo estos el protocolo OPC UA, para la comunicación entre las herramientas, y el software Wonderware System Platform R2 SP1, para el diseño gráfico de la interfaz. Los objetivos específicos son:

1. Conocer la comunicación OPC y su variante OPC UA (*Unified Architecture*).

2. Verificar la comunicación del servidor de la simulación y de la optimización con el software Wonderware System Platform R2 SP1.

3. Mostrar y editar los valores más relevantes del proceso de la simulación.

4. Manipular y controlar la ejecución de la simulación y optimización.

5. Crear un servidor utilizando el lenguaje de programación Python, que encapsule el trabajo del optimizador.

DESARROLLO DEL TFG

CAPÍTULO 1

Conceptos previos

1.1 COMUNICACIÓN OPC CLASSIC

La tecnología para la automatización y la comunicación industrial de los diversos componentes que forman las Redes de Control de Procesos se caracteriza por la interconexión física entre módulos de entrada/salida y lógicas cableadas. El desarrollo de los buses de campo permite aumentar la capacidad de comunicación y de conexión entre los diferentes elementos de campo, como pueden ser sensores, actuadores, controladores…

La necesidad de esta comunicación entre los diferentes dispositivos lleva a la industria a desarrollar un estándar de comunicaciones que garantice el funcionamiento en cualquier dispositivo utilizado para la automatización industrial. Para facilitar esta tarea de comunicación entre dispositivos, se ha desarrollado un protocolo de comunicaciones OPC (*OLE for Procces Control),* [1]. Este protocolo está basado en la tecnología OLE de Microsoft y después ha sido optimizada por la Fundación OPC.

Es una tecnología de protocolos de alto nivel, que se utiliza para poder desarrollar aplicaciones software compatibles entre sistemas distribuidos orientados al control y monitorización de procesos. Con estos protocolos se logra una comunicación abierta, independiente y estandarizada.

Se puede conseguir una integración total, ya que facilita que equipos de diferentes fabricantes puedan trabajar sin que se presente ningún tipo de conflicto entre ellos y facilitando que cada empresa pueda realizar su propia aplicación de control y medición adecuada a sus necesidades.

Para realizar esta comunicación se han definido una serie de interfaces de comunicación que se encargan de recoger las necesidades principales de la industria, estableciendo tres especificaciones básicas que OPC aborda como se muestra en la Tabla 1:

COMUNICACIÓN OPC	
Requerimientos de la Industria	*OPC Foundation*
Datos de Tiempo Real	OPC Data Access Specification (OPC DA)
Datos Históricos	OPC Historical Data Access Specification (OPC HAD)
Datos de Alarmas y Eventos	OPC Alarms & Events Specification (OPC A&E)

Tabla 1. Comunicación OPC

Estas interfaces han sido diseñadas con el objetivo de proporcionar un mecanismo de intercambio de información industrial estandarizado que permita, conocer en tiempo real los datos de los dispositivos, el estado de éstos mediante señales de alarma y la creación de una base de datos que permita almacenar los datos como históricos del proceso.

A continuación, se definen brevemente las interfaces de comunicación principales del protocolo de comunicaciones OPC:

- *OPC Data Access Specification*

  La interfaz OPC DA proporciona la lectura, escritura y monitorización de variables del proceso en tiempo real. El principal uso de esta interfaz es captar los datos proporcionados por PLCs, DCSs, y otros dispositivos de control.

  Para establecer dicha comunicación se tiene que crear un cliente OPC DA que permita seleccionar las variables que se deseen leer, escribir o monitorizar del servidor. Para detectar errores en la transmisión de datos en tiempo real, OPC DA cuenta con garantías de tiempo y calidad en los datos entregados, ya que si la conexión se interrumpe contempla tres tipos de calidad en los datos entregados:
  - Datos Precisos: *Quality Good*.
  - Datos No Disponibles: *Quality Bad*.
  - Datos Desconocidos: *Quality Uncertain*
- *OPC Historical Data Access Specification*

  OPC DAH proporciona acceso a los datos que hayan sido almacenados en un servidor de datos Históricos. Su principal funcionalidad es la lectura de dichos datos de tres formas diferentes:
  - Lectura a través de un fichero de datos.
  - Lectura mediante variables en un periodo definido.
  - Lectura mediante una base de Datos de Históricos.

  Para realizar la comunicación se debe implementar un cliente de Históricos, que conecte con la fuente de información.
- *OPC Alarms & Events Specification*

  La interfaz OPC A&E habilita la recepción de notificación de eventos y alarmas. Estos eventos y alarmas están asociados a diferentes situaciones de emergencia que puedan ocurrir durante un proceso, la información se transmite en forma de notificaciones de estado o mensajes de confirmación y respuesta.

  Para realizar la comunicación se debe crear un cliente de Eventos y Alarmas que conecte con los dispositivos de campo que transmitan dichas señales.

Con estas interfaces que proporciona OPC se puede conseguir una comunicación fiable y segura entre los diferentes elementos de campo y los

dispositivos que actúan como aplicaciones de usuario (HMI/SCADA), planteando una situación como la que se describe en la Figura 1:

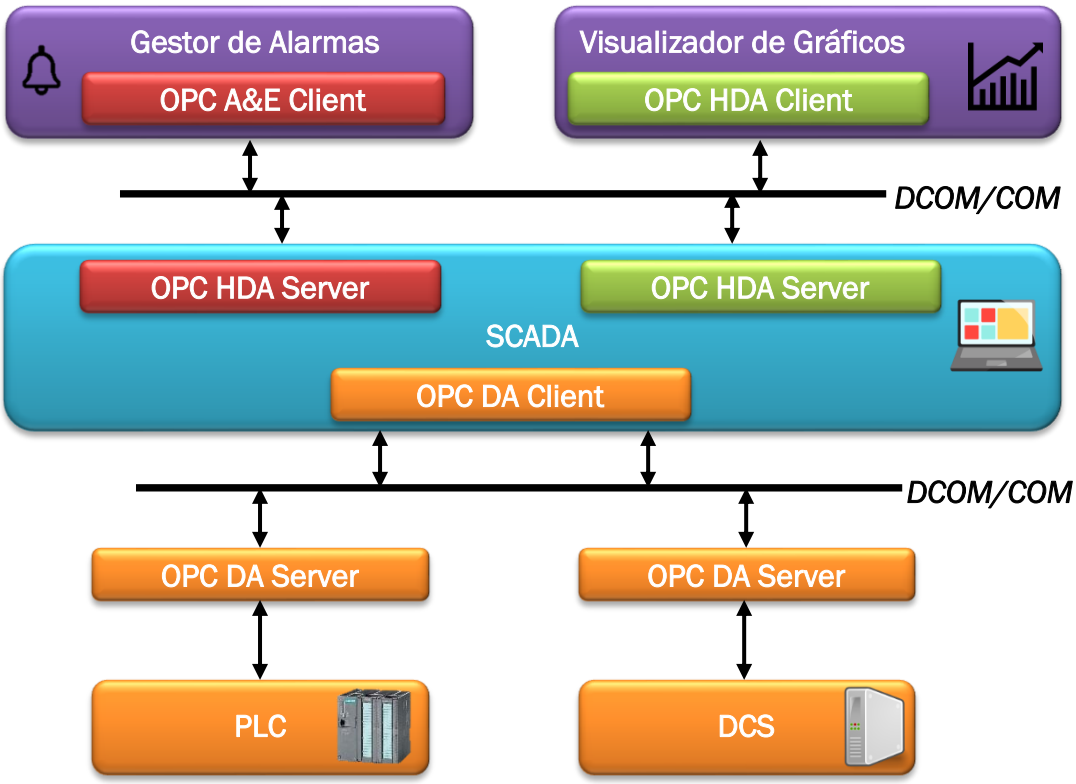


Figura 1. Situación de Comunicación OPC Classic

Como se puede observar en el diagrama, la arquitectura cliente/servidor está presente para la realización de la comunicación entre los diferentes elementos que formen el sistema. El cliente se encarga de solicitar los recursos requeridos por el usuario de la aplicación, mientras que el servidor realizará las acciones necesarias para satisfacer la demanda del cliente [2], [3], [4].

1.2. COMUNICACIÓN OPC UNIFIED ARCHITECTURE

Hoy en día se utiliza la comunicación OPC como interfaz estandarizada entre sistemas de automatización en diferentes niveles de la pirámide de automatización, incluso se usa en muchas áreas para las que no fue diseñado, y hay muchos fabricantes que desean integrar el estándar en sus dispositivos, pero no pueden usarlo debido a la dependencia con el puerto COM o las limitaciones para el acceso remoto utilizando DCOM.

También se detectaron problemas de interoperabilidad entre la interfaz OPC XML DA y los diferentes servicios web XML, además las compañías miembros de OPC *Foundation* presentaron el requisito de exponer datos y sistemas complejos, eliminando las limitaciones del OPC Clásico.

Para solventar estos requisitos se creó OPC *Unified Architecture*, que pretende reemplazar todas las especificaciones existentes basadas en COM sin perder ninguna característica o rendimiento. Además, se encarga de cubrir todos los requisitos para interfaces de sistemas independientes y del modelado que puedan describir sistemas complejos. Los requisitos más importantes que debe cumplir OPC UA se recogen en la Tabla 2:

COMUNICACIÓN OPC	
Comunicación entre Sistemas Distribuidos	*Datos de Modelado*
Fiabilidad para: ▪ Robustez y tolerancia a fallos. ▪ Redundancia	Modelo común para todos los datos OPC.
Escalabilidad.	Orientado a Objetos.
Alto rendimiento.	Sistema de tipo extensible
Seguridad y Control de Acceso.	Datos y Métodos Complejos
Interoperabilidad	Base para otros modelos de datos Estándar.

Tabla 2. Requerimientos para el estándar OPC Unified Architecture (UA)

Los componentes principales de OPC UA son los mecanismos de transporte de información y el modelado de datos. El transporte define diferentes mecanismos optimizados para diferentes casos, según su requerimiento. Se define un protocolo TCP optimizado para comunicaciones de alto rendimiento, así como un mapeo de los estándares de Internet aceptados, como servicios web, XML y HTTP para una comunicación de Internet compatible con los firewall. Ambos mecanismos de transporte de información utilizan el mismo modelo de seguridad basado en mensajes, que se utiliza en los servicios web. El modelo de comunicación abstracta, no depende de un mapeo de protocolo específico, esto permite agregar nuevos protocolos en el futuro.

Para el modelado de datos se definen unas reglas y una creación de bloques necesarios para exponer un modelo de información utilizando OPC UA. También define los puertos de entrada en el espacio de direcciones y los tipos de datos utilizados para crear una jerarquía de estos. Además, define algunos conceptos mejorados, como describir máquinas de estado (utilizadas en diferentes modelos de información).

Los servicios que proporciona este protocolo, es una interfaz de comunicación entre los servidores como proveedores de un modelo de información y los clientes como consumidores de ese modelo de información.

Para cubrir todas las características de la comunicación OPC Clásica, se diseñó una arquitectura de comunicación como muestra la Figura 2:

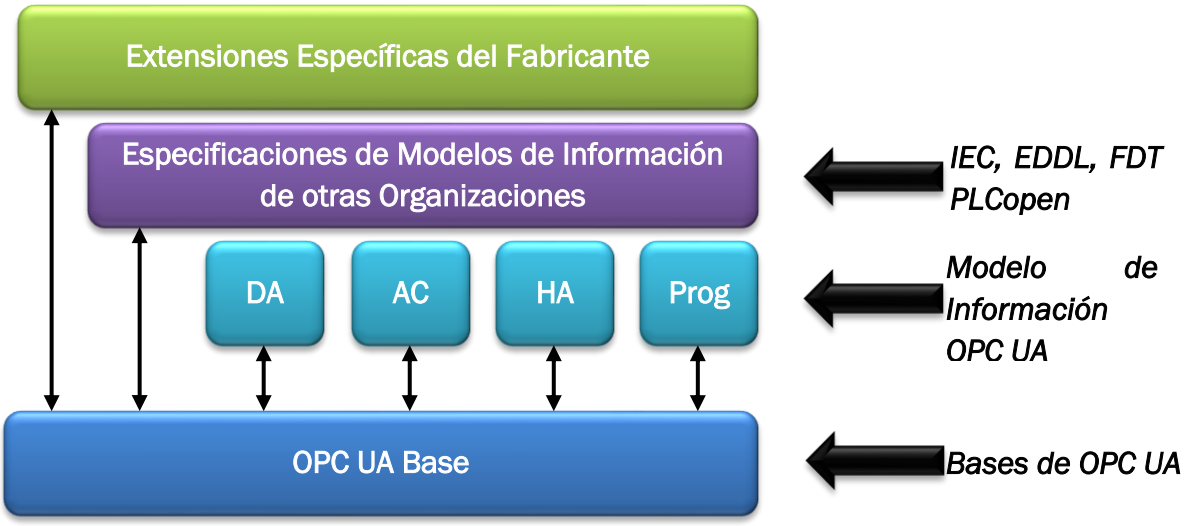


Figura 2. Arquitectura OPC UA

- *Data Automation (DA)*
  Define extensiones específicas de datos de automatización, como el modelado de datos discretos o analógicos, además de comprobar la calidad del servicio (*Quality*). El resto de las características DA están implementadas en la base.
- *Alarm & Condition (AC)*
  Especifica un modelo avanzado para gestión de alarmas de un proceso y la monitorización de condiciones de estado.
- *Historical Access (HA)*
  Define los mecanismos para acceder a los datos y eventos históricos.
- *Programs (Prog)*
  Especifica un mecanismo para iniciar, manipular y monitorizar la ejecución de programas.

Cabe destacar respecto a las mejoras en la comunicación, que OPC UA es capaz de realizar la comunicación de un modelo que emplee una gran cantidad de variables, mientras que OPC *Classic* tiene algunas limitaciones para transmitir un número elevado de variables, para obtener más información sobre la comunicación OPC UA, consultar [5], [6].

1.3. SCADA - WONDERWARE SYSTEM PLATFORM

Un SCADA es un sistema para la supervisión, control y adquisición de datos de una planta o proceso, utilizando las herramientas de comunicación correspondientes en cada caso [7]. La comunicación se realiza mediante una estación central (MTU *Master Terminal Unit)*, que será la encargada de enviar las órdenes correspondientes a una o varias unidades remotas (RTUs, *Receptor Terminal Unit)* para que éstas realicen el control y la adquisición de datos demandada por la estación central. El nombre SCADA proviene de las siglas *Supervisory Control And Data Acquisition* de las cuales se puede obtener las siguientes características que definen este sistema:

- *Adquisición de Datos*
  El sistema accede a los datos que proporcionan los elementos de campo en tiempo real. Se utilizan protocolos de comunicación fiables que garanticen que la información transmitida es correcta, tanto para datos analógicos como para datos discretos.
- *Control Supervisado*
  Proporciona información del proceso al usuario encargado de la estación central, para poder monitorizar el proceso y llevar a cabo las acciones de control que sean necesarias. Este tipo de control incluye al operario para la decisión final sobre el control del sistema o de la planta industrial.

Es importante señalar que el control directo se realiza a través de los controladores digitales o autómatas programables que, al estar conectados a la estación central, actúan como diálogo con el operador. El lugar que ocupa en la pirámide de automatización se muestra en la Figura 3:

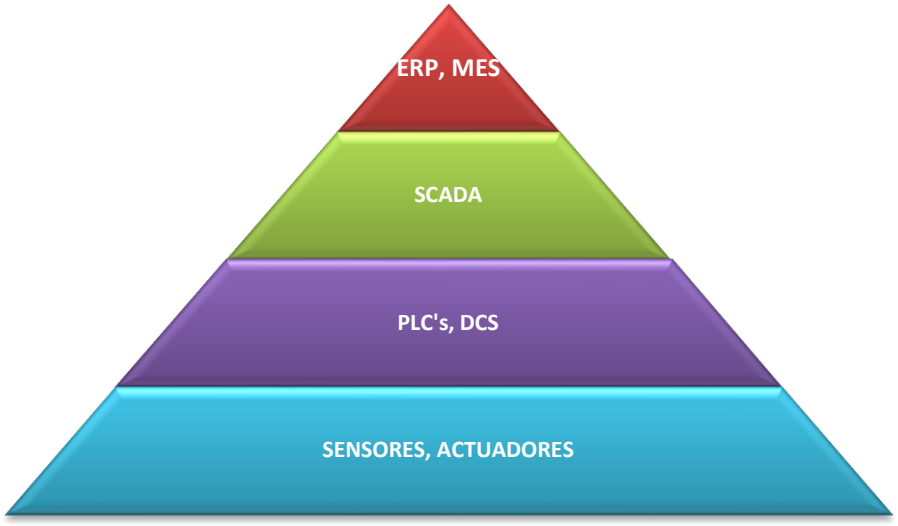


Figura 3. Pirámide de Automatización

En esta pirámide se pueden asociar los niveles según la función que desempeñan. Por ello se pueden diferenciar 4 niveles atendiendo a sus funcionalidades principales dentro de la empresa:

- *Nivel de Gestión y Planificación (ERP y MES)*
  En este nivel se recoge la planificación de los recursos empresariales utilizando un sistema de Planificación de Recursos Empresariales ERP, (*Enterprise Resource Planning*), que se encargará de la planificación de compras, ventas, logística... Mientras que el Sistema de Ejecución de la Fabricación MES, (*Manufacturing Executing System*) se encarga de la gestión de la producción, donde realiza funciones de mantenimiento, documentación, optimización...
- *Nivel de Supervisión (SCADA)*
  Este nivel acoge la definición de un sistema SCADA, que permite interaccionar entre los niveles superiores e inferiores entre los que se encuentra en la pirámide. Su principal función es, conocer en todo momento el estado del proceso para poder tomar las decisiones adecuadas y garantizar unos requisitos de producción en la empresa.
- *Nivel de Control (PLCs y DCS)*
  En este nivel se lleva a cabo la tarea de control, donde se establecen los dispositivos necesarios para que el proceso tenga un cierto grado de automatización. Este nivel actúa de intermediario entre el SCADA y los elementos de campo del proceso.
- *Nivel de Campo (Sensores y Actuadores)*
  A nivel de campo se tienen los sensores que enviarán señales de información a los controladores y actuadores que se encargaran de recibir las acciones demandadas por los controladores. Éstos deben ser capaces de comunicarse con el nivel de control, para realizar la supervisión de las partes más relevantes del proceso.

La arquitectura que implementan la mayoría de estos sistemas se corresponde con la Figura 4:

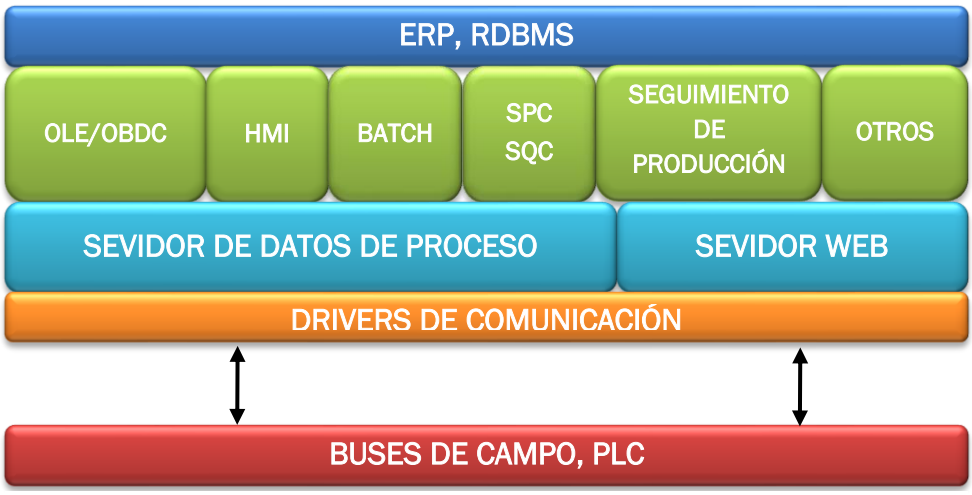


Figura 4. Arquitectura General de un Sistema SCADA

Para implementar un sistema SCADA generalmente se recurre a un paquete de software especializado. Este software permitirá diseñar los sinópticos que actúan como una interfaz gráfica entre el operario y el proceso, de esta forma es posible realizar las acciones anteriormente descritas. Para este TFG se ha utilizado el software industrial Wonderware System Platform 2014 R2.

Wonderware System Platform 2014 R2 es una marca de software industrial dedicado a la gestión de operaciones en tiempo real y la gestión de infraestructuras, utilizando la programación orientada a objetos. Wonderware está basado en la arquitectura ArchestrA creada a partir de la tecnología .NET de Microsoft y desarrollada para facilitar e impulsar la integración de dispositivos industriales y sistemas a distintos niveles, para obtener más información sobre el entorno Wonderware, consultar [8].

Los productos software que engloban la tecnología ArchestrA se pueden agrupar en función del flujo de comunicación de datos mediante programas a nivel de servidor y programas a nivel de cliente:

- *Software a nivel de Servidor:*
  El producto System Platform que incluye Wonderware es una base para el diseño y gestión de modelos industriales que incluye:
  - Application Server
    Es el motor de ejecución de la aplicación, se encarga de gestionar y configurar los programas que se definen a continuación:
    - **Wonderware Historian:** Es una base de datos de proceso de alto rendimiento que recupera toda la información operacional para una fácil entrega en su tratamiento.
    - **Wonderware Information Server:** Se encarga de la gestión y administración de servicios Web.
    - **Data Adquisition Server:** Es un programa que integra drivers de comunicación con dispositivos de campo utilizando estándares industriales.

- *Software a nivel de Cliente*
  - InTouch
    Es un sistema interactivo diseñado para la visualización, la supervisión y el control de procesos industriales.
  - System Manage Console
    Es un programa para la gestión y comunicación entre diferentes dispositivos en el entorno de control de procesos.
  - Integración con MS Office
    Permite la gestión de documentos mediante los paquetes de Microsoft Word y Microsoft Excel de Office.

Para comprender mejor la gestión de la comunicación y de los programas que se tendrán que utilizar para el diseño del HMI, se ha elaborado un diagrama de flujo con la arquitectura que utiliza Wonderware en sus productos:

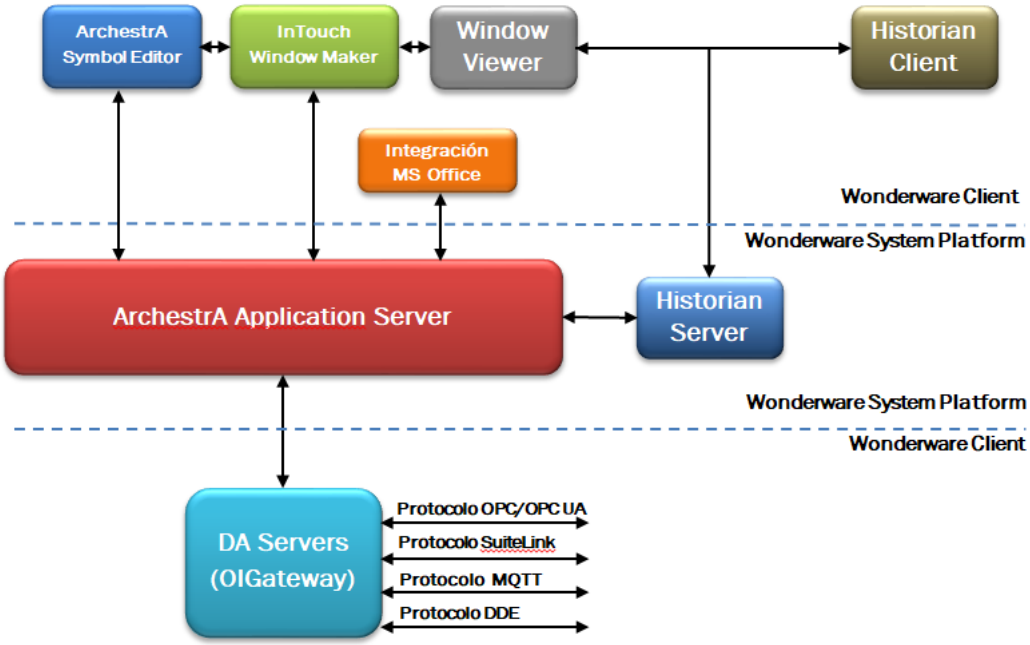


Figura 5. Arquitectura de Comunicación Wonderware System Platform R2 2014 SP1

Con estos productos Wonderware, bajo el entorno ArchestrA, es capaz de proporcionar un marco de servicios unificado en cuanto a gestión y administración de plantas industriales se refiere.

Comprendiendo la arquitectura de comunicación que utiliza Wonderware, se puede establecer un diagrama de comunicaciones que resuma los elementos necesarios para llevar a cabo el diseño de la aplicación HMI:

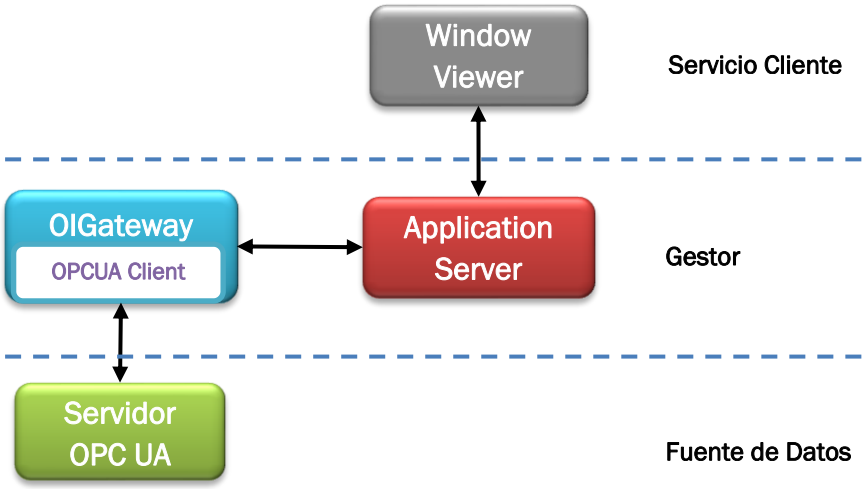


Figura 6. Situación Inicial de Comunicación Wonderware - OPC UA

1.4. CONEXIÓN CLIENTE-SERVIDOR OPC UA

El protocolo de comunicaciones OPC UA utiliza la arquitectura cliente/servidor subyacente para transmitir las distintas categorías de datos. Para una comunicación correcta y de forma coordinada, se debe incluir la misma especificación OPC UA tanto en el cliente como en el servidor.

- *Servidor*

  Un servidor OPC UA es una aplicación software desarrollado específicamente para cumplir con una o más funciones específicas para la comunicación con los elementos de campo.

  Son conectores que se pueden asimilar a traductores entre el mundo OPC UA y los protocolos nativos de una fuente de datos, la relación cliente OPC UA/servidor OPC UA es de tipo maestro/esclavo, es necesario que el cliente OPC pida los datos al servidor para que éste los proporcione.

  Estos servidores se pueden comunicar prácticamente con cualquier elemento de campo, cuyos datos puedan ser leídos o escritos por dispositivos electrónicos como, PLCs, DCSs, RTUs, instrumentos de medición... Para comunicarse con cualquiera de estos dispositivos se requiere el uso de un servidor OPC UA que utilice el protocolo o interfaz nativo apropiado. Si la configuración del servidor OPC UA se ha realizado correctamente, cualquier aplicación cliente OPC UA podrá empezar a comunicarse con éste.

  Para que esta comunicación se establezca correctamente, el servidor define un puerto de comunicación utilizando TCP/IP y una política de seguridad para la transmisión de información.

- *Cliente*

  Un cliente OPC UA es una aplicación software creada para comunicarse con servidores OPC UA. Para establecer la comunicación utiliza la mensajería definida por una especificación concreta, ya sea datos en tiempo real, datos históricos, datos de alarmas y eventos... definidas en el estándar de la OPC Foundation.

  El cliente OPC UA se utiliza como destino de datos, puede iniciar y controlar la comunicación establecida con el servidor OPC UA. La función principal que realiza el cliente es, crear la petición de comunicación de los datos que el usuario haya solicitado, para poder enviarla al servidor.

  Por ser un protocolo estandarizado, el cliente es capaz de conectarse a diferentes servidores OPC UA y gestionar su comunicación independientemente.

  Para que la comunicación sea correcta se debe configurar el cliente con el puerto de comunicación por el que transmite el servidor y configurar la política de seguridad.

Para administrar la comunicación cliente/servidor OPC UA, existen softwares que actúan como un cliente OPC UA, facilitando la gestión de la comunicación.

En este TFG se han utilizado los siguientes programas:

- *dataFEED OPC UA Client - Softing* [9]
- *UaExpert - Unified Automation* [10]
- *OPC UA TOOLBOX - Matlab 2017b* [11]
- *Object Viewer - Wonderware System Platform R2 2014 SP1* [12]

La utilización de estos clientes es sencilla e intuitiva conociendo los parámetros principales que define este tipo de protocolo. Para visualizar y comprender como administra la comunicación el cliente OPC UA y el interior de un servidor OPC UA, se presenta como ejemplo la conexión de un cliente desarrollado con Softing con un servidor OPC UA de prueba, proporcionado por el mismo software. Para que la conexión sea correcta, el servidor debe estar en ejecución y esperando las conexiones entrantes por un puerto definido en su implementación, la ventana de configuración que corresponde a esta situación se muestra en la Figura 7:

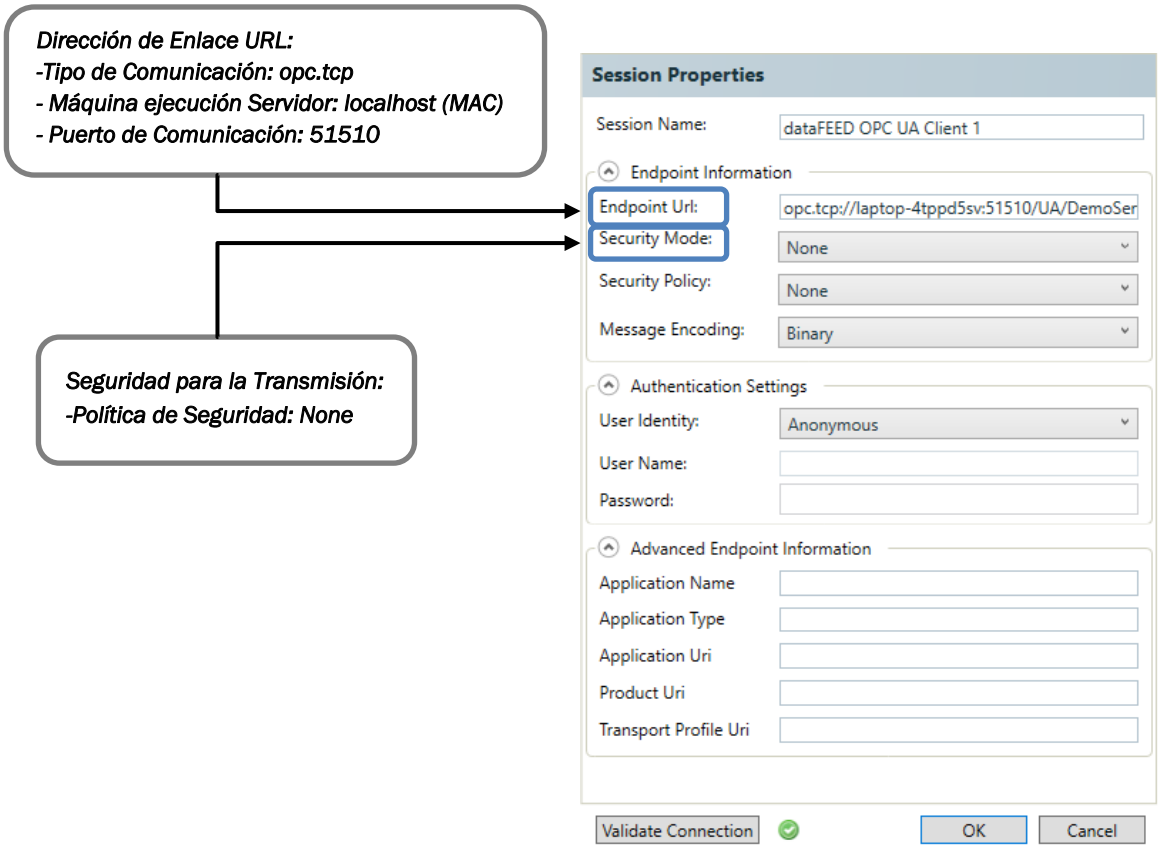


Figura 7. Configuración Inicial del Cliente para la Comunicación Cliente/Servidor OPC UA

Cuando la conexión ha sido validada por el programa, se puede empezar a gestionar la comunicación, para ello se debe acceder a la estructura jerarquizada del servidor, que representa la información que contiene éste para empezar a transmitir. En la Figura 8 se puede ver la estructura de este árbol y la monitorización de una variable:

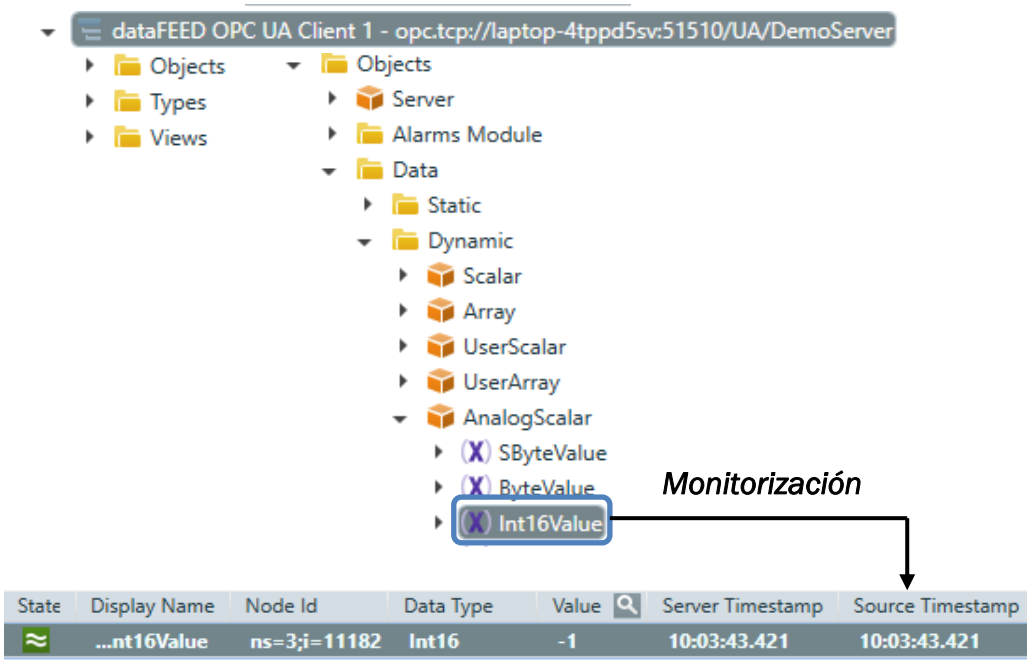


Figura 8. Cliente del Programa dataFEED OPC UA - Softing

Accediendo al servidor como se muestra en la imagen, se podrá monitorizar la variable que se desee. Con esta monitorización se podrá establecer un valor en la variable si se tienen permisos de escritura. El valor introducido debe corresponderse con el tipo de variable, sino se introduce un valor correcto, el servidor reportará un error y el valor no se cambiará. Los tipos de variables que utiliza el protocolo OPC UA son, *Boolean, Arrays, String, Int16/32, Double* y *Date-Time*.

- *Boolean:* Tipo de dato lógico, solo acepta dos tipos de valores, 0/False, 1/True.
- *Arrays:* Tipo de dato vector, almacena una cantidad finita de valores, ya sean valores numéricos o valores en forma de texto.
- *String:* Tipo de dato para almacenar caracteres alfanuméricos.
- *Int16/32:* Tipo de dato entero, variables que almacenan números enteros solamente.
- *Double:* Tipo de dato doble, almacena números reales.
- *Date*-Time: Tipo de dato para almacenar el tiempo y la fecha deseadas con un formato definido.

CAPÍTULO 2

Caso de estudio

2.1. INTRODUCCIÓN

Debido a la globalización del mercado del azúcar. En los últimos años el sector azucarero europeo, ha tenido que crear nuevas alternativas y promover su sostenibilidad, para asegurar su permanencia.

En el caso de estudio en el que se enmarca este TFG, se pretende reducir los costes de operación, utilizando una planta de cogeneración asociada que se gestionará según las posibilidades que ofrece el mercado diario eléctrico español.

En el mercado diario eléctrico español, productores y consumidores realizan diariamente pujas en las que demandan u ofertan una cantidad de electricidad a un precio determinado para cada hora del día siguiente. Con cada una de estas ofertas y demandas se construye una curva de oferta y una curva de demanda agregada, y el precio en el que ambas curvas se cortan decide el precio final de la electricidad. Este mercado presenta una serie de particularidades:

- El mercado es gestionado por un operador independiente, Operador del Mercado Ibérico de Energía (a partir de ahora OMIE).
- Las ofertas se introducen a través de Unidades de Oferta (UOF)
- Se permite presentar ofertas para las 24 horas del día siguiente a su cierre, siempre y cuando dicha oferta se presente antes de las 12:00 p.m.
- En caso de no cumplir con la oferta o demanda correspondiente la empresa sufrirá penalizaciones económicas.

Para más información acerca del mercado eléctrico, consultar [13].

La cogeneración es un sistema que se beneficia de la producción y la utilización conjunta de la energía térmica útil y electricidad, en un proceso secuencial, permitiendo mayores niveles de eficiencia, así como rebajar las emisiones de gases de efecto invernadero generadas.

Las características principales del sistema de cogeneración están relacionadas con la sostenibilidad ambiental, y el beneficio económico para los productores. Gracias al aprovechamiento simultáneo de la energía, los productores, consiguen generar mayor rendimiento potencial que una central convencional, y en consecuencia una mayor sostenibilidad medioambiental, puesto que promueve un menor consumo de combustible para su proceso. Esto influye de manera notable también en la inversión de materia prima y el gasto de producción, favoreciendo la competitividad en el mercado. Para encontrar más información sobre la cogeneración consultar [14].

Los elementos principales que constituyen un sistema de cogeneración son: Fuente de energía primaria, Elemento motor (Turbinas o Motores), Sistema de aprovechamiento de energía mecánica, Sistema de aprovechamiento del calor, Sistemas de refrigeración, Sistema de tratamiento de agua, Sistema de control y Sistema eléctrico [15].

Para evitar la pillería, existen una serie de leyes a nivel europeo que dictan cómo debe generarse la energía eléctrica de forma eficiente en sistemas de cogeneración. El cumplimiento de dichas leyes repercute en una serie de ventajas para el productor, como el hecho de prioridad en el despacho eléctrico.

El presente TFG parte de una herramienta de optimización desarrollada para lidiar con las dificultades que una empresa azucarera puede encontrar a la hora de participar en el mercado eléctrico diario. Dicha herramienta ha sido testeada utilizando una fábrica azucarera simulada comunicada con la optimización a través del protocolo OPC UA. Dado que la optimización se encuentra programada en Python, y puede ser difícil de utilizar e interpretar por personas que no tienen conocimientos de programación, en este TFG se busca desarrollar una interfaz amigable que sirva para que cualquier operario pueda lanzar optimizaciones y estudiar las soluciones propuestas por la herramienta de optimización.

Para comprobar el correcto funcionamiento de la interfaz, se utilizará la simulación mencionada anteriormente, ya que como se verá en los próximos apartados, la configuración desarrollada puede ser fácilmente adaptada para procesos reales. En el siguiente apartado se explica con más detalle los módulos de optimización y simulación de los que se ha partido.

2.2. MÓDULOS DE SIMULACIÓN Y OPTIMIZACIÓN

Las herramientas de simulación y optimización, vinculadas a la planta azucarera y al mercado eléctrico, han sido el punto de partida para diseñar la interfaz gráfica que sea capaz de establecer la comunicación entre ambas. Por este motivo es necesaria la compresión de ambas, y así conseguir el objetivo general de este TFG, por lo que a continuación se detallarán.

2.2.1. Módulo de simulación de la planta azucarera

El proceso de obtención del azúcar se clasifica como un proceso continuo en el que el azúcar es extraído de la remolacha. En esta extracción la materia prima pasa por diferentes subprocesos hasta que finalmente se obtiene el producto final. Estos subprocesos requieren un cierto suministro energético para poder llevarse a cabo. Para ello se cuenta con una planta de cogeneración que generará energía térmica y energía eléctrica.

En el caso de estudio presentado, este proceso está representado por una simulación, que utiliza un modelo de caja gris no lineal, [16]. En dicho

modelo, se tienen una serie de entradas y salidas definidas en la Tabla 3 y en la Tabla 4:

Entradas del Modelo	
Función	*Valores*
Tasa de producción de remolacha	[370 - 430 T/h]
Generación de energía eléctrica en las turbinas	[5000 - 11000 kW]
Presión de trabajo de evaporación	[2.2 - 3 barA]
Temperatura de vapor sobrecalentado en las calderas	[360 - 420 ºC]

Tabla 3. Entradas del modelo de simulación

Salidas del Modelo
Función
Consumo de energía eléctrica de la fábrica de azúcar
Caudal másico de gas natural necesario para el proceso
Tiempo promedio que pasa la remolacha en la zona de almacenamiento
Presión de vapor en el cuarto efecto de evaporación
Índices de la legislación europea
Consumo de energía térmica de la fábrica de azúcar

Tabla 4. Salidas del modelo de simulación

La implementación del modelo de la simulación se llevó a cabo con el software de modelado y simulación EcosimPro [17]. Esta herramienta permite el desarrollo de modelos complejos utilizando un lenguaje de modelado (EL, EcosimPro Language) orientado a objetos. También cuenta con una potente interfaz gráfica que facilita la creación y diseño de diferentes modelos utilizando librerías específicas para ello. Estas librerías están basadas en métodos numéricos y simbólicos que son capaces de procesar ecuaciones diferenciales en sistemas complejos. Por su programación orientada a objetos, permite un modelado jerárquico que simplifica la tarea de implementación en modelos de sistemas complejos de gran tamaño.

Con este programa se ha creado un servidor con el modelo implementado, éste utilizará un protocolo de comunicaciones para transmitir los resultados del modelo. El protocolo que se definió para la comunicación del servidor, se corresponde con el estándar OPC UA.

2.2.2. Módulo de optimización

La planta de cogeneración asociada a la industria azucarera, está formada principalmente por tres calderas que trabajan en paralelo y tres turbinas de contrapresión. Las calderas se encargan de generar el vapor de alta presión evaporando agua utilizando como fuente de combustible gas natural. El suministro de gas natural se controla mediante un regulador de presión, que gestiona el control de la válvula encargada de regular el suministro de combustible. Las turbinas son capaces de generar vapor

saturado de baja presión, que se aprovechará en uno de los subprocesos y a su vez generará una gran cantidad de energía eléctrica. Esta energía eléctrica será controlada por un regulador, el cual establecerá su consigna en función de la cantidad de energía eléctrica que se quiera generar. En ocasiones, esta energía no es consumida en su totalidad por los subprocesos que la requieren y por estas situaciones se debe realizar una optimización del proceso, para que el excedente de energía pueda ser vendido a la compañía eléctrica de acuerdo al mercado eléctrico español en función de los requerimientos de energía de este.

El sistema de cogeneración de la planta ha sido el objeto de estudio para la realización de la herramienta de optimización. Esta herramienta utiliza la cantidad de remolacha que va a llegar a la planta y el precio de la electricidad predicho para el día siguiente para calcular las entradas del modelo de simulación.

La función de la optimización es doble. Por un lado se busca obtener la cantidad de electricidad que la fábrica debe comprometerse a vender al día siguiente en el mercado eléctrico. Por otro lado, se obtiene la ley de control que debe utilizarse para obtener el excedente pactado.

El proceso de cálculo del optimizador, se desarrolló con el lenguaje de programación Python. Para ello se utilizó el módulo de Pyomo [18], que proporciona una gran variedad de características para facilitar la tarea de implementación software.

En último lugar, se considera importante señalar que las herramientas descritas han sido proporcionadas por la tutorización académica de este TFG.

La resolución de este caso de estudio, ha pasado por diferentes etapas durante su desarrollo, estas se explican en el capítulo 3 del presente documento.

Para facilitar la comprensión del proceso de cogeneración se adjunta una imagen que muestra las partes más relevantes del proceso, este esquema se utilizará como base para la representación del proceso en la interfaz gráfica. Este proceso puede visualizarse en la Figura 9.

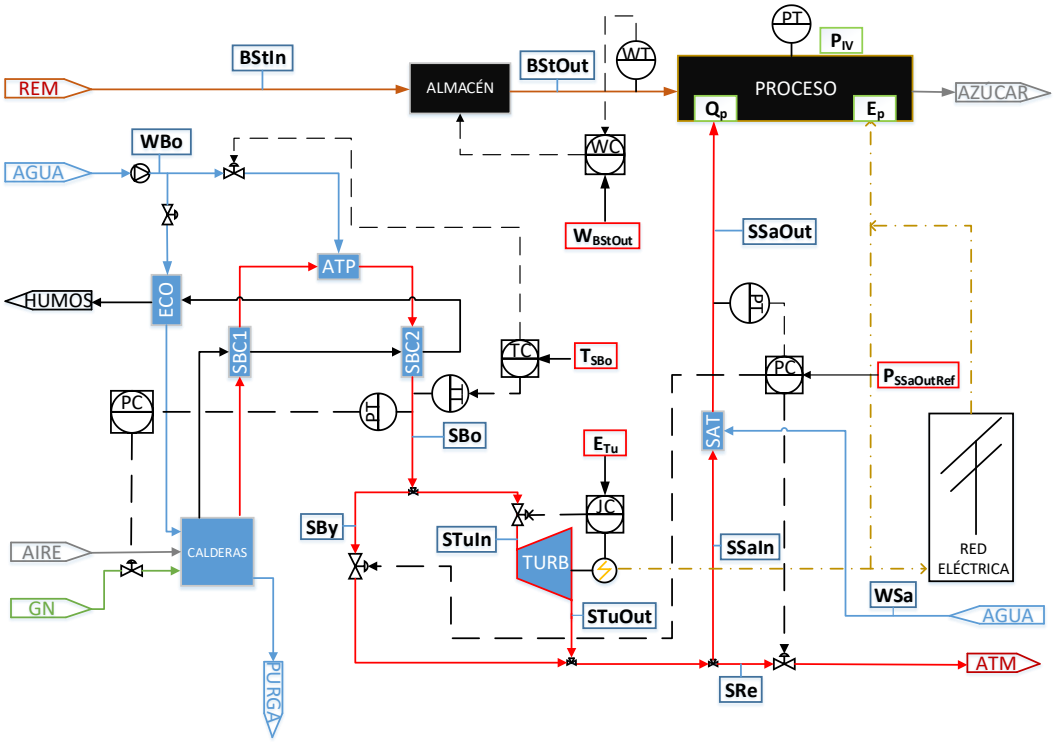


Figura 9. Proceso a implementar en la interfaz gráfica

CAPÍTULO 3.

Diseño de la aplicación

3.1. RESOLUCIÓN DEL CASO DE ESTUDIO

Anteriormente se ha descrito la problemática a la que se pretende dar solución, es decir, se intenta resolver la situación de comunicación estableciendo una interfaz gráfica, que se comunique con las herramientas disponibles.

En primer lugar, se ha tenido en cuenta que el servidor de la simulación incorpora una capa de comunicaciones con el protocolo OPC UA. Por lo que aprovechando las ventajas que ofrece este protocolo (Véase apartado 1.2) se ha utilizado como vía principal de comunicación de la interfaz gráfica.

Por otro lado, cabe señalar que la interfaz gráfica se ha encargado de recoger los parámetros necesarios para gestionar esta comunicación. Dicha interfaz gráfica ha sido diseñada con el entorno Wonderware que se ha descrito en el apartado 1.3.

En relación con el proceso de optimización y el establecimiento de la comunicación con este, se desarrolló un servidor que incorporaba los cálculos necesarios, utilizando una capa de comunicaciones con el protocolo OPC UA. Para ello se utilizó el lenguaje de programación Python, que es un lenguaje de código abierto, que cuenta con una gran variedad de librerías que facilitan su programación. La librería que se ha utilizado para implementar la capa de comunicación ha sido *freeOPCUA*, para mayor información sobre la implementación y la librería, consultar el Anexo II.

Durante el desarrollo del TFG, se detectaron problemas en la comunicación de datos de tipo “array” y en la configuración de gráficas para la representación de los valores de las variables. Por lo tanto, se decidió utilizar el paquete de integración MS Office que proporciona Wonderware, más concretamente, Microsoft Excel. Esta solución ha permitido introducir el escenario de precios y la llegada de remolacha, utilizando filas y columnas para simular los datos del tipo “array”. Estos datos serán los que leerá el servidor de la optimización para realizar los cálculos necesarios. También se ha utilizado este programa para la representación gráfica de los valores aportados en el cálculo de la optimización. Con esto se ha conseguido visualizar la evolución temporal de las entradas del modelo que se comunicarán al servidor de la simulación.

Es importante señalar que los resultados del cálculo de la optimización proporcionados por el servidor son en forma de “array”, incompatibles con la lectura del programa Wonderware. Por este motivo ha sido necesario añadir

una capa de comunicaciones que implemente un cliente OPC UA. Este cliente se encargará de pedir los datos del cálculo de la optimización para poder enviarlos al servidor de la simulación.

Gracias a la realización de esta interfaz y las cuestiones que se han tenido en cuenta en su desarrollo, se ha conseguido a través de la utilización de las herramientas que la comunicación sea efectiva y fiable. La situación que ha sido descrita puede observarse de forma resumida en el diagrama de flujo de la Figura 10. Diagrama de

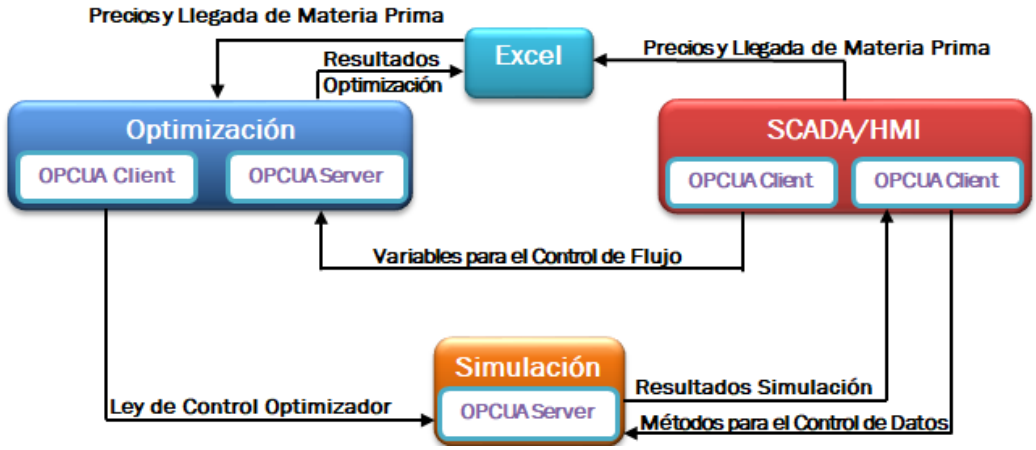


Figura 10. Diagrama de Flujo de Comunicaciones

3.2. ARQUITECTURA DE LA APLICACIÓN

En el diseño de la aplicación se puede definir una arquitectura de comunicaciones que represente la comunicación entre los tres tipos de módulos, el tipo de protocolo que se va a utilizar en su comunicación y el rol que desempeñan desde el punto de vista cliente/servidor, como se muestra en la Figura 11. Arquitectura de Comunicaciones de la Aplicación

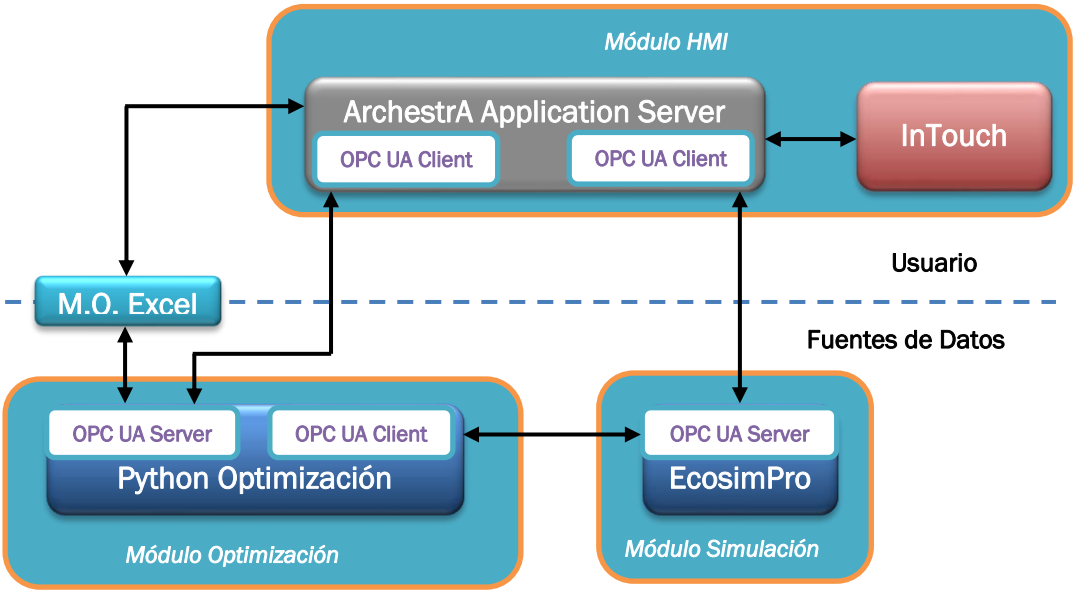


Figura 11. Arquitectura de Comunicaciones de la Aplicación

En este módulo únicamente se implementa un servidor de datos OPC UA, que recibirá las peticiones correspondientes de los clientes del optimizador y del HMI. La función principal de este servidor, es calcular la respuesta del sistema en función de una determinada ley de control que proporcionará el optimizador. Los resultados del sistema que proporcionará el servidor, se enviarán al módulo HMI. Para que el servidor calcule la respuesta del sistema, este incorpora una serie de métodos y parámetros que facilitarán la tarea del cálculo y la manipulación de la simulación. Los parámetros que se deben definir, se corresponden con el tiempo final de simulación y el valor para la integración del cálculo para la respuesta del sistema. Como las ofertas de energía eléctrica se producen al día siguiente de su solicitud, el tiempo final de simulación se establecerá en 24h, para obtener la evolución del sistema organizada en días y contabilizar los días en el que el optimizador envía los resultados al simulador. Por otro lado, el valor de la integración del cálculo, se definirá como un factor de aceleración, este factor se encargará de establecer la velocidad de ejecución de la simulación, permitiendo simular 24h del proceso en un tiempo mucho menor. Estos parámetros deben estar definidos previamente al uso de los métodos del servidor. Estos métodos, entre otras cosas, son capaces de: establecer el valor inicial de las variables para ejecutar la simulación (*command_run)*, ejecutar el valor de la integración del cálculo (*command_integ_cint)*, y reiniciar los valores de las variables de la simulación (*command_reset)*. Con estos métodos y parámetros se conseguirá tener un control total de la ejecución de la simulación en el módulo HMI, que será el encargado de gestionarlos.

- *Módulo Optimización*

  El módulo de optimización se comunicará con el módulo HMI para recibir la política de precios del mercado eléctrico y la llegada de remolacha, que utilizará para realizar el cálculo de la optimización y comunicárselo al módulo simulación. El cálculo que implementa la optimización debe ser capaz de proponer una ley de control que le llegará al servidor de la simulación en base a un escenario de precios y la llegada de materia prima. Estos resultados se mostrarán en gráficas Excel, para que sea el operario quien pueda valorar los resultados y aceptar que éstos son correctos para enviarlos al servidor de la simulación. Para realizar estas funciones, en el módulo de optimización se han implementado dos capas que realizan la función de cliente y de servidor. La capa del cliente OPC UA se encargará de recoger los resultados calculados por el servidor de la optimización y enviarlos al servidor de la simulación. Para gestionar estas funciones correctamente, en el servidor de la optimización se tiene una serie de métodos que definen, el inicio del cálculo de la optimización (*StarOpt)*, el inicio de la comunicación entre el cliente de la optimización y

el servidor de la simulación (*StartCom)* y el final de la comunicación entre cliente y servidor (StopCom). Con estos métodos se tendrá el control total de la optimización, para gestionar la comunicación desde el módulo HMI correctamente.

- *Módulo HMI*

  Este módulo establecerá dos tipos de comunicaciones, comunicación interna, donde el programa encargado del diseño gráfico de la aplicación se comunicará con el gestor de la aplicación, y la comunicación externa, que se realiza entre el gestor de la aplicación y los servidores de la simulación y de la optimización. La función principal de este módulo, es establecer los parámetros necesarios de cada servidor, para gestionar las comunicaciones externas con estos y así poder mostrarlos en la interfaz gráfica diseñada. El gestor de la aplicación se establece utilizando el software asociado "ArchestrA Application Server", encargado de administrar las comunicaciones externas e internas. Este gestor cuenta con un acceso directo a las variables administradas en el Gateway del sistema "OIGateway", para más información de la utilización y administración de las variables en el Gateway, véase Anexo I. Éste comunicará los datos de las variables administradas directamente al gestor de la aplicación para enviarlos al programa "InTouch" que será el encargado del diseño gráfico. En "InTouch" se establecerán los parámetros de cada servidor, siendo estos el factor de aceleración correspondiente al servidor de la simulación y el escenario de precios del mercado eléctrico y la llegada de remolacha al sistema, correspondientes al servidor de la optimización. También se mostrarán los resultados de las variables del servidor de la simulación, así como las gráficas en tiempo real de dichas variables.

El programa Excel no se ha incluido en ningún módulo, ya que actúa como elemento auxiliar de comunicación entre el módulo HMI y el módulo de la optimización.

3.3. DESARROLLO DE LA INTERFAZ GRÁFICA

Para el desarrollo de la interfaz gráfica, se han utilizado los programas asociados de Wonderware "InTouch" y "ArchestrA Application Server". El programa "ArchestrA Application Server" será el encargado de establecer la comunicación con el Gateway donde se encuentran administradas las variables de los servidores. Además, también se encargará de establecer la comunicación con "InTouch" para utilizar los objetos diseñados en el programa. Estos objetos se corresponden con los elementos gráficos de la aplicación, que estarán diseñados acordes a los requisitos de la interfaz gráfica, para más información véase Anexo I. Cada uno de ellos agrupa características comunes que permiten realizar una programación orientada a

objetos facilitando su implementación. Estos objetos podrán sincronizarse con "InTouch" para que puedan ser utilizados por este programa y ser incluidos en las ventanas diseñadas para la aplicación. Por ello "InTouch" se encargará de administrar todos los elementos gráficos de la aplicación. Estos se agruparán en ventanas según las necesidades de la aplicación.

Para que el usuario pueda interactuar con la interfaz gráfica que se pretende diseñar, Wonderware cuenta con una gran cantidad de funcionalidades que se le pueden asociar a los gráficos diseñados. Esta variedad en funcionalidades permitirá abordar todos los requisitos que se vayan estableciendo para el correcto funcionamiento de la aplicación. El software también cuenta con una potente librería de gráficos de alto rendimiento, con los elementos más comunes utilizados en la industria de procesos. Además, se permite la modificación de los elementos prediseñados de la librería, abriendo un gran abanico de posibilidades para implementar en el diseño. Es por ello que se utilizará esta librería como principal fuente de elementos gráficos.

Respecto a la realización de los gráficos de alto rendimiento, se ha seguido el trabajo desarrollado por Pérez [19], puesto que en el diseño de sinópticos ineficientes en interfaces gráficas para el control y la supervisión, llevan a la ineficiencia de las prácticas de operación.

En el informe de Pérez se proponen una serie de características y recomendaciones que deben tener los elementos gráficos dedicados a procesos industriales. Para la interfaz gráfica que se ha diseñado, se han seguido las principales características:

- Uso limitado del color. Solamente se utilizará en situaciones específicas y de manera consistente.
- Los fondos de las ventanas deberán estar en una escala de grises que minimicen el deslumbramiento al operario.
- Gráficas de variables con el formato correcto, proporcionando la información necesaria de un solo vistazo.
- Evitar gráficos con animaciones, excepto en situaciones de alarmas.
- Utilizar una representación 2D, evitando el contraste.
- Asociar técnicas que minimicen los errores de entrada de datos del operario.

Si se siguen estas características se logrará tener una interfaz gráfica potente y con gráficos de alto rendimiento, que faciliten la interacción con el usuario.

Para realizar el diseño de la interfaz gráfica se han establecido cuatro pasos. Estos se explicarán con detalle, ya que es el método que se ha seguido durante el desarrollo de la interfaz.

Paso 1: Creación del Elemento Gráfico

La creación de los gráficos correspondientes a cada uno de los objetos se realiza utilizando el programa de edición "ArchestrA Symbol Editor" que se encuentra dentro de "ArchestrA Application Server". Al iniciar este programa se tendrá la imagen que muestra la Figura 12.

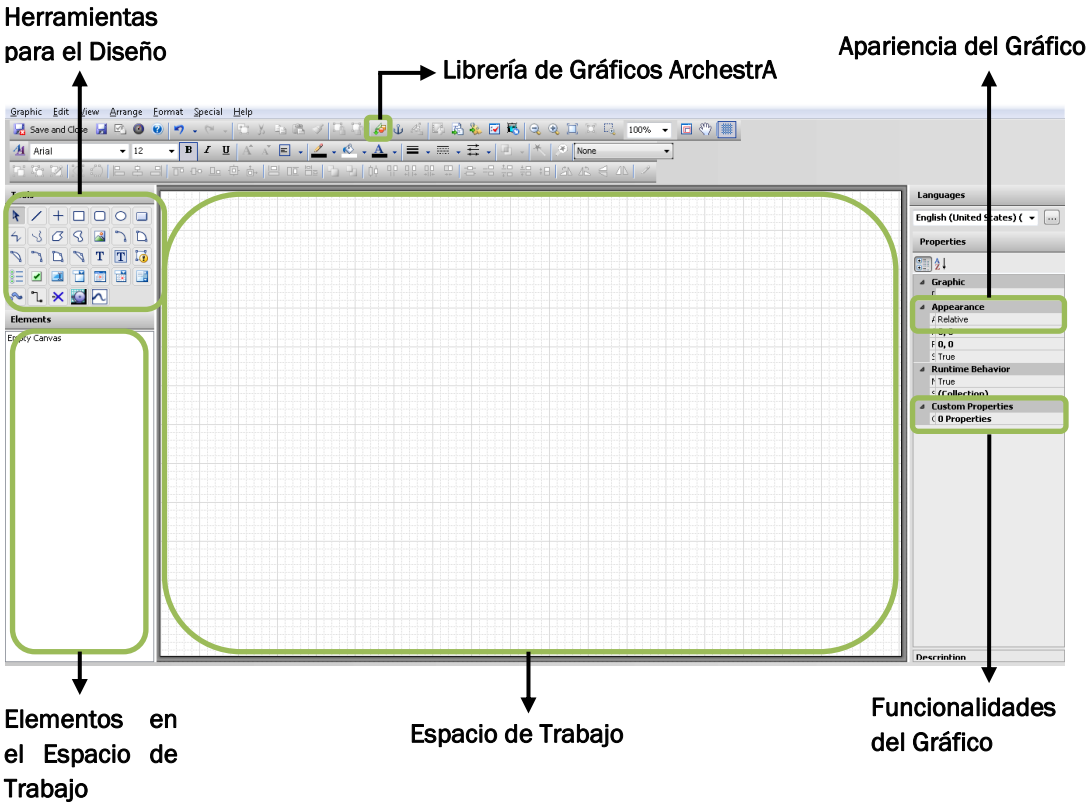


Figura 12. Ventana Principal ArchestrA Symbol Editor

Los elementos del proceso de cogeneración que se encuentren en la librería de "ArchestrA" se utilizarán realizando las modificaciones necesarias. Si por el contrario el elemento no se encontrara en la librería de gráficos, se realizará el diseño del elemento utilizando las herramientas de diseño y asociando las funcionalidades requeridas para dicho gráfico. La manera de implementar el gráfico es arrastrando la herramienta seleccionada y realizando la configuración adecuada para cada una de estas. Los elementos del proceso de cogeneración que se deberán representar se recogen en la Tabla 5. Elementos del Proceso de Cogeneración

Proceso de Cogeneración	
Bomba Centrífuga	Válvula Manual
Caldera	Transmisores
Turbinas	Etiquetas de Proceso
Reguladores	Bloques Gráficos
Válvulas Automáticas	

Tabla 5. Elementos del Proceso de Cogeneración

Para que el usuario pueda interactuar con la interfaz diseñada, "ArchestrA Symbol Editor" cuenta una serie de funcionalidades que se podrán asociar a los gráficos. Estas funcionalidades permiten abordar numerosas situaciones que podrán surgir durante el diseño de la interfaz. Estas se explican en el paso siguiente.

Paso 2: Asociación de Funcionalidades

Para poder realizar estas asociaciones correctamente, es necesario haber establecido comunicación con las variables de los servidores que se pretenden representar o utilizar en ciertos algoritmos de control, para realizar esta comunicación consultar el apartado 3.2.1. "ArchestrA Symbol Editor" clasifica las funcionalidades dependiendo de la tarea que desarrollen, como se puede observar en la Figura 13

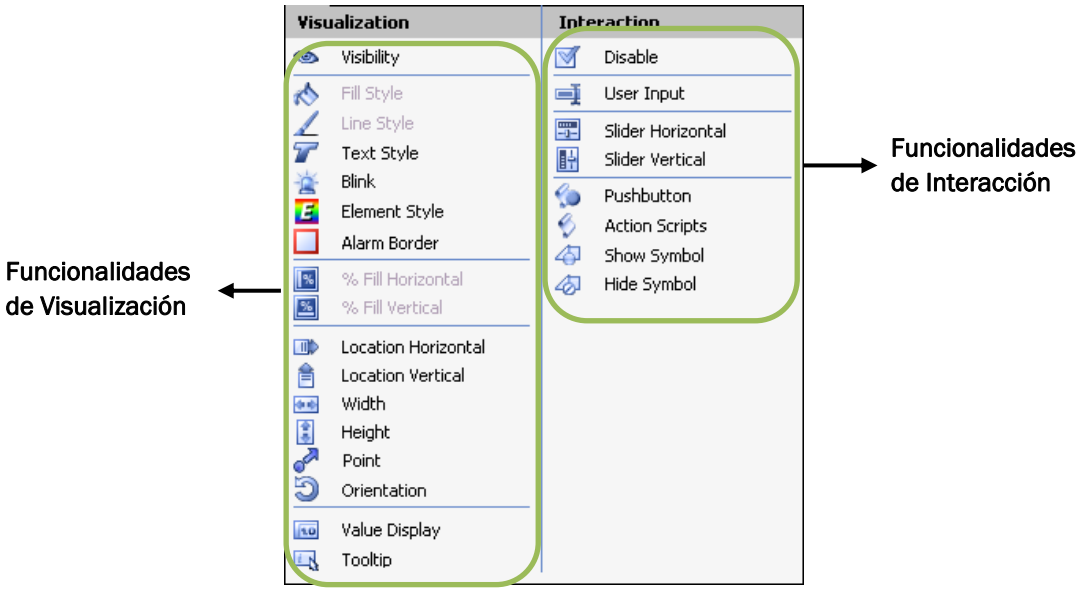


Figura 13. Funcionalidades para los gráficos de ArchestrA Symbol Editor

Con todas estas opciones se conseguirá solucionar cada uno de los requisitos necesarios para el correcto funcionamiento de la aplicación. A continuación, se realiza una breve descripción de las funcionalidades más utilizadas en el diseño:

- *Fill Style:* Se utiliza para rellenar con un color determinado el gráfico diseñado, este cambiará de color en función del valor de una o más variables.

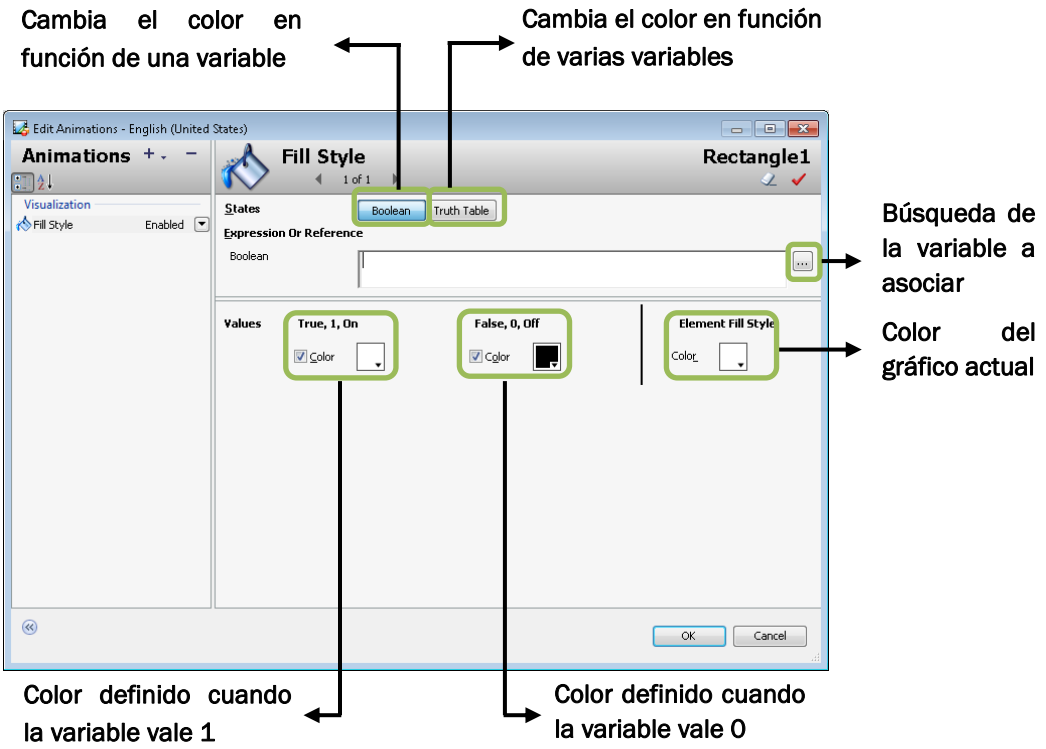


Figura 14. Funcionalidad "Fill Style" asociada a un elemento gráfico

- *Text Style:* Se utiliza para cambiar de color un panel de texto, este cambiará de color en función del valor de una o más variables.

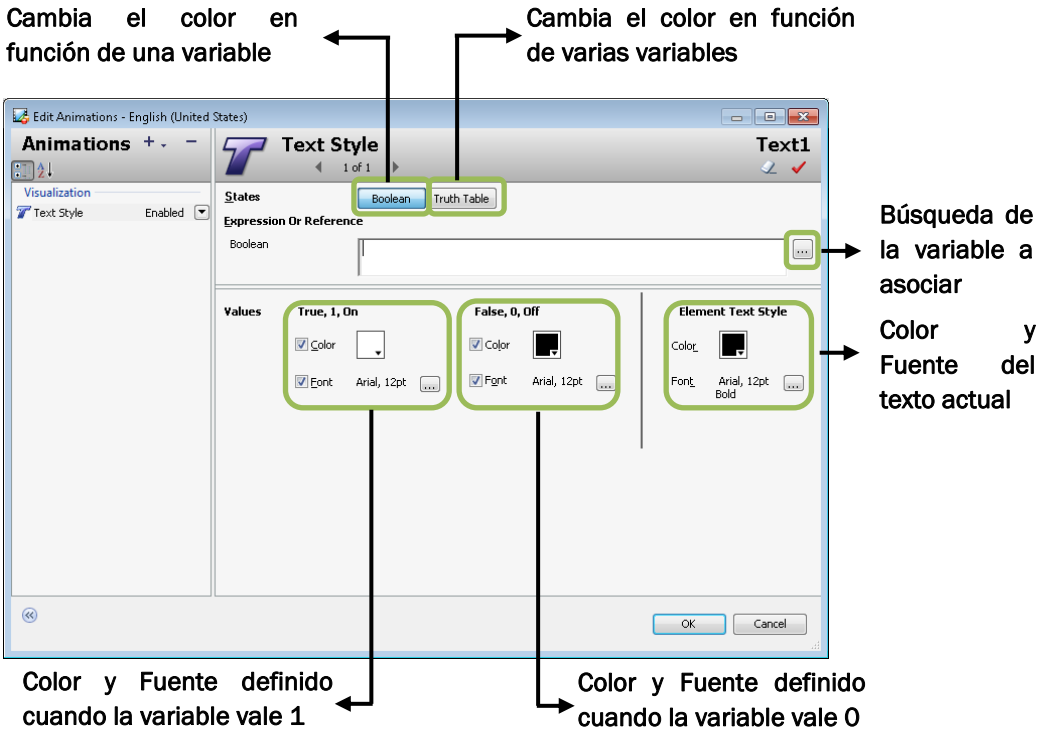


Figura 15. Funcionalidad "Text Style" asociada a un elemento gráfico

- *Slider Horizontal/Vertical:* Asocia a un elemento diseñado el movimiento horizontal o vertical, que permitirá el cambio automático del valor de la variable a la que esté asociada. La configuración se realiza de igual manera para el slider horizontal y vertical.

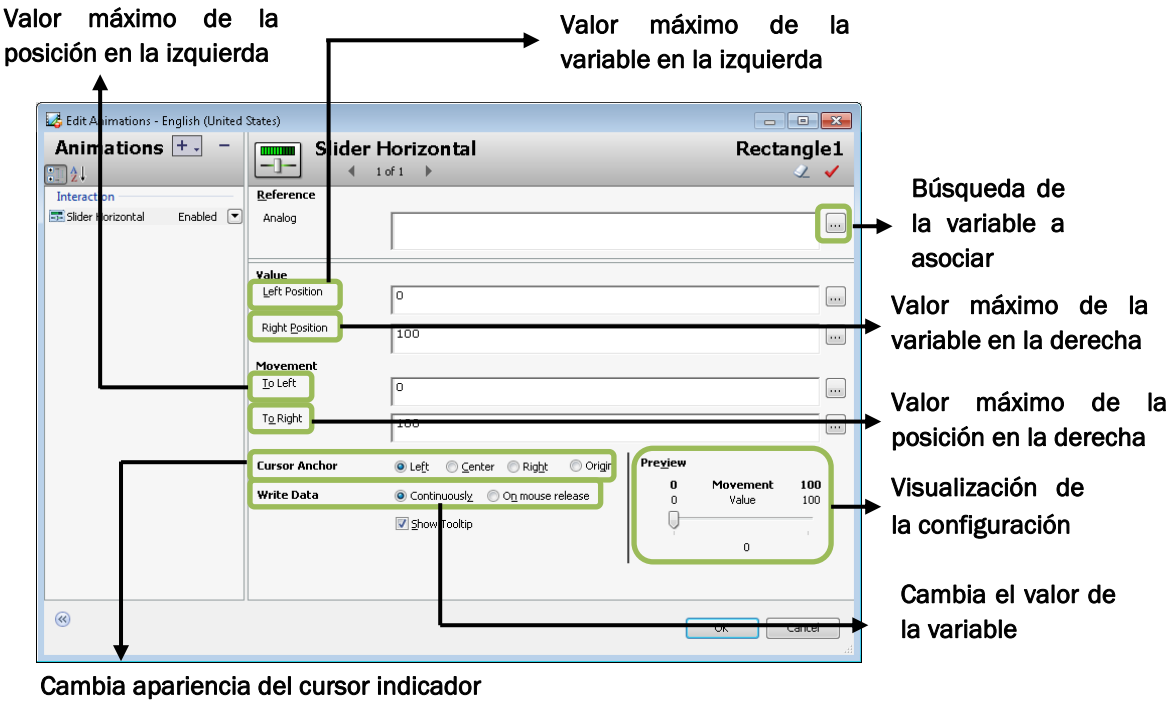


Figura 16. Funcionalidad "Slider Horizontal/Vertical" asociada a un elemento gráfico

- *Value Display*: Muestra el valor de la variable asociada sin que el usuario pueda editar su valor.

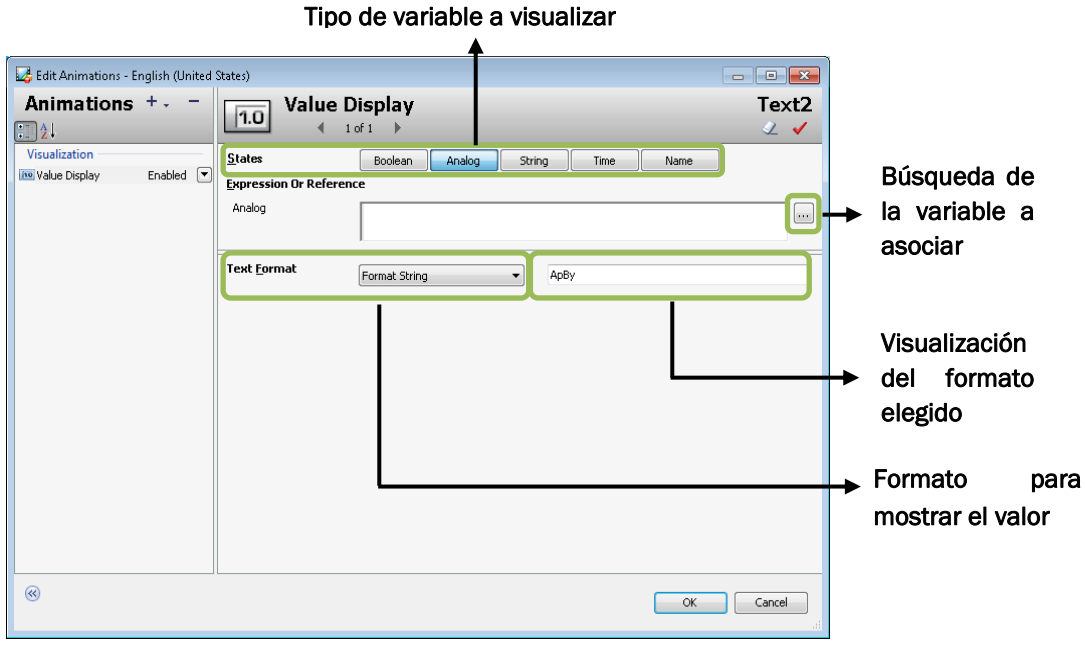


Figura 17. Funcionalidad "Value Display" asociada a un elemento gráfico

- *User Input*: Muestra un teclado virtual, que permite la edición del valor de la variable a la que esté asociada.

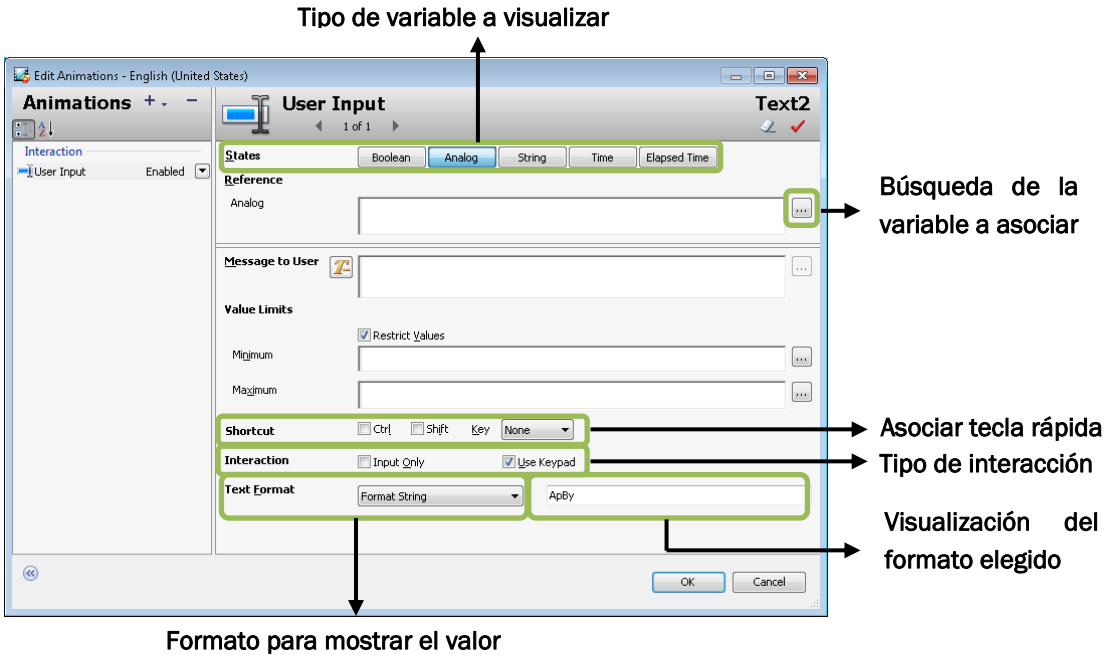


Figura 18. Funcionalidad "User Input" asociada a un elemento gráfico

- *Disable:* Desactiva o activa la interacción con un elemento diseñado, en función del valor una determinada variable.

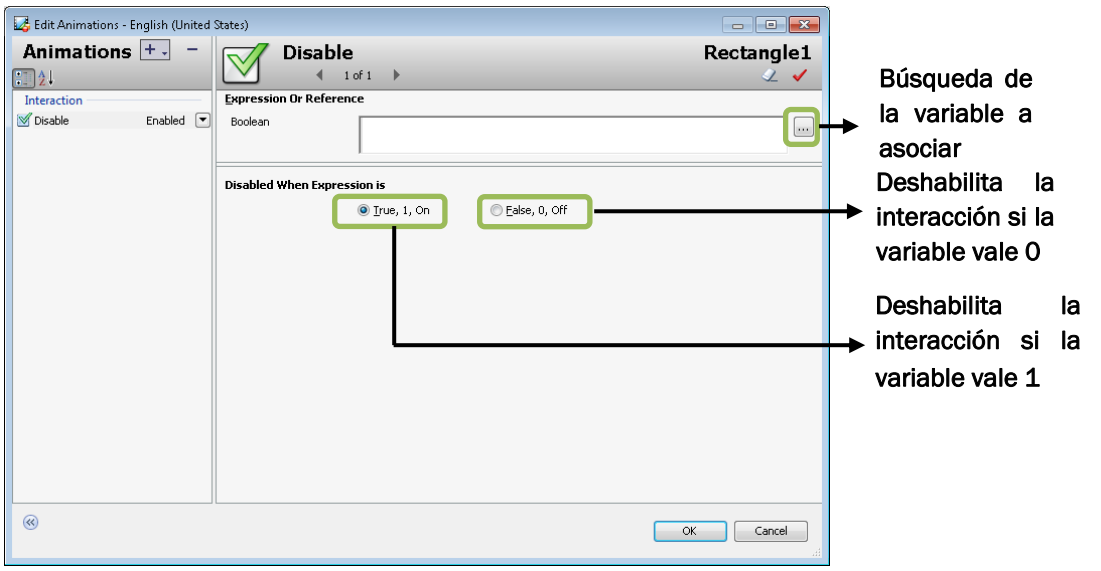


Figura 19. Funcionalidad "Disable" asociada a un elemento gráfico

- *Action Scripts:* Genera un botón de acción, cuando el usuario pulse sobre el elemento se ejecutará el algoritmo asociado a un script, previamente definido.

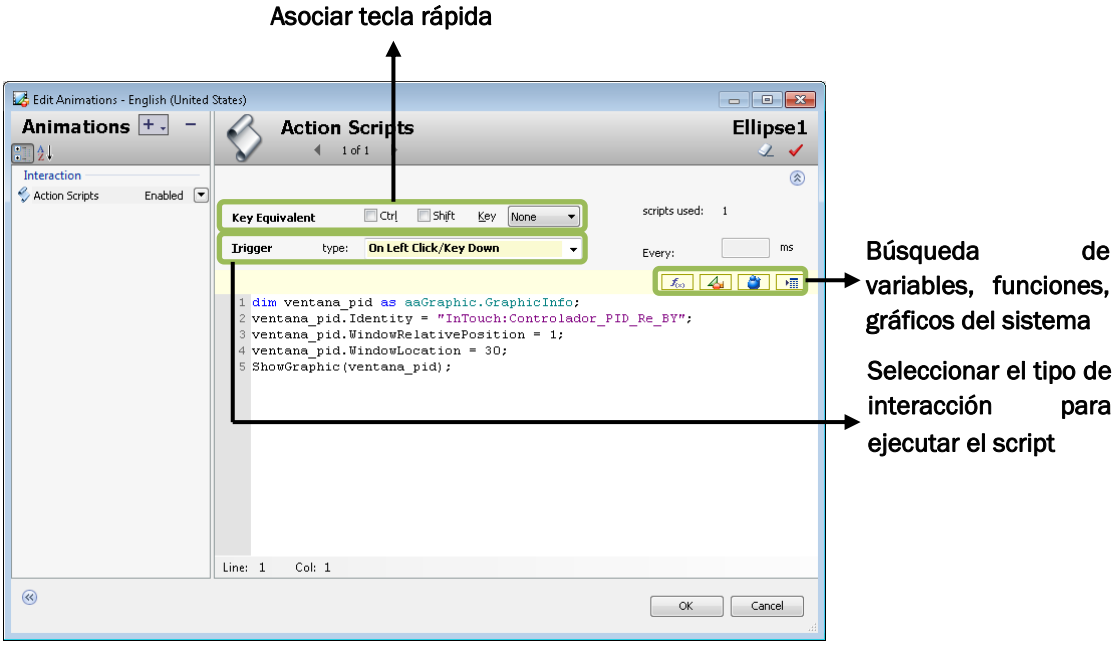


Figura 20. Funcionalidad "Action Script" asociada a un elemento gráfico

Paso 3: Creación de Ventanas de la Aplicación y asociación de Scripts

La creación de las ventanas que contendrán los gráficos de la aplicación se realiza en el programa "InTouch". Este programa cuenta con varios tipos de ventanas, que se deben conocer para realizar la configuración de estas correctamente:

- *Windows Replace: Se* refiere a una ventana que en su ejecución cerrará cualquier ventana que impida su visualización.
- *Windows Overlay: En* su ejecución se mostrará por encima de cualquier ventana que ya estuviera abierta.
- *Windows Popup:* Cuando se ejecuta se mantiene encima de las ventanas que ya estuvieran abiertas, aunque se seleccione otra.

Conociendo las posibilidades que ofrece el programa para la creación de ventanas, en la aplicación se han tenido en cuenta para resaltar ventanas de mayor importancia, frente a ventanas de consulta e información que deberán ocultarse o quedar en segundo plano.

La configuración de estas ventanas se realiza como se muestra en la Figura 21.

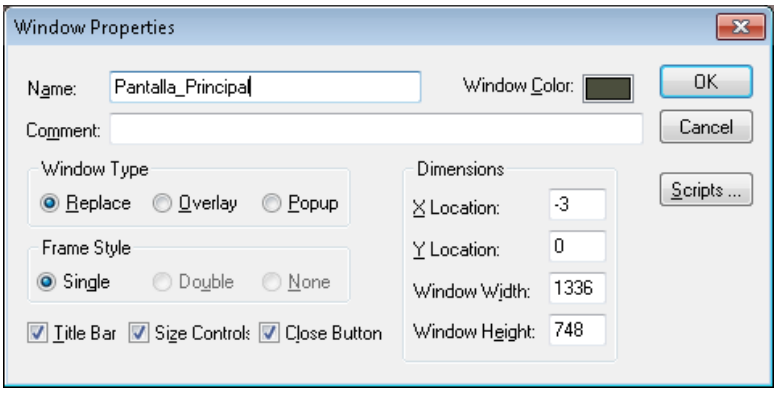


Figura 21. Configuración de ventanas en InTouch

- *Window Type:* En este apartado se configura el tipo de ventana que se desea implementar. En este caso al ser la pantalla principal, se utilizará la opción *Replace*.
- *Frame Style*: Para configurar el tipo de marco que se mostrará en la ventana se deben seleccionar los parámetros Title Bar/Size Control/Close Button. El marco mostrará los elementos seleccionados, los ocultará o no habrá marco, dependiendo de la selección realizada.
- *Dimensions:* Las dimensiones de la pantalla se pueden establecer en función de la situación y el tamaño de la pantalla. Esta pantalla se puede ubicar en cualquier zona del espacio de trabajo de "InTouch" y establecer el ancho y largo adecuado al diseño.
- *Window Color:* Las ventanas permiten establecer un color en el fondo de la pantalla, esto permitirá crear contrastes con elementos de la interfaz y mejorar la visualización de los gráficos.
- *Scripts:* En esta pestaña se configuran los scripts relativos a la ventana, es decir, se pueden asociar scripts cuando la ventana se vaya a mostrar, cuando la ventana esté activa o cuando la ventana esté oculta.

Como se puede observar en la Figura 21, "InTouch" cuenta con la asociación de scripts a ventanas. Los scripts son fragmentos de código en el cual se realizan ciertas acciones utilizando el lenguaje de programación que propone Wonderware. Se deben diferenciar tres tipos de scripts que se pueden asociar a cualquier ventana:

- *Script "On Show":* Se realizará la ejecución del script una sola vez, cuando la ventana se muestre por primera vez.
- *Script "While Showing":* Se realizará la ejecución del script periódicamente, se debe establecer el tiempo de periodicidad de la ejecución que se realizará mientras la venta se esté mostrando.
- *Script "On Hide":* Se realizará la ejecución del script una sola vez, cuando la ventana se oculte por primera vez.

Estos tipos de scripts se utilizarán para establecer el control de la ejecución de la simulación y de la optimización asociado a la ventana

principal de la aplicación. El aspecto para la configuración de estos se muestra en la siguiente imagen:

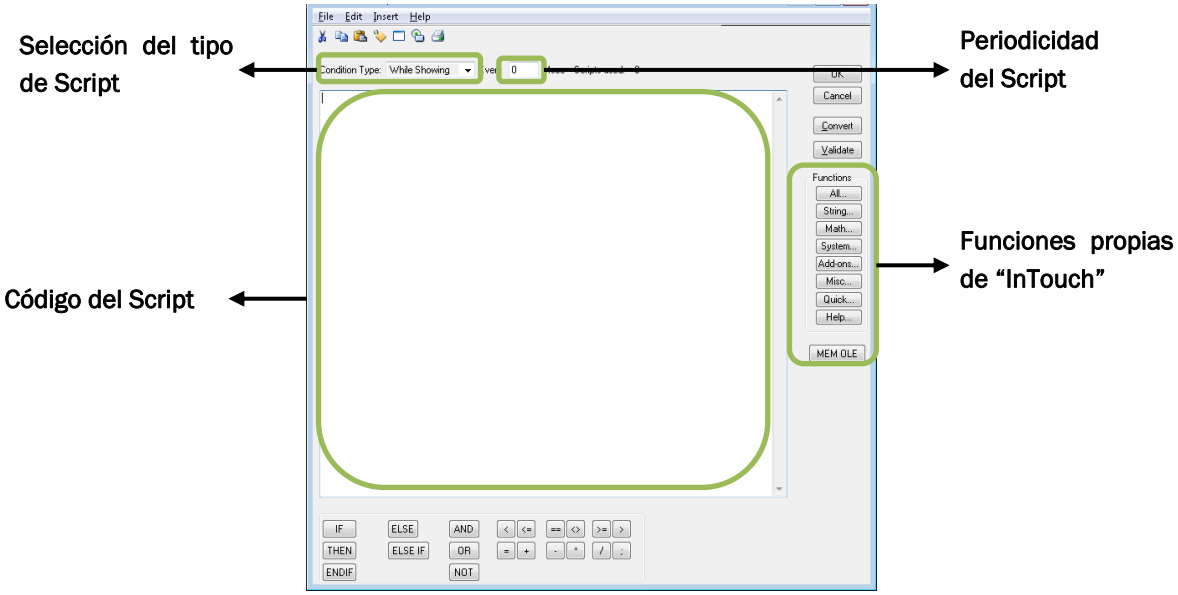


Figura 22. Editar un Script asociado a una ventana de InTouch

Una vez se hayan realizado los pasos descritos anteriormente se tendrá la siguiente configuración del programa que muestra la Figura 23:

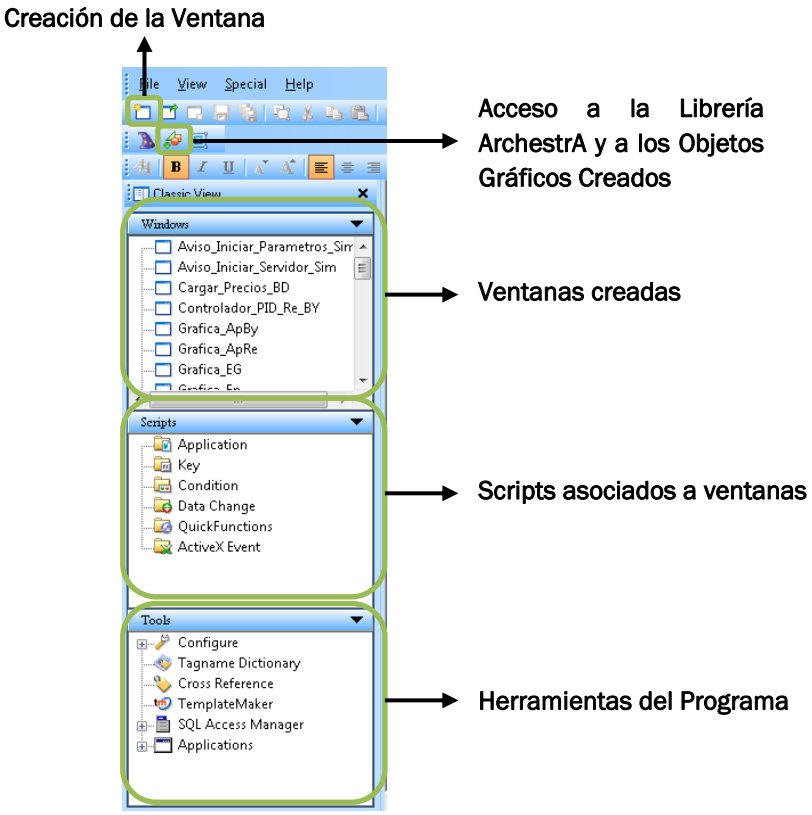


Figura 23. Configuración de InTouch para el diseño de la interfaz gráfica

Paso 4: Creación de Gráficas de Tiempo Real

El programa "InTouch" cuenta con dos tipos de representaciones gráficas, gráficas de tiempo real y gráficas de históricos. Las gráficas de tiempo real muestran el valor que va tomando la variable asociada a dicha gráfica utilizando el tiempo del sistema en el que se está ejecutando la aplicación. Con esto se consigue obtener los valores de las variables en tiempo real, realizando la configuración adecuada. Los valores de la variable que no se puedan representar por la escala elegida en el eje temporal, se irán eliminando para poder introducir los nuevos valores que vaya tomando la variable. Sin embargo, las gráficas de históricos almacenan en una base datos interna, los valores que van tomando las variables durante su representación. Como en este TFG interesan los valores en tiempo real que va tomando el proceso durante su manipulación, se han utilizado las gráficas de tiempo real que propone "InTouch".

A continuación, se muestra en la Figura 24 la ventana que debe configurarse para implementar este tipo de gráficas.

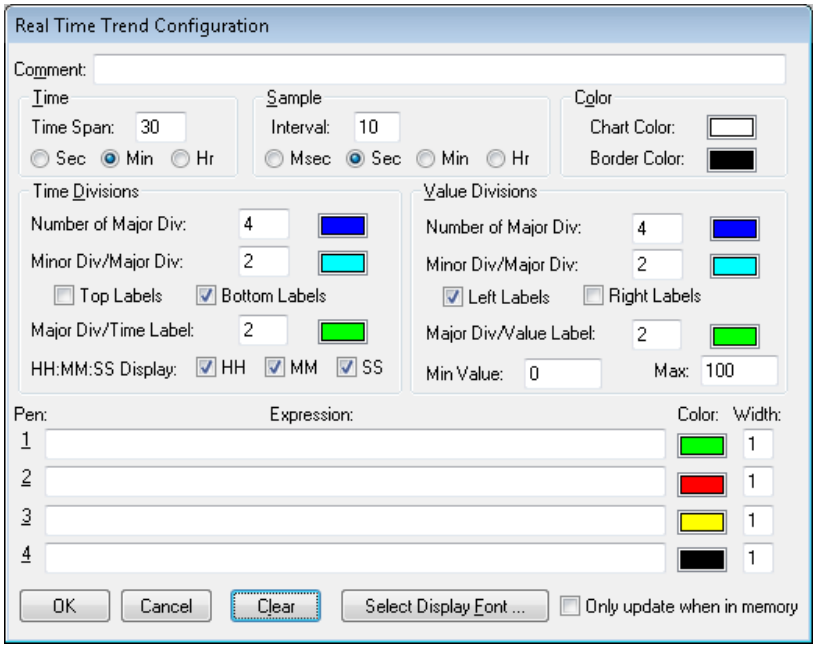


Figura 24. Configuración de gráficas en tiempo real en InTouch

Las características que se van a describir son las principales para implementar la gráfica, los demás parámetros solamente afectan a la visualización de la gráfica, cuya configuración no es relevante para la representación:

- *"Time Span"*: Esta opción se corresponde con el valor que se quiere mostrar en el eje horizontal, pudiendo establecer segundos, minutos y horas.
- *"Sample"*: Esta característica establece la frecuencia de muestreo con la que se obtendrá el valor de la variable para actualizar y representar en la gráfica, pudiendo establecerse en segundos, minutos y hora.
- *"Max/Min"*: En esta opción se puede establecer un valor máximo y mínimo para visualizar en el eje vertical.
- *"Pen"*: En esta característica se define el número de gráficas que se quiere representar en una sola visualización. Se pueden llegar a visualizar hasta 4 gráficas, para asociar la variable que se quiere representar, basta con hacer doble click en el número de gráfica que corresponda y buscar la variable que corresponda.

Realizando esta configuración correctamente se obtendrá la gráfica que se muestra en la Figura 25.

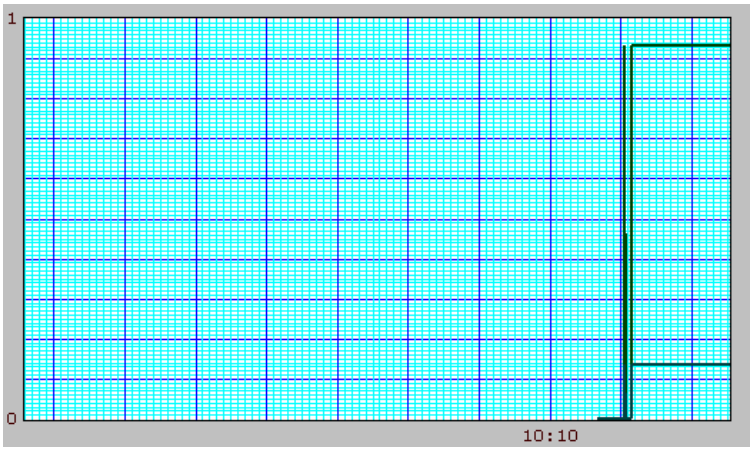


Figura 25. Gráfica de tiempo real proporcionada por InTouch

Como se puede ver en la imagen, la representación carece de mucha información que se necesita mostrar y captar a primera vista. Por lo tanto, se han realizado una serie de modificaciones que permitan al usuario identificar la variable que se está visualizando, el valor que va tomando en cada momento, las unidades en las que se están representando los ejes y deslizadores asociados a las variables que permitan la interacción con la variable que permita su edición y lectura. Se han establecido dos maneras de representar las gráficas, dependiendo si es una variable lo que se quiere representar o son varias variables.

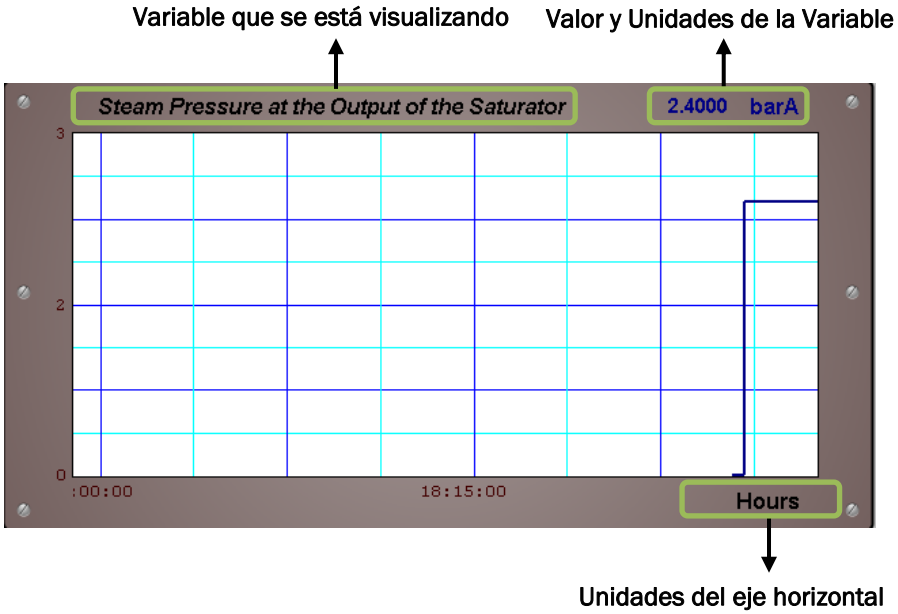


Figura 26. Gráfica de tiempo real editada para una variable

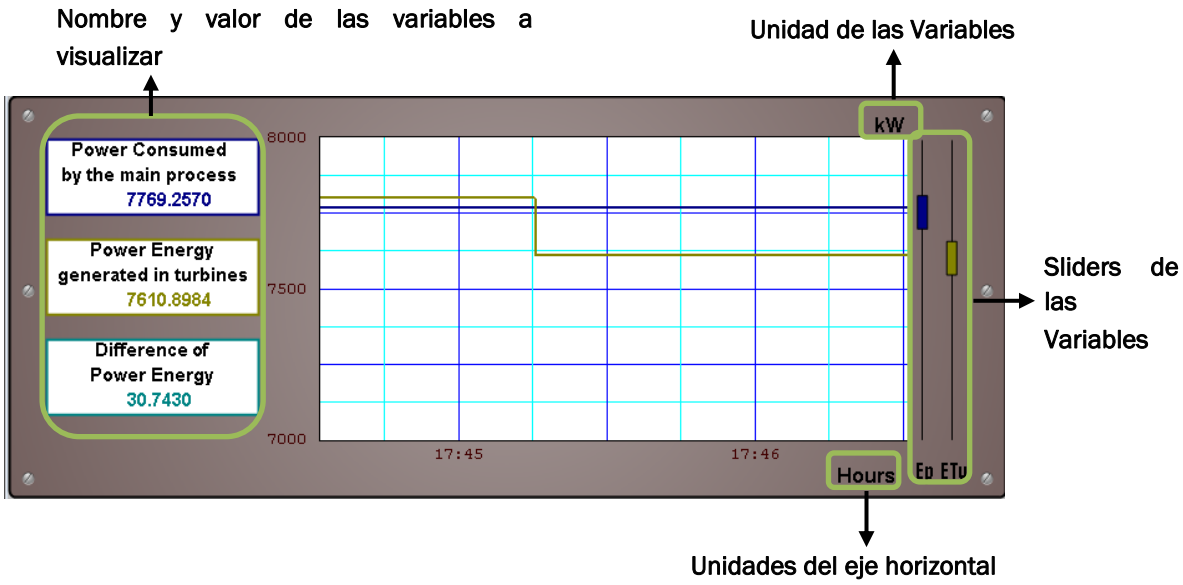


Figura 27. Gráfica de tiempo real editada para dos variables

Como se puede observar en ambas figuras, la información que muestran es mucho mayor que en la situación inicial. Es por ello que todas las gráficas que se necesiten representar en la aplicación se corresponderán con estos diseños. Estas gráficas se asociarán a los elementos gráficos del proceso para representar todas las variables que se estén visualizando, utilizando un script de acción se consigue este cometido.

3.3.1. Comunicación ArchestrA Application Server - OIGateway

El establecimiento de la comunicación debe ser administrado y gestionado, para esto se han realizado unas plantillas que proporciona el gestor de la aplicación (ArchestrA Application Server), permitiendo acceder a las variables de los servidores (OIGateway) que son requeridas en el diseño gráfico.

Para establecer la comunicación entre el OIGateway y el gestor de la aplicación, se debe crear una plantilla con el protocolo de comunicaciones OPC UA. Una vez creada la plantilla, esta se debe configurar para gestionar la comunicación con OIGateway como se muestra en la Figura 28.

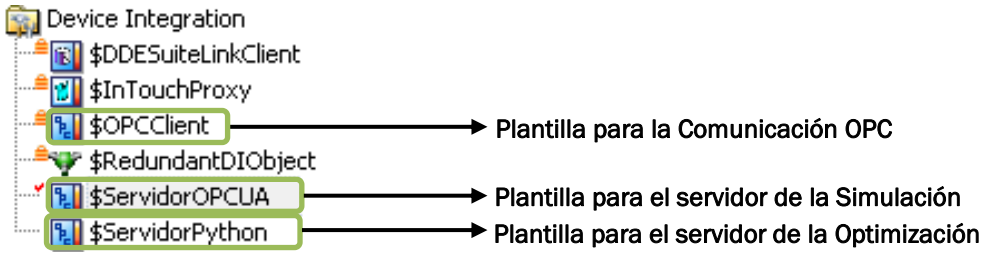


Figura 28. Plantillas para la Comunicación OPC UA

Cabe señalar la creación de dos plantillas, una para cada servidor, puesto que esto permitirá acceder a sus variables de forma organizada. Cada una de estas plantillas, debe configurarse por separado y acceder a la parte del Gateway donde estén almacenadas las variables del servidor correspondiente. Para mayor información sobre la configuración y la implementación de estas plantillas véase el Anexo I. Una vez que la configuración se ha realizado correctamente, las variables que se podrán gestionar de cada servidor utilizando ArchestrA Application Server se corresponden con las Figura 29 y Figura 30:

Gestión de las Variables del Servidor de la Optimización

Associated attributes for Comandos_opt:

Attribute	Item Reference
CINT_OPT	Servidor_Optimizacion.Comandos_control.CINT_OPT
StartCom	Servidor_Optimizacion.Comandos_control.StartCom
StartOpt	Servidor_Optimizacion.Comandos_control.StartOpt
StopCom	Servidor_Optimizacion.Comandos_control.StopCom
TSTOP_OPT	Servidor_Optimizacion.Comandos_control.TSTOP_OPT
reset	Servidor_Optimizacion.Comandos_control.reset

Figura 29. Variables del servidor de la optimización

Gestión de las Variables del Servidor de la Simulación

Associated attributes for Variables:

Attribute	Item Reference
ApBy	Servidor_V101.Items./Variable/ApBy
ApRe	Servidor_V101.Items./Variable/ApRe
CINT	Servidor_V101.Items./Variable/CINT
ETu	Servidor_V101.Items./Variable/ETu
Ep	Servidor_V101.Items./Variable/Ep
Epeq	Servidor_V101.Items./Variable/Epeq
Epi	Servidor_V101.Items./Variable/Epi
Epx1	Servidor_V101.Items./Variable/Epx1
Epx1'	Servidor_V101.Items./Variable/Epx1'
Epx2	Servidor_V101.Items./Variable/Epx2
Epx2'	Servidor_V101.Items./Variable/Epx2'

Associated attributes for Comandos:

Attribute	Item Reference
command_integ_cint	Servidor_V101.Commands./Command/command_integ_cint
command_reset	Servidor_V101.Commands./Command/command_reset
command_run	Servidor_V101.Commands./Command/command_run

Figura 30. Variables del servidor de la simulación

Estas plantillas permiten definir el nombre de las variables que se desee, facilitando su localización para la asociación con elementos gráficos. Como se puede ver en la figuras anteriores, cada una de las variables está vinculada a una dirección. Esta dirección se corresponde con la referencia que ha sido configurado en el Gateway. Para más información véase el Anexo I.

3.3.2. Desarrollo de la Interfaz Gráfica del Proceso

En primer lugar se ha buscado desarrollar una interfaz gráfica que permita ver qué es lo que está ocurriendo en la planta azucarera. Esto se ha realizado para facilitar la visualización de las partes más relevantes de este proceso de esta industria, más concretamente el sistema de cogeneración asociado.

Para facilitar la tarea de diseño, en cada apartado se propone una serie de requisitos que deben satisfacerse para el correcto funcionamiento de la aplicación. Estos pueden observarse en la Figura 31.

Requisitos Interfaz de Proceso

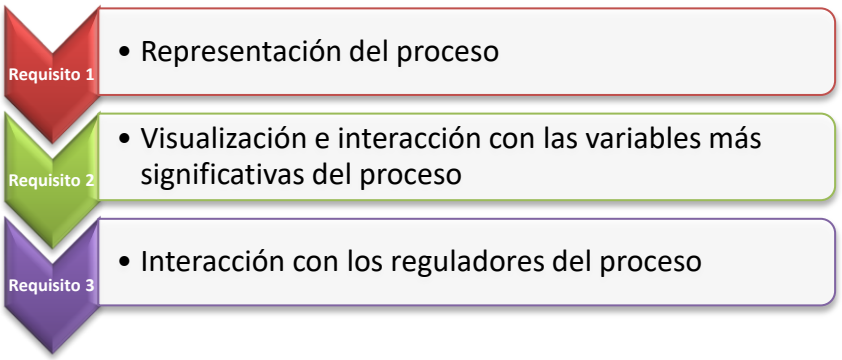


Figura 31. Requisitos para la interfaz del proceso

A continuación, se muestra la forma de afrontar estos requisitos utilizando las prestaciones de Wonderware, siguiendo los pasos descritos en el apartado 3.2 de este capítulo:

Requisito 1: Representación del proceso

Creando los objetos necesarios (Véase Anexo I) y utilizando la herramienta ArchestrA Symbol Editor, se ha realizado la implementación para la representación de los gráficos del proceso. A continuación, se detalla los objetos necesarios y su representación:

- Objeto: Dispositivos del Proceso
  En este objeto se han agrupado los elementos del proceso de cogeneración que hacen alusión a la bomba centrífuga, la caldera y las turbinas (por simplicidad se ha representado únicamente una turbina, en vez de las tres en paralelo que dispone el proceso). Estos elementos no necesitan ninguna funcionalidad para la interacción del usuario con el proceso, simplemente a modo de información se ha implementado en la turbina, un texto indicador del funcionamiento de ésta. Este indicador representará el estado de la turbina en todo momento:

Dispositivos del Proceso	
Elemento	*Creación del Gráfico*
Bomba Centrífuga	Librería ArchestrA
Caldera	ArchestrA Symbol Editor
Turbinas	ArchestrA Symbol Editor

Tabla 6. Gráficos del Objeto: Dispositivos del Proceso

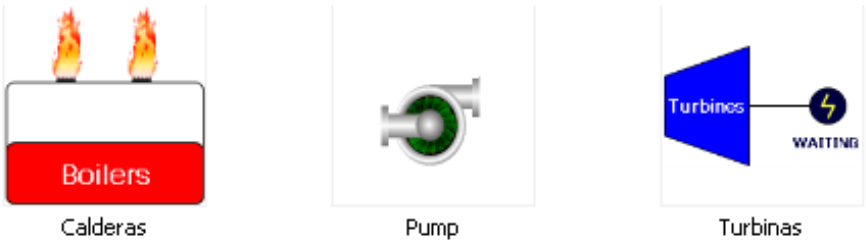


Figura 32. Elementos gráficos del Objeto: Dispositivos del Proceso

- Objeto: Elementos del Entorno
  En este se agrupan los elementos del entorno que representan fuentes de alimentación o salidas del proceso. En la tabla que recoge los elementos del proceso de cogeneración, se han identificado como etiquetas de proceso y bloques gráficos. En estos gráficos se ha intentado representar con la mayor fidelidad posible cada una de las situaciones ya que la interacción con ellos no es necesaria.

Elementos del Entorno	
Elemento	*Creación del Gráfico*
ATM	ArchestrA Symbol Editor
Compresor	Librería ArchestrA
Red Externa	Librería ArchestrA
Humos	Librería ArchestrA
Proceso Principal	Librería ArchestrA
Tanque de Gas Natural	Librería ArchestrA
Almacén	Librería ArchestrA
Azúcar	ArchestrA Symbol Editor
Camión de Transporte	Librería ArchestrA
Tanque de Agua	Librería ArchestrA

Tabla 7. Gráficos del Objeto: Elementos del entorno

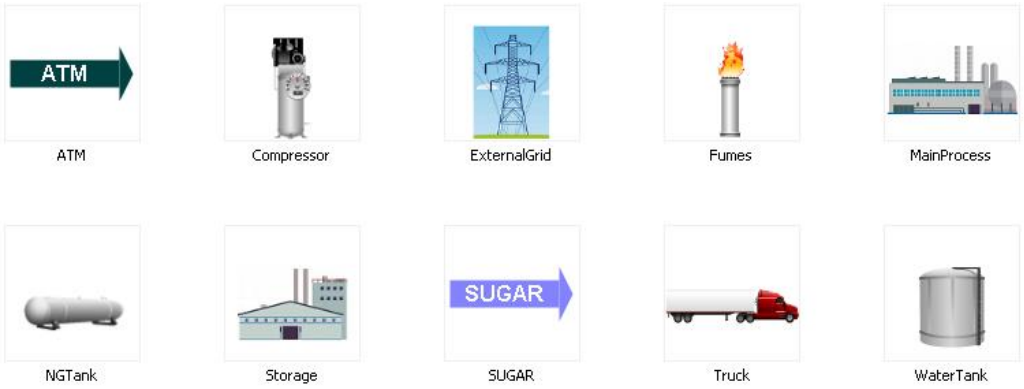


Figura 33. Elementos gráficos del Objeto: Elementos del entorno

- Objeto: Reguladores

  En este objeto se recogen los diferentes reguladores que se encargan de controlar el proceso de cogeneración. Con el objetivo de facilitar la identificación de cada tipo de regulador, se han asociado a cada uno de ellos una gama de colores que se asociará con su correspondiente transmisor. En este diseño se ha utilizado la nomenclatura y simbología que propone la ISA [20]. Para que el usuario conozca en todo momento el estado de los reguladores, estos presentarán un botón de interacción que permitirá al usuario accionar para obtener la información necesaria.

Reguladores	
Elemento	*Creación del Gráfico*
Regulador de Energía (JC)	ArchestrA Symbol Editor
Regulador de Presión (PC)	ArchestrA Symbol Editor
Regulador de Temperatura (TC)	ArchestrA Symbol Editor
Regulador de Peso (WC)	ArchestrA Symbol Editor

Tabla 8. Gráficos del Objeto: Reguladores

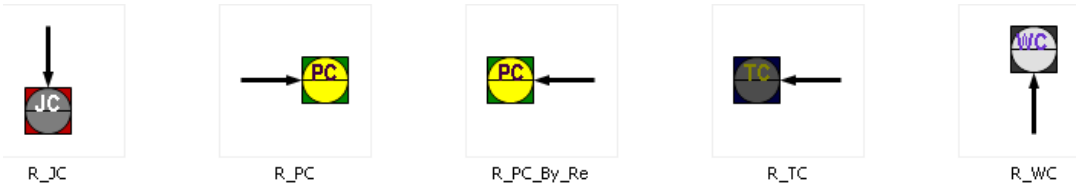


Figura 34. Elementos gráficos del Objeto: Reguladores

- Objeto: Controlador PI
  En este objeto se recoge la interfaz que proporción la librería de "ArchestrA" para interactuar con los reguladores de control. Este gráfico se ha asociado a los reguladores correspondientes para que el usuario pueda acceder a él pulsando sobre dicho regulador.

Controlador PI	
Elemento	*Creación del Gráfico*
Controlador PI	Librería ArchestrA

Tabla 9. Gráficos del Objeto: Controladores PI

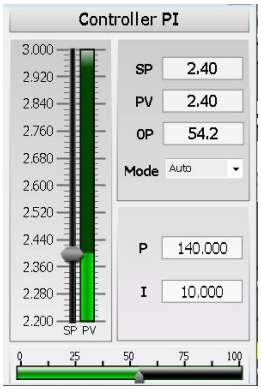


Figura 35. Elemento gráfico del Objeto: Controlador PI

- Objeto: Transmisores
  Este objeto recoge los transmisores asociados a cada uno de los reguladores de control. Como se ha mencionado en el anterior apartado, la gama de colores asociada a estos transmisores coincide con el regulador encargado de su control. La interacción de estos elementos con el usuario no es necesaria, por ello no se ha incorporado ninguna funcionalidad a los gráficos.

Transmisores	
Elemento	*Creación del Gráfico*
Transmisor de Energía (JC)	ArchestrA Symbol Editor
Transmisor de Presión (PC)	ArchestrA Symbol Editor
Transmisor de Temperatura (TC)	ArchestrA Symbol Editor
Transmisor de Peso (WC)	ArchestrA Symbol Editor

Tabla 10. Gráficos del Objeto: Transmisores

Figura 36. Elementos gráficos del Objeto: Transmisores

- Objeto: Válvulas
  Aquí se han agrupado dos tipos de válvulas, clasificándolas por su tipo de activación. En el proceso se tienen válvulas de regulación automática, que estarán controladas por algunos de los reguladores y una válvula de apertura y cierre manual.

Válvulas	
Elemento	*Creación del Gráfico*
Válvula Automática (JC/PC)	Librería ArchestrA
Válvula Manual	Librería ArchestrA

Tabla 11. Gráficos del Objeto: Válvulas

Figura 37. Elementos gráficos del Objeto: Válvulas

- Objeto: Visualización de Variables
  Para visualizar las variables que sean de interés para el proceso, se ha creado este objeto. Esta visualización se puede clasificar en variables (Véase Anexo III) que permiten la edición y lectura y en variables que solamente son de lectura. En este caso las variables que permiten la edición y lectura son las asociadas a los parámetros del modelo de simulación. La clasificación se ha realizado como se muestra a continuación:

Visualización de Variables (Lectura/Escritura)	
Elemento	*Creación del Gráfico*
ETu	ArchestrA Symbol Editor
PSBo	ArchestrA Symbol Editor
PSSaOutRef	ArchestrA Symbol Editor
TSBo	ArchestrA Symbol Editor
WBStIn	ArchestrA Symbol Editor
WBStOut	ArchestrA Symbol Editor

Tabla 12. Gráficos del Objeto: Visualización de Variables (Escritura/Lectura)

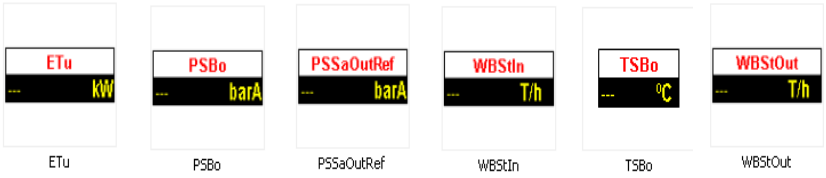


Figura 38. Elementos gráficos del Objeto: Visualización de Variables (Escritura/Lectura)

Visualización de Variables (Lectura)	
Elemento	*Creación del Gráfico*
ApBy	ArchestrA Symbol Editor
ApRe	ArchestrA Symbol Editor
Ep	ArchestrA Symbol Editor
Es	ArchestrA Symbol Editor
mSt	ArchestrA Symbol Editor
PIV	ArchestrA Symbol Editor
PSSaOut	ArchestrA Symbol Editor

Visualización de Variables (Lectura)	
Elemento	*Creación del Gráfico*
Qp	ArchestrA Symbol Editor
Wng	ArchestrA Symbol Editor
WSBo	ArchestrA Symbol Editor
WSRe	ArchestrA Symbol Editor
WSBy	ArchestrA Symbol Editor
WSTuIn	ArchestrA Symbol Editor
WWSa	ArchestrA Symbol Editor

Tabla 13. Gráficos del Objeto: Visualización de Variables (Lectura)

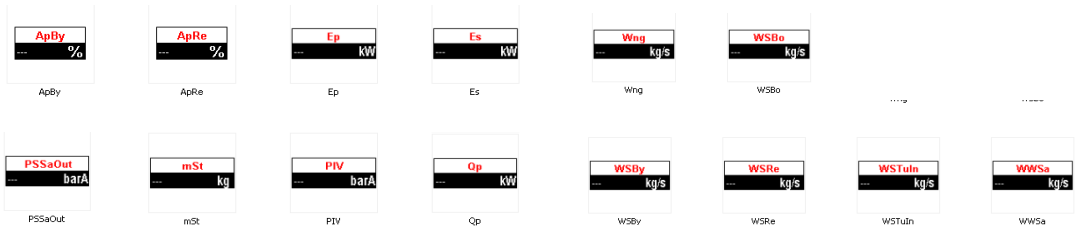


Figura 39. Elementos gráficos del Objeto: Visualización de Variables (Lectura)

Una vez definidos todos los objetos necesarios para el diseño de la interfaz, se puede realizar la sincronización con "InTouch". Esta, permitirá utilizar los objetos definidos como elementos gráficos para realizar el diseño de la aplicación gráfica, el resultado del diseño se puede ver en la Figura 40.

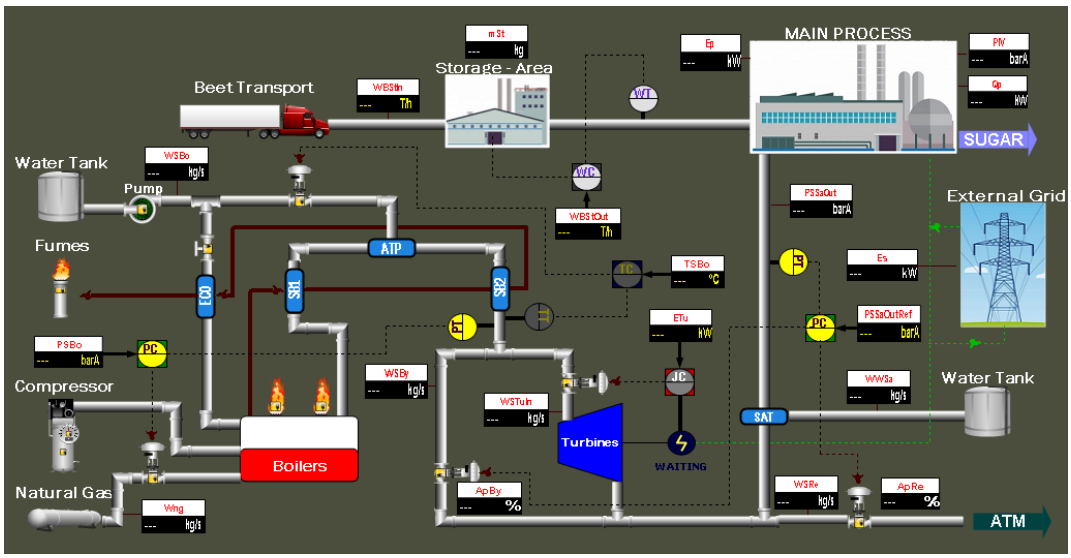


Figura 40. Resultado del diseño de la interfaz gráfica

Requisito 2: Visualización e interacción con las variables del proceso

Una vez realizada la comunicación con las plantillas OPC UA de cada servidor, se debe buscar que los paneles indicadores de las variables del proceso muestren los valores y se pueda interaccionar con estos. Para ello, se deben asociar las funcionalidades de *Value Display* e *User Input*, descritas en el paso 2 del apartado 3.2 de este capítulo. Asociando *Value Display,* se puede asegurar que el usuario no tendrá posibilidad de edición en el valor que se esté visualizando, por lo tanto, esta funcionalidad se asociará a las variables que solamente tienen permisos de lectura. Por otro lado, *User Input* se utilizará para las variables que tengan permisos de escritura, mostrando un teclado para la edición de estas como se muestra en la Figura 41.

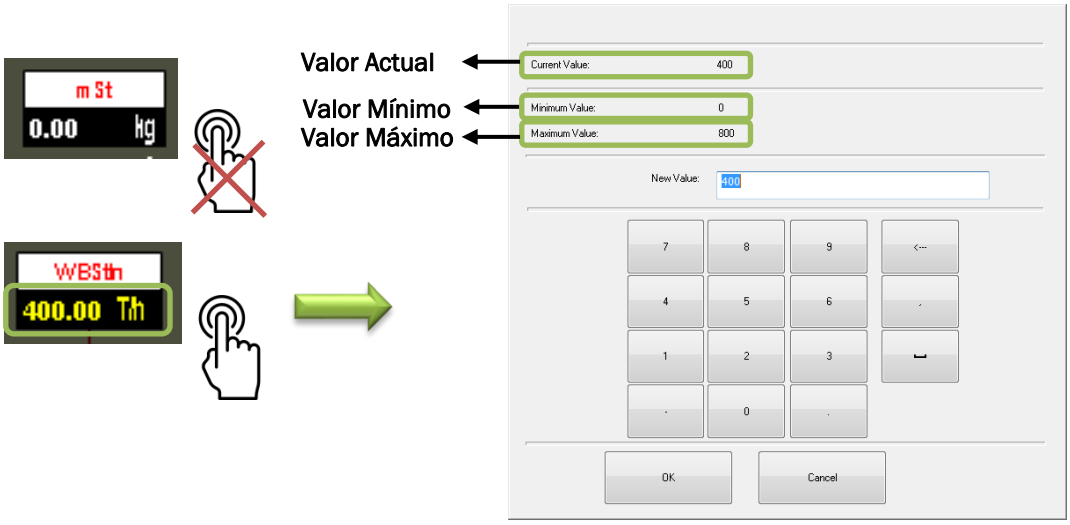


Figura 41. Interacción con las variables del proceso

Como se puede ver en la imagen, estos permisos se han clasificado utilizando el color blanco para variables de sólo lectura y el color amarillo para variables de lectura y escritura. Esta clasificación permitirá al usuario identificar rápidamente las variables que se pueden escribir en la interfaz. Hay que destacar que el teclado virtual que proporciona Wonderware es compatible con el teclado del ordenador, es decir, se puede realizar la escritura de la variable introduciendo los valores por el teclado del ordenador en el que se esté ejecutando la interfaz gráfica.

Requisito 3: Interacción con los reguladores de proceso

Otro de los requisitos del desarrollo de esta interfaz, es poder interactuar con los reguladores de control que se tienen en el proceso de cogeneración. Para ello se utilizará un script de acción, asociada a cada uno de los reguladores, con el objeto controlador PI. Con esto se conseguirá asociar un script al regulador y que, al ejecutar dicho script, se realice la apertura de una ventana que muestre el gráfico contenido en el objeto. Para realizar esta

tarea, se ha utilizado la funcionalidad "Action Script" junto con la correspondiente programación para la apertura de ventanas.

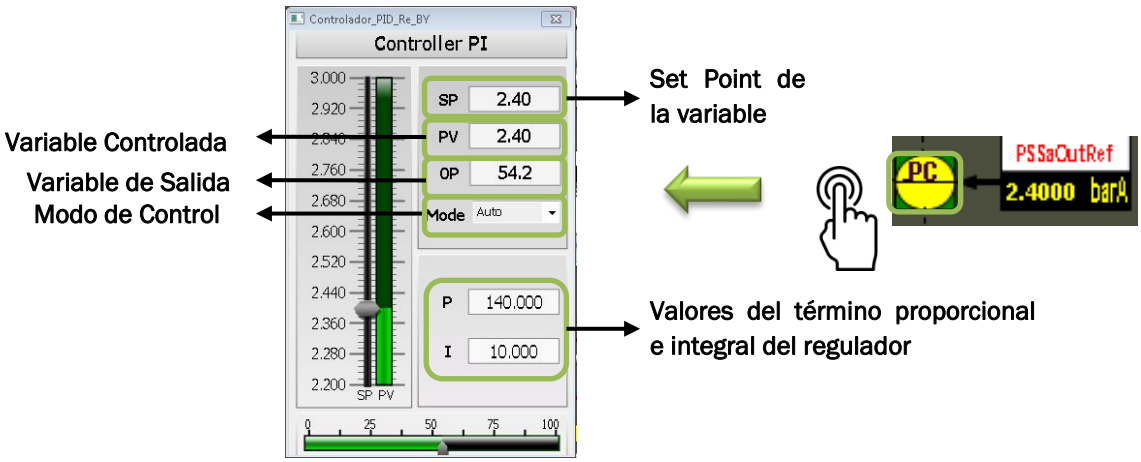


Figura 42. Interacción con los reguladores del proceso

Como se puede apreciar en la imagen, con el gráfico proporcionado por la librería "ArchestrA" se puede tener un control total sobre el regulador, por parte del usuario. Se puede realizar la modificación de los parámetros utilizando el slider de control o simplemente editando los valores correspondientes. También se puede establecer el valor necesario del término proporcional e integral del regulador.

Con la configuración realizada hasta ahora, se tiene una interfaz gráfica que representa el proceso de cogeneración, es capaz de visualizar las variables del proceso y editar los valores de las variables que se tenga permiso. Además, se puede acceder a los reguladores de control del proceso pudiendo configurar los parámetros de control más relevantes. Por lo tanto, el resultado final es el que se muestra en la Figura 43:

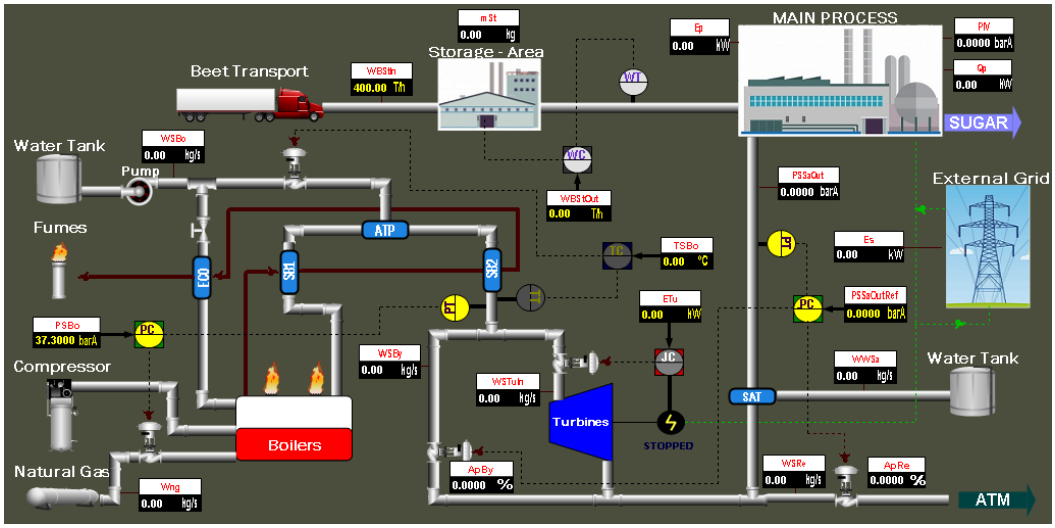


Figura 43. Resultado del diseño de la interfaz gráfica (II)

3.3.3. Desarrollo de la interfaz gráfica de simulación

La simulación es un proceso importante en este TFG, puesto que no se está realizando de forma real en una planta azucarera. Si esto se llevase a cabo en el proceso real no sería necesaria la simulación.

Hasta este apartado, el HMI solamente es capaz de visualizar los datos que le proporciona el servidor de la simulación y modificar las variables que permitan su edición.

Para manipular la simulación del proceso, se debe diseñar una interfaz que permita la interacción con el servidor para poder iniciarla, sin utilizar clientes externos. Además, para que el usuario tenga conocimiento del estado de la simulación y pueda tener control sobre la ejecución del servidor, se requieren paneles de estado, control e información que implementen estas funcionalidades. Una vez realizada la ejecución de la simulación, es necesario poder visualizar la evolución temporal de las variables, utilizando las gráficas de tiempo real, ya que si se realiza antes de la simulación, el valor indicado permanecería constante durante la visualización.

Para obtener un mejor control en la comunicación de la aplicación, se han creado avisos y manipulado los gráficos de manera que se ha obligado al usuario a seguir los pasos necesarios para el correcto funcionamiento de la simulación.

Todos los gráficos realizados para el diseño de la interfaz se han recogido en el objeto “simulación” mientras que los que tienen que ver con las gráficas de tiempo real, se han agrupado en el objeto “gráficas”. Con estos requerimientos se pueden establecer una serie de requisitos que permitan abordar la situación de una manera sencilla:

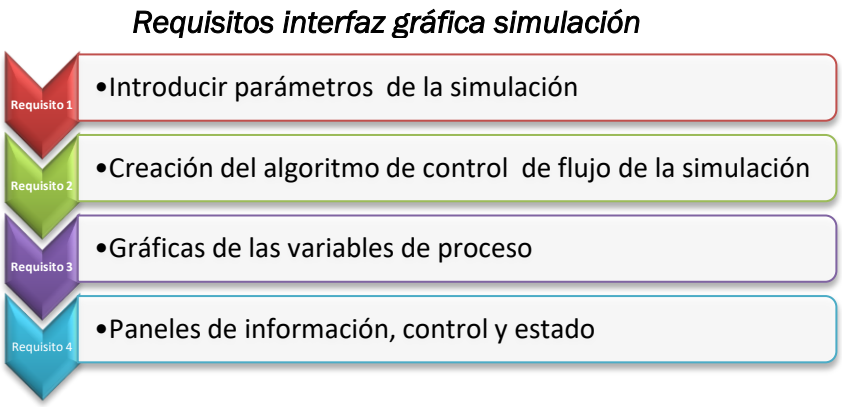


Figura 44. Requisitos para desarrollo de la interfaz gráfica de la simulación

A continuación, se explican cada uno de los requisitos a tener en cuenta para realizar el control de la simulación del proceso.

Requisito 1: Introducir parámetros de la simulación

Para poder activar la simulación mediante los métodos del servidor, ha sido necesario en este caso configurar el factor de aceleración de la simulación. El factor de aceleración que se ha utilizado se corresponde con el intervalo de comunicación, necesario para la ejecución de la simulación. Para facilitar su edición se ha diseñado un *slider* que ha permitido modificar el valor de este, dentro de un rango limitado. Una vez definido este parámetro, se ha diseñado un botón de activación capaz de iniciar la simulación. El resultado de lo descrito anteriormente se puede observar en la Figura 45.

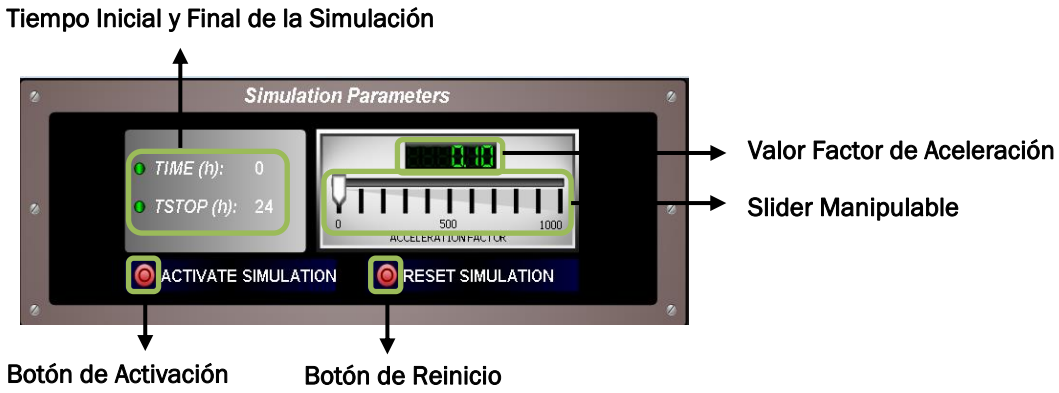


Figura 45. Ventana de parámetros de la simulación

Como se puede ver en el resultado del diseño, la ventana de parámetros contempla todos los elementos necesarios para iniciar la simulación, además de un botón de reinicio, el cuál reestablecerá los valores de todas la variables de proceso a un estado inicial. Se ha tratado de utilizar un diseño que permita al usuario identificar rápidamente la manera de interactuar con la ventana de parámetros. Cabe destacar que esta ventana tiene asociado un script *On Show,* que se ejecutará únicamente cuando se haya realizado la apertura de la ventana. Este script se encarga de cargar los valores iniciales en las variables del proceso, para que la simulación se realice correctamente.

El script está diseñado para que esta inicialización de variables se realice únicamente cuando se vaya a iniciar la simulación. Durante la simulación este script no tendrá ningún efecto, aunque la ventana se cierre y se muestre por primera vez.

Requisito 2: Creación del algoritmo de control de flujo de la simulación

El algoritmo de control de flujo de la simulación requiere una periodicidad adecuada, para ir obteniendo los datos que proporciona el servidor de la simulación. Para solucionar esta situación se ha utilizado un script asociado a la ventana principal de la aplicación gráfica del tipo “While Showing”. Este tipo de script es el más adecuado para desarrollar la tarea descrita. Por ello se ha

realizado la manipulación de un método que ejecuta la simulación hasta el siguiente intervalo de comunicación, garantizando la ejecución de la simulación.

La configuración de este script se muestra en la Figura 46. Es necesario ajustar el tiempo de ejecución, con el tiempo que se tarda en leer los datos del servidor, ya que, si el tiempo de ejecución es inferior, no se recogerán todos los datos que proporcionará el servidor.

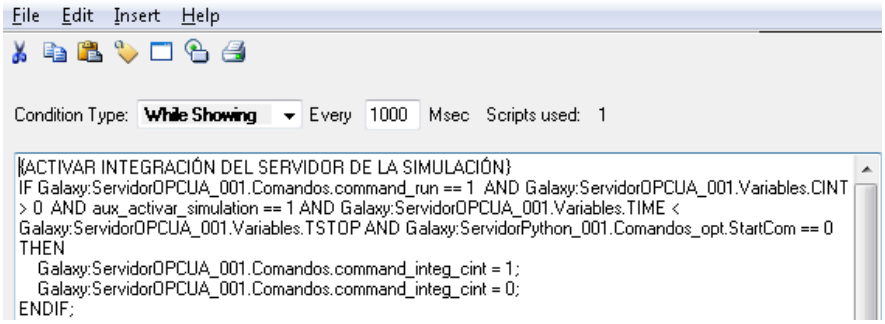


Figura 46. Código del script para ejecutar la simulación

La elección de asociar este script a la ventana principal de la aplicación se debe a que dicha ventana permanecerá activa y mostrándose al usuario en todo momento. Si esta ventana fuera cerrada por el usuario, la aplicación se cerraría, finalizando la comunicación con el servidor y por tanto el script no se podría ejecutar.

Requisito 3: Gráficas de las variables del proceso

La representación gráfica de la evolución temporal de las variables es necesaria en todas ellas. Por ello se han diseñado tantas gráficas como variables se están visualizando en la interfaz del proceso. La visualización de estas gráficas permanece oculta hasta que el usuario requiera de su activación. Esta activación está asociada a cada uno de los nombres que representan los paneles de visualización de las variables, como se muestra en la Figura 47:

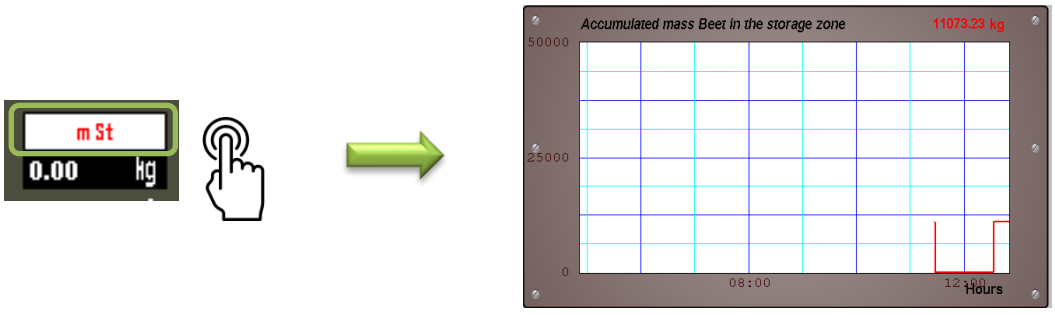


Figura 47. Visualización de las gráficas relativas a las variables del proceso

Requisito 4: Paneles de información, control y estado

Los paneles de información, control y estado de la simulación se han diseñado para facilitar la realización de estas tareas al usuario y conocer su estado en todo momento.

- Paneles de Información

  Los paneles de información se pueden clasificar en dos tipos, paneles de avisos al usuario y paneles de información de la aplicación. Los paneles de avisos al usuario se activan cuando el usuario no ha seguido la secuencia de pasos que se debe realizar para activar la simulación correctamente. Estos paneles mostrarán al usuario el procedimiento a seguir para que la activación de la simulación se realice de manera correcta. Estos, se muestran en la Figura 48. Ambos cuentan con un botón *Ok* que ocultará el aviso después de su activación.

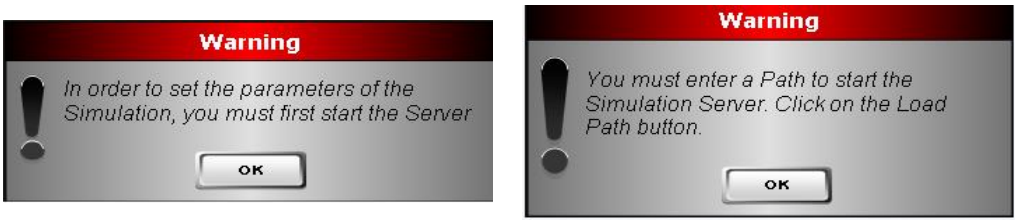


Figura 48. Paneles de aviso para el usuario

Para que el usuario tenga presente en todo momento los pasos que se deben seguir en el inicio de la simulación, se ha diseñado un panel de información de la aplicación. Este panel sugiere al usuario la secuencia de pasos que se debe seguir si se desea realizar una simulación correctamente. La Figura 49 muestra el panel descrito.

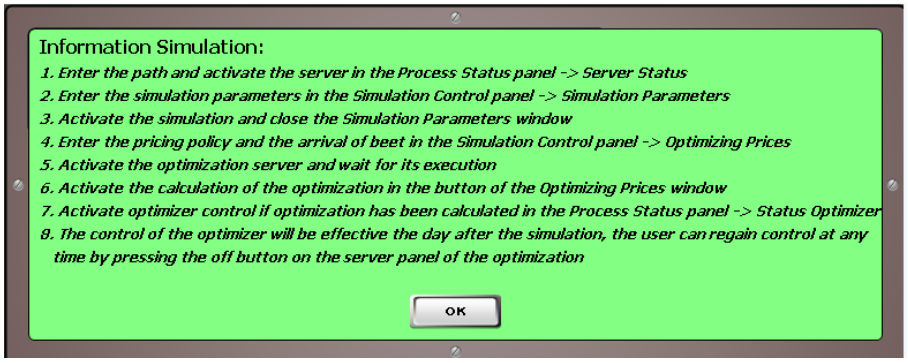


Figura 49. Panel de información para el usuario

Como en los paneles anteriores, se ha implementado el botón *Ok* que ocultará la ventana de información, para que el usuario pueda realizar los pasos descritos.

- Panel de Control

  Para que el usuario pueda activar el servidor desde la pantalla principal, se ha diseñado un panel de control que se encargue de realizar esta tarea. Para poder realizar una activación correcta, en primer lugar, se debe establecer una dirección, la cual sea el destino del servidor de la simulación. Para establecer la ruta del servidor, se ha diseñado un botón que active la ventana en la cual se introducirá esta. Si la ruta es correcta se tendrá un botón de activación para que se ejecute el archivo asociado al servidor. Si todo se ha realizado correctamente, el panel de control contará con indicador de texto que lo confirmará. Para mayor seguridad de la comunicación con el servidor, se ha implementado un botón que activa una ventana con los parámetros más significativos de la conexión. Esta ventana permitirá obtener la información necesaria para que el usuario pueda detectar la correcta o incorrecta activación del servidor. Esta situación se ve representada en la Figura 50.

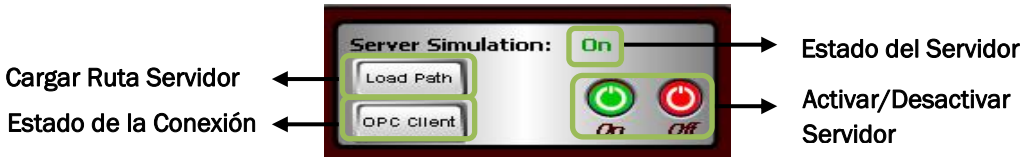


Figura 50. Panel de control del servidor de la simulación

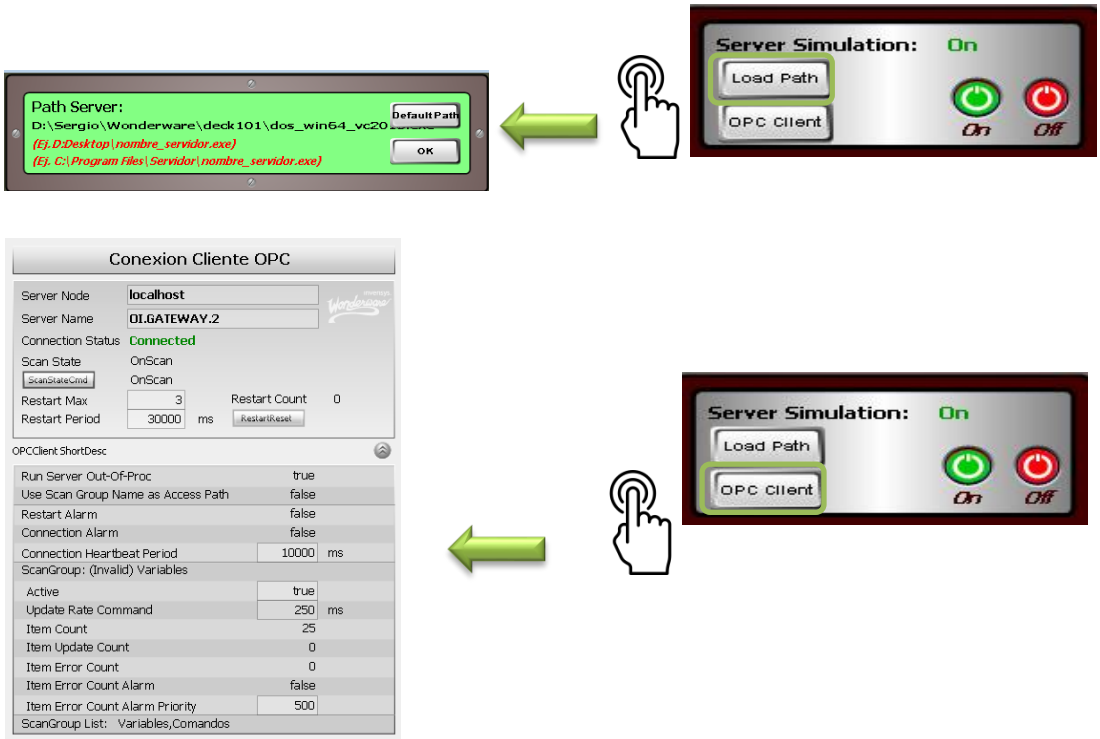


Figura 51. Funcionalidades del panel de control de la simulación

Con esta implementación el usuario será capaz de tener control total sobre la comunicación del servidor.

- Paneles de Estado

  Estos paneles se encargarán de mostrar la información necesaria, para que el usuario sepa en todo momento la situación en la que se encuentra cuando inicie la ejecución de la simulación y destacar los valores de ciertas variables del proceso de cogeneración. Esta información se ha representado utilizando sliders, que no permitan su manipulación, pero que se vayan actualizando en función de los valores que están representando. Estos se han situado en la pantalla principal, donde el usuario tenga acceso a su visualización en cualquier instante. Se han diseñado tres paneles, dos de ellos mostrarán la evolución temporal de dos variables índices de rendimiento del proceso de cogeneración y uno que mostrará el progreso de la simulación en forma de porcentaje. Estos se muestran la Figura 52.

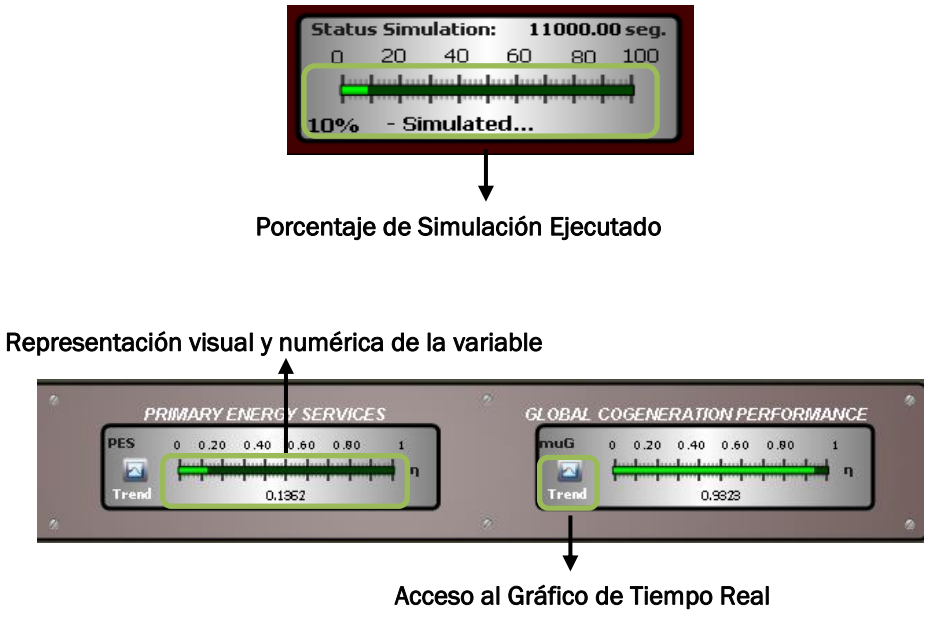


Figura 52. Paneles de información y estado de la ejecución de la simulación

Con el fin de establecer una organización que permita una mejor visualización de las opciones que contiene la interfaz gráfica, se han agrupado los paneles de estado y control relativos al servidor en un panel de control general. Mientras que los paneles de estado de los índices de rendimiento de las variables se han agrupado en otro panel.

Para ello se han diseñado unos marcos específicos para cada panel, que resalten la importancia de cada uno y mejoren el aspecto visual de la aplicación. De esta manera se tendrá todo agrupado en la pantalla principal. Para el acceso a la ventana de parámetros se ha implementado

un menú, que proporcione la posibilidad de abrir dicha ventana y que proporcione información sobre el funcionamiento de la aplicación. Los resultados de esta interfaz gráfica se pueden ver en la Figura 53 que recoge todas las funcionalidades de la interfaz de simulación.

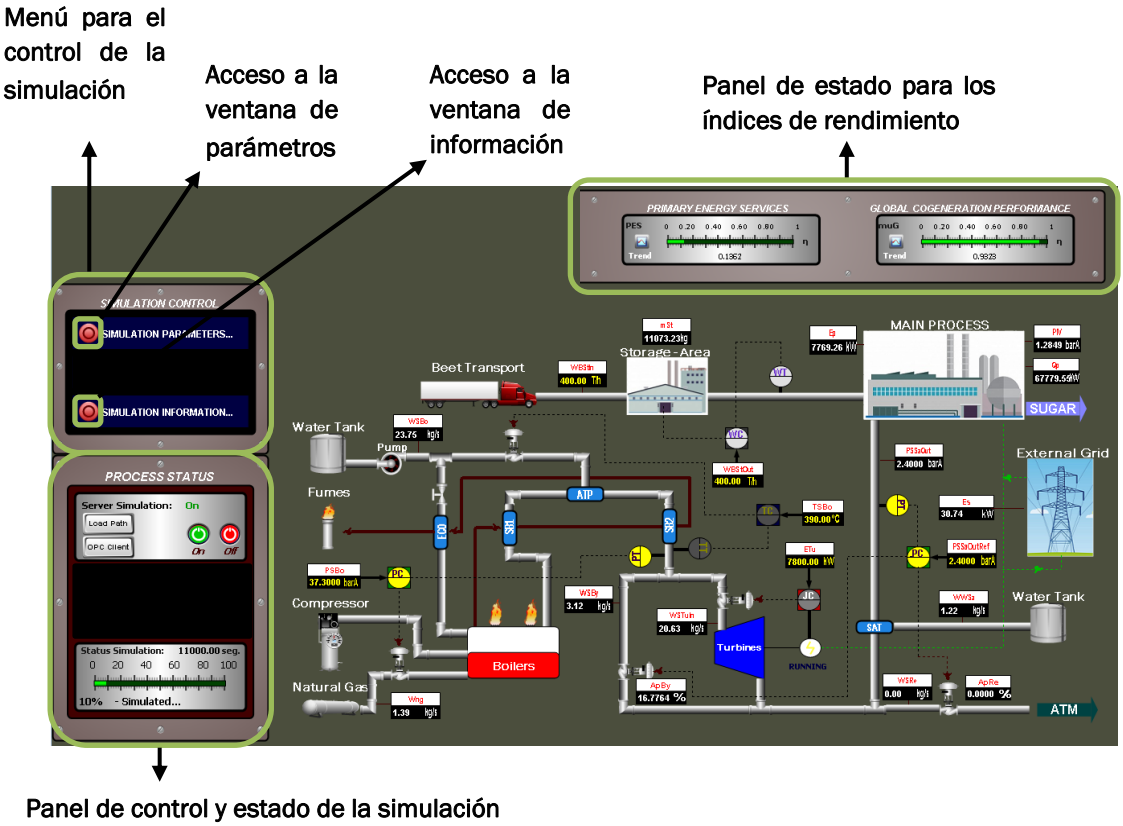


Figura 53. Resultado del diseño de la interfaz gráfica (III)

En conclusión, gracias a este proceso de simulación, se puede tener control total de la ejecución de esta, así como visualizar la evolución de las variables numéricas y gráficas de forma simultánea.

3.3.4. Desarrollo de la interfaz gráfica de optimización

La optimización calcula sus resultados en función de una política de precios del mercado eléctrico y la llegada de remolacha al proceso. Por lo tanto, se debe diseñar una ventana que sea capaz de establecer los valores anteriores, además de ser capaz de enviar la activación del optimizador para que empiece a realizar los cálculos necesarios. Para la creación de los gráficos, se ha definido un objeto *optimization* que recogerá todos los gráficos relativos a esta.

Como en los anteriores apartados se ha establecido una serie de requisitos que se deben abordar para realizar el control de la optimización.

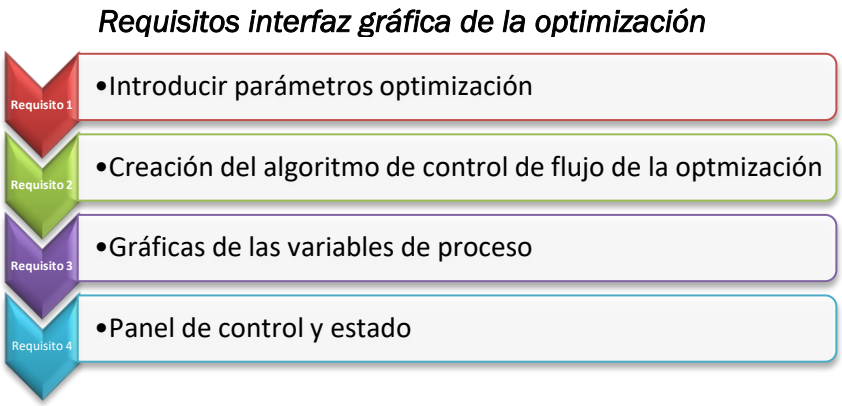


Figura 54. Requisitos para el desarrollo de la interfaz gráfica de la optimización

Requisito 1: Introducir parámetros optimización

Para la gestión de precios y llegada de remolacha, Wonderware cuenta con un elemento gráfico llamado "ListBox". Este elemento representa los datos ingresados en forma de lista de manera que el usuario pueda visualizar todos los datos añadidos de una manera sencilla e intuitiva. Para que el control sobre la lista sea total, se han diseñado varios botones que implementan las funcionalidades necesarias, como son, eliminar un precio, eliminar la lista, ingresar un solo precio o ingresar unos precios de una base de datos externa.

Estas funcionalidades se han conseguido utilizando scripts de acción en los botones y asociando métodos que dispone Wonderware para el tratamiento vía software de estas listas. El aspecto final de la ventana diseñada se corresponde con la Figura 55.

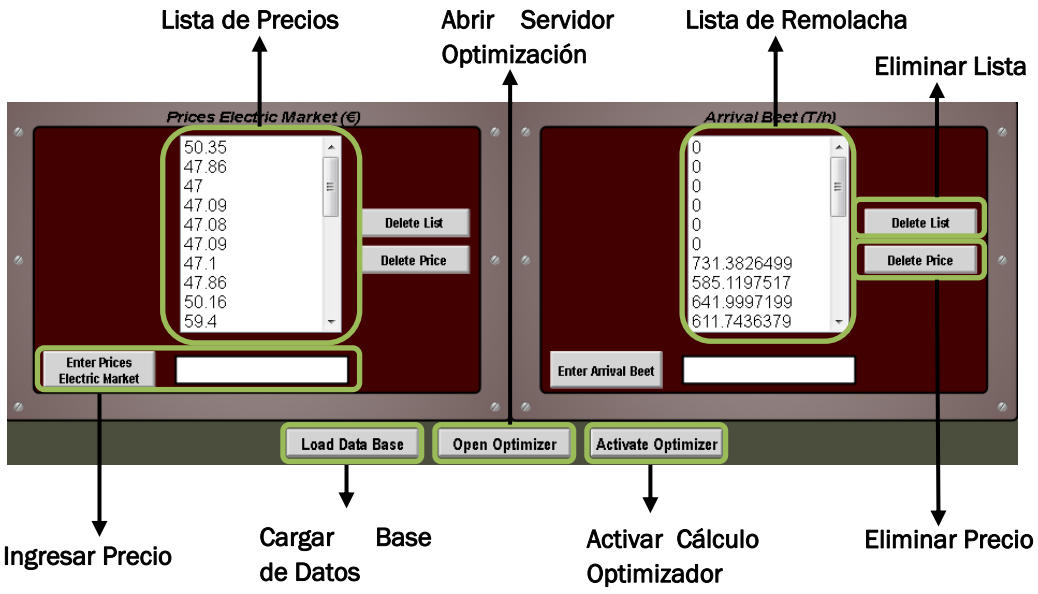


Figura 55. Ventana para introducir el escenario de precios y la llegada de remolacha

La ventana que contiene al gráfico descrito anteriormente, tiene asociada dos tipos de script, que permiten comunicarse con la hoja Excel en la que se deben escribir los precios y la llegada de remolacha. Se han implementado dos script para diferenciar si el usuario introduce los precios desde la interfaz gráfica o desde la base de datos. La base de datos que se dispone se encuentra clasificada por días en dos archivos Excel, por ello al pulsar el botón "Load Data Base" se abrirá la ventana indicada para que el usuario introduzca el día que desea cargar y se cargará automáticamente en la lista de precios y llegada de remolacha, como se puede ver en la Figura 56.

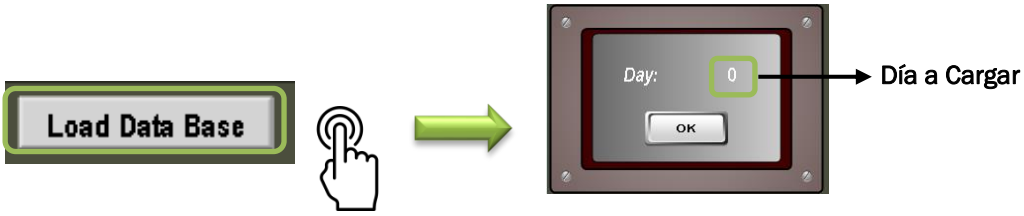


Figura 56. Acceso a la base de datos de precios y llegada de remolacha

Por lo tanto la interacción del usuario con esta ventana, está relacionada directamente con el archivo Excel que recoge el listado de precios y la llegada de remolacha. Por motivos de programación de los script para leer y escribir en Excel, es necesario que el archivo al que se desea acceder esté abierto en el instante de la lectura o la escritura. Si no estuviera abierto, la escritura en el archivo no se realizaría correctamente.

Para que el usuario pueda tener la contabilidad de los días que van pasando durante la ejecución de la simulación, se ha diseñado un calendario que sea capaz de realizar esta función. Como la simulación se va a ejecutar cíclicamente durante 86400 segundos (1 día), transcurrido ese tiempo, el calendario sumará un día en su resultado. Es por ello que el algoritmo que se encarga de controlar los días que marcará el calendario, se ha añadido al script de la venta principal. El diseño realizado para el calendario es el mostrado en la Figura 57. Este calendario se mostrará en la ventana principal de la aplicación donde se pueda visualizar rápidamente el día en el que se encuentra la simulación, junto a los paneles de estado de la simulación, donde su visibilidad sea notable.


```
{CONTADOR DE DÍAS}
IF Galaxy:ServidorOPCUA_001.Comandos.command_run == 1  AND Galaxy:ServidorOPCUA_001.Variables.TIME
> 0 AND aux_activar_simulation == 1 AND Galaxy:ServidorOPCUA_001.Variables.TIME == 86400 THEN
        Galaxy:ServidorOPCUA_001.Variables.TIME = 0;
        IF contador_dias_unidades == 10  THEN
          contador_dias_unidades = 0;
          contador_dias_decenas = contador_dias_decenas+1;
        ELSE
          contador_dias_unidades = contador_dias_unidades + 1;
        ENDIF;
ENDIF;
```

Figura 57. Calendario diseñado y script asociado

Requisito 2: Creación del algoritmo de control de flujo de la optimización

Para que se produzca la activación del cálculo de la optimización y que se establezca la comunicación directa con el servidor de la optimización, es necesario desarrollar un algoritmo en un script asociado a una ventana. Para su implementación se utilizará la ventana principal como se hizo en la simulación, ya que será la que esté en ejecución en todo momento. Como en la ventana principal también se implementó el algoritmo de control de la simulación, se deberán tener en cuenta las variables utilizadas para su implementación. Si la comunicación del optimizador está activa, la integración de la simulación se detendrá dejando el control de esta al optimizador.

El algoritmo desarrollado es capaz de detectar 4 situaciones posibles que se pueden producir durante la ejecución de la simulación:

- Situación 1: El usuario durante la simulación, después de haber realizado el cálculo de la optimización, decide delegar el control al optimizador. El algoritmo es capaz de detectar esta situación, y transcurrido un día después de su activación se establecerá la comunicación con el servidor de la optimización y el control será suyo.
- Situación 2: El usuario durante la simulación, decide realizar el cálculo de la optimización, pero los resultados no son los esperados y decide no delegar el control al optimizador. Este algoritmo detectará esta situación y la simulación seguirá ejecutándose sin realizar ningún cambio.
- Situación 3: El usuario durante el control del optimizador, decide para la comunicación y retomar el control de la simulación. El algoritmo detectará la situación y la comunicación con el servidor de la optimización se detendrá, retomando el control el algoritmo de la simulación.
- Situación 4: El usuario durante el control del optimizador, decide realizar otra optimización y aplicar los resultados obtenidos al día siguiente. Esta situación también está contemplada por el algoritmo, por lo tanto, podría realizarse sin dar lugar a problemas.

La implementación del algoritmo sería la que se muestra en la Figura 58:

```
{ACTIVAR COMUNICACIÓN CON EL SERVIDOR DE LA OPTIMIZACIÓN}
IF Galaxy:ServidorOPCUA_001.Variables.TIME == 86400 AND aux_activar_optimization == 1  AND
Galaxy:ServidorPython_001.Comandos_opt.StartCom == 0 THEN
        Galaxy:ServidorPython_001.Comandos_opt.StartCom = 1;
ENDIF;
```

→ Establece Comunicación con el servidor

```
{INICIALIZA VARIABLES DESPUÉS DEL CONTROL DEL OPTIMIZADOR}
IF Galaxy:ServidorOPCUA_001.Variables.TIME > 0 AND Galaxy:ServidorOPCUA_001.Variables.TIME == 86400
AND aux_activar_optimization == 1  AND Galaxy:ServidorPython_001.Comandos_opt.StartCom == 1 THEN
        aux_activar_optimization = 0;
        aux_estado_optimizador = 0;
ENDIF;
```

→ Ha pasado un día después de realizar el control el Optimizador

```
{INICIALIZA VARIABLES SINO SE HA INICIADO EL CONTROL DEL OPTIMIZADOR}
IF Galaxy:ServidorOPCUA_001.Variables.TIME > 0 AND Galaxy:ServidorOPCUA_001.Variables.TIME == 86400
AND aux_estado_optimizador == 1  AND Galaxy:ServidorPython_001.Comandos_opt.StartCom == 0 THEN
        aux_activar_optimization = 0;
        aux_estado_optimizador = 0;
ENDIF;
```

→ No se ha activado el control del Optimizador

Figura 58. Código del algoritmo de control de flujo de la optimización

Con este algoritmo se pueden abordar los requisitos de control que se presentaban inicialmente, proporcionando al usuario un control total sobre el servidor de la optimización y sobre los resultados aportados.

Requisito 3: Creación de Gráficas

Para que el usuario pueda verificar los resultados proporcionados por el optimizador, se han creado las gráficas en Excel. Estas gráficas muestran las consignas calculadas por el optimizador de las variables controladas del modelo, como se puede ver en la Figura 59:

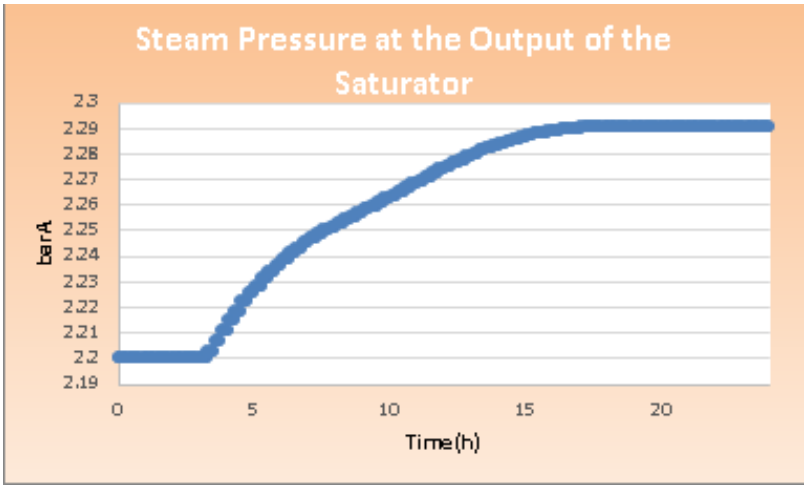


Figura 59. Gráfica en Excel de los resultados de la optimización

Estas gráficas servirán al usuario para verificar si los resultados obtenidos por el optimizador son correctos. Representan la evolución temporal de las consignas que se han calculado, valores que después tomarán los parámetros del modelo si el usuario acepta el control del optimizador.

Requisito 4: Panel de control y estado

Como en el desarrollo de la interfaz de la simulación del proceso, se ha diseñado un bloque que sea capaz de mostrar al usuario la interacción con el servidor de la optimización. Este bloque será capaz de mostrar:

- Si el optimizador está disponible para realizar los cálculos necesarios,
- Si el optimizador está realizando los cálculos,
- Si se ha encontrado la solución con los datos introducidos
- Si ha llegado a tomar el control una vez transcurrido el día después de su activación.

Para seguir con la organización de los gráficos inicial, este panel se ha introducido en el panel de control que engloba también los bloques de la simulación, agrupando los controles de los servidores, para facilitar la interacción con el usuario. El diseño realizado se corresponde con el de la Figura 60.

Figura 60. Panel de control y estado del servidor de la optimización

Cabe destacar en este bloque, que los botones de *On* y *Off* activan y desactivan el control del servidor de la optimización. No se debe confundir con el funcionamiento de los botones asociados al panel del servidor de la simulación, ya que estos no realizan la apertura de ningún archivo. Cuando el usuario decida delegar el control del sistema al optimizador y pulse el botón *On*, el control del optimizador se hará efectivo al día siguiente después de su activación. El control del optimizador puede ser cancelado en cualquier momento pulsando el botón *Off*, que finalizará la comunicación con el optimizador y devolverá el control del sistema al usuario.

El resultado final de la aplicación se puede apreciar en la Figura 61, que recoge cada uno de los desarrollos realizados anteriormente.

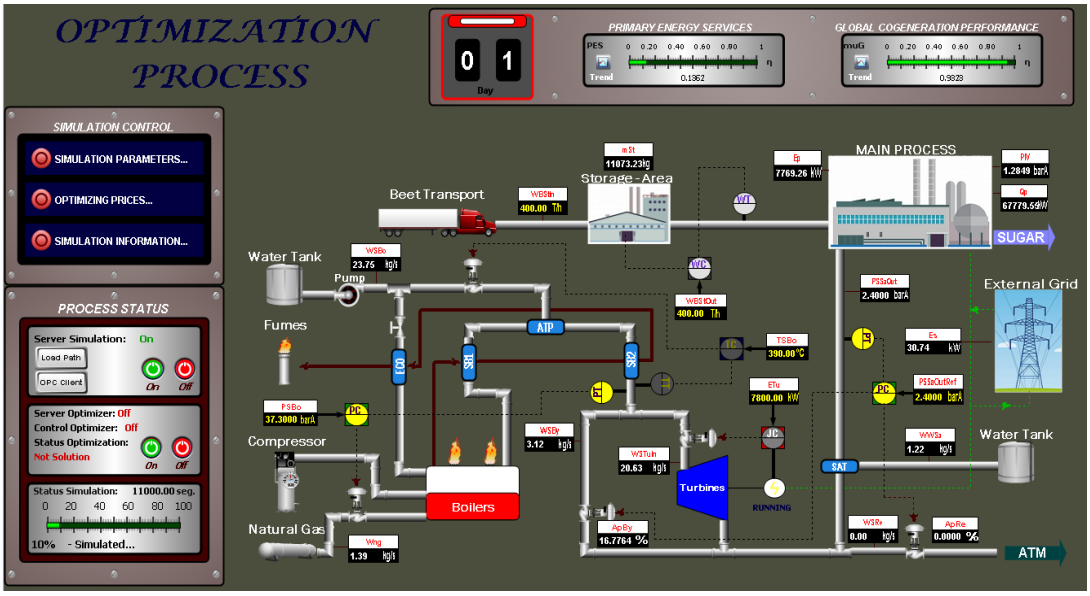


Figura 61. Resultado del diseño de la interfaz gráfica (IV)

Como se puede apreciar en la imagen, la interfaz gráfica quedaría definida completamente, integrando los requisitos necesarios para establecer la comunicación y el control entre el módulo de la optimización y el módulo de la simulación.

CAPÍTULO 4

Funcionamiento de la aplicación

4.1. INTRODUCCIÓN

En este capítulo se pretende validar el funcionamiento de la aplicación diseñada, verificando que se cumplen todos los requisitos propuestos en el apartado anterior. Para realizar dicha verificación se ha organizado una batería de pruebas que contemplan las posibilidades que puede realizar el usuario cuando utilice la aplicación. En primer lugar, se debe establecer la secuencia de funcionamiento para interactuar con la aplicación, una vez definida se realizará la batería de pruebas que verificará la validez de la aplicación. La secuencia de funcionamiento ya ha sido implementada en la interfaz gráfica mediante un panel de información, a continuación, se va a realizar una explicación detallada de cada uno de los pasos a seguir, aportando las imágenes necesarias para su comprobación. Para verificar el funcionamiento del optimizador, se ha realizado la simulación de un día de control del optimizador, analizando las gráficas obtenidas en Excel y la evolución temporal del proceso en el HMI.

Paso 1: Activación del servidor de la simulación

En este primer paso se debe activar el servidor correspondiente a la simulación, para ello se debe introducir la ruta de acceso, donde se encuentre dicho servidor.

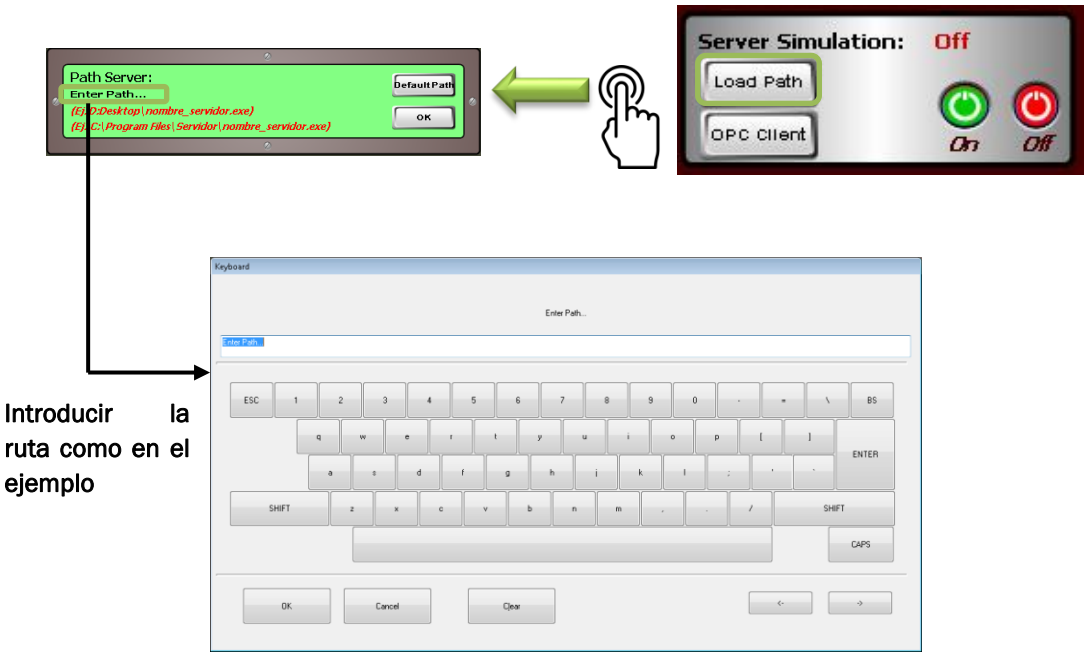


Figura 62. Ruta para la activación del servidor de la simulación

Si la ruta se ha introducido correctamente, se deberá pulsar el botón *On* que internamente se encargará de acceder a la ruta, y abrir el archivo que contiene al servidor de la simulación. La apertura correcta del servidor se notificará en el panel como se muestra en la Figura 63:

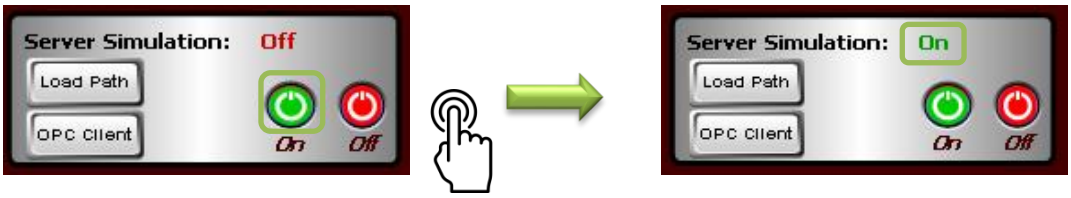


Figura 63. Activación del servidor de la simulación

En este punto, se tendría iniciada la comunicación con el servidor de la simulación. Por lo tanto, se comenzará a visualizar en los paneles indicadores, los valores iniciales de las variables que proporciona el servidor.

Paso 2: Introducir los parámetros para activar la simulación

Cuando se haya establecido correctamente la comunicación con el servidor de la simulación, se deberá introducir el factor de aceleración. Para configurar este parámetro se debe pulsar el botón *Simulation Parameters…* este realizará la apertura de la ventana de parámetros. Esta situación se muestra en la Figura 64:

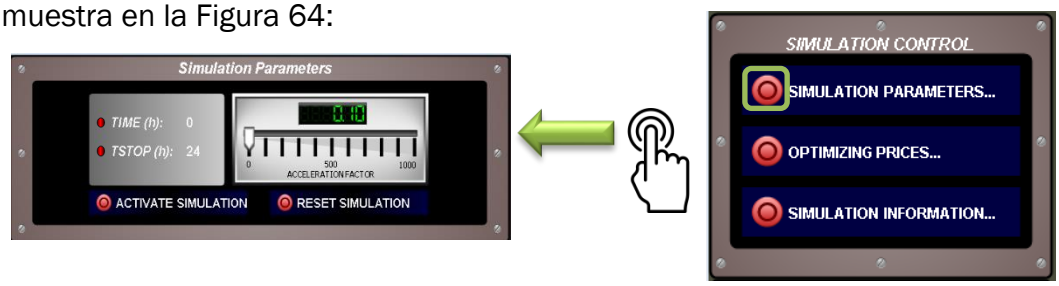


Figura 64. Acceso a la ventana de parámetros del servidor de la simulación

Como se puede observar en la imagen, el factor de aceleración tiene un valor configurado por defecto, este valor se obtiene al realizar la precarga de los parámetros para iniciar la simulación. Es importante destacar que el factor de aceleración tiene que tener un valor superior a 0 para iniciar la simulación, si este valor fuera 0 el botón de activación de la simulación no realizará su cometido. Después de la configuración del parámetro de aceleración se debe pulsar en el botón *Activate Simulation* y la simulación comenzará a ejecutarse:

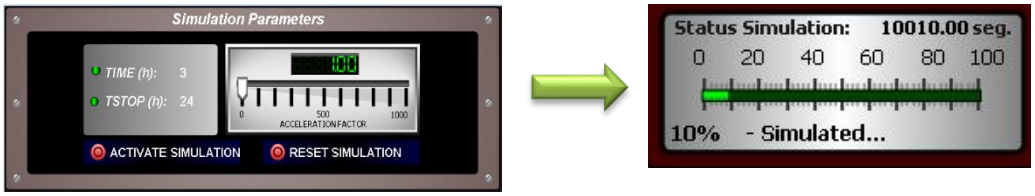


Figura 65. Verificación de la ejecución de la simulación

Para que el usuario pueda verificar el inicio de la simulación en la ventana de parámetros, como se puede ver en la Figura 65, los indicadores al lado de *TIME* y *TSTOP* pasarán de rojo a verde. Este color indicará que la simulación ha comenzado su ejecución, además de que el indicador numérico del tiempo comenzará a avanzar, mostrando el tiempo transcurrido en horas. Otra forma de comprobar el inicio de la simulación es consultar el panel de estado de la simulación, en este se podrá comprobar el porcentaje de simulación ejecutado, así como el tiempo en el que se encuentra la simulación en segundos.

En este paso se tendría iniciada la simulación del proceso, en la cual el usuario tiene el control total del sistema, pudiendo establecer valores de consigna adecuados en el proceso y consultar las gráficas de tiempo real en cualquier momento.

Paso 3: Introducir la política de precios y llegada de remolacha

Para introducir la política de precios y la llegada de remolacha se debe pulsar el botón *Optimizing Prices…*. Este botón abrirá la ventana que mostrará la interfaz gráfica para la optimización. Esta situación se muestra en la Figura 66.

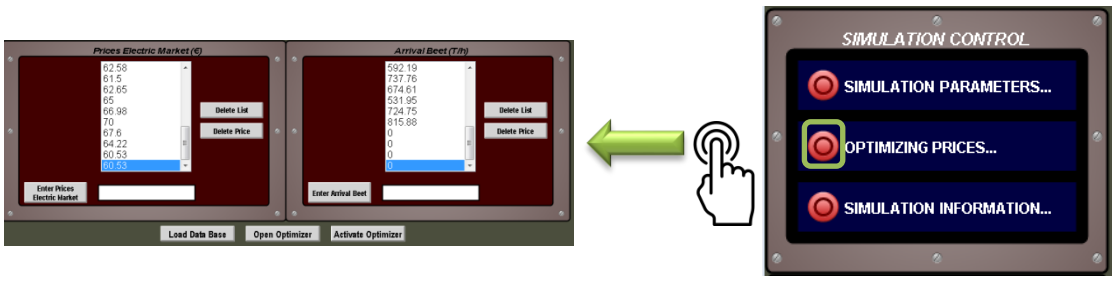


Figura 66. Acceso a la ventana de parámetros del servidor de la optimización

Inicialmente las listas de precios y de llegada de remolacha muestran el contenido del archivo Excel donde se escriben estos. Estos datos se corresponden con los últimos valores de precios y llegada de remolacha introducidos por el usuario en otra ocasión. Si se desea introducir unos valores diferentes, se debe eliminar cada una de estas listas pulsando en el botón *Delete List,* se obtendrá la imagen de la Figura 67.

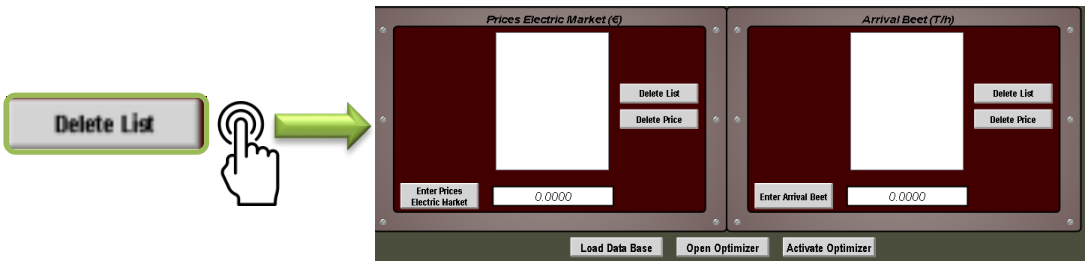
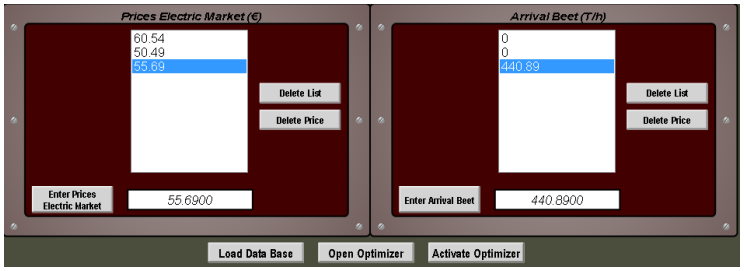


Figura 67. Manipulación de la lista de precios y remolacha

Como se puede observar en la imagen, las listas se han vaciado permitiendo ingresar nuevos valores en ambas. Para ingresar el precio en cada una de ellas, se debe pulsar en el panel donde se muestra el valor 0.000. Esto generará la visualización del teclado virtual de la aplicación y permitirá introducir el valor deseado. Una vez introducido el valor se debe pulsar en el botón *Enter Prices Electric Market* para ingresar los precios del mercado eléctrico y *Enter Arrival Beet* para ingresar la llegada de remolacha, según corresponda en cada caso. Automáticamente al ingresar el valor correspondiente en la lista, este se cargará en la hoja de Excel que se encarga de recoger estos valores. Esta situación se puede observar en la Figura 68:



A	B
Pe	WBStIn
60.54	0
50.49	0
55.69	440.89

Figura 68. Verificación de la escritura en Excel de los precios y la llegada de remolacha

Otra forma de ingresar los precios y la llegada de remolacha será cargar estos mediante una base de datos. Simplemente se debe pulsar sobre el botón *Load Data Base* y se abrirá la ventana emergente que corresponde al día que se desea cargar. Introduciendo el valor del día deseado (desde 1 hasta 101 días) se cargarán automáticamente los valores del día deseado en la interfaz gráfica de la optimización y en el Excel que recoge los valores necesarios para la optimización. Cabe destacar, que es necesario introducir 25 valores en cada una de las listas para poder pasar al siguiente paso. Esto se debe a que el optimizador, necesita 25 valores para poder realizar los cálculos necesarios.

Paso 4: Activar el servidor de la optimización

Cuando se han introducido correctamente los valores de los precios y de la llegada de remolacha, se debe activar el servidor de la optimización. La activación de éste se realiza desde la ventana dedicada a la interfaz gráfica de la optimización. En esta ventana se tiene un botón *Open Optimizer* que al pulsarlo ejecutará el servidor de la optimización utilizando la consola de Windows. La apertura de este servidor es muy diferente a la apertura del servidor de la simulación, ya que este está implementado en Python. Debido a la imposibilidad de crear un archivo ejecutable para este servidor (archivo .exe) por la cantidad de librerías necesarias, se realiza la apertura de este desde la ventana de comandos de Windows, accediendo al archivo Python que lo contiene. Esta situación se muestra en la Figura 69.

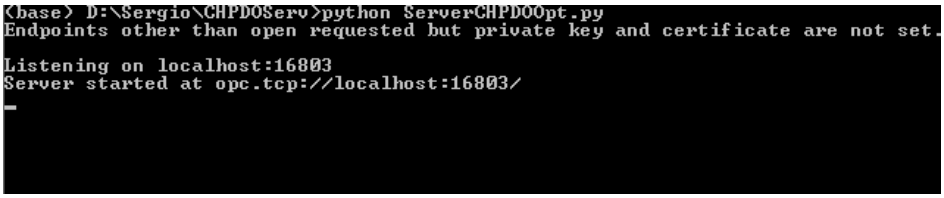


Figura 69. Activación del servidor de la optimización

Como en el servidor de la simulación, se puede comprobar la activación de este en el panel de control y estado del servidor de la optimización. Este panel mostrará en la información del estado del servidor *On,* mostrando que la comunicación con este se ha realizado correctamente, como se puede ver en la Figura 70.

Figura 70. Verificación de la activación del servidor de la optimización

En esta se puede observar como el estado del servidor se encuentra activo, el control del optimizador está apagado y el optimizador no tiene solución, porque no se han cargado los precios y la llegada de remolacha todavía.

Paso 5: Activar el cálculo del optimizador

Si el servidor ha sido activado correctamente, se podrá establecer la comunicación con este para que realice los cálculos necesarios. Es importante que el servidor se encuentre activo, ya que sino la secuencia de pasos que se está describiendo no tendría unos resultados correctos. Para que el servidor de la optimización comience a realizar los cálculos se de pulsar el botón *Activate Optimizer*. Este botón enviará la información necesaria al servidor para indicarle que debe comenzar a calcular utilizando los valores de los precios y la llegada de remolacha implementados anteriormente. La Figura 71 se encarga de mostrar la situación descrita.

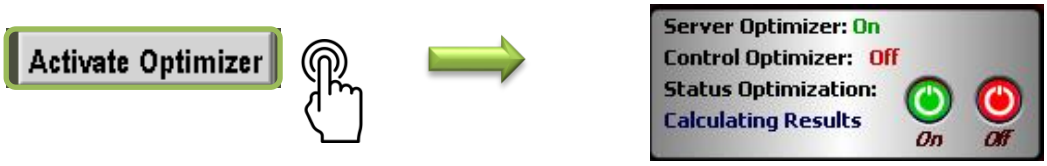


Figura 71. Activación del cálculo del optimizador

Como se puede observar en el panel de control y estado, el texto indicador del estado del cálculo de la optimización muestra el mensaje *Calculating Results.* Este mensaje informará al usuario de que el cálculo de la optimización se está realizando. También se puede apreciar, como el texto indicador del control del optimizador, se encuentra en *Off,* indicando que el usuario aún no ha delegado el control del proceso al optimizador. Cuando el cálculo de la optimización finalice, se notificará al usuario por medio del panel mostrando la información de la Figura 72.

Figura 72. Verificación de la finalización del cálculo del optimizador

Cuando se obtenga la solución a los precios y la llegada de remolacha introducida anteriormente, el usuario podrá observar los resultados del cálculo en las gráficas de Excel. Estas gráficas se encuentran en otra hoja diferente donde se han almacenado la política de precios y llegada de remolacha. De esta forma el usuario podrá evaluar las consignas que se irán estableciendo en el proceso, para valorar si los resultados que se están visualizando son correctos o no. Estas gráficas se muestran en la Figura 73.

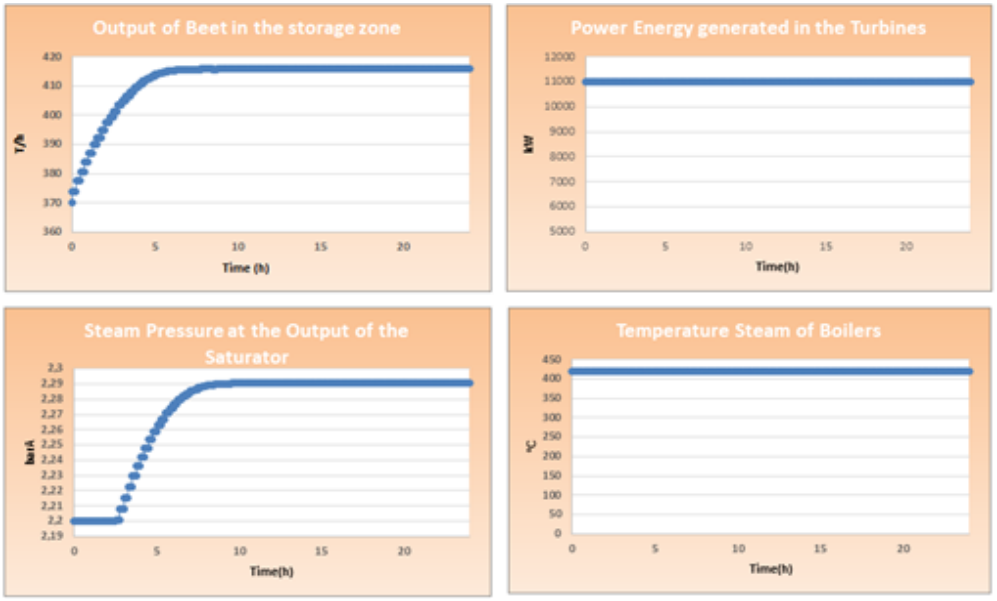


Figura 73. Gráficas de los resultados del cálculo del optimizador

Paso 6: Activar el control del optimizador

Si el usuario verifica los resultados obtenidos por el optimizador y decide aplicar esos resultados, deberá comunicarlo utilizando el panel de control y estado del servidor de la optimización. Para ello deberá pulsar en el botón *On* que muestra el panel, inmediatamente se notificará esta activación en este. La indicación que mostrará será *Wait,* informando al usuario que la aplicación ha recibido la notificación del control, pero este no se hará efectivo hasta que no comience el día siguiente. Esta situación se representa en la Figura 74.

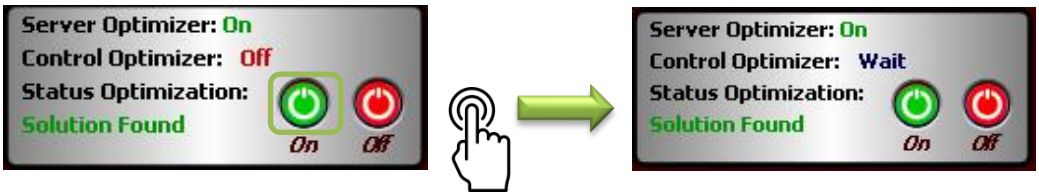


Figura 74. Activación del control del optimizador

Cuando llega el instante de tiempo 86400 segundos, momento en el cuál la simulación finaliza un día y comienza a simular otro, se activará inmediatamente el control del optimizador. A partir del segundo 0 del día siguiente el control lo tomará el optimizador, comenzando a introducir las consignas propuestas en el cálculo. Este control se puede verificar en el panel de control y estado del servidor que mostrará la imagen que se ve en la Figura 75.

Figura 75. Verificación del control del optimizador

En todo momento el usuario podrá volver a retomar el control del proceso, pulsando en el botón *Off* del panel. La pulsación de este botón cortará la comunicación con el servidor de la optimización y comenzará a controlar las consignas del proceso el usuario siguiendo la simulación en el instante en que se haya realizado la pulsación.

Las gráficas que verifican el control del optimizador se muestran a continuación. Se debe destacar que aunque las gráficas implementadas sean de tiempo real, la evolución temporal de las variables no es en tiempo real, ya que el tiempo de integración (factor de aceleración) en la simulación es de 50 segundos. Por lo tanto un segundo de tiempo real se corresponde con 50 segundos en la simulación del proceso.

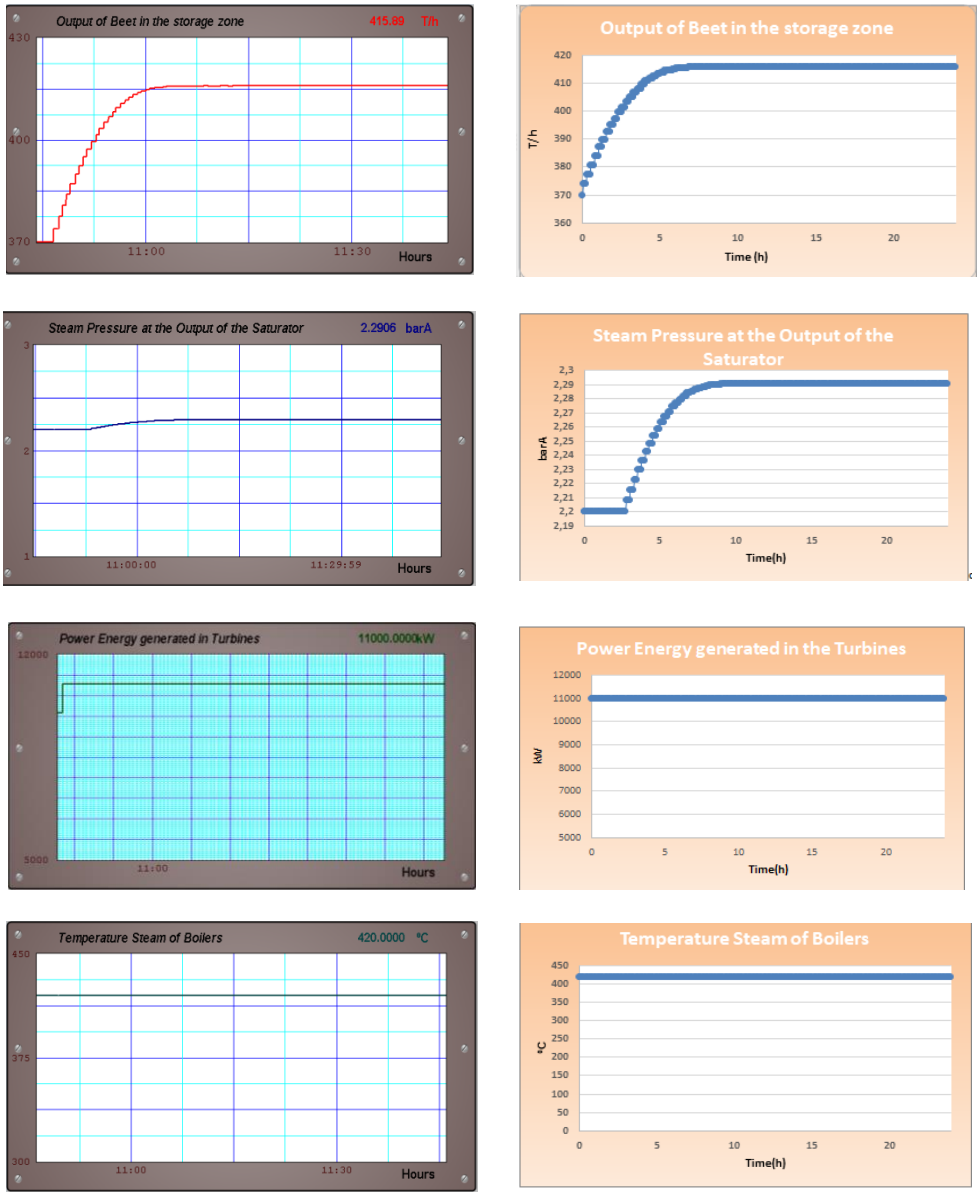


Figura 76. Relación gráficas tiempo real - gráficas Excel

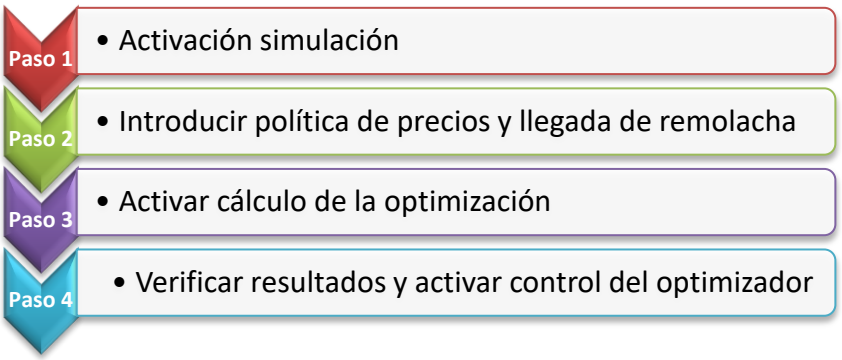
En esta Figura 76 se muestra la comparativa entre los resultados que proporciona el optimizador en Excel, y la evolución del sistema cuando el optimizador toma el control. Como cabía esperar, los resultados son los mismos, salvando el detalle de las escalas, se puede verificar el control del sistema mediante el optimizador.

4.2. VERIFICACIÓN DEL FUNCIONAMIENTO DE LA APLICACIÓN

Para verificar el funcionamiento correcto de la aplicación se ha desarrollado una serie de pruebas. Estas pruebas abordan todas las situaciones en las que el usuario puede hacer uso de la simulación y la optimización. Para ello se deben superar cada una de ellas hasta que la simulación llegue hasta el último día propuesto. El factor de aceleración para realizar las pruebas es de 50 segundos, por lo tanto ajustando las gráficas de las variables se conseguirá visualizar un día de simulación en una hora de tiempo real.

Como primera prueba para el día 1 de simulación se proponen los siguientes pasos a realizar y verificar su cumplimiento:

Prueba: Día 1



La activación de la simulación se ha realizado estableciendo los siguientes parámetros, para su correcto funcionamiento:

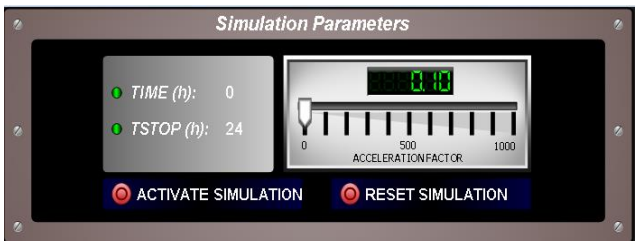


Figura 77. Prueba día 1: Activación de la simulación

Con la ejecución de la simulación activa, se puede proceder a introducir una política de precios y llegada de remolacha que se corresponde con la imagen que se muestra en la Figura 78:

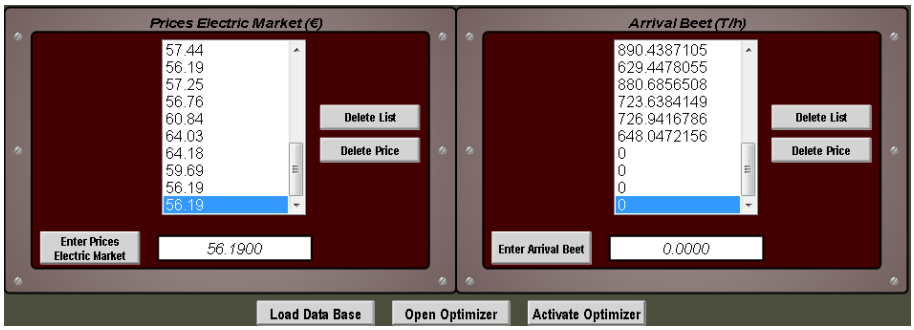


Figura 78. Ventana para introducir la política de precios y llegada de remolacha

Estableciendo los 25 valores correspondientes, se activa el servidor de la optimización, para proceder a activar el cálculo. Una vez activado el cálculo y que el optimizador haya encontrado la solución a los valores introducidos, se podrá establecer el control de este. Antes de aplicar el control, el usuario podrá verificar los resultados del optimizador en las gráficas Excel, que muestran a continuación.

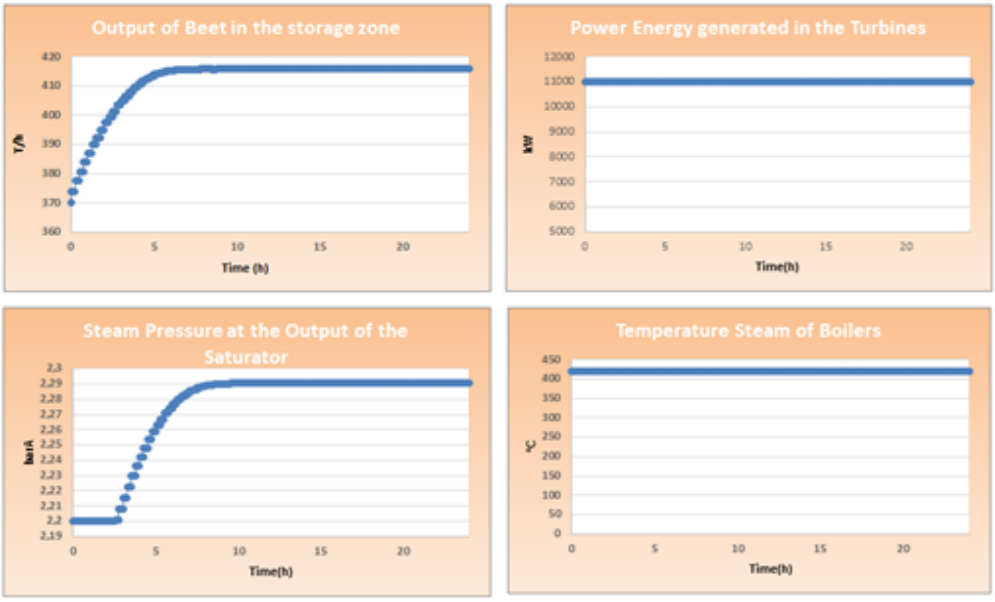


Figura 79. Prueba día 1: Resultados del cálculo del optimizador para el día 2 de simulación

Una vez verificados estos resultados, se ha decido que son apropiados para realizar el control, por lo tanto se activará el botón de control del optimizador, que enviará la petición de control al sistema. Para verificar que la petición de control del optimizador ha llegado a este se tiene la Figura 80 que muestra esta situación a través del panel de estado y control del servidor:

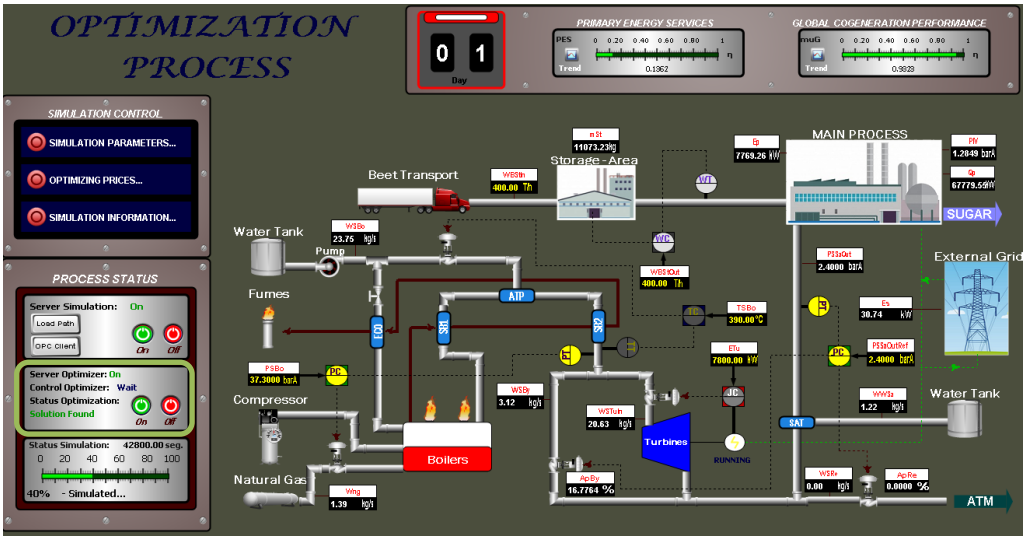


Figura 80. Prueba día 1: Verificación de la petición de control del optimizador

Prueba: Día 2

- **Paso 1** • Verificar control del optimizador
- **Paso 2** • Lanzar el cálculo de otra optimización
- **Paso 3** • Verificar resultados y aplicar control

Para verificar el control del optimizador, se debe esperar hasta el día 2 de simulación del proceso. En ese momento el panel de estado y control notificará el control de este, comenzando a manipular las consignas del proceso como se puede ver en la Figura 81.

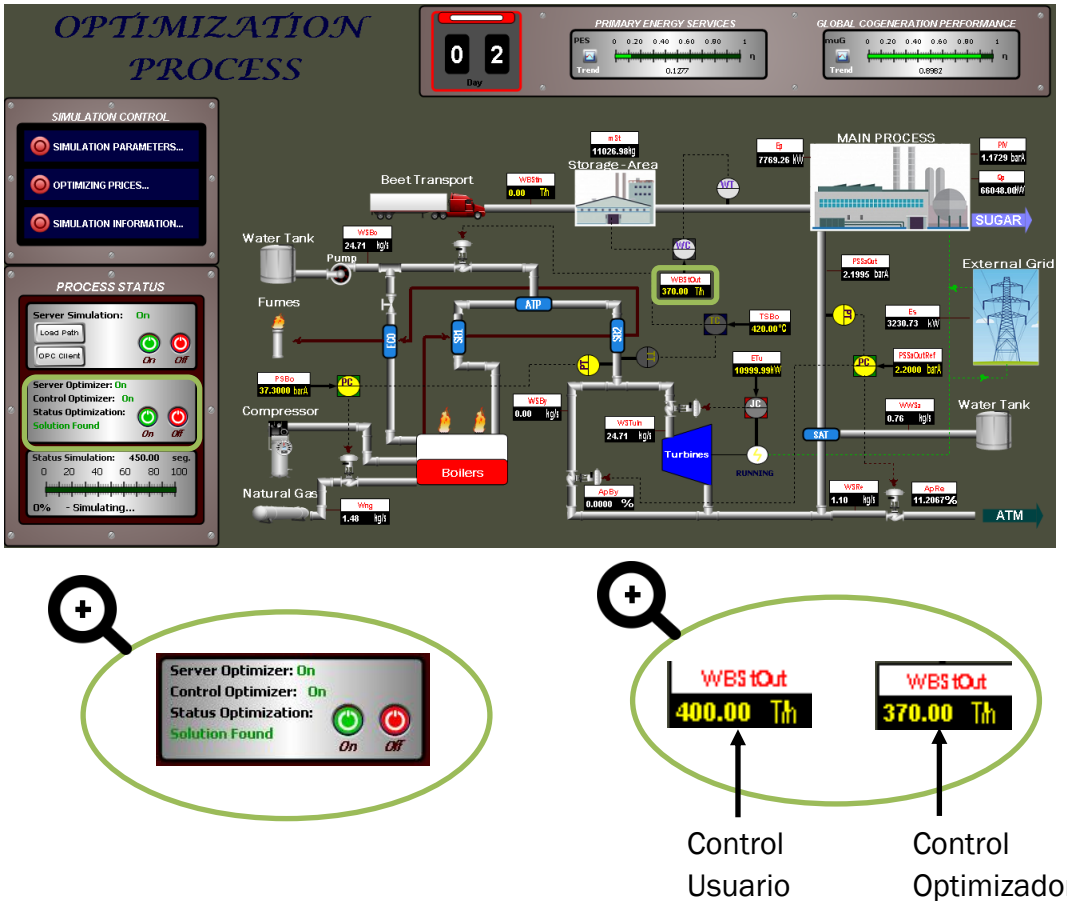


Figura 81. Prueba día 2: Activación del control del servidor de la optimización

En esta situación se puede observar como el sistema evoluciona de manera autónoma estableciendo las consignas calculadas y mostradas en las gráficas Excel del día 1. Durante el control del optimizador, se procede a lanzar otra optimización para que se haga efectiva el día 3 de simulación. Para ello se deberá realizar el mismo procedimiento que en el día 1, en el cual se introducían los precios y la llegada de remolacha. Una vez el optimizador realice los cálculos, se debe proceder a verificar el resultado de estos. Los resultados para el día 3 de simulación son los que se muestran en la Figura 82.

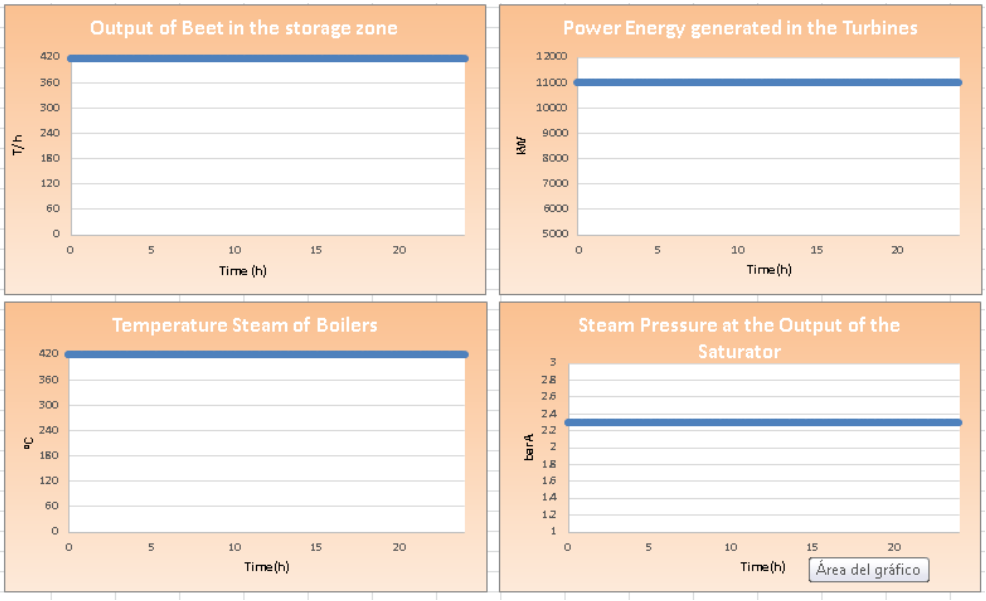


Figura 82. Prueba día 2: Resultados del cálculo del optimizador para el día 3 de simulación

Cuando finalice el tiempo de ejecución de la simulación del día 2, el optimizador deberá continuar con los resultados calculados para aplicar en el día 3 de simulación. Para verificar que el control del optimizador en el HMI se corresponde con los valores mostrados en las gráficas de resultados de la Figura 79, se ha hecho una captura de las gráficas en tiempo real que muestra el HMI. Como el factor de aceleración es de 50 segundos, se han ajustado las escalas en las gráficas para que se pueda mostrar los resultados de la simulación de un día entero.

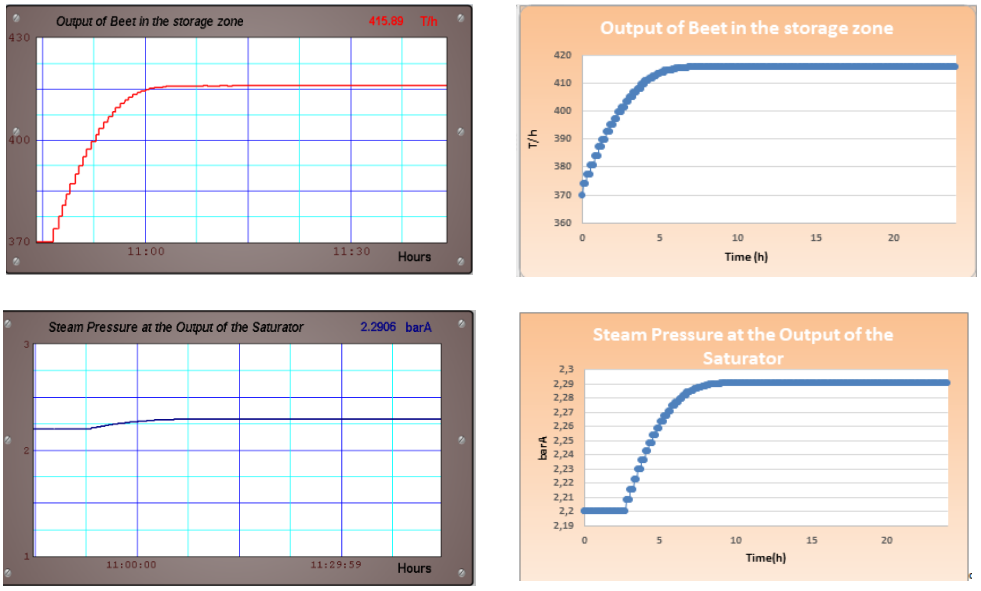


Figura 83. Prueba día 2: Comparación de resultados (I)

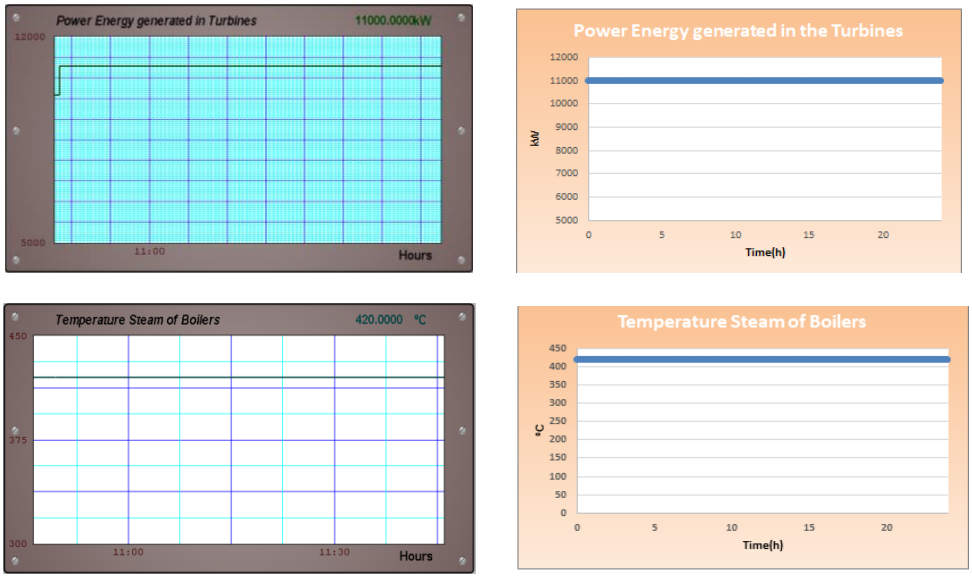
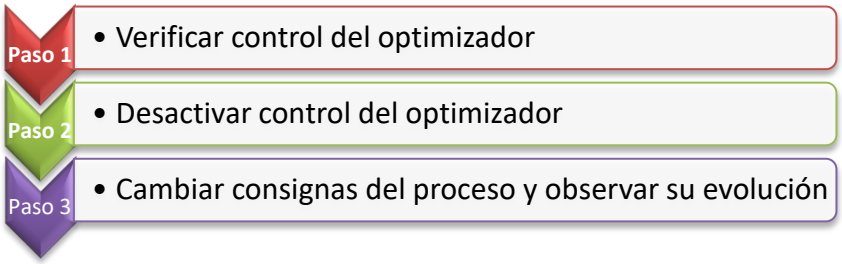


Figura 84. Prueba día 2: Comparación de resultados (II)

Como se puede apreciar en las imágenes, el control del optimizador evoluciona correctamente.

Prueba: Día 3



Durante el día 3 de la simulación, se debe verificar que el optimizador ha seguido con el control del sistema, para ello se ha hecho una captura que represente esta situación en la Figura 85:

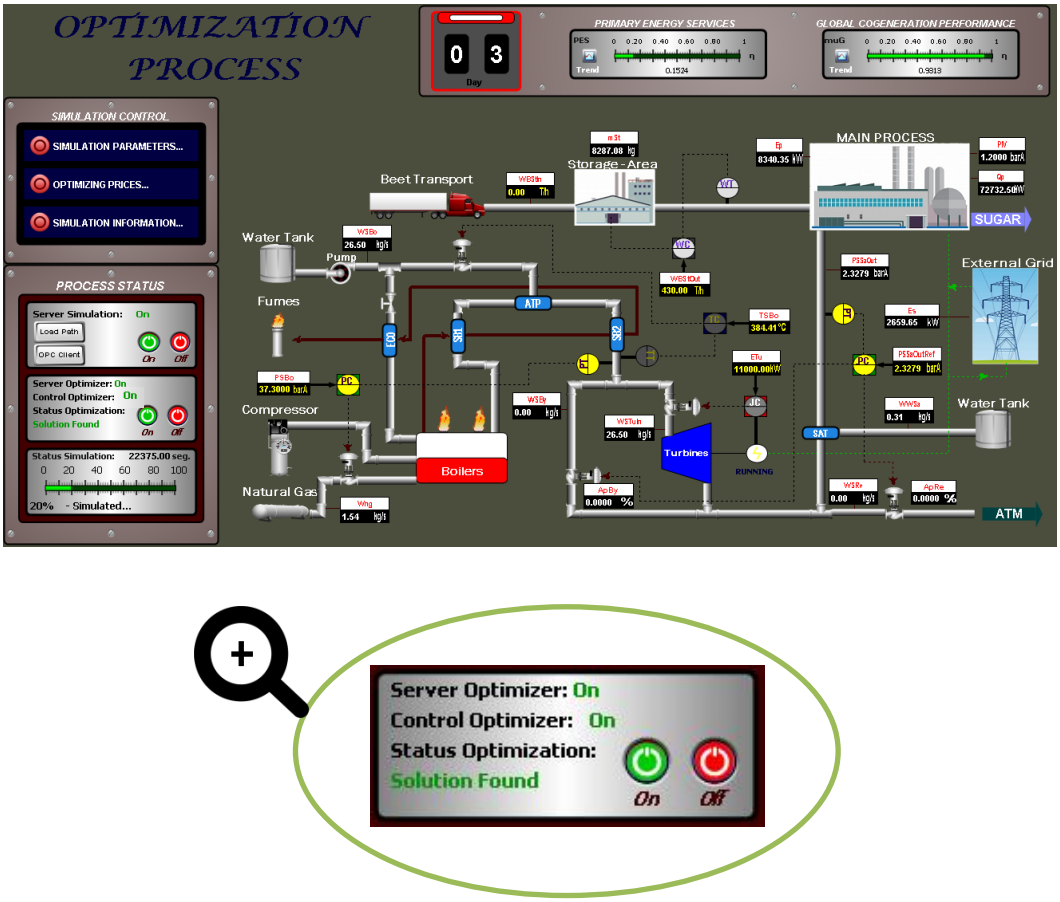


Figura 85. Prueba día 3: Verificación de la petición de control del optimizador

Durante el control del optimizador en este día, por una situación de emergencia en la planta, las consignas que está aportando el optimizador al proceso no son correctas, por lo tanto el operario deberá parar el control del optimizador y llevar el sistema a una situación segura. Esta situación se puede verificar cuando se capturan los resultados de la simulación completa del día 3, en las gráficas que muestra la aplicación gráfica, se observa un cambio brusco en el valor de las variables que indica la manipulación de estas por el operario, cancelando el control del optimizador.

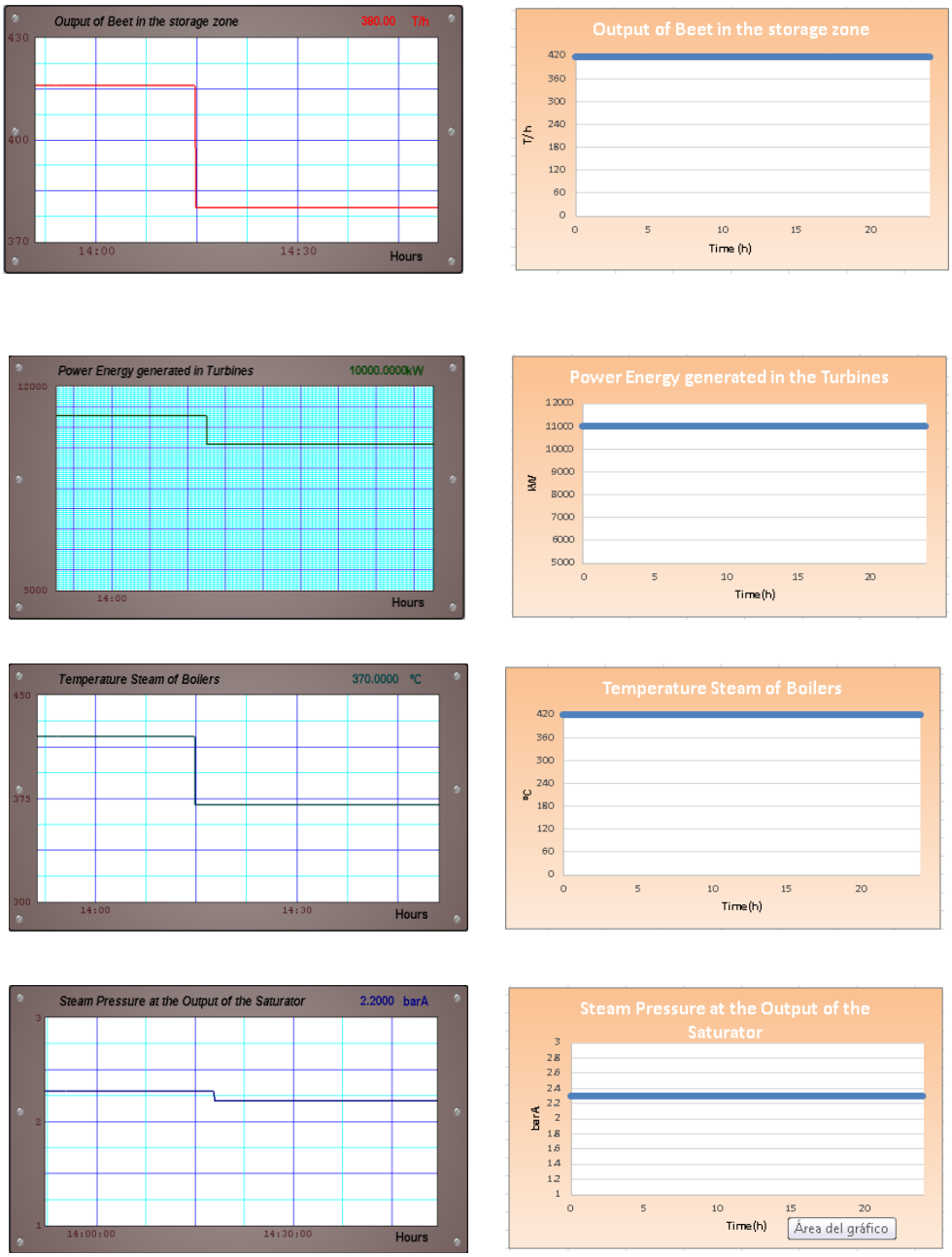


Figura 86. Prueba día 3: Comparación de resultados

Como se puede ver en los resultados, la descripción de la situación se ve reflejada en la evolución temporal del sistema. Con estas pruebas se puede garantizar que el operario tiene control total del sistema.

CONCLUSIONES

El presente TFG buscaba presentar el diseño de una interfaz gráfica que facilite la comunicación entre la herramienta encarga de optimizar el proceso de cogeneración asociado a una industria azucarera y la planta. Se puede observar en los resultados obtenidos y las gráficas presentadas, como la comunicación ha sido efectiva gracias a la aplicación gráfica diseñada.

Se ha podido constatar que utilizando OPC UA como principal protocolo de comunicaciones en la aplicación diseñada, esta es más fiable, puesto que se puede garantizar que el mensaje ha sido recibido por ambas entidades (cliente/servidor). Esto se debe a que si en algún momento el servidor deja de emitir datos o estos están dañados, el cliente obtendrá dicha información. Esta notificación ha sido de gran utilidad a la hora de detectar errores en la comunicación con los servidores en el proceso de ejecución de la aplicación.

A lo largo de este trabajo se ha señalado la importancia del programa Wonderware en el diseño de la aplicación gráfica. Esto viene justificado porque se ha podido corroborar su potencialidad en entornos industriales, puesto que su amplio abanico de funcionalidades ha facilitado la resolución de los requisitos propuestos, sin necesidad de utilizar otro tipo de programas externos, gracias a su integración en las comunicaciones y el diseño gráfico. Se considera necesario señalar, que este software tiene una gran complejidad, por lo que se ha visto la necesidad de ampliar los conocimientos para su utilización, a través de sus manuales, así como la posibilidad de continuar ampliado los conocimientos en un futuro.

Pyhton, se trata de una lenguaje de programación de fácil acceso y gratuito para todas las personas que lo deseen. Esto hace que sea muy utilizado y sus niveles de documentación sean muy amplios, por lo que la programación utilizando librerías ha facilitado su implementación en la creación de la capa de comunicaciones para el servidor OPC UA utilizado.

En último lugar y no menos importante, cabe señalar la gran utilidad de la aplicación gráfica diseñada. Esto viene justificado por su diseño pensado para la comprensión, utilizando la intuición lo cual facilita su lectura. Así mismo, los beneficios que aporta al sector industrial se pueden observar en diferentes ámbitos. En primer lugar a nivel económico, puesto que la venta de los excedentes de energía eléctrica aumenta los ingresos de la empresa, promoviendo un desarrollo económico sostenible en el tiempo. Del mismo modo, tiene beneficios medioambientales puesto que los niveles de contaminación disminuyen notablemente, ya que la aplicación tiene en cuenta los índices de rendimiento acogiéndose a la legislación europea que recompensa los procesos de eficiencia de cogeneración. Por lo que cabe concluir, que está tipología de aplicaciones gráfica son de gran utilidad e

importancia para el desarrollo de sociedades con modelos económicos globalizados, puesto que su base es la sostenibilidad.

BIBLIOGRAFÍA

[1] "Home Page - OPC Foundation." [En línea]. Disponible: https://opcfoundation.org/. [Accedido: 7-7-2019].

[2] J. M. Zamarreño Cosme, *Acceso a datos mediante OPC*. Andavira, 2010.

[3] D. Kominek, P. Eng Alberta, y C. -, "OPC: ¿De qué se trata, y cómo funciona? 'Guía para entender la Tecnología OPC.'"

[4] J. D. Lemos, D. M. Guerrero, y A. Arias, "OPC Como Alternativa a las Tecnologías Propietarias de Comunicación Industrial," 2006.

[5] S.-H. Leitner y W. Mahnke, "OPC UA – Service-oriented Architecture for Industrial Applications," *ABB Corp. Res. Cent.*, vol. 26, no. 4, pp. 1–6, 2006.

[6] W. Mahnke, S.-H. Leitner, y M. Damm, *OPC unified architecture*. Springer-Verlag, 2009.

[7] A. Rodríguez Penín, "Sistemas SCADA ❒ Sistemas SCADA ❒," Barcelona: Marcombo, 2003.

[8] "Wonderware Spain." [En línea]. Disponible: https://www.wonderware.es/. [Accedido: 7-7-2019].

[9] "dataFEED OPC Suite - Softing | Softing." [En línea]. Disponible: https://data-intelligence.softing.com/products/opc-software-platform/datafeed-opc-suite/. [Accedido: 7-7-2019].

[10] "UaExpert "UA Reference Client" - Unified Automation." [En línea]. Disponible: https://www.unified-automation.com/products/development-tools/uaexpert.html. [Accedido: 7-7-2019].

[11] "OPC UA Components - MATLAB & Simulink - MathWorks España." [En línea]. Disponible: https://es.mathworks.com/help/opc/ug/opc-ua-components.html. [Accedido: 7-7-2019].

[12] "Object Viewer User's Guide," Cambridge: AVEA, 2018.

[13] "Nuestros mercados de electricidad | OMIE." [En línea]. Disponible: http://www.omie.es/inicio/mercados-y-productos/mercado-electricidad/nuestros-mercados-de-electricidad. [Accedido: 7-7-2019].

[14] J. R. Morales, "Cogeneración y transición energética en España : más industria , más mercados y más gas," pp. 76–86.

[15] M. I. Sosa and A. Fushimi, "EL ROL DE LA REGULACIÓN EN EL DESARROLLO DE LA COGENERACIÓN," *Av. en Energías Renov. y Medio Ambient.*, vol. 8, 2004.

[16] C. Pablos, A. Merino, and L. F. Acebes, "Modeling On-Site Combined Heat and Power Systems Coupled to Main Process Operation," *Processes*, vol. 7, no. 4, p. 218, Apr. 2019.

[17] "EcosimPro | PROOSIS Modelling and Simulation Software." [En línea]. Disponible: https://www.ecosimpro.com/. [Accedido: 7-7-2019].

[18] "Pyomo." [En línea]. Disponible: http://www.pyomo.org/. [Accedido: 7-7-2019].

[19] H. Perez, "High Performance Graphics to Maximize Operator Effectiveness," 2012.

[20] O. P. Rivera, P. Asociado, P. Oscar, and P. Rivera, "Pagina 1 Norma ISA."

ANEXO I: COMUNICACIÓN CON EL SERVIDOR OPC UA

1. Comunicación en Wonderware cliente/servidor OPC UA

1.1. OIGATEWAY 3.0 (OPERATIONS INTEGRATION GATEWAY)

El paquete *OIGateway* es un software asociado a Wonderware, encargado de gestionar las comunicaciones con dispositivos externos. Es necesario destacar, que el entorno de Wonderware instalado para la realización del TFG, no contaba con este programa, sino con la versión anterior *FSGateway* que no era capaz de soportar el protocolo de comunicación OPC UA, de ahí que fuera necesaria su actualización.

Este programa cuenta con un Gateway que actúa como convertidor de protocolos para poder establecer la comunicación con diferentes dispositivos. Con este tipo de Gateway podemos establecer la comunicación con el servidor OPC UA.

La arquitectura de comunicaciones que presenta el Gateway se muestra a continuación, mostrando las diferentes posibilidades para la comunicación con dispositivos externos.

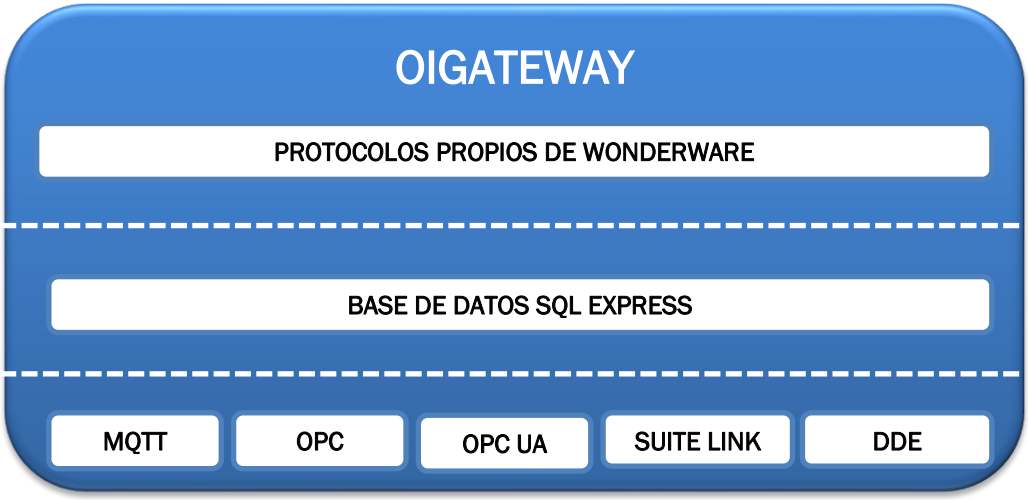


Figura 87. Arquitectura del Gateway *OIGateway 3.0*

1.2. CREACIÓN DEL CLIENTE OPC UA

Para la creación del cliente OPC UA se debe desplegar el árbol que muestra el programa *System Management Console* al iniciarse y localizar el Gateway, en este caso *OI Gateway*:

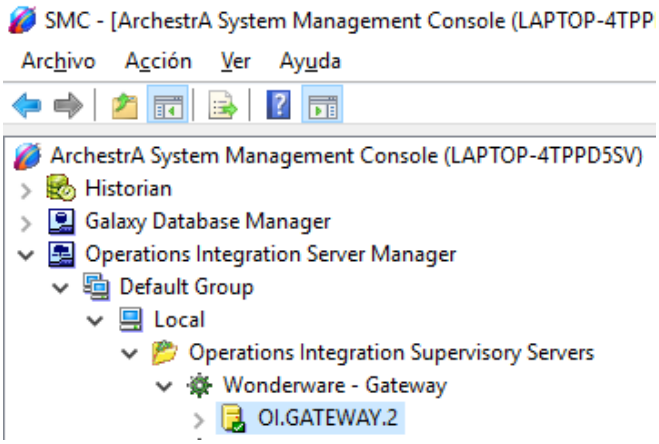


Figura 88. Árbol de Operaciones de System Management Console

Desplegando la pestaña correspondiente al Gateway, se mostrará una ventana de configuración del nodo que se utilizará para la comunicación entre cliente y servidor. Los parámetros por defecto que muestra Wonderware son los recomendados para establecer una comunicación con cualquier dispositivo. Sin modificar ninguno de estos parámetros se creará el cliente asociado a la comunicación OPC UA.

Para la creación del cliente pinchando con el botón derecho del ratón sobre *Configurations* se desplegará una pestaña con los diferentes tipos de comunicación permitidas, seleccionando *Add OPC UA Connection* se creará el cliente, el nombre de éste se puede modificar quedando en este caso como muestra la Figura 89.

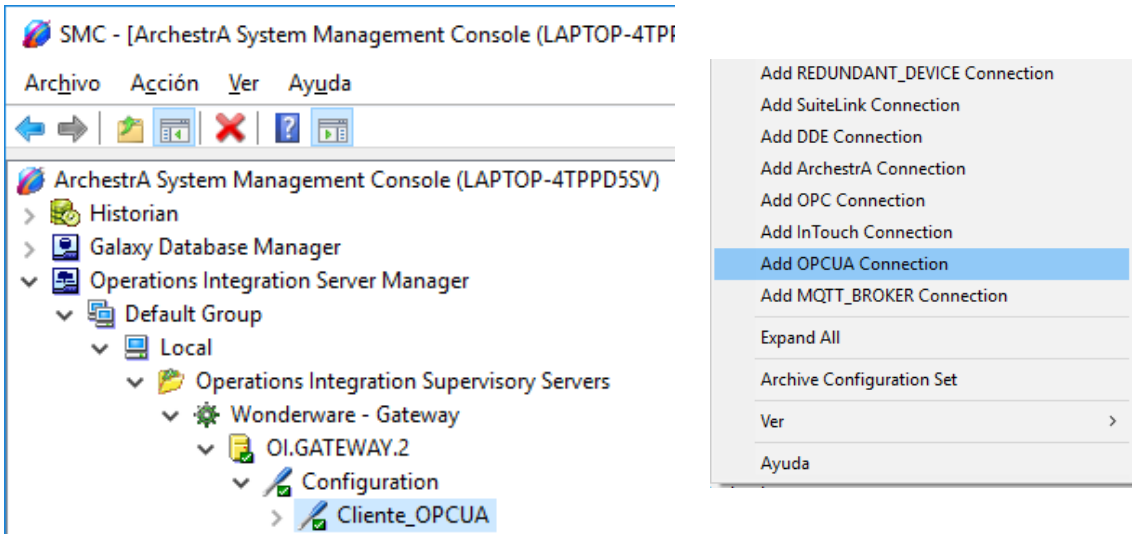


Figura 89. Creación del Cliente OPC UA

Una vez creado el cliente OPC UA se debe establecer una serie de parámetros para garantizar la correcta conexión y comunicación con el servidor OPC UA. La ventana de configuración muestra los siguientes parámetros:

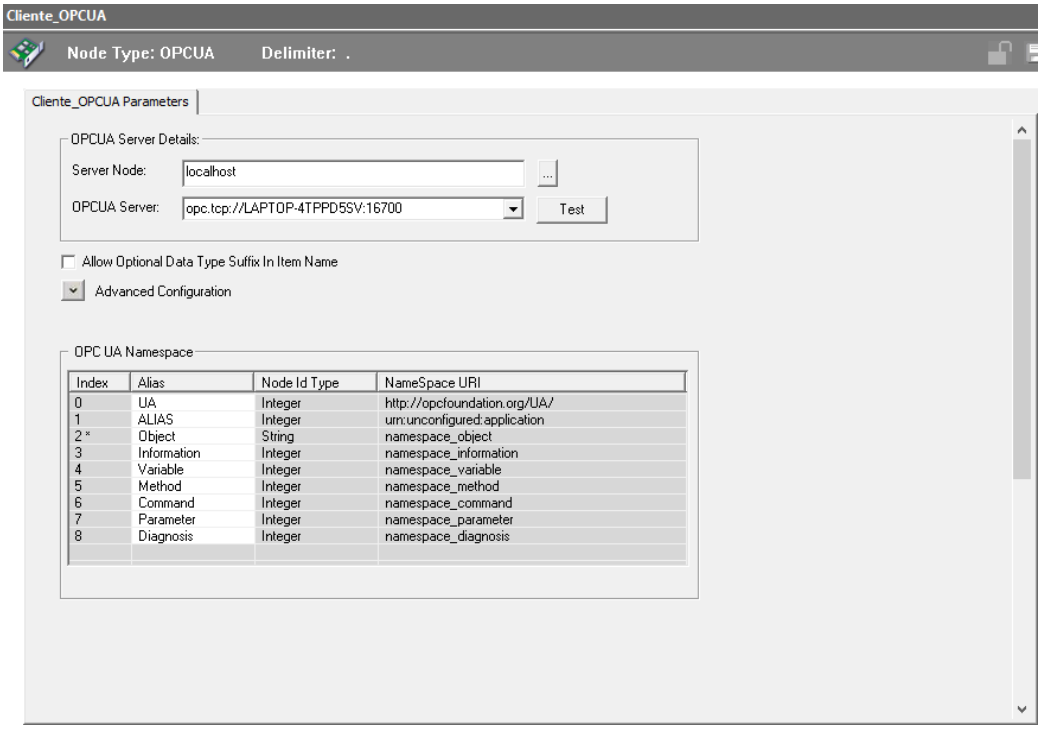


Figura 90. Parámetros para la conexión con el Servidor

Para la conexión con el servidor OPC UA, se debe indicar el nodo donde se encuentra el servidor (en este caso localhost, se encuentra en la misma máquina donde está instalado Wonderware), y la dirección URL para establecer la comunicación. Para comprobar que la comunicación es correcta se puede pulsar *Test*. A continuación, se detallan los parámetros de configuración:

- ***Server Node***: Nombre para establecer el nodo de conexión del servidor OPC UA, en este caso como el servidor se encuentra en el mismo ordenador, se puede dejar en vacío o introducir "localhost", ambos indicarán que el nodo está en el ordenador local.
- ***OPCUA Server***: Se establece la dirección y el puerto donde está conectado el servidor OPC UA, para realizar la conexión y comunicación. Sigue el formato:

opc.tcp//DireccióndelNodo:Puerto

- *OPC UA Namespaces:* Se establece un nombre para los campos a los que se puede acceder en el servidor, este nombre se reflejará cuando se añadan las variables asociadas al servidor.

Las opciones avanzadas que se muestran en la Figura 91, corresponden principalmente a la seguridad del servidor:

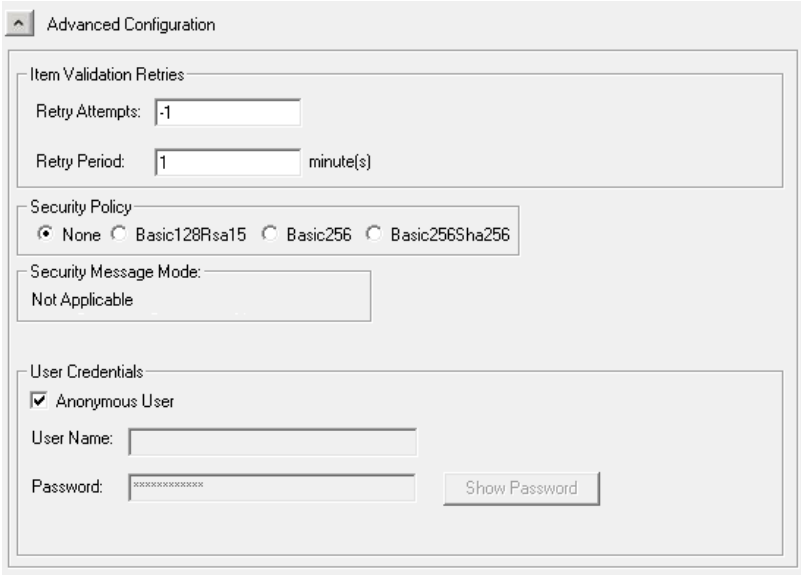


Figura 91. Configuración Avanzada para conexión con el Servidor OPC UA

- *Item Validation Retries*: Configuración de los intentos de reconexión del servidor y cada qué período se realiza esta reconexión.
- *Security Policy*: Se establece una seguridad de encriptación en los mensajes cuando se produce la comunicación con dicho servidor.
- *User Credentials*: Establece un acceso limitado a toda persona ajena al servidor.

En este caso, no se ha definido ningún tipo de seguridad ni modificado ningún parámetro por defecto. Una vez se ha configurado correctamente, se guardan los parámetros y pinchando con el segundo botón sobre el cliente se añade un nuevo grupo de ítems (variables) del servidor:

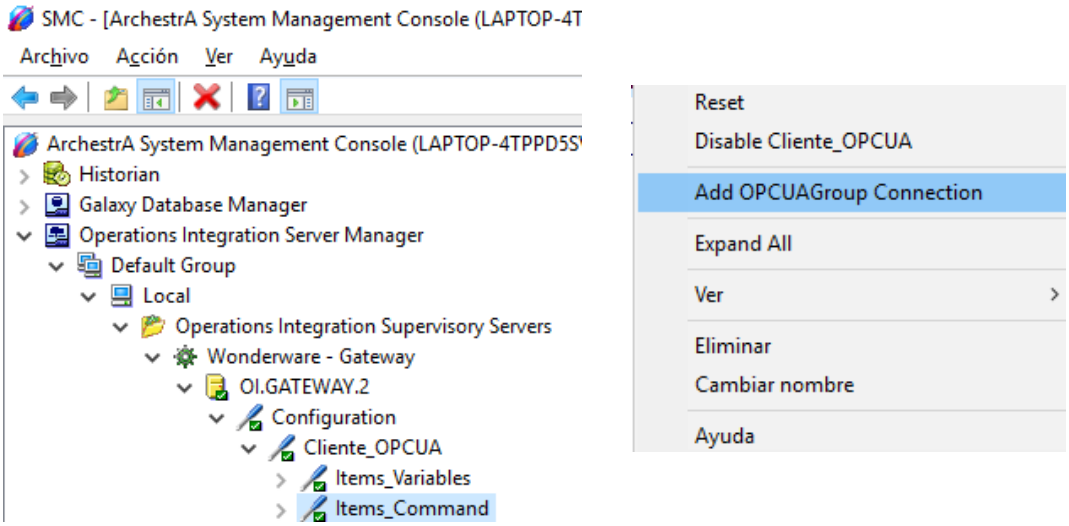


Figura 92. Creación de los ítems del Servidor OPC UA

En este caso se han creado dos grupos principales, definidos como "Variables" y "*Command*", que hacen alusión a las variables del proceso que contiene el servidor, y los métodos para poder arrancar la simulación del servidor. Una vez creados y definidos los nombres se accede a la siguiente ventana de configuración:

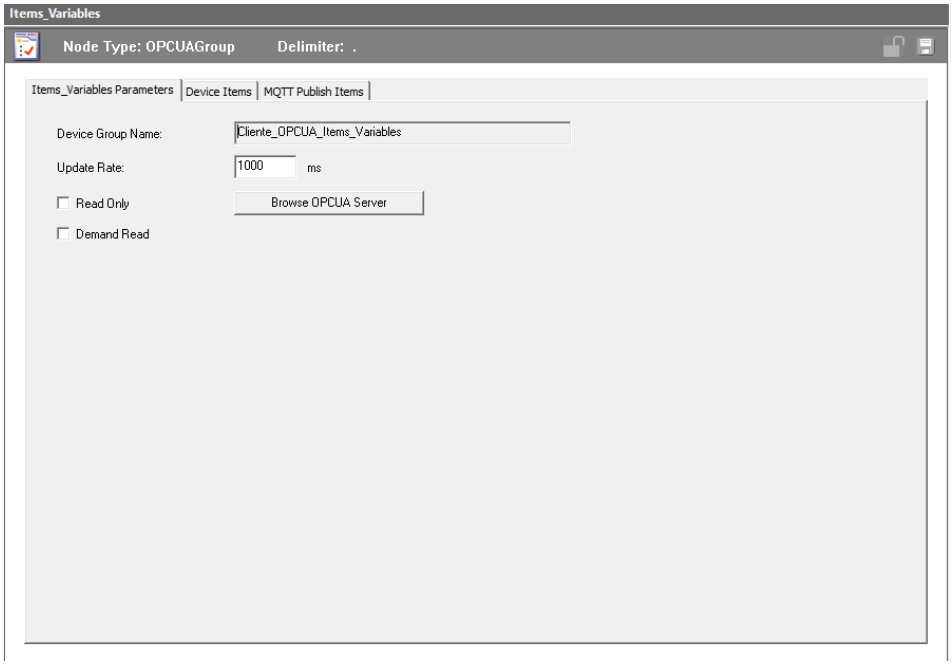


Figura 93. Configuración de los ítems del Servidor OPC UA

En esta configuración se puede definir el tiempo de actualización de las variables y las variables que se desea seleccionar. Accediendo a la pestaña *Browse OPCUA Server* se va al contenido del servidor donde se seleccionarán

las variables y métodos necesarios, una vez añadidos se mostrarán en la pestaña *Device Items:*

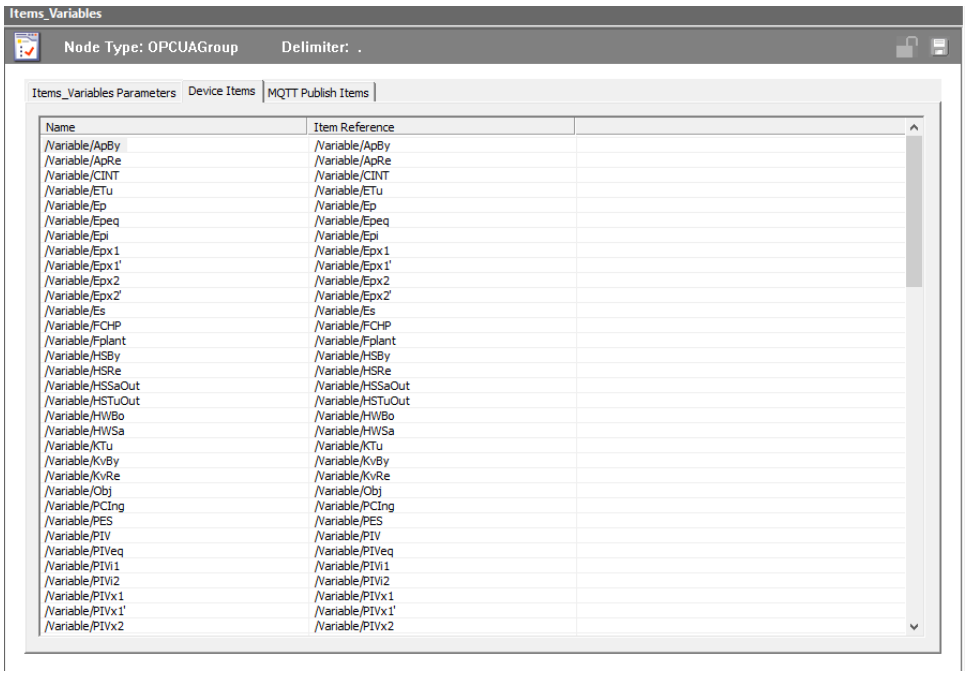


Figura 94. Ítems del Servidor añadidos

El nombre de los ítems añadidos se puede modificar, una vez configurada esta ventana se guardará el grupo añadido y se activará el servidor *OIGateway*, para poder establecer la comunicación con el servidor en cualquier momento (el servidor al que se conectará deberá estar activo). La activación del servidor de *OIGateway* se realiza accediendo a la configuración de este en el árbol que muestra la Figura 88. Para acceder a la configuración pinchando con el botón derecho del ratón sobre el servidor, se desplegaran las opciones de configuración, entre las cuales estará la de activar el servidor.

2. Creación de un elemento con variables del proceso

2.1. CREANDO UNA APLICATTION SERVER ARCHESTRA

Application Server Archestra, es la manera de crear una aplicación administrada utilizando el IDE que proporciona *ArchestrA*. Esta administración se debe a la posibilidad que ofrece *ArchestrA* de interactuar con los subprogramas disponibles en el entorno Wonderware. Creando este tipo de aplicación se podrá acceder a cada uno de los subprogramas que se utilizan en Wonderware para establecer comunicaciones con dispositivos externos, realizar diseños de interfaces gráficas, comunicaciones con paquetes MS Office...De esta manera se tendrá la administración de todo el entorno Wonderware desde una sola aplicación.

Para crear una aplicación de server *ArchestrA*, en primer lugar se debe conectar a Galaxy (Galaxia que administra las conexiones de Wonderware con otras Application Server), para ello, se debe indicar el nodo (elemento de conexión) donde se realizará la creación de la conexión, una vez creado si todo ha ido correctamente, se pulsará el botón *Connect* y se accederá a *ArchestrA IDE*:

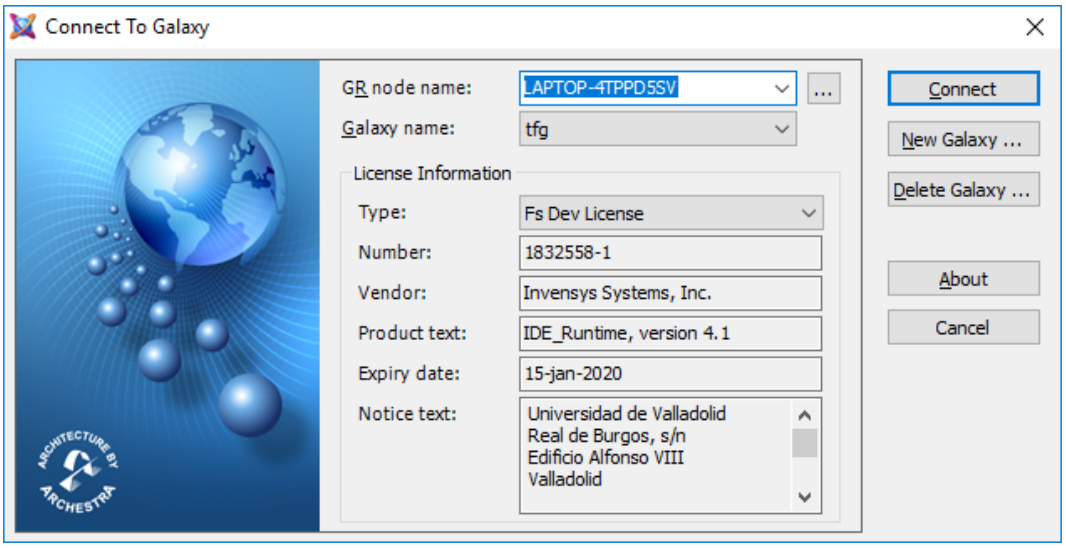


Figura 95. Conexión con Galaxy - ArchestrA IDE

En la ventana que abre ArchestrA IDE, se tiene dos herramientas de edición:

- ***Template Toolbox:*** Herramienta para la conexión con los software del programa y con dispositivos externos, se proporcionan una serie de plantillas preparadas para realizar la conexión y comunicación.

- ***Graphics Toolbox:*** Herramienta para la reutilización y diseño de elementos gráficos, proporciona una serie de elementos gráficos prediseñados que pueden ser editados e incluso definir un objeto por el usuario.

Esta aplicación permitirá conectar con el servidor OPC UA y administrar las variables que se hayan almacenado mediante el cliente OPC UA, una vez realizada la conexión se creará un elemento gráfico asociado a las variables que involucren a dicho gráfico en el proceso.

2.2. CONEXIÓN CON EL SERVIDOR OPC UA (TEMPLATE TOOLBOX)

Para realizar la conexión con el servidor OPC UA, se debe realizar subplantillas de los elementos que se muestran en la Figura 96:

- ***AppEngine*:** El objeto AppEngine necesita una plataforma en la que poder arrancar.
  - Almacena los objetos *ApplicationObjects, Devices Integration* y *Areas*.
  - Contiene la lógica para establecer la inicialización de los objetos cuando se han implementado en el sistema.
  - Contiene la lógica para borrar los objetos de *AppEngine* cuando no se han implementado en el sistema.
  - Determina el tiempo de búsqueda de todos los objetos que pertenecen a *AppEngine*.
- ***Area:*** Todos los objetos que se creen del sistema deben pertenecer a un área. Este objeto proporciona una organización en la agrupación de alarmas, para obtener información que después podrá ser utilizada por clientes que utilicen alarmas/eventos para supervisar sus áreas. Esta información se puede guardar como históricos del sistema.
  - Activar el contador de alarmas.
  - Contador de alarmas no reconocidas.
  - Deshabilitar (o silenciar) el contador de alarmas.
- ***InTouchViewApp:*** Este objeto debe tener un *ViewEngine* en el que ejecutarse.
  - Gestiona la sincronización de archivos requeridos por la aplicación *InTouch* asociada.
  - Proporciona acceso en tiempo de ejecución a los tags definidos en la aplicación *InTouch* asociada.

- ***ViewEngine:*** Este objeto necesita una plataforma en la cual ejecutarse.
  - Almacén los objetos relativos a *InTouchViewApp*.
  - Contiene la lógica para establecer la inicialización de los objetos cuando se han implementado en el sistema.
  - Contiene la lógica para borrar los objetos de *AppEngine* cuando no se han implementado en el sistema.
  - Determina el tiempo de búsqueda de todos los objetos que pertenecen a *ViewEngine*.

- ***WinPlatform:*** Es un objeto que se utiliza para almacenar otros objetos que permitan modelar el sistema. Este permite:
  - Calcular varias estadísticas relativas al nodo que se implemente en el sistema. Estas estadísticas pueden ser publicadas como atributos.
  - Monitorizar varias estadísticas relativas al nodo que se implemente en el sistema. Estas estadísticas pueden ser supervisadas como alarmas e históricos.

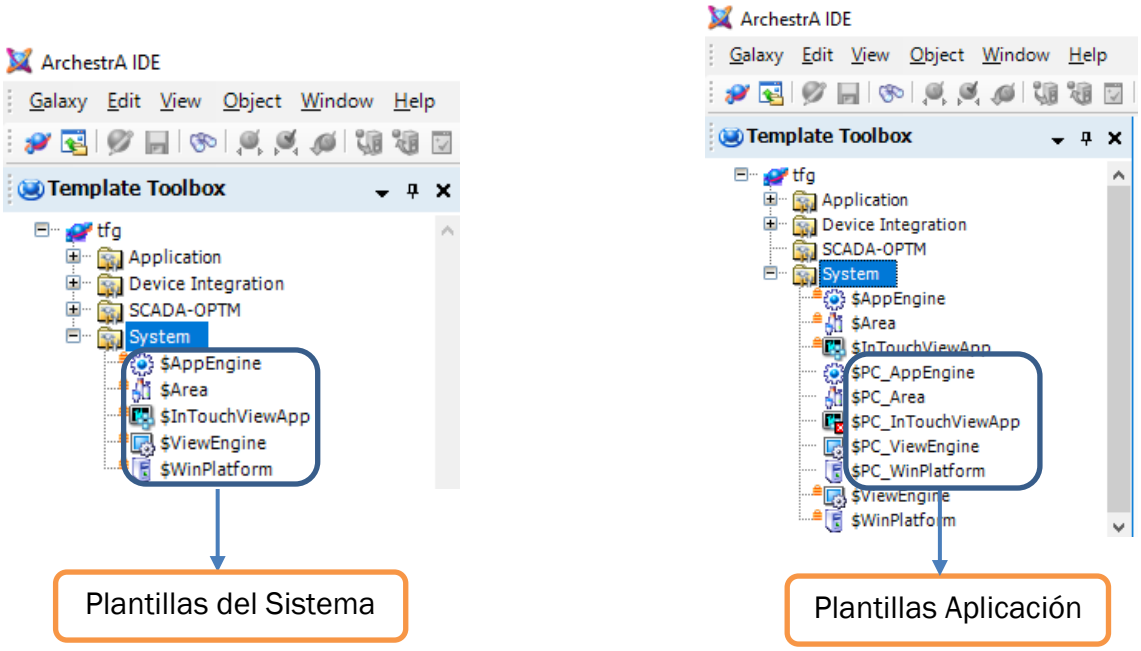


Figura 96. Creación de subplantillas - Template Toolbox

La creación de dichas subplantillas se realiza pinchando con el botón derecho del ratón sobre la plantilla del Sistema:

New -> Derived Template

Una vez creadas todas las subplantillas se renombrarán para identificarlas correctamente y se creará una subcarpeta que almacenará todas las subplantillas creadas.

Pinchando con el botón derecho sobre el nodo raíz del árbol, se desplegará la opción ***"New Template Toolset"***, esta carpeta será la que contendrá todas las subplantillas destinadas a la comunicación con el servidor.

Una vez arrastradas las subplantillas a dicha carpeta se realizará la configuración de la comunicación:

- ***"PC_InTouchViewApp"***
  Con la configuración de esta subplantilla creamos la conexión con la aplicación *InTouch*, de esta forma quedará referenciada al Galaxy creado.
  Abriendo la subplantilla creada se mostrará la configuración:

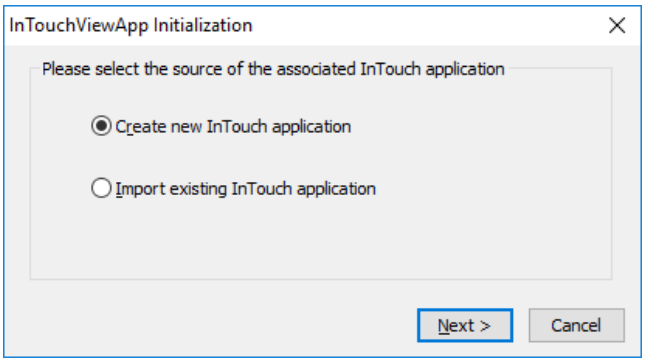


Figura 97. Inicialización Subplantilla - InTouchViewApp

Para este caso se creará una nueva aplicación en *InTouch*. A continuación, se debe elegir el nombre de la aplicación que se va a crear:

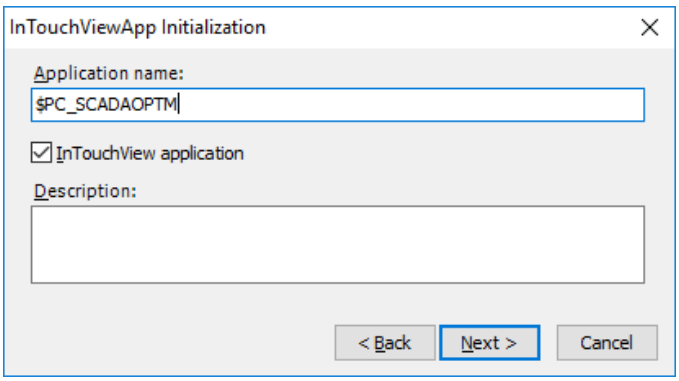


Figura 98. Nombre de la aplicación InTouch - InTouchViewApp

- *"PC_WinPlatform"*
  Esta subplantilla se corresponde con los parámetros del nodo que realizará la conexión con el servidor. Abriendo esta subplantilla se tendrá la siguiente ventana de configuración:

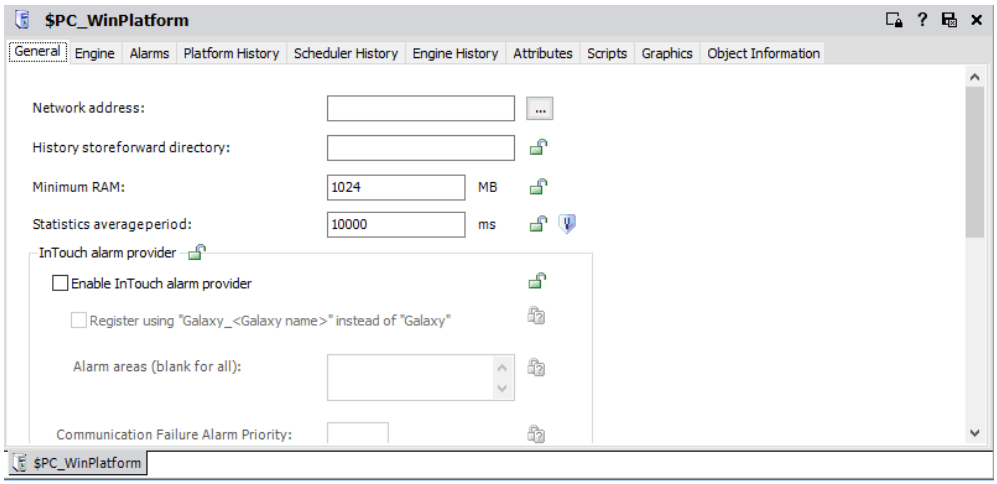


Figura 99. Configuración del Nodo de Conexión - WinPlatform

Se debe introducir el nombre del nodo donde se realizará la conexión, en este caso como está en el mismo ordenador se deberá poner ***"localhost"***. Una vez realizada la configuración se guardará y cerrará la ventana.

Con estos parámetros se tiene la configuración básica para realizar la conexión y el diseño del elemento gráfico. A continuación, los elementos creados se deben instanciar, para que después puedan cargarse en ArchestrA y referenciar todos los software con la aplicación creada.

La instancia se realiza pinchando con el segundo botón en cada una de las subplantillas creadas y accediendo:

New -> Instance

Al realizar en cada subplantilla una instancia, se añaden al árbol inferior izquierdo donde se muestran las instancias en una carpeta llamada ***"Unassigned Area"***, para que todo funcione correctamente se deben agrupar como se muestra en la Figura 100, teniendo en cuenta la pestaña ***"Model"*** y la pestaña ***"Deployment"***:

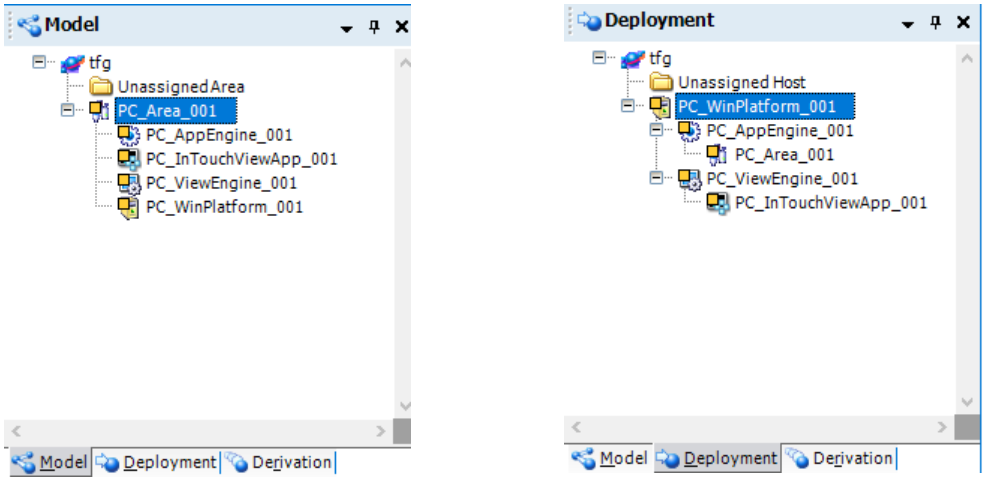


Figura 100. Configuración Model y Deployment

Para configurar la conexión del Galaxy creado, accedemos al árbol superior izquierdo a la carpeta ***"Device Integration"*** donde se creará una subplantilla (de igual forma que las anteriores) de la comunicación utilizada, en este caso OPC UA, como indica la Figura 101:

New -> Derived Template

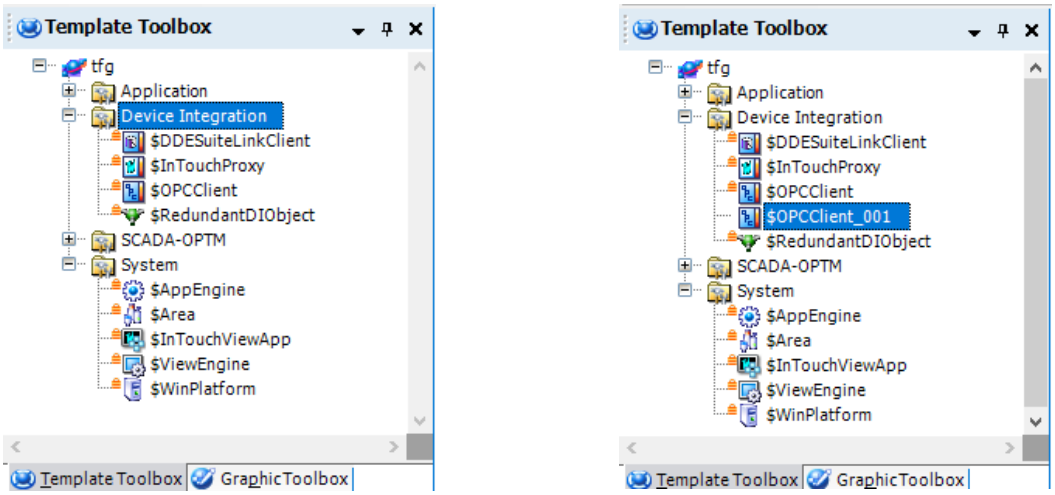


Figura 101. Creación Subplantilla para Conexión OPC UA - OPCClient

Cuando se haya creado se le dará el nombre del servidor con el que se va a conectar y abriendo la subplantilla se accede a la ventana de configuración de la comunicación:

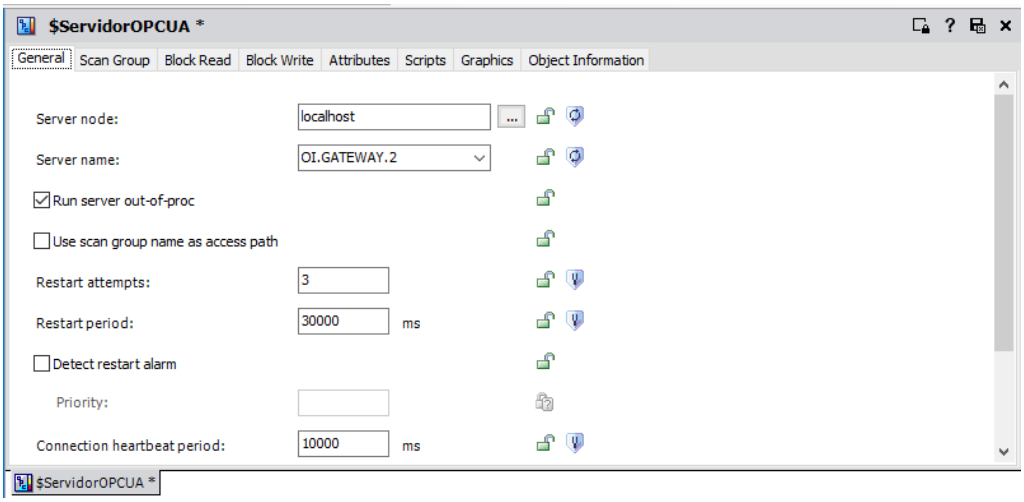


Figura 102. Configuración Conexión con Servidor OPC UA

En esta configuración se debe indicar donde se encuentra el Nodo que realizará la comunicación con el servidor OPC UA, para ello se debe indicar que el nodo se encuentra en el mismo ordenador (localhost) y que se utilizará el *OIGateway* para la conexión.

A continuación, se debe configurar la pestaña de ***"Scan Group"*** para administrar las variables que se considere procedentes del servidor OPC UA.

En esta pestaña accedemos al cliente creado en el programa *System Management Console* donde se encuentran las variables de interés del servidor OPC UA, para añadir dichas variables se configura la pestaña como la Figura 103:

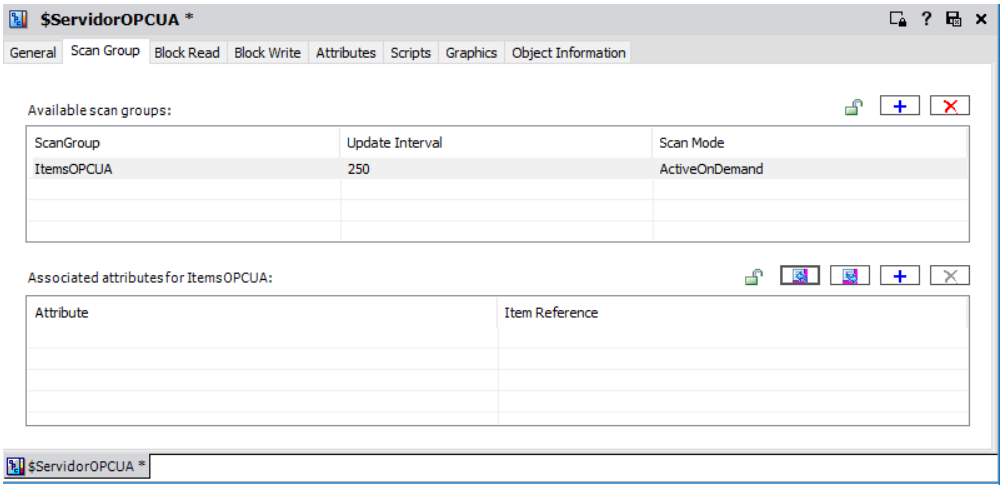


Figura 103. Creación del Grupo de Variables a utilizar

Cuando se crea un grupo de escaneo se puede acceder al Nodo configurado anteriormente, mostrándose la siguiente ventana:

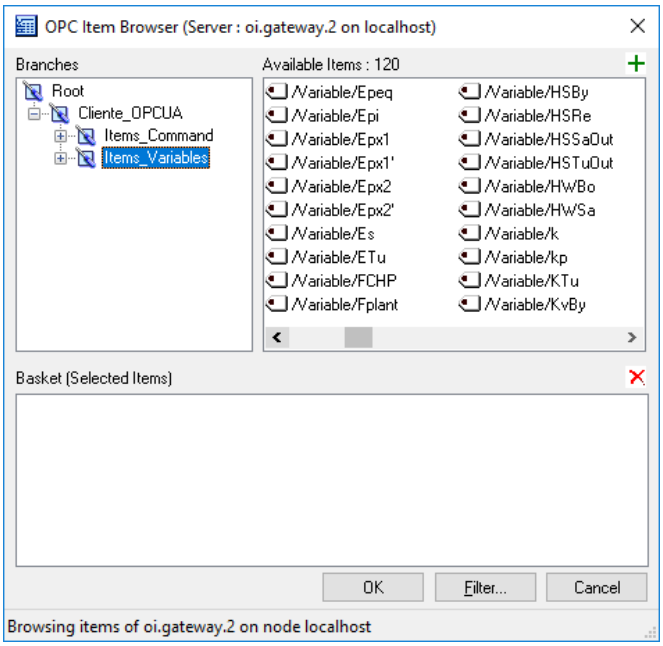


Figura 104. Elección de las Variables a Utilizar en el elemento Gráfico

Una vez seleccionadas las variables de interés, se guarda la configuración de la comunicación y se instancia dicha subplantilla quedando como muestra la Figura 105:

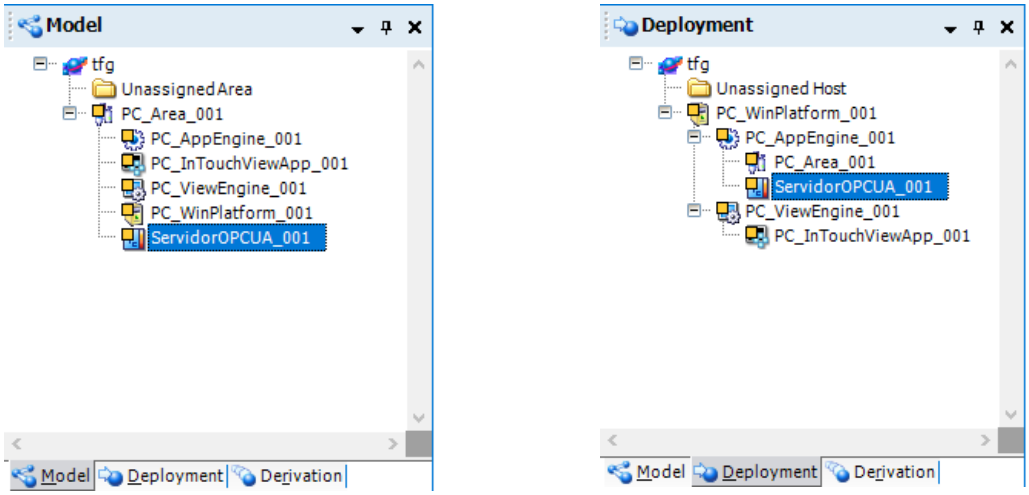


Figura 105. Conexión con el Servidor OPC UA

Realizando correctamente los pasos anteriores se tiene que desempaquetar la configuración realizada para que ArchestrA referencie a los programas involucrados en la configuración. Para ello se seleccionan las instancias que se indican en la Figura 106 en el espacio *Deployment* y con el botón derecho del ratón se selecciona *Deploy:*

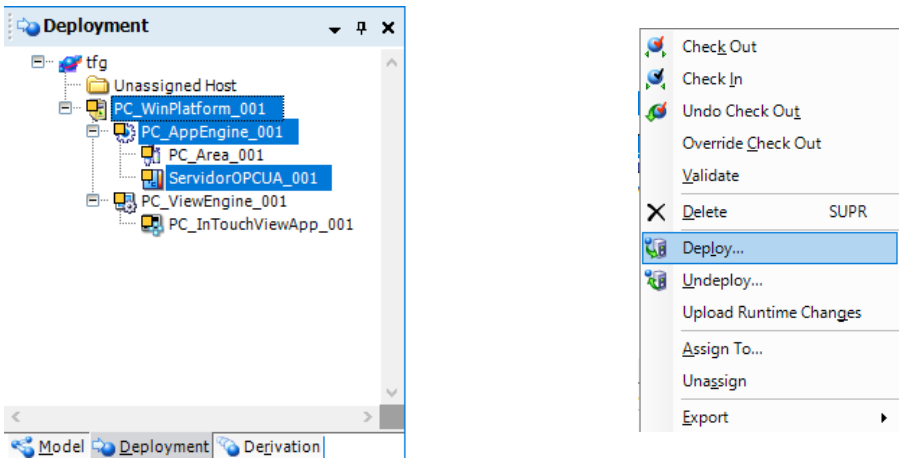


Figura 106. Sincronizar la configuración

Si todo se ha realizado correctamente la sincronización debe mostrar un mensaje de que la operación se ha realizado con éxito. Con esta configuración ya se tendría referenciada la conexión y comunicación con los demás software de Wonderware. A continuación, se comprueba la conexión realizada y se realiza la edición del elemento gráfico.

2.3. CONEXIÓN CON EL CLIENTE DE WONDERWARE (OBJECT VIEWER)

Para poder verificar la correcta conexión de *Application Server*, Wonderware cuenta con un cliente propio del sistema llamado *Object Viewer*. Este cliente realizará la conexión utilizando las instancias que se hayan realizado para la comunicación con los dispositivos correspondientes.

Para poder abrir el cliente, es necesario haber hecho un *Deployment* de la aplicación y que el resultado sea correcto, si se tuviera algún error el cliente no mostrará correctamente el acceso a la plantilla de comunicación. Pulsando el botón derecho del ratón sobre la plantilla de comunicaciones correspondiente en el espacio *Deployment* se tendrán las opciones para abrir el cliente, como muestra la Figura 107:

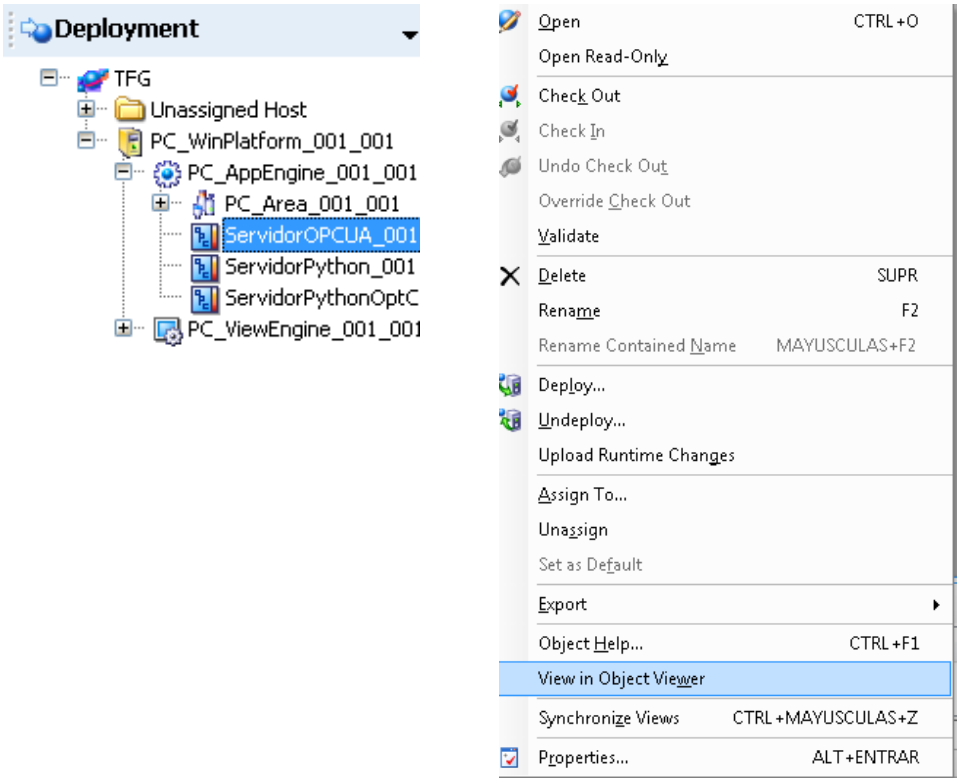


Figura 107. Apertura del cliente propio de Wonderware (Object Viewer)

Accediendo a la opción *View in Object Viewer* se abrirá dicho cliente, y se podrá empezar a monitorizar las variables que se hayan administrado en la instancia seleccionada. El aspecto de este cliente se muestra en la Figura 107, donde se pueden monitorizar las variables buscándolas en el espacio de direcciones o en el árbol que se presenta en la parte izquierda del programa. Una vez localizadas las variables que se desean monitorizar, se debe hacer doble clic en ellas para que se carguen en la ventana de monitorización donde se mostrará el valor de la variable, el tipo de variable y el estado de la conexión con dicha variables, además de otras opciones. Para poder localizar correctamente las variables, el nombre debe coincidir exactamente con el que se ha asignado en la plantilla de comunicaciones durante la administración de estas.

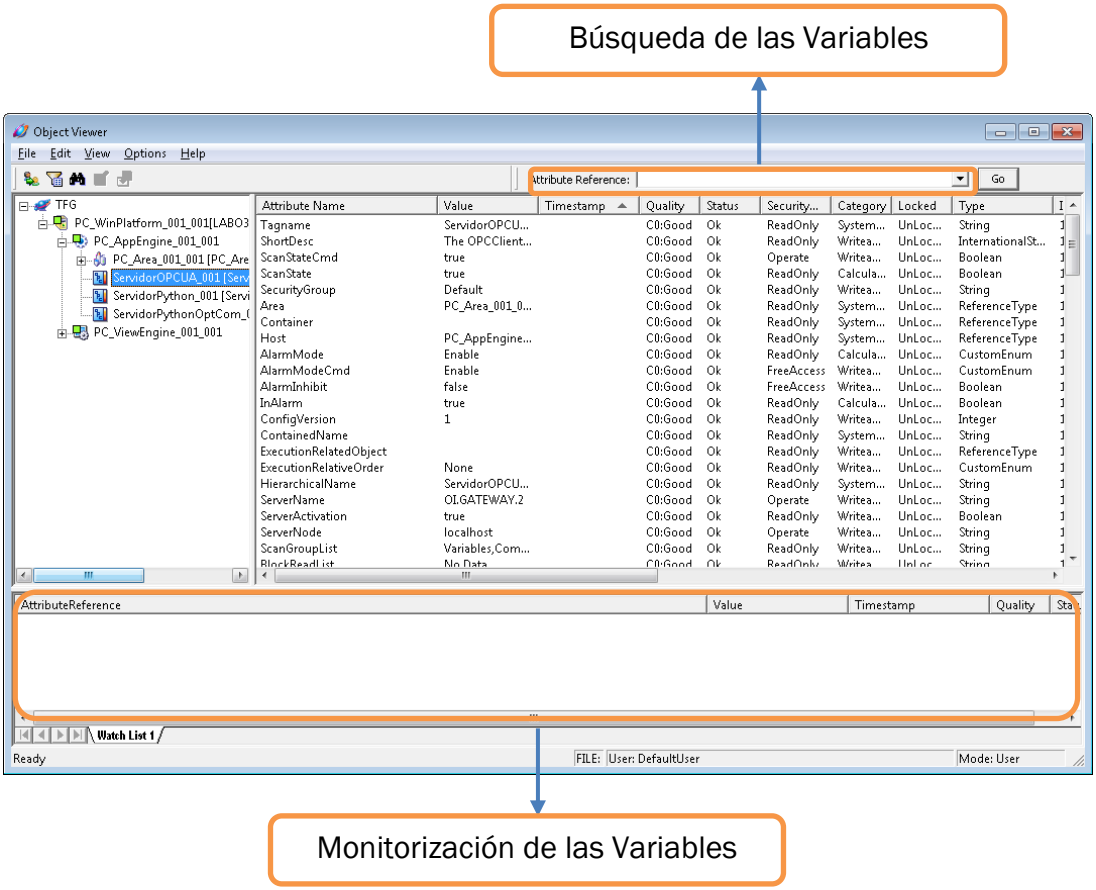


Figura 108. Monitorización de las variables

2.4. OBJETO GRÁFICO (TEMPLATE TOOLBOX)

Para crear un objeto gráfico definido por el usuario, se debe acceder en el árbol superior izquierdo a la carpeta ***"Aplicattion"*** y se mostraran las opciones de gráficos que se muestran a continuación:

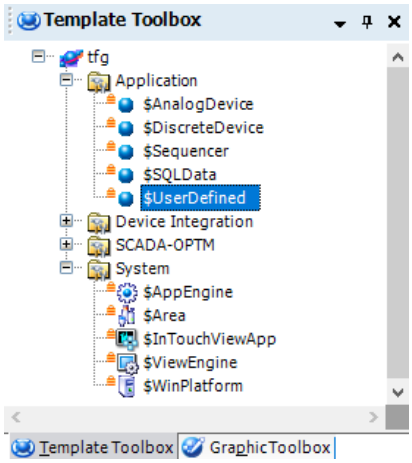


Figura 109. Creación de Elemento Gráfico

Para crear el gráfico, se debe

crear una subplantilla de igual forma que en la configuración anterior:

New -> Derived Template

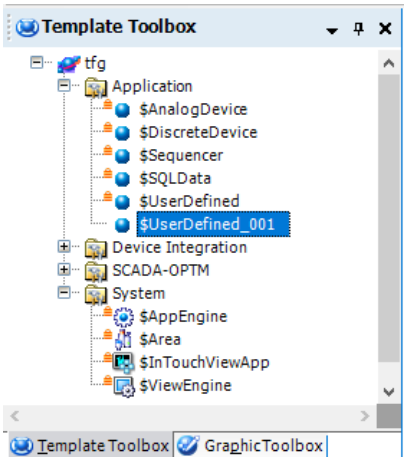


Figura 110. Creación de la subplantilla del Elemento Gráfico

Para configurar dicho objeto, se pincha dos veces sobre él y se mostrará una ventana donde se configuran todos los parámetros que tendrá el objeto.

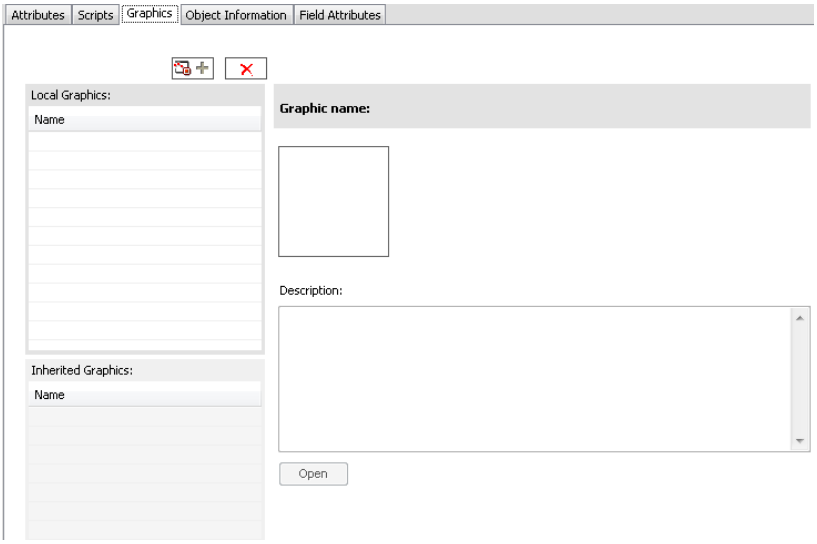


Figura 111. Ventana para la creación de gráficos

En este caso, la pestaña que interesa para la implementación de los gráficos es *Graphics*, donde se crearan tantos objetos gráficos como requiera el objeto creado. Para crear objetos gráficos, simplemente se debe acceder al botón de añadir, establecer un nombre para el objeto, y acceder a "ArchestrA Symbol Editor". Una vez realizada esta configuración el aspecto que se tendrá inicialmente para el editor gráfico será el que muestra la Figura 112

En este editor comenzará a diseñarse el gráfico que requiera el proceso,

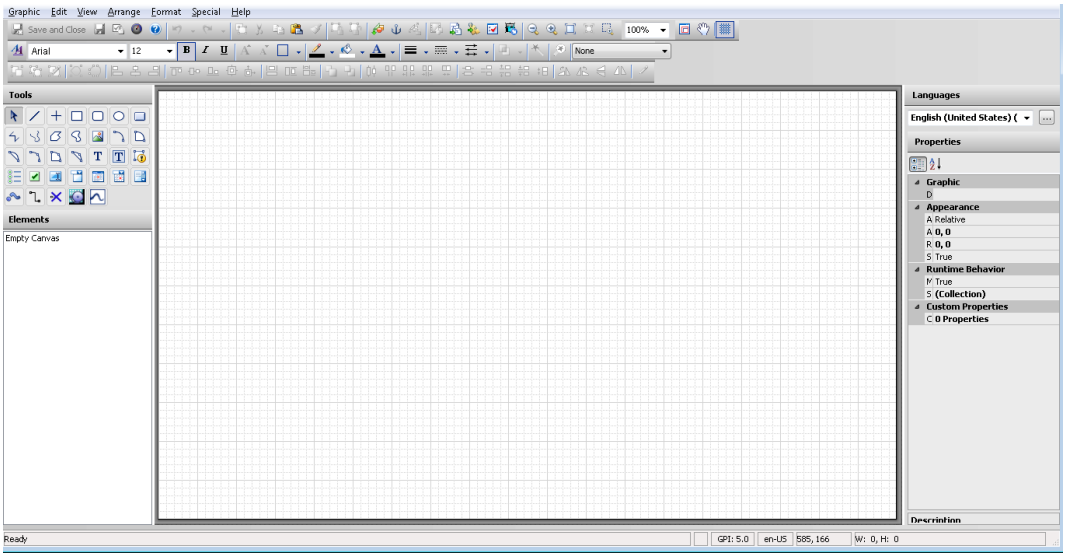


Figura 112. Ventana principal ArchestrA Symbol Editor

estableciendo las funcionalidades requeridas en cada caso. También se pueden añadir objetos gráficos de la librería que proporciona "ArchestrA" y modificarlos para que cumplan las necesidades que se desee.

ANEXO II: DESARROLLO DEL SERVIDOR OPC UA EN PYTHON

1. Implementación del Servidor OPC UA en Python

Para el desarrollo del servidor OPCUA de la optimización utilizando el lenguaje de programación Python, se ha utilizado la librería *freeOPCUA* con licencia LGPL que contiene los métodos necesarios para la implementación del servidor OPCUA de una forma intuitiva. Esta librería cuenta con una serie de ejemplos que facilitan la comprensión de los métodos a utilizar. Para realizar la implementación se han desarrollado tres pasos que se deben seguir para garantizar una correcta comunicación con dicho servidor.

- Paso 1: Creación del servidor.
- Paso 2: Definición de los nodos y variables
- Paso 3: Establecer permisos en las variables
- Paso 4: Bucle de ejecución

Paso 1: Creación del Servidor

La creación de un servidor OPCUA se debe a cuatro parámetros fundamentales:

Dirección URL: Esta dirección será la utilizada por el servidor para recibir y enviar los datos.

Política de Seguridad: Proporciona una encriptación en la transmisión de los datos recibidos y enviados por el servidor.

Nombre del Servidor: Asocia un nombre al servidor que se vaya a crear, para que pueda ser identificado por el cliente que vaya a conectarse a este.

Registrar Espacio de Direcciones: Se registra el espacio de direcciones del servidor, para asociar los nodos y variables correspondientes a este.

Para la implementación de estos parámetros, la librería contiene una serie de métodos que se encargarán de gestionarlos de una manera sencilla como se muestra en la Figura 113:

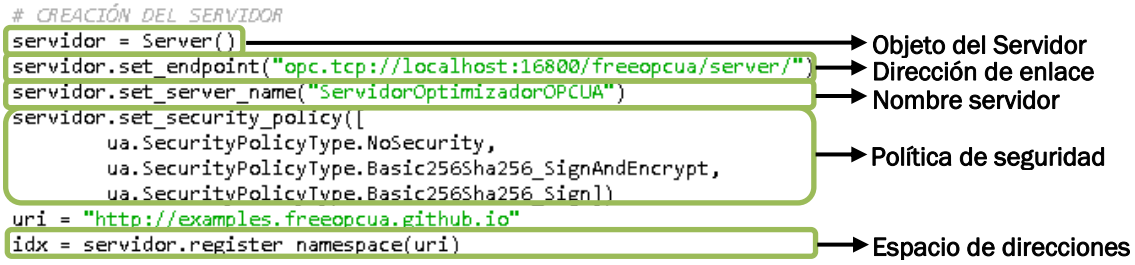


Figura 113. Código para la creación del servidor

Con esta parte del código el servidor quedaría registrado y definido para realizar la conexión con cualquier cliente OPCUA, en el siguiente paso se establecerán las variables que contendrá dicho servidor.

Paso 2: Definición de Nodos y Variables

La definición del contenido del servidor sigue un árbol de jerarquía donde se deben enlazar los nodos que se encargaran de administrar las variables del servidor. La forma de realizar esta jerarquía es la que se muestra en la Figura 114:

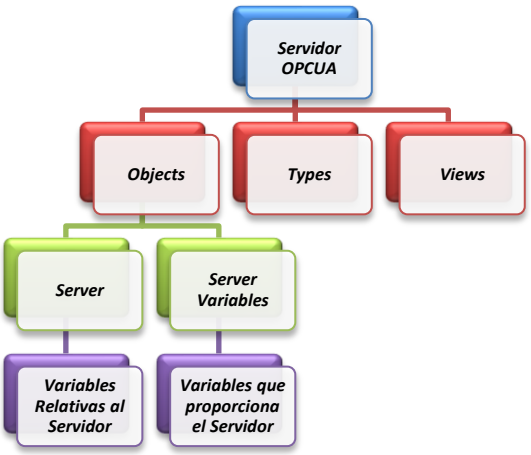


Figura 114. Organización interna del servidor OPC UA

Para realizar el cometido de este TFG, la implementación del servidor se realiza de manera que pueda funcionar correctamente con las características más básicas, no es objeto de esta aplicación profundizar en el contenido del servidor sino de establecer la comunicación entre el cliente y servidor OPC UA correctamente. Por ello esta implementación se centrará solamente en el nodo *Objects* que será el que contendrá las variables que utilizará para comunicar los datos al cliente que esté conectado a este. Implementando el siguiente fragmento de código que se muestra en la Figura 115, se consigue crear un nodo donde se almacenen las variables del servidor de la optimización y de las variables del modelo que se encargarán de proporcionar las consignas correspondientes al cálculo de la optimización.

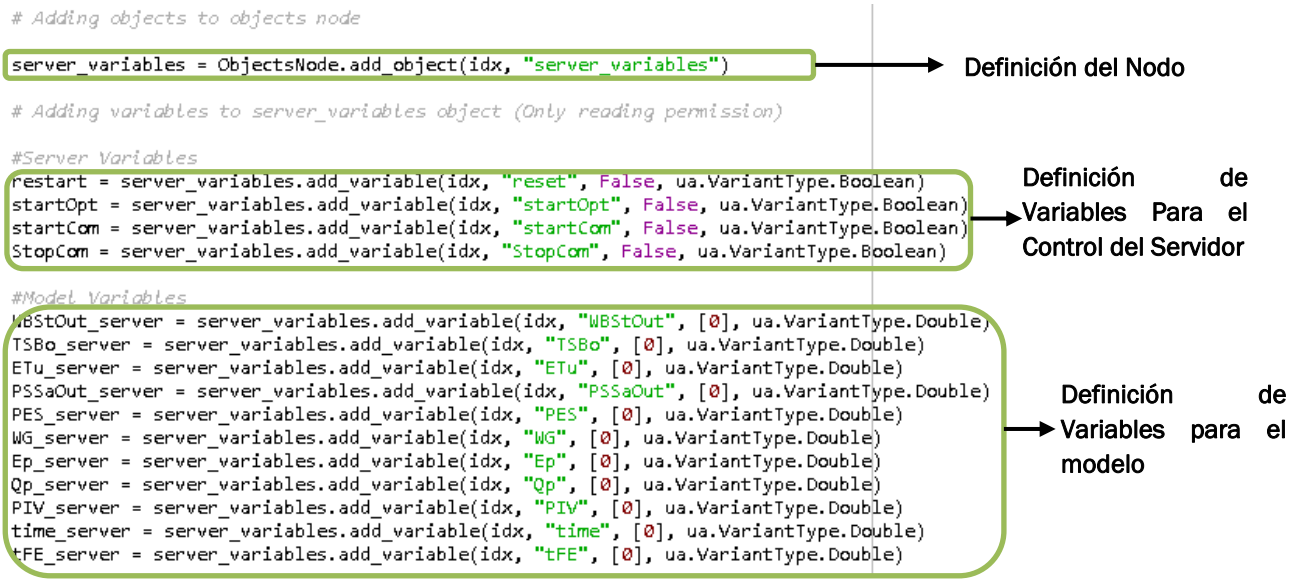


Figura 115. Código para definir el interior del servidor OPC UA

Con la implementación de este código se consigue definir las variables que el servidor va a emplear para transmitir los datos en su comunicación. Una vez definidas estas variables se debe establecer el permiso de escritura de estas.

Paso 3: Establecer permisos en las variables

Para que las variables puedan ser editadas por el cliente que se conecte al servidor, se deben establecer unos permisos que den lugar a la escritura de las variables que sea necesario. Para que una variable se le conceda solamente permisos de lectura, no es necesario llamar a ningún método, únicamente es necesario para las variables que se desee conceder permisos de escritura. Esta situación se representa en la Figura 116 que contiene el siguiente fragmento de código:

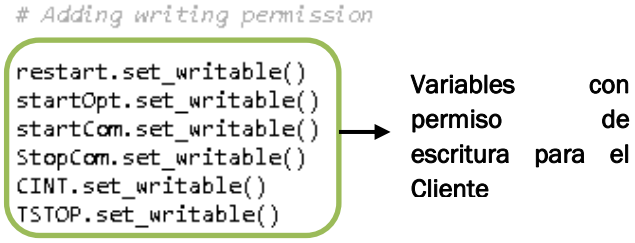


Figura 116. Código para dar permisos de escritura a las variables

Con esta parte del código se tendría definido el servidor con las necesidades básicas para establecer una correcta comunicación.

Paso 4: Bucle de Ejecución

Para que el servidor desarrolle una tarea y las variables vayan adquiriendo los valores de esa tarea, es necesario crear un bucle infinito que lo realice. En este bucle se implementará la tarea que se quiere desarrollar, que en este caso será el cálculo de la optimización.

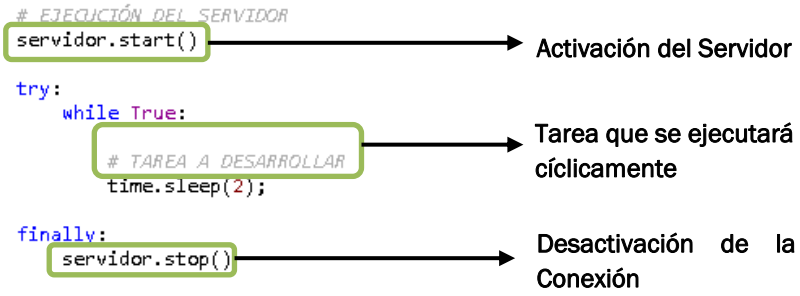


Figura 117. Código para la ejecución continua del servidor OPC UA

Finalmente añadiendo esta parte del código se tendría implementado correctamente un servidor OPCUA utilizando el lenguaje de programación Python.

ANEXO III: VARIABLES DEL PROCESO

1. Descripción de las variables

DENOMINACIÓN DE LAS VARIABLES UTILIZADAS		
Nombre de las Variables	*Descripción*	*Unidades*
ApBy	Grado de apertura de la válvula del bypass	%
ApRe	Grado de apertura de la válvula de recirculación	%
Ep	Energía eléctrica consumida por el proceso principal	kW
Es	Diferencia entre la energía consumida por el proceso principal y la energía generada en las turbinas	kW
ETu	Energía eléctrica generada en las turbinas	kW
mSt	Cantidad de remolacha acumulada en la zona de almacenaje	kg
muG	Índice global de cogeneración	%
PES	Índice de los servicios de energía primarios	%
PIV	Presión de vapor en el cuarto efecto de la evaporación	barA
PSBo	Presión de vapor en la salida de la caldera	barA
PSSaOut	Presión de vapor en la salida del saturador	barA
PSSaOutRef	Presión de vapor de referencia en la salida del saturador	barA
Qp	Consumo de energía térmica de la fábrica de azúcar	kW
TSBo	Temperatura de vapor sobrecalentado en las calderas	ºC
WBStIn	Tasa de llegada de remolacha	T/h
WBStOut	Tasa de producción de remolacha	T/h
Wng	Caudal másico de gas natural necesario para el proceso	kg/s
WSBo	Caudal másico de vapor en la caldera	kg/s
WSBy	Caudal másico de vapor en el bypass	kg/s
WSRe	Caudal másico de vapor en la recirculación	kg/s
WSTuIn	Caudal másico de vapor en la entrada de las turbinas	kg/s
WWSa	Caudal másico de agua en el saturador	kg/s

Tabla 14. Listado de las variables del proceso mostrado en la interfaz